Université de Lille
École Doctorale Mathématiques, sciences du numérique et de leurs interactions
Institut de Recherche Centre Inria de l'Université de Lille

Thèse préparée par **Jana Koteich**
en vue de l'obtention du grade de **Docteur en Informatique**
Discipline **Informatique et Applications**

Routage opportuniste tenant compte du contexte dans les réseaux sans fil

Thèse soutenue le 20 septembre 2024 devant le jury composé de :

Fabrice Valois Professeur des Universités, INSA Lyon	**Président et Examinateur**
André-Luc Beylot Professeur des Universités, ENSEEIHT	**Examinateur**
Aline Carneiro Viana Directrice de Recherche, Inria	**Rapporteure**
Thomas Noël Professeur des Universités, Université de Strasbourg	**Rapporteur**
Nathalie Mitton Directrice de Recherche, Inria	**Directrice de Thèse**

University of Lille
Doctoral School MAthematics and DIgital Sciences
Research Institute Inria Centre of the University of Lille

Thesis prepared by **Jana Koteich**
with the view of obtaining the degree of **PhD in Computer Science**
Discipline **IT and Applications**

Novel Context Aware Opportunistic Data Forwarding Strategy in Wireless Networks

Thesis defended the 20th September 2024 in-front of the jury comprised of:

Fabrice Valois Full Professor, INSA Lyon	**President and Examiner**
André-Luc Beylot Full Professor, ENSEEIHT	**Examiner**
Aline Carneiro Viana Research Director, Inria	**Reviewer**
Thomas Noël Full Professor, Université de Strasbourg	**Reviewer**
Nathalie Mitton Research Director, Inria	**Supervisor**

"For every oppressed person..."

Acknowledgements

This thesis was carried out at Inria, France, from October 2021 to September 2024. It has been a rewarding experience marked by intellectual achievement and personal growth. This work would not have been possible without the support and guidance of several people who have contributed in countless ways. To each of them I express my sincere gratitude.

First and foremost, I would like to express my heartfelt gratitude to my supervisor, Ms. Nathalie Mitton, for making my Ph.D. journey a successful and outstanding phase of my life. Words can't do justice to what I mean, but her guidance, patience, support, and confidence in me gave me the strength and motivation to continue. Nathalie has had a great impact on my life, on a scientific and human level, she is the true meaning of a great leader and supervisor. Thank you Nathalie for your moral support, for believing in me and for caring about me, especially during the hard times I went through! It has been my greatest pleasure to work with you, and I will forever cherish this opportunity and carry the lessons you have taught me throughout my life. No matter what I say, it will not be enough, but I owe you so much!

I am also deeply thankful to the members of my jury and the reviewers for their time and valuable insights: Mr. Fabrice Valois, Mr. André-Luc Beylot, Ms. Aline Carneiro Viana, and Mr. Thomas Noël.

A big thank you to my colleagues in the FUN team during these three and a half years! Marwa, Hazem and Carol (thank you for taking care of me and for your honest friendship), Christian (my former Master's supervisor, none of this would have been possible without your initial trust and support!), Damien (Thank you for all the discussions and for having so much confidence in me), Gérardo (Thank you for being supportive and for the deep valuable discussions and unique friendship), Bingying (Thank you for believing in me more than myself, it is my greatest blessing to have you as a close friend), Hervé, Nina, Adriana, Lucille, Prakriti, Nasima, Edward, Brandon, Yasir, Ildi, Jiali, Selina...the list is long and I appreciate each and every one of you, mentioned or not.

As part of my Ph.D., I had the great pleasure to be a visiting researcher at Stellenbosch university working with Dr. Riaan Wolhuter. I am deeply grateful to Dr. Wolhuter for his warm welcome and extreme generosity. Working with and knowing Dr. Wolhuter has been a milestone in my PhD, so I extend my sincere heartfelt gratitude for his valuable time and discussions. Also to my great colleagues that I met there from the DSP lab, namely Christiaan, Julian, Quintin, Triven, Emi, Nikola...Thank you for taking care of me during my stay in Stellenbosch, and for making me feel at home. And here I would like to deeply thank Christiaan for his help in my academics and moral support all the way, and unique friendship, and for always believing in me. A sincere thank you to the Steenkamp family for hosting me in SA, especially Maretha and Christine who became my second family.

My deepest gratitude to my Lebanese friends and family who held me during these three years: Mira, Zahraa, Fatima K, Fatima G, Sara, Ghina, Doha, Zianab, Alaa, Hussein, Fatima H, Ali, Aya, Mohamad... Thank you for carrying me this far.

Israa and Zainab Issa, thank you for being my support system through thick and thin. Thank you for making me feel at home and for genuinely caring about me. I owe you a lot!

Finally, to my family, who made all of this possible, who supported me and my dreams. My mom who will always be my teacher, my dad who gave me strength, my five awesome siblings: Mohamad, Fatima, Bilal, Ali, Ghina, and my awesome sister-in-law Malak, You are the reason for being who I am, and giving my best in this life; I owe all that I am to you.

Résumé

Aujourd'hui, le partage des données et du contenu numérique est essentiel dans diverses applications, notamment dans les domaines de la santé, de l'éducation et de l'agriculture. Les réseaux câblés traditionnels et les réseaux sans fil sont les deux principaux modes de communication, ces derniers présentant davantage de difficultés en raison de l'absence de chemin physique guidé. Le routage opportuniste apparaît comme une solution prometteuse dans les régions dépourvues d'infrastructures de communication, en particulier dans les pays les moins avancés. Dans cette thèse, nous proposons une solution alternative aux approches basées sur l'infrastructure pour fournir des données indépendamment de toute infrastructure existante. Cette solution repose sur des dispositifs de communication et de stockage peu coûteux qui peuvent intégrer différentes technologies de communication, ce qui permet de créer un système global de partage de données préservant la vie privée et basé sur la mobilité naturelle des foules. Pour ce faire, nous analysons les schémas de mobilité de la foule afin d'attribuer une probabilité de livraison à un message en fonction de son schéma de mobilité. Tout d'abord, nous avons généré l'ensemble de données PILOT, une collection de données préservant la confidentialité des technologies de communication sans fil. L'ensemble de données se compose de quatre types d'informations collectées conjointement dans différents contextes de mobilité. Il comprend trois technologies de communication sans fil : les réponses des sondes WiFi, les balises BLE (Bluetooth Low Energy) et les paquets LoRa (Long Range Radio), ainsi que des informations supplémentaires sur l'accélération, le roulis et le tangage, toutes collectées simultanément. L'ensemble des données a été collecté pendant environ 90 heures, avec une taille de 200 Mo, en utilisant les dispositifs FiPy de Pycom. Nous avons fourni les clés permettant de reproduire cette collecte de données et partagé les ensembles de données déjà collectés sur GitHub. Après avoir généré l'ensemble de données, nous avons traité les traces collectées de WiFi et de BLE pour générer un modèle de classification capable d'estimer la situation réelle d'un appareil. Le premier modèle créé, appelé modèle B, vise à identifier si un appareil est stationnaire ou mobile. Par la suite, un modèle complémentaire, le modèle M, a été créé pour déterminer une situation plus précise de l'appareil dans la vie réelle, comme à la maison, au bureau, dans un bus, un train, etc. Enfin, nous avons exploité l'ensemble des données collectées et les modèles d'apprentissage automatique entraînés pour concevoir un protocole de routage en établissant des probabilités de livraison conditionnées par le contexte déterminé de l'appareil. Nous testons et validons notre approche en utilisant le simulateur ONE, qui est conçu pour un environnement de réseau opportuniste.

Abstract

Today, sharing data and digital content is essential across various applications, particularly in health, education, and agriculture. Traditional wired networks and wireless networks are the two main modes of communication, with the latter presenting more challenges due to the absence of a guided physical path. Opportunistic routing emerges as a promising solution in regions lacking communication infrastructure, especially in Least Developed Countries.

In this thesis, we propose an alternative solution to infrastructure-based approaches for delivering data independently of any existing operated infrastructure. This solution relies on low-cost communication and storage devices that can embed different communication technologies, resulting in a global privacy-preserving data-sharing system based on natural crowd mobility. To achieve this, we analyze crowd mobility patterns to assign a delivery probability for a message based on its mobility pattern. First, we generated the PILOT dataset, a privacy-preserving data collection of wireless communication technologies. The dataset consists of four types of jointly collected information in different mobility contexts. It includes three wireless communication technologies: WiFi probe responses, BLE (Bluetooth Low Energy) beacons, and LoRa (Long Range Radio) packets, as well as additional information on acceleration, roll, and pitch, all collected simultaneously. The dataset was collected over approximately 90 hours, with a size of 200 MB, using FiPy devices from Pycom. We provided the keys to reproduce such data collection and shared the datasets already collected on GitHub. After generating the dataset, we processed the collected traces of WiFi and BLE to generate a classification model that can estimate the real-life situation of a device. The first created model, called the B-model, aims to identify whether a device is stationary or mobile. Subsequently, a complementary model, the M-model, was created to determine a more precise real-life situation of the device, such as being at home, in the office, on a bus, train, etc. Finally, we exploited the collected dataset and the trained machine learning models to design a routing protocol by setting delivery probabilities conditioned by the determined context of the device. We are testing and validating our approach using the ONE simulator, which is designed for an opportunistic network environment.

Contents

Acknowledgements **vii**

Résumé **viii**

Abstract **ix**

Contents **x**

1 Introduction **1**
1.1 Context and Motivation 1
1.2 Delay Tolerant Networks 1
1.2.1 Routing Protocols in DTN 2
1.3 Thesis Objective and Methodology 4
1.3.1 Importance of Mobility Models in Routing 4
1.3.2 Leveraging Natural Mobility 4
1.3.3 Motivation behind mobility context 5
1.4 Contributions and Thesis Organization 6
1.4.1 Contributions 6
1.4.2 Thesis Organization 7
1.5 List of Publications 7

2 State of the Art **9**
2.1 Routing Protocols in DTN 9
2.1.1 Geographic-based Routing Protocols 9
2.1.2 Link-State Aware Routing Protocol 10
2.1.3 Cross Layer Opportunistic Routing Protocol 11
2.1.4 Probabilistic Routing Protocols 11
2.1.5 Optimization-based Routing Protocols 13
2.1.6 Context-aware Routing Protocols 15
2.2 Leveraging Radio Technology 16
2.3 Human Mobility in DTN Routing Protocols 17
2.3.1 Machine learning-based approaches 17
2.3.2 Analyzing Human Mobility 18
2.4 Summary 19

3 PILOT Dataset **21**
3.1 Background and Motivation 21
3.1.1 Online Datasets and literature review 21
3.1.2 Purpose of the chapter 23
3.2 Experimental Design 23
3.2.1 Active and Passive Scanning 24
3.2.2 Data Collection Devices 25
3.2.3 Configuration 25
3.3 Dataset Collection Methodology 29
3.3.1 Study Area and Different Scenarios 29
3.3.2 Privacy Preserving Technique 29
3.3.3 Dataset Description 30

3.4 Dataset Insights . . . 33
3.5 Challenges and Open Issues . . . 34
3.5.1 Hardware Level . . . 34
3.5.2 Software Level . . . 34
3.6 Conclusion and Integration with Thesis . . . 34

4 Machine Learning Approach **37**
4.1 Objective and Problem Definition . . . 37
4.2 Model Design . . . 37
4.2.1 Data Description . . . 38
4.2.2 Feature Selection . . . 38
4.2.3 WiFi and BLE Selected Features . . . 39
4.2.4 Feature Extraction: Phase 1 . . . 39
4.2.5 Feature Extraction: Phase 2 . . . 40
4.2.6 Model Architecture . . . 41
4.3 Models Training and Evaluation . . . 42
4.3.1 Training Process or Model training . . . 42
4.3.2 Evaluation Metrics . . . 42
4.3.3 Models Evaluation . . . 43
4.3.4 B-Model: Binary Classification . . . 44
4.3.5 M-model: Multi-class Classification . . . 45
4.4 Conclusion and Discussion . . . 49

5 Routing Protocol **51**
5.1 BatonRelay Routing Protocol . . . 51
5.1.1 Design Principles . . . 52
5.2 Simulation Setup . . . 53
5.2.1 Tools and Scenario Description . . . 54
5.2.2 Parameters . . . 54
5.2.3 Performance Metrics . . . 54
5.2.4 Statistical Robustness in Simulation . . . 55
5.2.5 Confidence Interval Calculation . . . 55
5.3 Results and Discussion . . . 56
5.3.1 Performance Analysis of BatonRelay Protocol . . . 56
5.3.2 Comparison with existing Protocols . . . 57
5.4 Conclusion and Ongoing Work . . . 60

6 General Conclusion and Perspective **63**
6.1 Conclusion . . . 63
6.2 Future Work and Perspective . . . 65
6.2.1 Future Work . . . 65
6.2.2 Perspective . . . 65

Bibliography **67**

List of Figures

1.1 Infrastructure based network . . . 1
1.2 Ad hoc network . . . 1

1.3 Transition between real-life context and network model . . . 4
1.4 Illustration of different network observations . . . 5
1.5 Full overview . . . 6

2.1 Epidemic Routing [38]. . . . 14
2.2 Spray and Wait Routing [35]. . . . 14

3.1 FiPy microcontroller from Pycom . . . 25
3.2 Final setup for scanning . . . 25
3.3 A simple design for encapsulating the whole setup . . . 25
3.4 External shield - Pysense . . . 26
3.5 External shield - Pytrack . . . 26
3.6 Data collection process diagram . . . 27
3.7 Examples of the log files of the collected dataset for each technology . . . 28
3.8 Map of the collected data places . . . 29
3.9 WiFi data from an office . . . 30
3.10 WiFi data from a bus . . . 30
3.11 WiFi data from a car . . . 30
3.12 WiFi data from a train . . . 33
3.13 Results from node L (dwell time 120ms) . . . 33
3.14 Results from node S (dwell time 20ms) . . . 33
3.15 Dwell time experimental setup . . . 34
3.16 Gap due to hardware disconnection . . . 34

4.1 Dataset distribution . . . 38
4.2 Data collection process diagram . . . 41
4.3 Period comparison for B-model . . . 46
4.4 Accuracy comparison for three different input datasets . . . 46
4.5 Confusion Matrix . . . 49

5.1 Illustration of Baton-relay protocol . . . 52
5.2 Helsinki Map . . . 54
5.3 TTL comparison . . . 59
5.4 TTL comparison . . . 60

List of Tables

2.1 Comparison of Routing Protocols [45]. 15

3.1 Records from Mobile Scenarios 31
3.2 Records from Static Scenarios 32

4.1 Records from dataset analysis 41
4.2 M-model binary classification period comparison 45
4.3 Comparison between best three models 47
4.4 B-model models evaluation and comparison 47
4.5 M-model with B-model input feature 48
4.6 Class-wise accuracy 48

5.1 Routing Protocol Parameters 54
5.2 Routing Protocol Parameters 55
5.3 Comparative Performance of BatonRelay vs. BatonRelay-Reversed Routing Protocols 57
5.4 Comparison of Routing Protocol Performance 58

Introduction | 1

1.1 Context and Motivation

In today's interconnected world, the sharing of data and digital content has become a fundamental aspect of nearly every application. Whether it involves business operations, educational resources, social media interactions, or scientific research (in particular, to support health, education, and agriculture), the ability to seamlessly exchange information and multimedia is critical. So, as a network is built for the data flow, broadly there are two main modes of the communication system, the traditional wired networks and the wireless networks. While traditional networks function on end-to-end wired communication, wireless networks become more and more challenging since the transfer medium is not guided by a physical path of wiring, and different possibilities arise to decide the path of a message. Thus for the wireless medium, there exists the Infrastructure mode (Figure 1.1), where the nodes communicate through an access point. But there still exist some remote or isolated regions, especially in some Least Developing Countries, where the communication infrastructure does not exist whereas this is where it could bring the most [1]. Even when it exists, it is prone to failure caused by a power failure, traffic overload, a disaster destroying the physical infrastructure or a cyber-attack. There is thus a crucial need to develop new innovative concepts to potentially complement powered and operated infrastructures but independently from them, to reduce the stress on communication infrastructures and deliver data even to isolated areas. Here, the other mode of communication appears, which is the Ad Hoc mode, where the nodes communicate directly with each other due to the absence of infrastructure (Figure 1.2). In our research, we are interested in addressing these challenges by exploring innovative solutions that improve the resilience and efficiency of data transmission in Delay Tolerant Networks (DTN). By addressing the challenges in such a network, we aim to contribute to the development of more robust and secure communication strategies for remote and challenging environments.

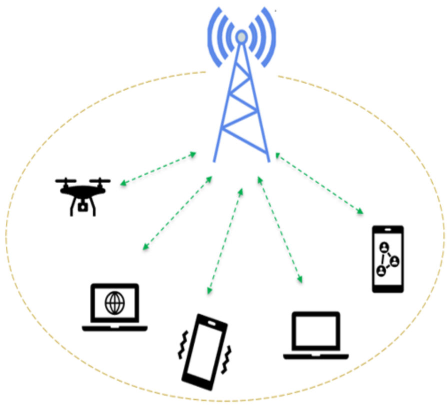


Figure 1.1: Infrastructure based network

1.2 Delay Tolerant Networks

The development of DTNs has been driven by the need to overcome communication barriers in situations where traditional network assumptions do not apply. Examples of such scenarios include deep space communication, disaster recovery, and rural areas that are remote and hard to reach. In a delay-tolerant network, the nodes do not know the topology initially until they discover it. If a new node comes into the network, it should declare its presence and listens to broadcasts from neighboring nodes. Every node present within the network studies the neighbour's node's behaviour and how to reach them.

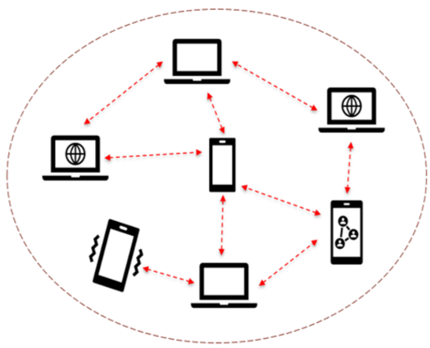


Figure 1.2: Ad hoc network

DTNs are designed to function effectively in situations where traditional networks fail due to limited connectivity, extended delays, and recurrent disruptions. They are characterized by their ability to store and forward data, adapt to changing network conditions, and maintain communication even in the face of disruptions. They are typically decentralized, self-organizing and fault tolerant. Here are the key characteristics and features of DTNs that are also considered as challenges in such networks:

- Network Connectivity: DTNs often operate in environments with limited or no network connectivity, such as in disaster-stricken areas, remote regions, or interplanetary space.
- End-to-End Communication: In DTNs, data packets can be stored and forwarded by intermediate nodes before reaching their destination. This results in a time-consuming process, as packets may be queued and stored in nodes' buffers for extended periods.
- Energy Efficiency: Devices in DTNs, such as smartphones or sensors, often have limited battery life. Ensuring that these devices can communicate without draining their batteries is a significant challenge. Energy efficiency is crucial for the sustainability and scalability of DTNs.
- Security and Trust: DTNs rely on trust relationships between nodes to ensure the integrity of data transmitted across the network. In such networks, nodes may forward packets without verifying their authenticity, making them vulnerable to malicious attacks or data corruption. Developing robust security measures and building trust among nodes is essential for the reliable operation of DTNs.
- Routing Protocols: DTNs require specific routing protocols to manage the storage and transfer of data packets across the network. The selection of routing protocols can significantly impact the overall performance of the network.
- Bandwidth constraints: Many factors are affecting wireless communication such as multiple access, interference conditions, noise and signal fading.

In summary, the difficulties encountered in a Delay Tolerant Network (DTN) highlight the need for advanced techniques to ensure efficient and secure data management and transmission. DTNs often operate under severe resource constraints, including limited bandwidth, storage capacity and power, requiring careful resource management. The intermittent connectivity in DTNs requires the development of robust data storage and transmission mechanisms to cope with frequent interruptions. In addition, the significant latency inherent in these networks requires effective buffering and scheduling methods to manage delays in data delivery. In addition, protecting user identities and sensitive information in dynamic and potentially untrusted environments creates an additional layer of complexity which arises the importance of anonymity and privacy. Overcoming these challenges is critical to optimising the performance and reliability of DTNs.

1.2.1 Routing Protocols in DTN

In any type of communication, we need routing protocols to establish the connection between nodes by the selection of routes on which data packets are to be transmitted. The traditional routing protocols in wireless network performs best path routing that pre-selects one or more optimized fixed routes before the actual transmission starts (This strategy simply applies the main operations and principles inherited from routing solutions in wired networks). The traditional wireless routing protocols are not efficient in dynamic wireless environment variation, since they trigger excessive link-level re-transmissions and waste of network resources[2]. From this situation, a challenging question arises: *How to disseminate data to the users with the absence of infrastructure?* To this end, DTN routing protocols were designed to tackle such challenges by using store-and-forward techniques. In this approach, data packets are temporarily stored at intermediate nodes until a dependable path to the final destination becomes accessible. These protocols take account of the dynamic nature of the network when determining the best routes for data packets to transit from their source to their destination. In this case, the message will start an unknown *Journey* to reach the destination.

The advancement of routing protocols in DTNs represents a persistent effort to balance the trade-offs between delivery probability, latency, and resource utilization. From the early days of flooding-based routing like Epidemic routing [3] and probabilistic methods like PRoPHET protocol [4] to more sophisticated techniques that employ social dynamics and probabilistic models. Each development has played a role in making DTNs a feasible solution for communication in challenging and sporadically connected environments. As research continues to advance, new developments in DTN routing protocols are anticipated to further improve their efficiency, dependability, and scalability, allowing for their application in a broader range of situations.

Thus, several researchers have made the effort to study human mobility to determine people's fine-grained activities like using GPS positioning [5], but GPS-based mobility characterization raises many issues such as spotty coverage and battery consumption [6]. Other attempts and tools have been developed to predict global mobility [7] in general or only the next step [8]. Some studies [9] aimed to characterize the mobility of people based on data traffic, but only for a particular subset of people (students in higher education) and over very limited areas. Recently, some studies investigated the use of human mobility but mainly in the COVID-19 context to anticipate contamination [10]. Such approaches are different in the sense that they mainly aim to trace contacts between devices and not necessarily their mobility. Other approaches used mobility models to evaluate the forwarding protocols, while **Context-Aware and Adaptive Routing** appeared, these protocols consider various contextual information, such as node mobility patterns, energy levels, and current network conditions. By adapting to the current context, these protocols aim to optimize routing decisions dynamically.

Very promising are those approaches that exploit information on the context the users are surrounded by, like the other users they met, the places they visit, etc [11]. But these solutions still assume that people will be equipped with laptops and energy supply, which is not always the case, and data transit through an external operator, a source of potential personal data leaks, as well such approaches requires a large history and memory storage, that is why it is important to take the decision for routing a message from a global context-aware perspective, taking into advantage the crowd mobility of devices while being privacy-friendly and leveraging the various wireless technologies in a network.

