



HAL
open science

Enhancing routing protocols in dense nanonetworks

Carole Al Mawla

► **To cite this version:**

Carole Al Mawla. Enhancing routing protocols in dense nanonetworks. Networking and Internet Architecture [cs.NI]. Université Bourgogne Franche-Comté, 2024. English. NNT : 2024UBFCD033 . tel-04903264

HAL Id: tel-04903264

<https://theses.hal.science/tel-04903264v1>

Submitted on 21 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

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

CAROLE AL MAWLA

Enhancing routing protocols in dense nanonetworks

Amélioration des protocoles de routage dans les nanoréseaux denses

Thèse présentée et soutenue à Montbéliard, le 29 Novembre 2024

Composition du Jury :

RAMDANE-CHERIF AMAR	Professeur des universités, Univ. Paris Saclay	Président du jury
LORENZ PASCAL	Professeur des universités, Univ. Haute-Alsace	Rapporteur
RONDEAU ÉRIC	Professeur des universités, Univ. Lorraine	Rapporteur
DEDU EUGEN	Maître de conférences HDR, Univ. Franche-Comté	Directeur de thèse
DHOUTAUT DOMINIQUE	Maître de conférences, Univ. Franche-Comté	Codirecteur de thèse
BEYDOUN KAMAL	Maître de conférences, Université Libanaise	Codirecteur de thèse

ACKNOWLEDGEMENTS

I extend my heartfelt gratitude to Professor Eugen Dedu, my primary research supervisor, for his unwavering support throughout my Ph.D. journey. His guidance, consistent attention, and valuable insights have been a constant source of motivation over the past years. I am deeply indebted to Professor Dedu for his pivotal role in the successful completion of my thesis.

I would also like to express my sincere appreciation to my co-supervisors, Dr. Kamal Beydoun and Dr. Dominique Dhoutaut, for their invaluable contributions to my research. Dr. Beydoun's insightful perspectives and encouragement have been instrumental in shaping my work. I am grateful to Dr. Dhoutaut for his assistance and clarity in various aspects of my thesis. The collaborative efforts of both have significantly enhanced the quality of my research, and I am truly thankful for their guidance and support.

My sincere thanks also go to the members of the committee for their active participation in my Ph.D. defense.

Finally, I want to dedicate this work and express my deepest love and gratitude to my family, especially my father, **Abdullah Fawzi Al Mawla**, and my mother, **Zaynab Ballout**, for their unwavering encouragement, support, and confidence throughout my academic journey. Their steadfast presence and belief in me have been a tremendous source of strength and inspiration.

CONTENTS

I	Context	1
1	Introduction	3
1.1	Scientific and institutional context	3
1.2	Contribution to nanonetworks	5
1.3	Summary in French	6
1.4	Thesis outline	9
2	Background	11
2.1	Nanonetworks	11
2.1.1	Nanotechnology and nanonetwork	11
2.1.2	Nanonetwork applications	13
2.2	TS-OOK modulation	15
2.3	SLR (Stateless Linear-path Routing) protocol	17
2.4	Nanonetwork simulators	18
2.4.1	Introduction	18
2.4.2	Simulator desired features	19
2.4.3	Other simulators	20
2.4.3.1	Nano-Sim	21
2.4.3.2	TeraSim	22
2.4.3.3	Vouivre	22
2.4.4	BitSimulator	23
2.4.4.1	Introduction	23
2.4.4.2	User's guide	25
2.4.4.3	BitSimulator output and VisualTracer visualisation tool	29

II	Contribution	31
3	A classification of single-channel communication concurrency	33
3.1	Introduction	33
3.1.1	Motivations	34
3.1.2	Contributions	34
3.1.3	Scope of the chapter	35
3.2	Classification of concurrent communications	36
3.2.1	Flow-level concurrency	36
3.2.2	Packet-level concurrency	37
3.2.3	Bit / symbol-level concurrency	38
3.2.4	Pulse-level concurrency	42
3.2.4.1	Concurrency using pulse injection	43
3.2.4.2	Concurrency using pulse shape changing	44
3.2.5	Summary	44
3.2.6	Amount of concurrency	45
3.3	Conclusions	46
4	EIDA, a best effort equitable distributed id assignment mechanism	47
4.1	Introduction	48
4.2	Related work	49
4.2.1	Random assignment	49
4.2.2	Ideal assignment	50
4.2.3	Other assignment methods	50
4.3	EIDA in one-hop networks	52
4.3.1	Formalization	52
4.3.2	EIDA functioning	54
4.3.3	Memory and energy considerations	56
4.4	Application to multi-hop networks	56
4.4.1	Applications	56
4.4.2	Network division in zones using SLR	57

4.4.3	Zone density computation	57
4.4.4	EIDA algorithm in multi-hop network	58
4.5	Evaluation	59
4.5.1	EIDA results	59
4.5.2	Method comparison	60
4.6	Conclusion	62
5	FR-SLR, forwarder reduction in SLR routing protocol through zone splitting	63
5.1	Introduction	63
5.2	Related work	64
5.2.1	Cluster-based approaches	64
5.2.2	Flooding protocols	65
5.2.3	Destination-oriented protocols	66
5.2.4	Sleeping-based approaches	66
5.3	Forwarder reduction in SLR routing (FR-SLR)	67
5.3.1	FR-SLR description	67
5.3.2	Integration of FR-SLR with existing id assignment mechanisms	68
5.3.2.1	Random id assignment	68
5.3.2.2	Ideal id assignment	69
5.3.2.3	EIDA assignment mechanism	69
5.4	Evaluation	70
5.4.1	Die-out avoidance in FR-SLR with EIDA	71
5.4.2	Comparison of FR-SLR/EIDA to other routing protocols	71
5.4.3	Discussion on FR-SLR vs counter-based SLR	72
5.5	Conclusion	73
6	Network parameter influence on communications in dense wireless nanonet-	75
	works	
6.1	Introduction	75
6.2	Related work	76
6.3	Network parameter influence on communications	77

6.3.1	Base network scenario used for simulations	77
6.3.2	Impact of β	78
6.3.3	Impact of the communication range	81
6.3.4	Impact of the pulse duration	83
6.3.5	Impact of node density	83
6.4	Conclusion	84
III	Conclusion	85
7	Conclusion and future work	87
IV	Appendix	105
A	Reproducibility information	107
A.1	EIDA mechanism	107
A.2	FR-SLR protocol	108
A.3	Network parameter influence	111

|

CONTEXT

INTRODUCTION

1.1/ SCIENTIFIC AND INSTITUTIONAL CONTEXT

I started my thesis on January 1, 2021, at the FEMTO-ST Institute (DISC department within the OMNI team) and University of Bourgogne Franche-Comté, France. I was under the supervision of Prof. Eugen Dedu, Dr. Dominique Dhoutaut, and Dr. Kamal Beydoun.

For most of the time, my Ph.D. was pursued remotely from the L'Arکید laboratory at the Lebanese university, Lebanon. A significant 73-day staying at the FEMTO-ST laboratory at Montbéliard, France, under the direct supervision of Prof. Eugen Dedu, provided a valuable experience, enhancing my research perspective.

My research, which concluded in November 2024, centered on designing and developing routing protocols in dense nanonetworks, necessitated by the unique challenges posed by nanonetwork density and scale, a key research interest of the OMNI team.

First, my research delved into a fundamental concept in network communication: the levels and mechanisms of communication concurrency (pulse, bit, packet, flow). This exploration aimed to provide a comprehensive toolbox for future research and for developing of new routing protocols that effectively leverage concurrency. These insights are important in advancing network efficiency and performance, thus significantly contributing to the field of nanonetwork communications.

Afterward, as shown on figure 1.1, I focused on improving routing protocols in dense nanonetworks, particularly through the method of traffic splitting. A significant part of my work involved developing a new mechanism for assigning IDs in these networks. This ID assignment process is crucial for managing dense nanonetworks and plays a key role in enhancing routing methods.

Finally, my research involved investigating the impact of network parameters—such as β , communication range, pulse duration, and node density—on network metrics like collision, reception, emission, and delivery rates. These insights provide guidelines for se-

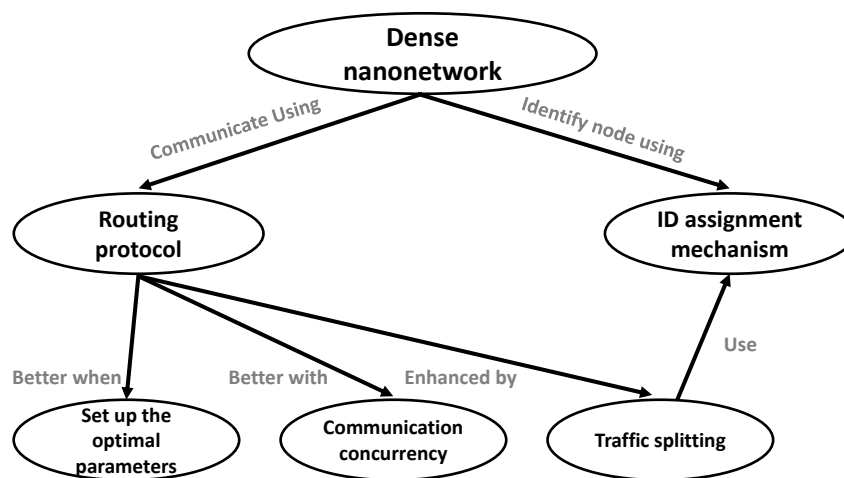


Figure 1.1: Context overview.

lecting appropriate network parameters.

During the PhD, I co-authored the following articles:

Publications in [conferences](#):

1. Carole Al Mawla, Eugen Dedu, Kamal Beydoun, and Dominique Dhoutaut, "[EIDA, a Best Effort Equitable Distributed id Assignment Mechanism for Heterogeneous Dense Nanonetworks](#)", in the 10th ACM International Conference on Nanoscale Computing and Communication (NanoCom), UK, pp. 33–39, Sep. 2023, IEEE.
2. Carole Al Mawla, Eugen Dedu, Kamal Beydoun, and Dominique Dhoutaut, "[FR-SLR, forwarder reduction in SLR routing protocol through zone splitting](#)", in 2023 International Wireless Communications and Mobile Computing (IWCMC), Marrakesh, Morocco, pp. 1316–1321, June 2023, IEEE, CORE B.
3. Carole Al Mawla and Eugen Dedu, "[Network Parameter Influence on Communications in Dense Wireless Nanonetworks](#)", in the 10th ACM International Conference on Nanoscale Computing and Communication (NanoCom), Milan, Italy, Oct. 2024, 6 pages

Submission to [journal](#):

1. Carole Al Mawla and Eugen Dedu, "[A classification of single-channel communication concurrency](#)", in Nano Communication Networks, 8 pages.

1.2/ CONTRIBUTION TO NANONETWORKS

Dense nanonetworks (detailed in section 2.1.1) consist of a huge number of very small-scale devices (nano-devices) connected in a network. These networks present unique challenges and opportunities due to their scale and the capabilities of individual nano-devices.

Our contribution to nanonetwork in this thesis involves studying, analyzing, and developing novel algorithms specifically designed for dense nanonetworks, focusing on the network layer. The work is structured into four main components:

- A classification of single-channel communication concurrency.
- EIDA (a Best Effort **E**quitable Distributed **id** Assignment Mechanism).
- FR-SLR (**F**orwarder **R**eduction in **SLR** routing protocol).
- Network parameters influence on communication in dense wireless nanonetworks.

The network layer of the OSI (Open Systems Interconnection) model, also known as layer 3, is responsible for packet routing. It handles routing and forwarding of data packets between two parties by determining the best path for data to travel from the source to the destination. This layer is crucial due to the unique challenges posed by the nanoscale environment, such as limited energy and computational capabilities. Efficient routing should conserve energy and prolong the operational lifetime of these devices. Optimizing routing protocols for nanonetworks is a must to enhance reliability and reduce latency.

Using **concurrency** in routing protocol enhances the overall quality of network communication. This realization leads to our first contribution (presented in chapter 3), which provides valuable insights to protocol creators and users. We review the mechanisms that provide concurrency, and identify and detail the various levels of communication concurrency, including pulse, bit, packet, and flow levels.

Our second contribution, **FR-SLR** protocol (presented in chapter 5), is a routing protocol based on the SLR routing protocol (presented in section 2.3). FR-SLR aims to reduce the number of packets forwarded along the path between the sender and receiver. This protocol uses an id assignment mechanism to divide nodes in groups, selecting only one group of nodes to forward packets in each transmission.

The challenges of assigning ids in a dense nanonetwork and the id assignment mechanism needed in FR-SLR lead to our third contribution, **EIDA** (presented in chapter 4). EIDA is a new, configurable, and distributed id assignment mechanism that assigns each group of nodes a specific id. This mechanism combines both random and ideal assignments, overcoming the limitations of using only one method. By leveraging the strengths

of both approaches, EIDA ensures a more equitable distribution of ids than purely random assignment and requires less packet exchanges than the ideal assignment.

After implementing and testing our first three contributions, we began to consider how to further improve a routing protocols. Is it enough to have a good protocol, or are there other factors we need to consider? This question led us to our fourth contribution (presented in chapter 6). This contribution provides guidelines for selecting appropriate **network parameters** by evaluating how these parameters impact communication quality in a dense nanonetwork. The parameters studied include β , communication range, node density, and pulse duration in both homogeneous and heterogeneous nanonetworks. We focus on their effect on network communication quality, such as the number of packets collided, received, emitted, and delivered. These guidelines help in fine-tuning network parameters to achieve optimal performance.

1.3/ SUMMARY IN FRENCH

J'ai commencé ma thèse en janvier 2021 à l'Institut FEMTO-ST (département DISC, au sein de l'équipe OMNI), à l'Université de Franche-Comté, en France. J'étais sous la supervision du Prof. Eugen Dedu, du Dr. Dominique Dhoutaut et du Dr. Kamal Beydoun. Pendant la majeure partie de mon doctorat, j'ai travaillé à distance depuis le laboratoire L'Aricod à l'Université Libanaise. Un séjour important de 73 jours au laboratoire FEMTO-ST à Montbéliard, en France, sous la supervision directe du Prof. Eugen Dedu, m'a offert une expérience précieuse, enrichissant ainsi ma perspective de recherche.

Les nanoréseaux denses multi-sauts se composent d'un nombre incroyablement élevé de nano-nœuds de connectivité de petite taille (moins de 10 μm), potentiellement de milliers à des millions, avec de nombreux voisins engagés dans la communication et l'interaction. Ces nanodispositifs sont capables d'effectuer uniquement des tâches très simples telles que le calcul, le stockage de données, la détection et l'actionnement. Ils peuvent être minuscules, mais ils sont suffisamment intelligents pour accomplir des tâches importantes, comme le traitement de l'information, la détection de leur environnement et la prise de mesures en fonction de ce qu'ils détectent. Bien que chaque nanodispositif ne puisse accomplir que des tâches basiques, ils peuvent réaliser des tâches remarquablement complexes lorsqu'ils collaborent au sein d'un réseau.

Dans ma recherche doctorale, je me suis concentrée sur l'amélioration des protocoles de routage dans les nanoréseaux denses, notamment par la méthode du partage de trafic. FR-SLR est un protocole de routage que je propose, basé sur le protocole de routage SLR. SLR (Stateless Linear-path Routing) est un protocole de routage conçu pour les nanoréseaux denses et proposé en 2017. Il s'agit d'un protocole d'adressage

spatial et de routage unicast où les messages sont envoyés à une seule destination. SLR comprend deux phases : la phase d'initialisation/adressage et la phase de routage. L'objectif de la phase d'initialisation est de diviser le réseau en zones et de leur attribuer des coordonnées. Ces coordonnées sont définies comme un nombre entier de sauts à partir de quelques nœuds spéciaux appelés ancres. L'objectif de la phase de routage est de diriger les paquets de données sur un chemin de routage linéaire basé sur les coordonnées attribuées lors de la phase d'initialisation. Dans SLR, tous les nœuds d'une zone participent ensemble au processus de routage, ce qui entraîne une consommation inutile de ressources.

FR-SLR vise à réduire le nombre de paquets transmis le long du chemin entre l'émetteur et le récepteur. Ce protocole utilise un mécanisme d'attribution d'identifiants (id) pour diviser les nœuds d'une zone en plusieurs groupes afin que le routage soit effectué par groupe plutôt que par zone, c'est-à-dire en sélectionnant seulement un groupe de nœuds pour retransmettre les paquets dans chaque zone du chemin de transmission au lieu de tous les nœuds. Dans FR-SLR, tous les nœuds d'un groupe partagent le même id, et les groupes ont des ids différents. Seuls les nœuds sur le chemin de transmission dont l'id correspond à celui indiqué dans le paquet retransmettent le paquet (au lieu de tous les nœuds, comme dans SLR).

Mon travail a également porté sur le développement d'un nouveau mécanisme d'attribution d'identifiants dans les nanoréseaux denses. Ce processus d'attribution d'id permet de gérer ces réseaux et joue un rôle clé dans l'amélioration des méthodes de routage. Ma proposition, EIDA, est un mécanisme d'attribution d'id distribué, configurable, qui attribue à chaque groupe de nœuds un id spécifique. Il combine à la fois une attribution aléatoire (où chaque nœud choisit un id aléatoire sans échange de paquets) et une attribution idéale (où les nœuds attribuent alternativement un id unique entre 0 et $n-1$ en échangeant n paquets). En tirant parti des forces des deux approches, EIDA surmonte les limites de l'utilisation d'une seule méthode, assurant une répartition plus équitable des ids que l'attribution purement aléatoire et nécessitant moins d'échanges de paquets qu'une attribution idéale. EIDA fonctionne en permettant à un sous-ensemble de nœuds d'attribuer leurs ids par échange de paquets jusqu'à ce qu'un niveau prédéterminé d'équité soit atteint. Au-delà de ce point, le mécanisme passe à une attribution d'id aléatoire pour les nœuds restants, minimisant ainsi la communication supplémentaire. Cette stratégie vise à équilibrer les méthodes d'attribution idéale et aléatoire, en recherchant le meilleur compromis pour une répartition équitable des ids. Ce mécanisme utilise deux paramètres choisis par l'utilisateur en fonction des besoins de l'application. La redondance représente le nombre de nœuds attendu (souhaité) dans un groupe, et la garantie représente le nombre minimum de nœuds assuré dans chaque groupe (par exemple, nous avons besoin d'au moins 2 nœuds par groupe).

De plus, mes recherches ont exploré un concept fondamental de la communication en réseau : les niveaux et mécanismes de concurrence en communication. Cette exploration visait à fournir une boîte à outils complète pour la recherche future et pour le développement de nouveaux protocoles de routage tirant efficacement parti de la concurrence. Les niveaux de concurrence incluent le niveau des impulsions (présence ou absence d'impulsions discrètes à des moments spécifiques), le niveau des bits (plusieurs bits sont transmis simultanément dans un canal), le niveau des paquets (le canal ayant plusieurs paquets en même temps, mais sans bits simultanés), et le niveau des flux (une transmission significative de bout en bout entre deux pairs). Parmi les mécanismes explorés, on trouve l'entrelacement de bits au niveau des paquets (les paquets sont dans le canal en même temps, mais aucun bit n'est simultanément présent), l'effet de capture (parmi plusieurs signaux, seul le plus fort peut être correctement démodulé par un récepteur) et l'interférence constructive (plusieurs nœuds transmettent simultanément le même paquet, ce qui entraîne l'addition constructive de leurs signaux au niveau du récepteur), au niveau des bits, etc.

En outre, mes recherches ont porté sur l'étude de l'impact des paramètres réseau sur les métriques de réseau. Nous fournissons des recommandations pour choisir les paramètres réseau appropriés en évaluant leur impact sur la qualité de communication dans un nanoréseau dense. Les paramètres étudiés incluent β (le taux de symboles, rapport du temps entre deux impulsions consécutives et la durée de l'impulsion), la portée de communication, la densité des nœuds et la durée des impulsions dans les nanoréseaux homogènes et hétérogènes. Nous nous concentrons sur leur effet sur la qualité de la communication réseau, telle que le nombre de paquets en collision, reçus, émis et livrés. Nos résultats montrent qu'une augmentation de β (jusqu'à un certain point) réduit les collisions de paquets tout en augmentant le nombre de réceptions, d'émissions et de livraisons ; une augmentation de β au-delà de ce point n'a pas d'effet significatif. L'augmentation de la portée de communication ou de la densité des nœuds augmente les collisions, les réceptions, les émissions et les livraisons. L'augmentation de la durée des impulsions réduit le nombre de collisions de paquets tout en augmentant le nombre de paquets reçus et émis (jusqu'à un certain point). Ces recommandations aident à ajuster les paramètres réseau pour atteindre des performances optimales.

En conclusion, comme illustré dans la figure 1.1, ma thèse se concentre sur l'amélioration des protocoles de routage pour les nanoréseaux denses via le partage de trafic (FR-SLR). Cette amélioration utilise un mécanisme d'attribution d'identifiants (EIDA) qui garantit la fiabilité. De plus, un protocole de routage s'améliore en définissant des paramètres optimaux et en exploitant la concurrence.

1.4/ THESIS OUTLINE

This thesis is organized to provide a clear view of the research contributions and innovations in the field of nanonetworks. It is structured into three main chapters that guide the reader through the context, contributions, and conclusions of the research.

Chapter I: Context sets the stage for the research. It begins with an introduction that establishes the scientific and institutional context of the thesis, highlighting the relevance and importance of nanonetworks in current research and technology. This is followed by a discussion of the advancements and innovations introduced by the research, detailing how the thesis contributes to the field of nanonetworks. The chapter concludes with an outline of the thesis structure, providing a roadmap for the subsequent sections.

It also delves into the essential context necessary to understand the research contributions. It starts with a comprehensive overview of nanonetworks, covering the fundamental concepts of nanotechnology and its applications. The chapter then addresses TS-OOK modulation, an important technique used in the study, and explains Stateless Linear-path Routing (SLR), a key protocol relevant to the research. Additionally, it reviews various nanonetwork simulators, including Nano-Sim, TeraSim, Vouivre, and BitSimulator, offering insight into their features, functionalities, and their relevance to the study.

Chapter II: Contribution details the primary contributions of the thesis.

- It presents a classification of single-channel communication concurrency, providing a structured approach to understanding concurrency in nanonetworks communication. This section reviews concurrency mechanisms and classifies the levels of communication concurrency (pulse, bit, packet, and flow levels) along with examples of network technologies and protocols that support each level.
- It introduces the EIDA (Equitable ID Assignment) mechanism, describing its design and implementation in both one-hop and multi-hop network scenarios. This section elaborates on the formalization of EIDA, its functioning, and its evaluation and compares it to other ID assignment methods.
- It presents the FR-SLR (Forwarder Reduction in SLR) protocol, explaining how it reduces the number of forwarders in SLR routing through zone splitting. This section examines the integration of FR-SLR with existing ID assignment mechanisms, evaluates its effectiveness, and compares it with other routing protocols.
- It investigates the influence of network parameters on communication in dense wireless nanonetworks. It analyzes how different parameters (such as β , communication range, node density, and pulse duration) affect various network metrics, including packet collisions, receptions, emissions, and deliveries to destination.

Chapter III: Conclusion summarizes the key findings of the thesis, the research contributions, and discusses potential directions for future work. It provides a comprehensive overview of the main results, their significance within the field, and their potential impact on future developments.

Additionally, the Appendix in chapter IV addresses the reproducibility of the research, ensuring that the findings can be reliably replicated by others in the field.

BACKGROUND

As I started exploring the vast field of nanotechnology, my first step was to immerse myself in reading. By studying many published articles, I began to build the knowledge needed for my research.

Traditional communication methods prove ineffective in dense nanonetwork, necessitating specialized approaches like Time Spread On-Off Keying (TS-OOK) modulation and Stateless Linear-path Routing protocol (SLR).

In the following sections, I will provide an in-depth examination of these advanced communication techniques, the concept and applications of nanonetworks, and present nanonetwork simulators. Specifically, I will discuss how BitSimulator, the simulation tool employed in this research, contributes to evaluating and validating the proposed solutions.

2.1/ NANONETWORKS

2.1.1/ NANOTECHNOLOGY AND NANONETWORK

Nanonetworks Lemic et al. (2021) are all about small-scale connectivity, operating at the tiny level of nanometers (less than 10 μm in dimension). The “nano” in nanonetworks refers to this super small scale. These networks are made up of extremely small devices called nanodevices, which are just a few nanometers in size. Think of them like miniature machines that can talk to each other.

Nanotechnology got its start in the 1980s thanks to some big discoveries, such as the scanning tunneling microscope in 1981 and fullerenes in 1985. These findings got scientists and inventors excited about the possibilities of tiny technology. However, it wasn't until the early 2000s that we began to see nanotechnology making its way into everyday items. That's when companies started using it to create products that we use in our daily lives.

These **nanodevices**, also known as nanonodes, are capable of performing only very sim-

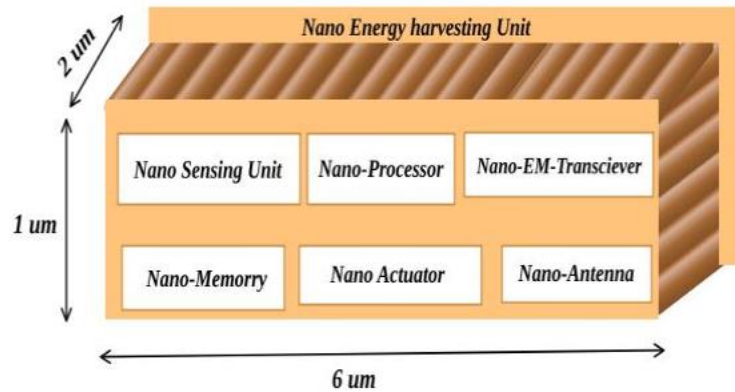


Figure 2.1: Nanosensor architecture Oukhatar et al. (2021).

ple tasks such as computing, data storage, sensing, and actuation. They might be tiny, but they're smart enough to handle important jobs like processing information, sensing their environment, and taking action based on what they detect (don't underestimate them!). Even though each nanodevice can only handle basic tasks, when they collaborate within a network, they can accomplish remarkably complex tasks.

For instance, consider a **nanosensor architecture** Oukhatar et al. (2021), depicted in Figure 2.1, comprising nano-processors, nano-memory, nano-batteries, nano-transceivers, nano-antennas, nano-actuators, and nano-sensing unit. Nano-processors execute computational tasks, while nano-sensors play a pivotal role in detecting novel events at the nanoscale, such as identifying viruses and bacteria within biological samples. Nano-memory, serving as the storage unit of the nanonode, is constrained in size. Nano-antennas facilitate communication between nano-devices. Nano-transceivers convert the recognition of nanoscale events into a measurable physical phenomenon, such as alterations in electrical resistance. Lastly, nano-actuator enables the nano-sensors to interact with their surrounding events.

In nanonetworks, the **network layer** is crucial for ensuring data transmission between THz ($1 \text{ THz} = 10^{12} \text{ Hz}$) nanonodes, regardless of their distance apart. This involves utilizing intermediate nanonodes, known as hops, to relay information and maintain successful communication. The forwarding functionality, available to either all nanonodes or a subset, aids in the transmission process. Additionally, routing functionality is essential for directing data through the hopping process to the correct destinations. It ensures that the paths between the data's origin and its intended destination are effective. However, routing protocols in nanonetworks face unique challenges, such as limited energy and unpredictable conditions. This necessitates the development of innovative approaches to guarantee reliable and efficient message delivery. Given these challenges, traditional communication methods must be rethought for the nanoscale.

Electromagnetic and molecular communication are the primary options for nanonetwork communication. **Electromagnetic communication** in nanonetworks involves the transmission of electromagnetic signals, akin to traditional wireless communication but adapted for the nanoscale. Nanodevices utilize electromagnetic waves, such as radio or light waves, to convey data between nodes, offering relatively high data rates and compatibility with existing infrastructure. In contrast, **molecular communication** relies on the exchange of signaling molecules between nanonodes, conveying information through concentration, timing, or type of molecules released. While molecular communication is well-suited for nanoscale applications due to compatibility with biological systems, it may entail slower data rates compared to electromagnetic methods.

