Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2018

Rui Filipe Marques de Oliveira

Forwarding Strategies for Opportunistic Data Gathering in Mobile IoT

Estratégias de Encaminhamento para Recolha Oportunística de Informação em Redes Móveis de Internet das Coisas

Universidade de Aveiro Departamento de Electrónica, Telecomunicações e Informática,
2018

Rui Filipe Marques de Oliveira

Forwarding Strategies for Opportunistic Data Gathering in Mobile IoT

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requesitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica da Professora Doutora Susana Sargento, Professora Associada com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro e co-orientação científica do Doutor Miguel Luís, Investigador Auxiliar do Instituto de Telecomunicações de Aveiro.

o júri / the jury

presidente / president	**Professor Doutor António Luís Jesus Teixeira** Professor Associado com Agregação da Universidade de Aveiro (por delegação da Reitora da Universidade de Aveiro)
vogais / examiners committee	**Professor Doutor João Paulo da Silva Machado Garcia Vilela** Professor Auxiliar da Universidade de Coimbra - Faculdade de Ciências e Tecnologia
	Doutor Nuno Miguel Abreu Luís Investigador Auxiliar, Instituto de Telecomunicações de Aveiro (Co-Orientador)

agradecimentos / acknowledgements

É com muito prazer que agradeço às pessoas que diretamente ou inderetamente me ajudaram no meu percurso académico, rematado pela realização desta dissertação.

Gostaria de agradecer à minha família, que tanto contribuíram para a minha educação. Neste grupo, é de louvar o meu Pai, António Oliveira, por todo o esforço e dedicação. Não posso deixar de frisar que foi um esforço conjunto. Aos avós, quer paternos quer maternos, agradeço pelos ensinamentos e carinho. De forma mais geral, agradeço também a todo o resto da família pelo apoio.

Pela estabilidade e segurança, quero agradecer à minha namorada Catarina. É de notar o apoio incondicional dado, a capacidade de me fazer esquecer o trabalho, e principalmente a capacidade de me ouvir.

O percurso académico foi marcado também por imensas personalidades que merecem agradecimentos, nomeadamento: Louro, António, Figueiredo, Portela, Pika, Jomi, Samuel, Pi, Pedro, Enes, Carretos, Resende, Jorge, Tixa e Indiano (não está ordenado, o Pika é sempre o favorito).

Aos colegas de laboratório do NAP, um agradecimento pelo conhecimento partilhado e pela ajuda nos testes realizados. Um especial agradecimento ao Rodrigo pela integração no trabalho previamente desenvolvido e pela disponibilidade demonstrada.

Ao Doutor Miguel Luís, um especial agradecimento pelo acompanhamento em todos os passos do processo, nomeadamente na discussão científica e na maneira de expor os resultados obtidos. Prontidão será talvez a palavra mais adequada.

Um especial agradecimento à professora Susana Sargento pela introdução à área das redes e integração neste grupo, pela oportunidade e pela orientação ao longo desta dissertação.

Agradeço à Fundação Portuguesa para a Ciência e Tecnologia pelo suporte financeiro através de fundos nacionais e quando aplicável cofinanciado pelo FEDER, no âmbito do Acordo de Parceria PT2020 pelo projecto MobiWise através do programa Operacional Competitividade e Internacionalização (COMPETE 2020) do Portugal 2020 (POCI-01-0145-FEDER-016426).

Resumo

A elevada mobilidade em cenários veiculares urbanos origina descontinuidades de comunicação entre veículos, um fator altamente importante quando se desenha uma estratégia de encaminhamento para redes veiculares. Mecanismos de *store, carry and forward* (guardar, carregar e entregar) possibilitam a recolha de dados de sensores em aplicações da Internet das Coisas, contribuindo para plataformas de cidades inteligentes.

Este trabalho é focado em dois tópicos principais de forma a melhorar a decisão de encaminhamento: i) estratégias de encaminhamento que fazem uso de métricas sociais e de localização para efetuar a seleção de vizinhos, ii) e mecanismos de seleção de pacotes que qualificam a rede com qualidade de serviço. A seleção de vizinhos é feita através de múltiplas métricas, resultando em três estratégias de encaminhamento: Gateway Location Awareness (GLA), uma classificação baseada em localização que faz uso de velocidade, ângulo de direção e distância até uma *gateway*, para selecionar os veículos com maior probabilidade de entregar a informação num menor período temporal, distinguindo os veículos através dos seus padrões de movimento. Aging Social-Aware Ranking (ASAR) explora os comportamentos sociais de cada veículo, onde é atribuída uma classificação aos veículos com base num histórico de contactos, diferenciando veículos com um alto número de contactos de outros com menos. Por fim, por forma a tirar partido das distintas características de cada uma das destas estratégias, é proposta uma abordagem híbrida, Hybrid between GLA and ASAR (HYBRID).

Aliado ao critério de encaminhamento, são propostos dois mecanismos de seleção de pacotes que focam distintas funcionalidades na rede, sendo estes: Distributed Packet Selection, que foca em primeiro lugar na prioritização de determinados tipos de pacotes e em segundo lugar, no tempo de vida que resta ao pacote na rede; e Equalized Packet Selection, que usa métricas da rede para calcular a classificação de cada pacote em memória. Para tal, é usado o número de saltos do pacote, o tipo de dados do pacote e o tempo de vida que resta ao pacote na rede.

De forma a avaliar os mecanismos propostos, foram realizadas experiências em emulador e em cenário real. Para cada estratégia de encaminhamento, é avaliada a influência de vários parâmetros de configuração no desempenho da rede. Para além disso, é feita uma avaliação comparativa entre as várias estratégias em diferentes cenários. Resultados experimentais, obtidos usando traços reais de mobilidade e conetividade de uma rede veicular urbana, são utilizados para avaliar a performance dos esquemas GLA, ASAR e HYBRID. Posteriormente, a viabilidade destas estratégias é também validada em cenário real. Os resultados obtidos mostram que estas estratégias são um bom *tradeoff* para maximizar a taxa de entrega de dados e minimizar a sobrecarga de dados na rede. Para avaliar os mecanismos de seleção de pacotes, um simples mecanismo *First In First Out* é utilizado como base, contrapondo com as técnicas propostas mais orientadas a objectivos concretos. Os resultados obtidos mostram que os mecanismos propostos são capazes de proporcionar à rede diferentes funcionalidades, desde prioritização de determinado tipos de dados a melhoramentos no desempenho da rede.

Abstract

High vehicular mobility in urban scenarios originates inter-vehicles communication discontinuities, a highly important factor when designing a forwarding strategy for vehicular networks. Store, carry and forward mechanisms enable the usage of vehicular networks in a large set of applications, such as sensor data collection in IoT, contributing to smart city platforms.