[2]: Nessrine Chakchouk. 'A Survey on Opportunistic Routing in Wireless Communication Networks'. In: *IEEE Communications Surveys Tutorials* 17.4 (2015). DOI: 10.1109/COMST.2015.2411335

[3]: Amin Vahdat, David Becker, et al. *Epidemic routing for partially connected ad hoc networks*. 2000

[4]: Anders Lindgren, Avri Doria, and Olov Schelén. 'Probabilistic routing in intermittently connected networks'. In: *ACM SIGMOBILE mobile computing and communications review* 7.3 (2003)

[5]: Gonzalo M. Vazquez-Prokopec, Donal Bisanzio, Steven T. Stoddard, Valerie Paz-Soldan, Amy C. Morrison, John P. Elder, Jhon Ramirez-Paredes, Eric S. Halsey, Tadeusz J. Kochel, Thomas W. Scott, and Uriel Kitron. 'Using GPS Technology to Quantify Human Mobility, Dynamic Contacts and Infectious Disease Dynamics in a Resource-Poor Urban Environment'. In: *PLOS ONE* 8.4 (Apr. 2013). DOI: 10.1371/journal.pone.0058802

[6]: Li Liu. 'Interest-based prediction routing protocol in socially-aware opportunistic networks'. In: *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*. 2013. DOI: 10.1109/IMSNA.2013.6743293

[7]: Douglas Teixeira, Aline Carneiro Viana, Jussara Marques Almeida, and Mário S Alvim. 'Revealing challenges in human mobility predictability'. In: *ACM Transactions on Spatial Algorithms and Systems* (2021)

[8]: Douglas Teixeira, Mário Alvim, and Jussara Almeida. 'On the predictability of a user's next check-in using data from different social networks'. In: *Proc. of the 2nd ACM SIGSPATIAL Workshop on Prediction of Human Mobility*. 2018

[9]: Joon Heo, Kyong-Mee Chung, Sanghyun Yoon, Sung Bum Yun, Jong Won Ma, and Sungha Ju. 'Spatial-data-driven student characterization in higher education'. In: *Proc. of the 1st ACM SIGSPATIAL Workshop on Prediction of Human Mobility (PredictGIS)*. 2017

[10]: Enrique Hernández-Orallo and Antonio Armero-Martínez. 'How Human Mobility Models Can Help to Deal with COVID-19'. In: *Electronics* 10.1 (2021). DOI: 10.3390/electronics10010033

[11]: Chiara Boldrini. 'Design and Analysis of Context-Aware Forwarding Protocols for Opportunistic Networks'. In: *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*. MobiOpp '10. Pisa, Italy: Association for Computing Machinery, 2010. ISBN: 9781605589251. DOI: 10.1145/1755743.1755788. URL: https://doi.org/10.1145/1755743.1755788

1.3 Thesis Objective and Methodology

The aim is to build a distributed low cost large scale privacy-friendly and mobility-aware delay tolerant data delivery network based on multi-technology rather than relying on full Internet connectivity between expensive powerful personal laptops or mobile phones. Our aim is to design a novel approach for routing protocols in Delay Tolerant Networks that relies on the natural crowded based mobility small data-storage communicating devices that can embed different communication technologies offering both some short range technologies with high throughput (such as WiFi, BlueTooth) and long-range technologies with low energy consumption (such as LoRa). These devices will embed a battery and this energy needs to be used with parsimony. Data stored on the device can be written and accessed through a USB, Bluetooth or WiFi connection with a PC or a smart phone. Data will travel between devices in a seamless manner through wireless connections, leveraging the natural mobility of population (and device holders).

1.3.1 Importance of Mobility Models in Routing

Mobility models play a crucial role in the performance of routing protocols, particularly in DTNs and ad hoc networks, where node movement significantly impacts network dynamics. Mobility models simulate the movement patterns of nodes, providing insights into network behavior and helping to develop more effective routing strategies.

Understanding and modeling humans and device mobility have fundamental importance in mobile computing, with implications ranging from network design and location-aware technologies to urban infrastructure planning [12]. So inferring mobility states such as being stationary, walking, or driving is critical for several applications. The fact that these days users carry several devices such as smartphones, laptops, and smartwatches equipped with radio communication technologies with each device offering a different set of services resulting in different usage and mobility, provides new opportunities and means for studying human mobility.

[12]: Amee Trivedi. 'Human Mobility Monitoring using WiFi: Analysis, Modeling, and Applications'. PhD thesis. University of Massachusetts Amherst, USA, 2021

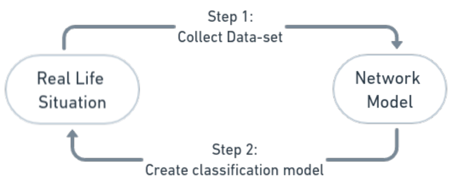


Figure 1.3: Transition between real-life context and network model

1.3.2 Leveraging Natural Mobility

Based on what we mentioned as a main objective, our approach will rely on the knowledge that could be extracted from natural crowd mobility to translate each 'real-life' situation into a network model. As 'real-life' situation, latter referred as context, we refer to the mobility and environment of a device: static, low mobility (e.g. pedestrian), high mobility (e.g. car, bus), at home, in a public place, in an urban or rural environment, etc. We define as a network pattern the number, density, and stability of the wireless links that a device can observe at a given time for each wireless technology.

But the question is, how to determine the context of a device through the observation of the surrounding network? In this section, we motivate the concept behind studying mobility context through radio beacons,

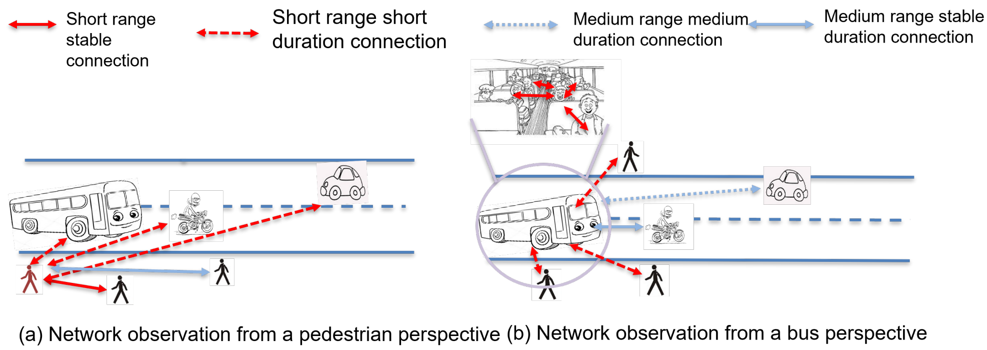


Figure 1.4: Illustration of different network observations

and how it is possible to leverage wireless links to determine a network context.

A wireless connection between two devices can be established if they are close enough to communicate. The maximum distance required between them to establish a contact depends in particular on the environment and the communication technology. So we believe the number of direct links a single device can establish at a given time and the rate at which these links break and appear are strong indicators of the context in which a device evolves (static vs. mobile, urban vs. rural environment, isolated vs. social, transport publication vs. individual locomotion, velocity, etc.). Figure 1.4 illustrates two different perspectives in a single scenario. Connections are observed from a device held by a pedestrian (Fig. 1.4a) or from a device traveling on a bus (Fig. 1.4b). When in a bus, several stable short range communications can be established with other bus passengers. They are completed by a set of longer range intermittent communications that can be sporadically established with devices exterior to the bus, such as pedestrians or cars. In contrast, when a pedestrian is holding the position, there are more or less stable short-range communications and few brief longer-range communications. Thus, from a network perspective we can translate the mobility and the surrounding environments of a device into a network pattern, i.e. when a user walks in an urban area, on average, what and how many connections are they supposed to have and at what rate are they changing. The same question arises when a user is in a bus, a cab, biking, etc. and in various scenarios. To achieve this, we first need to observe the variations of different wireless links in different scenarios. This requires collecting dataset from each real-life context. The idea is to translate the real-life situation into a network model, as later from the network model we will be able to guess the real-life situation of a device (Figure 1.3).

1.3.3 Motivation behind mobility context

Figure 1.5 summarizes the global overview of our objective and how the different challenges arise. When data has to be sent to a given destination (Starting point on Figure 1.5), the holding device scans its network environment (Step 1) to determine in what 'real life situation' it is (Step 2). This allows it to estimate a delivery probability for itself, but also for its neighbors (Step 3). Then, given different delivery probabilities in the

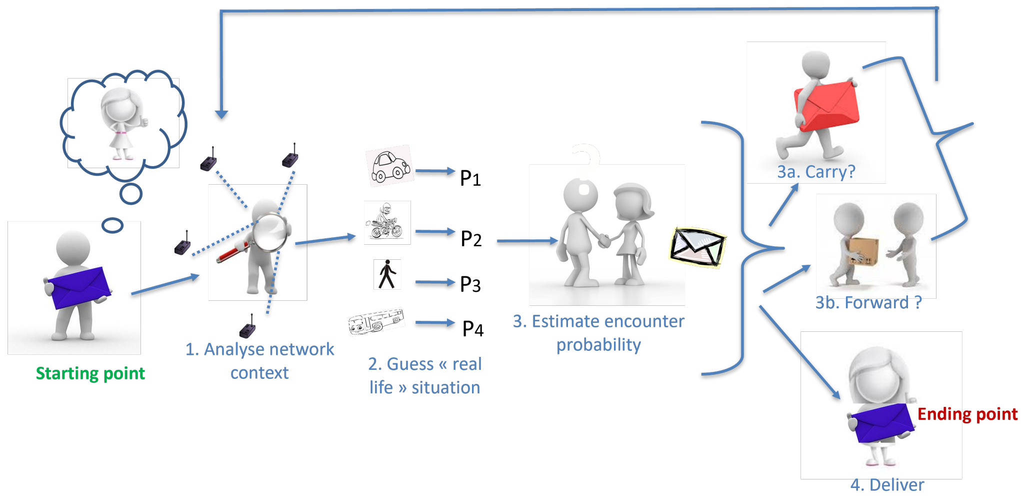


Figure 1.5: Full overview

device's environment, it will decide to carry the message or forward it, taking care of mobility-enabled path diversity, and so until either the data is delivered to its destination (Step 4 - Ending point) or it continues processing the same steps from Step 1.

Accordingly, three outstanding challenges have been identified and will be addressed, each challenge consisting in answering a part of the fundamental underlying question for designing the routing protocol, i.e. when, to whom and how to send the data. These challenges are:

- Challenge C1: model different network views generated from natural user mobility (Figure 1.3),
- Challenge C2: understand and identify them to determine the real-life situation of a device,
- Challenge C3: design the routing protocol to decide whether to send a data or carry it based on a probability derived from the mobility of a device.

We examine various wireless technologies and data collection methods and discuss how this data can be analyzed to gain insights into crowd behavior.

1.4 Contributions and Thesis Organization

1.4.1 Contributions

In order to address the identified challenges and achieve the overall objective, the work of the thesis project will be divided into 4 main contributions as follows:

- Contribution 1: In the first contribution, we collected PILOT dataset to get the data we needed for our analysis
- Contribution 2: In the second contribution, we analyzed the dataset to address Challenge 1.
- Contribution 3: In the third contribution, we designed a machine learning model to address Challenge 2

- Contribution 4: Finally in the last contribution, we exploit the understanding of the mobility type to design a routing protocol and this is how we tackled the last challenge.

1.4.2 Thesis Organization

The thesis is organised into six chapters. As we first introduced the general motivation and objective behind this work, the rest of the thesis is organized as follows:

- In Chapter 2 we present a comprehensive and in-depth review of the current state of the art in routing protocols for Delay Tolerant Networks (DTNs).
- In Chapter 3 we introduce the first and the second contributions. In this chapter we explain the reason behind collecting our own dataset and illustrate all the steps needed for generating it.
- In Chapter 4 we present the third contribution that exploits the dataset to build a machine learning model to determine the real-life situation of the devices.
- In Chapter 5 we present the last contribution which is the design of the routing protocol, where the decision of routing is based on the mobility of a device.
- In Chapter 6 we summarise the contributions of this work, outlining possible current research directions and perspectives.

1.5 List of Publications

Journals

Jana Koteich, Christian Salim, and Nathalie Mitton. 'Image processing based data reduction technique in WVSN for smart agriculture'. In: *Computing* 105.12 (2023)

International Conferences

Jana Koteich and Nathalie Mitton. 'Dataset Collection of Multi-Communication Technologies Monitored in Different Mobility Contexts'. In: *The 20th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE. 2024

Jana Koteich and Nathalie Mitton. 'Machine Learning Approach for Mobility Context Classification using Radio Beacons'. In: *2023 31st International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE. 2023

International Workshops

Jana Koteich, Christian Salim, and Nathalie Mitton. 'Spatio-temporal data reduction technique in WVSN for smart agriculture'. In: *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2022

Jana Koteich, Christian Salim, and Nathalie Mitton. 'Data reduction and frame rate adaptation in wvsn'. In: *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2021

National Conferences

Jana Koteich and Nathalie Mitton. 'Radio Beacon Base Context Identification'. In: *CoRes 2024: 9èmes Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*. Saint-Briac-sur-Mer, France, May 2024. URL: https://hal.science/hal-04554003

Jana Koteich and Nathalie Mitton. 'PILOT Dataset: A Collection of Multi-Communication Technologies in Different Mobility Contexts'. In: *CoRes 2023-8th Francophone Meeting on Protocol Design, Performance Evaluation and Communication Network Experimentation*. 2023

2 State of the Art

Delay-tolerant networks (DTNs) are characterized by intermittent connectivity, frequent disconnections, and high latency. In DTNs, routing frequently employs node mobility to help deliver messages through a store-carry-forward method. This section offers a comprehensive summary of the most advanced routing protocols in DTNs that utilize node mobility. They are classified according to their underlying approaches, which are then evaluated using various mobility models.

To generate a comprehensive state of the art, we structure the information chronologically, focusing on the evolution of routing approaches, the importance of human mobility and the rationale for studying node mobility in routing protocol design.

2.1 Routing Protocols in DTN

Routing in DTN drew great attention within the last decades, thanks to its numerous advantages over traditional routing. It can be applied to all kinds of wireless multihop networks, such as ad hoc, mesh, and sensor networks, as long as omni-directional antennas are used [2]. The approaches appeared in the literature are mainly classified into five major categories: Geographic, Link State Aware, Probabilistic, Optimization-based and Cross Layer Opportunistic Routing.

2.1.1 Geographic-based Routing Protocols

Geographic based routing protocols in DTN leverage the location information to make forwarding decisions. It is a useful approach in scenarios where nodes have access to their geographic positions. For example, the authors in [20] proposed a novel geographic routing protocol designed for Mobile Ad-hoc Networks (MANETs) operating in delay-tolerant situations. The protocol is designed to use one-hop information only to make routing decisions, reducing overhead and complexity. This protocol employs a utility function that is responsible for evaluating the potential next hops, and such a function likely considers factors such as geographic position and other relevant metrics to determine the best forwarding option. An improved approach has been proposed and presented in [21]. The authors presented a novel approach to geographic routing in opportunistic ad-hoc networks (OppNets) where end-to-end connectivity is infeasible due to intermittent connections and node mobility. The proposed protocol utilizes both one-hop and two-hop neighbor information, which includes location data and movement directions to make the routing decision. In [22], Zorzi *et al* proposed a geographic routing approach called GeRaF which focuses on the multi hop performance of such a solution, in terms of average number of hops to reach a destination as a function of the distance and of the average number of available neighbors. They evaluated the performance of this protocol in terms of the average number of hops to reach a destination.

Geographic-based routing protocols in DTN offer several advantages. They are scalable and well-suited for high node mobility patterns, as they do not require route discovery and management [23]. These protocols can make efficient forwarding decisions using location information, which is particularly useful in scenarios where nodes have access to their geographic positions [24]. However, these approaches also have some weaknesses. They typically rely on GPS or other positioning services, which may not always be available or accurate, especially in environments like tunnels where satellite signals are absent [23]. Additionally, while protocols using only one-hop information (like the one in [20]) reduce overhead and complexity, they may miss potentially better routing opportunities that could be identified with more comprehensive network knowledge [25]. On the other hand, protocols using two-hop information (like in [21]) may provide better routing decisions but at the cost of increased overhead and complexity. Furthermore, geographic routing protocols in DTN environments must deal with the challenges of intermittent connectivity and potential long delays. While they can use store-carry-forward mechanisms to handle these issues, this approach can lead to increased end-to-end delay and resource consumption on intermediate nodes [26].

2.1.2 Link-State Aware Routing Protocol

On the other hand, Link-State Aware routing protocols focus on the quality and characteristics of the links between the different nodes to make the best decision, aiming to improve the reliability and efficiency of the data transmission in critical environments where the connectivity is intermittent and unpredictable.

Several researchers have proposed Link-State Aware routing protocols for DTNs. For instance, in [27], He et al. introduced the Link State aware Hierarchical Road routing (LSHR) protocol for 3D scenarios in Vehicular Ad-hoc Networks (VANETs). This protocol selects the next intersection based on distance and road connectivity, prioritizing neighbors with the largest two-hop transmission range as forwarders. Another example is the LSGO (Link State aware Geographic Opportunistic) routing protocol for VANETs, proposed by Zhu et al., which combines geographic location and link state information for forwarder selection. In the context of hybrid DTNs, Mayer and Waldhorst explored routing protocols that incorporate link-state information to improve performance in networks with both DTN and non-DTN characteristics [28]. Their work highlights the potential of using link-state routing protocols in conjunction with DTN-specific approaches to enhance routing efficiency.

Link-State Aware Routing Protocols in DTNs offer improved reliability and performance by considering link quality for informed decision-making. They enhance adaptability to changing network conditions and can achieve higher throughput and lower packet dropping rates in dynamic environments. However, these protocols face challenges such as increased overhead for collecting and maintaining link state information, higher complexity in algorithms, and potential scalability issues in large networks. They also heavily depend on the accuracy and timeliness of link state information, which can be difficult to maintain in highly dynamic DTN scenarios. Additionally, frequent changes in link states could lead to routing instability if not managed properly.

2.1.3 Cross Layer Opportunistic Routing Protocol

Cross Layer Opportunistic Routing adopts the interaction and coordination between different layers of the network stack to enhance the data or message delivery where the end-to-end communication is unreliable or intermittent. Authors in [29] combined both the Link-State Aware and the Cross-Layer routing approaches and proposed a new routing protocol called *ExOR*, Extremely Opportunistic Routing. A new uni-cast routing technique for multi-hop wireless networks. It exploits the multiple transmission opportunities that the broadcast nature of the wireless medium creates. Jing Zuo et al [30] examined how cross-layer optimization across different network layers can be used to improve the energy efficiency of ad-hoc networks.

Cross Layer Opportunistic Routing offers several advantages. It improves reliability by leveraging metrics from multiple layers, enhancing data transmission in unreliable environments [31]. These protocols can significantly improve energy efficiency, which is crucial for networks with limited resources like Wireless Sensor Networks [32]. By combining information from different layers, they can make more informed routing decisions, leading to better utilization of network resources and optimal forwarder selection. However, these protocols also have some weaknesses. The integration of multiple layers increases the complexity of the routing algorithm, potentially leading to higher computational requirements and implementation challenges. There's often increased overhead due to the collection and processing of information from multiple layers, which may affect overall network performance, especially in high-traffic scenarios. Scalability can be an issue as the network grows, making it difficult to manage cross-layer interactions and maintain necessary state information. Additionally, the performance of these protocols heavily relies on the accuracy of the information gathered from different layers, and inaccurate or outdated information can lead to suboptimal routing decisions [33].

2.1.4 Probabilistic Routing Protocols

As the name suggests, in probabilistic routing protocols, a node makes routing decisions based on the probability of encountering the destination or the intermediate nodes that are more likely to meet the destination. This probability is often derived from historical encounter data. Each node maintains a probability estimate of delivering messages to each destination which is updated dynamically based on encounters and exchanges with other nodes. An innovative routing protocol designed specifically for the opportunistic network is given in [34]. The authors proposed a probabilistic routing model that leverages the meeting probabilities of nodes to route the data packets to the destination effectively. The models presented adapt to network's changing conditions by considering factors such as the last encounter time between nodes and utilising acknowledgement tables to manage the network load and prevent congestion. This protocol showed the ability to enhance the probability of message delivery while keeping overhead and latency within acceptable limits. In [3], the authors proposed Epidemic routing protocol, which is an early sparse probabilistic routing protocol proposed for DTN. It assumes that each node has unlimited storage space and bandwidth. Therefore, every node can store all the messages transmitted during "contact" phase. This uses the concept of database replication. Each node maintains a list of messages in the database called summary vector. This epidemic strategy is practically possible in case of a very sparse network and small sized message.

PROPHET (Probabilistic Routing Protocol using History of Encounters and Transitivity)

A part of the probabilistic routing protocol, **PROPHET** [4, 35] is a sophisticated routing protocol. It is designed to address the challenges imposed by intermittent connectivity and long delay paths, through leveraging a combination of historical encounter data and probabilistic decision making to facilitate effective data forwarding. The PROPHET operates on the principle that a node mobility patterns and historical interactions can be adopted to predict the likelihood of successful message delivery. For this purpose, each node maintains a record of its encounters with other nodes, tracks how frequently and for how long these encounters occur. This historical data is used to calculate the probability that a given node will eventually come into contact with a particular destination node. PROPHET employs the concept of transitivity, where the probability of successful delivery to a destination is inferred from intermediate nodes. If a node frequently encounters another node that, in turn, frequently encounters the destination node, PROPHET infers that there is a higher likelihood of successful delivery through these intermediary nodes. This adaptive mechanism allows PROPHET to make informed routing decisions even in the absence of continuous network connectivity, improving the chances of message delivery in challenging environments. By continuously updating encounter histories and leveraging probabilistic estimates, PROPHET offers a dynamic and efficient routing solution tailored for the unpredictable nature of delay-tolerant networks.

Probabilistic routing protocols, such as PROPHET, offer several advantages in DTNs. They can adapt to dynamic network conditions by leveraging historical encounter data, which improves the probability of successful message delivery. These protocols are particularly effective in environments with intermittent connectivity, as they do not rely on continuous end-to-end paths. Additionally, they can manage network load and prevent congestion by utilizing acknowledgment tables and probabilistic decision-making. However, probabilistic routing protocols also have some weaknesses. They often require significant storage and computational resources to maintain and update encounter histories, which can be challenging in resource-constrained environments. The reliance on historical data means that their performance can degrade if the mobility patterns of nodes change unpredictably. Furthermore, protocols like Epidemic routing assume unlimited storage and bandwidth, which is impractical in many real-world scenarios. This can lead to excessive overhead and resource consumption, particularly in dense networks.

2.1.5 Optimization-based Routing Protocols

On the other hand, Optimization-based routing protocols are protocols that aim to enhance the data delivery process through the application of mathematical and computational optimization techniques to determine the best route for message forwarding. Such protocols consider various network metrics and constraints to optimize routing decisions, often balancing trade-offs between delivery probability, latency and resource consumption. For instance, the author in [36] evaluated a method that utilizes optimization algorithms to enhance the routing in DTNs. Another optimization approach is given in [37] by which the authors discussed the optimization in terms of energy consumption in DTNs through the adoption of machine learning techniques to maintain a long lifespan for the network. [34] a combined approach is proposed through the integration of Cross-Layer and Optimization approaches. The authors presented a cross-layer optimization approach through the integration of network, transport and application layer functionalities, for IoT enabled opportunistic networks. The approach emphasizes the development of delay-tolerant routing algorithms and message ferrying strategies to ensure timely delivery of critical data.

Epidemic Routing

The epidemic routing protocol [35, 38] is known as a highly robust routing protocol designed for DTNs, where traditional routing strategies might fail due to the intermittent connectivity and high latency typical of these networks. This protocol is inspired by the spread of epidemics in biological systems, it operates on a principle similar to the propagation of a disease through a population. Each node in the network acts like an infectious agent, continuously seeking out other nodes to exchange data. Once two nodes are in contact, they exchange all the messages they have making sure that each message reaches as many nodes as possible as given in Figure 2.1. Such approach maximizes the likelihood of message delivery even in highly unpredictable network conditions. Although this routing protocol guarantees the eventual delivery of messages, it does so at the cost of significant resource consumption, from bandwidth and storage perspective due to the redundancy of message copies that multiply in number throughout the network. The redundancy can lead to inefficiencies and increased overhead especially when dealing with a limited resources network. Despite these drawbacks, the Epidemic protocol is valued for its simplicity and effectiveness in scenarios where maintaining connectivity is challenging, ensuring that messages have a higher probability of reaching their intended destinations over time [39]. Various studies evaluated the epidemic protocol. For example, [40] the authors provided a comprehensive analysis of different epidemic routing strategies including P-Q epidemic, epidemic with Time-To-Live (TTL) and epidemic with Encounter Count (EC). In [41] the authors proposed an enhanced version of the epidemic protocol for DTNs with the aim to improve the efficiency and reliability of the basic epidemic routing approach.

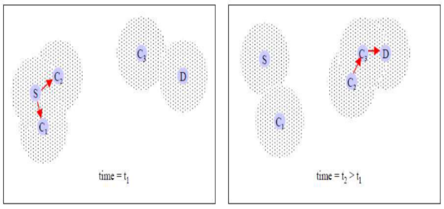


Figure 2.1: Epidemic Routing [38].