2.1.2/ NANONETWORK APPLICATIONS

The importance of nanonetworks lies in their potential to revolutionize various fields, like healthcare and environmental monitoring, construction engineering, etc., by enabling communication and data exchange at the nanoscale. Imagine tiny sensors that can travel through your body to check for health problems, or tiny machines that can clean up pollution in the environment. Here are several key aspects highlighting their importance:

Healthcare: The development of nanoscale sensors enables the early detection of disease biomarkers, leading to more effective treatments. Nanonetworks offer significant potential in healthcare, facilitating targeted drug delivery and real-time health monitoring applications, including important measurements like blood pressure and oxygen saturation Dorj et al. (2017). In this monitoring process, the nanosensor data captured from patients is displayed on their smartphones, as depicted in Figure 2.2, enabling efficient monitoring and management of health conditions. Another notable application in healthcare is the diagnosis of cancer onset and metastasis from blood-derived factors, as well as monitoring the response to chemotherapy treatments Salvati et al. (2015).

A known architecture in this field is illustrated in Figure 2.3. It depicts the architecture of the IoNT-enabled healthcare system, comprising a set of nanosensors and smart devices deployed on, in, or around the human body, all connected to the gateway to the internet Pramanik et al. (2020).

Environmental Monitoring: Nanonetworks can be deployed for real-time monitoring of environmental parameters such as pollution levels, air quality, water contamination, and crop health monitoring. These networks can provide valuable data for environmental management and help mitigate environmental risks. For instance, crop monitoring, as shown in Figure 2.4, is possible because plants release specific chemical compounds that can be analyzed to assess environmental conditions and plant growth Yang et al. (2020); Afsharinejad et al. (2016).

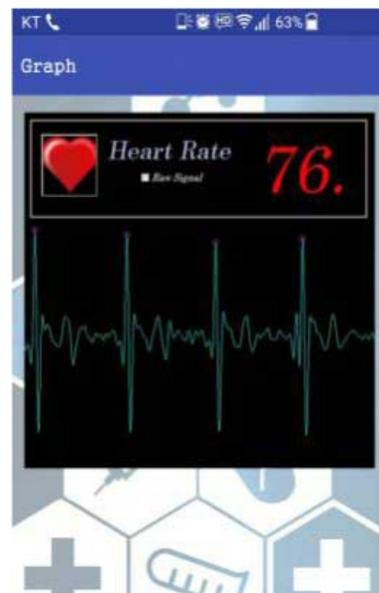


Figure 2.2: Result page of the ECG nanosensor Dorj et al. (2017).

For instance, environmental pollution, mainly caused by toxic chemicals from industrial, vehicular, and agricultural sources, impacts air, water, and soil quality. Consequently, nanotechnology plays an important role in addressing these multifaceted challenges through remediation techniques such as nanofiltration for cleanup and purification purposes. These techniques leverage the unique properties of nanomaterials to efficiently remove pollutants and restore environmental integrity. Additionally, nanotechnology facilitates advanced sensing and detection methodologies, enabling the accurate identification and quantification of contaminants in environmental matrices. Furthermore, proactive pollution prevention strategies are being developed, leveraging nanomaterial-based solutions to mitigate pollution at its source Yunus et al. (2012).

Construction engineering Nanotechnology offers significant benefits in construction engineering, notably by streamlining the implementation of structural condition monitoring systems in large-scale civil projects. Nanonodes, for instance, provide proactive solutions for constructions like bridges by serving as effective monitors for early signs of degradation and cracking, empowering authorities to take timely action. An example of nanotechnology's evolution is the development of graphene-based nanosensors. With graphene's outstanding electrical and thermal conductivity Imoni et al. (2023), these sensors excel at detecting changes in critical parameters such as temperature, humidity, or gas concentrations. Consequently, it enables the development of advanced sensors and infrastructure systems, offering significant improvements in monitoring and management.

Moreover, embedding nanonodes into roads and structures bolsters structural integrity (results in robust and durable structures) Aiken et al. (2011) while also yielding cost savings and enhancing maintenance efficiency. For example, carbon nanotubes have shown

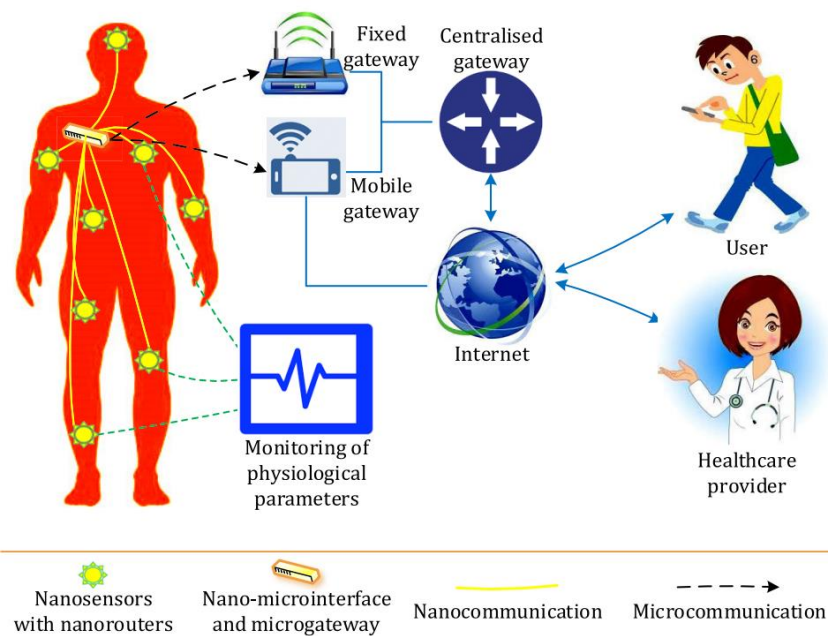


Figure 2.3: Architecture of the IoNT enabled healthcare system Pramanik et al. (2020).

exceptional mechanical strength, making them suitable for reinforcing concrete and improving the structural integrity of buildings Imoni et al. (2023).

Military technology: Nanonetworks are used in various ways for military applications. For example, a nanosensor can accurately and reliably detect biological weapons and harmful chemicals Kabir et al. (2021). Nanonetworks can be employed in advanced nuclear, biological, and chemical defenses Simić et al. (2018), enhancing detection and neutralization capabilities against these threats. Additionally, nanonetworks are crucial in sophisticated damage identification systems for common structures, warriors' shields, and combat wagons/vehicles.

2.2/ TS-OOK MODULATION

Nanonetworks diverge from conventional network structures, and the constraints of nanoscale hardware introduce unique challenges that necessitate innovative solutions. These constraints have led to the development of specific communication protocols and signal modulation techniques, distinct from those used in traditional network systems.

Nanonetworks are characterized by their limited energy capacities and dense configurations, presenting numerous challenges and unique characteristics vital to this study. Due to the small size of nanonodes, a single node can often communicate with hundreds of others within its communication range, leading to densely packed networks. This density introduces complexities in communication protocols.

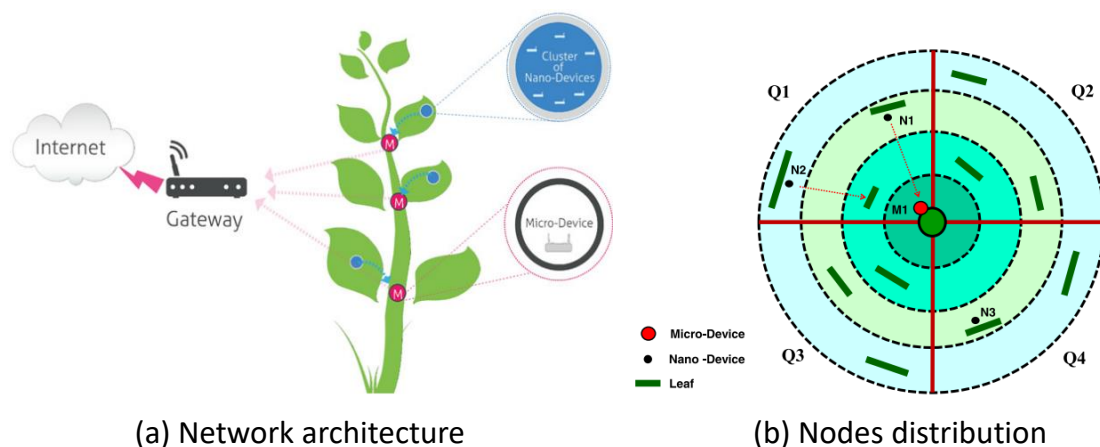


Figure 2.4: Nanonetworks for plant monitoring.

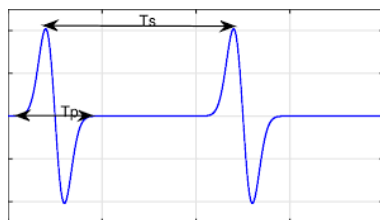


Figure 2.5: TS-OOK modulation.

Among these specific protocols is the Time Spread On-Off Keying (TS-OOK) modulation technique, proposed by Jornet and Akyildiz (2014a). This method is made for the communication of nanodevices operating in the radio terahertz channel.

TS-OOK uses extremely short, femtosecond-long pulses to transmit data. Information is encoded through these pulses, interleaved by predetermined duration intervals T_s . In this system, a '1' bit is represented by a power pulse of duration T_p , while a '0' bit is signified by a period of silence. The parameters of this modulation method are defined by the pulse duration T_p and the time between symbols T_s , with the spreading ratio ($\beta = T_s/T_p$) being a key factor.

The unique aspect of TS-OOK lies in its handling of the temporal spacing between individual bits, as shown in Figure 2.5. Due to the minuscule scale of nanodevices and the limitations of their hardware, it is not feasible to emit consecutive pulses without adequate gaps. Therefore, each bit in TS-OOK is separated by a time interval T_s , which is significantly longer than the pulse duration.

The proposed duration T_p for a single pulse in TS-OOK is 100 femtoseconds. Given the need for a significant gap between consecutive bits due to hardware limitations, the spreading ratio ($\beta = T_s/T_p$) often reaches high values, such as 1000. This ratio plays a critical role in the effective bandwidth of the channel ($1/T_p = 10 \text{ Tb/s}$).

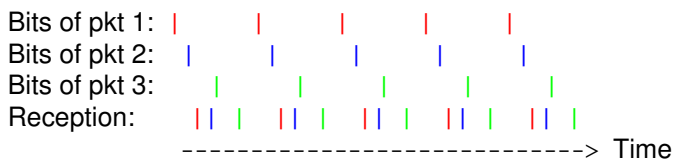


Figure 2.6: Packet overlapping by bit interleaving in TS-OOK.

Another feature of TS-OOK is the potential for packet overlapping, as shown in Figure 2.6, which is the interleaving of bits from multiple packets across a shared transmission channel. This feature is similar to time-division multiple access TDMA but operates at the frame level. While this allows for high channel capacity, it can also lead to inefficiencies in frame retransmissions in multi-hop forwarding scenarios if not properly managed. Additionally, hardware constraints play a significant role, particularly in limiting the number of frames a node can process simultaneously. This limitation can lead to packet loss and significantly reduce the effective channel capacity.

In TS-OOK, collisions are a unique phenomenon. They occur when a '0' bit (silence) is being received, and a '1' bit arrives simultaneously, overshadowing the '0'. The nature of collisions in TS-OOK is unique. They do not necessarily impact all receivers on the channel, but only those at specific distances from the source and receiving bits simultaneously. Not all collisions are detrimental (destructive); for instance, collisions involving two '0' bits do not interfere with each other since both represent periods of silence.

2.3/ SLR (STATELESS LINEAR-PATH ROUTING) PROTOCOL

Traditional routing protocols, designed to ensure that a packet reaches a destination, are not suitable for dense wireless nanonetworks. This is due to the challenges arising from constraints on resources, energy, memory, and data processing capabilities.

SLR (Stateless Linear-path Routing) Tsioliariidou et al. (2017) is a spatial addressing and unicast routing protocol where a message is sent to one destination only. All nanonodes are placed within a cubic space. Specifically tailored for nanonetworks, SLR comprises two phases: (1) the initialization/addressing phase, and (2) the routing phase.

The goal of the (1) initialization phase is to divide the network into zones and assign them coordinates. These coordinates are defined as an integer number of hops from a few special nodes called anchors. During this phase, two anchors placed at the vertexes of 2D network broadcast a beacon, as shown in Fig. 2.7. These anchors are positioned at the top-left and bottom-left corners of the network. The packets emitted include one field representing/storing the current number of hops (similar to a TTL, time to live, field) from the anchors. This field is initialized to zero and increments with each retransmission. At

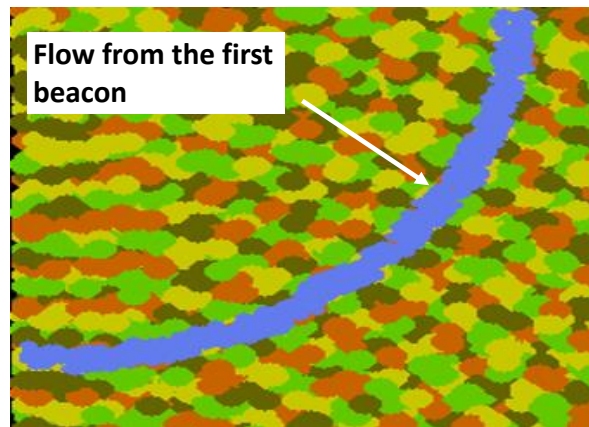


Figure 2.7: VisualTracer sketch for the SLR initialization phase.

the end of this phase, each zone will have unique coordinates representing the distance in hops to each anchor. Note that **all nodes within the same zone share the same coordinates (zone coordinates)** and that all the zones are disjoint (i.e. each node belongs to one zone only).

The goal of the (2) routing phase is to route the data packets in a linear routing path based on the coordinates assigned in the initialization phase, as shown in Fig. 2.8. During this phase, each node that receives a packet checks, using a simple formula, whether it is on the path (between sender and receiver) based on its own coordinates and the source and destination coordinates (found in the packet). If so, the data packet is forwarded; elsewhere, the packet is discarded.

2.4/ NANONETWORK SIMULATORS

2.4.1/ INTRODUCTION

A nanonetwork is an environment where an incredibly large number of small devices, potentially ranging from thousands to millions, are engaged in communication and interaction. The huge volume of these devices, combined with their limited capabilities, cost, privacy, and security, makes it impractical, if not impossible, to conduct real-world experiments at such a high density. As a result, in order to effectively evaluate protocols and applications for these networks, we rely on simulation tools. These tools allow us to model and analyze the behavior and performance of nanonetworks, evaluating protocols without the need for physical experiments, which would be challenging given the scale and complexity involved.

An example of a physical simulator is COMSOL Multiphysics ¹, a cross-platform tool

¹<https://www.comsol.com/>

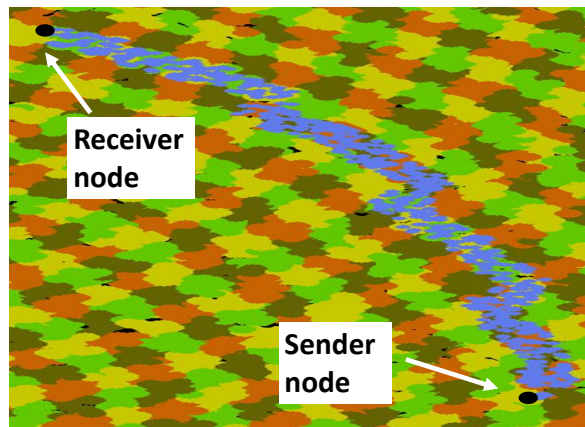


Figure 2.8: VisualTracer sketch for the SLR routing phase.

known for its detailed physical modeling capabilities. It proves highly effective in providing accurate simulations across various scientific and engineering problems. While COMSOL excels in modeling a wide range of phenomena, including structural mechanics, acoustics, and fluid flow, its high cost can be a limiting factor. Indeed, due to its computational requirements, COMSOL is better suited for simulations with a few nodes only, making it less practical for very large-scale (dense) network simulations.

In this section, we initially outline the essential features of a good (effective) simulator. We then provide an overview of specific nanonetwork simulators, including Nano-Sim, TeraSim, and Vouivre, to provide a comprehensive overview. Afterward, our focus shifts to an in-depth analysis of BitSimulator, identified as the optimal tool for our study.

2.4.2/ SIMULATOR DESIRED FEATURES

An efficient simulator is characterized by several key features. Among these, the most crucial are:

Efficient Computation and Performance: The simulator should be optimized for performance, ensuring that simulations are completed in a reasonable time frame, even when dealing with complex scenarios or large-scale networks. For example, a simulation of 20 000 nodes can be completed in 3.5 seconds, as shown later in section 4.5.1.

Scalability: It should be capable of handling the simulation of a large number of devices or nodes, scaling up to thousands or millions if necessary, without loss in accuracy.

Flexibility and Customization: The simulator needs to offer flexibility, allowing researchers to modify parameters, experiment with different scenarios, and customize aspects to suit specific research needs.

Detailed Reporting: It should provide detailed reporting capabilities, allowing for an in-depth examination of simulation results and behaviors.

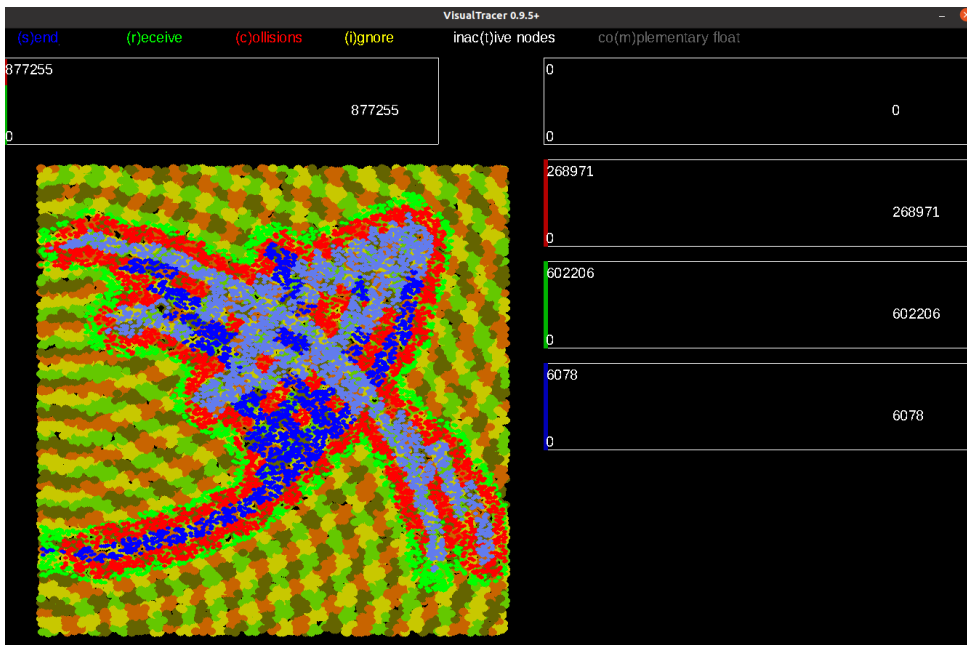


Figure 2.9: Sketch for a nanonetwork scenario in visual tracer (BitSimulator visualizer)

Reliability: The simulator must be reliable, with minimal bugs and errors, ensuring that the results it produces are consistent and trustworthy.

Advanced Visualization Tools: Effective simulators should be equipped with advanced visualization tools for a clear and detailed representation of simulation processes and results. These tools should offer graphical interfaces to display network topology, frame transmission paths, and the real-time state of nodes. The ability to visually track and analyze the dynamics of nanonetwork communications is important for understanding complex interactions and identifying potential areas for improvement. For example, BitSimulator incorporates a visualization tool named VisualTracer, which enables the observation of simulation events, as shown in Figure 2.9.

Comprehensive Documentation: Clear and detailed documentation is crucial for any software, as it promotes user adoption and simplifies the learning process, particularly for newcomers in the research field.

Adding these features ensures that the simulator can effectively match the behavior of nanonetworks. This provides researchers with valuable insights and data that would be challenging to obtain through physical experimentation.

2.4.3/ OTHER SIMULATORS

Several simulators have been developed to model and analyze nanonetworks, each offering distinct features and capabilities. The choice of a specific simulator over others

hinges on a range of factors, including scalability, accuracy, efficiency, and the particular needs of the research. In the following section, we will explore several existing simulators, examining their unique characteristics. This comprehensive overview sets the stage for a detailed analysis of BitSimulator, highlighting why it stands out as the preferred choice for our study. We will delve into BitSimulator's functionalities and specific features that align with our research requirements in nanonetworks.

2.4.3.1/ NANO-SIM

Nano-Sim, an extension of the ns-3, is an open-source network simulator Piro et al. (2013) designed to enable the simulation of electromagnetic nanonetworks and their interactions with larger networks, such as the Internet. Nano-Sim is distinguished by its implementation of TS-OOK modulation, a key element in the field of nanonetwork communications.

In Nano-Sim, three types of nodes are supported: nanonodes, nanorouters, and nanointerfaces. Nanonodes are characterized by their small size and limited energy, computational, and storage capabilities, making them suitable for tasks such as environmental monitoring and data collection. Nanorouters, with greater capabilities than nanonodes, play a key role in controlling, aggregating, and processing information received from nanonodes. Nanointerfaces serve as essential gateways, bridging the nano-scale network (WNSN) with larger-scale networks, thereby enabling communication with the broader world.

Nano-Sim, while a popular choice in nanonetwork simulation, faces certain challenges and limitations Dedu et al. (2014). A notable issue is its lack of recent updates; the tool hasn't seen recent updates and lacks comprehensive documentation. Additionally, it lacks critical models for propagation delay and energy consumption, which may affect the accuracy of certain simulations.

At its physical layer, Nano-Sim operates with a uniform symbol rate β and assumes that collisions occur when bits arrive at the receiver simultaneously. However, this does not entirely reflect the reality of destructive collisions in nanonetworks, typically happening when two different bits collide.

Furthermore, Nano-Sim's capacity is limited to simulating approximately a thousand nodes, and its incompatibility with ns-3's visualization modules poses challenges for projects where detailed visual analysis and large-scale simulations are important.

Given these limitations, especially in terms of scalability and advanced visualization capabilities, along with a deeper exploration of the β parameter, Nano-Sim was not chosen as the primary simulator for this research. The search for a more dynamic and visually capable tool was essential to meet the specific requirements of this study.

2.4.3.2/ TERASIM

TeraSim is another extension of the ns-3 network simulator specifically designed to model communication networks within the terahertz band Hossain et al. (2018). Its architecture is structured to include both nanoscale and macroscale applications.

A key component of TeraSim is that it incorporates the ALOHA and CSMA MAC protocols, each adapted to the unique properties of their respective physical layers.

Due to its reliance on the extensive infrastructure of ns-3, TeraSim struggles with large-scale networks, with a thousand nodes being the optimistic limit for simulations. TeraSim simplifies collision handling, which may result in packet loss when packets overlap, as it relies on the SINR (Signal to Interference plus Noise Ratio) to determine packet success.

While TeraSim overcomes some of Nano-Sim's limitations and offers features like propagation delay, path loss, molecular absorption, spreading loss, energy models, and advanced protocols, it still shares some common drawbacks. These include treating non-destructive collisions as collisions and a lack of support for large-scale networks. Additionally, TeraSim does not support visualization through advanced ns-3 modules, which is a critical aspect of large-scale network analysis.

Given these considerations, particularly regarding scalability, payload management, and visualization, TeraSim was not deemed suitable for the requirements of this dissertation.

2.4.3.3/ VOUIVRE

Vouivre, a C++ THz nano-wireless simulation library, plays an important role in extending the capabilities of the Dynamic Physical Rendering Simulator (DPRSim) Boillot et al. (2013). Originally developed for the Claytronics project, which focuses on simulating numerous Claytronics micro-robots (catoms), DPRSim initially lacked the ability for wireless transmission for the original catoms. Vouivre overcame this limitation by introducing wireless communication capabilities to the catoms.

Despite its innovative features, Vouivre has its challenges. The simulator focuses on a statistical approach, omitting the study of packet payload and bits. Furthermore, it lacks support for visualization. Although Vouivre can simulate tens of thousands of nodes in the nanonetwork Boillot et al. (2015), it is not widely used, not regularly updated, and does not align with our work's specific requirements.

2.4.4/ BITSIMULATOR

In the initial phase of my thesis, I acquired knowledge about BitSimulator Dhoutaut et al. (2018). This encompassed the understanding of the simulator's core functionalities, including the simulation phases it encompasses and delving into comprehending how the nodes are distributed within the simulated environment within BitSimulator. This foundational knowledge formed the basis for subsequent investigations and analyses within the context of my research on nanonetworks.

This section will provide a thorough exploration of BitSimulator's features, capabilities, and VisualTracer tool. In addition to examining its role in facilitating detailed investigations into nanonetwork communication, we will provide a guide that aims to offer a comprehensive understanding of how to use BitSimulator.

2.4.4.1/ INTRODUCTION

Understanding communication complexity in resource-constrained environments like nanonetworks is crucial for optimizing protocols and enabling advanced applications. BitSimulator is an advanced C++ simulator specifically targeting wireless nanonetworks. It is free software able to support a large number of nodes, up to hundreds of thousands of nodes².

BitSimulator processes tasks based on a discrete event model, executing events sequentially, i.e. it is a sequential program. However, it can be executed several times in parallel with different parameters (several instances can run in parallel) by specifying different output files. This is useful because usually a scenario is executed several times with different random seeds.

Unlike conventional network simulators, BitSimulator offers unique capabilities:

- **High detail:** Accurately simulates collisions and channel interactions even at the bit level, allowing investigations into coding schemes, channel access protocols, routing strategies, and congestion control mechanisms.
- **Scalability and efficiency:** Supports simulations of up to hundreds of thousands of nodes, making it suitable for exploring large-scale scenarios while maintaining fast execution times.
- **Customization:** Enables researchers to tailor simulations to their specific needs by adjusting parameters like β , experimenting with diverse scenarios, and customizing aspects like node density and communication range.

²<http://eugen.dedu.free.fr/bitsimulator>

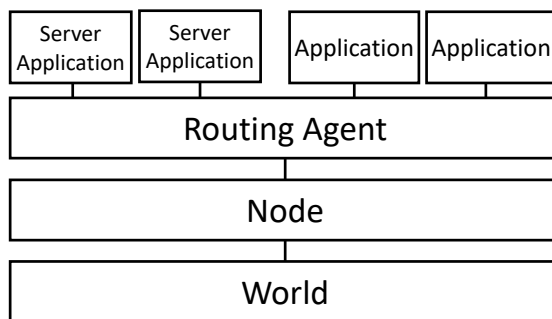


Figure 2.10: Main C++ classes interaction.

- **Reproducibility:** Ensures reliable and verifiable results through the use of random seeds and readily available software, facilitating comparison and collaborative research.
- **Bit-by-bit transmission and error computation:** BitSimulator provides detailed simulations of collisions and channel interactions at the bit level, allowing for in-depth investigations into channel access protocols and routing strategies. The mechanisms influencing the bit error rate and the distribution of errors depend on the coding and payload characteristics. This enables researchers to evaluate the impact of different coding strategies on error propagation and overall network performance.
- **Visualization:** VisualTracer, a visualization program, complements simulations by graphically displaying simulation events on a 2D or 3D map, as shown in Figure 2.9, enhancing comprehension and analysis. It is free software and has been used to validate the results of several papers.