This work focuses on two main topics to enhance the forwarding decision: i) forwarding strategies that make use of location-aware and social-based to perform neighborhood selection, ii) and packet selection mechanisms to provide Quality of Service (QoS).

The neighborhood selection is performed through multiple metrics, resulting in three forwarding strategies: (1) Gateway Location Awareness (GLA), a location-aware ranking classification making use of velocity, heading angle and distance to the gateway, to select the vehicles with higher chance to deliver the information in a shorter period of time, thus differentiating nodes through their movement patterns; (2) Aging Social-Aware Ranking (ASAR) that exploits the social behaviours of each vehicle, where nodes are ranked based on a historical contact table, differentiating vehicles with a high number of contacts from those who barely contact with other vehicles; (3) and to merge both location and social aforementioned algorithms, a hybrid approach emerges, thus generating a more intelligent mechanism.

Allied to the forwarding criteria, two packet selection mechanisms are proposed to address distinct network functionalities, namely: Distributed Packet Selection, that focuses primarily on data type prioritization and secondly, on packet network lifetime; and Equalized Packet Selection, which uses network metrics to calculate a storage packet ranking. To do so, the packet number of hops, the packet type and packet network lifetime are used.

In order to perform the evaluation of the proposed mechanisms, both real and emulation experiments were performed. For each forwarding strategy, it is evaluated the influence of several parameters in the network's performance, as well as comparatively evaluate the strategies in different scenarios. Experiment results, obtained with real traces of both mobility and vehicular connectivity from a real city-scale urban vehicular network, are used to evaluate the performance of GLA, ASAR and HYBRID schemes, and their results are compared to lower- and upper-bounds. Later, these strategies' viability is also validated in a real scenario. The obtained results show that these strategies are a good tradeoff to maximize data delivery ratio and minimize network overhead, while making use of moving networks as a smart city network infrastructure. To evaluate the proposed packet selection mechanisms, a First In First Out packet selection technique is used as ground rule, thus contrasting with the more objective driven proposed techniques. The results show that the proposed mechanisms are capable of provide distinct network functionalities, from prioritizing a packet type to enhancing the network's performance.

Contents

List of Figures

List of Tables

Acronyms

6LoWPAN	IPv6 over Low power Wireless Personal Area Networks
ACK	Acknowledgement
ASAR	Aging Social-Aware Ranking
BLE	Bluetooth Low Energy
DCU	Data Collecting Unit
DTN	Delay Tolerant Netowork
FIFO	First In First Out
GLA	Gateway Location Awareness
GPS	Global Position System
HYBRID	HYBRID between GLA and ASAR
I_2C	Inter-Integrated Circuit
IEEE	Institute of Electrical and Eletronics Engineers
IoT	Internet of Things
IP	Internet Protocol
IPC	Inter-Process Communication
LoRa	Long Range
LPWAN	Low Power Wide Area Network
M2M	Machine-to-Machine
M2P	Machine-to-Person
MAC	Medium Access Control
mOVE	mobile Opportunistic VEhicular
mOVERS	mobile Opportunistic Vehicular Emulator for Real Scenarios
NFC	Near-Field-Communications
OBU	On-Boart Unit
PCB	Printed Circuit Board
QoS	Quality of Service
RFID	Radio-Frequency IDentification
RSU	Road Side Unit
RX	Receiver
TCP	Transmission Control Protocol
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol
V2I	Vehicle-to-Infrastructure

V2V Vehicle-to-Vehicle
VANET Vehicular Ad-hoc NETwork

Chapter 1

Introduction

1.1 Context and Motivation

The IoT paradigm refers to a network of interconnected things. These devices may have sensors and/or actuators, be capable of communicating with other devices, have storage and process units. IoT aims to provide Machine-to-Machine (M2M) and Machine-to-Person (M2P) communications on a massive scale [1]. According to multiple studies [2, 3, 4], it is expected a significant increase in both the total number of devices connected and revenues. For example, Figure 1.1 predicts the evolution of the number of connected devices until 2021, regarding the different sectors.

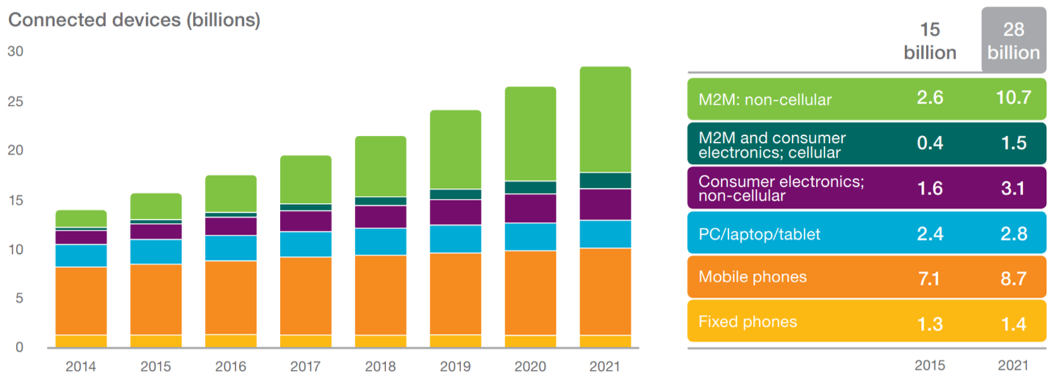


Figure 1.1: Growth in connected devices [5].

The possibilities of applications in IoT platforms are enormous. Different application areas have specific requirements and considerations [6, 2]. With hindsight, a wide variety of technologies must be considered. Extremely short-range systems (Near-Field-Communications (NFC)), short-range passive and active radio frequency (Radio-Frequency IDentification (RFID)) systems, systems based on the family of Institute of Electrical and Eletronics Engineers (IEEE) 802.15.4 standards (ZigBee, IPv6 over Low power Wireless Personal Area Networks (6LoWPAN)), Bluetooth-based systems (including Bluetooth Low Energy (BLE)), Proprietary systems (Z-Wave, CSRMesh, EnOcean), systems based on IEEE 802.11/Wi-Fi and cellular networks are options for IoT applications [7, 8]. Supporting wide urban coverage applications, a set of technologies named Low Power Wide Area Networks (LPWANs) are

IoT enablers, such as: NB-IoT[1], Long Range (LoRa)[2], SigFox[3], Ingenu[4], Weigthless[5].

Considering the requirement of IoT platforms, a wide spectrum of applications arise [2, 9]. In Massive IoT applications - typically sensors that report to the cloud on a regular basis - the end-to-end cost must be low enough for the business case to make sense. The range of applications can be seen in Figure 1.2.