Spray and Wait

It comes as an enhancement of the Epidemic approach, where the core idea behind the Spray and Wait routing protocol is to balance the efficiency of resource utilization with the effectiveness of message delivery [35]. It operates in two distinct phases: the spray phase and the wait one as shown in Figure 2.2. During the spray phase, the source node initiates the message delivery process by spraying a fixed number of message copies referred to as the *spray limit*, to a set number of nodes within the network. These nodes are chosen based on their likelihood of improving connectivity and advancing the message closer to its destination. Each node that receives a copy of the message can further spray additional copies to other node, spreading the message across the network. The aim is to increase the likelihood of encountering the destination node while conserving network resources and avoiding excessive redundancy. When the first phase is completed, the protocol moves towards the wait phase. During this phase, the nodes that hold message copies enter a period of waiting and monitoring. Each node maintains its copy and continues to attempt direct delivery to the destination node whenever it encounters it. The key feature of this phase that it allows the nodes to optimize their transmission efforts based on their current network conditions, such as contact opportunities and node mobility, thereby enhancing the chances of successful delivery without further spraying. The Spray and Wait protocol effectively addresses the challenges of DTNs by combining a strategic initial distribution of messages with a flexible, opportunistic delivery mechanism. This approach helps in managing network resources efficiently, reducing the overhead associated with message forwarding, and improving the likelihood of successful message delivery even in highly dynamic and unpredictable network environments [42]. As mentioned before, this protocol was an improvement of the epidemic one. Spyropoulos *et al* proposed this Spray and Wait protocol and it was first introduced in [43]. Through this study, the authors improved the performance of Epidemic Routing by controlling the number of generated copies of a message in the network. This is achieved by assigning a constant number of logical tickets to each message, and thus they reduced the routing overhead caused by redundant message copies transmission but achieved a lower packet delivery ratio.

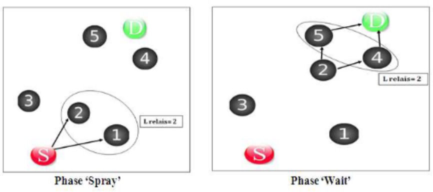


Figure 2.2: Spray and Wait Routing [35].

MAxProp

The MaxProp routing protocol [35, 44] is designed for DTN networks based on forwarding and on the other hand on mechanisms that optimize two routing metrics: the message delivery rate and the average latency or delivery delay. In other words, the main aim of this routing protocol is to improve the delivery rate of messages while minimizing latency and resource consumption which are significant challenges in DTN environments. This is done through the integration of message prioritization and scheduling mechanisms. When a node encounters another node, MaxProp prioritizes the messages to be transmitted based

Table 2.1: Comparison of Routing Protocols [45].

Routing Protocol	Latency	Energy Consumption	Bandwidth Efficiency	Storage Requirements	Reliability	Scalability	Additional Features
Epidemic	High due to the uncontrolled nature of flooding the network with message copies.	High energy consumption because each node transmits multiple copies of each message.	Low bandwidth efficiency due to the large number of redundant message copies which may lead to network congestion.	High since nodes need to buffer multiple copies of each message.	High since multiple copies increase the chances of message delivery.	Low due to exponential growth in message copies as network size increases.	Simple implementation, suitable for high-mobility scenarios but not resource efficient.
ProPHET	Moderate latency, it depends on encounter rates and mobility patterns.	Moderate energy consumption, it reduces the number of transmissions and uses probabilistic decisions to minimize unnecessary message forwarding.	Moderate through the minimization of redundant transmissions. It uses historical encounter data to make informed forwarding decisions.	Moderate as nodes need to store encounter history and message probabilities.	Moderate reliability by leveraging historical encounter data to improve delivery chances.	Moderate, it depends on encounter predictability and node encounters stability.	Ideal for environments where node encounters follow predictable patterns, such as daily commutes or routine movements.
Spray and Wait	Moderate latency, it balances between immediate delivery and waiting for contact opportunities.	Moderate due to controlled dissemination of message copies	Moderate by restricting the spread of message copies.	Moderate with nodes only storing a limited number of message copies.	Moderate, it balances spreading enough copies for reliable delivery and minimizing resource use.	Moderate compared to epidemic protocol.	Suitable for networks where nodes have moderate mobility and contact opportunities are somewhat predictable.
MaxProp	Low latency due to prioritized message scheduling and forwarding.	High energy consumption due to complex algorithms for prioritizing and scheduling messages. Nodes frequently compute and update priorities.	Moderate by prioritizing messages and reducing unnecessary transmissions.	High due to the need for large buffers to store prioritized messages.	High since messages are prioritized based on delivery likelihood and deadline.	Moderate scalability, effective in network with varying sizes and densities.	Effective in scenarios that require timely delivery of critical data such as emergency response or vehicular networks.

on the likelihood of successful delivery. This likelihood is This likelihood is calculated using historical data about encounters and the estimated delivery probabilities. Messages with higher probabilities of delivery are given higher priority, ensuring that they are transmitted first. One of the key features of MaxProp is its use of a hop count and acknowledgments to manage message priorities. Since they are more likely to arrive at their destination quickly, messages that have traveled through fewer hops are given priority over those that have traveled through several nodes.Additionally, MaxProp uses acknowledgments to track which messages have been successfully delivered, thereby preventing unnecessary retransmissions and reducing network congestion.

A brief comparison between the described optimization-based and probability-based routing protocols in DTN networks is given in Table 2.1.

2.1.6 Context-aware Routing Protocols

To overcome these limitations, recent work in the literature exploits the advantages of context-awareness to sense the physical environment of the devices and adapt their behavior for forwarding the message accordingly. To this end, in [46] the authors proposed a new routing protocol named Habit, a novel multi-layered approach to content dissemination in DTN that enables relevant content to reach interested nodes while minimising the load on uninterested intermediaries. The main parameter used to achieve the forwarding strategy is the users' co-location, which is logged and processed in order to learn regularity in human movement (physical layer). Second, the social network of users, which is dynamically propagated during periods of co-location (application layer). Finally, these two layers are combined to compute routes that content should follow, in order to have high probability of reaching interested node. While in [6] the authors presents a routing protocol named interest based prediction routing protocol (InP), which is based on the self-interest and second-interest information to predict the future meet opportunities. Each mobile node records and maintains self-interest for itself and second interest according to contact times in a time window and successful delivery record, and an ID that represents the sensitive strength of the interest. Based on this information, InP selects the higher meet probability to the destination device as forwarder in order to improve the efficiency. The simulation shows that Inp obtains the higher efficiency than Epidemic and PROPHET in higher delivery ratio, lower overhead and closer average latency.

Based on the importance of context-aware forwarding protocols, different mobility models should be investigated to help design protocols with less communication overhead and used resources. Traces generated by human mobility models can be used in the simulations of wireless ad hoc networks [47]. Very recently, some studies investigates the use of human mobility but mainly in the COVID-19 context in order to anticipate contamination [48]. The literature has widely investigated mobility models but rather to measure their impact on network performances and not to leverage their characteristics to improve the performances [49]. Thus in our objective, the generated mobility model will take a part in the decision of forwarding the message rather than only evaluation. With time we will investigate more in the literature regarding mobility models.

2.2 Leveraging Radio Technology

Some techniques in the literature have used radio frequency (RF) technologies for different use cases. This approach is important to validate the uniqueness of our application which is leveraging the presence of multiple communication technologies and assigning estimation probabilities based on the collected traces from the scanned network. In [50], Krieg *et al*, proposed SmartPhone, a system for real-time parking information that relies on two major functionalities to deliver on its promise: 1) Transportation mode detection, and 2) Location matching. SmartPark enables the automatic detection of a user vacating a parking space by leveraging the smartphone's sensors and the ubiquitous Wi-Fi and cellular infrastructure. Through 2 of sensor readings, the Accelerometer and Gyroscope, they detected the transportation mode. Then they introduces the Location matching. SmartPark creates a location profile by leveraging the ubiquitous Wi-Fi and cellular infrastructure. This is achieved by mainly taking into account the proportion of APs in common between two location profiles and comparing the RSSI values of the two respective profiles. While in [51], the authors aim to build a mobile phone application that can detect the evacuation of parking spaces, while minimizing the impact on the user's mobile device. Parksense Leverages the ubiquitous Wi-Fi beacons in urban areas for sensing unparking events. It uses a novel approach based on the rate of change of Wi-Fi beacons to sense if the user has started driving. The authors motivates the use of Wi-Fi scanning (*Wireless based sensing)* by comparing it with *GPS based location* and *Network based location* and shows how these two approaches consume more power than wifi scanning and observing beacon rate of change. In [52], Frieson *et al*, discuss a complete data collection system deployed at the university of Manitoba that utilizes a variety of wireless networking technologies and devices to collection inferred traffic data at an intersection along a major thoroughfare in an urban setting. A wireless sensor network (XBee module, GSM module and Bluetooth module) of slave probes was designed and implemented with the objective to collect Bluetooth device information for this purpose. The aim is to collect statistical representation of traffic density and flow by scavenging data from consumer devices using Bluetooth communications. In [53], Vu *et al*, introduces a new frame work that collects location information and ad hoc contacts of humans at a university. The Bluetooth mac address is used to infer contact information and Wi-Fi mac addresses are used to infer physical location of the phone. Then they evaluated the performance of the hybrid epidemic data dissemination protocol on their collected dataset.

Leveraging RF technologies offers several strengths. These technologies enable low power consumption and efficient use of existing infrastructure, making them suitable for a wide range of applications. RF technologies can provide real-time data collection and analysis, enhancing the accuracy and timeliness of information. Furthermore, they facilitate seamless integration with various communication systems, such as Wi-Fi, Bluetooth, and cellular networks, allowing for versatile and scalable solutions. The ability to utilize multiple communication technologies also enables robust and resilient systems that can adapt to different environments and conditions.

2.3 Human Mobility in DTN Routing Protocols

Several approaches have studied human mobility patterns to improve routing in Delay Tolerant Networks (DTNs). These studies leverage the predictable aspects of human movement to enhance message delivery efficiency. Here are some key approaches:

2.3.1 Machine learning-based approaches

Recent work has explored using machine learning techniques to predict human mobility patterns and optimize routing decisions. These approaches leverage ML techniques to predict node behavior, optimize routing decisions, and improve overall network performance. Here are some key insights and references from recent research: In [54], the authors design an intelligent DTN routing system based on machine learning techniques. The system aims to classify and predict the best routing paths by leveraging historical data and node behavior patterns. But this approach requires memory storage to manage historical data and model parameters. In [55], the paper discusses using machine learning classifiers to predict the most suitable neighbor nodes for message delivery in DTNs. The approach involves analyzing network traffic statistics and historical delivery information to make informed routing decisions.

In conclusion, machine learning-based approaches offer significant potential to improve DTN routing protocols. By leveraging predictive analytics and adaptive algorithms, these approaches can enhance delivery ratios, reduce latency, and optimize resource usage in challenging network environments. However, there are also some weaknesses associated with these approaches. Machine learning models require substantial computational resources and memory storage to manage and process historical data and model parameters.Thus, while machine learning techniques offer promising improvements for DTN routing, further research is needed to address these challenges and enhance the applicability of ML-based approaches in diverse and resource-constrained DTN scenarios.

2.3.2 Analyzing Human Mobility

The analysis of human mobility is not new and has been widely exploited to optimize infrastructure deployment, especially for mobile telephony, mainly 5G [56] or edge computing. The literature has widely studied human mobility and investigated mobility models, which is a step to measure their impact on network performance [57]. To this end, recognition of physical mobility states and activities has been studied because they provide useful information for investigating in human mobility [6]. Several studies in the literature used GPS positioning to infer physical activities. In [58] and [59], the authors uses GPS data with external knowledge about bus routes and bus stops to infer and predict a user's transportation mode such as walking, driving or taking a bus by applying Bayes filters and Rao-Blackwellised particle. Although such algorithms perform well, they use computationally expensive models, thus it would be preferable to have simpler models that infer mobility states. Moreover, GPS data sampling is power-consuming. In [50], Krieg et al. proposed a transportation mode detection that helps in real-time parking, as through only 2 of sensor readings Accelerometer and Gyroscope, their system can decide whether a user is using one of the following transportation modes: walking, bicycle, bus, car, subway, motorbike, train, tram, airplane. Though this approach is applicable but it couldn't classify the real status of a device when it is static. Other approaches leveraged the information from LoRa, BLE and WiFi wireless links but rather for indoor and outdoor localization systems such as in [60], [61], and [62]. Thus our work is different in the sense that we adopt and exploit WiFi and BLE jointly to determine the status of the device in its real-life situation. To the best of our knowledge this is considered as a new approach for analyzing network context.

Other research work has explored the landscape of opportunistic communication and Device-to-Device (D2D) networks, emphasizing human-aware strategies. In [63], Waqas et al provide a comprehensive survey of mobility-aware D2D communications, while in [64] Costa et al. introduce TOOTS, a human-aware approach leveraging social and spatial factors for enhanced content delivery. Nunes et al. [65] combine spatial and social awareness in D2D opportunistic routing, and in [66], Thilakarathna et al. focus on friend-to-friend content dissemination in mobile environments. These studies highlight the potential of human-aware, socially-informed strategies in improving opportunistic and D2D communications. The main advantages of these approaches include improved content delivery

[63]: Muhammad Waqas, Yong Niu, Yong Li, Manzoor Ahmed, Depeng Jin, Sheng Chen, and Zhu Han. 'A comprehensive survey on mobility-aware D2D communications: Principles, practice and challenges'. In: *IEEE Communications Surveys & Tutorials* 22.3 (2019)

[64]: Rafael L Costa, Aline C Viana, Artur Ziviani, and Leobino N Sampaio. 'On building human-aware opportunistic communication strategies for cost-effective content delivery'. In: *Computer Communications* 192 (2022)

[65]: Ivan O Nunes, Clayson Celes, Igor Nunes, Pedro OS Vaz de Melo, and Antonio AF Loureiro. 'Combining spatial and social awareness in D2D opportunistic routing'. In: *IEEE Communications Magazine* 56.1 (2018)

[66]: Kanchana Thilakarathna, Aline Carneiro Viana, Aruna Seneviratne, and Henrik Petander. 'Design and analysis of an efficient friend-to-friend content dissemination system'. In: *IEEE Transactions on Mobile Computing* 16.3 (2016)

performance, more efficient routing, and better utilization of existing social structures. However, they also face challenges such as increased computational complexity, potential privacy concerns related to using social information, varying effectiveness depending on network environments, and difficulties in accurately predicting user mobility patterns. Despite these challenges, the overall trend suggests that incorporating human behavior and social factors into network designs can lead to significant improvements in opportunistic communication systems.

2.4 Summary

The evolution of DTN routing protocols demonstrates a clear trend toward incorporating a more sophisticated understanding of network dynamics, particularly the mobility of nodes. This progression underscores the essential role of examining human mobility patterns and node movement when designing effective DTN routing protocols. By focusing on mobility-focused approaches, researchers can develop more efficient, context-aware, and resource-optimized routing strategies for challenging DTN environments.

Reviewing the current state-of-the-art in delay-tolerant network (DTN) routing protocols, it is clear that the incorporation of node mobility studies has made a significant impact in the development of these protocols. The foundation for more advanced methodologies was established by early attempts, which were typified by probabilistic methods and basic flooding mechanisms. But the emphasis on social-based and context-aware routing that is present now emphasizes how important it is to understand node and human mobility. Routing decisions can be improved by predicatively leveraging trends and behaviours found in node mobility studies. Delivery success rates can be raised, overhead can be decreased, and resources may be allocated more effectively when node encounters and movements are anticipated.

Our approach distinguishes itself by bringing a new perspective to this ongoing evolution. By focusing on a deeper understanding of mobility patterns, our methodology aims to refine and extend current practices. We intend to develop a routing protocol that not only incorporates existing mobility insights but also innovates in the way these patterns are analyzed and applied. This novel approach promises to improve the efficiency, reliability and overall performance of routing in DTNs, offering a fresh and impactful contribution to the field.

PILOT Dataset 3

In this chapter, we introduce the PILOT dataset. First, we explain the reason behind collecting our own dataset, then illustrate all the steps needed for generating it. The dataset is composed of four joint information from wireless communication technologies and sensors. PILOT dataset provides a new generation of collected data that we believe will help provide keys to study human mobility or other applications.

3.1 Background and Motivation

These days, the rapidly evolving information technology and the development of wireless and mobile networks have promoted a new wave of information and industrial tide [67]. Several promising applications such as smartphone tracking, traffic monitoring, crowd dynamics monitoring, and other scientific research are based on the gathering and analysis of measurement data. This increases the urge to create new forms of datasets that provide a new perspective to analyze data and present new measurements. The collected datasets are mainly characterized by the *Model* and *Parameters* recorded. Each type of dataset can serve a new form of analysis, which is why there are always newly collected datasets with different characteristics. Thus, based on the requirements of our project, we need to validate the hypothesis illustrated in Section 1.3 to determine the real life situation of a device through wireless communication traces.

Our initial investigation focused on identifying existing datasets relevant to our research question. We conducted a thorough review of the literature and explored publicly available online repositories to check whether a publicly published dataset would serve to our contribution. However, we identified a lack of a collective dataset that includes several traces of wireless communication technologies recorded at the same time in different mobility contexts. Given the absence of a suitable existing dataset, we opted to collect our own data specifically tailored to our research objectives.

3.1.1 Online Datasets and literature review

Before generating our own dataset, we looked into the literature to check online datatsets if they can serve for our usecase. Thus, in this part, we provide two literature reviews, one dedicated to the different generated datasets for wireless communication traces in the literature and some presenting methodologies and tools for collecting them to highlight the uniqueness of our approach. The other subsection provides different use cases that used wireless traces in their research work to demonstrate the importance and use cases of such datasets.

Datasets: During the last decade, several research studies have been conducted to collect data efficiently from wireless networks and Internet of Things (IoT) devices for analysis [68]. In [52], Friesen et. al. present a

[68]: Louiza Mansour and Samira Moussaoui. ‘CDCP: Collaborative Data Collection Protocol in Vehicular Sensor Networks’. In: *Wireless Personnal Communications* 80.1 (2015). DOI: 10.1007/s11277-014-2000-z

[52]: Marc Friesen, Rory Jacob, Paul Grestoni, Tyler Mailey, Marcia R. Friesen, and Robert D. McLeod. ‘Vehicular Traffic Monitoring Using Bluetooth Scanning Over a Wireless Sensor Network’. In: *Canadian Journal of Electrical and Computer Engineering* 37.3 (2014)

complete data collection system developed at the University of Manitoba that uses a variety of wireless networking technologies and devices to collect inferred traffic data. They used XBee, GSM and Bluetooth modules for designing and implementing a slave probe network with the objective of collecting Bluetooth device information. In [53], Vu et. al. introduce a new framework that collects location information and ad hoc contacts of humans at the University of Illinois campus. The Bluetooth MAC address is used to infer contact information and Wi-Fi MAC addresses are used to infer physical location of the phone. In [69], a data collection campaign and a dataset of BLE beacons for detecting and analysing human social interactions is is collected in a High School. CRAWDAD [70] dataset, is a repository of wireless network datasets, including datasets for WiFi, Bluetooth, and cellular networks collected at Dartmouth. UJIIndoorLoc [71], is a dataset that includes WiFi signal strength measurements and location information for a multi-story building, which can be used for indoor positioning research. Although useful, these approaches are limited to contact tracing applications in a single environment or for indoor positioning systems.

[53]: Long Vu, Klara Nahrstedt, Samuel Retika, and Indranil Gupta. 'Joint Bluetooth/Wifi Scanning Framework for Characterizing and Leveraging People Movement in University Campus'. In: MSWIM '10. Bodrum, Turkey: Association for Computing Machinery, 2010. ISBN: 9781450302746. DOI: 10.1145/1868521.1868563. URL: https://doi.org/10.1145/1868521.1868563

[69]: Michele Girolami, Fabio Mavilia, and Franca Delmastro. 'A bluetooth low energy dataset for the analysis of social interactions with commercial devices'. In: *Data in Brief* 32 (2020). DOI: https://doi.org/10.1016/j.dib.2020.106102

[70]: https://ieee-dataport.org/collections/crawdad

[71]: https://www.kaggle.com/datasets/giantuji/UjiIndoorLoc

In [72], the authors designed a system architecture to collect data from the IoT environment relying on BLE technology only and smartphones. Cecchinel et. al. [73] proposed an architecture to support the big data collected from IoT. This architecture is restricted to data generated from sensors like sonar and temperature sensors and is presented from a software perspective only. While [74] generated different types of network traffic data with the FIT IoT-LAB testbed, their work includes a single technology (IEEE 802.15.4) and focuses on the delay and throughput of links under different message sizes and frequencies. Another technology under study is 5G. The authors of [75] produce 5G datasets that can be used to study malicious attacks in 5G traffic and their characteristics. In [76], the authors proposed a new paradigm for generating 5G datasets for ISAC high precision positioning, named Multilevel-FSM. In [77] the dataset is designed for multimodal machine learning research in wireless communication. It consists of various scenarios where multimodal sensing and communication data samples are collected. Although it is a large-scale rich dataset, we still offer more diversity. To the best of our knowledge, no previous work has collected a labeled dataset from multi-communication wireless technologies and sensors at the same time in different mobility contexts, as we propose with a new form of collected *parameters* in different *mobility* scenarios.

[72]: Abd Elwahab Boualouache, Omar Nouali, Samira Moussaoui, and Abdessamed Derder. 'A BLE-based data collection system for IoT'. In: *NTIC*. 2015

[73]: Cyril Cecchinel, Matthieu Jimenez, Sebastien Mosser, and Michel Riveill. 'An architecture to support the collection of big data in the internet of things'. In: *IEEE World congress on services*. 2014

[74]: Nina Santi, Rémy Grünblatt, Brandon Foubert, Aroosa Hameed, John Violos, Aris Leivadeas, and Nathalie Mitton. 'Automated and Reproducible Application Traces Generation for IoT Applications'. In: *Q2SWinet 2021*. Alicante, Spain

Dataset Usecases: Several approaches have leveraged from WiFi beacons or probe-response to characterize people's flow. As in [78], the author collected WiFi probe requests transmitted by people's smartphones, then used this data to characterize people's flows through a machine learning approach. While in [79], Gebru et. al. presented two possible methodologies for people counting and mobility detection based both on off-the-shelf hardware and commercial devices to scan the WiFi spectrum for probe requests in an urban environment, then by applying a ML-based scheme, they show how the data collected can be used to characterize people's flows. This approach is amenable to being deployed at the edge of the network. But as a further notice, Bluetooth has been exploited for some applications like traffic monitoring. As mentioned earlier, in [52], the authors exploited the collected data from Bluetooth devices to collect statistical representation of traffic density and flow. While in [80], Kulkarni et. al., deployed road-side Bluetooth scanners for traffic data collection and from Bluetooth information they have extracted traffic parameters for road traffic management. So, as we can see wireless communication has been pervasive in our daily lives. As such, collecting and analysing data from the wireless environment is an important approach for several use cases like traffic monitoring, characterizing people's flows which would help in studying human mobility, and for other services like positioning systems and others.

3.1.2 Purpose of the chapter

As illustrated in Section 3.1.1, in the rapidly evolving field of wireless communication and sensor technology, the collection and analysis of diverse data types is critical to advance our understanding and capabilities. This chapter details the comprehensive data collection methodologies employed in our research, focusing on a diverse array of data types, including WiFi prob-responses, Bluetooth Low Energy (BLE) advertisements, LoRa packets, and sensor-based measurements of acceleration, roll and pitch. This dataset meticulously collected to explore the ability to guess the real-life situation of a device through the variation of observed wireless links. In this chapter, we introduce a new approach to collect a dataset, characterized by mainly two novel approaches for collecting data, at the level of *Model* type and *Parameters* recorded, as follows:

- Collecting different types of data from sensors and wireless communication technologies at a time: WiFi probe response, BLE beacons, LoRa packets, and from the sensors: Acceleration, Roll, and Pitch information in a seamless and anonymous manner, involves using micro-controllers to gather data in real-time.
- Data are collected in different mobility scenarios and are classified mainly into two categories: *Static* vs *Mobile*.

We will outline the systematic approach taken in gathering and preparing the dataset that forms the backbone of this research. This chapter aims to:

- Ensure Replicability: Provide a detailed account of the methods and technologies used, enabling other researchers to replicate or build upon this work.
- Methodology: Justify the choices made during the data collection process, from the selection of data types to the configuration of collection parameters, illustrating how these decisions align with our research objectives.
- Challenges: Explain the challenges and limitations of the used setup on a hardware and software level.

3.2 Experimental Design

Based on the objective of the thesis, the goal is to identify the real-life situation through wireless traces; thus, the aim is to collect the possible information from the surrounding environment taking into consideration the limitations of the hardware devices used to collect the data set. For this reason, in this section we define the collected data and its functionalities.