Figure 2.10 illustrates the detailed interactions among the main C++ classes in BitSimulator. Together, the Node class and the World class intricately implement the physical and channel access control layers. **Nodes** within BitSimulator are equipped with nanotransceivers featuring customizable ranges, utilizing the default TS-OOK modulation scheme characterized by 100 fs pulses and a configurable β parameter per frame. The **Routing** class executes network layer protocols such as SLR or flooding, providing the flexibility to integrate new routing agents. BitSimulator's application layer distinguishes between two types: **Application**, facilitating packet transmission, and **Server Applications**, capable of both sending and receiving packets through a logical port that allows demultiplexing and distribution of incoming packets to the correct ServerApplication. Notably, multiple instances of Application(s) and ServerApplication(s) can be attached to a given node, significantly enhancing adaptability. Data flow through the network is encapsulated by the **Packet** class, which includes the binary payload alongside various headers such as source, destination, packet, and flow identifiers. Upon correct reception, packets are passed to a Server Application instance running on Nodes, completing the communica-

tion process within BitSimulator. Furthermore, the flexibility extends to the user's ability to customize and attach instances of these classes to nodes, providing additional adaptability to the simulation setup.

In this thesis, a primary focus is on the development and evaluation of novel routing protocols within the BitSimulator framework, as detailed in the forthcoming context chapter. The **Routing** class, integrated with the Node and World classes, assumes a central role in executing network layer protocols. Moreover, a significant aspect of our investigation centers around the ID (identifier) assignment within the simulation setup. This aspect is crucial for understanding and optimizing the communication process. The assignment and management of identifiers play a pivotal role, and this subject will be thoroughly explored and discussed throughout the thesis.

Bitsimulator, with its primary focus on nanonetworks, does have some **limitations**. As it is designed for nanomachines and nanoantennas, both transmissions and receptions are omnidirectional. This characteristic may not accurately represent scenarios where directional communication is essential. Additionally, it currently lacks an energy model implementation. Considering the crucial role of energy considerations in analyzing new protocols, this absence may limit the comprehensive evaluation of protocols that rely on energy-efficient strategies. Researchers should be mindful of these constraints when interpreting simulation results and proposing energy-aware protocols in Bitsimulator.

2.4.4.2/ USER'S GUIDE

To begin the testing and analysis of a new routing protocol through simulation, the initial step involves downloading BitSimulator. Following this, the new routing protocol must be implemented within the simulator. This enables researchers to utilize BitSimulator's capabilities for comprehensive testing and evaluation of the proposed routing protocol in the context of nanonetworks.

In BitSimulator, the prescribed **sequence of protocol execution** during simulation adheres to the following temporal framework: (1) Initialization of DEDeN at time 0 fs (enabled for nodes to know their local density), (2) Establishment of SLR setup at 200×10^9 fs, (3) Activation of Sleep enabling mechanisms at 500×10^9 fs, and (4) Commencement of CBR at 1000×10^9 fs. The initial timing can be adjusted within the XML file. Furthermore, the inclusion of a novel protocol at a designated moment is achievable, as elucidated in the discussion of the EIDA protocol in Chapter 4. Existing routing protocols in BitSimulator are pure flooding, probabilistic flooding, backoff flooding, Stateless Linear-path Routing (SLR) and their derivations,

The essential parameters for simulating a specific protocol within BitSimulator and replicating its outcomes are specified in the **scenario.xml** file outlined below (Listing 2.1). This

XML file presents detailed information regarding the setup of a homogeneous network, including the spatial distribution of nodes, modulation parameters, routing configurations, and additional specifications. By comprehensively defining the simulation scenario, the `scenario.xml` file serves as a fundamental guide for researchers aiming to reproduce and investigate the intricacies of diverse network protocols within the BitSimulator framework.

Listing 2.1: BitSimulator Scenario example in homogeneous network

```
<scenario>
  <world sizeX_nm="6000000" sizeY_nm="0" sizeZ_nm="6000000">
    <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="5000000" posY_nm="0" posZ_nm
      ="5000000"/>
    <node id="3" posX_nm="3000000" posY_nm="0" posZ_nm
      ="1000000"/>
    <genericNodes count="50000" positionRNGSeed="1"/>
  </world>
  <modulation>
    <ts-ook pulseDuration_fs="100" defaultBeta="1000"
      defaultCommRange_nm="285000" maxConcurrentReceptions
      ="1000000" minIntervalBetweenSends="0"
      minIntervalBetweenReceiveAndSend="0"/>
  </modulation>
  <routing defaultBackoffWindow="10000" backoffRNGSeed="1"
    backoffRedundancy="5">
    <SLRRouting commRangeSetup_nm="220000" anchor1id="0"
      anchor2id="1" useDeviation="false"
      useCounterBasedForwarding="false"/>
  </routing>
  <applications>
    <cbr flowId="123" srcId="2" dstId="3" port="3001"
      packetSize="100" interval_ns="100000" repetitions="1"
      startTime_ns="1000000" beta="1000"/>
  </applications>
  <log/>
</scenario>
```

This scenario outlines a network with dimensions of 6mm x 6mm, housing a dense configuration of 50,000 nodes. Each node is uniquely identified by its ID. Notably, Nodes 0 and 1 play crucial roles of anchors during the initialization phase of the SLR protocol. Node 2

functions as the predefined source for Flow 123, while Node 3 is assigned the role of the destination, determining the path for Flow 123. Flow 123 initiates at 1,000,000 nanoseconds with a packet size of 100 and a β value of 1000. Moreover, in this scenario, each node's communication range varies based on the simulation phase. The communication range is set at 285 μm during the setup phase, and to 220 μm in the routing phase. Modulation specifications define the TS-OOK scheme with a 100 femtosecond pulse duration, default β of 1000, and a maximum of one million concurrent receptions.

In BitSimulator, node distribution is a pivotal element of scenario setup, offering the flexibility to statically position nodes within the network. By assigning unique IDs and specifying precise X, Y, and Z coordinates in nanometers, researchers can strategically position nodes within the simulated environment. For instance, nodes 2 and 3 exemplify this approach, with Node 2 located at (5 000 000, 0, 5 000 000) and Node 3 at (3 000 000, 0, 1 000 000). This spatial control allows researchers to strategically position nodes, facilitating the exploration of various network configurations tailored to specific experimental scenarios.

In addition to the homogeneous network, BitSimulator allows for the creation of heterogeneous networks. For example, in the scenario outlined below (Listing 2.2), the network is partitioned into six parts, each characterized by a different density.

Listing 2.2: BitSimulator Scenario example in heterogeneous network

```
<scenario>

<world sizeX_nm="6000000" sizeY_nm="0" sizeZ_nm="6000000">

<node id="0" posX_nm="0" posY_nm="0" posZ_nm="0"/>
<node id="1" posX_nm="100000" posY_nm="0" posZ_nm="6000000"/>
<node id="2" posX_nm="5000000" posY_nm="0" posZ_nm="5000000"/>
<node id="3" posX_nm="500000" posY_nm="0" posZ_nm="1000000"/>

<genericNodes count="0" positionRNGSeed="10"/>
  <area shape="rectangle" x_nm="0" y_nm="0" z_nm="0" sizeX_nm
    ="2000000" sizeY_nm="0" sizeZ_nm="3000000" distribution
    ="uniform" nodesCount="8000" positionRNGSeed="11"> </
    area>
  <area shape="rectangle" x_nm="2000000" y_nm="0" z_nm="0"
    sizeX_nm="2000000" sizeY_nm="0" sizeZ_nm="3000000"
    distribution="uniform" nodesCount="6000" positionRNGSeed
    ="12"> </area>
```

```

<area shape="rectangle" x_nm="4000000" y_nm="0" z_nm="0"
  sizeX_nm="2000000" sizeY_nm="0" sizeZ_nm="3000000"
  distribution="uniform" nodesCount="5000" positionRNGSeed
  ="13"> </area>
<area shape="rectangle" x_nm="0" y_nm="0" z_nm="3000000"
  sizeX_nm="2000000" sizeY_nm="0" sizeZ_nm="3000000"
  distribution="uniform" nodesCount="4000" positionRNGSeed
  ="11"> </area>
<area shape="rectangle" x_nm="2000000" y_nm="0" z_nm
  ="3000000" sizeX_nm="2000000" sizeY_nm="0" sizeZ_nm
  ="3000000" distribution="uniform" nodesCount="10000"
  positionRNGSeed="12"> </area>
<area shape="rectangle" x_nm="4000000" y_nm="0" z_nm
  ="3000000" sizeX_nm="2000000" sizeY_nm="0" sizeZ_nm
  ="3000000" distribution="uniform" nodesCount="7000"
  positionRNGSeed="13"> </area>
</world>

<modulation>
  <ts-look pulseDuration_fs="100" defaultBeta="1000"
    defaultCommRange_nm="285000" maxConcurrentReceptions
    ="1000000" minIntervalBetweenSends="0"
    minIntervalBetweenReceiveAndSend="0"/>
</modulation>

<routing defaultBackoffWindow="10000" backoffRNGSeed="1"
  backoffRedundancy="5">
  <SLRRouting commRangeSetup_nm="220000" anchor1id="0"
    anchor2id="1" useDeviation="false"
    useCounterBasedForwarding="false"/>
</routing>

<applications>
  <cbr flowId="111" srcId="2" dstId="3" port="3001" packetSize
    ="100" interval_ns="100000" repetitions="1" startTime_ns
    ="1000000" beta="100"/>
</applications> <log/>

</scenario>

```

2.4.4.3/ BITSIMULATOR OUTPUT AND VISUALTRACER VISUALISATION TOOL

In the context of BitSimulator's output and visualization tools, the simulator offers a visualization program called VisualTracer, designed to visually represent simulation outcomes on both 2D and 3D maps. During the simulation, BitSimulator not only displays real-time information, such as simulation speed, total simulated time, and the overall number of events on the standard output but also systematically saves comprehensive results in various files. For instance, 'results-events.log' file captures detailed information about the number of packets sent, received, and collided by each node, while another file records the precise spatial coordinates of individual nodes. The analysis of trace files produced by BitSimulator is facilitated through the use of VisualTracer, a dedicated visualization tool. This tool imports log files generated by the simulator and provides an intuitive representation of protocol behavior, message propagation, and potential design issues. The graphical outputs generated by VisualTracer enhance the clarity of simulation results and enable researchers to navigate through events in a time range, facilitating a deeper understanding and analysis of the simulated nanonetwork scenarios.

As illustrated in Figure 2.11, the flow follows the specifications outlined in the scenario.xml file shown in Listing 2.1, originating from the source node 2 and directed towards the destination node 3. The color-coded representation, as shown in Figure 2.9, aids in interpreting the simulation results: nodes highlighted in blue indicate instances where a packet transmission has occurred. Nodes marked in green signify successful receptions, where the nodes correctly received a transmitted packet. The color yellow designates nodes that received a packet but encountered issues in processing or treating the information. Lastly, the color red designates nodes that received a packet that collided with others, resulting in potential data corruption. This visual representation offers a clear and insightful overview during the simulation, enabling researchers to identify and analyze various communication scenarios and potential challenges within the network.

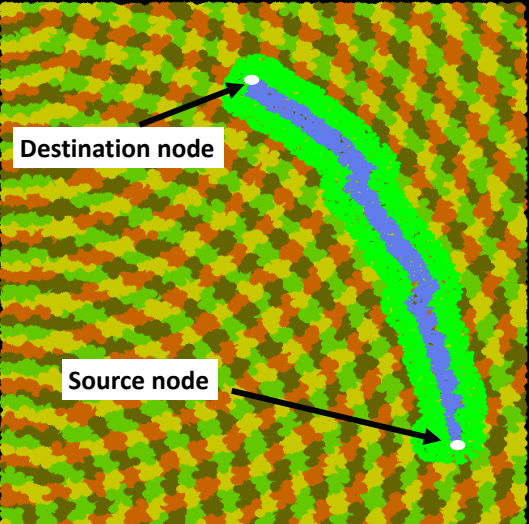


Figure 2.11: Visual Tracer Sketch for two flow scenario.



CONTRIBUTION

A CLASSIFICATION OF SINGLE-CHANNEL COMMUNICATION CONCURRENCY

When people think about “concurrency” in computer networks, they involuntarily associate it with flow-level concurrency, a characteristic of the well-known Internet stack (TCP and IP), where several flows (and not packets) are transferred at the same time through the each network link. However, this is not the only form of concurrency provided by networks. Moreover, this term is vague and has different meanings in different articles. In this chapter, we first review the mechanisms that provide concurrency. Afterwards, we identify and detail the various levels of communication concurrency, namely pulse, bit, packet, and flow levels, with examples of network technologies and protocols supporting each of them, and quantify the amount of this concurrency. We focus on concurrency at the same channel, where interference can occur, since using different channels does not create any interference. By expanding the understanding of the concept of communication concurrency, this chapter contributes to existing knowledge and provides valuable insights to protocol creators and users.

3.1/ INTRODUCTION

In various communication systems, the concept of concurrency, which enables multiple data transmissions to occur simultaneously, is of paramount importance, especially with the continuous increase of the number of users. Concurrent communication can take two distinct forms: concurrent communication on a single channel, or on different channels. A communication channel refers to a communication medium, such as a wire or a radio channel, through which information is transmitted between devices. In concurrency on the same channel, multiple data streams coexist and can interfere within this single channel,

while in multi-channel, data is transmitted simultaneously across multiple independent channels (e.g. different paths or sub-channels, or using Orthogonal Frequency Division Multiplexing OFDM) and the different flows do not interfere with each other. This chapter focuses on concurrency within a single communication channel.

3.1.1/ MOTIVATIONS

The concept of **concurrent** data transmission, which describes multiple senders simultaneously transmitting packets, is wide and confusing. This concept may have different meanings across different articles, where the word “concurrency” and its synonyms (such as simultaneous and parallel) are usually not defined, and are used in various contexts with **different** meanings. For example, for a reader familiar with the concurrency term (at flow-level as in IP protocol), reading about PiP (bit-level) concurrency will be confusing. Some random examples of concurrency contexts are:

- for several **channels** (or paths): “Multipath TCP provides the ability to simultaneously use multiple paths between peers” Raiciu et al. (2020).
- for several **packets**, but not bits, received concurrently: “TS-OOK enables robust and concurrent communication among nano-devices” Jornet and Akyildiz (2014b);
- for several **bits** received at the same time: “the interference to the transmission over a link comes from the concurrent transmitters” Liu et al. (2013), and “PiP exploits packet concatenation to achieve concurrent data collections from multiple neighboring nodes in a single transmission slot” Zhang et al. (2018);
- for several **pulses** received at the same time: “provide a scalable concurrent ranging solution that can be practically implemented on off-the-shelf UWB devices” Großwindhager et al. (2018);

3.1.2/ CONTRIBUTIONS

The goal of this chapter is to gather the knowledge of the various meanings of the concept of concurrency, and provide a classification of them into different levels: flow, packet, bit, and pulse levels, as shown in Fig. 3.1, along with examples of their use. To the best of our knowledge, this is the first classification of communication concurrency.

This classification considers **where** concurrency appears rather than the **goal** it is used for (its intended purpose). As an example, the goal of Glossy Ferrari et al. (2011) is to reconstruct packets, but the concurrency occurs at the bit level, hence it will be classified as bit-level (and not packet-level) concurrency.

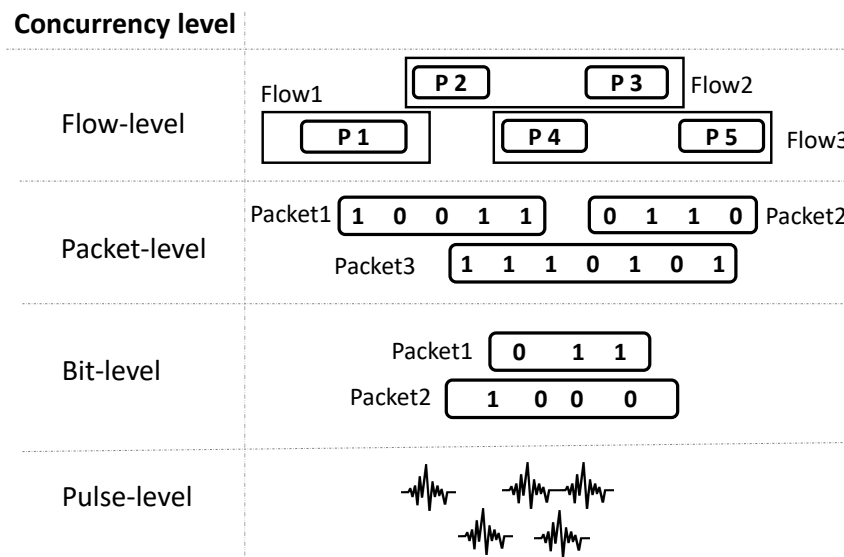


Figure 3.1: The various concurrency levels.

3.1.3/ SCOPE OF THE CHAPTER

As it was already stated, this chapter focuses on communications using a **single channel**, or the same channel (or link). However, it is important to understand that the word channel has several meanings in the literature. In this article, we use the word **channel** for a medium where, at some point in space and time, only one bit (or one pulse for pulse-based technologies) at a time can travel, and if a second one exists, then it interferes with the first one. The channel can be either dedicated (e.g. a wire) or allocated a specific frequency range.

For example, in the sense of this chapter, an Ethernet physical cable is a **channel** because two bits cannot exist in the cable at the same time without interference. Likewise, 802.11b (Wi-Fi) provides 14 channels, and each of them is a **channel** (although this is not completely true, as two consecutive channels overlap each other). For LoRa, a **channel** means using the same frequency and the same spreading factor (SF 5..12) Augustin et al. (2016).

In this chapter, we are interested in one channel because this is the case where several communications disturb each other, cause interference, and as such lead to lost packets.

As written above, this chapter **is not about several** (or different) channels, because in this case several packets can travel concurrently with no interference/collision. To clarify the scope of the chapter, in the following we give some examples using **several** channels, which are thus out of the scope.

The concurrency of this article should not be confounded with multi-channel concurrency, which is formed by independent links/channels, and as such the concurrency is harm-

less. For example, Wi-Fi provides 14 channels, and a sender might send concurrently 14 packets on the 14 channels. As another example, at the network level, **Multipath TCP** (MPTCP) Raiciu et al. (2020) is a protocol that enhances traditional TCP by enabling concurrent data transmission over multiple network paths. Another example, in WSN (wireless sensor networks), is RCB-MRT Zhang et al. (2020), where a sensor node divides each data packet into several redundant sub-packets, transmitted concurrently through several paths to the aggregation node (sink), and, upon reception, the sink node performs data fusion.

Similarly, the concurrency provided by a full-duplex system is out of the scope of this chapter. Full-duplex communication Jain et al. (2011) means sending one packet and receiving one packet at the same time, which is different from concurrent communication in this chapter, which means sending multiple data packets at the same time (or receiving them at the same time).

Section 3.2 presents the four levels of concurrency, namely pulse, bit, packet, and flow, along with the mechanisms used by various technologies and protocols to support concurrency. Finally, Section 3.3 presents the conclusions.

3.2/ CLASSIFICATION OF CONCURRENT COMMUNICATIONS

This section presents in detail the classification of concurrency on the same channel and the mechanisms used to achieve this concurrency, followed by a summary and information about the amount of concurrency.

In general, information (data) is transmitted in a network as a flow, which is divided into packets, which in turn consist of several bits; moreover, for pulse-based technologies, each bit is transmitted as one or several pulses. Thus, we classify the concurrency into different levels, covering flow, packet, bit, and pulse-level of concurrency.

3.2.1/ FLOW-LEVEL CONCURRENCY

The concept of **flow** is software-specific. At transport/routing layers, a flow denotes a single, meaningful end-to-end transmission between two peers.

Flow-level concurrency, the most popular form of concurrency, means that multiple flows are multiplexed on the same channel, otherwise said packets from one flow are transported **between** packets of other flows, as shown at top of Fig. 3.1. In other words, at a given time, the link transports a packet from one flow, but the second flow has already started and has not yet finished. It is essential to understand that while flows are pro-

cessed **concurrently**, packets are processed **sequentially**, i.e. on any link packets are transmitted sequentially, one **after** the other.

Being software-related, the flow-level concurrency is natively supported by TCP and IP protocols. Each IP packet belonging to a TCP flow contains a header with information such as source and destination addresses. IP routers receive numerous packets that belong to the same or distinct flows and, thanks to this header, can process them independently from each other, a process known as **packet switching**. Thus, this header enables routers to make routing decisions concurrently for several flows.

Because the flow-level concurrency could be processed by software (IP for example), the flow-level concurrency is supported by all network technologies. This means that flow-level concurrency can be used in any network setup without requiring specialized hardware or modifications at the physical layer.

Most of the current widely-used technologies do not support any concurrency by themselves, because, as they use signal carrier waves, two concurrent packets generate collisions. Thus, they only support software-based flow-level concurrency. Examples of such technologies are Ethernet (IEEE 802.3 standard Law et al. (2013)), widely employed in both local and wide-area networks, which has a collision detection mechanism, and Wi-Fi and Bluetooth Bisdikian (2001) radio technologies, that operate at a high frequency (such as 2.4 GHz).

However, other technologies and techniques do provide flow-level concurrency (and possibly packet-level concurrency too). For example, Time-Division Multiple Access (TDMA), allows to share one channel among several nodes, however, each node has exclusive access at a given time, and its data **does not collide** with data of other nodes, hence the data flows are transmitted at the same time. Similarly, duty-cycling technologies, such as LoRa, use natively the flow-level concurrency by allocating a limited transmission time for each device (e.g. 1% of the time). Thus, if a device needs to send more data than it can send in one time slot, it has to cut the data in smaller packets and send each of them between packets from other devices, leading to flow-level concurrency.

3.2.2/ PACKET-LEVEL CONCURRENCY

Concurrency at packet level appears in a channel having multiple packets at the same time (as shown in Fig. 3.1), but with no concurrent bits (in the latter case it should go to bit-level concurrency).

One **mechanism** to achieve this is, for example, through bit interleaving, as shown in Fig. 3.1: the three packets are at the same time on the channel, but no two bits are simultaneously found in the channel. Bit interleaving, used in TS-OOK for example, is

used to send multiple packets simultaneously by injecting bits of one packet between the bits of another packet.

Note that packet-level concurrency results in flow-level concurrency, i.e. a protocol supporting packet-level will naturally support flow-level concurrency too. Indeed, multiple packets from different flows being sent at the same time naturally leads to flow-level concurrency because these packets represent different data flows.

Ultra Wide Band (UWB) and UWB-based Time Spread On-Off Keying (TS-OOK) are technologies based on radio **pulses** (contrary to signal carriers like conventional wireless technologies). They support packet-level concurrency and are described in the following.

Unlike traditional systems that modify sinusoidal wave properties (amplitude, frequency, and phase), UWB systems transmit data by emitting radio energy at specific time intervals Minoli and Occhiogrosso (2018). UWB communication uses a wide bandwidth, exceeding 500 MHz as a minimum Kim and Tran-Dang (2019). UWB can use several basic modulations (PPM, BPM, OOK, and PAM) Ghavami et al. (2007), allowing to add digital information onto analog pulses. Time Spread On-Off Keying (TS-OOK) Jornet and Akyildiz (2014b); Maded (2017) is a particular UWB modulation used in terahertz band. Bit 1 is transmitted as a femtosecond-long pulse, whereas bit 0 is “sent” as silence. The time between two consecutive bits sent is very high, much higher than the duration of a pulse (e.g. 1000 times Jornet and Akyildiz (2014b)). This time is fixed and known by both sender and receiver, so the receiver can track all the bits of a packet. These high time gaps mean that the bits from several packets can be interleaved, and several packets are received concurrently, as shown in Fig. 2.6, leading to packet-level concurrency (but not bit-level concurrency, as TS-OOK has no provision to decode concurrently-received bits).

The concurrency can be exploited to define the duty-cycling period Medlej et al. (2021) or the channel occupancy in dense networks Arrabal et al. (2018b).

3.2.3/ BIT / SYMBOL-LEVEL CONCURRENCY

Concurrency at the bit level appears when several bits are transmitted at the same time in a channel, as shown in Fig. 3.1. Of course, this also leads to packet-level and flow-level concurrency.

Among the **mechanisms** which enable bit-level communication concurrency we cite:

- **Capture effect**, as considered in SIC and CT-LoRa for example (described below), refers to a situation where, among multiple signals, only the strongest one can be correctly demodulated by a receiver, and the other signals are ignored. The stronger signal must be greater than the sum of all the other signals, for example by at least 3 dB as shown in Fig. 3.2.

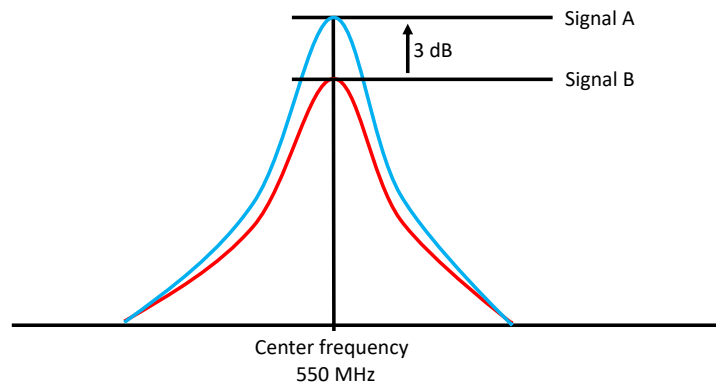


Figure 3.2: Capture effect between two signals.

- **Successive interference cancellation (SIC)** Sen et al. (2013), used in SAR for example, is a physical capability that allows a receiver to decode several bits or packets that arrive concurrently. It takes advantage of the capture effect several times, for each received packet. With traditional signal extraction, in the case of packet collisions, only the packet with the strongest signal can be decoded, while all other signals are treated as interference and ignored; consequently, if the signal of interest is not the strongest, it is not recovered. SIC overcomes this limitation and enables the recovery of weaker signals of interest. With SIC, the bits of the strongest signal are decoded as before, then the original signal corresponding to those bits is reconstructed and subtracted (canceled) from the received signal (containing the combined signals). Similarly, the next strongest signal is decoded from this residue to extract the weaker signal, and canceled if needed, as illustrated in Fig. 3.3. Thus, decoding accuracy depends on the received signal strength of the signal involved.
- **Constructive interference**, used in Glossy for example (described below), is when multiple nodes transmit the **same** packet simultaneously, resulting in the constructive addition of their signals at the receiver. It occurs when a receiver can detect the superposition of the base-band signals generated by multiple transmitters of the same signal; if it cannot detect the two signals, destructive interference occurs.

Interference generates collisions. In conventional technologies, bit collisions occur (signal overlap/collide) in the channel. It should be noted that this is not always the case, for example, in TS-OOK bits are transmitted as pulses (presented in the next section), and consequently, bit collision occurs at **receiver**: two bits sent by two different nodes do not collide in the air, and not at all receiving nodes, but only at nodes which receive them in the same capture window (the receiving time depends on the distance between sender and receiver) Jornet and Akyildiz (2014b).

To prevent collisions, it is necessary to avoid multiple nodes transmitting data simultane-

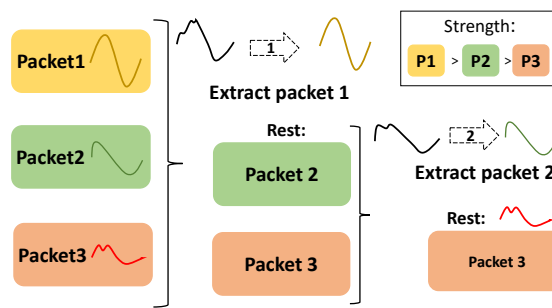


Figure 3.3: Successive interference cancellation.

ously, through techniques like carrier sensing, channel reservations, arbitrated medium access methods like TDMA, and random backoff mechanisms. Alternatively, CDMA (Code-Division Multiple Access) is a technique that uses different spreading codes (each sender uses a different power), so the receiver can extract the collided data (the wider signal can be extracted by order).