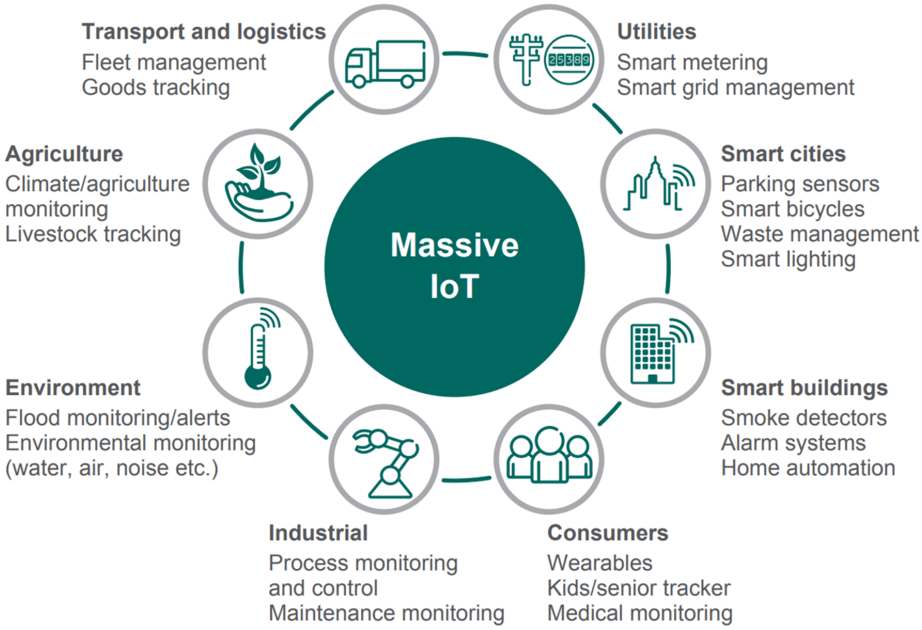


Figure 1.2: Differing requirement for Massive and Critical IoT applications [2].

To enable these future applications, vehicles are expected to be equipped with communication devices, mostly connected with smart city platforms [10]. Performing the connection bridge between vehicles, buildings and other devices embedded with software, sensors and actuators will enable a new myriad of applications in IoT and Vehicular Ad-hoc NETworks (VANETs). As expected, these devices will generate huge amounts of traffic, thus consuming a substantial portion of the communication resources [11]. Generally speaking, data traffic in smart cities can be categorized according to two taxonomies: delay-tolerant traffic and delay-sensitive traffic [12]. Thus, an opportunity emerges through the exploitation of the vehicles' mobility and also the store, carry and forward mechanisms to offload delay-tolerant traffic from existing telecommunication networks, not only benefiting the delay-sensitive data flows through network congestion relief but also enabling smart city applications [13].

Vehicular networks comprise a group of nodes that communicate with each other despite the lack of a fixed infrastructure support. Given the challenging propagation channels in high-mobility VANETs, the information in these networks must be transmitted to destinations through multi-hop wireless communications via intermediate vehicles [14]. Therefore, these

[1] http://www.3gpp.org/news-events/3gpp-news/1785-nb_iot_complete → Accessed: 26-01-2018
[2] https://www.lora-alliance.org/ → Accessed: 26-01-2018
[3] https://www.sigfox.com/en → Accessed: 26-01-2018
[4] https://www.ingenu.com/ → Accessed: 26-01-2018
[5] https://http://www.weightless.org/about/what-is-weightless → Accessed: 26-01-2018

networks must be able to deal with intermittent connectivity, typically presenting long and non-constant delays due to the unpredictable mobility. Vehicles equipped with On-Boart Units (OBUs) are able to provide Vehicle-to-Vehicle (V2V) communications, and are also able to connect to Road Side Units (RSUs) by Vehicle-to-Infrastructure (V2I) communications. Even though urban scenarios can be challenging from the signal propagation point of view, due to the high density of buildings, vehicles and other objects, the high number of vehicles, their social behaviours and their respective mobility patterns can be exploited to overcome such difficulties [15, 16].

1.2 Objectives

The main goal of this dissertation is to improve an under-development architecture and enhance its data gathering mechanism to cope with Smart City applications in IoT platforms. Therefore, this dissertation has the following objectives:

- Perform the first steps (for this architecture) in the energy expenditure analysis for some of its elements, and provide a solution based on a lower energy consumption technology.
- Study the state of the art on forwarding strategies through Delay Tolerant Netoworks (DTNs), and understand which are the more relevant metrics to evaluate forwarding strategies.
- Design and implement forwarding strategies with location-aware and social-based metrics and network management mechanisms in a DTN.
- Design and implement packet selection mechanisms to address distinct scenarios presented by Smart City applications.
- Evaluate the forwarding strategies and packet selection mechanisms through real and emulated mobile experiments.

1.3 Contributions

The work developed in this dissertation led to the following contributions:

- Extend the functionalities to the under-development architecture.
- Conclusions about the viability of neighboring discovery through WiFi scan versus BLE scan.
- Proposal of new forwarding strategies for vehicular networks' data gathering, and conclusions about the proposed approaches in emulated and real environments.
- Conclusions about the packet selection mechanisms through emulation.
- Validation and verification of the feasibility of the proposed architecture as a real deployment.

The work developed in this thesis, namely the proposed forwarding strategies, was submitted to the IEEE Transactions on Vehicular Technology Journal (IF 4.432), on the Connected Vehicles Series, currently under evaluation. Preliminary work has also been published on Sensors Open Access Journal, MDPI Sensors, on April 2018 [17] (as a second author).

1.4 Document Structure

The remainder of this thesis is organized as follows:

- *Chapter 2 - State of the Art* - presents the related work, where several forwarding mechanisms are exposed and compared;
- *Chapter 3 - Proposed Architecture* - overviews the network architecture and the network components. It also presents the analytic models for the proposed forwarding strategies and packet selection mechanisms;
- *Chapter 4 - Implementation and Integration* - describes the software work-flows that were necessary to the implementation of the proposed work;
- *Chapter 5 - Evaluation* - evaluates the implemented solution through real and emulated experiments;
- *Chapter 6 - Conclusions and Future Work* - presents the conclusions and directions for future work.

Chapter 2

State of the art

2.1 Introduction

This chapter aims to provide the reader with the fundamental concepts required to understand the work presented in this dissertation. In addition, related work already developed by the scientific community is exhibited.

Section 2.2 presents the concept of DTN. A brief overview and definition are presented. Moreover, it also presents the architecture implementation used as baseline for this work, and a summary of the existing routing and forwarding strategies that fit the DTN concept.

Section 2.3 presents use cases of Smart Cities' deployments that are in related with data gathering or with the above mentioned subjects.