To collect the dataset, we are mainly interested in obtaining wireless signal traces over short and long ranges. So, before introducing the scanning setup, we will first explain the different strategies for scanning, then detail the hardware used for collecting the data and the various wireless communication functionalities collected within the dataset. For the dataset intended for public release, we did not have limitations regarding the collected data. Our objective was to gather extensive data across various technologies.

PILOT dataset provides a group of collected packets from three different wireless communication technologies: WiFi, BLE and LoRa, and as additional useful information, the dataset includes the acceleration, roll, pitch and the battery voltage. These traces have been jointly collected all together in several mobility contexts using FiPy 3.1 microcontrollers from pycom [81]. The duration of each scanned dataset is between 10 min and 3 hours, and each recorded log file is labeled by its specific mobility scenario of scanning with a description of the conditions of scanning. The dataset is uploaded to Github* as a collection of text files and CSV files.

[81]: https://docs.pycom.io/datasheets/development/fipy/

3.2.1 Active and Passive Scanning

Active and passive scanning are two methods used to discover wireless networks (WiFi) and devices (Bluetooth), knowing that LoRa does not have the scanning option. Thus in this section, we will explain briefly the scanning properties for WiFi and Bluetooth. As **active scanning**, the device actively sends probe requests on all available channels. Access points (APs) respond with probe responses containing information about the network (SSID, supported rates, security features, etc.). This technique guarantees a faster discovery of networks since it does not wait for beacon frames. Same for Bluetooth, as the device actively sends inquiry requests to identify nearby Bluetooth devices, and nearby devices respond with their device addresses and, sometimes, additional information. While for **passive scanning**, in WiFi, the device passively listens for beacon frames that are periodically broadcast by APs and contain information about the network. Same for Bluetooth, the device listens for advertisement packets sent by Bluetooth devices. As a summary, **active scanning** involves actively sending requests and waiting for responses, leading to faster discovery but higher power consumption and increased network traffic. While **passive scanning** involves listening for broadcast signals, resulting in lower power consumption and minimal network interference but slower discovery. Since in our case, our aim was just to obtain the dataset rather than to build an optimized setup for scanning, we applied active scanning for WiFi and passive scanning for BLE (more details in Section 3.2.3.)

* https://github.com/Janakoteich/PILOT-Dataset-Collection-of-Multi-communication-Technologies

3.2.2 Data Collection Devices

The experimental setup described in this section is the basis for the collected data and the formation of the desired dataset. The aim is to scan the different wireless communication activities in the range of the scanning device. To do so, we used Pycom FiPy devices [81] since they support the three wireless communication technologies: WiFi, Bluetooth, and LoRa. FiPy also provides SigFox and LTE-CAT M1/NB1, but these are not included in our scanning process as they require a subscription to an operator. Focusing on WiFi, BLE and LoRa for the time being is sufficient for our main objective, eliminating the need for additional effort and investment to obtain LTE information.

[81]: `https://docs.pycom.io/datasheets/development/fipy/`

[82]: `https://docs.pycom.io/datasheets/expansionboards/pytrack2/`

[83]: `https://docs.pycom.io/datasheets/expansionboards/pysense2/`

The FiPy (Figure 3.1 is connected to a Pytrack [82] (Figure 3.5) or Pysense [83] (Figure 3.4) expansion board that includes Accelerometer and an SD card module. The FiPy development boards include an onboard WiFi and Bluetooth antenna, and for LoRa, an external antenna is connected.

Figure 3.1: FiPy microcontroller from Pycom

Since one FiPy supports several technologies at the same time, we first implemented the code to collect information for the three technologies on a single microcontroller. However, we observed a significant delay in receiving BLE beacons and WiFi probe response when one device was scanning and listening to all packets. So, to avoid the delay caused by interruption methods, the scanning process is distributed on four devices for more precision. To get the actual time, the FiPy is connected to an access point (AP), to be able to connect to the Network Time Protocol (NTP) server that will help to synchronize the real-time clock (RTC) and get the current timestamp. The collected data by each device is saved on a secure digital (SD) card. So, the implementation runs on four FiPy devices, each dedicated to collecting a separate type of data as follows:

- Node W (N_W): Scans for WiFi APs every 2s.
- Node B (N_B): Scans for Bluetooth devices every 1s.
- Node L (N_L): Listens for LoRa packets on different frequencies almost every 2s.
- Node X (N_X): Gets the acceleration, roll, pitch, and battery voltage/percentage.

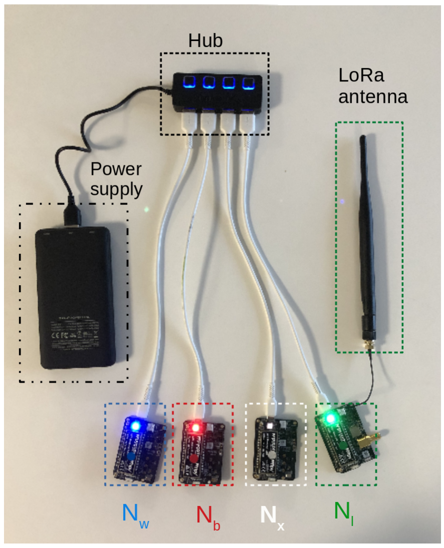


Figure 3.2: Final setup for scanning

Figure 3.2 shows the final setup for collecting the data, where the FiPys are connected to the power supply through a hub. However, to make the scanning process easier and more practical to hold the devices with the power supply in different places, the setup is designed in a box to be easily carried everywhere, as shown in Figure 3.3.

3.2.3 Configuration

The configuration of each wireless technology is as follows:

Figure 3.3: A simple design for encapsulating the whole setup

WiFi Node (N_W) **WiFi:** WiFi is a widely used wireless networking technology that enables devices to connect to the internet or communicate with each other over a local area network (LAN). It operates on the 2.4 GHz and 5 GHz frequency bands, offering high bandwidth capabilities, which makes it ideal for applications requiring fast data transfer. Key features of WiFi include:

Figure 3.4: External shield - Pysense

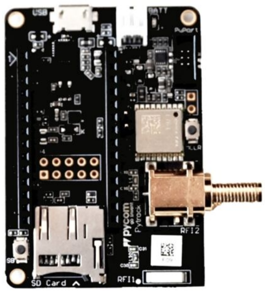


Figure 3.5: External shield - Pytrack

- **High Data Rates**: WiFi supports high-speed data transmission, suitable for streaming video, downloading large files, and online gaming.
- **Range**: Typically covers a range of up to 100 meters indoors and 300 meters outdoors, depending on the environment and obstacles.
- **Connectivity**: Allows multiple devices to connect simultaneously to a network, facilitating extensive device interaction and internet access.

The WiFi node is configured to start active scanning, as the device radio transmits a probe request and listens for a probe response from an AP or active devices such as phones or laptops. The FiPy device supports 802.11b/g/n so the radio scanning is in the 2.4-GHz to 2.4835-GHz spectrum. Upon detecting a probe response, scanners log several pieces of information related to that probe, as it indicates the following named tuples: (SSID, BSSID, sec, channel, RSSI). The saved log mainly includes 1) the timestamp related to the probe request detection, 2) the service set identifier (SSID) which is the name of the AP, 3) the basic service set identifier (BSSID) which is the MAC address of the AP, 4) the *sec* that stands for security, 5) the channel number, which is in the range of 1 to 11, and 6) the received signal strength (RSSI). The value of *sec* attribute defines the type of security where each value means the following: '0' is open, '1' is WEP, '2' is WPA-PSK, '3' is WPA2-PSK, and '4' is for WPA/WPA2-PSK. Figure 3.7d shows a sample of the saved logs of the received WiFi packets.

Bluetooth Node (N_B) The BLE node is configured for passive scanning to receive the advertising packets (PDUs) that are retrieved every second. First, we created a Bluetooth object, then started performing a scan listening for BLE devices sending advertisements. The following named tuple with the advertisement data is received during scanning: (mac, $addr\_type$, adv_type, $rssi$, $data$), where *data* contains the complete 31 bytes of the advertisement message. Then this *data* is parsed to get the following information: adv_flag, $scan\_tx\_pwr$, $conn\_tx\_pwr$, tx_range and adv_tx_pwr. So as with WiFi, the log is first saved with a timestamp since the advertisement is received, with the MAC address of the sender and the RSSI. Other information is retrieved like the advertising flag, where for some beacons it is unknown. Same for the $name$, in most of the received beacons the name of the device is unknown. Then we have the other information like scanning transmission power, connection transmission power, transmission range, and advertising transmission power. Figure 3.7b shows a sample of the saved logs for BLE beacons.

LoRa Node (N_L) The LoRa Alliance developed by Semtech [84] allows long-range, low-power and low-throughput communications. It operates on the **433-, 868- or 915-MHz** ISM bands, depending on the region in which it is deployed. So in our location, EU868 is supported. There are no beacons in LoRa, so we cannot scan to detect gateways as passive detection of a public LoRAWAN gateway is unreliable. Thus, we aimed to detect LoRa traffic by listening on different channels. The microcontroller will be listening to different frequencies and switch between them every second. When changing frequencies, the display will show what frequency it is

[84]: https://lora-alliance.org/

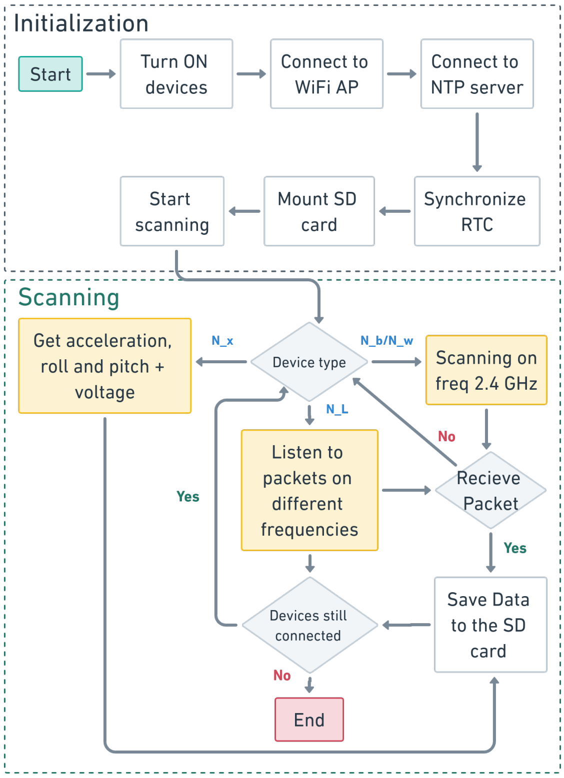


Figure 3.6: Data collection process diagram

monitoring and the number of valid packets that have been previously seen on that frequency. In this way, we can receive some LoRa packets that are operating on the following frequencies: 863000000, 864000000, 865000000, 866000000, 867000000, 868000000, 869000000, 864862500, 865062500, 865402500, 865602500, 865985000, 866200000, 866400000, 866600000. These frequencies are defined randomly in the configuration of the FiPy. As we increase the number of frequencies to listen on, the delay will increase for switching between selected frequencies, and as a consequence, the chance of receiving packets will decrease. So, if by chance we received data while listening on a specific frequency, the data will be saved in the file with mainly the following information: The timestamp, the spreading factor, the data itself (which will be masked for privacy issues), the frequency, RSSI, the signal-to-noise ratio (SNR), and other information as shown in the log sample in Figure 3.7c.

```
2022-06-09 17:43:02: {'Acceleration': '(0.2675781, -0.1103516, 0.9726563)', 'Roll': '-15.3816', 'battery_voltage': 4.556237, 'battery_percentage': 88.95696, 'Pitch': '6.242762'}
2022-06-09 17:43:03: {'Acceleration': '(0.2646484, -0.1101074, 0.9729004)', 'Roll': '-15.2174', 'battery_voltage': 4.556237, 'battery_percentage': 88.95696, 'Pitch': '6.232354'}
2022-06-09 17:43:03: {'Acceleration': '(0.2667236, -0.1104736, 0.9689941)', 'Roll': '-15.39001', 'battery_voltage': 4.561255, 'battery_percentage': 89.45872, 'Pitch': '6.23429'}
2022-06-09 17:43:04: {'Acceleration': '(0.2658691, -0.1096191, 0.9736328)', 'Roll': '-15.44312', 'battery_voltage': 4.566273, 'battery_percentage': 89.96055, 'Pitch': '6.285924'}
2022-06-09 17:43:05: {'Acceleration': '(0.2670898, -0.1098633, 0.9730225)', 'Roll': '-15.44367', 'battery_voltage': 4.556237, 'battery_percentage': 88.95696, 'Pitch': '6.236812'}
2022-06-09 17:43:06: {'Acceleration': '(0.2672119, -0.1097412, 0.9681396)', 'Roll': '-15.4298', 'battery_voltage': 4.556237, 'battery_percentage': 88.95696, 'Pitch': '6.260544'}
```


(a) Log file of acceleration data


```
2022-06-09 17:43:02: {'adv_flag': None, 'def_tx_pwr': 3, 'mac': b'26...8d', 'rssi': -75, 'name': None, 'scan_tx_pwr': 3, 'conn_tx_pwr': 64, 'tx_range': None, 'adv_tx_pwr': 3}
2022-06-09 17:43:03: {'adv_flag': None, 'def_tx_pwr': 3, 'mac': b'26...8d', 'rssi': -81, 'name': None, 'scan_tx_pwr': 3, 'conn_tx_pwr': 64, 'tx_range': None, 'adv_tx_pwr': 3}
2022-06-09 17:43:04: {'adv_flag': None, 'def_tx_pwr': 3, 'mac': b'26....8d', 'rssi': -85, 'name': None, 'scan_tx_pwr': 3, 'conn_tx_pwr': 64, 'tx_range': None, 'adv_tx_pwr': 3}
2022-06-09 17:43:05: {'adv_flag': None, 'def_tx_pwr': 3, 'mac': b'26...8d', 'rssi': -84, 'name': None, 'scan_tx_pwr': 3, 'conn_tx_pwr': 64, 'tx_range': None, 'adv_tx_pwr': 3}
2022-06-09 17:43:05: {'adv_flag': None, 'def_tx_pwr': 3, 'mac': b'267...8d', 'rssi': -72, 'name': None, 'scan_tx_pwr': 3, 'conn_tx_pwr': 64, 'tx_range': None, 'adv_tx_pwr': 3}
2022-06-09 17:43:06: {'adv_flag': None, 'def_tx_pwr': 3, 'mac': b'267...8d', 'rssi': -69, 'name': None, 'scan_tx_pwr': 3, 'conn_tx_pwr': 64, 'tx_range': None, 'adv_tx_pwr': 3}
```


(b) Log file of BLE data


```
2022-06-09 17:43:02:
    {'spreading_factor': 7, 'data': b'\xa4\xcdq\...\xf1\x9c', 'frequency': 865062500, 'bandwidth': 0}
    {'rx_timestamp':2901527116, 'rssi':-122, 'snr':-11.0, 'sfrx':7, 'sftx':0, 'tx_trials':0, 'tx_power':14, 'tx_time_on_air':0, 'tx_counter':0, 'tx_frequency':0}
2022-06-09 17:43:04:
    {'spreading_factor': 7, 'data': b'9..7w', 'frequency': 865402500, 'bandwidth': 0}
    {'rx_timestamp':2901527116, 'rssi':-122, 'snr':-11.0, 'sfrx':7, 'sftx':0, 'tx_trials':0, 'tx_power':14, 'tx_time_on_air':0, 'tx_counter':0, 'tx_frequency':0}
2022-06-09 17:43:06:
    {'spreading_factor': 7, 'data': b'', 'frequency': 865602500, 'bandwidth': 0}
    {'rx_timestamp':2901527116, 'rssi':-122, 'snr':-11.0, 'sfrx':7, 'sftx':0, 'tx_trials':0, 'tx_power':14, 'tx_time_on_air':0, 'tx_counter':0, 'tx_frequency':0}
```


(c) Log file of LoRa data


```
2022-06-09 17:43:02: {'ssid':'edm', 'bssid':b'<Q\...\xa0', 'sec':5, 'channel':1, 'rssi':-66}
2022-06-09 17:43:02: {'ssid':'guest', 'bssid':b'<Q\...\xa2', 'sec':0, 'channel':1, 'rssi':-66}
2022-06-09 17:43:04: {'ssid':'IA', 'bssid':b'<Q\...\x84#', 'sec':0, 'channel':6, 'rssi':-83}
2022-06-09 17:43:04: {'ssid':'IA-intr', 'bssid':b'<Q\...\x84$', 'sec':5, 'channel':6, 'rssi':-83}
2022-06-09 17:43:06: {'ssid':'IA-guest', 'bssid':b'<Q\...\xa2', 'sec':0, 'channel':1, 'rssi':-64}
2022-06-09 17:43:06: {'ssid':'edm', 'bssid':b'<Q\...\xa0', 'sec':5, 'channel':1, 'rssi':-65}
```


(d) Log file of WiFi data

Figure 3.7: Examples of the log files of the collected dataset for each technology

3.3 Dataset Collection Methodology

3.3.1 Study Area and Different Scenarios

The dataset is collected in different scenarios, with different variations. Using the final setup (Figure 3.2) described in Section 3.2, the devices are connected to the power supply to start collecting data. Each log is saved with a real timestamp in a text file on the SD card. The goal is to observe and record the variations of wireless technologies in different mobility contexts, which are mainly categorized into two: *Static* and *Mobile* scenarios. For *Static* we defined the following cases: Home, Office, Restaurant, Bus station, University and Meetings, and for *Mobile* we have the following scenarios: Pedestrian, car, bus, metro and trains. We took into consideration the different conditions of each scenario, as we have rural areas (autoroute), urban areas like cities, and less crowded places like villages. Some of the scenarios that fall under the category of having several conditions are illustrated below.

Train: Knowing that there are several types of trains, the data is collected in almost all different types that exist, like fast trains that travel between provinces more likely between countrysides (eg., TGV in France), or slower trains that travel between cities (like TER and Intercités in France), or those that travel between areas in a big city (eg., RER in France).

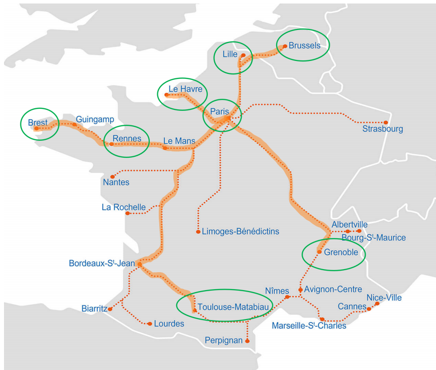


Figure 3.8: Map of the collected data places

Bus: Mainly there are two kinds of buses. 1) Those that travel inside cities (urban areas) with a specific trajectory and fixed interruptions on bus stops every specified short time, and 2) buses that travel between cities (rural areas) for a long time and distance (the Autocar).

Car and Pedestrian: For Cars or pedestrians, there are mainly two conditions, being in an urban or rural place.

Home: There are several types of homes, such as ordinary apartments in rural areas, studio apartments in urban areas, collocations, platforms in a crowded city, a platform in a village, separate homes in rural areas, a Hotel, etc.

Meetings: In this category, we defined the static scenarios where people meet for professional activities such as conferences, workshops, seminars, etc.

Office: The office can be in a building surrounded by other companies and buildings. The office scenario can be categorized into a rural office and an urban office.

Figure 3.8 shows a map of the main train lines in France. The highlighted orange routes show the train lines where the data was collected. The cities marked with a green circle represent the main locations of the data collected in the other scenarios (bus, home, hotel, university, etc.).

3.3.2 Privacy Preserving Technique

According to the General Data Protection Regulation (GDPR) [85], the MAC address and the device name are considered as personal data, so to ensure the published dataset is privacy-preserving, we pseudonymize the device names which is a foundational technique to mitigate data

[85]: `https://ec.europa.eu/info/law/law-topic/data-protection/reform/`

protection risks. This is achieved by replacing the actual name of each unique MAC address with any random symbol and the MAC address is hashed using SHA-224 [86].

[86]: https://www.rfc-editor.org/rfc/rfc3874.txt

3.3.3 Dataset Description

One of the main characteristics of the dataset is that it is labeled. This is achieved by noting the places and the time of scanning; then the data is retrieved from the four devices and saved in a file with a label based on the scanning conditions that were noted. In the first step, we get four text files (wifi.txt, ble.txt, Lora.txt, acc.txt), then each text file is transferred to a comma-separated values (CSV) file. The data is classified and uploaded on GitHub. Tables 3.1 and 3.2 briefly describe the organization of the dataset for mobile and static collected scenarios respectively. For each scenario, we have sub-folders named by the prefix of the name of this scenario, and each of them includes the scanned data. The table describes each scanned folder by its category, the collected information, the approximate scanning interval with the duration, and finally the description of the scanning scenario.

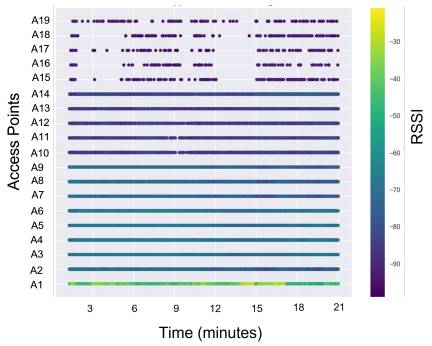


Figure 3.9: WiFi data from an office

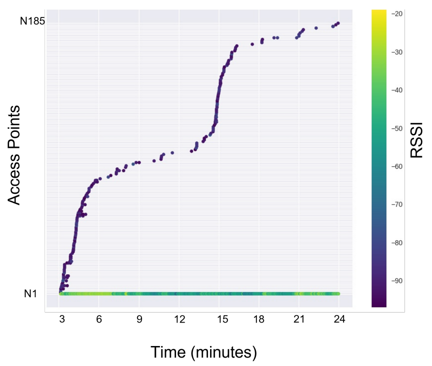


Figure 3.10: WiFi data from a bus

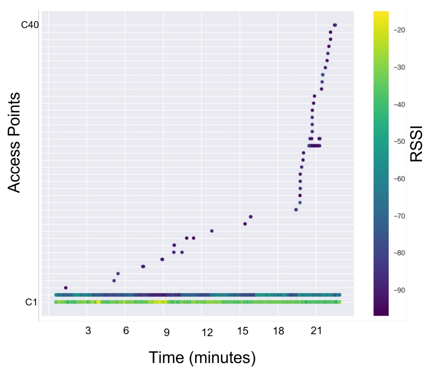


Figure 3.11: WiFi data from a car

Table 3.1: Records from Mobile Scenarios

Mobility Scenarios	Mobile								
	Label	*WiFi*	*BlE*	*LoRa*	*Acceleration*	*From*	*To*	*Duration*	*Description*
Bus	B1	✓	✓	✓	✓	12:20:00	12:55:00	35min	Autocar between city and village - crowded
	B2	✓	✓	✓	✓	13:42:00	14:25:00	43min	Autocar between village and city - crowded
	B3	✓	✓	✗	✓	17:07:04	17:56:00	49min	Bus in a city
	B4	✓	✓	✗	✓	14:23:00	15:00:00	37min	Bus in a city
	B5	✓	✗	✓	✓	20:06:00	20:38:00	32min	Bus in a city - very crowded
	B6	✓	✓	✓	✓	09:13:00	09:22:00	9min	Bus in a city
	B7	✓	✓	✗	✓	18:45:00	18:58:00	13min	Bus in a city
	B8	✓	✓	✓	✓	17:01:00	17:12:00	11min	Bus in the city - crowded
Car	C1	✓	✓	✓	✓	17:26:00	18:03:00	37min	Auto-Route - rural area
	C2	✓	✓	✗	✓	13:21:00	14:04:00	43min	Auto-Route then between houses in villages
	C3	✓	✓	✗	✓	09:06:00	09:24:00	18min	Auto-Route then between houses in villages
Metro	M1	✓	✓	✓	✗	21:04:00	21:35:00	31min	Short metro in a city
	M2	✓	✓	✗	✓	11:00:00	11:22:00	22min	Short metro in a city
	M3	✓	✓	✓	✓	21:23:00	21:33:00	10min	Long metro in a city
	M4	✓	✗	✓	✓	13:32:00	13:48:00	16min	Long metro in a city
	M5	✓	✗	✓	✓	16:39:00	16:52:00	13min	Short metro in a city
	M6	✓	✓	✓	✗	15:07:00	15:20:00	13min	Short metro in a city
Train	T1	✓	✓	✓	✓	16:21:00	18:17:00	1hr, 56min	Intercités - crowded
	T2	✓	✓	✓	✓	08:04:00	08:30:00	26min	RER - crowded
	T3	✓	✓	✓	✗	20:17:00	21:02:00	45min	TER between two cities
	T4	✓	✓	✓	✓	09:35:00	10:23:00	48min	TER between two cities
	T5	✓	✓	✗	✓	11:46:00	12:37:00	51min	TER between two cities
	T6	✓	✓	✓	✓	10:15:00	13:12:00	2hr, 57min	TGV
	T7	✓	✓	✓	✓	11:47:00	12:57:00	1hr, 10min	TGV
	T8	✓	✓	✓	✓	15:17:00	16:18:00	1hr, 1min	TGV
	T9	✓	✓	✗	✓	07:30:00	08:51:00	1hr, 21min	TGV
	T10	✓	✓	✓	✓	21:49:00	23:05:00	1hr, 16min	TGV
	T11	✓	✓	✓	✓	14:06:00	14:45:00	39min	TGV
	T12	✓	✓	✓	✓	14:53:00	15:26:00	33min	TGV
	T13	✓	✓	✓	✓	18:27:00	19:07:00	40min	TGV
	T14	✓	✗	✓	✓	17:15:00	18:05:00	50min	TGV
	T15	✓	✗	✓	✗	19:14:00	19:56:00	42min	TER between two cities
	T16	✓	✗	✓	✓	18:47:00	19:43:00	56min	TGV
Pedestrian	P1	✓	✓	✓	✓	09:23:00	09:34:00	11min	University Campus
	P2	✓	✓	✗	✓	18:58:00	19:07:00	9min	Crowded city
	P3	✓	✓	✓	✗	19:56:00	21:00:00	1hour, 4min	Downtown - crowded
	P4	✓	✓	✓	✓	12:55:00	13:08:00	13min	Rural area - Countryside
	P5	✓	✓	✓	✓	13:03:00	13:13:00	10min	Rural area - Countryside
	P6	✓	✓	✓	✗	10:27:00	12:06:00	1hr, 39min	Mall in a city
	P7	✓	✓	✓	✓	21:41:00	22:16:00	35min	Rural area - next to an auto-route
	P8	✓	✓	✗	✓	15:00:00	16:17:00	1hr, 17min	Mall in a city
	P9	✓	✓	✓	✗	17:12:00	17:28:00	16min	No Information
	P10	✓	✓	✓	✗	07:57:00	08:04:00	7min	University campus