Concurrent bits can interfere destructively, leading to packet collisions, hence reducing communication efficiency. In a collision, several received bits overlap (interfere) with each other. It can also interfere constructively when the colliding bits correspond to the same data. Without constructive interference, even if the two nodes have to send a similar packet to a common receiver, they need coordination to determine which node sends the packet first. The capture effect can also be exploited if the time displacement of these two signals is less than the capture window, Otherwise, the bits are extracted based on constraints such as time displacement.

The above mechanisms might need to consider the time delta between two bits of packets sent concurrently (for constructive interference), the received signal strength RSS (for capture effect), the radio technology used (modulation and encoding), and the option to send an identical payload or not.

The following examples of protocols use bit-level concurrency.

Concurrent transmission in wireless sensor networks (WSN) can be achieved through the capture effect and constructive interference Yuan and Hollick (2013). The capture effect allows receivers to pick up signals from interfering transmitters, even when nodes are not perfectly synchronized. For instance, when the stringent time synchrony condition of $0.5 \mu\text{s}$ cannot be met, the capture effect can still occur since it has much looser time synchrony requirements. However, if the hardware-based implementation of rebroadcasting fully eliminates the software delay, thus, the one-hop constructive interference can be guaranteed.

One example of a flooding scheme in WSN is Glossy Ferrari et al. (2011). For it, interference is utilized as an advantage rather than a problem. It achieves fast network flooding

by making nodes transmit simultaneously the **same** packet. These transmissions create constructive interference, and the packets are successfully decoded if the temporal offset between concurrent transmitters is less than $0.5 \mu\text{s}$. With Glossy, nodes have their radios active only at specific scheduled times, during which they listen for communications over the medium and relay packets **immediately** after receiving them.

An example of a communication scheme in WSN is Packet-in-Packet (PiP) Zhang et al. (2018). PiP exploits constructive interference and the capture effect, as derived from the essence of Glossy. In PiP, a node initiates communication by broadcasting a “beacon” packet, and upon reception, neighboring nodes transmit their own packets concurrently. If packet contents are identical, then the receiver can correctly receive the packet due to constructive interference which relies on accurate synchronization. Otherwise, if the packet contents differ, the receiver can capture the packet with the highest power thanks to the capture effect. PiP also introduces the capability of packet concatenation, allowing multiple nodes to transmit simultaneously while dynamically adjusting the power amplifier. The goal is to ensure that the receiver can extract diverse packets from all senders during a single transmission round.

In Disco (deliberate synchronized constructive interference) Wang et al. (2015), multiple senders in WSN coordinate to transmit their packets at the same time. Its key modules include a chip-level synchronization algorithm and a link selection and alignment algorithm. The chip-level synchronization algorithm enables concurrent transmissions to be synchronized in $0.5 \mu\text{s}$ by compensating for propagation and radio processing delays. The link selection and alignment algorithm determines which co-senders participate in simultaneous transmissions. One of the nodes, the trigger node, senses the channel using CSMA/CA, and when it is free, it broadcasts a synchronization packet, which instructs all the co-senders about the destination and the time of the transmission. Thus, all the co-senders simultaneously transmit the packet, and the receivers use constructive interference to decode the packet.

Multi-hop LoRa networks benefit from the concurrent communication flooding multi-hop protocol (CT-LoRa), which combines the LoRa physical layer with concurrent transmission techniques to improve network efficiency Liao et al. (2017). This protocol enables multiple nodes to simultaneously transmit packets carrying the same content, rather than avoiding packet collisions. It exploits the capture effect, i.e. it extracts the packet with the strongest power during collision. After the initiator sends the initial packet, each node that receives it for the first time immediately retransmits it as another broadcast. This flooding eliminates the need and overhead of collision avoidance mechanisms and enables rapid dissemination of packets throughout the network.

Unlike other protocols, mZig protocol Kong and Liu (2015) enables a ZigBee receiver to decode/decompose multiple concurrent packets from one collision directly (from different

transmitters). A chip, the smallest unit carrying information, consists of multiple samples. A chip 1 is represented by a positive half-sine shaping, and a 0 is represented by a negative half-sine. Every four bits are spread to their specified 32 chips. When packets are sent simultaneously, their base-band signals overlap if they have a chip-level time offset. In mZig, if some collision-free samples (of packet 1), along with the sample shaping, are known, the transmitted chip of the first packet can be extracted. Then, the extracted chip is subtracted from the collided packet to obtain the first collision-free chip of the second packet. Repeating the operations of subtraction and estimation decomposes the collided packet into two packets chip-by-chip. However, if there is no chip-level time offset, the collision-free samples method cannot be used. Therefore, mZig decomposes packets by leveraging the amplitude difference since the amplitude of all chips in one packet is equivalent.

SAR (SIC Aware Routing) protocol Liu et al. (2013) uses SIC mechanism (differentiate several concurrent signals at link layer), thus allowing the network layer to receive and process multiple packets simultaneously. Routing in SAR is done in three steps: source node initiates route discovery, intermediate nodes update path information while avoiding loops, and destination node selects the optimal path with minimum interference (to optimize spatial resource consumption) and answers with the selected path.

Chiron Li et al. (2018) enables the concurrent transmission and concurrent reception of two signals from two different protocols, one Wi-Fi and one ZigBee, without interference, by using different amplitudes and bandwidths. The reception process involves signal separation, demodulation, and the application of an error correction mechanism to rectify distorted signals.

To conclude, several methods are used to achieve bit-level concurrency. Some protocols (like Glossy, PiP, and CT-LoRa) exploit constructive interference and capture effect, either of the same packet (for Glossy and CT-LoRa), or for different packets (like PiP), possibly with a power amplifier (like PiP). Other protocols (like mZig) use collision-free samples along with the sample shaping in order to extract the overlapped signal.

3.2.4/ PULSE-LEVEL CONCURRENCY

Conventional wireless technologies, such as Wi-Fi and LoRa, are based on carrier waves, and information (bits) is encoded by varying the properties of the carrier signal, including amplitude, frequency, or phase. In contrast, in pulse-based communication systems, the fundamental units of information transmission are discrete pulses, which convey data, for example, through the presence or absence of these pulses at specific time instances. The modulation scheme represents information through analog radio pulses and defines whether a single or multiple pulses are used to represent a single bit.

Pulse-level concurrency implies bit-level concurrency (as one or multiple pulses are used to represent one bit; for example, sending three pulses that collectively represent one bit from multiple senders is equivalent to sending two bits), which in turn, as mentioned in the previous section, implies packet and flow-level concurrency.

Some **mechanisms** enabling pulse-level communication concurrency are pulse injection (or pulse interleaving) and pulse shaping. In the following, we detail them through the use of Ultra-wideband (UWB), one example of pulse-based technology, where communication and ranging, its two main applications, use concurrent transmission and reception techniques.

3.2.4.1/ CONCURRENTLY USING PULSE INJECTION

Pulse injection transmits additional data by exploiting the time displacement of two pulses and injecting other data pulses or extra information in between them.

One method to implement pulse injection is time hopping. Time hopping (TH) allocates pseudo-random transmission times to pulses emitted by distinct users (Time Hopping Impulse Radio, TH-IP) Molisch (2009), aiming to reduce collisions and thus facilitating a multitude of simultaneous communications. TH UWB system needs accurate timing for the generation of the transmitted sequence and subsequent reception.

TH can be implemented by changing the preamble code in front of each transmitted packet, which defines the pseudo-random sequence used for time-hopping during the transmission of that packet Vecchia et al. (2019). Preamble codes are thus envisaged as a mechanism to enable multiple non-interfering accesses to the wireless medium.

TH can also be implemented by dividing the time into frames of fixed duration with precisely one pulse transmitted per frame Colli-Vignarelli et al. (2009). However, the choice of the frame of each pulse is randomized to avoid catastrophic collisions using a pseudo-random time-hopping sequence (THS).

In practice, the IEEE 802.15.4-compliant DW1000 communication chip DecaWave Ltd (2016) achieves non-interfering communication through a different pulse repetition frequency (PRF) with different preamble codes. The PRF signifies the number of pulses sent per second, including the pulse and the pause time. The preamble is part of the packet preceding the actual data, indicating that someone is about to transmit data, and its purpose is synchronization.

Another method to implement pulse injection is provided by Chorus Corbalán et al. (2019). It is a UWB ranging (localization) scheme in which several localization anchors at known positions send a pulse **at the same time**. The target device, given that it is at different distances from anchors, receives them at **different** times, and computes the distance

Table 3.1: Concurrency levels supported by various technologies and protocols.

	Concurrency level			
	Pulse	Bit	Packet	Flow
Network technologies:				
Carrier-based (Ethernet, Wi-Fi, etc.)	X	X	X	✓
Pulse-based (UWB, TS-OOK)	✓	✓	✓	✓
Network protocols:				
Glossy, PiP, CT-LoRa, mZig, SAR	X	✓	✓	✓
Pulse injection and pulse shape-changing	✓	✓	✓	✓
Others	✓/ X	✓/ X	✓/ X	✓

using the time difference of their arrival time (like in GPS).

3.2.4.2/ CONCURRENCY USING PULSE SHAPE CHANGING

Modifying the shape of pulses enables the receiver to reliably detect signal peaks associated with different responders, even in the case of overlapping responses. For instance, DW1000 radio chip can widen the transmitted pulses, thus allowing pulse shaping. This is accomplished by altering the shape of the transmitted pulses based on the responder ID, using a specific pulse template Großwindhager et al. (2018). As these distinct pulse shapes are clearly visible in the channel, the receiver can easily detect the responses of the neighbors. The receiver encodes the responder ID using these different pulse shapes. The number of pulse shapes that can be supported determines the degree to which responders can be differentiable through this pulse-shaping technique.

To sum up, pulse-level concurrency can be achieved either by exploiting the time displacement between two pulses (sent at the same time or at different times by several senders), or by changing the shape of the transmitted pulses.

3.2.5/ SUMMARY

Table 3.1 provides a summary of the communication concurrency levels on the same channel across the technologies and the protocols presented in this section.

We recall that pulse-level concurrency can be achieved by pulse injection or by pulse shape-changing; bit-level concurrency means that two bits exist on the channel at the same time (combined); packet-level means having multiple packets simultaneously in the channel (without bit-level concurrency); and flow-level means having multiple flows (but not multiple packets) simultaneously in the same channel.

Pulse-level concurrency exists only in pulse-based technologies (like UWB), and not in carrier-based technologies (like Wi-Fi and Ethernet, where concurrent packets create destructive collisions). Flow-level concurrency is software-based and as such is supported

by all the technologies and protocols.

We recall that concurrency level support automatically enables support for the “right-side” levels too, i.e. bit-level concurrency implies packet and flow-level concurrency as well, as confirmed in the table.

Note that this table is not exhaustive, some network protocols exist which support for example only packet and flow-level concurrency, as shown in the last row of the table (column Others).

Different methods can be used to allow concurrency. For instance, signal-based technologies may exploit the capture effect and constructive interference, while pulse-based technologies may use pulse injection or pulse shaping. The capture effect ensures the demodulation of the strongest signal, effectively filtering out interference, while SIC systematically extracts weaker signals from overlapping transmissions. Constructive interference enhances reception when identical signals align (exploit their combined strength). Bit interleaving and pulse injection organize data bits/pulses and inject additional information within signal intervals.

Finally, a fundamental question is whether the **concurrency level** is a characteristic of the **network technology** (hardware, physical layer), or of the **protocol** (software). As shown in the table, the level of concurrency supported depends either on network technology (based on carriers or on pulses) in pulse-level concurrency, or on protocol in pulse, bit, and packet-level concurrency.

3.2.6/ AMOUNT OF CONCURRENCY

The ability to achieve concurrent communication is influenced by several parameters of network technologies and protocols, and also by **node** capability Vinyals et al. (2011). They also give the **amount** of concurrency, analyzed in the following.

Node capability is given by hardware factors, including device transmission, reception, and processing capabilities, and sometimes also by its size, which might prevent complex operations in hardware. For example, nodes in nanonetworks are tiny and thus do not have enough memory to store numerous concurrently received packets: “the receiver can simultaneously track a fixed number of incoming packets, K ” Jornet et al. (2012).

Another parameter impacting the concurrency is the **time**. In TS-OOK, this amount is quantified by a parameter known as β . We recall that in TS-OOK, pulses are sent with a large space between them, and β is the ratio between two consecutive pulses and the length of a pulse. Other pulses can be transmitted in between. The higher β , the higher the number of possible concurrent packets. The value of 1000 is found in the literature (“The ratio between the time between pulses and the pulse duration is kept constant and

equal to $\beta=1000$ " Jornet and Akyildiz (2014b)), which means that up to 1000 packets can be transmitted concurrently. Likewise, in UWB, concurrency is achieved through time displacement, and the extent (amount) of concurrency achievable is limited by the temporal spacing between pulses.

Another parameter is the **signal strength**. In SIC Sen et al. (2013), the amount of concurrency (number of concurrent bits/packets that can be decoded) is given by the differences in signal strength of the received signals. If one signal is stronger than all the other signals together (the residual), then the strongest signal can be extracted through the capture effect Liu et al. (2013), and recursively the same procedure can be applied to the residual, thus extracting the concurrent bits one by one. To conclude, the number of bits that can be decoded concurrently (the amount of concurrency) is given by the differences in signal strengths.

Another parameter is **pulse shaping** (in pulse-based technologies only). To achieve concurrency, several pulses with different widths can be used (by increasing pulse width). For example, four senders can simultaneously transmit four pulses with different widths, and the receiver can differentiate and extract all of them. Of course, to effectively differentiate between received pulses, each of them must have a distinct shape. Consequently, the number of unique pulse shapes available determines the maximum amount of concurrency.

3.3/ CONCLUSIONS

This chapter provides a comprehensive understanding of concurrency in data communication. It classifies concurrency in different levels, namely pulse, bit, packet, and flow levels. It introduces the mechanisms used to enable concurrency, and gives numerous examples of network technologies and protocols in each level. Carrier-based network technologies support only flow-level, whereas pulse-based technologies support all the levels of concurrency through pulse injection, and pulse shaping (pulse, bit, packet, and flow levels). Network protocols support various levels, using mechanisms like capture effect, constructive interference, pulse injection, and pulse shaping.

EIDA, A BEST EFFORT EQUITABLE DISTRIBUTED ID ASSIGNMENT MECHANISM

Assigning identifiers (ids) in dense nanonetworks presents significant challenges due to the large number of nodes, potential for interference and signal overlap, and the difficulty in ensuring fair and efficient id assignment.

Existing solutions, such as ideal and random assignments, have their own limitations and unsuitability. Ideal assignments may not be practical in dense networks due to the huge number of packet exchanges required. Conversely, random assignments can lead to inefficiencies and a lack of coordination among nodes, resulting in unequal distribution (no-equitability) of node ids.

To overcome these limitations, we introduce EIDA (best effort Equitable Id Assignment), a configurable, and distributed mechanism designed for dense nanonetworks. EIDA combines the principles of ideal and random assignments to assign ids in the best effort of equitability while exchanging a determined number of packets.

EIDA functions by allowing a subset of nodes to assign their ids through packet exchanges until a predetermined level of equitability is reached. Beyond this point, the mechanism transitions to random id assignment for the remaining nodes, thereby minimizing further communication. This strategy aims to balance between ideal and random assignment methods, aiming for best effort of equitable id distribution.

Our simulations on ultra-dense nanonetworks have demonstrated the effectiveness of the EIDA mechanism in id assignment. A notable advantage of EIDA is its applicability in enhancing routing protocols. For example, it allows these protocols to select a desired (optimal) number of nodes for data packets forwarding at each hop, thereby minimizing the huge number of packet exchanges in network communication.

4.1/ INTRODUCTION

Dense multi-hop networks are composed of numerous nodes with high density, i.e. numerous neighbors. These networks present significant challenges for efficient communication, requiring unique node identification (id) as a fundamental primitive for several operations. One such operation is multi-hop routing, where packets are relayed through multiple nodes to reach their destination. However, without proper routing mechanisms, this process can be highly inefficient and can cause energy waste.

To avoid this inefficiency, nodes can be partitioned into zones, with each zone having a designated cluster head (only one node) responsible for forwarding packets. This approach can significantly improve network efficiency by reducing the number of nodes involved in packet forwarding. However, zones need to elect their cluster head and have very high load unbalancing, and these nodes deplete their battery very fast.

Another method is to partition nodes into groups, and hence split traffic among potential forwarders, where each packet is processed by only one group of nodes. To achieve equitable traffic splitting, i.e. avoiding that one flow is processed by e.g. 18 nodes and the other by only 2 nodes, groups need to have equal sizes. One way to achieve this is by assigning a unique id to each node and then using a simple modulo operation (id divided by the number of groups) to divide them into groups. FR-SLR Al Mawla et al. (2023b) is an example of routing protocol, where only nodes with specific id participate in the routing process along the transmission path, and EIDA can be used to assign ids to nodes.

EIDA is an id assignment mechanism in dense networks, in particular in nanonetworks, where nodes are very small and limited in memory storage, energy, and computing capabilities. Designing such a method is challenging due to the large number of nodes, potential interference and signal overlap, and difficulties in ensuring fair and efficient assignments.

Existing assignment methods have drawbacks. For example, the classical ideal assignment, where each node receives consecutive ids $(0, 1, \dots, n - 1)$, requires numerous packet exchanges among the nodes, which is unacceptable in a dense nanonetwork. The random assignment does not guarantee equitable distribution of ids, i.e. two groups may have 5 nodes while the third one has no node, leading to routing failures (die-out) when the latter group is used for routing.

To tackle this problem, we propose EIDA (Equitable Id Assignment), a distributed algorithm to assign ids in a best-effort equitable way. EIDA combines ideal and random assignments, where some nodes assign their ids using packet exchanges (like in the ideal assignment) until the required minimum guarantee is fulfilled, and the remaining nodes assign ids randomly without further communication. It works in one-hop nanonetworks, and in multi-hop nanonetworks after division into several independent zones processed

separately by EIDA. It assigns ids based on the node density and two configurable parameters: redundancy and guarantee. While the network density is usually a fact and is fixed, the two parameters can be chosen by the user based on the application's needs. The **redundancy** represents the expected (desired) number of nodes in a group, and the **guarantee** represents the minimum ensured number of nodes in each group (e.g. we need at least 2 nodes in each group).

The evaluation of id assignment is done on electromagnetic nanonetworks, made up of tiny nodes of nanometric size.

The contribution of EIDA is twofold:

- We introduce EIDA, a novel best-effort equitable id assignment mechanism.
- Simulations show a good trade-off between the amount of packet exchanged and the equitability while ensuring a minimum number of nodes in each group, as compared with alternative methods.

The remainder of this chapter is organized as follows. Section 4.2 presents the related work on id assignment methods. Section 4.3 presents the assumptions, formalization, and description of the proposed EIDA mechanism. Section 4.4 shows how EIDA can be applied to multi-hop nanonetworks and the motivation for using it. Section 4.5 evaluates EIDA. Finally, the conclusion is drawn in section 4.6.

4.2/ RELATED WORK

This section first explains the random and ideal assignment mechanisms on which our method is based and then proceeds with other methods. The methods are compared with our EIDA mechanism with respect to the number of packets exchanged (an important parameter in our case of ultra-dense nanonetworks), the guarantee of minimum group size, and the equitability of group sizes.

4.2.1/ RANDOM ASSIGNMENT

Random assignment is a simple mechanism where each node in the network chooses a random number as its id. These numbers are generated by each node in a distributed manner using an RNG (random number generator) Nishimura (2000) with different seeds on each node, i.e. there is no centralized node such as the base station in telecommunication networks that generates different ids for each node.

Usually, an RNG is used to generate random or unpredictable numbers. Instead, in this case, the RNG is used on a large number space to get **non-duplicate numbers**.

The main advantage of the random assignment mechanism is that no packet exchange (communication) is needed for nodes to choose their ids. Conversely, the major drawback is that it is not equitable, since there is no guarantee that nodes will be approximately equally distributed into the groups (when groups are small).

4.2.2/ IDEAL ASSIGNMENT

In the ideal assignment, nodes assign alternatively a unique id between 0 and $n - 1$ (0, 1, 2, ..., $n - 1$). This assignment is considered ideal since node ids are uniformly distributed without requiring a large space number.

One way to achieve this ideal assignment is to start with a random assignment, where each node chooses a random number within an interval much bigger than n to avoid two nodes choosing the same number. Nodes then transform these random numbers into ids by sending packets containing their number using a high backoff to avoid packet collisions. Once all the packets have been sent, all nodes know all the numbers, sort them, and get their ids. For example, if the randomly generated numbers are 58, 23, and 88, nodes will replace them with ids 1, 0, and 2, respectively.

The advantage of the ideal assignment is that it is 100% equitable. For example, using modulo operation generates the same number of nodes in each group (± 1). Its drawback is that it requires (generates) n packet exchanges, where n denotes the number of nodes.

The method presented in this article is a trade-off between random and ideal assignments, where the number of packets exchanges required is much smaller than n , while still providing a configurable guarantee on minimum group size.

4.2.3/ OTHER ASSIGNMENT METHODS

Other id assignment methods include: during the manufacturing process, using tree data structures, and centralized approaches. We explain them and provide their unsuitability for our case below.

Ids can be assigned during the manufacturing process Tian et al. (2008), using a globally unique id per node, such as the MAC address in traditional networks. This approach requires very long ids (to make them globally unique) and extra steps in the manufacturing process. Otherwise, it is equivalent to random assignment in a big space; hence, it has the same advantages and drawbacks.

Assaker et al. (2020) studies the unique node id assignment in modular robots. First, it builds a tree of nodes with a leader and calculates the number of children of every node (nodes below it in the tree). Then, a message is sent at each level, from parent to its

children, based on the number of children already calculated. This process is repeated from parent to child until all modules in the system receive their globally unique ids. This mechanism leads to unique id assignments. The message exchange complexity of this algorithm is high for our purposes, $O(n)$ (order of node density).

SIDA (self-organized id assignment) Lin et al. (2007) is a local and distributed variable-length id assignment mechanism. The id space for the assignment expands in real-time when more nodes enter the network. It builds an overlay binary tree in which each position is mapped to a unique id, which is then assigned to a node. Distant nodes from the sink receive shorter ids, while closer nodes are assigned longer ids. The algorithm runs in three phases: exchange control information between neighboring nodes, broadcast the maximal length to the sink node, and assign the unique ids of nodes. Each node sends 4 messages on average during these phases, which is too high.

Ould-Ahmed-Vall et al. (2005) presents a distributed algorithm to assign unique ids using the minimum number of bytes. It has three phases: create the tree structure and assign temporary long ids, report the size of sub-trees from leaf nodes to the root (using the temporary ids), and assign the unique ids of nodes (each parent node assigns ids to its children nodes, starting from the root). By knowing the size of the network (using phase 2), the initiator can compute the minimum number of bytes required to assign a unique id to each node in the tree. This assignment requires numerous packets to be exchanged, for instance, it ends in about 5 minutes in a network of 1000 nodes Ould-Ahmed-Vall et al. (2005).

The efficient topology discovery protocol, ETDP Zhao et al. (2020), assigns node ids using a tree structure. It establishes the layered network structure by transmitting topology discovery (TD) packets. Nodes are classified into different layers based on the TD packet exchange. The protocol begins with the root node sending a TD packet, and nodes in its communication range that receive it are considered its children. Each node in the network computes its unique id by exchanging id assignment packets with other nodes using network topology information, the father's id, a timer, the number of neighbors, and the layer number. This protocol requires more than $2 * O(n)$ packet exchanges.

Ryu et al. (2011) presents three algorithms: ids swapping assignment, probabilistic simulated annealing assignment, and distributed ids swapping assignment. The first two algorithms are centralized and appropriate for wireless sensor networks with fixed infrastructure, which is not our case; they exchange node ids between two nodes swapped if graph connection rules are respected. The last algorithm consists of four main steps: (1) announce its own node id to its physical neighbors, (2) collect candidate ids that want to be swapped, (3) select the best-fit node id to be swapped, and (4) swap node ids and update each logical neighbor table of its physical neighbors. It is distributed and efficient for ad hoc WSN. This algorithm generates unique but not ideal id assignments, with the

same drawbacks as above.

It is worthwhile to note that Assaker et al. (2020); Lin et al. (2007); Ould-Ahmed-Vall et al. (2005); Zhao et al. (2020) generate unique (i.e. non-duplicate) ids, whereas our objective is to assign ids into different groups. To be able to use this assignment as in our case, we need to consider the number of groups. However, even when applying the modulo operation (%), the generated assignment might be unsatisfactory. For example, using $g = 2$ groups and unique ids 11, 33, 55, all three nodes are assigned to the same group (group 1), since $11\%2 = 33\%2 = 55\%2 = 1$.

4.3/ EIDA IN ONE-HOP NETWORKS

EIDA is a method to assign ids to nodes in a best-effort equitable manner with a guarantee on the minimum. We present it in two steps: this section presents EIDA in a dense one-hop nanonetwork (i.e. each node is in the communication range of any other node), and the next section shows how it can be used in a multi-hop nanonetwork.

This section first outlines the assumptions used by EIDA. Afterward, it formalizes the proposed mechanism, describes it in detail, and finally considers its memory and energy requirements.

As a reminder, EIDA is a combination of the ideal and random assignments and solves the problem of the high number of packet exchanges of the first and the non-equity of the second.

EIDA has the following assumptions:

- Nodes do not appear or disappear in the network during the id assignment process.
- Nodes do not process concurrent packets, i.e. a node cannot send a packet while other packets are in the sending or receiving state. Indeed, some networks allow this type of concurrency, such as nanonetworks, and EIDA does not work if appropriate countermeasures are not taken (a high backoff in our case, as shown later).
- During the algorithm, packets reach the destination without loss.

4.3.1/ FORMALIZATION

In a dense nanonetwork with n nodes, each node is within communication range of all the others. Our goal is to assign a unique id to each node, which allows us to assign every node to a group using a simple modulo operation (groupable id assignments). The

number of groups g is computed based on a configurable parameter r (for redundancy, the expected number of nodes in each group), where $g = \lceil n/r \rceil$ (ceiling operation).

We want to assign each node to one of the g groups. Given that nodes are distributed, they are allowed to exchange packets.

We formalize this mechanism as follows: Let's assume we have a network of n nodes and a redundancy r . We search for a distributed method to divide the nodes into g groups, each with $r = \lfloor \frac{n}{g} \rfloor$ or $\lfloor \frac{n}{g} \rfloor + 1$ nodes, where the symbol $[X]$ denotes the integer part of X . Formally, the method should output $X(n, r, i)$ as the set of nodes in the group g_i so that:

$$|X(n, r, i)| = r \text{ or } r + 1, \quad i = 0, \dots, g - 1 \quad (4.1)$$

where the symbol $|X|$ denotes the cardinality of the set X .

This distribution corresponds to the ideal assignment of ids to nodes (nodes are assigned to groups equitably). However, to achieve this distribution, all nodes need to exchange packets containing their ids (or the group they belong to). Consequently, this approach can be very costly in terms of resources, especially in a dense nanonetwork where n exchanged packets are required.

To avoid the huge amount of packet exchange required, we assume that some applications do not need an ideal assignment but only a minimum guarantee. Thus, we replace the constraint on r (equation 4.1) with another constraint, m , which specifies that each group must have **at least** m nodes (with $m \leq r$). This allows for an assignment as equitably as possible.

This new problem is formalized by the following new equation, which replaces equation 4.1:

$$|X(n, r, i)| \geq m, \quad i = 0, \dots, g - 1 \quad (4.2)$$

For a better understanding, here is a numerical example:

- $n = 100$ nodes
- $r = 5$ redundancy, i.e. the **expected** number of nodes in each group
- $g = \lceil n/r \rceil = 20$ groups
- $m = 2$ guarantee, i.e. the **minimum** number of nodes in each group.