2.2 Delay Tolerant Networks

2.2.1 Overview

Delay tolerant networks (DTNs) are mobile and ad-hoc wireless networks characterized by intermittent connections between the nodes [18], high node mobility and short radio range, very high delivery delays and lack of end-to-end paths and stable infrastructure, and the possibility of environmental interferences [19].

In the IoT paradigm, enabling delay tolerant communications will allow smart objects to better communicate even in the presence of disruption in their connectivity [20]. However, DTN schemes have to be tailored to IoT applications to fit their specific requirements, such as: heterogeneity, huge amount of exchanged messages, information-centric based protocols and intermittent connections.

The underlying principle behind DTN is the ability to route data in a store-carry and forward, in which intermediate mobile nodes store data to be transmitted until they find an appropriate relay node (to forward the message) in the path towards the destination (Figure 2.1).

The DTN particular properties led to a wide number of routing protocols. In a DTN, three indicators are used to evaluate a routing algorithm performance: delivery ratio, latency, delivery cost (network overhead) [22]:

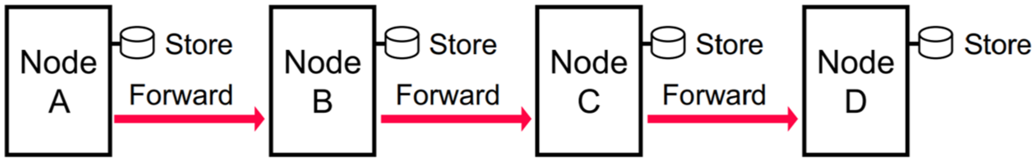


Figure 2.1: Store-Carry-and-Forward mechanism [21].

- **Delivery ratio** is the number of messages delivered divided by the total number of messages generated.
- **Latency or delay** is the total amount of time elapsed between the creation of a message and its respective delivery.
- **Delivery cost** refers to the amount of data used to deliver the generated information.

In order to implement the conceptual idea of a DTN, software architectures implement distinct functionalities. DTN2 [23], IBR-DTN [24] and mOVE are examples of practical implementations of a DTN. DTN2 is the base implementation of the Bundle Protocol [25]. The main architectural modules are: Bundle Router and Bundle Forwarder, Convergence Layers, Persistent Storage, Fragmentation Module and Contact Manager. IBR-DTN was developed for embedded systems and works in OpenWRT[1]. The main architectural modules are: Event Switch, Discovery Agent, Connection Manager, Base Router, Bundle Storage, Wall Clock and IBR-DTN API. The mOVE architecture will be addressed in further detail in the next section.

2.2.2 Mobile Opportunistic VEhicular (mOVE)

mobile Opportunistic VEhicular (mOVE) [26] is a DTN architecture developed in Network Architectures and Protocols (NAP) research group [27], using WiFi technology IEEE 802.11a/b/g to communicate. This software was designed to be highly modular and extensible. In [28] further developments were done.
In order to implement a DTN, different modules compose this architecture. These modules are illustrated in Figure 2.2, which are: Neighbouring, API Management, Storage, Socket, Receiver (RX), Routing, DCU Listener, Sensor Listener and LoRa Socket.

Neighbouring This module is responsible for listening to the media searching from other neighbour nodes. Each node advertises its presence and updates its internal neighboring tables. It can operate with different types of neighbouring nodes depending on the communication interface that is used.

Socket This module is a User Datagram Protocol (UDP) socket that provides an abstraction layer to send/receive packets to/from neighbouring nodes.

[1]https://openwrt.org/ → Accessed: 29-10-2018

RX (Receiver) This module is constantly checking if any data was received in the UDP socket. A classification is made based on internal parameters of these packets. This module, depending on the packet type, either neighbouring control packets or data packets, forwards them to the neighbouring and routing modules, respectively. The RX module also enables an unicast or broadcast transfer of information.

API Management A mOVE node interacts with external applications through this module. In detail, it uses UNIX sockets Datagram Communications (connectionless) to manage data and control messages between mOVE applications and mOVE.

Storage This module is responsible for storing several packets and other information that is relevant for the forwarding decision. Developed in a robust way, it is able to deal with unexpected power outages. Also, it allows a binary storage of network packets and performs fast queries to accommodate routing decisions.

Routing This module makes decisions, based on the routing strategy, of which packets should be sent, in which order, to which neighbors. Some metrics are considered such as: maximization of the delivery of useful information to its destination and the sent information during transfer windows, minimization of the CPU consumption, and balancing of the load between nodes. Furthermore, it aims to minimize the replicas in the network and the packets maintained in Storage.

DCU Listener Connection socket established when a vehicle finds a DCU creating a communication channel between the DCU and the respecting vehicle.

Sensor Listener Socket responsible to forward sensor data collected by the vehicle to the mOVE platform.

LoRa Socket Socket responsible to forward desired data packets from the mOVE platform to be delivered through the LoRa technology.

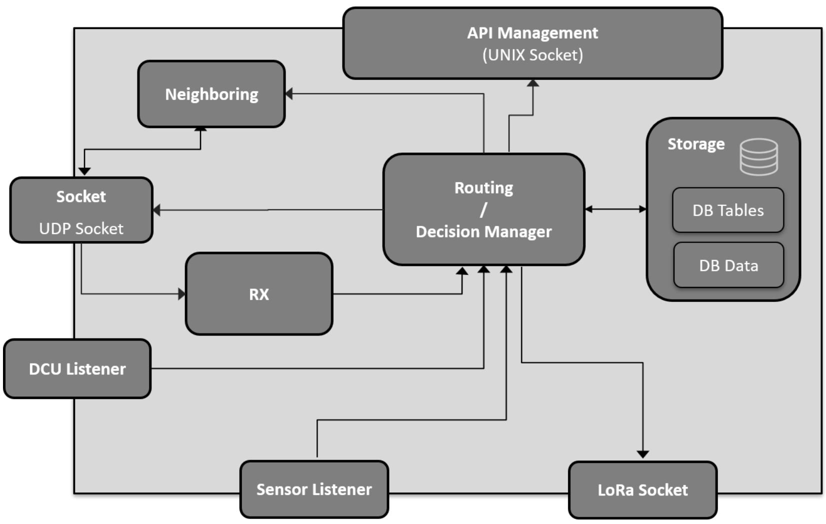


Figure 2.2: mOVE architecture [28].

2.2.3 Routing Strategies

The forwarding strategies that comprise the state of the art for this thesis were selected considering taxonomies of interest to the development of this work, namely: (1) Single-copy routing, where packets only have one hop available until reaching a gateway; (2) Flooding mechanisms, where packets are widely spread in the network since every connection is used to replicate packets (controlled flooding mechanisms are also explored); (3) Probabilistic forwarding explores historical encounters to calculate the probability of neighboring viability; (4) Geographic routing uses location mechanisms to perform forwarding decisions, where mobility and movement prediction schemes make use of geolocation to predict and anticipate events; (5) Social forwarding explores community patterns to perform the routing decisions; (6) Opportunistic routing makes the decision on the move, where connectivity metrics are used in the forwarding decision.