Table 3.2: Records from Static Scenarios

Static Scenarios	**Static**								
	Label	***WiFi***	***BlE***	***LoRa***	***Acceleration***	***From***	***To***	***Duration***	***Description***
Home	H1	✓	✓	✓	✓	01:14:00	02:36:00	1hr, 22min	Student residence
	H2	✓	✓	✓	✗	12:31:00	13:33:00	1hr, 2min	Studio in a crowded city
	H3	✓	✓	✓	✓	08:08:00	08:43:00	35min	Apartment in a building - village
	H4	✓	✓	✓	✓	08:38:00	10:42:00	2hr, 4min	Apartment in a building - village
	H5	✓	✓	✓	✓	10:06:00	11:37:00	1hr, 31min	Student residence in a city
	H6	✓	✓	✓	✓	19:08:00	19:35:00	27min	Hotel in a crowded city
	H7	✓	✓	✓	✓	06:08:00	07:05:00	57min	Hotel in a crowded city - urban area
	H8	✓	✓	✓	✓	10:50:00	12:13:00	1hr, 23min	Hotel in a village - rural area
	H9	✓	✓	✓	✓	08:17:00	08:42:00	25min	Hotel in a village - rural area
	H10	✓	✗	✓	✓	10:38:00	12:16:00	1hr, 38min	Apartment in a building - village
University	U1	✓	✓	✓	✗	09:14:00	11:12:00	1hr, 58min	University campus
Office	O1	✓	✓	✓	✓	09:54:00	11:00:00	1hr, 6min	Office in a rural area
	O2	✓	✓	✓	✓	16:55:00	18:07:00	1hr, 12min	Office in a rural area
	O3	✓	✗	✓	✓	17:38:00	19:16:00	1hr, 38min	Office in a rural area
	O4	✓	✗	✓	✓	09:38:00	10:35:00	57min	Office in a rural area
	O5	✓	✗	✓	✓	16:31:00	17:54:00	1hr, 23min	Office in a rural area
	O6	✓	✗	✓	✓	10:12:00	11:18:00	1hr, 6min	Office in a rural area
	O7	✓	✗	✓	✓	16:17:00	16:47:00	30min	Office in a rural area
	O8	✓	✓	✓	✗	17:02:00	18:13:00	1hr, 11min	Office in an urban area
Bus_station	BS1	✓	✓	✓	✓	13:13:00	13:42:00	29min	No Information
	BS2	✓	✓	✗	✓	14:16:00	14:23:00	7min	No Information
	BS3	✓	✓	✓	✓	16:47:00	17:00:00	13min	In a city, not crowded
Conference	C1	✓	✓	✓	✓	10:37:00	11:04:00	27min	Conference at rural area
	C2	✓	✓	✓	✓	10:01:00	11:04:00	1hr, 3min	Conference at a university campus
	C3	✓	✓	✓	✓	16:04:00	17:07:00	1hr, 3min	Conference at a university campus
	C4	✓	✓	✓	✓	08:58:00	10:14:00	1hr, 16min	Conference at a university campus
	C5	✓	✓	✓	✓	13:58:00	14:04:00	6min	Conference at a rural area in a village
	C6	✓	✓	✓	✓	09:09:00	10:12:00	1hr, 3min	Conference at a rural area in a village
Restaurant	R1	✓	✓	✓	✓	10:49:00	12:02:00	1hr, 13min	Restaurant in the city, not crowded.

3.4 Dataset Insights

In this section, some primary observations of the collected data are presented. Figures 3.14, 3.10 represent the received frames from each access point over time, with the corresponding RSSI. In Figure 3.14, which shows the WiFi data collected from an office, we can observe the reception of the probe response frames throughout the scanning time. Knowing that the scanner is fixed would indicate that the access points detected by the scanner are also fixed. While in Figure 3.10 which is related to the data collected from a bus in an urban area, the frames appear only for a very short duration, this is because we are losing connection to the access point due to the mobility of the bus. From these observations, we can verify whether each scenario has its unique pattern of received probe response frames which would confirm our intuition as described in Section 1.3. What is more interesting is that several datasets for the same scenario, show the same behavior of received frames. Such results could be important for the building of new knowledge and bringing a new perspective to the analysis of human mobility.

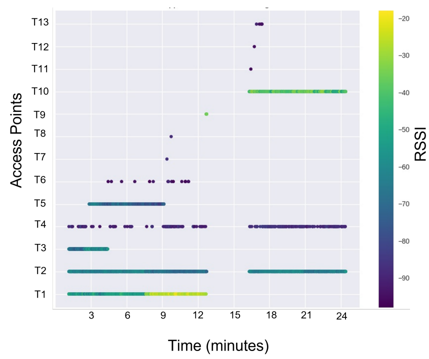


Figure 3.12: WiFi data from a train

As a further analysis, normally we should keep receiving a probe response frame from an AP as long as the AP is in the range of the node that is sending the probe request, but based on Figure 3.9, we observed the absence of the frames for some time-slots even when the access point is near to the scanner and static. To investigate this issue, we had to check the configuration for scanning. Based on the code implemented, the dwell time is set to *20 ms*, as this field is used to control how long the scan dwells on each channel. So, the node listens for 20 ms on each channel for frames and then switches to the other. To see the impact of dwell time on scanning quality, we performed a simple experiment (Figure 3.15) as described below. First we selected two nodes as follows:

[87]: Taehwa Choi, Yohan Chon, and Hojung Cha. 'Energy-efficient WiFi scanning for localization'. In: *Pervasive and Mobile Computing* 37 (2017). DOI: https://doi.org/10.1016/j.pmcj.2016.07.005

- **Node S:** Configured with 20ms dwell time on each channel.
- **Node L:** Configured with 120ms dwell time on each channel.

Both nodes are located next to each other, and started scanning at the same time with the same conditions for 15 minutes. Figures 3.14 and 3.13 show the results of the data collected from *Node S* and *Node L* respectively. From the figures we have the following observations:

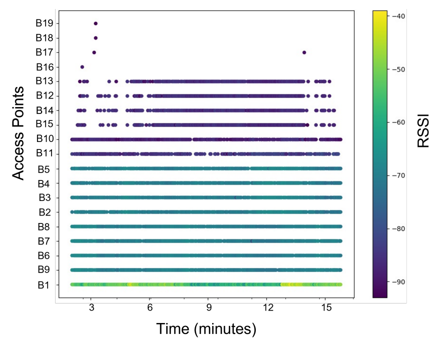


Figure 3.13: Results from node L (dwell time 120ms)

- *Node S* discovered 15 access points, while *Node L* discovered 19 access points.
- *Node L* received more beacons than *Node S* from the access point **B12**.

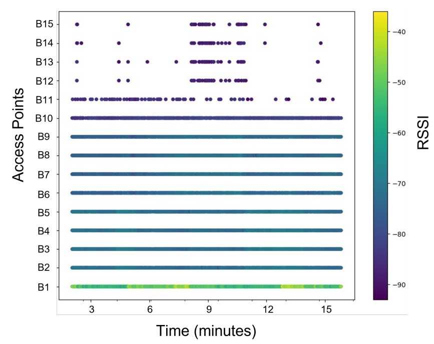


Figure 3.14: Results from node S (dwell time 20ms)

These observations show the impact of the dwell time on the number of received probe response frames. From these results we can deduce that to detect more access points and receive more probe response frames, it is suggested to increase the dwell time on a channel to increase the chance of discovering more access points. This observation is supported by the work done in [87] with further analysis.

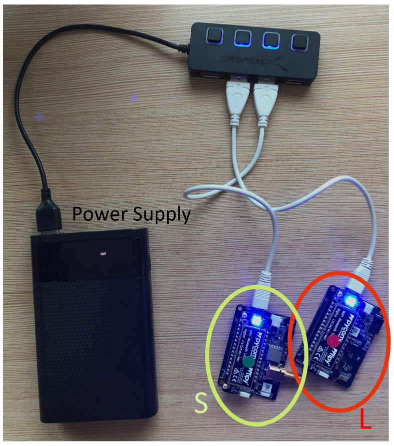


Figure 3.15: Dwell time experimental setup

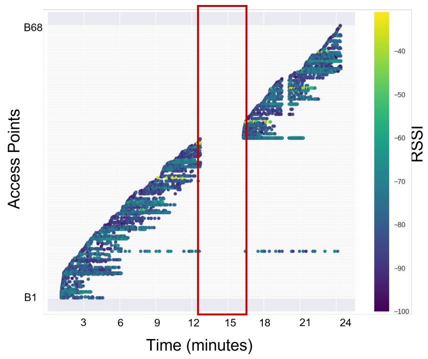


Figure 3.16: Gap due to hardware disconnection

3.5 Challenges and Open Issues

3.5.1 Hardware Level

Based on our experience, the primary challenges in collecting the dataset are hardware related. During scanning, the FiPy device can occasionally disconnect or reboot, resulting in gaps in the collected data. Figure 3.16 shows an example of a dataset where the FiPy is disconnected for awhile then continued scanning. Another main challenge is time synchronization. The FiPy lacks an internal Real-Time Clock (RTC) module for time synchronization, necessitating a connection to an NTP server via WiFi. However, this method is not always feasible due to radio coverage limitations or connection instability. Lastly, the SD card can sometimes become corrupted, leading to loss of data.

3.5.2 Software Level

Besides the hardware challenges, there is another open issue to be investigated to improve the quality of scanning which concerns LoRa. There are no beacons in LoRa, so we cannot scan to detect gateways as passive detection of a public LoRAWAN gateway is unreliable. A FiPy device can only listen on one channel and DataRates (DR) at a time; thus, we can only listen to a specific channel/data rate combination and see if by chance someone else uses the same. So, we can deduce that the radio can only monitor one frequency at a time, and we must receive an entire LoRa packet while listening to parse it, but the more time we spend on one channel, the more we might be missing from others. So, in our approach, we are still limited by the selected frequencies mentioned in Section 3.2 for listening to LoRa packets, while the goal is to capture as many as possible packets and discover the presence of LoRa base stations. Finally, knowing that the methodology for scanning and getting all four joined pieces of information at the same time was challenging, further investigations could be held to optimize the scanning setup while still getting the same quality of the output dataset.

3.6 Conclusion and Integration with Thesis

In this Chapter, we have explained the purpose behind collecting PILOT dataset that includes four joint collected pieces of information from multi-communication wireless technologies and sensors. The overall collected data till now spans about 90 hours in total in different mobility scenarios collected using a Micropython enabled microcontroller called FiPy device. The traces are collected in different mobility scenarios mainly categorized as *Static* and *Mobile* scenarios. The dataset is a collection of WiFi probe-responses, BLE beacons, LoRa packets, and additional sensor information such as acceleration, roll, and pitch. The methodology to reproduce the dataset is illustrated, with annotations to the collected data provided on GitHub. After collecting the desired dataset, we can now investigate in the data collected to see whether it is possible to exhibit mobility related patterns and so determine the real-life situation of a device based on the wireless dynamics observed in its range. Further details are explained in the next chapter.

Machine Learning Approach 4

In this chapter, we shift our focus from data collection to application, specifically through the development of a machine learning model. We will detail how we use the dataset described in Chapter 3 to construct a model that validates our initial hypothesis: that it is possible to accurately infer the real-life situation of a device based on wireless traces alone. By integrating various machine learning techniques and methodologies, we will demonstrate the potential of our dataset to not only support, but also improve our understanding of device behaviour in different environments.

4.1 Objective and Problem Definition

In this chapter we explain the steps to investigate in the collected wireless traces to study whether we can identify specific patterns that would allow us to determine the real-life situation of a device as explained in Section 1.3. First we visualized the data to observe and select the important features to build a machine learning model designed to classify mobility contexts using WiFi and Bluetooth Low Energy (BLE) packets. The model aims to achieve high accuracy in distinguishing between stationary and mobile states and further identifying the specific context (or 'real-life' situation) of mobile states.

The primary objective is to design a machine learning model capable of accurately classifying the mobility context of a device based on data collected from WiFi and Bluetooth Low Energy (BLE) beacons. The model aims to perform two main tasks:

- Binary Classification (B-model): Determine whether the device is stationary or mobile.
- Multiclass Classification (M-model): Identify the specific context of the device (e.g., Home, Office, Bus, Train) when it is mobile or static.

4.2 Model Design

In this section, we delve into the details of the models. First, we provide an in-depth overview of the dataset, highlighting its structure and the specific data points used for the analysis. Subsequently, we explore the feature selection process, where we identify and refine the variables that are most critical to achieve accurate model performance. Lastly, we outline the architecture of our machine learning model and discuss the reasoning behind the selection of specific algorithms.

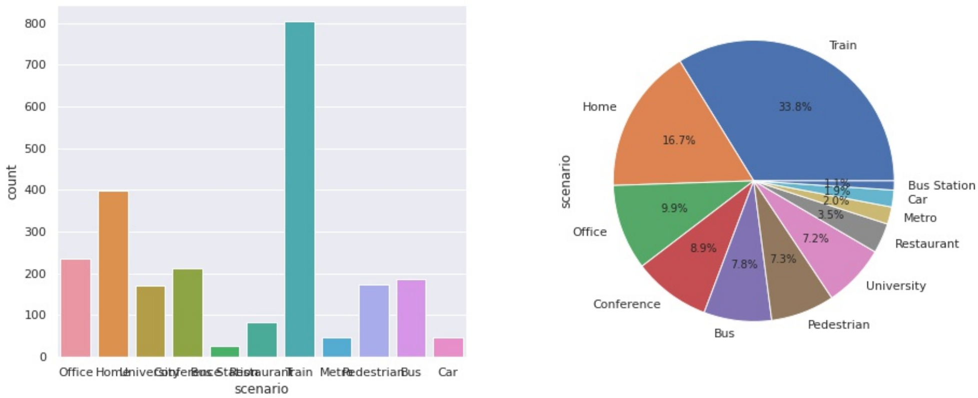


Figure 4.1: Dataset distribution

4.2.1 Data Description

We use the dataset collected and described in Chapter 3. The distribution of the dataset is shown in Figure 4.1. This distribution represents the number of input samples of each label ready for training. After checking the scanned information of each technology, we found that the following parameters are essential to exhibit pattern to design a machine learning-based approach able to determine the device's real-life situation:

- MAC address, which provides insights into the density of the observed networks.
- Signal strength (RSSI), which can indicate the mobility of devices or access points; this will be explained in detail in the upcoming section.
- Timestamp, as it is crucial for determining the duration that a specific network remains within the scanner's range.

4.2.2 Feature Selection

Features are derived from the detected WiFi APs and BLE packets, focusing on:

- Contact duration with the scanner.
- Received Signal Strength Indication (RSSI).
- Number of unique MAC addresses detected.

Data Pre-processing and Feature Engineering

As illustrated in Chapter 3.2, the datasets are saved as csv files. Since each scenario (label) has two separately scanned files (WiFi and BLE) that share the same time-stamp, both files are combined to facilitate the analysis and observation of both datasets at the same time. Now, for each scenario, we have one file that includes the data from WiFi and BLE at a certain time-slot. Each dataset is in the form of an n x 3 matrix, where $n > 0$ is a variable number that equals to the number of received probe-responses and advertisements from all detected APs during scanning time t. But feeding data into a model must be a column matrix and not an n x m matrix. So, we need to transform the matrix n x 3 into a 1 x $(f + 1)$ where f is the number of selected features of both WiFi and BLE (to be defined in the next section) plus at the end the *label*. Thus, in this case, each dataset will be represented by a single row. To transform the shape of the dataset from 2D to 1D, the dataset undergoes two main phases:

- ϕ_1: Get the main statistics for each AP, thus as a result we will get an m x f_1 matrix, where m is the number of unique AP that appeared during scanning, and f_1 is the number of extracted features (defined in section 4.2.4). So, the dataset still has the 2D form at this phase.
- ϕ_2: Transform each 2D dataset from ϕ_2 to a 1D vector (explained in section 4.2.5).

4.2.3 WiFi and BLE Selected Features

During scanning, the device frequently receives beacons from APs that are in its communication range. If the scanner moves away from the AP, after some time the connection between the two devices is lost, and the scanner stops receiving beacons from that AP, and vice versa if the AP is mobile and the scanner is fixed. This behavior is therefore considered an important metric for identifying the mobility of a device (as explained in Chapter 3).

From these observations, we can see how each scenario has its unique pattern of received beacons and the importance of the contact duration between the scanning device and surrounding APs in differentiating between different scenarios. Thus, as a result, the time-stamp, the RSSI and MAC address will be selected for feature engineering, since they are the main attributes to give insights for the collected datasets.

4.2.4 Feature Extraction: Phase 1

In this section, data processing and feature extraction are illustrated in details. As mentioned in Section 4.2.3, contact duration will be calculated for each AP, with the mean and standard deviation of the RSSI. As a result, we will end up with two dataframes that represent the statistics of the data collected for each wireless technology.

1. Contact Duration: Let M be the set of all unique MAC addresses appeared during Time t of scanning. The contact duration is calculated as follows:

$$duration(m) = t_i - t_0, \quad \forall m \in M \tag{4.1}$$

Where i is the last beacon appeared during the scanning.

2. Signal Strength (RSSI) Mean and Standard Deviation: The RSSI value tends to fluctuate even if the device is fixed because of external factors influencing radio waves (interference, diffraction, etc.), or when a Wi-Fi receiver is moving, the signal strengths it observes are noisier than when it is not moving. For this reason, for each AP, the average mean of the values recorded over time is calculated as follows:

$$\bar{X}_{x(m)} = \frac{1}{n}\sum_{i=1}^{n} x_i(m), \quad \forall m \in M \tag{4.2}$$

where $x(m)$ represents the RSSI from the MAC address m, x_i is the $i-th$ RSSI value in the sample, n is the total number of appearances of the beacon from the AP of the same MAC address during the scanning time, and x_i is the $i-th$ RSSI value in the sample. Then the standard deviation is calculated (eq. 4.3) to see how dispersed the data is in relation to the mean as this could indicate if the device is moving.

$$\delta_{x(m)} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{X}_{x(m)})^2}{n-1}} \tag{4.3}$$

4.2.5 Feature Extraction: Phase 2

Now as a result we have two data-frames that summarise the main information for each AP. Still we need to transform both 2D files to a 1D vector. This is the second phase (ϕ_2) to obtain the important information from the dataframe for our model. Table 4.1 shows the number of AP that appeared in different scenarios. We can see how the number of APs differs from one context to another as well as the average contact duration (Δ_t) with the scanner. Knowing that the contact duration is an important metric for differentiating between scenarios, as mentioned in Section 4.2.3, then the dataframes will be transformed to a 1D vector by displaying the access point (AP) statistics based on their contact duration (Δ_t). The features are mainly categorized into three main conditions: Long Δ_t, medium Δ_t, and short Δ_t according to the following criteria: First, get the percentage of contact duration as follows:

$$\%\Delta_t(m) = \frac{\Delta_t(m)}{t} \times 100 \quad \forall m \in M \tag{4.4}$$

1. L: Set of APs that have $\Delta_t > 70\%$ of the total time t.
2. M: Set of APs for which $30\% < \Delta_t < 70\%$.
3. S: Set of APs that have a $\Delta_t < 30\%$ of the total time t.

Then for each Set (L, M, S), the mean and standard deviation of the RSSI of the access points that belong to each set is calculated.

As a result, we end up with a vector (V_i) that includes 24 features (12 features extracted from WiFi and 12 extracted from BLE), plus the label of the scanned scenario i. The same process is done for all the datasets that are collected in different scenarios, thus we end up with a dataframe of V_n input vectors, where n is the total number of datasets (see Figure 4.2).

Table 4.1: Records from dataset analysis

	Number of Wifi MAC	Wifi Δ_t	Number of BLE MAC	BLE Δ_t
Office	19	9min, 16sec	27	4min, 27sec
Bus	185	17sec	43	2min, 32sec
Car	6	4min	6	3min, 56sec
Train	3	6min, 9sec	68	3min, 13sec

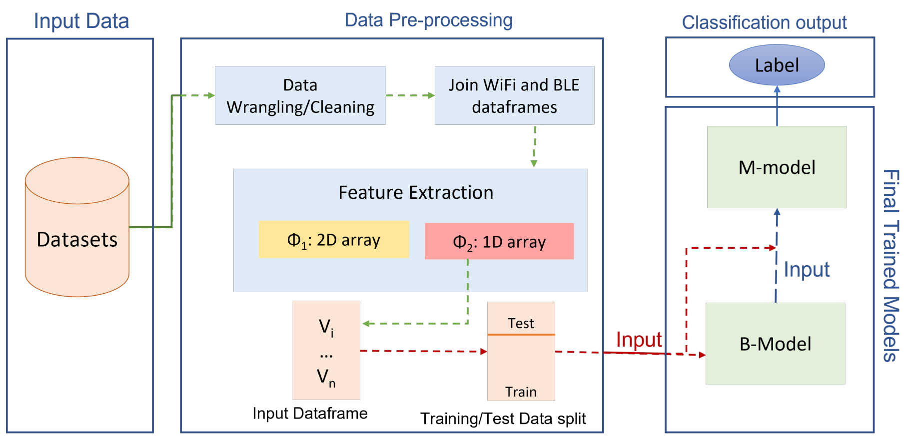


Figure 4.2: Data collection process diagram

4.2.6 Model Architecture

The global architecture consists of two models:

1. B-model (Binary Model):
   - Input: Aggregated statistics from WiFi and BLE scans.
   - Algorithm: Decision-tree-based ensemble methods (LGBM-Classifier and XGBClassifier).
   - Output: Binary label (stationary or mobile).
2. M-model (Multiclass Model):
   - Input: Features from WiFi and BLE scans, including the output from the B-model.
   - Algorithm: Decision-tree-based ensemble methods (LGBM-Classifier and XGBClassifier).
   - Output: Multiclass label (e.g., Home, Office, Bus, Train).

In this section, the model architecture is illustrated. As explained in Section 1.3, the aim is to extract knowledge from natural crowd mobility through radio beacons, by translating each "real-life" situation into a network model. The real-life situation refers to the context of a network, which is the mobility kind of the environment of a device. Thus, first we simplified the use case by classifying the context into two main categories: *Static* and *Mobile*.