We will see later that the number of packets needed to fulfill the constraint on the minimum is:

$$\text{max_pkts} = n \frac{m}{r} \quad (4.3)$$

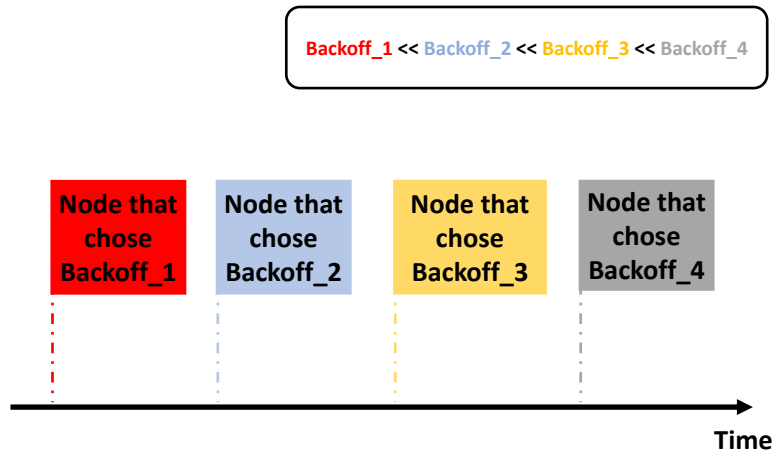


Figure 4.1: Node activation order for Initiating EIDA, ordered by the backoff.

i.e. 40 packets, much smaller than $n = 100$ packets required to fulfill the constraint on full equity (for the ideal assignment).

4.3.2/ EIDA FUNCTIONING

Nodes can start the EIDA algorithm at the same time or at different times. In the latter case, when a node starts the algorithm, the packet it sends will “awake” the other nodes upon receipt, prompting them to start the algorithm as well.

Each node starts by choosing a random backoff in a very large window. The window is large in order to prevent two nodes from starting the algorithm simultaneously and to avoid two packets being received at the same time (because nanonetworks allow this), as shown in figure 4.1. It is proportional to the number of nodes n , the packet size, and the time required for the packet to reach the receiving nodes.

During the backoff waiting period, when a node receives an EIDA packet, it increments its *crt_id* (id counter) value (representing the number of nodes having already assigned their ids), which is initially set to 0. Given that all nodes receive the packet (one-hop nanonetwork) and only one packet is sent at a time, all nodes will come up with the same *crt_id* value.

At backoff expiration, nodes compare the number of packets received with a threshold that is computed from the specified minimum guarantee. If smaller, it chooses its id as the value of *crt_id* and sends a packet to inform all the nodes of its choice of *crt_id*. Elsewhere, it chooses its id randomly without any communication, aiming for best-effort equitability.

Thus, the algorithm consists of two phases: phase 1, where nodes choose their ids incrementally (like in ideal assignment) and send a packet with their id, and phase 2,

where nodes choose their ids using a random value (like in random assignment). As a minor point, in phase 2, instead of generating a new random id, each node can reuse the value of its chosen backoff since this value is already random.

Packets sent during the EIDA algorithm contain two fields: the specific type EIDA not to confound it with other packets in the network, and the id chosen by the sender.

Each node stops the algorithm at the end of its id assignment. Note that the communication between nodes stops when the threshold condition is fulfilled, i.e. when the maximum number of exchanging packets ($max_pkts = mg = mn/r$) is reached, as explained above. At this point, all the nodes without assigned ids at their backoff choose their ids in the addressing space based on their backoff value.

The full algorithm is shown in Algorithm 1.

EIDA is designed to handle packet loss during the assignment process. If a node n_i sends a packet that gets lost, the algorithm continues as usual, and the other nodes continue at their backoff as planned. This ensures that the assignment process remains reliable and guarantees the same results as if there were no packet loss. The only difference is that node n_i will retain its id from phase 1 rather than phase 2, though it is still considered to be randomly assigned in phase 2.

Algorithm 1 ID assignment algorithm.

Input:

n ▷ network density, number of nodes
 m ▷ guarantee, minimum nb of nodes in each group, e.g. 2
 r ▷ redundancy, expected nb of nodes in group, e.g. 5

$g \leftarrow n/r$ ▷ number of groups
 $crt_id \leftarrow 0$ ▷ next id to assign
 $max_pkts \leftarrow m * g$ ▷ max number of packets exchanged
 $x \leftarrow rand()$ ▷ backoff value

wait (x)
if $crt_id < max_pkts$ **then** ▷ phase 1
 $id \leftarrow crt_id \% g$
 send_packet (EIDA, ids)
else ▷ phase 2
 $id \leftarrow x \% g$
end if

Upon packet reception:

if $packet_type == EIDA$ **then**
 $crt_id ++$
end if

4.3.3/ MEMORY AND ENERGY CONSIDERATIONS

At the nanonetwork scale, memory and battery energy are severely limited Lemic et al. (2021).

The memory utilization of nodes is optimized in this algorithm since they only store the current ids, regardless of the network density.

The energy consumption depends, among others, on the number of exchanged packets, which is mn/r (cf. eq (4.3)). Note that only a subset of the nodes exchange packets, while others do not send any, so the energy consumption varies among nodes.

4.4/ APPLICATION TO MULTI-HOP NETWORKS

For EIDA to work in a multi-hop nanonetwork, the network needs to be partitioned into disjoint zones and to apply EIDA in each zone separately. This is the case for networks divided into clusters (each cluster being a separate zone) or GPS (Global Positioning System) zones.

This section discusses the need for an id assignment mechanism in some applications. In addition, we present SLR Tsioliariidou et al. (2017), Stateless Linear-path Routing, a specific protocol used to divide the nanonetwork into zones and for routing. Afterward, we show how to apply EIDA in multi-hop nanonetworks using SLR.

4.4.1/ APPLICATIONS

This section presents the need for an id assignment mechanism in some applications and the motivation to use EIDA for these applications.

In a dense nanonetwork, where numerous nodes act as potential routers, node ids can be used to divide communication bandwidth or other resources, such as memory and energy, among the nodes. For example, without ids, nodes have to communicate by flooding the entire network, while adding the source and destination node ids to packets allows for efficient routing during communication by delivering the packet in a linear path from source to destination. EIDA can be used here to split traffic based on the ids (groups). Consequently, the packet is delivered in a linear path as well, but instead of having all the nodes in the zones (belonging to the traffic) retransmit the packet, only one group in each zone retransmits.

Other applications that use id assignment exist, including monitoring the temperature of a certain area where multiple nodes in the same area can hold the same ids (belong to the same group), object tracking, and location-aware ids where the id assignment of nodes

depends on the coordinate system (hop count) like in phase 1 of SLR (section 4.4.2) Liu and Ni (2006).

A possible application in routing is FR-SLR zone splitting Al Mawla et al. (2023b) (presented in detail in chapter 5) in a multi-zone nanonetwork, where only some of the nodes in the zones of the path participate in the routing process. Each node has an id assigned by EIDA, and a modulo operation decides whether the node forwards the packet or not. For example, consider three nodes, A , B , and C , with ids 1, 1, and 0, respectively, where modulo 2 is used (number of groups) to decide the forwarding. When they receive the packet 0, only node C forwards it, since only $\text{id}[C]\%2 == 0$. Using EIDA avoids transmission die-out as is the case when using random assignment that might assign ids 1, 1, and 1, respectively (this case can appear in $1/2^3 = 1/8$ of cases), so none of the nodes retransmits packet 0 (the modulo always returns 1). In FR-SLR, the parameters n_i , r and m refer to each zone. EIDA is applied independently in each zone. For any transmitted packet, we need $r \leq n_i$ retransmitters in each zone. The retransmitters must not be the same for all the packets but evenly distributed among the n_i nodes, i.e. all the $g_i = \lceil n_i/r \rceil$ groups should be used. In other words, for packet 1, it should choose a group of r retransmitters; for packet 2 it should choose another group of r retransmitters; and so on for all packets.

In summary, EIDA is a useful mechanism that improves the efficiency and reliability of communication and routing. By reducing congestion, improving routing efficiency, and allocating resources more equitably among nodes, EIDA can lead to a more efficient and reliable network that can handle increased traffic and reduce the risk of communication failures.

4.4.2/ NETWORK DIVISION IN ZONES USING SLR

As explained in section 2.3, SLR Tsioliaridou et al. (2017) is a spatial addressing and routing protocol that comprises two phases: initialization and routing. The goal of the initialization phase is to assign coordinates to nodes (the same unique coordinates for nodes in each zone) that are defined as an integer number of hops from the anchors. The goal of the routing phase is to route the data packets from source to destination in a linear routing path based on the coordinates assigned previously, as shown in Fig. 4.2.

4.4.3/ ZONE DENSITY COMPUTATION

To run properly, each node in EIDA needs to know the density of its zone. To this end, we implemented a zone density estimator during the SLR initialization phase. We remind that in this phase, each node in every zone sends one SLR packet, so we take advantage

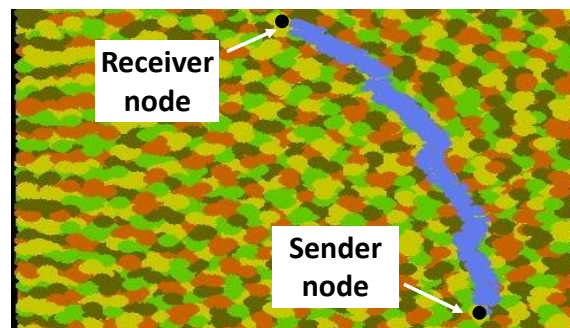


Figure 4.2: VisualTracer sketch for SLR routing phase.

of that and append the sender coordinates to it. Afterward, each time a node receives a packet in this phase, it compares the coordinates of the packet sender with its own coordinates: If they are the same, it increments the density of its zone (sender), as shown in Algorithm 2.

Algorithm 2 Zone density computation

```

ZoneDensity  $\leftarrow$  0
while node receives a packet do
  if packet_type == SLR_setup then
    if sender_XY == receiver_XY then
      ZoneDensity ++
    end if
  end if
end while

```

In this algorithm, *sender_XY* and *receiver_XY* denote the SLR_X and SLR_Y of sender and receiver respectively. *packet_type* denotes the type of received packet (*SLR_setup* in our case), which allows to differentiate it from other packets in the network.

4.4.4/ EIDA ALGORITHM IN MULTI-HOP NETWORK

EIDA algorithm (Algorithm 1) is executed in each zone separately and independently. The algorithm is updated as follows:

1. Packets contain a third field, *slr_xy*, which denotes the sender *slr_x* and *slr_y*, computed as shown in Section 4.4.2: `send_packet (EIDA, ids, slr_xy)`
2. *n* denotes the density of the current zone instead of the network density. This value is computed during the SLR initialisation phase, as explained in Section 4.4.3.
3. Upon packet reception, an additional check is performed on the equality between the *slr_xy* value in the packet (sender) and that of the current node.
if *packet_type* == *EIDA* AND *slr_xy* == *my_xy*

Table 4.1: Simulation parameters.

Size of simulated area	6 mm * 6 mm
Number of nodes	20 000
Communication radius	350 μ m
Packet size	100 bit

To conclude, EIDA mechanism can be applied to multi-zone nanonetworks with slight updates, allowing it to run in each zone separately and independently.

4.5/ EVALUATION

This section evaluates EIDA, in particular the termination and correctness of the algorithm, and compares it with two other assignments using several metrics.

As real experiments are not possible with such a dense nanonetwork, we evaluate our assignment mechanism through simulations. Several nanonetwork simulators exist, as shown in section 2.4.3, but only one, BitSimulator Dhoutaut et al. (2018), is scalable and allows to simulate dense networks Sahin et al. (2021). BitSimulator, presented in section 2.4.4, can simulate tens of thousands of nodes, especially at routing and transport levels. VisualTracer tool is used to display the simulation events.

We implemented in BitSimulator EIDA and zone density computation described in Sec. 4.4.3. All the other protocols were already implemented. The reproduction of the simulation results are presented in appendix A.1.

The simulation parameters are shown in Table 4.1. The nodes are randomly placed in the 2D network using a uniform distribution. We use standard values for TS-OOK modulation: the duration of one pulse (bit) is $T_p = 100$ fs (cf. “very short symbol duration T_p (i.e., ≈ 100 fs)” Jornet and Akyildiz (2014a)), and the time spreading ratio $\beta = T_s/T_p = 1000$ (cf. “The ratio between the time between pulses and the pulse duration is kept constant”).

4.5.1/ EIDA RESULTS

In the simulation, we use $r=5$ (the expected number of nodes in each group) and $m=2$ (the guaranteed minimum number of nodes in each group). A nanonetwork is multi-hop, hence the algorithm is executed by each zone separately and independently. To show the simulation result, we choose a representative zone, of coordinates (31,25). The density of this zone is $n_i=31$, as calculated by Algorithm 2. Given these inputs, $g = 6$ groups, and $max_pkts = 12$ (cf. Algorithm 1).

The results of the assignment are presented in Table 4.2. In this table, the nodes (1, 2,

Table 4.2: Node id assignment simulation results.

Phase 1		Phase 2			
Using packet exchange		Using backoff			
Node backoff order	id	Node backoff order	id	Node backoff order	ids
1	0	13	0	25	3
2	1	14	1	26	1
3	2	15	5	27	0
4	3	16	5	28	0
5	4	17	0	29	1
6	5	18	0	30	0
7	0	19	1	31	3
8	1	20	4		
9	2	21	4		
10	3	22	1		
11	4	23	5		
12	5	24	2		
Continue to 13		Continue to 25			

..., 32) are ordered from the smallest to the highest backoff. The table confirms that only the first $max_pkts = 12$ nodes assign their ids in the ideal case (in phase 1, with packet exchange), because they get increasing ids (0, ..., 5). The simulator took 3.5 seconds to finish the id assignment of the 20 000 nodes.

Figure 4.3 shows the number of nodes that choose each id: using packet exchange (phase 1), using backoff (phase 2), and their sum. In phase 1, the red values show that groups contain an equal number of nodes assigned to them (an equal partition depending on $m = 2$). In phase 2, the blue values show that without communication, the groups are split into a best effort for equitability, but still non-equal partitions (different numbers of assigned ids to nodes in groups). The sum of these two values shows that EIDA does achieve the guarantee (from phase 1 using packet exchange: there are at least $m = 2$ nodes in each group in the ideal case of assignment), i.e. it presents the correctness of the algorithm.

4.5.2/ METHOD COMPARISON

The works cited in Sec. 4.2.3 either assign sparse ids, like the random assignment mechanism (leading to nonequity), or consecutive ids, like the ideal assignment (leading to a high number of packet exchanges). Thus, this section compares the id assignment results of EIDA, random, and ideal assignment mechanisms. The methods are applied on the same zone, which contains 31 nodes.

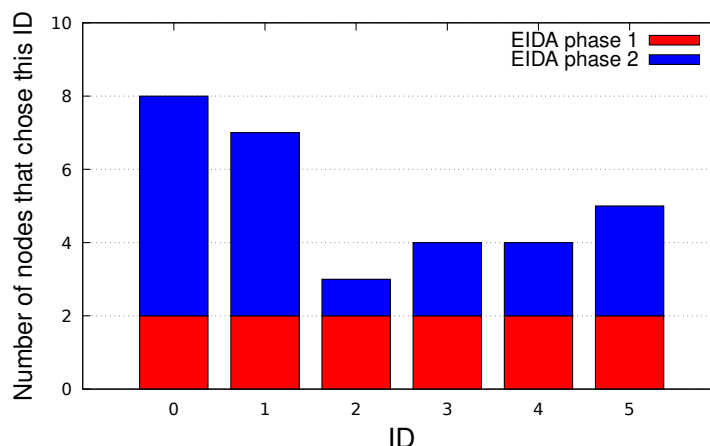


Figure 4.3: Number of nodes that assign their ids to 0, 1, ..., 5 during packet exchange (phase 1 of EIDA), during backoff (phase 2 of EIDA), and after both phases (sum of both phases).

Table 4.3 shows the assignment results. The ideal assignment assigns ids sequentially to nodes and incrementally (0, 1, ..., 5, 0, 1, ..., 5, ...), hence, ids are equitably assigned (5 or 6 nodes per group). EIDA results are taken from Fig. 4.3. Random assignment results are taken from a simple C++ program using the classical Mersenne Twister RNG with `seed=101`.

Table 4.4 compares the three methods in terms of minimum guarantee, number of exchanged packets, and equitability. The equitability uses the least-squares method, calculated by the following equation (where r_i denotes the number of nodes in group i , and $r = 31/6 = 5.1$ their average):

$$S = \sum_{i=0}^5 (r_i - r)^2 \quad (4.4)$$

Random assignment does not provide any guarantee: in Table 4.3 none of the nodes has id =1, which is critical in some applications, such as the routing algorithm presented in Sec. 4.4.1.

The ideal assignment uses $n - 1 = 30$ packets to assign ids to nodes in this zone, whereas EIDA needs 12 packets (*max_pkts*), and random assignment needs none.

Looking at the equitability results in Table 4.3, most groups in the random assignment are far from the ideal assignment, as evidenced by a group with 8 nodes and another one with 0 nodes. In EIDA, each group has at least $m=2$ nodes, and most of the groups are close to the ideal assignment. The least squares result shows that EIDA ($S = 18.8$) is closer to the ideal assignment ($S = 0.8$) than to the random one ($S = 50.8$).

To conclude, EIDA offers a better trade-off than either the random or the ideal assignment

¹`mt19937_64 rng(10); for(i=0;i<31;i++) cout<<rng()%6;`

Table 4.3: Number of nodes in each group after id assignment.

	0	1	2	3	4	5
Ideal	6	5	5	5	5	5
EIDA	8	7	3	4	4	5
Random	7	0	5	3	8	8

Table 4.4: Comparison of ids mechanisms.

	Guarantee ensured	Exchanged packets	Equitability (least squares)
Ideal	yes	$n - 1 = 30$	0.8
EIDA	yes	$nm/r = 12$	18.8
Random	no	0	50.8

mechanisms alone. It guarantees a minimum number of ids per group while reducing the number of packets exchanged.

4.6/ CONCLUSION

This chapter presents EIDA, a mechanism for assigning ids in the context of dense nanonetworks. EIDA aims to assign best-effort equitable ids to nodes in order to create groups of nodes. It is a combination of ideal and random assignments that depends on the configurable desired guarantee m and the number of nodes r desired in each group.

Compared to the ideal assignment, EIDA uses fewer resources by exchanging fewer packets, while assigning partially equitable ids based on the guarantee value, i.e. not fully equitable as in an ideal assignment case. Compared to random assignment, EIDA requires a specific amount of packet exchange to fulfill a minimum guarantee, making it useful in some applications, contrary to random assignment. Evaluations using a dense nanonetwork simulator illustrate the results and benefits of EIDA.

FR-SLR, FORWARDER REDUCTION IN SLR ROUTING PROTOCOL THROUGH ZONE SPLITTING

Current routing protocols are inefficient in dense networks, where nodes have numerous neighbors. Protocols appropriate to dense networks are needed. An example of such a protocol is SLR, which divides the network into zones, with all nodes in a zone sharing the same coordinate. As such, all nodes in a zone participate together in the routing process, leading to the useless consumption of resources. For such networks, we propose FR-SLR, a routing protocol based on SLR, aiming to reduce the number of forwarders in each zone by dividing the nodes into groups and making only one group of nodes forward the packet in each zone. FR-SLR uses an id assignment mechanism to split the nodes in a zone into several groups so that the routing is done per group instead of per zone, thus reducing the number of nodes involved in routing. Simulations on an ultra-dense nanonetwork show the effectiveness of the proposed routing protocol in reducing the number of traffic forwarders.

5.1/ INTRODUCTION

Nanonetwork, presented in section 2.1.1, differs from other types of networks by node resources, energy, and memory capabilities that influence the network lifetime, the huge bandwidth, and the ability to multiplex many frames over the same period of time, among others.

Given the tiny size of nodes, nanonetworks are multi-hop with a high node density, such as hundreds of neighbors per node. The same trend exists in macronetworks, as ultra-dense nanonetworks are emerging as one of the most promising technologies to cope with the tremendously increasing volume of mobile traffic.

In dense nanonetworks, current routing protocols perform poorly, because numerous nodes are involved in the routing process. One such example is Stateless Linear-path Routing (SLR) Tsioliaridou et al. (2017), presented in section 2.3, on which our work is based. It is a spatial addressing and routing protocol.

Given that in SLR all the nodes in a zone share the same coordinates, **all the nodes** in the routing zones retransmit the packet. In this chapter, we propose a method to reduce the number of nodes during packet routing. It works by dividing the nodes in each zone into different groups, using a node id assignment mechanism, and making only one group reforward the packet instead of all the nodes in each zone. Consequently, this reduces the number of packets forwarded, leading to less resource consumption and energy, and increasing the network lifetime.

The contribution of this chapter is to propose FR-SLR, an SLR-based routing protocol efficient in a multi-zone dense network, based on the aforementioned grouping idea. Simulations done on a scalable nanonetwork simulator show a notable reduction in terms of number of forwarders when compared with related forwarding methods.

This chapter is organized as follows. The related work on reducing the number of forwarders is presented in section 5.2. Section 5.3 describes the proposed FR-SLR routing protocol using various id assignment mechanisms. Section 5.4 evaluates FR-SLR using one such assignment and compares it to other flooding and destination-oriented protocols. Finally, the conclusion is drawn in section 5.5.

5.2/ RELATED WORK

This section presents routing protocols that have the same goal as FR-SLR, which is to deliver the packet from source to destination with a reduced number of forwarders. We group them based on the approach they use.

5.2.1/ CLUSTER-BASED APPROACHES

Cluster-based approaches first divide the network into clusters and afterward route packets using cluster heads.

To create clusters, Park et al. (2020) bundle adjacent small cell base stations into a group. Clustering in this paper is done in two steps, (1) cluster splitting/creation and (2) combining them with a weighted k-means algorithm. The k-means group the small base stations into k clusters according to their distances (location and traffic load of the sub-cluster). Zahedi et al. (2016) proposes a swarm intelligence-based fuzzy routing protocol to generate balanced clusters over the network with the ability to determine the precise number

of clusters by considering the residual energy, distance to the sink, and distance from the cluster centroid to select cluster heads.

Routing is done exclusively by cluster heads (CH), and each cluster has a cluster head. This consumes the energy of the CH and creates a load balancing problem. On the contrary, in FR-SLR, a **different** group of nodes is chosen each time in each zone to retransmit the packet. For example, Xu et al. (2019) proposes a hierarchical clustering method where the nodes in the cluster can transmit data to the CH in one hop. Then, the CH nodes transfer data to the nanocontroller (NC) through multi-hop, which has a larger volume and stronger calculation processing capability than the nanonode. It solves the common problem of balancing the energy consumption in the network by continuously updating CH, where the node with the highest residual energy is selected as CH. In this paper, the energy of NC is not limited, which is not the same case as in a dense nanonetwork. As another example, Rikhtegar et al. (2017) introduces a fuzzy logic-based mobility management solution for mobile cluster-based WNSNs to deliver the data packet to static NC. The communication between the nanosensor and the NC of its cluster is done in one hop. The decision to select the NC (stationary nodes) is based on three criteria: the distance of mobile nanonode from NC, residual energy, and the traffic load of this NC. Subsequently, the nanocontroller assigns specific slots for data transmission. Afterward, every nanonode, according to the TDMA schedule, transmits packets to its parent nanocontroller, which then aggregates and forwards them to the nano-micro interface. Yet this protocol cannot be adopted in dense networks because of its computational complexity, the large memory needed, and the need for stationary nodes (NC) distributed within the network.

5.2.2/ FLOODING PROTOCOLS

Pure flooding is the simplest routing method, where every node in the network forwards once each received packet. This flooding generates a significant amount of messages through the network, resulting in broadcast storms in dense networks.

In probabilistic flooding, nodes forward the packet with a certain probability Reina et al. (2015). The chosen probability can be fixed and can depend on several factors, such as density, distance, and speed. The probability needs to be tuned to prevent broadcast storms and guarantee message delivery.

Backoff flooding is a high-quality flooding that uses a counter to count the copies received and a waiting time called "backoff" for the packet forwarding Arrabal et al. (2018a). The backoff value is selected randomly from the backoff window. This window is very large and is proportional to the neighbor density. A node forwards the packet only if it has not received r (redundancy) copies of the same packet during the backoff time.

5.2.3/ DESTINATION-ORIENTED PROTOCOLS

In flooding methods, the packet is flooded into the whole network instead of being transmitted toward the destination, leading to useless packet reception and resource waste (e.g., bandwidth and energy). Destination-oriented protocols, such as SLR, counter-based SLR, and Maze-routing, reduce packet dissemination in unnecessary directions and reduce the number of forwarders.

SLR has been presented in the introduction. Counter-based SLR combines SLR with backoff flooding: A node that receives a packet forwards it only if it has seen fewer copies of the packet than a given threshold.

Maze-routing Fattah et al. (2015) is a distributed routing algorithm that guarantees delivery or indicates that the destination is unreachable by adding extra fields to a message. It allows routing around fault regions without requiring nanonodes to store any information other than their coordinates or the need for a routing table (close to the idea of SLR). Yet, the route may be far from optimal, whereas FR-SLR uses the optimal path (much shorter in some cases).

5.2.4/ SLEEPING-BASED APPROACHES

Another way to reduce the number of forwarders is to use a sleeping mechanism, also known as duty cycling, which turns sensor nodes on and off when necessary. In such a mechanism, a node always wakes up long enough to allow the reception of one or more frames. Afterward, it goes back to sleep by turning off most of its reception and processing capabilities. For example, in cluster synchronization Wang et al. (2016), sensor nodes are grouped into synchronized clusters. In the same cluster, sensor nodes wake up or go to sleep at the same time. But clusters act together with others asynchronously. In Wang et al. (2016), all the decisions are made by the controller, which is usually connected to a constant power supply, which is not appropriate for dense networks.

Furthermore, the sleeping mechanism can be decentralized; each node has its own wake-up and sleep schedule. In Ye and Zhang (2017), the time axis is divided into a number of time slots, and each node autonomously decides to sleep, listen, or transmit in a time slot. The decision is based on its current situation and an approximation of its neighbors' situations without the need for communication with neighbors.

SERENA Minet et al. (2010) is a decentralized node activity scheduling algorithm based on three-hop coloring. It allows nodes to sleep while ensuring end-to-end communication by assigning time slots to nodes. Slots are assigned to nodes based on their color. Any node stays awake only during its slots to transmit and the slots assigned to its 1-hop neighbors to receive their messages; it sleeps the remaining time. Two nodes that are 1,

2, or 3-hop neighbors have different colors. Hence, color is reused four hops away.

FR-SLR is a destination-oriented protocol, which is why, for a fair comparison, the proposed protocol is compared to destination-oriented protocols (SLR and counter-based SLR) and not only to flooding (pure, backoff, and probabilistic) protocols. This comparison takes place in section 5.4.2 and uses the same scenario.

5.3/ FORWARDER REDUCTION IN SLR ROUTING (FR-SLR)

We recall that SLR has a big drawback: during routing, **all** the nodes in the zones on the path from source to destination retransmit the packet, as shown in Fig. 2.8. In ultra-dense networks, this consumes a lot of energy and reduces network lifetime. FR-SLR aims to avoid this resource waste and is described in the following.

5.3.1/ FR-SLR DESCRIPTION

In FR-SLR, nodes from each zone are divided into g groups; all the nodes in a group have the same id, and groups have different ids. Each packet has an additional field specifying an id, initialized by the source of the packet with its own id. In each SLR zone on the routing path, only the nodes whose id is the same as the id field in the packet retransmit the packet (instead of all of them like in SLR).

Node ids are selected using any id assignment mechanism, provided that the following condition is met: ids start from 0 and follow in sequence, without holes, i.e. 0, 1, 2, ..., g , where g is chosen by the user and given as input in the assignment mechanism. g value is known by all the nodes. We will present a few such mechanisms in the next section.