Single-copy Routing

First Contact

First Contact [29] is a single-copy routing protocol which aims to minimize the usage of bandwidth and resources (e.g., energy, storage). In First Contact, a message is forwarded along an edge chosen randomly among all the current contacts. This results in a random search for the destination node. Therefore, messages may be delivered to a dead end. Furthermore, the delay is high and the delivery capacity is very low.

Direct Delivery

Direct delivery [30] is a single copy routing protocol where there is no need of knowledge about the network to make forwarding decisions. The source node carries a message until it meets its final destination. The use of bandwidth and resources is minimum, but the delivery capacity is very low and the delay is very high (in specific scenarios, infinite).

Flooding

Epidemic Routing Protocol

The Epidemic protocol [31] is a multi-copy protocol. The basic operation is simple: data source nodes and intermediate nodes flood messages to all their neighbours to mitigate the effects of a single path failure in order to maximize the possibility that the message may arrive to the destination node. Messages are quickly distributed through the neighbourhood, but significant resources from the network and nodes are wasted in this process. Therefore, no previous knowledge of the network is required. This approach can achieve high delivery ratios, low latency and very high overhead. This can be considered the optimal solution in an environment with no buffer space limitations, power spent and bandwidth boundaries. Clearly, this protocol spoils storage, energy and bandwidth in comparison with other protocols [12].

In order to avoid replication of messages already in the buffers of nodes, summary vectors are exchanged between nodes (Figure 2.3) and it is maintained a list of nodes that recently established contact. Also, a hop count is used to prevent infinite message replication. Messages with hop count equal to one will only be forwarded to the destination node.

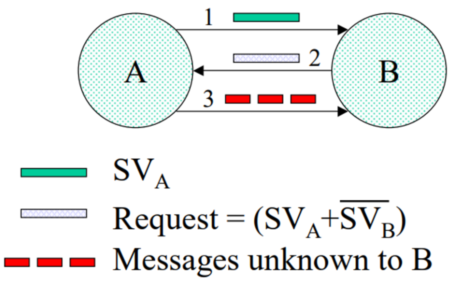


Figure 2.3: Message exchange in the Epidemic Routing Protocol [31].

MaxProp

MaxProp protocol [32] addresses scenarios in which either transfer duration or storage is a limited resource in the network, trying to increase the delivery rate and lowering the latency of packets. The MaxProp protocol principle is based on a ranked list of the peers' stored packets. The list is built through a cost (an estimate of delivery likelihood) assigned to each destination. Moreover, MaxProp deals with the problem of scheduling packets for transmission to other peers and deleting packets when buffers are low on space recurring to multiple mechanisms, illustrated in Figure 2.4. In addition, MaxProp uses acknowledgements sent to all peers in order to notify them of packet delivery. Furthermore, the transmission of messages to other hosts is done in a specific order, which takes into account message hop counts and message delivery probabilities of previous encounters.

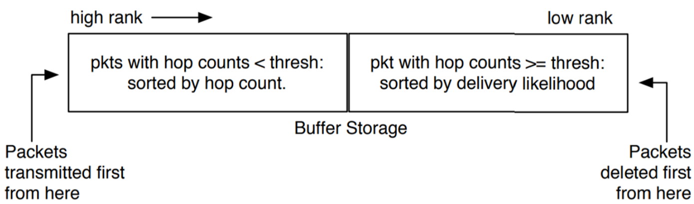


Figure 2.4: MaxProp routing strategy [32].

Spray and Wait

Spray and Wait [33] is a zero-knowledge routing protocol that reduces flooding of redundant messages in a DTN. Is it accomplished by limiting the number L of bundle copies created per bundle. The Spray and Wait protocol is composed of two phases, the spray phase and the wait phase:

- Spray phase: L message copies are initially spread to L distinct intermediate nodes, either by the source or other nodes receiving a copy.
- Wait phase: In case the destination is not reached in the spraying phase, each of the L nodes carrying a message copy is restricted to only forward the message to the destination (direct transmission).

Moreover, two spraying modes are proposed, the normal spray mode and the binary spray mode:

- Normal spray mode: Source node forwards one of the L copies to each encountered node.
- Binary spray mode: Source hands over $\frac{L}{2}$ and keeps $\frac{L}{2}$ for itself when it encounters another node with no copies. When one node is left with only one copy, it switches to direct transmission.

Since the number of message copies L is a parameter that can be adjusted, the number of copies will increase with the size of the network. Therefore, the network scalability is adaptable.

Probability based forwarding

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

The PROPHET protocol [34, 35] estimates the delivery probability based on the history of encounters. A metric called Delivery Predictability, $P_{(a,b)} \in [0,1]$, is calculated for every node a for each known destination b. Suppose that a node a has a message m for the destination d. When a contact occurs between a pair of nodes a and b, node a forwards the message m to node b only if b has a greater Delivery Predictability to the destination d, that is, $P_{(a,d)} < P_{(b,d)}$. During the contact, in addition to the exchange of messages, the Delivery Predictability for each node can be updated.

The Delivery Predictability calculation is divided in three parts. When two nodes meet each other, they immediately update the Delivery Predictability as shown in (2.1).

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) \times P_{init} \tag{2.1}$$

where $P_{init} \in [0,1]$ is an initialization constant.

If, for a period of time, a pair of nodes does not encounter each other, then the Delivery Predictability metric is updated by the nodes as shown in (2.2).

$$P_{(a,b)} = P_{(a,b)old} \times \gamma^k \tag{2.2}$$

where $\gamma \in [0,1]$ is an aging constant, and k is the number of time slots elapsed since the metric was updated for the last time.

If node a frequently encounters node b, and node b frequently encounters node c, the transitive property is applied. Therefore, node c presents itself as a good choice to forward messages to node a. The transitive property affects the Delivery Predictability as shown in (2.3).

where $\beta \in [0,1]$ is a constant that determines the impact of transitivity on the Delivery Predictability.

$$P_{(a,c)} = P_{(a,c)old} + (1 - P_{(a,c)old}) \times P_{(a,b)} \times P_{(b,c)} \times \beta \quad (2.3)$$

NECTAR

The NECTAR protocol [36] uses the occurrence of an opportunistic contact to calculate a Neighbourhood Index and spread messages in a controlled manner. The functions of NECTAR protocol can be summarized in three procedures: Neighbourhood Index calculation, Message Scheduling Algorithm and Message Discard Policy.