The static scenario refers to any fixed scenario such as home, office, restaurant, library, etc. For further explanation, device A that is considered as static could be a fixed sensor located somewhere, or a device held by a person in an office, though the person could be moving in their office sometimes, but still will be considered as stationary since their status is still in the office (as no frequent movement over time is happening). On the other hand, mobile scenarios are assigned to devices that are in a moving context, like bus, car, train, pedestrian, etc. Similarly, if a device is in a bus, it will be considered mobile, since our reference is the global context of the scenario which is the bus, and not the individual reference with other devices in the bus. To this end, to achieve the main goal which is understanding mobility from which we can determine the real-life situation of a device (e.g. being in a bus, or train, or pedestrian, or at home, etc.), two models are defined. First, since the input data are labeled, our models will undergo supervised learning approaches. We started with the binary model called *B-model* that can determine the general situation of the device, which could be static or mobile. The output of this model helped to improve the performance of the second main model which is called *M-model*, by reducing its complexity, that stands for multi-class classification model, which aims to determine ten different status of a device that are: Train, Home, Office, Conference, Bus, Car, Metro, Pedestrian, Restaurant, and University. The output of the B-model will be one of the input features of the M-model.

4.3 Models Training and Evaluation

4.3.1 Training Process or Model training

The training process involves:

1. Collecting and preprocessing the data to ensure quality and consistency.
2. Extracting relevant features and transforming them into a suitable format for training.
3. Training the B-model to distinguish between stationary and mobile states.
4. Using the B-model's output as an additional feature for training the M-model to classify specific contexts.
   Two models were developed:
   - B-model: This model determines whether a device is stationary or mobile.
   - M-model: This model further classifies the device's precise context, such as home, office, bus, train, etc.

4.3.2 Evaluation Metrics

In this section, we give a brief analysis of the models' performance to select the best one for the mentioned use case. Typically various metrics will be reported that assess how well the model is able to make predictions on new, unseen data. The choice of the metrics depends on the type of the problem, so for both models B-model and M-model the following metrics will be evaluated: Accuracy, Balanced accuracy, and F1 score. Finally (AUC-ROC), an additional metric will be added for evaluating B-model, that stands for *area under the receiver operating characteristic curve*, it is a commonly used metric to evaluate the performance of binary classification models.

In general, F1 score, precision, and recall are metrics used in binary classification to evaluate the performance of a machine learning model where the *Precision* metric measures how often the model is correct when it predicts a positive sample. The formula for precision is:

$$Precision = \frac{TruePositives}{(TruePositives + FalsePositives)} \tag{4.5}$$

While the *Recall* metric measures how often the model correctly identifies a positive sample out of all the positive samples in the data. The formula for recall is:

$$Recall = \frac{TruePositives}{(TruePositives + FalseNegatives)} \tag{4.6}$$

But the F1 score is the harmonic mean of precision and recall. It combines both metrics to provide a single score that represents the model's overall performance. The formula for F1 score is:

$$F1Score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \tag{4.7}$$

Since Precision and Recall do not always provide a complete picture of a model's performance, and the F1 score balances the trade-off between precision and recall, as it provides a single score that summarizes both, then it will be used for the evaluation. Balanced accuracy in binary and multiclass classification problems is a metric to deal with imbalanced datasets. It is defined as the average of recall obtained in each class.

4.3.3 Models Evaluation

As described in Section 4.2.1, the dataset is labeled and collected over one year in different scenarios. The size of the dataset is 44,4 MB. The distribution of the datasets is unbalanced. To this end, with the available dataset for the moment, classical machine leaning algorithms will be suitable for our case since the dataset is small, and since it is a labeled dataset, supervised learning techniques will be applied to meet the final goal for constructing a classification model to guess the categorical label. In this section, several simulations are performed to evaluate our model to select the one with the best performance. Knowing that each dataset is scanned with a different time duration, first we divide all datasets to equal time spans. To achieve so, we define a period, and all datasets are divided by this period to finally have datasets with equal time duration. But to determine the best value of the period that will best perform on training the models, we will first define several period values and test the models on each of them to select the best period for each model. To this end, four periods are defined, P1: 5 min, P2: 3 min, P3: 2 min, P4: 1 min. Then we have selected 7 main classification algorithms that are: Boosting Decision Tree, KNN, Voting Classification, Gradient Boosting, Decision Tree, Neural Network, SVM, Naive Bayes, Random Forest Classifier, and Logistic Regression to be trained on the different selected periods, then compared the performance of each one.

4.3.4 B-Model: Binary Classification

The aim of this model is to determine whether the device is in a static or mobile context as defined in Section 4.1. First, we will train the seven selected models that are defined in Section 4.3.3 on the different defined periods. For each model, the accuracy after cross validation is calculated. Here is a description of the distribution of the datasets for each specified period:

- (P1: 5 min): 921 input vectors
- (P2: 3 min): 1576 input vectors
- (P3: 2 min): 2375 input vectors
- (P4: 1 min): 4764 input vectors

The distribution of datasets is as follows: 33.9% train, 17% Home, 10% office, 9% conference, 7.5% bus, 7.3% pedestrian, 7.3% university, 3.5% restaurant, 1.6% metro, and 2% car. The percentage may slightly change from one period to another. So, each model is trained on the different defined periods. To ensure that the models are robust and can generalize well, the dataset was split into 70% for training and 30% for testing. Given that the dataset is unbalanced, with varying proportions across different categories, the Stratify parameter was used in the splitting function. This ensures that each subset (train and test) maintains the same distribution of classes as the original dataset. This approach helps to prevent the model from being biased towards the more frequent classes and ensures a fair evaluation of its performance across all categories.

Figure 4.3 represents a clear comparison between the accuracy of different models. We can see that a period of one minute gave a higher accuracy in almost all selected models, while five minutes has the lowest accuracy. Thus a period of one minute will be selected to divide the datasets into equal fragments for training B-model.

After determining the best period for training the models, now 27 classification models are selected for training. From the 27 models, we have selected the most known models to compare, as table 4.4 displays 11 models, each with its calculated evaluation metrics. We can see that the LGBMClassifier and XGBClassifier gave an accuracy of 99%, and knowing that the trained dataset is not balanced, the *Balanced Accuracy* is calculated as it also gives a 99% accuracy, and same for F1 score. As a result, in our case, the boosting Ensemble method (LGBMClassifier and XGBClassifier) improved model performance since we have unbalanced datasets, as they assign higher weights to misclassified examples in the minority class.

To justify the importance of using information from both WiFi and BLE jointly for such a model, we repeated the training using only WiFi data, then only BLE, and compared the results with the models trained on both technologies jointly. Figure 4.4 displays the accuracy value (from cross validation) for each trained model for the three scenarios. The green curve represents the values from models that are trained with only BLE input data, and the red curve for WiFi input data only, thus we can see that WiFi information gives better accuracy than BLE input data in almost all trained models. The violet curve represents WiFi and BLE trained data jointly, and the results show a higher accuracy from such data, thus

Table 4.2: M-model binary classification period comparison

Model	*1 min*	*2 min*	*3 min*	*5 min*
LGBMClassifier	0.92	0.93	0.91	0.92
XGBClassifier	0.91	0.93	0.90	0.92
RandomForestClassifier	0.91	0.93	0.90	0.93
BaggingClassifier	0.89	0.91	0.88	0.90
SVC	0.89	0.90	0.88	0.86
AdaBoostClassifier	0.50	0.51	0.51	0.51
DecisionTreeClassifier	0.85	0.87	0.85	0.83
KNeighborsClassifier	0.84	0.83	0.83	0.80
LogisticRegression	0.85	0.87	0.87	0.84
RidgeClassifier	0.74	0.75	0.74	0.75
LinearDiscriminantAnalysis	0.19	0.12	0.17	0.25

we can conclude that both WiFi and BLE jointly gives better accuracy for estimating the output.

4.3.5 M-model: Multi-class Classification

Now, the aim of this model is to determine a specified context of the network. We have tested 26 different machine learning classifiers to train the multiclass model. Knowing that M-model has a different objective from B-model model, we need to repeat the same process for selecting the best period for classifying as in B-model. Table 4.2 shows the accuracy of 11 trained models on different period values. These models are selected from the 26 trained models as the most known models and with the highest accuracy among the others. We can see that a period of two minutes and five minutes give a better accuracy in almost all tested algorithms. Thus, (P3 = 2 min) will be selected as the period to train the models. We can see that the highest accuracy is equal to 93% from the LGBMclassifier and XGBclassifier and RandomForestClassifier as an example. To improve even more the accuracy, we added the results of B-model as an input feature to M-model to see how could the static/mobile information enhance the performance of the models. Table 4.5 shows the results of the same 11 models but with input of the B-model. We can see that the accuracy increased by 0.01 (or 1%) in almost all models, thus this indicates the importance of the information from B-model to give a better accuracy in determining the real-life situation of the device from M-model. After determining the best period for the classification which is two minutes, and improving the accuracy by the input values from B-model, the first three models that have higher accuracy that are: The LGBMClassifier, XGBClassifier, and RandomForestClassifier are selected. We will test our chosen models again by getting the accuracy after cross validation and calculating the Time consumed for training and prediction.

As shown in Table 4.3, XGBClassifier has a higher accuracy among the other selected models after cross-validation, with the shortest prediction time, and then it will be selected for hyper parameter tuning. After hyper parameter tuning, the accuracy remained the same, therefore, the XGBoost classifier has the best accuracy with approximation to 94%.

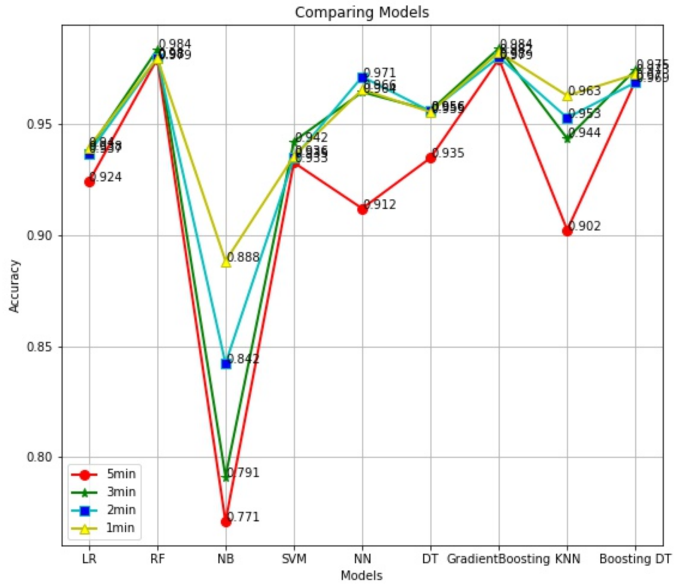


Figure 4.3: Period comparison for B-model

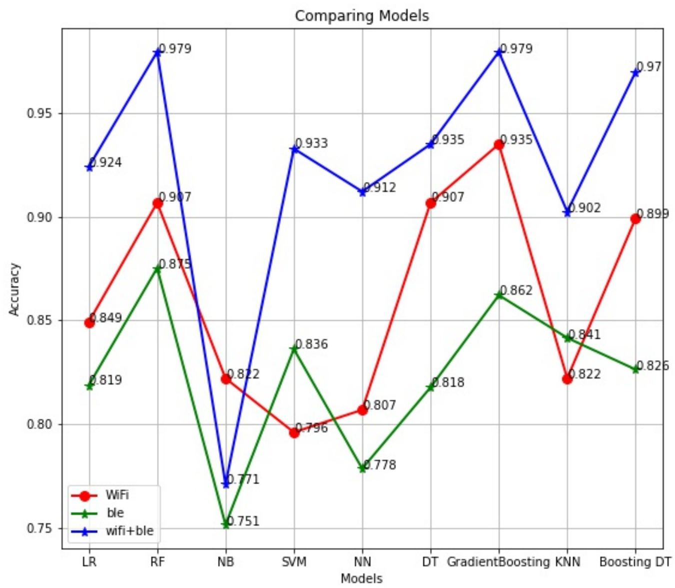


Figure 4.4: Accuracy comparison for three different input datasets

Table 4.3: Comparison between best three models

Model	*Accuracy*	*Training Time*	*Pediction Time*
XGBClassifier	93.62%	0.868	0.00436
LGBMClassifier	92.95%	0.907	0.00706
RandomForestClassifier	93.38%	0.433	0.01808

Table 4.4: B-model models evaluation and comparison

Model	*Accuracy*	*Balanced*	*F1 Score*
LGBMClassifier	0.99	0.99	0.99
XGBClassifier	0.99	0.99	0.99
RandomForestClassifier	0.98	0.98	0.98
BaggingClassifier	0.98	0.98	0.98
SVC	0.97	0.97	0.97
AdaBoostClassifier	0.97	0.97	0.97
DecisionTreeClassifier	0.95	0.95	0.95
KNeighborsClassifier	0.94	0.94	0.94
LogisticRegression	0.93	0.93	0.93
RidgeClassifierCV	0.92	0.92	0.92
LinearDiscriminantAnalysis	0.92	0.92	0.92

Model Performance Analysis

Model performance evaluation is crucial to understand not only the overall accuracy but also the behavior of the model across different classes, particularly in multiclass classification tasks. To this end, we present both a confusion matrix (Figure 4.5) and a class-wise accuracy table (Table 4.6) to highlight the strengths and weaknesses of the M-model. In both the confusion matrix and the class-wise accuracy table, each number (0 to 9) represents a specific class, corresponding to different environments in the dataset. For example, Class 0 represents "Home," Class 1 is "Office," and so on, as detailed in the class-wise accuracy table. The same numerical labels are used in the confusion matrix, where each row corresponds to the actual class and each column to the predicted class. This ensures a consistent reference for interpreting both the confusion matrix and the class-wise performance.

The confusion matrix provides a detailed view of actual versus predicted classifications. High diagonal values indicate strong performance in correctly predicting certain classes, such as "Home" (118 correct predictions) and "Train" (233 correct predictions). However, misclassifications become evident for classes like "Metro," where only 10 instances were correctly classified, with several others misclassified as "Bus" or "Train." This is reflected in the class-wise accuracy table, where "Metro" has a notably low accuracy of 28.57%, in contrast to other classes like "Home" (99.16%) and "Train" (96.68%).

Despite the lower accuracy for certain classes like "Metro" and "Bus," the overall model accuracy remains high at 94%. This is due to the dominance of correctly classified instances in larger classes, such as "Home," "Train," and "Office," which contribute significantly to the overall performance. Therefore, while the model performs well in general, attention is needed

Table 4.5: M-model with B-model input feature

Model	*Accuracy*	*Balanced*	*F1 Score*
XGBClassifier	0.94	0.86	0.93
LGBMClassifier	0.93	0.86	0.93
RandomForestClassifier	0.94	0.85	0.93
BaggingClassifier	0.92	0.86	0.92
LogisticRegression	0.88	0.74	0.86
SVC	0.90	0.75	0.88
DecisionTreeClassifier	0.88	0.80	0.87
KNeighborsClassifier	0.85	0.69	0.84
RidgeClassifierCV	0.77	0.51	0.72
AdaBoostClassifier	0.51	0.20	0.36

Table 4.6: Class-wise accuracy

	Class	*Accuracy*
0	Home	99.16%
1	Office	97.14%
2	Conference	95.24%
3	Bus	78.57%
4	Metro	28.57%
5	Train	96.68%
6	Pedestrian	94.23%
7	University	96.08%
8	Restaurant	72.00%
9	Car	71.43%

to improve its performance in more challenging classes like "Metro" to ensure balanced and robust results across all environments. This could be achieved by enriching the datasets for Metro and Bus or adding a new feature for training the model, like acceleration.

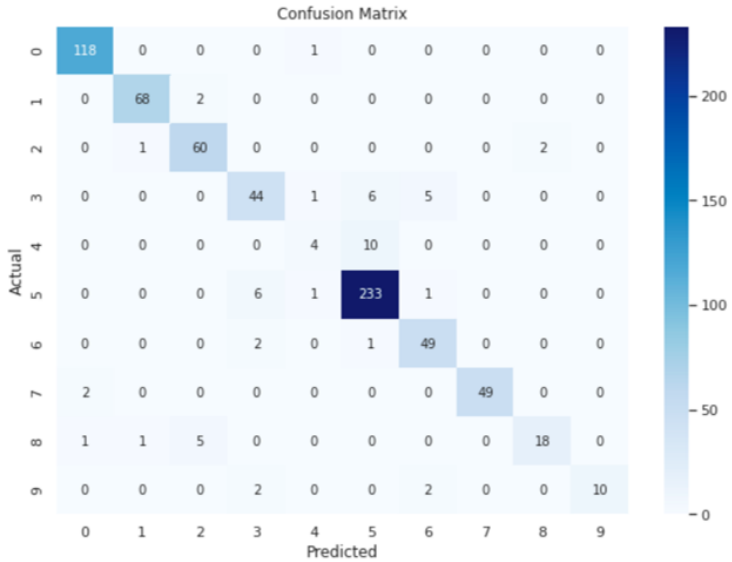


Figure 4.5: Confusion Matrix

4.4 Conclusion and Discussion

In this Chapter, we have proposed a novel approach for inferring human mobility by determining a device's status within its network context through wireless communication technologies, namely WiFi and BLE, working in conjunction. Firstly, we trained a model to ascertain whether a device is in a mobile or static network context. Subsequently, a complementary model was trained, providing a more precise classification of the device's real-life situation. These models were trained using real datasets collected over one year, for 90 hours, across various scenarios and conditions. In our initial approach, we achieved 94% accuracy in classifying among 10 scenarios using a lightweight classical machine learning algorithm (XGBClassifier). More analysis will be done in this work like getting the accuracy for each model, investigating in the feature selection and finally checking the fairness of the model.

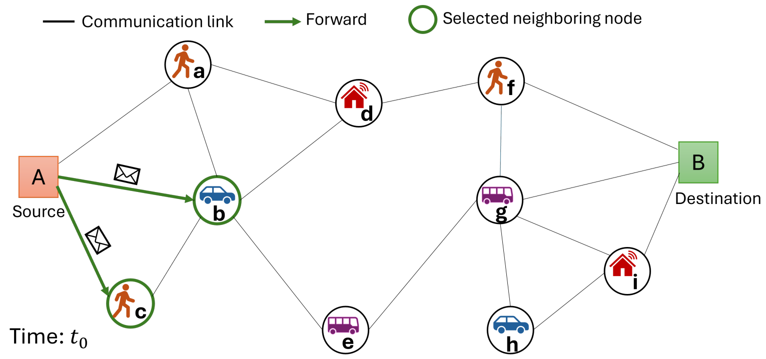
To this end, we have validated the hypothesis of the capability of determining the devices' real-life situation through the observations of wireless traces in the range of a device. This also shows the added value of using jointly several kinds of information, and to get better accuracy we will investigate later how to also leverage LoRa and acceleration data. We will carry more analysis for the routing. This method will assist the routing protocol that will be based on the mobility of the devices to take a decision for forwarding a message that is illustrated in the next chapter.

Routing Protocol | 5

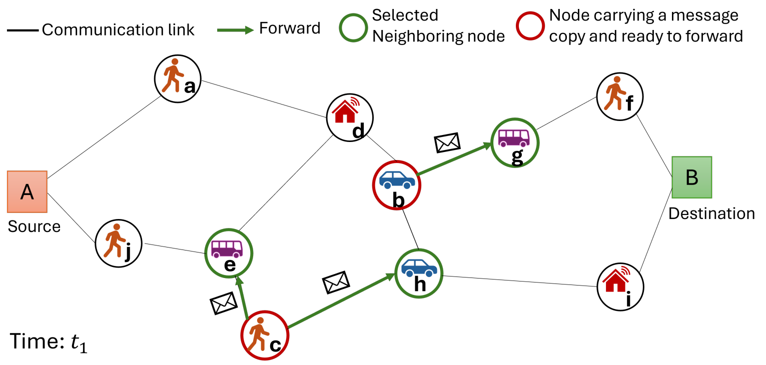
In this chapter, we present 'BatonRelay' a privacy-aware and mobility-aware routing protocol. This protocol is based on the mobility of nodes in a network. After investigating the possibility of inferring the mobility context of a device from the observation of wireless technologies in its range (in Chapter 4.1), we will study the concept of leveraging the context or mobility of a token to make the decision for routing. This will help to analyse the impact of the mobility context and how it is taken into account on the performance of routing protocols in DTN. The encounter probability to carry or forward the message is based on a mobility level assigned to each token. First, we implement 'BatonRelay' and then present a valuable analysis of the results, and finally compare to the most known DTN protocols presented in the literature, and discuss the limitations and ongoing work to improve 'BatonRelay'. The main goal is to implement a robust simple routing protocol that will base the forwarding decision in real-time without extensive computations or huge storage history to encounter a forwarding probability and based on the knowledge of mobility patterns in a privacy-preserving approach.

5.1 BatonRelay Routing Protocol

This routing protocol prioritises connections with higher mobility values for message forwarding. The logic ensures that messages are sent to nodes with higher mobility, which are likely to carry the message further and faster. If no suitable connections are found, the message is carried by the current host until a better connection is available. We believe such a structured approach would allow efficient message dissemination in a DTN, where the network topology is dynamic and the connectivity is intermittent. Thus, to prove this claim, we will first explain the algorithm and the behaviour of the protocol. Figure 5.1 illustrates the scenario for forwarding a message in 2 consecutive time-slots. First, each device will determine its local context through a machine learning model by scanning its network environment as illustrated in 4.1, then share this information with its neighboring devices. In Figure 5.1a, device A which is static, wants to send a message to the destination B. From the knowledge of device's A context, a delivery probability (f to be defined later) is computed to decide whether to forward the message or carry it. Then device A estimates whether it needs to send two copies of the message to devices b and c because they have a higher mobility than itself. Then crowds move and wireless connections dynamically appear and disappear. In Figure 5.1b, device b, which has carried the message, will repeat the same process of forwarding the message to the devices with higher mobility, thus it will exclude device h since it has the same mobility context as itself and thus there is no added value to forward the message for such device. This process will be repeated on each device that is holding a copy of the message until the message reaches its destination.



(a) Scenario at time 0



(b) Scenario at time 1

Figure 5.1: Illustration of Baton-relay protocol

5.1.1 Design Principles

To design the routing protocol, we need as a first approach to investigate how the mobility of a node affects the routing protocol at different levels (delivery probability, buffer time, etc.). To this end, we implemented the first approach for the DTN routing protocol that is based on a threshold assigned based on the mobility type of a node.

First, let's assume we have a group of nodes:

- **Nodes and Messages**:
  - Let $N = \{n_1, n_2, \ldots, n_k\}$ be the set of nodes in the network.
  - Let M be the message to be delivered from a source node n_s to a destination node n_d.

  Now, we will assign a mobility level μ] for each mobility type as follow:

  $$\mu : N \rightarrow \mathbb{R}$$

  where $\mu(n_i)$ represents the mobility level of node n_i.
  And each node n_i has a threshold T_i associated to its mobility level, i.e., $T_i = f(\mu(n_i))$, where we define the identity function f defined as $f(x) = x$ as a first step for simplicity. When node n_i encounters node n_j, they exchange their mobility levels $\mu(n_i)$ and $\mu(n_j)$.

Algorithm Description

First we start with initialization, where the source node n_s initiates the process with the message M and its mobility level $\mu(n_s)$. Upon encountering a new node n_j, the current node n_i will share its mobility level $\mu(n_i)$ with n_j and receive $\mu(n_j)$, and then compare with its threshold T_i. If $\mu(n_j) > T_i$, then node n_i forwards the message M to node n_j, otherwise, node n_i retains the message M and continues to carry it. The process continues until the message M reaches the destination node n_d or TTL expires.

The recursive process at node n_i could be described as follows:

$$\text{while } n_i \nrightarrow n_d \text{ or } TTL = 0 \text{ do}$$

$$\text{for each encounter } (n_j) :$$

$$\begin{cases} \text{if } \mu(n_j) > T_i \\ \quad \text{then forward}(M, n_j); \\ \quad \text{based on } T_i : \\ \quad \text{carry}(M, n_i) \text{ or remove}(M, n_i) \\ \text{else carry}(M, n_i) \end{cases}$$

So, BatonRelay relies on the concept of node mobility as a metric to improve message forwarding in a delay tolerant network. By assigning mobility levels to nodes based on their mobility types, the algorithm ensures that messages are forwarded to nodes with higher mobility, which we believe could have a higher chance of encountering the destination or other high-mobility nodes. Algorithm 1 explains the behavior of the routing protocol.