Users can select g based on multiple factors, such as the density of zones, zone size, and application. If the user needs, for example, 5 forwarders in each zone and the zone density is 10 nodes, the user must set $g = 2$, whereas if zone density is 100, the user must set $g = 20$. Additionally, for example, if the zone space is small and all the nodes are within the communication range of each other, a small g can be chosen, whereas if zones are large, then g needs to be bigger to prevent die out. Also, critical applications might choose a small value for g in order to have more nodes retransmit the packet.

We note that, as it is, the method leads to a load balancing problem in some cases. For example, in a scenario where a source node sends 1000 packets, all the packets contain the same id, hence **the same group** in each zone of the path reforward the packet. To avoid this case, FR-SLR nodes also use the packet sequence number to decide the group (provided that packets have such a field), so that packets from the same source

are processed by **different** groups.

The full algorithm is shown in Algorithm 3. `is_on_path()` is the SLR function that determines whether the current node is on the path between the source and destination based on their coordinates.

FR-SLR differs from SLR by adding an additional test (condition). This test compares the id equality of the packet sender to the id of the current host. Thus, the current host only forwards the packet if this condition is met, i.e., if they are holding the same id, which means that they belong to the same group.

Algorithm 3 FR-SLR pseudocode

```

Upon packet pkt reception in node i:
coord_i ← get_coord(node_i)
coord_src ← get_source_coord(pkt)
coord_dst ← get_destination_coord(pkt)
if is_on_path(coord_i, coord_src, coord_dst) then
    src_id ← get_source_id(pkt)
    id ← get_id(node_i)
    seq_no ← get_sequence_nb(pkt)
    if (src_id + seq_no) mod g == id then
        forward_packet(pkt)
    end if
end if

```

To conclude, in FR-SLR, only a part of the nodes in each zone on the path retransmit the packet based on node id.

5.3.2/ INTEGRATION OF FR-SLR WITH EXISTING ID ASSIGNMENT MECHANISMS

FR-SLR uses node ids to divide the forwarders (reduce the number of forwarders) of each packet. In this section, we recall various mechanisms to assign each node an id, taken from previous chapter, and show why we chose EIDA.

5.3.2.1/ RANDOM ID ASSIGNMENT

In the random id assignment, each node assigns itself a random id from 0 to $g - 1$. An example of random assignment is shown in Table 5.1. Its results are taken from a simple C++ program using the classical Mersenne Twister RNG with `seed=10`¹. This table also shows that the number of nodes in the group of `id=1` is zero. Thus, when a packet is sent by a sender with $src_id + seq_no = 1$ and reaches a zone where group 1 is empty,

¹`mt19937_64 rng(10); for(i=0;i<31;i++) cout<<rng()%6;`

Table 5.1: Number of nodes in each group after assignment.

	0	1	2	3	4	5
Ideal	6	5	5	5	5	5
EIDA	9	4	5	4	5	4
Random	7	0	5	3	8	8

the packet is not forwarded, leading to a die-out. We will therefore not use the random assignment in the evaluation.

5.3.2.2/ IDEAL ID ASSIGNMENT

The ideal id assignment divides the nodes into g groups, each with the same (\approx) number of nodes, $\lfloor \frac{n}{g} \rfloor$ or $\lfloor \frac{n}{g} \rfloor + 1$ nodes. Thus, the partition is $\approx 100\%$ equitable, as shown in Table 5.1. However, to obtain this equal partition, the number of packet exchanges in each zone is $n - 1$, a big number in ultra-dense networks, hence we will not use this assignment in our evaluation.

5.3.2.3/ EIDA ASSIGNMENT MECHANISM

EIDA, presented in chapter 4, partitions several nodes into groups of the same id Al Mawla et al. (2023a). It assumes that nodes can communicate with each other. Thus, multi-hop networks need to be partitioned into zones first, and afterward, EIDA can be used in each zone separately and independently. SLR does such a partitioning in the initialization phase, and EIDA needs to start after the SLR initialization phase.

EIDA provides two configurable parameters, r and m , where r is the expected number of nodes in each group and m the minimum number of nodes in each group (with $m \leq r$). EIDA assigns each node to one of the g groups (0, 1, 2, ..., $g - 1$) in a best-effort equitable manner, respecting the condition on m . The number of groups g is computed as $g = \lceil \frac{n}{r} \rceil$ (ceiling operation), where n denotes the density of the current zone. EIDA results are shown in Table 5.1.

EIDA has a useful property in our case. Given that any group has at least m nodes (as shown in Table 5.1), during routing each packet is forwarded by at least m nodes in each zone, thus preventing the die-out problem. We will therefore use EIDA to assign node ids for FR-SLR evaluation.

It should be noted that EIDA has a limitation affecting FR-SLR: when combined with EIDA, FR-SLR does not work in **heterogeneous** networks. The reason is that, contrary to the other assignment methods, EIDA does not use a fixed g value as input but instead

Table 5.2: Simulation parameters.

Parameter	Value
Size of simulated area	6 mm * 6 mm
Number of nodes	20 000
Communication radius	285 μ m
Packet size	100 bit

computes g based on m and r . In heterogeneous networks, where zones have different densities, the g value can be different, otherwise said, some zones can have e.g. 5 groups (50 nodes divided by $r=5$), others 2 groups (20 nodes divided by $r=5$). Thus, in the case where the source is in group 3, the packet will have id 3 and when crossing zones with only 2 groups, no node exists with group 3, hence the packet is not retransmitted and a die-out occurs.

5.4/ EVALUATION

This section evaluates the FR-SLR routing protocol with the EIDA mechanism and compares it to other routing protocols. Furthermore, the motivation for this chapter is revealed when the simulation compares the number of packet exchanges using SLR and FR-SLR.

An example of dense networks is nanonetworks. They currently use the TS-OOK, presented in section 2.3, which is based on femtosecond-long pulses in the terahertz band, appropriate to the very limited energy of nanonodes.

As real experiments are not possible with such a dense network, we evaluate our routing protocol through simulations. Implementation and evaluation of the FR-SLR algorithm are done using BitSimulator Dhoutaut et al. (2018), the only simulator allowing simulation of ultra-dense nanonetworks (tens of thousands of nodes), as presented in 2.4.

We implemented FR-SLR forwarding in BitSimulator. The other routing protocols were already included in the simulator.

The simulation parameters are shown in Table 5.2. The nodes are placed randomly in the 2D network, using a uniform distribution, and are static. We use standard low-level parameters for TS-OOK modulation, i.e. duration of one pulse (bit) $T_p = 100$ fs Jornet and Akyildiz (2014a)), and the time spreading ratio $\beta = T_s/T_p = 1000$ (cf. “The ratio between the time between pulses and the pulse duration is kept constant”).

The reproduction of the simulation results are presented in appendix A.2.

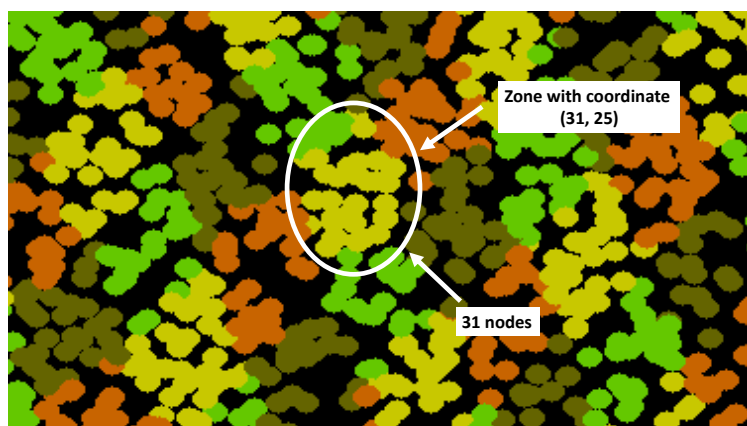


Figure 5.1: VisualTracer sketch for the chosen zone, of coordinates (31,25).

5.4.1/ DIE-OUT AVOIDANCE IN FR-SLR WITH EIDA

This section shows that FR-SLR/EIDA avoids the die-out, i.e. packets reach the destination.

As a node id assignment mechanism, we use EIDA because it uses fewer packet exchanges than the ideal assignment, and the random assignment can lead to die-out, as shown in Sec. 5.3.2. EIDA parameters are $r=5$ (expected number in each group) and $m=2$ (minimum number of nodes in each group). An example of an assignment is presented in the previous Table 5.1, for the zone of coordinates (31,25), shown in Fig. 5.1, with 31 nodes. The table confirms that EIDA guarantee of m is achieved because each group contains at least $m = 2$ nodes.

In the scenario, we chose a sender with coordinates (35,25) and a receiver with coordinates (16,29). The simulation uses the first packet, hence $seq_no = 0$.

In the FR-SLR simulation, Table 5.3 presents the number of forwarders in each zone that belong to the transmission path between the sender (35,25) and receiver (16,29). This table shows that for each zone, the number of forwarders is greater than the guarantee ($m = 2$). Thus, FR-SLR inherits from EIDA the guarantee of minimum group size, so no die-out can occur.

To conclude, FR-SLR with EIDA avoids the die-out.

5.4.2/ COMPARISON OF FR-SLR/EIDA TO OTHER ROUTING PROTOCOLS

In this section, we compare FR-SLR/EIDA to five routing protocols: SLR, counter-based SLR, pure flooding, backoff flooding, and probabilistic flooding. The comparison is based on the number of forwarders using the aforementioned scenario.

Counter-based SLR uses a redundancy of $r=5$, similar to FR-SLR (for a fair comparison).

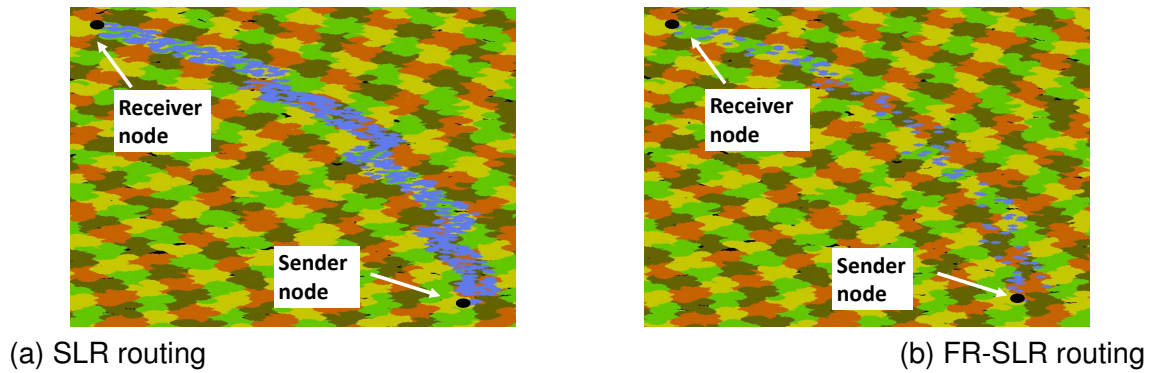


Figure 5.2: VisualTracer sketch for SLR and FR-SLR routing phase.

Table 5.4 presents the number of forwarders in each routing protocol. In SLR, there are 619 forwarders, because **all the nodes** in a zone on the path reforward the packet. In FR-SLR, since only **one group** of nodes in each zone retransmits the packet, there are 116 forwarders, as can be computed by summing up the numbers in the last column in Table 5.3 (+1 for the first packet). The forwarders in both cases are presented as blue points in Fig. 5.2. Counter-based SLR has a comparable number of forwarders (a bit higher in this case).

In a zone with a density of around 30 nodes and a parameter of $r = 5$ for FR-SLR, there are $\lceil \frac{30}{5} \rceil = 6$ groups in this zone. Therefore, $\frac{1}{6}$ of the nodes (one group) in each zone reforward the packet. If the linear path between the sender and receiver contains 23 zones, the total number of forwarded packets in FR-SLR should be reduced by a factor of 6 compared to SLR. This is proved in Table 5.4, where the number of forwarded packets of FR-SLR is approximately equal to $\frac{1}{6}$ of the number of forwarded packets of SLR.

Table 5.4 shows the time to deliver the packet from source to destination (the delay). FR-SLR is much faster than counter-based SLR and backoff flooding (because these protocols need a large backoff during each forwarding). The difference in time, of thousands of ns, i.e. 1 μ s, seems small because it is at a small scale of 23 zones between source and destination, but on longer paths, the difference becomes bigger.

5.4.3/ DISCUSSION ON FR-SLR VS COUNTER-BASED SLR

Given that counter-based SLR and our FR-SLR method have a comparable number of forwarders, in this section we compare them more thoroughly. Table 5.5 compares both protocols analytically.

In FR-SLR, if all the nodes that are chosen to retransmit the packet (based on the group id) fail, a die-out occurs. On the contrary, in counter-based SLR, no die-out occurs since nodes are chosen randomly each time. We plan to improve on this in future work.

FR-SLR, given that it uses the id of nodes, requires an id initialization mechanism before the routing protocol starts, as opposed to counter-based SLR, where no initialization phase is needed.

For each packet forwarding, the counter-based SLR protocol uses a large waiting time (backoff), whereas in FR-SLR there is no backoff since nodes simply compare the id of the packet sender and the current host (itself) to decide whether to forward the packet or not.

In the counter-based SLR protocol, redundancy applies to the communication range, which can contain one or multiple zones, whereas in FR-SLR redundancy applies per zone, i.e., in each zone r nodes reforward the packet.

The redundancy has many consequences on die-out. Some die-out problems appear in counter-based SLR, depending on the communication ranges used in the initialization and in routing SLR phases. The same communication range (220) for both phases results in a die-out problem approximately in the middle of the transmission path. The cause is that the redundancy is used in the communication range area, not the zone area as in FR-SLR. Hence, when the communication range of the nodes responsible for transmitting the packet does not propagate to the next zone in the transmission path, a die-out occurs. The same problem appears if the difference between the two communication ranges is small; for example, using $r=1$ and (220, 250) for the (initialization, routing) communication ranges, a die-out occurs again since the communication range of the node responsible for retransmission is too small to reach the next zone.

To conclude, the proposed routing protocol FR-SLR uses generally fewer forwarder packets and smaller time to deliver packets compared to the mentioned flooding (pure flooding, probabilistic flooding, backoff flooding) and destination-oriented protocols (SLR, counter-based SLR), albeit the communication ranges of nodes play an important role in this comparison.

5.5/ CONCLUSION

This chapter introduces FR-SLR, an enhanced SLR routing protocol, using the EIDA mechanism (a combination of ideal and RNG assignments) in the context of dense networks. It reduces the number of forwarders during packet routing by grouping nodes in each zone. Grouping is done using the id assignment method.

Evaluations are done using a dense nanonetwork simulator and illustrate the benefits of FR-SLR when combined with EIDA as an assignment mechanism. FR-SLR outperforms the compared protocols, either flooding or destination-based ones, in terms of the number of forwarders and time to deliver packets.

Table 5.3: Number of forwarders in each zone in FR-SLR.

Zone	SLR coordinates	Number of forwarders
0	35, 26	6
1	34, 26	10
2	33, 26	6
3	32, 26	2
4	31, 26	4
5	31, 27	7
6	30, 27	4
7	29, 27	3
8	28, 27	6
9	27, 27	4
10	26, 27	4
11	26, 28	5
12	25, 28	4
13	24, 28	3
14	23, 28	5
15	22, 28	2
15	21, 28	4
17	21, 29	6
18	20, 29	9
19	19, 29	6
20	18, 29	5
21	17, 29	5
22	16, 29	5

Table 5.4: Comparison of various routing protocols.

Method	Number of forwarders	Simulation time (ns)
Flooding methods:		
Pure flooding	20 004	158
Probabilistic flooding	718	277
Backoff flooding	575	1544
Destination-oriented methods:		
SLR	619	168
Counter-based SLR	147	5361
FR-SLR	116	198

Table 5.5: FR-SLR vs counter-based SLR.

Property	FR-SLR	Counter-based SLR
Works in node failures	no	yes
Initialization phase	yes	no
Backoff during routing	no	yes
Redundancy	per zone	per communication range

NETWORK PARAMETER INFLUENCE ON COMMUNICATIONS IN DENSE WIRELESS NANONETWORKS

This chapter evaluates how various network parameters impact communication quality in a dense electromagnetic nanonetwork Al Mawla and Dedu (2024). The parameters studied are β (equal to the ratio of time between two consecutive pulses sent and the pulse length), communication range, node density, and pulse duration. The quality is measured by packet collisions, receptions, emissions, and deliveries to destination. The evaluation considers homogeneous and heterogeneous networks, and single and multiple packets per flow. Simulation results show how these parameters influence network communication quality; for example, increasing β reduces collisions and increases receptions, deliveries, and emission rates up to an optimal threshold, beyond which further increases in β have no significant effect. These insights provide guidelines for selecting appropriate network parameters.

6.1/ INTRODUCTION

In this chapter, we consider dense nanonetworks, in which nodes have a high number of neighboring nodes. Some challenges of dense nanonetworks are to ensure reliable communication and to reduce the interference and collision rates due to the high node density. Addressing these challenges requires a comprehensive understanding of how various network parameters influence overall network communication quality.

Network communication quality can be measured by several metrics, such as the number of collided, received, delivered, and transmitted packets. Network protocols have a high influence on these metrics. In this chapter, we demonstrate that even within the same protocol, outcomes can vary significantly due to different parameter values. The param-

eters we consider in this chapter are β , node density, communication range, and pulse duration. We recall that in nanonetworks, β , i.e. the symbol rate, represents the ratio of time between two consecutive pulses and the pulse length, pulse duration represents the pulse length, communication range defines the effective distance over which communication occurs, and node density is the number of neighboring nodes per communication range.

This chapter contributes by revealing the impact of network parameters on packet collisions, receptions, emissions, and deliveries. This result can be used as a guideline for selecting appropriate network parameters in simulation studies.

The remainder of this chapter is organized as follows. Section 6.2 presents the related work. Section 6.3 presents how choosing network parameters such as β , communication range, density, and pulse duration influences network communication quality like collisions, emissions, reception, and delivery. Finally, the conclusion and future perspective are drawn in section 6.4.

6.2/ RELATED WORK

Hoteit et al. (2023) discusses the influence of β and source packet rate on nanonetwork performance, concluding that higher β reduces the number of collisions. In contrast, we show that β has an optimal value (beyond which collisions do not decrease anymore) and give examples on how to find it, and evaluates the effect of other parameters too (communication range, pulse duration, and node density).

Yu et al. (2015) proposes a channel-aware forwarding scheme in nanonetworks, where data is forwarded either to all nodes (one-hop transmission) or only to neighboring nodes (multi-hop transmission). To achieve that, nodes use frequencies known to have small or big attenuation in the given transmission channel. They conclude that multi-hop forwarding yields higher capacity than single-hop transmission, while maintaining comparable delay. In contrast to this chapter, we discuss the effect of various parameters (including various communication ranges) to collisions, receptions, and emissions in multi-hop scenarios.

In underwater acoustic multi-hop, Jouhari et al. (2019); de Souza et al. (2016) show that reducing the number of hops (through larger communication ranges) significantly reduces energy consumption per bit. Instead, we do not consider energy, but network metrics (collisions, receptions, etc.) using various parameters (communication range, density, etc.)

In this chapter, we consider node density as a parameter. The density is also studied in Garg et al. (2013), which describes the relationship between density and dissemination

time, which denotes the duration for data to spread/share information throughout the entire network. It concludes that beyond a certain threshold of node density, the impact of node density does not significantly affect the data dissemination time, without compromising the achieved data quality. However, if the node density is smaller, then few people exchange data, leading to a slower data dissemination process.

Other papers also discuss various parameters and aspects that affect network behavior. For instance, Toh et al. (2002) shows that end-to-end performance is influenced by route length, communication throughput depends on packet size and route length, route discovery time is affected by channel conditions and route length, and packet loss is not significantly affected by packet size or route length.

6.3/ NETWORK PARAMETER INFLUENCE ON COMMUNICATIONS

In this section, we aim to explore and analyze the impact of various parameters on communications.

Our analysis needs ultra-dense nanonetworks with tens of thousands of nodes and several flows. Due to the impracticality of conducting real experiments in such a dense network and to the complexity, we need to resort to simulations. As previously, we will use BitSimulator.

6.3.1/ BASE NETWORK SCENARIO USED FOR SIMULATIONS

The simulations involve a 2D network with nodes randomly placed using a uniform distribution. The nodes remain static throughout the simulation.

Simulations were conducted on the SLR routing protocol. SLR Tsioliariidou et al. (2017) is a spatial addressing and routing protocol that comprises two phases: setup to assign coordinates to nodes and routing to route the data packets from source to destination in a linear routing path based on the coordinates assigned previously, as shown in Fig. 2.8 and detailed in section 2.3.

We consider two node placements, a homogeneous one where all the nodes are placed in the whole network, and a heterogeneous one divided in six parts in a grid as shown in Fig. 6.1; for example, for 10 000 nodes, the six parts have 2 000, 1 500, 1 250, 1 000, 2 500 and 1 750 nodes, respectively.

Unless otherwise stated, the parameters used are shown in Table 6.1. We use the standard duration of pulse for TS-OOK modulation (presented in chapter 2.2) $T_p = 100$ fs (cf. Jornet and Akyildiz (2014a)).

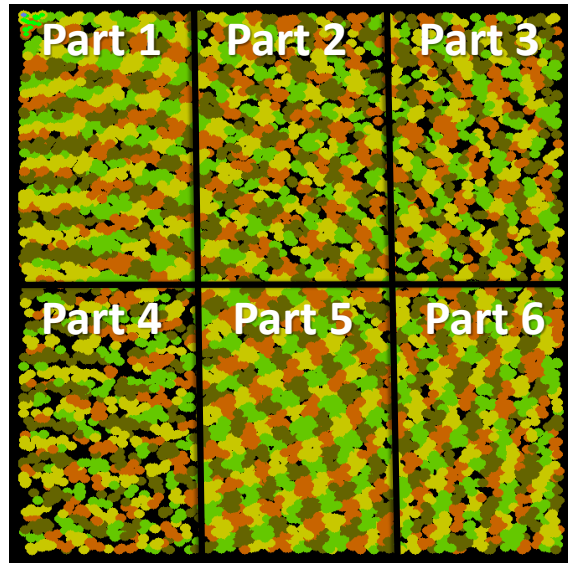


Figure 6.1: Heterogeneous network used in simulations.

Simulations are executed on two numbers of packets, 1 and 20 packets per flow (chosen randomly). There are 9 distinct flows, illustrated in Fig. 6.2. They use SLR routing protocol, described in Sec. 2.3.

To ensure the reproducibility of results and enhance trust in our study, all the presented results are easily reproducible through a shell script, provided, along with all the related information, in appendix A.3.

6.3.2/ IMPACT OF β

This section evaluates the impact of the parameter β (defined as T_s/T_p , where T_s is the time between two pulses, and T_p the pulse duration) on the communications, namely the number of packet collisions and packet deliveries at destination, and also on packet emissions and receptions at any nodes.

To allow fairer comparisons between different values of β , we vary β only during the routing phase, and keep it constant (equal to 1 000) during the SLR setup phase, so that the SLR zones are the same.

We evaluate a dozen values of β , chosen empirically. Table 6.2 presents only five representative values, allowing to discover the optimal β : a small value, three values close to the optimal β , and a big value.

The results in the homogeneous network are shown in Table 6.2. They show **two intervals**. In the first, called **improvement interval**, increasing β results in a reduction in packet collisions up to a certain point. In the second interval, called **stable interval**, increasing β does not influence packet collision. The value in between, called **optimal β** , marks the be-

Table 6.1: The parameters used in the evaluation (unless otherwise stated).

Parameter	Value
Size of simulated area	$6\,000 \times 6\,000 \mu\text{m}^2$
Number of nodes	10 000
Communication range for the SLR setup phase	$220 \mu\text{m}$
Communication range in routing	$285 \mu\text{m}$
β for the SLR setup phase	1,000
β in routing	100
Pulse duration	100 fs
Packet size	100 bit

gining of stabilization. In the following, we detail both intervals, and discuss the optimal value.

During the **improvement interval**, as the value of β increases, the occurrence of packet collisions decreases, resulting in a reduction in collision events. For example, when considering 1 packet per flow and 10 000 nodes (first column in the table), the number of packet collisions decreases from 79 096 to 8 307 when β increases from 50 to 600. Similarly, for 20 packets per flow and 30 000 nodes, the number of packet collisions decreases from 26 497 920 to 4 953 223 when β increases from 50 to 800. This correlation between β and collisions can be explained by the nature of β , defined as T_s/T_p . Increasing β essentially means increasing T_s , which, in practical terms, translates to increasing the time spacing between two consecutive pulses, thereby allowing more flows to access the channel without collision. This relationship underscores the importance of optimizing β to minimize packet collisions.

As β increases and the number of collisions decreases, more packets are correctly received, hence the number of receptions increases. Indeed, in the table, for 20 000 nodes and 1 packet per flow, the number of received packets increases from 342 677 to 812 905 as β rises from 50 to 1 000; similarly, for 40 000 nodes and 20 packets per flow, the number of packets delivered (received at the destination zone) increases from 2 140 to 10 622 when β rises from 50 to 1 000.

More received packets generate more retransmitted packets. So, increasing β leads to an increase of emissions across all nodes, particularly visible in the dense network shown in the table.

During the **stable interval**, occurring when β is large, such as 1 000 and 50 000, the number of packet collisions remains the same, as exemplified for 30 000 nodes where it stabilizes at 315 738 in 1 packet per flow and 4 953 223 in 20 packets per flow. This stability is due to the probability of collision. For example, when sending or receiving 10 flows with $\beta = 800$, the probability of collision is very low, close to 0. Further increasing β to 1 000 does not change anything, i.e. the probability remains close to 0. Indeed, the probability

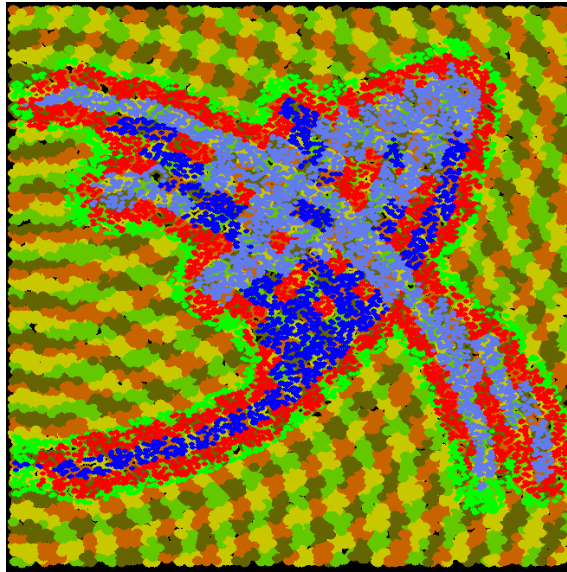


Figure 6.2: The network and the flows used in evaluation.

of zero collision can be calculated as the product of the probabilities of each packet not colliding (1 minus probability of zero collision):

- probability of collision-free transmission for $\beta=800$: $\frac{799}{800} \times \frac{798}{800} \times \frac{797}{800} \times \frac{796}{800} \times \frac{795}{800} \times \frac{794}{800} \times \frac{793}{800} \times \frac{792}{800} \times \frac{791}{800} = 0.94$
- Probability of collision-free transmission for $\beta = 1000$: $\frac{999}{1000} \times \frac{998}{1000} \times \frac{997}{1000} \times \frac{996}{1000} \times \frac{995}{1000} \times \frac{994}{1000} \times \frac{993}{1000} \times \frac{992}{1000} \times \frac{991}{1000} = 0.95$

Thus, the probabilities of collision in scenarios with $\beta = 800$ and $\beta = 1000$ are $1 - 0.94 = 0.06$ and $1 - 0.95 = 0.05$ respectively, which are nearly the same.

Given that the collision rate does not change, the number of received and emitted packets does not change either, as it can be seen in the table, for instance in one packet per flow and 10 000 nodes, when collision stabilizes at 8307, the rate of packet reception and emission also remain stable at 195 912 and 2 828 respectively.