- **Neighbourhood Index calculation:** each node keeps a neighbourhood index table, which stores the information about the meeting frequency of every inter-node in the network. The node with higher meeting frequency will be assigned a higher index value. When a node needs to forward the message to a particular destination, it selects from the relay nodes the one that has the highest index value for the respective destination.

  In practice, the neighbourhood index for a particular node is calculated based on the time that this node and the neighbour node remain in contact, a distance metric and an aging metric. But, in case another node has a better neighbourhood index to the destination, the neighbourhood index table is updated in a weighted fashion.

- **Message Scheduling Algorithm:** this procedure determines the delivery priority of each message in the storage area. Messages whose destination is the current contact node are sent first. To avoid network congestion, messages have a time-to-leave field which is decreased by one in each intermediate node. After this, nodes update their respective Neighbourhood index.

  During a specified time, the node that delivered the message to the destination keeps a copy of the header message in a different storage area to avoid receiving redundant copies of the same message.

  Furthermore, the NECTAR protocol can operate as the Epidemic Protocol, but with limitations on the maximum and minimum number of packets to flood the network.

- **Message Discard Policy:** this procedure is based on the number of replicas and the number of time slots elapsed since the receipt of a message from a specific node. This information is used to perform the message aging index calculation. When the storage area is fully occupied, message discard based on the aging index occurs.

EASE (Exponential Age Search)

EASE protocol [37] is a last encounter routing (LER) algorithm (i.e., it computes routes purely based on last encounters), which translates into no energy spent to explicitly diffuse location information or maintain a view of the network topology.

The basic idea behind EASE is for a node to follow the trajectory to the destination between and in "jumps" of decreasing length. The end-points of such jumps are called anchors. No particular routing algorithm is defined when a packet goes from one anchor to the next, leaving space for any position-based routing algorithm. Thereafter, the EASE algorithm operates in two alternating phases. In the first phase, when a packet has reached an anchor, it performs a local search around that anchor to find the next anchor. In the second phase, an existing position-based routing algorithm is used to route the packet towards the new anchor.

However, it is clear that EASE ignores a lot of potentially useful information. Therefore, some modifications were made to the algorithm, resulting into GREASE (Greedy EASE) that checks the age of the last encounter with the destination at each hop. If it encounters a node that has a more recent estimate of the destinations location than the anchor is currently headed to, then it becomes the new anchor.

HPR (Hybrid of Probability and message Redundancy)

The HPR algorithm [35] implements parallel message transfer in multi-path mode in order to improve the delivery ratio of a massage. In this process, a maximum number of copies is established per source node. These messages are generated based on a binary tree which aims to reduce the overhead. Moreover, to minimize the overhead of the network, this algorithm adopts the nodes delivery probability value as the basis to forward the message, meaning that the direction in which the delivery probability value is increasing is the eligible direction to forward the message.

Geographic based forwarding

GeoSpray

The GeoSpray routing protocol [38, 39] assumes that the nodes are aware of their location (geographical position), provided by a positioning device like a Global Position System (GPS) navigation system.

Since multi-copy routing schemes are noted for high delivery ratios, low bundle delivery delays, but high overheads due to duplicated copies, GeoSpray considers the replication approach of the Spray and Wait protocol to limit the amount of duplicated copies. In this process, a small/fixed number of bundle copies is distributed to distinct nodes in the network. However, GeoSpray makes further considerations in comparison to Spray and Wait protocol: instead of doing blind replication, it guarantees that bundle copies are only spread to network nodes that go closer (and/or arrive sooner) to the bundle destination; and instead of waiting until one of these network nodes meets the destination and delivers its bundle copy in the waiting phase, it allows each node to forward its bundle copy further to another node that can take the data closer to the destination (or sooner in time).

Furthermore, GeoSpray uses the concept of active receipts to explicitly clear delivered bundles. These bundles, which are buffered at intermediate nodes, are removed in order to improve storage capacity for upcoming bundles.

CaD (Converge and Diverge)

CaD [40] uses two phases to route bundles to the mobile destination, converge and diverge phases. The converge phase enables message replication to support fast convergence to the edge of this range, reducing the latency. CaD estimates the movement radius of the destination using its historical location, speed and time elapsed. During the converge phase, the node carrying the bundle replicates it to the encountered node only if that node is moving faster towards the movement area of destination (distance from the area, speed and angle of its direction to the area are considered). The diverge phase is started once the node is within the destination movement area. The angle of the moving node is not taken into account and a replication candidate is chosen based on how fast it can diverge to cover the area of estimated movement of the destination node. This process aims to reduce the latency.

To enhance routing efficiency, it is used delegation replication (similar to delegation forwarding [41]), that originally enables messaging to cache an updated threshold value equal to the recorded utility metric for the message destination, promoting message replication to the candidate node with a better utility metric in comparison to this cached threshold.

Greedy Perimeter Coordinator Routing (GPCR)

GPCR [42] takes advantage of the fact that streets and junctions form a natural planar graph, thus not requiring any global or external information such as a static street map. It consists of two parts: a restricted greedy forwarding procedure which considers that junctions are the only places where actual routing decisions are taken; and a repair strategy which is based on the topology of real-world streets and junctions.

To avoid congestion problems and improve the performance of GPCR, a congestion aware algorithm (Congestion-Aware GPCR) is proposed in [43], which takes not only the distance between the next node and destination, but also the free buffer queue size into consideration to select the best next node in the restricted greedy forwarding procedure of GPCR.

GreedyFlow

GreedyFlow [44] proposes a distributed greedy routing algorithm for efficient packet routing between landmarks in DTNs. GreedyFlow builds a local traffic map and a global landmark map on each node. Each node collects node transit frequencies between landmarks in the area it primarily visits, and uses such information to build a local traffic map. The global landmark map shows the distribution of landmarks in the system and is built offline. In the forwarding decision, the global landmark map shows the general packet forwarding direction, while the local traffic map helps determine the next-hop landmark on the fastest path in the forwarding direction. As a result, packets are greedily forwarded toward their destination landmarks. The routing efficiency is further improved by two advanced components that enhance the consistency of distributively maintained local traffic maps and exploit node-based forwarding to reduce the expected delay.

Mobility and movement prediction

HVR (History based Vector Routing)

HVR [45] is an extension of the approach in [46] so that each node maintains its own location history as well as the location history of previously encountered nodes. When two nodes meet, information vectors are exchanged. Also, an estimate of the rendezvous probability for a given bundle is considered. The rendezvous probability is calculated using a geometric method based on this predicted area. The rendezvous probability of a node, exemplified in Figure 2.5, is defined as $\frac{Area_{ints}}{Area_{pred}}$, where $Area_{pred}$ is the predicted area of the destination nodes' location, and $Area_{ints}$ is the intersection between the nodes transmission range and $Area_{pred}$. Although the rendezvous probability of a node at the current time, t_{curr}, is zero because the expected area of the destination node is located out of its transmission range, the probability may increase if the node has been moving towards the destination node for a period of time Δt. To take such movement into account, an additional movement estimation from t_{curr} to $t_{curr} + \Delta t$ is also considered. A bundle is replicated to its neighbour node with the highest rendezvous probability regarding its destination node.