Algorithm 1 BatonRelay, run on node n_i holding a message M to destination n_d

Input: M, $\mu(n_i)$
for each encountered n_j **do**
    // Step 2: Share mobility levels between n_i and n_j
    receive $\mu(n_j)$
    send $\mu(n_i)$
    // Step 3: Check if node_j's mobility level is greater than node_i's threshold
    if $\mu(n_j) > T_i$ **then**
        // Step 4: Forward the message M from n_i to n_j
        forwardMessage(n_i, n_j)
        // Decide if n_i will carry or remove the message based on the threshold T_i
        carry(M,n_i) or remove(M, n_i)
    end if
end for

5.2 Simulation Setup

In this section, we provide an overview of the environment (ONE simulator) that is used to implement the BatonRelay and other approaches from the literature: Prophet, Spray & Wait, and Prophet protocols. Next, we define the settings for the simulation and the metrics for evaluation.

5.2.1 Tools and Scenario Description

In our evaluation of the proposed routing protocol, we used the Opportunistic Network Environment (ONE) Simulator [88]. ONE is a Java-based simulation tool designed for evaluating delay-tolerant networking protocols. It provides a platform for simulating various network scenarios and mobility models, enabling comprehensive performance analysis of routing protocols. The selection of the ONE Simulator is based on its strong DTN simulation capabilities, and availability of many integrated mobility models and network configurations. It is especially helpful for evaluating the mobility-based routing protocol because of its ability to display network performance data and node movements.

[88]: Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. 'The ONE simulator for DTN protocol evaluation'. In: *Proceedings of the 2nd international conference on simulation tools and techniques*. 2009

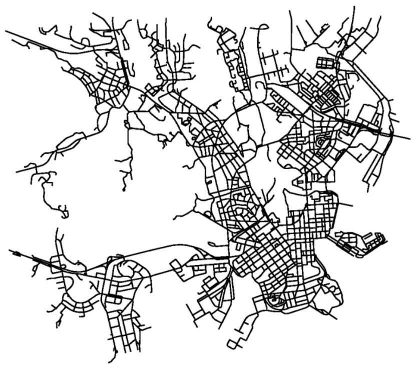


Figure 5.2: Helsinki Map

5.2.2 Parameters

Table 5.1 summarizes the general configuration of the simulation used to evaluate and analyze the performance of the routing protocols. However, Table 5.2 shows in more detail the configuration for each mobility type used (number of nodes and buffer size).

We defined 5 groups of nodes each with different mobility types: 80 pedestrians (separated into two groups with different buffer sizes), 40 cars, 6 buses, and 8 fixed nodes, so in total we have 133 nodes in the network. Furthermore, the release contains map data of the Helsinki downtown area (Figure 5.2) (roads and pedestrian walkways) that the map-based movement models can use.

5.2.3 Performance Metrics

To evaluate our routing protocol, we investigate the performance of all four protocols, i.e. Epidemic [3], Spray and Wait[43], PRoPHET [4], and BatonRelay, through simulations. The performance of these four

[3]: Amin Vahdat, David Becker, et al. *Epidemic routing for partially connected ad hoc networks*. 2000

[43]: Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. 'Spray and wait: an efficient routing scheme for intermittently connected mobile networks'. In: *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. 2005

[4]: Anders Lindgren, Avri Doria, and Olov Schelén. 'Probabilistic routing in intermittently connected networks'. In: *ACM SIGMOBILE mobile computing and communications review* 7.3 (2003)

Table 5.1: Routing Protocol Parameters

Parameter	Value
Simulation Time	43200 seconds (12 hours)
Interface Class	Bluetooth
Transmit Speed	250 KBps
Message size	500KB to 1MB
Transmit Range	10 meters
Number of Node Groups	14
Movement Model	Shortest Path Map-Based Movement (SPMBM)
Router	BatonRelay, Epidemic, Spray&Wait, ProPHet
Buffer Size	5M, 10M, 50M, 200M, 500M
TTL (Message Time to Live)	300, 240, 180, 120, 60, and 30 minutes.

Table 5.2: Routing Protocol Parameters

Group ID	Number	Buffer size
Pedestrian P1	40	5M
Pedestrian P2	40	10M
Cars C1	40	50 M
Bus B1	6	200M
Fixed F1	8	500M

protocols are investigated with different traffic patterns and TTL values. Performance is evaluated using the following five metrics:

- Copies overhead (CO) = the ratio of the number of copies against the number of original messages. This metric will calculate the cost of extra generated copies in the network as follows:

$$CO = \frac{N - M}{M} \tag{5.1}$$

  Where N is the created copy and M is the total number of original messages created.
- Delivery Ratio: the ratio of successful message deliveries to the total number of original messages created and sent.
- Hop count average: is the average number of hops each node has relayed on to reach the destination.
- Buffer time average: the average time that a message spends in the buffer of a node before it is successfully delivered or dropped.
- Average Latency: the average time the message takes to be forwarded from the source to the destination.

5.2.4 Statistical Robustness in Simulation

To ensure the robustness and reliability of our simulation results for the different Delay Tolerant Network (DTN) routing protocols, we employed a comprehensive simulation strategy involving multiple runs and various traffic scenarios. This approach aims to reduce the likelihood of unusual results caused by particular message structures or network conditions.

Message Creation

1500 messages are created to be routed in the simulated environment. The source and the destination for each created message is selected randomly in a range from 0 to 133, where 133 is he total number of nodes in the network as defined in the previous section. The selection of the nodes of the message origin and destination is randomized to ensure a broad representation of possible network interactions. This variability in the traffic files helps to test the protocols under a range of conditions, thus enhancing the validity of our findings.

5.2.5 Confidence Interval Calculation

Each routing protocol was subjected to 20 independent simulation runs. For each simulation, a unique traffic file was used. To determine whether

[89]: https://en.wikipedia.org/wiki/Confidence_interval

20 simulation runs are sufficient to provide reliable results, we calculated the confidence intervals [89] for key metrics: delivery probability and average latency. The confidence interval gives us a range in which we can expect the true population parameter (mean in this case) to fall with a certain level of confidence. Here is how the confidence interval is calculated:

$$CI = \bar{x} \pm z\left(\frac{s}{\sqrt{n}}\right)$$

Where:

- $\bar{x}$ is the sample mean of the metric (e.g., delivery probability or latency).
- z is the z-score corresponding to the desired confidence level (e.g., 1.96 for 95% confidence).
- s is the sample standard deviation.
- n is the number of samples (20 simulations).

The choice of the 95% confidence level is typical in empirical research, providing a good balance between confidence and precision. For our simulations, the resulting confidence intervals for both delivery probability and latency is [0.898, 0.905], which is considered narrow, indicating a small difference between the lower and upper bounds. This consistency across simulations suggests that the sample of 20 runs is sufficiently robust, reflecting stable and reliable results under varying network conditions.

5.3 Results and Discussion

In this section, we first analyse the routing protocol individually, then in the second part, we compare the results of BatonRelay with the other routing protocols known in the literature.

5.3.1 Performance Analysis of BatonRelay Protocol

As explained in Section 5.1, the decision to forward or carry a message in BatonRelay protocol is based on the node with a higher mobility. To validate this intuition and validate its effectiveness, we implemented a reverse version of BatonRelay which is called 'BatonRelay-Reversed'. The message in BatonRelay-Reversed is forwarded to the node with a lower mobility. So, if the sender is in a car and has in its range a bus and a pedestrian, the message will be forwarded to the pedestrian and not to the bus. The simulation is done for both protocols under the same conditions and traffic file. Table 5.3 displays the results after comparing the approaches.

Based on the comparative performance data between BatonRelay and BatonRelay-Reversed routing protocols, BatonRelay demonstrates overall a better performance. In particular, BatonRelay shows a significantly higher delivery probability at 0.85 compared to just 0.59 for BatonRelay-Reversed. This metric is crucial as it indicates a more reliable message

Table 5.3: Comparative Performance of BatonRelay vs. BatonRelay-Reversed Routing Protocols

Metric	BatonRelay	BatonRelay-Reversed
Delivery Probability	0.85	0.59
Latency Average (s)	2556.89	4306.34
Buffer Time Average (s)	12492.70	12475.54
Copies overhead	15.70	15.19

delivery under the BatonRelay protocol. Additionally, BatonRelay has a lower average latency of 2556.89 seconds, which is nearly 1750 seconds faster than the 4306.34 seconds recorded for BatonRelay-Reversed, indicating more efficient message handling.

The better performance of the BatonRelay protocol over that of the BatonRelay-Reversed can be explained by the strategic use of mobility characteristics to improve the efficiency of message delivery. As mentioned earlier, in the BatonRelay protocol, forwarding decisions are based on selecting nodes with higher mobility. This approach naturally increases the likelihood that the message will be carried over larger distances in shorter periods of time, using the faster or more widely travelling nodes to speed up message dissemination. Higher mobility nodes, such as buses or cars, are more likely to encounter a greater number of other nodes, thereby increasing the probability of delivery and reducing the overall latency of the network. In contrast, the BatonRelay-Reversed protocol, which routes messages to low-mobility nodes, naturally limits the potential distance and speed at which a message can travel. Low-mobility nodes, such as pedestrians, have restricted movement and fewer encounters with other nodes, leading to increased chances of message delay and a lower probability of successful delivery.

Thus, while both protocols theoretically test the impact of mobility on routing effectiveness, BatonRelay's strategy of using higher-mobility nodes capitalises on the rapid and extensive spread of messages across the network, as this was proven by simulation and thus validates our assumption.

5.3.2 Comparison with existing Protocols

After validating the choice of BatonRelay protocol for choosing the nodes with higher mobility to relay on, in this section we investigate in the performance of BatonRelay routing protocol compared to Epidemic [3], Spray&Wait [90] and PRoPHET [4] protocols. More details related to the functionality of the protocols are illustrated in Chapter 3.1.1. Having established confidence in our simulation setup, we proceeded to compare the performance of each routing protocol across the 20 simulations. To ensure that the results reflect the typical performance of the protocols under a variety of scenarios, we averaged the results (e.g., delivery ratio, latency, buffer time) of each protocol from the 20 simulations.

[3]: Amin Vahdat, David Becker, et al. *Epidemic routing for partially connected ad hoc networks*. 2000

[90]: Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. 'Spray and wait: an efficient routing scheme for intermittently connected mobile networks'. In: *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. 2005

[4]: Anders Lindgren, Avri Doria, and Olov Schelén. 'Probabilistic routing in intermittently connected networks'. In: *ACM SIGMOBILE mobile computing and communications review* 7.3 (2003)

Table 5.4 displays the statistical results of the four routers with the main metrics that are important to give a logical comparison between the different approaches. From the results, we can see that BatonRelay achieves a delivery probability of 0.85, indicating reliable performance,

Table 5.4: Comparison of Routing Protocol Performance

Protocol	Delivery Probability	Latency Avg (s)	Buffer Time Avg (s)	Hop Count Avg	Copies Overhead
BatonRelay	0.85	2556.89	12492.7	2	15.70
Epidemic	0.90	936.67	13000.3	4	71.34
Prophet	0.90	1037.99	13800.4	4	68.55
Spray and Wait	0.85	2521.55	17649.72	2	3.21

slightly lower than the Epidemic and Prophet protocols, which both exceed a delivery probability of 0.90.

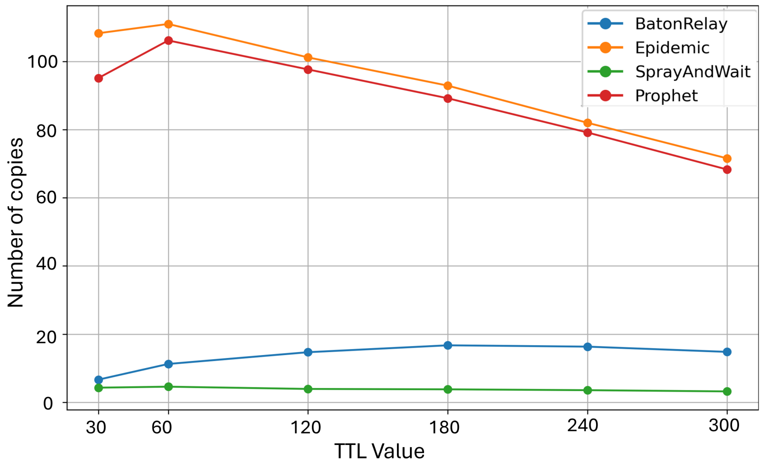
Although Epidemic and PRoPHET protocols display exceptionally low latency averages of 936.67 s and 1037.99 s, respectively, BatonRelay overcomes these protocols buffer time average, hop count, and copies overhead. As Epidemic and PRoPHET has a higher a higher average hop count, which increases the network overhead and complexity. Similarly for the copies overhead, as it presents a clear trade-off between delivery assurance and network resource utilisation. Epidemic and Prophet have high overheads of 71.34 and 68.55 respectively, which can lead to network congestion and increased power consumption, while BatonRelay, with a copies overhead of 15.70, provides a balanced approach, ensuring reasonable delivery success without excessive resource usage.

Finally, we can see that Spray&Wait overcomes BatonRelay in copies overhead, while having the same delivery probability, latency average, and hop count. However, BatonRelay shows a significant improvement for the buffer time average compared to Spray&Wait (and the other protocols as well). However, there is always a trade-off to consider when selecting a routing protocol and choosing according to the application requirements. Thus, in environments with limited storage or buffer capacities, BatonRelay will be a better choice and a better compromise over copies overhead compared to Spray&Wait, especially that Baton-relay does not require history storage and guarantees devices' privacy.

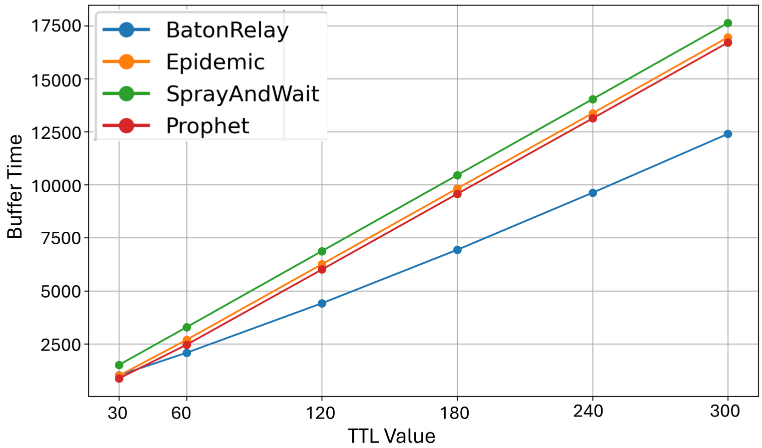
TTL Analysis

To further explore the dynamics of the BatonRelay protocol, we conducted additional analysis by varying the Time-to-Live (TTL) settings to observe their impact on overall performance. This investigation aimed to understand how different TTL values affect metrics such as delivery probability, latency, average buffer time and copies overhead. The results of these experiments are comprehensively illustrated in the following figures (5.3a, 5.3b, 5.4a, 5.4b) , providing a detailed visualisation of how TTL adjustments affect the routing efficiency and effectiveness of the BatonRelay protocol. Five TTL values are considered: 300 min, 240 min, 180 min, 120 min, 60 min and 30 min. To analyze the provided graphs, we will summarize the trends and patterns observed in each graph. Figure 5.3a shows the copies overhead values as function of different TTL values for the four routing protocols. Figure 5.3b shows the average buffer time, Figure 5.4b displays the average latency for each protocol and finally Figure 5.4a shows the delivery ratio.

We can observe that for BatonRelay, the number of copies increases linearly. Prophet produces the most copies, which might indicate higher



(a) Cost of extra number of copies

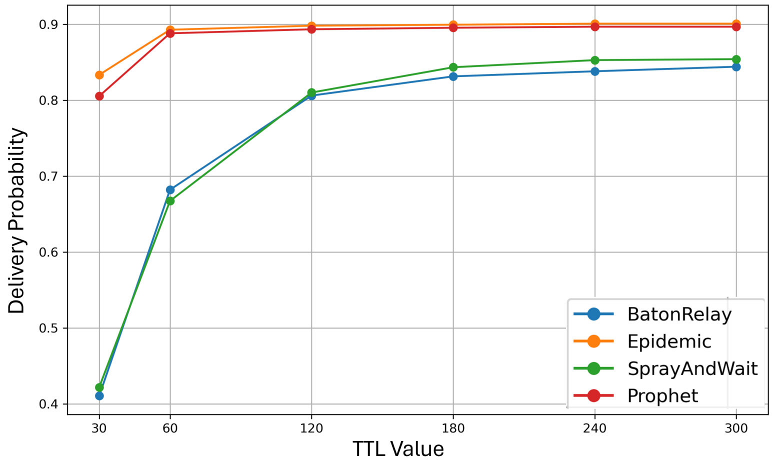


(b) Buffer Time

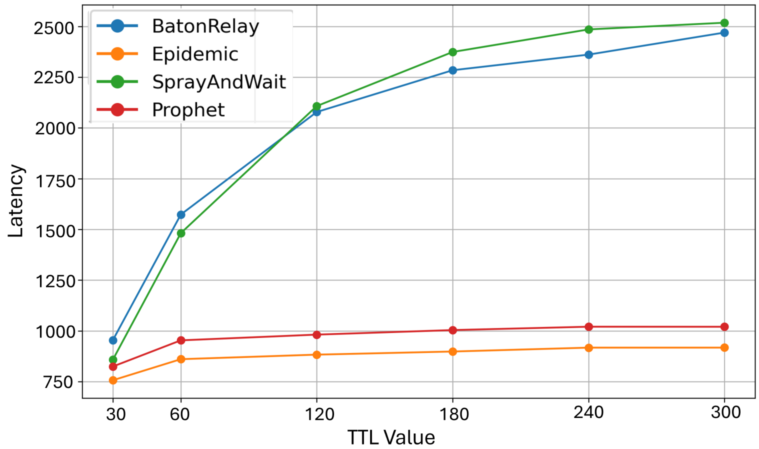
Figure 5.3: TTL comparison

resource usage. BatonRelay is the most conservative in terms of the number of copies, which might be beneficial in resource-constrained environments.

BatonRelay has the most favorable buffer time characteristics, with an initial increase that stabilizes quickly. Epidemic and Prophet show high buffer times, which might be a drawback in time-sensitive applications.



(a) Delivery Ratio



(b) Latency

Figure 5.4: TTL comparison

5.4 Conclusion and Ongoing Work

As a first approach to analyse the impact of the mobility of a token on the routing decision, we implemented the BatonRelay routing protocol. BatonRelay emerges as a balanced protocol, ideally suited for environments where efficient resource utilisation is crucial without significantly compromising delivery probability and latency. BatonRelay's strategic use of high-mobility nodes to propagate messages is recommended for urban environments with a mix of vehicular and pedestrian traffic. The idea of BationRelay is that it is simple and not based on extensive mathematical calculations and does not require a huge memory or buffer, but yet robust as it guarantees a reasonable probability delivery ratio and most importantly it is privacy preserving.

As an ongoing work, we are analysing the algorithm with different conditions to see where it could perform the best, like testing the effect of node density and mobility movement models. This will help to understand how different concentrations of nodes and different mobility movement nodes influence the efficiency and reliability of the protocol compared to other protocols. In addition, further analysis will be done to analyse the delivery probability and latency values over number of hops and investigate in the number of copies successfully delivered, and the designed function f, which will help in reducing the number of generated copies. Finally, an updated version of the protocol is being implemented to improve its performance (delivery probability, copies overhead, etc.), by refining the algorithms used to assign delivery probabilities. Enhancements could focus on more accurately predicting node interactions based on previous connections information and current network conditions, thereby increasing message delivery rates.

General Conclusion and Perspective 6

6.1 Conclusion

In this thesis, we proposed and evaluated a novel Delay Tolerant Network (DTN) routing protocol, called BatonRelay. This protocol has been designed to leverage the mobility patterns of nodes for efficient message forwarding. The primary aim behind this work was to overcome and address the limitations of existing DTN routing protocols through the introduction of the *Context-aware* approach that prioritizes nodes with higher mobility for message delivery. This approach has been thoroughly tested and validated through extensive simulations using the *Opportunistic Network Environment (ONE) Simulator.*

The key contributions in this thesis are summarised as follows:

1. **PILOT Dataset**: The creation of a unique dataset capturing multi-communication technologies in various mobility contexts.
2. **Machine Learning Models**: Analysis of collected data and development of models to infer the real-life situation of devices based on wireless communication traces.
3. **BatonRelay Routing Protocol**: Introduction of a novel DTN routing protocol that prioritizes nodes with higher mobility for efficient message forwarding.

The PILOT dataset was introduced in Chapter 3. The main aim behind this dataset is to address the lack of existing datasets that include multiple traces of wireless communication technologies recorded simultaneously in different mobility contexts. The dataset incorporates WiFi beacons, Bluetooth Low Energy (BLE) advertisements, LoRa packets, and sensor-based measurements such as acceleration and pitch. This data was collected using FiPy microcontrollers configured to capture information in various static and mobile scenarios. The dataset was meticulously gathered to ensure replicability, using four FiPy devices dedicated to different types of data collection to avoid delays and improve accuracy. Each device scans and records data with real-time synchronization, saving logs on SD cards. The data collection covers diverse environments, including homes, offices, public transportation, and various outdoor settings. The chapter outlines the methods and technologies used for data collection, justifies the choices made, and discusses the challenges faced, such as hardware limitations and time synchronization issues. The dataset is labeled based on the scenarios and is privacy-preserving, complying with GDPR regulations by pseudonymizing device names and hashing MAC addresses. The overall collected data spans about 90 hours in total in different mobility scenarios, and it is uploaded to GitHub.

In Chapter 4, a machine learning-based model for mobility contexts classification using WiFi and Bluetooth Low Energy (BLE) advertisement packets was introduced using the developed PILOT dataset. The main objective was to design a model capable of performing multiclass classification to identify the specific context of the device (e.g., Home, Office, Bus, Train) when it is mobile or static. The data collected was pre-processed and features were extracted focusing on contact duration with the scanner, Received Signal Strength Indication (RSSI) and the number of unique MAC addresses detected. Two models were developed as follows:

- **B-Model (Binary Model)**: To classify the device as stationary or mobile.
- **M-Model (Multi-class Model)**: To identify the specific context of the device when it is mobile.

Both models utilized decision-tree-based ensemble methods (LGBMClassifier and XGBClassifier) for their balance between performance and computational efficiency. The models were trained and validated using cross-validation to ensure robustness. The evaluation metrics used included accuracy, balanced accuracy, and F1 score, with additional AUC-ROC for the B-Model. The B-Model achieved high accuracy in distinguishing between stationary and mobile states, and the M-Model successfully identified specific contexts, demonstrating the effectiveness of using WiFi and BLE data for mobility context classification.

After validating the assumption that it is possible to determine the real-life situation of a device through wireless technologies (WiFi and BLE), the design of the routing protocol *BatonRelay* that leverages this knowledge about mobility was carried out. As described in Chapter 5, this protocol leverages the mobility type of a device to forward messages. BatonRelay routing protocol was implemented by assuming knowledge of the node's mobility type, rather than integrating the model directly into the simulator, so it operates under the assumption that nodes with higher mobility are more likely to efficiently forward messages to their destination. Thus, the design principle is based on the mobility-based forwarding, where nodes are assigned a mobility level, and the protocol prioritizes connections with higher mobility values for message forwarding that is learnt from knowledge of previous chapter. If a node with a higher mobility value is encountered, the message is forwarded; otherwise, the current node continues to carry the message. The protocol is unique due to its simplicity and efficiency. It is designed to be a simple yet robust routing protocol that makes real-time forwarding decisions without extensive calculations or large storage requirements. The performance of BatonRelay was compared with other well-known DTN protocols such as Epidemic, Spray and Wait, and PRoPHET. Key metrics for evaluation included delivery probability, latency, buffer time, and copies overhead. BatonRelay demonstrated a balanced performance, offering a high delivery probability and moderate latency without excessive resource consumption as it has the best buffer time usage compared to the other protocols, and it maintained a reasonable copies overhead, ensuring efficient and robust use of network resources without causing congestion and incorporating privacy measures. The results obtained from simulations and real-world data collection validate the hypothesis that context-aware routing based on human mobility significantly improves the performance of DTNs. The proposed BatonRelay protocol, in particular, demonstrates superior performance in terms of delivery probability and latency compared to traditional protocols like Epidemic and Spray and Wait.