Concerning the **optimal value of β** , it varies according to the number of nodes and the number of packets transmitted (it increases when the number of nodes or number of packets sent increases). This variation is illustrated in Table 6.2: for scenarios with 1 packet per flow, the optimal β is 600 for 10 000 nodes, and 800 for 20 000, 30 000, and 40 000 nodes. Conversely, in scenarios with 20 packets per flow, the optimal β is 800 for 10 000 and 20 000 nodes, and 1 000 for 30 000 and 40 000 nodes.

To check the existence of the intervals and whether the optimal β is constant across other protocols, we conduct the same series of simulations in the probabilistic flooding (for 1 packet per flow). In probabilistic flooding, nodes forward packets with a predefined

Table 6.2: Simulation results for various β values (and various number of nodes).

C:Collisions, R:Receptions, E:Emissions, D:Deliveries

	1 packet per flow				20 packets per flow			
	C	R	E	D	C	R	E	D
10 000 nodes								
$\beta=50$	79 096	120 532	2 763	96	1 604 771	2 409 555	55 589	1 734
$\beta=600$	8 307	195 912	2 828	157	224 523	3 865 111	56 627	3 022
$\beta=800$	8 307	195 912	2 828	157	224 515	3 865 119	56 627	3 022
$\beta=1\ 000$	8 307	195 912	2 828	157	224 515	3 865 119	56 627	3 022
$\beta=50\ 000$	8 307	195 912	2 828	157	224 515	3 865 119	56 627	3 022
20 000 nodes								
$\beta=50$	513 624	342 677	5 976	103	10 251 368	6 529 716	117 131	1 916
$\beta=600$	62 614	812 850	6,106	270	1 491 852	16 057 697	122 342	5 385
$\beta=800$	62 559	812 905	6 106	270	1 683 221	15 875 253	122 399	5 345
$\beta=1\ 000$	62 559	812 905	6 106	270	1 683 221	15 875 253	122 399	5 345
$\beta=50\ 000$	62 559	812 905	6 106	270	1 683 221	15 875 253	122 399	5 345
30 000 nodes								
$\beta=50$	1 419 223	527 299	9 099	121	26 497 920	10 480 326	173 083	2 158
$\beta=600$	315 845	1 704 604	9 445	467	5 009 060	35 160 560	187 820	10 317
$\beta=800$	315 738	1 704 711	9 445	467	4 953 301	35 422 690	188 761	10 243
$\beta=1\ 000$	315 738	1 704 711	9 445	467	4 953 223	35 422 768	188 761	10 243
$\beta=50\ 000$	315 738	1 704 711	9 445	467	4 953 223	35 422 768	188 761	10 243
40 000 nodes								
$\beta=50$	2 525 690	725 299	11 346	117	49 626 133	14 521 433	224 045	2 140
$\beta=600$	570 410	2 668 848	11 280	536	10 395 777	59 168 229	242 766	10 631
$\beta=800$	562 700	2 676 558	11 280	541	10 845 635	59 006 143	243 780	10 610
$\beta=1\ 000$	562 700	2 676 558	11 280	541	10 843 521	59 008 257	243 780	10 622
$\beta=50\ 000$	562 700	2 676 558	11 280	541	10 843 521	59 008 257	243 780	10 622

probability, discarding them if the probability threshold is not met. This is in contrast to pure flooding, which passes **all** messages to neighboring nodes. Using a sufficiently high probability, but much smaller than 1, probabilistic flooding maintains reliable message delivery while reducing the number of intermediate nodes involved in the relay process. We do not present all the results (they can be found on the reproducibility appendix A.3 specified previously), but the important point is that the variation of β yields an improvement and a stable interval, and an optimal β , like in SLR protocol. However, the optimal β differs with the routing protocol: it is 800 (in probabilistic protocol), compared to 600 (in SLR protocol) for 10 000 nodes; and 1 000 compared to 800 for 30 000 nodes.

To conclude, increasing β up to a point (called optimal β) reduces packet collisions and increases the number of packet receptions and transmissions; after this point, further maximizing β does not affect the number of collisions or reception and transmission. The optimal value of β increases with network density (number of nodes), and depends on the protocol, number of packets per flow, among others.

6.3.3/ IMPACT OF THE COMMUNICATION RANGE

In this section, we delve into the impact of communication range (the distance up to which a node receives the packet) on both the occurrence and mitigation of packet collisions, reception, and emission. We address a fundamental question: is it advantageous to use

Table 6.3: Simulation results for various communication ranges.

C:Collisions, R:Receptions, E:Emissions, D:Deliveries

Comm. range [μm]	Homogeneous network				Heterogeneous network			
	C	R	E	D	C	R	E	D
285	39 960	162 733	2 807	117	37 966	183 821	2 951	111
350	72 730	249 519	2 969	162	73 226	267 375	3 011	127
450	178 075	360 315	3 016	172	169 028	392 268	3 004	161
500	269 332	390 008	3 000	180	243 107	451 081	3 020	157

a larger communication range?

We present a series of simulations where we adjust the communication range while keeping the other parameters constant (given in Table 6.1). Note that, given that during the SLR setup phase the communication range impacts the size of zones during routing, for fairness and consistency across all simulations, we vary the communication range **only during the routing phase** and use the same communication range for the SLR setup phase, set to $220 \mu\text{m}$ (cf. Table 6.1). This ensures uniformity in the formation of zones in all our simulations. We also notice that the results for 1 packet and for 20 packets per flow are similar, hence, we present results only for 1 packet.

The results are presented in Table 6.3. They show that **increasing the communication range results in a higher number of collisions** in both homogeneous and heterogeneous networks. For instance, in the homogeneous network, a larger communication range results in an increase in collisions; for communication ranges of $285 \mu\text{m}$ and $500 \mu\text{m}$, 39 960 and 269 332 packets collide, respectively. Similarly, in the heterogeneous network, the number of collisions rises from 37 966 to 243 107 as the communication range increases from $285 \mu\text{m}$ to $500 \mu\text{m}$.

This phenomenon occurs because, as the communication range increases, more nodes communicate over a wider spatial area, enhancing their ability to interact with more neighboring nodes. As a result, having more nodes in reach increases the likelihood of overlapping transmissions and collisions within the network. Moreover, the extended communication range enables previously out-of-range nodes to potentially interfere with each other's transmissions, further increasing collision occurrences.

Table 6.3 also shows that a larger communication range increases the number of sent (emissions column in the table) and received packets (receptions and deliveries columns). A larger communication range implies more neighboring nodes to fall within reach, consequently, more packets are received, which in turn results in increased retransmissions (emissions) within the relay nodes.

In summary, the larger the communication range, the higher the collision rate and the number of both received and sent messages.

Table 6.4: Simulation results for various pulse durations.

C:Collisions, R:Receptions, E:Emissions, D:Deliveries

Pulse duration [fs]	Homogeneous network				Heterogeneous network			
	C	R	E	D	C	R	E	D
10	63 172	139 994	2 813	106	72 126	150 683	2 961	89
50	49 823	153 442	2 815	105	57 154	164 371	2 948	94
150	29 868	173 690	2 819	130	32 402	189 307	2 953	100
800	24 859	100 743	1 765	116	19 046	153 514	2 243	91

Finally, we note that it is up to the application to choose between both more collisions and more message deliveries (with a large communication range), and both fewer collisions and fewer message deliveries (with a small communication range).

6.3.4/ IMPACT OF THE PULSE DURATION

Pulse duration T_p refers to the length of time a signal remains in a high or low state, also known as its occupation time. This duration is typically determined by the characteristics of the modulation scheme and the hardware used for transmission and reception.

This section analyzes how pulse duration impacts the number of packet collisions, receptions, and emissions. In the simulations, we vary pulse duration while maintaining the same network scenario (given in Table 6.1) with 1 packet per flow.

The results are shown in Table 6.4. **An increase in pulse duration results in a decrease in the number of packet collisions.** For example, when the pulse duration increases from 10 to 800 fs, the number of packet collisions decreases from 63 172 to 24 859 for the homogeneous network, and from 72 126 to 19 046 for the heterogeneous network.

The increase in pulse duration correlates with a rise in the number of **packet receptions** (up to a point) at both the destination zone (deliveries in the table) and zones along the transmission path (receptions). For instance, in the homogeneous network, the number of packets received at intermediate zones increased from 139 994 to 173 690 as the pulse duration increased from 10 to 150 fs and then reduced to 100 743 as the pulse duration increased from 150 to 800 fs. Similarly, in the heterogeneous network, the number of packets received at the destination zone (deliveries) increased from 89 to 100 and then decreased to 91.

6.3.5/ IMPACT OF NODE DENSITY

This section presents how node density affects communications. In the simulations, we vary the number of nodes (hence the density) while maintaining the same network

Table 6.5: Simulation results for various node densities.

C:Collisions, R:Receptions, E:Emissions, D:Deliveries

Node density nodes	1 packet per flow				20 packets per flow			
	C	R	E	D	C	R	E	D
70	8 307	195 912	2 828	157	224 515	3 865 119	56 627	3 022
141	62 559	812 905	6 106	270	1 683 221	15 875 253	122 399	5 345
212	315 738	1 704 711	9 445	467	4 953 223	35 422 768	188 761	10 243
283	562 700	2 676 558	11 280	541	10 843 521	59 008 257	243 780	10 622

scenario and a homogeneous network. The node density can simply be computed as $\rho = n(\pi r^2/S)$, where n is the total number of nodes, $r = 285 \mu\text{m}$ is the communication range, and $S = 6000 * 6000 \mu\text{m}^2$ is the network surface (cf. Table 6.1). Thus, for 10 000, 20 000, 30 000, and 40 000 nodes in the network, the densities obtained are 70, 141, 212, and 283 neighbors per node, respectively.

Table 6.5 shows that increasing the density results in an increased collision rate and an increase in the number of packet receptions, emissions, and deliveries. Indeed, increasing density results in more neighboring nodes and more packets being transmitted, so more collisions and receptions occur.

6.4/ CONCLUSION

This chapter presented the influence of four network parameters (β , communication range, pulse duration and node density) on communications (packet collisions, emissions, receptions, and deliveries) in electromagnetic wireless nanonetworks.

Our findings are that increasing β (up to a point) reduces packet collisions while increasing the number of receptions, emissions, and deliveries; increasing β past this point has no significant effect. Increasing communication range or node density increases collisions, receptions, emissions, and deliveries. Increasing pulse duration reduces the number of packet collisions while increasing the number of packets received and emitted (up to a point).

These insights provide valuable guidance in choosing network parameters to enhance network usage.



CONCLUSION

CONCLUSION AND FUTURE WORK

This thesis, entitled "Innovative Algorithms in Dense Networks", presented a study on the design and development of routing protocols tailored for dense electromagnetic nanonetworks.

In this section, I review and summarize the key contributions and advancements achieved during my PhD research, highlighting the novel protocol and mechanism developed to address the specific challenges inherent in this field.

The first part of this thesis introduces the concept of dense nanonetworks and their applications, emphasizing their significance in fields such as healthcare, environmental monitoring, and military technology. It also outlines the constraints faced by nanonetwork machines (nanonodes), such as resources limitation.

Additionally, this thesis provides an overview of specific nanonetwork simulators, including Nano-Sim, TeraSim, and Vouivre, with an in-depth analysis of BitSimulator, identified as the optimal tool for our study. We also present the modulation scheme (TS-OOK) and the protocol (SLR) that form the basis of our research.

First, my initial study aimed to review the mechanisms that provide concurrency on the same channel. For example, bit interleaving where multiple packets occupy the channel at the same time but no two bits, capture effect where only the strongest signal can be correctly demodulated by the receiver, constructive interference where multiple nodes transmit the same packet simultaneously, resulting in the constructive addition of their signals at the receiver, pulse shape changing where the shape of pulses allows the receiver to reliably detect signal peaks associated with different responders, etc. We identified and detailed the various levels of communication concurrency, namely pulse, bit, packet, and flow levels, with examples of network technologies and protocols that support each. For instance, the SAR protocol supports bit, packet, and flow-level concurrency. This understanding offers valuable insights to protocol creators and users intending to use implement concurrency in their networks.

Second, we aimed to address the challenges of id assignment in dense nanonetworks,

where the large number of nodes makes traditional approaches impractical. To tackle this, I proposed the EIDA (Equitable Distributed ID Assignment) mechanism, a configurable and distributed mechanism designed for dense nanonetworks.

EIDA strikes a balance between ideal and random id assignment methods. While ideal assignments ensure equitability, they are impractical in dense networks due to the significant number of packet exchanges required. On the other hand, random assignments, though less resource-intensive, can lead to inefficiencies and unequal distribution of node IDs. EIDA combines these approaches, allowing nodes to assign their IDs through packet exchanges until a predetermined level of equitability (threshold) is reached. Beyond this threshold, the mechanism transitions to random id assignment for the remaining nodes, thereby minimizing further communication.

Simulations demonstrate the effectiveness of the EIDA mechanism in id assignment by showing that it uses fewer fewer packet exchanges—compared to ideal assignment (fewer resources), while still achieving a specific level of equitability (above the threshold), unlike random assignment.

Third, since existing routing protocols are not well-suited for the unique challenges presented by dense networks, a new, dedicated routing protocol is necessary, especially for dense nanonetworks. This need led to the development of FR-SLR (Forwarder Reduction in SLR Routing Protocol), a novel protocol designed to address these challenges.

FR-SLR is based on the SLR protocol. SLR is a unicast routing protocol that transmits messages to a single destination by dividing the network into zones and assigning each zone a coordinate. In SLR, only the zones along the transmission path are responsible for forwarding the packet.

FR-SLR improves upon the SLR protocol by aiming to reduce the number of packet forwarders. It enhances SLR by dividing each zone into multiple groups, where only one group in each zone along the transmission path forwards the packet. This approach decreases the number of forwarders between the sender and receiver. The number of groups within each zone can be specified by the user, providing greater flexibility and efficiency.

FR-SLR also uses the EIDA mechanism for id assignment, assigning each node in a zone a group id. Evaluations demonstrate the benefits of FR-SLR, particularly when combined with EIDA, in reducing the number of forwarders and improving packet delivery times.

Fourth, we evaluate how various network parameters impact communication quality in a dense electromagnetic nanonetwork. Evaluation done in homogeneous and heterogeneous networks, and across scenarios with single and multiple packets per flow, demonstrate several key insights, as follows. Increasing β (up to a point) reduces packet collisions while increasing the number of receptions, emissions, and deliveries; increasing β

past this point has no significant effect. Increasing communication range or node density increases collisions, receptions, emissions, and deliveries. Increasing pulse duration reduces the number of packet collisions while increasing the number of packets received and emitted (up to a point). These insights provide valuable information in selecting network parameters to enhance network usage.

To **conclude**, my thesis presents four significant contributions to the field of dense electromagnetic nanonetworks. First, the exploration of concurrency mechanisms sheds light on various levels of communication concurrency, offering valuable guidance for future protocol design and implementation. Second, the development of the EIDA mechanism addresses the challenge of manageable equitable ID assignment, balancing the limitations of ideal and random methods. Third, the creation of the FR-SLR protocol provides a dedicated routing solution tailored to the unique demands of dense nanonetworks, reducing the number of forwarders. Finally, the evaluation of network parameters reveals critical insights that guide the optimization of communication quality in dense environments.

Future work will focus on:

- **Compare concurrent and sequential communication:** Analyzing the benefits and drawbacks of concurrent communication (e.g. at the packet level) compared to sequential communication. For example, concurrent communication allows multiple packets to be transmitted simultaneously, potentially increasing throughput but introducing challenges such as increased packet collisions. The focus will be also on assessing the feasibility of nodes to support concurrency based on their characteristics, such as processing power, memory, and energy constraints. Understanding these trade-offs is key to determining the best approach for different network environments.
- **Create a new routing protocol that supports concurrency:** To fully leverage the potential of concurrent communication, it will be essential to develop a routing protocol that supports concurrency without introducing packet loss. This involves implementing and evaluating various mechanisms, such as capture effect and constructive interference, to manage concurrent transmissions efficiently. The protocol will be tested in different network scenarios to ensure it can handle high traffic loads while maintaining reliability and minimizing latency.
- **Exploring solutions for preventing routing protocol failures:** Improve FR-SLR to prevent transmission die-out in the case of node failures. This can be achieved by introducing a test to monitor the number of transmitted packets in each zone, allowing the protocol to detect potential issues and adjust routing strategies accordingly.

- **Enhancing the EIDA mechanism for better network adaptability:** Further refine the EIDA assignment mechanism to ensure uniform group distribution across all network zones, particularly in heterogeneous networks, to prevent routing instability. This approach guarantees that even if the number of nodes in each zone differs, packets can still be transmitted because the sender's id exists across all zones.
- **Extending the analysis of network communication quality:** Conduct a more comprehensive analysis of network communication quality metrics, including packet collisions, receptions, emissions, and deliveries, by considering a broader range of network parameters and protocols.
- **Investigate adaptive routing protocols:** Adaptive routing protocols are designed to adjust dynamically based on current network conditions, such as congestion levels or link quality. The focus of this work will be to develop and analyze such protocols for dense nanonetworks. The routing decisions will be based on real-time feedback from nodes, allowing the protocol to quickly change paths when congestion or poor link quality is detected. This approach can significantly reduce latency and improve overall network performance by ensuring data is always transmitted through the most optimal paths.
- **Analyze packet prioritization schemes:** Different types of data have different urgency levels, and packet prioritization schemes aim to ensure that time-sensitive data is delivered as quickly as possible. Future work will investigate various prioritization methods, such as Quality of Service (QoS)-based routing, where certain packets are given priority over others based on their importance. For example, critical sensor data could be prioritized over routine status updates. This will help reduce latency for high-priority packets and ensure timely delivery of critical information.
- **Implement dynamic load balancing:** Nanonetworks often operate with small, energy-constrained nodes, making dynamic load balancing essential to optimizing resource usage. By evenly distributing traffic across nodes, load balancing prevents certain nodes from becoming overloaded, which would increase their energy consumption and risk early failure. Dynamic load balancing can also help avoid network bottlenecks, reducing latency by ensuring that no single area of the network is overwhelmed by traffic. This approach will prolong the lifetime of individual nodes and enhance the overall reliability and performance of the network.

BIBLIOGRAPHY

- [Afsharinejad et al. 2016] AFSHARINEJAD, Armita ; DAVY, Alan ; JENNINGS, Brendan: **“Dynamic channel allocation in electromagnetic nanonetworks for high resolution monitoring of plants”**. In *Nano Communication Networks* 7 (2016), Mar, pages 2–16
- [Aiken et al. 2011] AIKEN, George R. ; HSU-KIM, Heileen ; RYAN, Joseph N.: **“Influence of Dissolved Organic Matter on the Environmental Fate of Metals, Nanoparticles, and Colloids”**. In *Environmental Science & Technology* 45 (2011), Mar, number 8, pages 3196–3201. – PMID: 21405118. DOI: 10.1021/es103992s
- [Al Mawla and Dedu 2024] AL MAWLA, Carole ; DEDU, Eugen: **“Network Parameter Influence on Communications in Dense Wireless Nanonetworks”**. In *10th ACM International Conference on Nanoscale Computing and Communication (ACM NanoCom)* (2024), Oct, pages 6
- [Al Mawla et al. 2023a] AL MAWLA, Carole ; DEDU, Eugen ; BEYDOUN, Kamal ; DHOUTAUT, Dominique: **“EIDA, a best effort equitable distributed id assignment mechanism for heterogeneous dense network”**. In *10th ACM International Conference on Nanoscale Computing and Communication (ACM NanoCom)* (2023), Sep, pages 1–7
- [Al Mawla et al. 2023b] AL MAWLA, Carole ; DEDU, Eugen ; BEYDOUN, Kamal ; DHOUTAUT, Dominique: **“FR-SLR, forwarder reduction in SLR routing protocol through zone splitting”**. In *International Wireless Communications & Mobile Computing Conference (IWCMC)*. Marrakesh, Morocco : IEEE, jun 2023, 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 Communication and Networks (ICCCN)* IEEE (event), 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, November 2018, pages 385–393

- [Assaker et al. 2020] ASSAKER, Joseph ; MAKHOUL, Abdallah ; BOURGEOIS, Julien ; DEMERJIAN, Jacques: **“A unique identifier assignment method for distributed modular robots”**. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Online (Las Vegas, NV, USA) : IEEE, Oct 2020, pages 3304–3311
- [Augustin et al. 2016] AUGUSTIN, Aloÿs ; YI, Jiazi ; CLAUSEN, Thomas ; TOWNSLEY, William M.: **“A study of LoRa: Long range & low power networks for the internet of things”**. In *Sensors* 16 (2016), Sep, number 9, pages 1466
- [Bisdikian 2001] BISDIKIAN, Chatschik: **“An overview of the Bluetooth wireless technology”**. In *IEEE Communications Magazine* 39 (2001), Dec, number 12, pages 86–94
- [Boillot et al. 2013] BOILLOT, Nicolas ; DHOUTAUT, Dominique ; BOURGEOIS, Julien: **“Efficient simulation environment of wireless radio communications in mems modular robots”**. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. Beijing, China : IEEE, Aug 2013, pages 638–645
- [Boillot et al. 2015] BOILLOT, Nicolas ; DHOUTAUT, Dominique ; BOURGEOIS, Julien: **“Going for large scale with nano-wireless simulations”**. In *Proceedings of the Second Annual International Conference on Nanoscale Computing and Communication*. Boston, MA : Association for Computing Machinery, Sep 2015, pages 1–2
- [Colli-Vignarelli et al. 2009] COLLI-VIGNARELLI, James ; VERNEZ, Jérôme ; MERZ, Ruben ; DEHOLLAIN, Catherine ; ROBERT, Stephan ; LE BOUDEC, Jean-Yves: **“Concurrent transmissions in IR-UWB networks: an experimental validation”**. In *IEEE International Conference on Ultra-Wideband (ICUWB)*. Vancouver, Canada : IEEE, Sept 2009, pages 337–342
- [Corbalán et al. 2019] CORBALÁN, Pablo ; PICCO, Gian P. ; PALIPANA, Sameera: **“Chorus: UWB concurrent transmissions for GPS-like passive localization of countless targets”**. In *18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. Montreal, Canada : IEEE, Apr 2019, pages 133–144
- [DecaWave Ltd 2016] DECAWAVE LTD: *DW1000 User Manual. V2.18. 2016*
- [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
- [Dhoutaut et al. 2018] DHOUTAUT, Dominique ; ARRABAL, Thierry ; DEDU, Eugen: **“Bit-Simulator, an electromagnetic nanonetworks simulator”**. In *5th ACM/IEEE Inter-*

- national Conference on Nanoscale Computing and Communication (NanoCom)*. Reykjavik, Iceland : ACM/IEEE, Sep 2018, pages 1–6
- [Dorj et al. 2017] DORJ, Ulzii-Orshikh ; LEE, Malrey ; CHOI, Jae-young ; LEE, Young-Keun ; JEONG, Gisung ; OTHERS: **“The intelligent healthcare data management system using nanosensors”**. In *Journal of Sensors* 2017 (2017), Dec
- [Fattah et al. 2015] FATTAH, Mohammad ; AIROLA, Antti ; AUSAVARUNGNIRUN, Rachata ; MIRZAEI, Nima ; LILJEBERG, Pasi ; PLOSILA, Juha ; MOHAMMADI, Siamak ; PAHIKKALA, Tapio ; MUTLU, Onur ; TENHUNEN, Hannu: **“A low-overhead, fully-distributed, guaranteed-delivery routing algorithm for faulty network-on-chips”**. In *9th International Symposium on Networks-on-Chip*. New York, NY, USA : Association for Computing Machinery, Sep 2015, pages 1–8
- [Ferrari et al. 2011] FERRARI, Federico ; ZIMMERLING, Marco ; THIELE, Lothar ; SAUKH, Olga: **“Efficient network flooding and time synchronization with Glossy”**. In *10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. Chicago, IL, USA : IEEE, Apr 2011, pages 73–84
- [Garg et al. 2013] GARG, Kamini ; GIORDANO, Silvia ; FÖRSTER, Anna: **“A study to understand the impact of node density on data dissemination time in opportunistic networks”**. In *2nd ACM workshop on High Performance Mobile Opportunistic Systems*. Barcelona, Spain : ACM, November 2013, pages 9–16
- [Ghavami et al. 2007] GHAVAMI, Mohammad ; MICHAEL, Lachlan ; KOHNO, Ryuji: *Ultra wideband signals and systems in communication engineering*. John Wiley & Sons, 2007
- [Großwindhager et al. 2018] GROSSWINDHAGER, Bernhard ; BOANO, Carlo A. ; RATH, Michael ; RÖMER, Kay: **“Concurrent ranging with ultra-wideband radios: From experimental evidence to a practical solution”**. In *38th IEEE International Conference on Distributed Computing Systems (ICDCS)*. Vienna, Austria : IEEE, July 2018, pages 1460–1467
- [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
- [Hoteit et al. 2023] HOTEIT, Farah ; DEDU, Eugen ; SEAH, Winston K. ; DHOUTAUT, Dominique: **“Influence of beta and source packet rate on electromagnetic nanocommunications”**. In *28th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*. Nanjing, China : ICPADS, January 2023, pages 443–449

- [Imoni et al. 2023] IMONI, Samson ; ONYEBUCHI, Mogbo ; TIZA, Michael T. ; AKANDE, Ebenezer O. ; ONUZULIKE, Collins: **“Nano Revolution: Advancing Civil Engineering through Nanomaterials and Technology”**. In *Journal of Novel Engineering Science and Technology* 2 (2023), Oct, number 03, pages 94–103
- [Jain et al. 2011] JAIN, Mayank ; CHOI, Jung I. ; KIM, Taemin ; BHARADIA, Dinesh ; SETH, Siddharth ; SRINIVASAN, Kannan ; LEVIS, Philip ; KATTI, Sachin ; SINHA, Prasun: **“Practical, real-time, full duplex wireless”**. In *17th Annual International Conference on Mobile computing and networking (ACM MobiCom)*. Las Vegas, NV, USA : ACM, Sept 2011, pages 301–312
- [Jornet and Akyildiz 2014a] JORNET, Josep M. ; AKYILDIZ, Ian F.: **“Femtosecond-Long Pulse-Based Modulation for Terahertz Band Communication in Nanonetworks”**. In *IEEE Transactions on Communications* 62 (2014), May, number 5, pages 1742–1754
- [Jornet and Akyildiz 2014b] JORNET, Josep M. ; AKYILDIZ, Ian F.: **“Femtosecond-Long Pulse-Based Modulation for Terahertz Band Communication in Nanonetworks”**. In *IEEE Transactions on Communications* 62 (2014), May, number 5, pages 1742–1753
- [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* 3 (2012), March, number 1, pages 74–81
- [Jouhari et al. 2019] JOUHARI, Mohammed ; IBRAHIMI, Khalil ; TEMBINE, Hamidou ; BEN-OTHTMAN, Jalel: **“Underwater wireless sensor networks: A survey on enabling technologies, localization protocols, and internet of underwater things”**. In *IEEE Access* 7 (2019), July, pages 96879–96899
- [Kabir et al. 2021] KABIR, Md H. ; ISLAM, SM R. ; SHRESTHA, Anish P. ; ALI, Farman ; BADSHA, Md A. ; PIRAN, Md J. ; DO, Dinh-Thuan: **“Electromagnetic nanocommunication networks: Principles, applications, and challenges”**. In *IEEE Access* 9 (2021), Dec, pages 166147–166165
- [Kim and Tran-Dang 2019] KIM, Dong-Seong ; TRAN-DANG, Hoa: **“Ultra-Wideband Technology for Military Applications”**. In *Industrial Sensors and Controls in Communication Networks (CCN)*. Springer, 2019, pages 197–204
- [Kong and Liu 2015] KONG, Linghe ; LIU, Xue: **“mZig: Enabling multi-packet reception in ZigBee”**. In *21st Annual International Conference on mobile computing and networking (ACM MobiCom)*. Paris, France : ACM, July 2015, pages 552–565