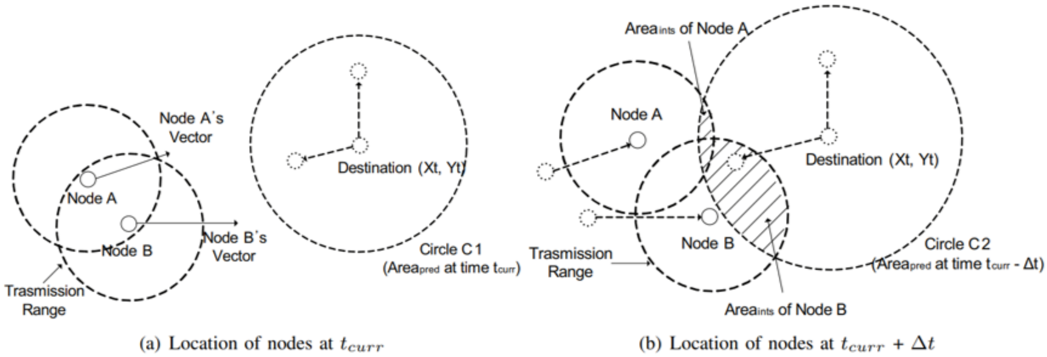


Figure 2.5: Calculation of rendezvous probability [45].

VeloSent (Velocity and direction)

VeloSent routing protocol [47] operates in three consecutive phases. First, an analysis of the environment is performed to detect neighbouring nodes and their respective contexts. Then, the neighbours that visited the destination more recently are considered using the context information about time, location and velocity of the destination node. Secondly, from this information, an estimation of the new location of the destination node is obtained (a straight line route is considered). In the third phase, the estimated location of the destination is used in order to determine which of the neighbouring nodes will most likely meet the destination node. This process is possible by mathematically estimating the timing and the location of a likely encounter between them.

Greedy with Min Cost Reliable Path (GMCRP)

GMCRP [48] makes use of a timeliness-aware trajectory data mining algorithm to mine frequent trajectories and generate movement patterns so that nodes' future location can be predicted. Using the predicted spatial and time information, a time-space graph model is constructed.

The path with the least cost among all minimum cost reliable paths between a node pair in each round is picked, based on two metrics: reliability degree of links and link overhead. Furthermore, a reliability threshold is considered, where undesirable paths are removed.

Novel Trajectory-based Routing (NTR)

NTR's [49] main objective is to efficiently transfer packets among vehicles using their past moving trajectories. It works in two stages: the vehicle trajectory prediction and comparison stage; and the packet forwarding stage. In the first stage, a data mining technique is adopted to build the trajectory tree of each vehicle according to its past moving trajectory data. The trajectory tree makes use of the following vehicle's information: intersections passed with timestamps and average velocities. By comparing intersections with their associated timestamps among vehicle trajectory trees, the vehicle encounter tree is constructed, where the information regarding all the possible encounters in the past is recorded. Afterwards, the vehicle encounter tree is transformed into a packet delivery graph, which predicts the best packet delivery path from the source vehicle to the destination one and calculates the optimal number of packet copies, i.e., the value of tokens, sprayed to each relay vehicle in the second stage. Also, a recovery mechanisms is proposed to handle three exception cases if the prediction for vehicle encounters in the first stage fails.

Social-based forwarding

GrAnt (Greedy Ant)

GrAnt [50] uses ant colony optimization considering the social metrics like degree centrality, betweenness centrality and social proximity.

Figure 2.6 presents its execution overview in a small network, where a Forward Ant (FA) k is sent together with a data message m toward a destination d (Figure 2.6 (a)). The path to d is constructed based on the knowledge acquired by this FA, which dictates the forwarding decision at a node and tries to infer the capability of good next forwarders to d. While being forwarded, each FA k collects the quality information (Q_x) of every node x along the path to d (Figure 2.6 (b)). After finding the destination, a Backward Ant (BA) is created and sent through the reverse path indicated by the FA. The BA stores the total quality Q^k_{path} of the path found by the FA and deposits a pheromone (proportional to the total quality) at links between nodes along the reverse path (from d to the source) (Figure 2.6 (c)). If subsequent messages are forwarded to the same destination, the already deposited pheromone is reinforced at those links and helps the forwarding of future FAs to the same destination (Figure 2.6 (d)). To direct DTN traffic to the most promising contacts, GrAnt uses information about opportunistic social connectivity between nodes.

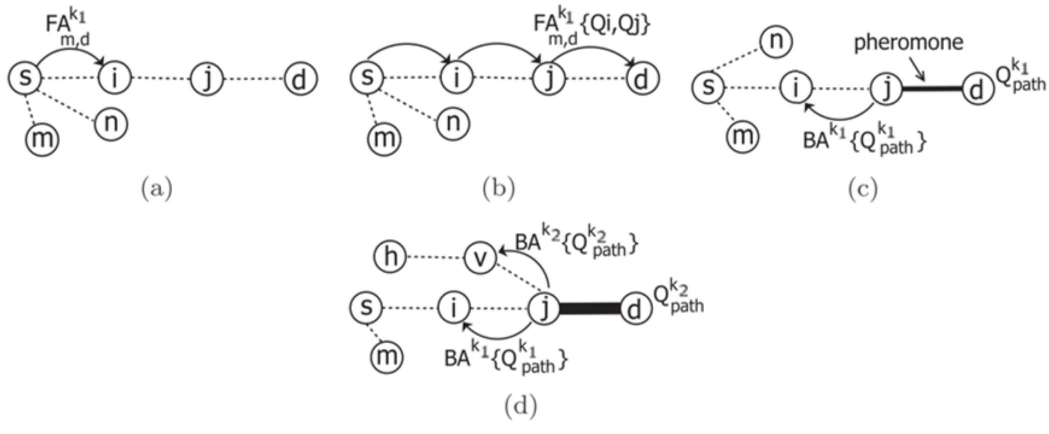


Figure 2.6: Overview of the GrAnt protocol execution [50].

SimBet routing

SimBet routing [51] uses betweenness centrality and similarity metrics to take forwarding decisions. Betweenness centrality is the measurement of a nodes' bridging capabilities between different communities. This metric can help in performing efficient forwarding between two different communities, and thus, a node can forward the bundle to another node with better betweenness centrality. When two nodes belong to the same community, they are more likely to meet each other. Thus, forwarding the bundle to a node with better similarity with the destination node has a better chance of a quick delivery. Therefore, the similarity metric is useful in the later stages of forwarding.