6.2 Future Work and Perspective

6.2.1 Future Work

While the results achieved in this thesis are promising, there are several avenues for future work that can further enhance the robustness and applicability of the BatonRelay protocol, illustrated below:

- **Integration of Additional Data Sources**: Incorporating data from additional sources such as acceleration sensors and LoRa could improve the accuracy of mobility detection and routing decisions. This would provide a more comprehensive understanding of node behaviors and improve forwarding efficiency.
- **Advanced Time Series Analysis**: Implementing time series analysis with a larger dataset could enable more precise predictions of node movements, enhancing the ability to forecast the context or next steps of the nodes. This predictive capability can be integrated into the routing protocol to improve real-time forwarding decisions.
- **Enhanced Routing Protocols**: Developing advanced routing protocols that adapt to the changing contexts of the network will be crucial. This includes considerations for energy efficiency, security, and evolving network conditions. Integrating security measures. Our ambition is to implement BatonRelay protocol in real-world environments. This implementation will move beyond simulation, allowing us to test the protocol under diverse and unpredictable conditions. Field tests will help validate the protocol's performance and robustness, providing practical insights that can refine and improve its operational parameters. Deploying the protocol in various scenarios, such as urban areas, rural regions, and disaster-prone zones, will demonstrate its versatility and effectiveness in different contexts.

By pursuing these future directions, we aim to significantly improve the efficiency and robustness of DTN routing protocols, making them more suitable for a wide range of real-world applications. This will ultimately contribute to better connectivity and communication in challenging network environments, supporting the growing demand for reliable data transmission in mobile and opportunistic networks.

6.2.2 Perspective

Contextual Analysis

Further analysis of the dataset could yield insights into various network contexts, such as detecting whether a node is in a crowded place or moving at different speeds. This contextual information can be leveraged to refine the routing protocol, making it more adaptable to dynamic network conditions. Moreover, the study of human mobility and flow has significant implications beyond telecommunications. For instance, understanding movement patterns can help design more efficient transportation systems, manage public health through the study of virus propagation, and improve telecommunication networks by predicting and mitigating congestion. These insights can thus contribute to a wide range of applications, enhancing both technological and societal outcomes.

Deploying Resilient Routing Protocols

Additionally, one of the primary objectives of this research is to design a routing protocol for Delay Tolerant Networks (DTN) that can connect the unconnected, with no reliance on existing infrastructure or authorities. This is particularly crucial in remote or underserved areas where traditional network infrastructure is either unavailable or unreliable. By leveraging the inherent mobility patterns and contextual information derived from the dataset, the proposed routing protocol can enable robust and efficient communication in challenging environments. The goal of these developments is to link the disconnected without relying on the government or the existing infrastructure. This approach facilitates information access by guaranteeing resilience, autonomy, and privacy in network connectivity, while simultaneously fostering inclusion and independence from centralized authority.

Societal Dimension

In the context of societal dimensions, our objective is to develop a protocol that enhances user engagement and participation by focusing on incentive measures like security and data privacy. To attract and retain users, we aim to build a trust system by instituting robust security measures to protect user data. Emphasizing data privacy is essential, as it ensures that user information is protected, thereby fostering trust and confidence among users. This increased confidence, driven by strong security and privacy protocols, will encourage more users to participate and engage with the network. As user participation grows, the protocol's performance will improve due to a larger number of data points, enhanced network stability, and overall better functionality. Eventually, a larger, more engaged user base will lead to a practical and efficient protocol, demonstrating the significant societal impact of our approach.

Bibliography

[1] Broadband Commission for Sustainable Development. *The State of Broadband 2016: Broadband Catalyzing Sustainable Development*. International Telecommunication Union (ITU), United Nations Educational, Scientific, and Cultural Organization (UNESCO), 2016 (cited on page 1).

[2] Nessrine Chakchouk. 'A Survey on Opportunistic Routing in Wireless Communication Networks'. In: *IEEE Communications Surveys Tutorials* 17.4 (2015), pp. 2214–2241. DOI: 10.1109/COMST.2015.2411335 (cited on pages 3, 9).

[3] Amin Vahdat, David Becker, et al. *Epidemic routing for partially connected ad hoc networks*. 2000 (cited on pages 3, 12, 54, 57).

[4] Anders Lindgren, Avri Doria, and Olov Schelén. 'Probabilistic routing in intermittently connected networks'. In: *ACM SIGMOBILE mobile computing and communications review* 7.3 (2003), pp. 19–20 (cited on pages 3, 12, 54, 57).

[5] Gonzalo M. Vazquez-Prokopec et al. 'Using GPS Technology to Quantify Human Mobility, Dynamic Contacts and Infectious Disease Dynamics in a Resource-Poor Urban Environment'. In: *PLOS ONE* 8.4 (Apr. 2013), pp. 1–10. DOI: 10.1371/journal.pone.0058802 (cited on page 3).

[6] Li Liu. 'Interest-based prediction routing protocol in socially-aware opportunistic networks'. In: *2013 2nd International Symposium on Instrumentation and Measurement, Sensor Network and Automation (IMSNA)*. 2013, pp. 371–374. DOI: 10.1109/IMSNA.2013.6743293 (cited on pages 3, 15, 18).

[7] Douglas Teixeira et al. 'Revealing challenges in human mobility predictability'. In: *ACM Transactions on Spatial Algorithms and Systems* (2021) (cited on page 3).

[8] Douglas Teixeira, Mário Alvim, and Jussara Almeida. 'On the predictability of a user's next check-in using data from different social networks'. In: *Proc. of the 2nd ACM SIGSPATIAL Workshop on Prediction of Human Mobility*. 2018 (cited on page 3).

[9] Joon Heo et al. 'Spatial-data-driven student characterization in higher education'. In: *Proc. of the 1st ACM SIGSPATIAL Workshop on Prediction of Human Mobility (PredictGIS)*. 2017 (cited on page 3).

[10] Enrique Hernández-Orallo and Antonio Armero-Martínez. 'How Human Mobility Models Can Help to Deal with COVID-19'. In: *Electronics* 10.1 (2021). DOI: 10.3390/electronics10010033 (cited on page 3).

[11] Chiara Boldrini. 'Design and Analysis of Context-Aware Forwarding Protocols for Opportunistic Networks'. In: *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*. MobiOpp '10. Pisa, Italy: Association for Computing Machinery, 2010, pp. 201–202. DOI: 10.1145/1755743.1755788 (cited on page 3).

[12] Amee Trivedi. 'Human Mobility Monitoring using WiFi: Analysis, Modeling, and Applications'. PhD thesis. University of Massachusetts Amherst, USA, 2021 (cited on page 4).

[13] Jana Koteich, Christian Salim, and Nathalie Mitton. 'Image processing based data reduction technique in WVSN for smart agriculture'. In: *Computing* 105.12 (2023), pp. 2675–2698 (cited on page 7).

[14] Jana Koteich and Nathalie Mitton. 'Dataset Collection of Multi-Communication Technologies Monitored in Different Mobility Contexts'. In: *The 20th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE. 2024 (cited on page 7).

[15] Jana Koteich and Nathalie Mitton. 'Machine Learning Approach for Mobility Context Classification using Radio Beacons'. In: *2023 31st International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*. IEEE. 2023, pp. 1–8 (cited on page 7).

[16] Jana Koteich, Christian Salim, and Nathalie Mitton. 'Spatio-temporal data reduction technique in WVSN for smart agriculture'. In: *2022 18th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2022, pp. 345–350 (cited on page 7).

[17] Jana Koteich, Christian Salim, and Nathalie Mitton. 'Data reduction and frame rate adaptation in wvsn'. In: *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE. 2021, pp. 127–132 (cited on page 8).

[18] Jana Koteich and Nathalie Mitton. 'Radio Beacon Base Context Identification'. In: *CoRes 2024: 9èmes Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*. Saint-Briac-sur-Mer, France, May 2024 (cited on page 8).

[19] Jana Koteich and Nathalie Mitton. 'PILOT Dataset: A Collection of Multi-Communication Technologies in Different Mobility Contexts'. In: *CoRes 2023-8th Francophone Meeting on Protocol Design, Performance Evaluation and Communication Network Experimentation*. 2023 (cited on page 8).

[20] Lei You et al. 'A one-hop information based geographic routing protocol for delay tolerant MANETs'. In: *International Journal of Ad Hoc and Ubiquitous Computing* 20.2 (2015), pp. 107–122 (cited on pages 9, 10).

[21] Mohd-Yaseen Mir, Hengbing Zhu, and Chih-Lin Hu. 'Enhanced Geographic Routing with One-and Two-Hop Movement Information in Opportunistic Ad Hoc Networks'. In: *Future Internet* 14.7 (2022), p. 214 (cited on pages 9, 10).

[22] Michele Zorzi and RR Rao. 'Energy and latency performance of geographic random forwarding for ad hoc and sensor networks: multihop performance'. In: *IEEE Transactions on mobile computing* (2003), pp. 337–348 (cited on page 9).

[23] B Maria et al. 'Recent study of routing protocols in VANET: Survey and taxonomy'. In: *The 1st International Workshop on Vehicular Networks and Telematics (WVNT 13), Marrakech-Morocco*. 2013 (cited on page 10).

[24] Radityo Anggoro, I RoyyanaMuslim, and M Hatem AA. 'A Survey on non-DTN and DTN Geographic based Protocols for VANETS'. In: *3rd International Conference on Intelligent Systems and Image*. 2015 (cited on page 10).

[25] Mohamed Atef Ali Madni, Saeid Iranmanesh, and Raad Raad. 'DTN and Non-DTN routing protocols for inter-cubesat communications: A comprehensive survey'. In: *Electronics* 9.3 (2020), p. 482 (cited on page 10).

[26] Ahmed Mohamed Abdalla and Salem H Salamah. 'Performance comparison between delay-tolerant and non-delay-tolerant position-based routing protocols in VANETs'. In: *International Journal of Communications, Network and System Sciences* 15.1 (2022), pp. 1–14 (cited on page 10).

[27] Renu Dalal et al. 'Proliferation of opportunistic routing: A systematic review'. In: *IEEE access* 10 (2021), pp. 5855–5883 (cited on page 10).

[28] Christoph P Mayer and Oliver P Waldhorst. 'Routing in hybrid delay tolerant networks'. In: *Computer Communications* 48 (2014), pp. 44–55 (cited on page 10).

[29] Sanjit Biswas and Robert Morris. 'Opportunistic Routing in Multi-Hop Wireless Networks'. In: *SIGCOMM Comput. Commun. Rev.* 34.1 (2004), pp. 69–74. DOI: 10.1145/972374.972387 (cited on page 11).

[30] Jing Zuo et al. 'Cross-layer aided energy-efficient routing design for ad hoc networks'. In: *IEEE Communications Surveys & Tutorials* 17.3 (2015), pp. 1214–1238 (cited on page 11).

[31] Umer Fiaz Abbasi, Azlan Awang, and Nor Hisham Hamid. 'A Cross-Layer Opportunistic MAC/Routing protocol to improve reliability in WBAN'. In: *The 20th Asia-Pacific Conference on Communication (APCC2014)*. 2014, pp. 36–41. DOI: 10.1109/APCC.2014.7091601 (cited on page 11).

[32] Jing Zuo et al. 'Cross-layer aided energy-efficient opportunistic routing in ad hoc networks'. In: *IEEE Transactions on Communications* 62.2 (2014), pp. 522–535 (cited on page 11).

[33] Xin Xu et al. 'A Cross-Layer Optimized Opportunistic Routing Scheme for Loss-and-Delay Sensitive WSNs'. In: *Sensors* 18.5 (2018). DOI: 10.3390/s18051422 (cited on page 11).

[34] Ayman Khalil and Besma Zeddini. 'Cross-Layer Optimization for Enhanced IoT Connectivity: A Novel Routing Protocol for Opportunistic Networks'. In: *Future Internet* 16.6 (2024), p. 183 (cited on pages 11, 13).

[35] Sammou El Mastapha. 'A Hierarchical Routing Tree Topology Based on Probabilistic Selection of Cluster Head and Super Cluster Head to Overcome Inter-Cluster and Intra-Cluster Routing Problems in DTN Networks'. In: *Journal of Theoretical and Applied Information Technology* 101.3 (2023) (cited on pages 12–14).

[36] P Kumar. 'Routing Protocol using optimization algorithm in Delay Tolerant Network'. In: *Journal of Networking and Communication Systems (JNACS)* 4 (Apr. 2021), pp. 16–23. doi: 10.46253/jnacs.v4i2.a3 (cited on page 13).

[37] Junran Wang and Milena Radenkovic. 'Optimization of Energy Consumption in Delay-Tolerant Networks'. In: *arXiv preprint arXiv:2406.03580* (2024) (cited on page 13).

[38] Amin Vahdat, David Becker, et al. *Epidemic routing for partially connected ad hoc networks*. 2000 (cited on pages 13, 14).

[39] Tianxiang Li, Zhaoning Kong, and Lixia Zhang. 'Supporting delay tolerant networking: A comparative study of epidemic routing and ndn'. In: *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE. 2020, pp. 1–6 (cited on page 13).

[40] Zhenxin Feng and Kwan-Wu Chin. 'A unified study of epidemic routing protocols and their enhancements'. In: *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*. IEEE. 2012, pp. 1484–1493 (cited on page 13).

[41] Prachi Garg et al. 'Enhanced epidemic routing protocol in delay tolerant networks'. In: *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE. 2018, pp. 396–401 (cited on page 13).

[42] Evjola Spaho et al. 'Performance evaluation of energy consumption for different DTN routing protocols'. In: *Advances in Network-Based Information Systems: The 21st International Conference on Network-Based Information Systems (NBiS-2018)*. Springer. 2019, pp. 122–131 (cited on page 14).

[43] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. 'Spray and wait: an efficient routing scheme for intermittently connected mobile networks'. In: *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. 2005, pp. 252–259 (cited on pages 14, 54).

[44] John Burgess et al. 'MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks.' In: *Infocom*. Vol. 6. Barcelona, Spain. 2006 (cited on page 14).

[45] SAMMOU EL MASTAPHA. 'Performance Evaluation of DTN Routing Protocols on Map-Based Social Mobility Models for DTN Networks'. In: *Journal of Theoretical and Applied Information Technology* 101.22 (2023) (cited on page 15).

[46] Afra J. Mashhadi, Sonia Ben Mokhtar, and Licia Capra. 'Habit: Leveraging human mobility and social network for efficient content dissemination in Delay Tolerant Networks'. In: *2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops*. 2009, pp. 1–6. doi: 10.1109/WOWMOM.2009.5282467 (cited on page 15).

[47] Gurkan Solmaz, Mustafa Akbaş, and Damla Turgut. 'A Mobility Model of Theme Park Visitors'. In: *IEEE Transactions on Mobile Computing* 14 (Dec. 2015), pp. 2406–2418. doi: 10.1109/TMC.2015.2400454 (cited on page 16).

[48] Enrique Hernández-Orallo and Antonio Armero-Martínez. 'How Human Mobility Models Can Help to Deal with COVID-19'. In: *Electronics* 10.1 (2021). doi: 10.3390/electronics10010033 (cited on page 16).

[49] Tanweer Alam. 'Fuzzy control based mobility framework for evaluating mobility models in MANET of smart devices'. In: *Journal of Engineering and Applied Sciences* 12 (Aug. 2017), pp. 4526–4538 (cited on page 16).

[50] Jean-Gabriel Krieg et al. 'Unlocking the Smartphone's Sensors for Smart City Parking'. In: *Pervasive and Mobile Computing* 43 (2018). An early version of this work appeared in the Proc. of IEEE ICC 2016 (Krieg et al., 2016), pp. 78–95. doi: 10.1016/j.pmcj.2017.12.002 (cited on pages 16, 18).

[51] Sarfraz Nawaz, Christos Efstratiou, and Cecilia Mascolo. 'ParkSense: A Smartphone Based Sensing System for on-Street Parking'. In: *Proceedings of the 19th Annual International Conference on Mobile Computing amp; Networking*. New York, NY, USA: Association for Computing Machinery, 2013, pp. 75–86. doi: 10.1145/2500423.2500438 (cited on page 16).

[52] Marc Friesen et al. 'Vehicular Traffic Monitoring Using Bluetooth Scanning Over a Wireless Sensor Network'. In: *Canadian Journal of Electrical and Computer Engineering* 37.3 (2014) (cited on pages 17, 21, 22).

[53] Long Vu et al. 'Joint Bluetooth/Wifi Scanning Framework for Characterizing and Leveraging People Movement in University Campus'. In: MSWIM '10. Bodrum, Turkey: Association for Computing Machinery, 2010, pp. 257–265. DOI: 10.1145/1868521.1868563 (cited on pages 17, 22).

[54] Stephane Cedric Koumetio Tekouabou, Yassine Maleh, Anand Nayyar, et al. 'Towards to intelligent routing for DTN protocols using machine learning techniques'. In: *Simulation Modelling Practice and Theory* 117 (2022), p. 102475 (cited on page 17).

[55] Rachel Dudukovich and Christos Papachristou. 'Delay tolerant network routing as a machine learning classification problem'. In: *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE. 2018, pp. 96–103 (cited on page 17).

[56] Rahul Arun Paropkari, Anurag Thantharate, and Cory C. Beard. 'Deep-Mobility: A Deep Learning Approach for an Efficient and Reliable 5G Handover'. In: *CoRR* abs/2101.06558 (2021) (cited on page 18).

[57] Tanweer Alam. 'Fuzzy control based mobility framework for evaluating mobility models in MANET of smart devices'. In: *ARPN Journal of Engineering and Applied Sciences* (2017) (cited on page 18).

[58] Donald J. Patterson et al. 'Inferring High-Level Behavior from Low-Level Sensors'. In: *Proc. Ubiquitous Computing*. Ed. by Anind K. Dey, Albrecht Schmidt, and Joseph F. McCarthy. 2003 (cited on page 18).

[59] Lin Liao et al. 'Learning and inferring transportation routines'. In: *Artificial Intelligence* 171.5 (2007), pp. 311–331. DOI: https://doi.org/10.1016/j.artint.2007.01.006 (cited on page 18).

[60] Ka-Ho Lam, Chi-Chung Cheung, and Wah-Ching Lee. 'RSSI-Based LoRa Localization Systems for Large-Scale Indoor and Outdoor Environments'. In: *IEEE Transactions on Vehicular Technology* 68.12 (2019), pp. 11778–11791. DOI: 10.1109/TVT.2019.2940272 (cited on page 18).

[61] Pooyan Shams Farahsari et al. 'A Survey on Indoor Positioning Systems for IoT-Based Applications'. In: *IEEE Internet of Things Journal* 9.10 (2022), pp. 7680–7699. DOI: 10.1109/JIOT.2022.3149048 (cited on page 18).

[62] Priya Roy and Chandreyee Chowdhury. 'A survey on ubiquitous WiFi-based indoor localization system for smartphone users from implementation perspectives'. In: *CCF Transactions on Pervasive Computing and Interaction* 4.3 (2022), pp. 298–318 (cited on page 18).

[63] Muhammad Waqas et al. 'A comprehensive survey on mobility-aware D2D communications: Principles, practice and challenges'. In: *IEEE Communications Surveys & Tutorials* 22.3 (2019), pp. 1863–1886 (cited on page 18).

[64] Rafael L Costa et al. 'On building human-aware opportunistic communication strategies for cost-effective content delivery'. In: *Computer Communications* 192 (2022), pp. 106–118 (cited on page 18).

[65] Ivan O Nunes et al. 'Combining spatial and social awareness in D2D opportunistic routing'. In: *IEEE Communications Magazine* 56.1 (2018), pp. 128–135 (cited on page 18).

[66] Kanchana Thilakarathna et al. 'Design and analysis of an efficient friend-to-friend content dissemination system'. In: *IEEE Transactions on Mobile Computing* 16.3 (2016), pp. 702–715 (cited on page 18).

[67] Ruichun Gu et al. 'A Scalable and Virtualized Testbed for IoT Experiments'. In: *TRIDENTCOM*. 2018 (cited on page 21).

[68] Louiza Mansour and Samira Moussaoui. 'CDCP: Collaborative Data Collection Protocol in Vehicular Sensor Networks'. In: *Wireless Personnal Communications* 80.1 (2015). DOI: 10.1007/s11277-014-2000-z (cited on page 21).

[69] Michele Girolami, Fabio Mavilia, and Franca Delmastro. 'A bluetooth low energy dataset for the analysis of social interactions with commercial devices'. In: *Data in Brief* 32 (2020), p. 106102. DOI: https://doi.org/10.1016/j.dib.2020.106102 (cited on page 22).

[70] https://ieee-dataport.org/collections/crawdad (cited on page 22).

[71] https://www.kaggle.com/datasets/giantuji/UjiIndoorLoc (cited on page 22).

[72] Abd Elwahab Boualouache et al. 'A BLE-based data collection system for IoT'. In: *NTIC*. 2015 (cited on page 22).

[73] Cyril Cecchinel et al. 'An architecture to support the collection of big data in the internet of things'. In: *IEEE World congress on services*. 2014 (cited on page 22).

[74] Nina Santi et al. 'Automated and Reproducible Application Traces Generation for IoT Applications'. In: *Q2SWinet 2021*. Alicante, Spain (cited on page 22).

[75] Omesh A Fernando, Hannan Xiao, and Joseph Spring. 'Developing a Testbed with P4 to Generate Datasets for the Analysis of 5G-MEC Security'. In: *IEEE WCNC*. Austin, USA, 2022. DOI: 10.1109/WCNC51071.2022.9771755 (cited on page 22).

[76] Kaixuan Gao et al. 'Toward 5G NR High-Precision Indoor Positioning via Channel Frequency Response: A New Paradigm and Dataset Generation Method'. In: *IEEE Journal on Selected Areas in Communications* 40.7 (2022), pp. 2233–2247. DOI: 10.1109/JSAC.2022.3157397 (cited on page 22).

[77] Ahmed Alkhateeb et al. 'DeepSense 6G: A large-scale real-world multi-modal sensing and communication dataset'. In: *IEEE Comm. Magazine* (2023) (cited on page 22).

[78] Kalkidan Gebru. 'A Privacy-preserving Scheme for Passive Monitoring of People's Flows through WiFi Beacons'. In: *IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*. 2022 (cited on page 22).

[79] Kalkidan Gebru et al. 'Edge-based passive crowd monitoring through WiFi Beacons'. In: *Computer Communications* 192 (2022), pp. 163–170. DOI: https://doi.org/10.1016/j.comcom.2022.06.003 (cited on page 22).

[80] Ashish Rajeshwar Kulkarni, Narendra Kumar, and K. Ramachandra Rao. 'Efficacy of Bluetooth-Based Data Collection for Road Traffic Analysis and Visualization Using Big Data Analytics'. In: *Big Data Mining and Analytics* 6.2 (2023), pp. 139–153. DOI: 10.26599/BDMA.2022.9020039 (cited on page 23).

[81] https://docs.pycom.io/datasheets/development/fipy/ (cited on pages 24, 25).

[82] https://docs.pycom.io/datasheets/expansionboards/pytrack2/ (cited on page 25).

[83] https://docs.pycom.io/datasheets/expansionboards/pysense2/ (cited on page 25).

[84] https://lora-alliance.org/ (cited on page 26).

[85] https://ec.europa.eu/info/law/law-topic/data-protection/reform/ (cited on page 29).

[86] https://www.rfc-editor.org/rfc/rfc3874.txt (cited on page 30).

[87] Taehwa Choi, Yohan Chon, and Hojung Cha. 'Energy-efficient WiFi scanning for localization'. In: *Pervasive and Mobile Computing* 37 (2017), pp. 124–138. DOI: https://doi.org/10.1016/j.pmcj.2016.07.005 (cited on page 33).

[88] Ari Keränen, Jörg Ott, and Teemu Kärkkäinen. 'The ONE simulator for DTN protocol evaluation'. In: *Proceedings of the 2nd international conference on simulation tools and techniques*. 2009, pp. 1–10 (cited on page 54).

[89] https://en.wikipedia.org/wiki/Confidence_interval (cited on page 56).

[90] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. 'Spray and wait: an efficient routing scheme for intermittently connected mobile networks'. In: *Proceedings of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking*. 2005, pp. 252–259 (cited on page 57).