- [Law et al. 2013] LAW, David ; DOVE, Dan ; D'AMBROSIA, John ; HAJDUCZENIA, Marek ; LAUBACH, Mark ; CARLSON, Steve: **“Evolution of Ethernet standards in the IEEE 802.3 working group”**. In *IEEE Communications Magazine* 51 (2013), Aug, number 8, pages 88–96
- [Lemic et al. 2021] LEMIC, Filip ; ABADAL, Sergi ; TAVERNIER, Wouter ; STROOBANT, Pieter ; COLLE, Didier ; ALARCÓN, Eduard ; MARQUEZ-BARJA, Johann ; FAMAËY, Jeroen: **“Survey on Terahertz Nanocommunication and Networking: A Top-Down Perspective”**. In *Journal on Selected Areas in Communications* 39 (2021), Feb, number 6, pages 1506–1543
- [Li et al. 2018] LI, Yan ; CHI, Zicheng ; LIU, Xin ; ZHU, Ting: **“Chiron: Concurrent high throughput communication for IoT devices”**. In *16th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*. Munich, Germany : ACM, June 2018, pages 204–216
- [Liao et al. 2017] LIAO, Chun-Hao ; ZHU, Guibing ; KUWABARA, Daiki ; SUZUKI, Makoto ; MORIKAWA, Hiroyuki: **“Multi-hop LoRa networks enabled by concurrent transmission”**. In *IEEE Access* 5 (2017), Sept, pages 21430–21446
- [Lin et al. 2007] LIN, Jialiu ; LIU, Yunhuai ; NI, Lionel M.: **“SIDA: self-organized ID assignment in wireless sensor networks”**. In *IEEE International Conference on Mobile Adhoc and Sensor Systems*. Pisa, Italy : IEEE, Oct 2007, pages 1–8
- [Liu et al. 2013] LIU, Runzi ; SHENG, Min ; LUI, King-Shan ; SHI, Yan: **“SIC aware high-throughput routing in multihop wireless networks”**. In *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)* IEEE (event), 2013, pages 3229–3233
- [Liu and Ni 2006] LIU, Yunhuai ; NI, Lionel M.: **“Location-aware ID Assignment in Wireless Sensor Networks”**. In *International Conference on Mobile Ad Hoc and Sensor Systems*. Vancouver, Canada : IEEE, Nov 2006, pages 525–529
- [Mabed 2017] MABED, Hakim: **“Enhanced spread in time on-off keying technique for dense Terahertz nanonetworks”**. In *IEEE Symposium on Computers and Communications (ISCC)*. Heraklion, Greece : IEEE, Jul 2017, pages 710–716
- [Medlej et al. 2021] MEDLEJ, Ali ; DEDU, Eugen ; BEYDOUN, Kamal ; DHOUTAUT, Dominique: **“Self-configuring asynchronous sleeping in heterogeneous networks”**. In *ITU Journal on Future and Evolving Technologies* 2 (2021), October, number 7, pages 51–62
- [Minet et al. 2010] MINET, Pascale ; MAHFOUDH, Saoucene ; CHALHOUB, Gerard ; GUITTON, Alexandre: **“Node coloring in a wireless sensor network with unidirectional**

- links and topology changes**". In *IEEE Wireless Communication and Networking Conference*. Sydney, Australia : IEEE, Apr 2010, pages 1–6
- [Minoli and Occhiogrosso 2018] MINOLI, Daniel ; OCCHIOGROSSO, Benedict: **"Ultrawideband (UWB) technology for smart cities IoT applications"**. In *IEEE International Smart Cities Conference (ISC2)*. Kansas City, MO, USA : IEEE, Sept 2018, pages 1–8
- [Molisch 2009] MOLISCH, Andreas F.: **"Ultra-wideband communications: An overview"**. In *URSI Radio Science Bulletin* (2009), Jun, number 329, pages 31–42
- [Nishimura 2000] NISHIMURA, Takuji: **"Tables of 64-bit Mersenne twisters"**. In *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 10 (2000), Oct, number 4, pages 348–357
- [Oukhatar et al. 2021] OUKHATAR, Ayoub ; BAKHOUYA, Mohamed ; EL OUADGHIRI, Driss: **"Electromagnetic-Based Wireless Nano-Sensors Network: Architectures and Applications."** In *J. Commun.* 16 (2021), Jan, number 1, pages 8
- [Ould-Ahmed-Vall et al. 2005] OULD-AHMED-VALL, ElMoustapha ; BLOUGH, Douglas M. ; HECK, Bonnie S. ; RILEY, George F.: **"Distributed unique global ID assignment for sensor networks"**. In *Ad Hoc Networks* 7 (2005), Nov, number 6, pages 2051–2055
- [Park et al. 2020] PARK, Jae-Wan ; KIM, Eung-Hyo ; HONG, Een-Kee: **"Cluster splitting and combining in ultra-dense networks"**. In *IET Communications* 14 (2020), April, number 9, pages 1438–1443
- [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
- [Pramanik et al. 2020] PRAMANIK, Pijush Kanti D. ; SOLANKI, Arun ; DEBNATH, Abhinaba ; NAYYAR, Anand ; EL-SAPPAGH, Shaker ; KWAK, Kyung-Sup: **"Advancing modern healthcare with nanotechnology, nanobiosensors, and internet of nano things: Taxonomies, applications, architecture, and challenges"**. In *IEEE Access* 8 (2020), Apr, pages 65230–65266
- [Raiciu et al. 2020] RAICIU, Costin ; HANDLEY, Mark ; BONAVENTURE, Olivier ; PAASCH, Christoph: *TCP Extensions for Multipath Operation with Multiple Addresses*. mar 2020. – RFC 8684

- [Reina et al. 2015] REINA, Daniel G. ; TORAL, SL ; JOHNSON, Princy ; BARRERO, Federico: **“A survey on probabilistic broadcast schemes for wireless ad hoc networks”**. In *Ad Hoc Networks* 25 (2015), pages 263–292
- [Rikhtegar et al. 2017] RIKHTEGAR, Negar ; JAVIDAN, Reza ; KESHTGARI, Manijeh: **“Mobility management in wireless nano-sensor networks using fuzzy logic”**. In *Journal of Intelligent & Fuzzy Systems* 32 (2017), Jan, number 1, pages 969–978
- [Ryu et al. 2011] RYU, Junghun ; YU, Jaewook ; NOEL, Eric ; TANG, K W.: **“Node ID assignment in group theoretic graphs for WSNs”**. In *Wireless Telecommunications Symposium (WTS)*. New York, NY, USA : IEEE, Apr 2011, pages 1–8
- [Sahin et al. 2021] SAHIN, Emre ; DAGDEVIREN, Orhan ; AKKAS, Mustafa A.: **“An Evaluation of Internet of Nano-Things Simulators”**. In *6th International Conference on Computer Science and Engineering (UBMK)*. Ankara, Turkey : IEEE, September 2021, pages 670–675
- [Salvati et al. 2015] SALVATI, Elisa ; STELLACCI, Francesco ; KROL, Silke: **“Nanosensors for early cancer detection and for therapeutic drug monitoring”**. In *Nanomedicine* 10 (2015), Nov, number 23, pages 3495–3512
- [Sen et al. 2013] SEN, Souvik ; SANTHAPURI, Naveen ; CHOUDHURY, Romit R. ; NELAKUDITI, Srihari: **“Successive interference cancellation: Carving out MAC layer opportunities”**. In *IEEE Transactions on Mobile Computing* 12 (2013), Feb, number 2, pages 346–357
- [Simić et al. 2018] SIMIĆ, Danica ; MARJANOVIĆ, Milica ; VITOROVIĆ, Todorović Maja ; BAUK, Sonja ; LAZIĆ, Dragana ; SAMOLOV, Aleksandra ; RISTOVIĆ, Novica: **“Nanotechnology for military applications: A survey of recent research in Military technical institute”**. In *Scientific Technical Review* 68 (2018), Aug, number 1, pages 59–72
- [de Souza et al. 2016] SOUZA, Fabio A. de ; CHANG, Bruno S. ; BRANTE, Glauber ; SOUZA, Richard D. ; PELLEZ, Marcelo E. ; ROSAS, Fernando: **“Optimizing the number of hops and retransmissions for energy efficient multi-hop underwater acoustic communications”**. In *IEEE Sensors Journal* 16 (2016), May, number 10, pages 3927–3938
- [Tian et al. 2008] TIAN, Ye ; SHENG, Min ; LI, Jiandong: **“Virtual Grid Spatial Reusing Algorithm for MAC Address Assignment in Wireless Sensor Network”**. In *International Journal of Distributed Sensor Networks* 4 (2008), Apr, number 2, pages 142–154
- [Toh et al. 2002] TOH, C-K ; DELWAR, Minar ; ALLEN, Donald: **“Evaluating the communication performance of an ad hoc wireless network”**. In *IEEE Transactions on Wireless Communications* 1 (2002), July, number 3, pages 402–414

- [Tsioliariidou et al. 2017] TSIOLIARIDOU, Ageliki ; LIASKOS, Christos ; DEDU, Eugen ; IOANNIDIS, Sotiris: **“Packet routing in 3D nanonetworks: A lightweight, linear-path scheme”**. In *Nano Communication Networks* 12 (2017), Jun, pages 63–71
- [Vecchia et al. 2019] VECCHIA, Davide ; CORBALÁN, Pablo ; ISTOMIN, Timofei ; PICCO, Gian P.: **“Playing with fire: Exploring concurrent transmissions in ultra-wideband radios”**. In *16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. Boston, MA, USA : IEEE, June 2019, pages 1–9
- [Vinyals et al. 2011] VINYALS, Meritxell ; RODRIGUEZ-AGUILAR, Juan A. ; CERQUIDES, Jesus: **“A survey on sensor networks from a multiagent perspective”**. In *The Computer Journal* 54 (2011), Mar, number 3, pages 455–470
- [Wang et al. 2016] WANG, Yanwen ; CHEN, Hainan ; WU, Xiaoling ; SHU, Lei: **“An energy-efficient SDN-based sleep scheduling algorithm for WSNs”**. In *Journal of Network and Computer Applications* 59 (2016), May, pages 39–45
- [Wang et al. 2015] WANG, Yin ; LIU, Yunhao ; HE, Yuan ; LI, Xiang-Yang ; CHENG, Dapeng: **“Disco: Improving packet delivery via deliberate synchronized constructive interference”**. In *IEEE Transactions on Parallel and Distributed Systems* 26 (2015), Mar, number 3, pages 713–723
- [Xu et al. 2019] XU, Juan ; ZHANG, Yan ; JIANG, Jiaolong ; KAN, Jiali: **“An energy balance clustering routing protocol for intra-body wireless nanosensor networks”**. In *Sensors* 19 (2019), Nov, number 22, pages 4875
- [Yang et al. 2020] YANG, Ke ; BI, Dadi ; DENG, Yansha ; ZHANG, Rui ; RAHMAN, Muhammad Mahboob U. ; ALI, Najah A. ; IMRAN, Muhammad A. ; JORNET, Josep M. ; ABBASI, Qammer H. ; ALOMAINY, Akram: **“A comprehensive survey on hybrid communication in context of molecular communication and terahertz communication for body-centric nanonetworks”**. In *IEEE Transactions on Molecular, Biological and Multi-Scale Communications* 6 (2020), Aug, number 2, pages 107–133
- [Ye and Zhang 2017] YE, Dayong ; ZHANG, Minjie: **“A self-adaptive sleep/wake-up scheduling approach for wireless sensor networks”**. In *IEEE Transactions on Cybernetics* 48 (2017), March, number 3, pages 979–992
- [Yu et al. 2015] YU, Hang ; NG, Bryan ; SEAH, Winston K.: **“Forwarding schemes for EM-based wireless nanosensor networks in the terahertz band”**. In *Second Annual International Conference on Nanoscale Computing and Communication (NanoCom)*. Boston, MA, USA : NanoCom, September 2015, pages 1–6
- [Yuan and Hollick 2013] YUAN, Dingwen ; HOLLICK, Matthias: **“Let’s talk together: Understanding concurrent transmission in wireless sensor networks”**. In *38th*

Annual IEEE Conference on Local Computer Networks (LCN). Sydney, Australia : IEEE, Oct 2013, pages 219–227

- [Yunus et al. 2012] YUNUS, Ian S. ; HARWIN ; KURNIAWAN, Adi ; ADITYAWARMAN, Dendy ; INDARTO, Antonius: **“Nanotechnologies in water and air pollution treatment”**. In *Environmental Technology Reviews* 1 (2012), Oct, number 1, pages 136–148
- [Zahedi et al. 2016] ZAHEDI, Zeynab M. ; AKBARI, Reza ; SHOKOUHIFAR, Mohammad ; SAFAEI, Farshad ; JALALI, Ali: **“Swarm intelligence based fuzzy routing protocol for clustered wireless sensor networks”**. In *Expert Systems with Applications* 55 (2016), Aug, pages 313–328
- [Zhang et al. 2020] ZHANG, De-gan ; WU, Hao ; ZHAO, Peng-zhen ; LIU, Xiao-huan ; CUI, Yu-ya ; CHEN, Lu ; ZHANG, Ting: **“New approach of multi-path reliable transmission for marginal wireless sensor network”**. In *Wireless Networks* 26 (2020), Feb, pages 1503–1517
- [Zhang et al. 2018] ZHANG, Peilin ; MA, Xiaoyuan ; THEEL, Oliver ; WEI, Jianming: **“Concurrent transmission-based packet concatenation in wireless sensor networks”**. In *43rd Annual IEEE Conference on Local Computer Networks (LCN)*. Chicago, USA : IEEE, Oct 2018, pages 421–424
- [Zhao et al. 2020] ZHAO, Ruiqin ; LIU, Yuan ; DOBRE, Octavia A. ; WANG, Haiyan ; SHEN, Xiaohong: **“An efficient topology discovery protocol with node id assignment based on layered model for underwater acoustic networks”**. In *Sensors* 20 (2020), Nov, number 22, pages 6601

LIST OF FIGURES

1.1	Context overview.	4
2.1	Nanosensor architecture Oukhatar et al. (2021).	12
2.2	Result page of the ECG nanosensor Dorj et al. (2017).	14
2.3	Architecture of the IoNT enabled healthcare system Pramanik et al. (2020).	15
2.4	Nanonetworks for plant monitoring.	16
2.5	TS-OOK modulation.	16
2.6	Packet overlapping by bit interleaving in TS-OOK.	17
2.7	VisualTracer sketch for the SLR initialization phase.	18
2.8	VisualTracer sketch for the SLR routing phase.	19
2.9	Sketch for a nanonetwork scenario in visual tracer (BitSimulator visualizer)	20
2.10	Main C++ classes interaction.	24
2.11	Visual Tracer Sketch for two flow scenario.	30
3.1	The various concurrency levels.	35
3.2	Capture effect between two signals.	39
3.3	Successive interference cancellation.	40
4.1	Node activation order for Initiating EIDA, ordered by the backoff.	54
4.2	VisualTracer sketch for SLR routing phase.	58
4.3	Number of nodes that assign their ids to 0, 1, . . . , 5 during packet exchange (phase 1 of EIDA), during backoff (phase 2 of EIDA), and after both phases (sum of both phases).	61
5.1	VisualTracer sketch for the chosen zone, of coordinates (31,25).	71
5.2	VisualTracer sketch for SLR and FR-SLR routing phase.	72
6.1	Heterogeneous network used in simulations.	78

6.2 The network and the flows used in evaluation. 80

LIST OF TABLES

3.1	Concurrency levels supported by various technologies and protocols. . . .	44
4.1	Simulation parameters.	59
4.2	Node id assignment simulation results.	60
4.3	Number of nodes in each group after id assignment.	62
4.4	Comparison of ids mechanisms.	62
5.1	Number of nodes in each group after assignment.	69
5.2	Simulation parameters.	70
5.3	Number of forwarders in each zone in FR-SLR.	74
5.4	Comparison of various routing protocols.	74
5.5	FR-SLR vs counter-based SLR.	74
6.1	The parameters used in the evaluation (unless otherwise stated).	79
6.2	Simulation results for various β values (and various number of nodes). . . .	81
6.3	Simulation results for various communication ranges.	82
6.4	Simulation results for various pulse durations.	83
6.5	Simulation results for various node densities.	84

IV

APPENDIX

REPRODUCIBILITY INFORMATION

Reproducing research results is a cornerstone of the scientific method, ensuring that others can independently verify findings. It provides confidence in the conclusions' validity and allows other researchers to build upon the work. By sharing reproducible results, we foster transparency, facilitate collaboration, and drive further advancements in the field.

This chapter presents the reproduction and patch code required to reproduce all the results discussed in the Contribution chapter (chapter II).

All simulations use BitSimulator version 0.9.5. To get started, download the simulator and compile it using classical `./configure` and `make`. Finally, run `make install`, or alternatively add BitSimulator and VisualTracer in the executable path (`$PATH` variable).

A.1/ EIDA MECHANISM

This section provides the steps to reproduce the results of the EIDA mechanism presented in section 4.

After downloading BitSimulator, apply `diff.patch` to BitSimulator (e.g. run `(patch -p1 <diff.patch)` in BitSimulator directory). Then, (re)compile the simulator.

The `scenario.xml` file describes the scenario to be executed (including the network topology, number of nodes, anchors place, etc.). Execute the scenario using the following command: `(bitsimulator scenario.xml)`.

Figure 4.2 can be reproduced with the following command: `(visualtracer --initialTimeSkip 899640000)`

Figure 4.3 can be reproduced using Gnuplot by considering the IDs occurrence presented in Table 4.2:

First create a file called `bar.dat`:

```
# EIDA1      EIDA2      EIDA
```

0	2	6	8
1	2	5	7
2	2	1	3
3	2	2	4
4	2	2	4
5	2	3	5

Second, create a file called `variation.gnuplot`:

```
set terminal pdf
set output 'output.pdf'
red = "#FF0000"; green = "#00FF00"; blue = "#0000FF";
set yrange [0:10]
set style data histogram
set style histogram rowstacked
set style fill solid border -1
set boxwidth 0.75
set xtics format ""
set grid ytics
set xlabel "ID"
set ylabel "Number of nodes that chose this ID"
plot "bar.dat" using 2:xtic(1) title "EIDA phase 1" linecolor rgb red, \
      "bar.dat" using 3 title "EIDA phase 2" linecolor rgb blue
```

Then, run Gnuplot using the command line in the location of the files with the following command: (`gnuplot variation.gnuplot`).

Table 4.2 shows the results of the EIDA simulation using the command `bitsimulator`, presenting each node choose which ID.

A.2/ FR-SLR PROTOCOL

This section provides the steps to reproduce the results of the FR-SLR protocol presented in section 5.

Similar to EIDA, after downloading `BitSimulator`, apply `diff.patch` to `BitSimulator` (e.g. run (`patch -p1 <diff.patch`) in `BitSimulator` directory). Then, compile the simulator.

For Scenario files, execute (`mkdir FM-SLR`) and (`cd FM-SLR`). Then download the two scenarios `scenario.xml` and `scenario_slrcounter.xml` in the `FM-SLR` directory. These two files describe the scenario to be executed.

Figure 5.1 can be reproduced using (visualtracer) command.

Figure 5.2 can be reproduced using:

- `(visualtracer --initialTimeSkip 899640000 -s 100000000000)` before applying the patch for figure a
- `(visualtracer --initialTimeSkip 899640000 -s 100000000000)` after applying the patch for figure b

Table 5.1 reproduction: EIDA assignment can be reproduced using the addition of two-phase (ideal+random) using scenario.xml

- `(bitsimulator 2>&1 | grep "sending packet holding id: 0" | wc -l)` to count the number of nodes in group 0 of the first phase (ideal assignment).
- `(bitsimulator 2>&1 | grep "choose id based on the random backoff: 0" | wc -l)` to count the number of node in group 0 of the second phase (random assignment).
- The total number of nodes in group zero is calculated by adding the number of the first and of the second phase.

Table 5.3 uses the results of FM-SLR simulation using the commands

- `(bitsimulator 2>&1 | grep "host Coordinate: 25 , 28" | wc -l)` to count the forwarder number in zone 25,28.
- The same code is used to count the number of forwarders for the other zones as appear in the table by just changing the coordinate value for example for zone 35,26 we execute: `(bitsimulator 2>&1 | grep "host Coordinate: 35 , 26" | wc -l)`.

In Table 5.4, the number of packets sent uses the results of different routing protocol simulations.

Scenario.xml is used with the following commands:

- Pure flooding protocol:
 - `bitsimulator --routing PureFloodingRouting --prefix pureflooding`
 - `visualtracer --prefix pureflooding -s 100000000000`
- Probabilistic flooding protocol:

- bitsimulator --routing ProbaFloodingRouting --probability .04
--prefix probaflooding
- visualtracer --prefix probaflooding -s 100000000000
- Backoff flooding protocol:
 - bitsimulator --routing BackoffFloodingRouting --prefix
backoffflooding
 - visualtracer --prefix backoffflooding -s 100000000000
- SLR protocol:
 - bitsimulator before applying the patch
 - visualtracer --initialTimeSkip 1000000000000 -s 100000000000
- FM-SLR protocol:
 - bitsimulator after applying the patch
 - visualtracer --initialTimeSkip 1000000000000 -s 100000000000

slrcounter.xml is used with the following commands:

- Counter-based SLR protocol:
 - bitsimulator scenario_slrcounter.xml
 - visualtracer scenario_slrcounter.xml --initialTimeSkip
1000000000000 -s 100000000000

Simulation time in [Table 5.4](#) is calculated using the event file of each simulation:

Pure flooding protocol:

- (grep '^0 [0-9]* 2 [0-9]* 43' pureflooding-events.log) produces the event where the source sent the packet:
0 1000009901438 2 2 1000 100 0 555 0 0 -1 -1 -1
- (grep '^1 [0-9] 3 [0-9] 43' pureflooding-events.log) produces the events where the destination receives the packet. Since the destination receives the packet several times, only the first line (first time the destination receive the packet) is considered:
1 1000168336747 3 15419 1000 100 555 0 0 -1 -1 -1
- the substration of the second number (simulation time) of the reception event (1000138626774) from the second number of the sent event (1000009901438) represents the time taken for the packet the reach the destination.

Probabilistic flooding protocol:

- (grep '^0 [0-9]* 2 [0-9]* 43' probabilisticflooding-events.log) for the sent event.
- (grep '^1 [0-9] 3 [0-9] 43' probabilisticflooding-events.log) for the reception event

Backoff flooding protocol:

- (grep '^0 [0-9]* 2 [0-9]* 43' backoffflooding-events.log) for the sent event
- (grep '^1 [0-9] 3 [0-9] 43' backoffflooding-events.log) for the reception event

counter-based SLR protocol:

- (grep '^0 [0-9]* 2 [0-9]* 43' slrcounter-events.log) for the sent event
- (grep '^1 [0-9] 3 [0-9] 43' slrcounter-events.log) for the reception event

SLR, FM-SLR and protocols (after the simulation of each protocol):

- (grep '^0 [0-9]* 2 [0-9]* 43' events.log) for the sent event
- (grep '^1 [0-9] 3 [0-9] 43' events.log) for the reception event

A.3/ NETWORK PARAMETER INFLUENCE

This section provides the steps to reproduce the simulation results presented in chapter 6.

After downloading and compiling the simulator, download and extract the Zip folder NanoComSimulation.zip. This folder contains the data files to reproduce all the tables (scenario.xml, bash.sh, and bash.awk files).

Figure 6.1 can be reproduced by running Bitsimulator and visualtracer command in "NanoComSimulation/NanoCom Simulation/Table 3 - Communication Range/Heterogeneous network" path.

Figure 6.2 can be reproduced by running bitsimulator in any of the 'NanoCom Simulation' folders and run (visualtracer --initialTimeSkip 10000000000000 -s 10000000000000).

After extracting the downloaded NanoCom Simulation folder and to reproduce the tables presented in chapter 6 use the following:

- **Table 2 - Beta** folder to get the beta simulation results of Table 6.2
- **Table 3 - Communication Range** folder to get the communication range simulation results of Table 6.3
- **Table 4 - Pulse duration** folder to get the pulse duration simulation results of Table 6.4
- **Table 5 - Density** folder to get the density simulation results of Table 6.5

Note that each of the listed folders has subfolders. For example, in **Table 3 - Communication Range** folder there are 2 subfolders ("Homogeneous network" and "Heterogeneous network"). Run `./bash.sh` in each of them to get the results of the homogeneous network of Table 6.3 from the subfolder "Homogeneous network", and the heterogeneous network of Table 6.3 from the subfolder "Heterogeneous network".

Title: Enhancing routing protocols in dense nanonetworks

Keywords: Dense networks, routing, communication, nanonetwork

Abstract:

With the rapid development of dense electromagnetic nanonetworks, there is a pressing need for specialized routing protocols that can face the unique challenges of these ultra-dense environments. Nanoscale devices provide high bandwidth and face significant complexities in communication.

This thesis addresses the lack of viable routing protocols for dense nanonetworks by developing and evaluating a novel algorithm specifically designed for this context. A key contribution is the introduction of the EIDA (Equitable Distributed ID Assignment) mechanism, which balances the trade-offs between ideal and random ID assignments in dense networks. EIDA is then integrated into the FR-SLR (Forwarder Reduction in SLR Routing) protocol, where it serves as the ID assignment mechanism. FR-SLR optimizes the existing SLR protocol by reducing the

number of packet forwarders and ensuring best-effort equitable packet forwarding.

Further, this thesis explores the impact of various network parameters (β , communication range, density, and pulse duration) on communication quality (number of packet collisions, reception, emission, and delivered), providing valuable insights for selecting appropriate network parameters. It also delves into concurrency mechanisms such as constructive interference and capture effect, identifying the levels of concurrency that can be effectively supported in nanonetworks (pulse, bit, packet, and flow level).

Due to the current challenges in fabricating nanomachines, the proposed protocol, mechanism, and evaluation were validated through simulations using BitSimulator, a simulator allowing the simulation of ultra-dense nanonetworks.

Titre : Algorithmes novateurs dans les réseaux denses

Mots-clés : Réseaux denses, protocole de routage, communication, nanoréseaux

Résumé :

Avec le développement rapide des nanoréseaux électromagnétiques denses, il existe un besoin urgent des protocoles de routage spécialisés capables de relever les défis uniques de ces environnements ultra-denses. Les dispositifs à l'échelle nanométrique offrent une bande passante élevée et sont confrontés à des complexités de communication importantes.

Cette thèse aborde le manque de protocoles de routage viables pour les nanoréseaux denses en développant et en évaluant un nouvel algorithme spécifiquement conçu pour ce contexte. Une contribution clé est l'introduction du mécanisme EIDA (Equitable Distributed ID Assignment), qui équilibre les compromis entre les attributions d'id idéales et aléatoires dans ces réseaux. EIDA est ensuite intégré au protocole FR-SLR (Forwarder Reduction in SLR Routing), où il sert de mécanisme d'attribution d'id. FR-SLR optimise le protocole SLR existant en réduisant le nombre de redirecteurs de paquets et en garantissant une transmission

équitable des paquets au meilleur effort.

De plus, cette thèse explore l'impact de divers paramètres réseau (β , portée de communication, densité et durée d'impulsion) sur la qualité de la communication (nombre de collisions de paquets, réception, émission et livraison), fournissant des informations précieuses pour sélectionner les paramètres réseau appropriés. Elle s'intéresse également aux mécanismes de concurrence tels que l'interférence constructive et l'effet de capture, en identifiant les niveaux de concurrence qui peuvent être efficacement pris en charge dans les nanoréseaux (au niveau d'impulsion, du bit, du paquet, et du flux).

En raison des défis actuels dans la fabrication de nanomachines, le protocole, le mécanisme et l'évaluation proposés ont été validés par des simulations à l'aide de BitSimulator, un simulateur permettant la simulation de nanoréseaux ultra-denses.