SimBetTS [52] extended utility to include tie strength, resulting in an improvement of the above protocol. The tie strength indicators are: Frequency, Intimacy/Closeness, Long period of time (longevity), Reciprocity, Recency, Multiple social context and Mutual confiding (trust). In an encounter, if a node has higher utility for a given destination, messages are exchanged and removed from the queue based on replication definition.

BUBBLE Rap

BUBBLE Rap [53] uses community affiliation labels with betweenness centrality measures to forward messages. Two types of centrality metrics are calculated for different nodes: global and local. When a source sends a packet to a destination, then a global bubble forwarding takes place so that the packet is hierarchically forwarded using global centrality until it reaches a node belonging to the community of the destination. After that, local centrality is used to forward the packet inside the community until it reaches the destination.

A modified version of BUBBLE, BUBBLE-B [54], deletes the message from the buffer of the original carrier once the message is transferred to the destination community. Decreasing the number of copies (further reducing the cost) does not negatively impact the delivery ratio, hence being a good addition.

DelQue (Delegation Query)

DelQue [55] focuses on sources initializing interest-based queries and selects relays by considering their capabilities for both query and response. The chosen relays take charge of both querying the relevant interest data and returning it to the demander. DelQue uses geo-community and mobility prediction in its algorithm. The Geo-community concept is used to explore location information from a community, taking advantage of its geolocation. Mobility prediction is used when the source has mobile properties. Spatio-temporal prediction is also used, in order to exploit the information that some nodes obtain at a given location and at a given interval of time.

CAF (Community Aware Framework)

CAF [56] considers a combination of geolocation with social characteristics. This scheme relies on the fact that rank-based forwarding algorithms operate normally within the same sub-community (SC). So, messages will be forwarded relying on rank-based techniques toward nodes which belong to the same sub-community.

On the other hand, particular nodes will operate as an inter-community backbone and circulate messages to other sub-communities where they will then be forwarded according to a rank-based forwarding rule. These backbone-like nodes are called MultiHomed nodes (MH). MultiHomed nodes are characterized by their higher mobility and belong to multiple sub-communities (i.e., as they move from one sub-community to another). These MultiHomed nodes (MH_{rank}) are ranked according to the number of subcommunities they belong to (SC(i) is equal to the number of sub-communities node i belongs to). Therefore, MH nodes carrying a message, forward it to other MH nodes according to a non-decreasing MH_{rank}.

In the process of computing the global node ranking in the whole system, the overhead introduced is relatively negligible compared to the overhead induced by legacy algorithms.

Active Area based Routing (AAR)

AAR [57] analyzes the vehicle traces and searches for i) active subareas frequently used and ii) vehicles that frequently meet each other inside each subarea. Then, the routing algorithm distributes the packet copy to each active subarea of the target vehicle using a traffic-aware shortest path spreading algorithm, and then in each subarea, each packet carrier tries to forward the packet to a vehicle presenting a higher encounter frequency with the target vehicle. An improved version, Advanced AAR (AAAR), exploits the spatio-temporal correlation of the visiting times of target vehicles on different RSUs to improve the routing efficiency in each sub-area.

Tie and Duration Based Routing Protocol (TDRP)

To select a relay node, TDRP [58] takes two social metrics into consideration: community and centrality. A distributed K-Clique [59] community detection is adopted to divide vehicles into different communities, and calculate global and local centralities of vehicles by making full use of strong and weak ties, as well as the duration of historical connections. To do so, each vehicle records a neighbor list within the transmission range, as well as a connection history list. Then, strong and weak ties are used, as well as duration of historical connections to calculate global and local centralities.

When the transmission occurs in the same community, the vehicle carrying the message chooses the next relay vehicle with the highest local centrality among its neighbors and itself. If this case is not feasible, the vehicle carrying the message selects the next hop with the highest global centrality among its neighbors and higher global centrality than itself.

Opportunistic forwarding

Best Quality Contact

The Best quality contact [60] bases forwarding decisions on the link quality between neighbours. The quality of the connection is calculated taking into account three metrics between nodes: Signal Strength Intensity (SSI), connection stability and bandwidth. SSI refers to the difference between the receptor antenna power and a reference, while the connection stability measures the ratio between received packets in comparison with the total packets sent in a recent time window. The bandwidth refers to the bitrate supported in the connection between two nodes.

In order to avoid backwards retransmission, if the node where the packet came from is the neighbour with the best quality, a new neighbor is selected, the subsequent one regarding the link quality. Since the forwarding strategy makes the decision only based on the link quality, no considerations are made regarding the forwarding to the pretended gateway. To improve the strategy, the author presents another version of this strategy, Best quality contact to the gateway.

In the Best quality contact to the gateway, each node has a routing table which contains some gateway information, namely: its Endpoint Identifier (EID), the respective quality, the number of hops and the EID of the next neighbour chosen to send the information. The quality is obtained multiplying the successive qualities between nodes to the gateway. This strategy consists in sending packets to the neighbour that has the best quality to the gateway and, if two nodes have the same quality, the number of hops to the gateway is the tiebreaker.

Controlled Replication

Controlled Replication [28] implements two mechanisms to deal with network congestion, namely the Loop Avoidance mechanism and the Congestion minimization mechanism. Respectively, these aim to reduce the number of packet loops and limit the number of hops in the network.

On the other hand, to perform the opportunistic decision, it is considered a neighbourhood rank classification. Two ways of performing the decision are considered: node contacts based classification and mobility based classification. In node contacts based classification, the number of contacts that a node had over a predefined period of time and the last recorded timestamp in which a node had contact with a Road Side Unit (RSU) are used to perform the evaluation. In mobility based classification, the node mean velocity and the node heading angle are used to perform the classification. An assumption that a node moving in the opposite direction has a lower probability of delivering the packet is made, since the node will follow the already travelled path. Therefore, the usage of the heading angle aims to identify nodes that are moving in the opposite direction. Moreover, an evaluation based on the mean velocity can prevent packets from being sent to static nodes.

2.2.4 Routing Strategies Summary

This subsection aims to summarize all the addressed routing strategies, showing the differences and trade-offs among them, which are presented in Table 2.1.

Distinct taxonomies have been explored, all of them with characteristics of interest to the development of this dissertation. Single copy routing is a very interesting approach from a resource point of view, as it keeps the resource usage of the network at a low level, lacking mostly in fields like delivery ratio and average delay. On the other hand, flooding mechanisms achieve a very high delivery ratio and low delays, but a lot of network resources are wasted. Admittedly, these two mechanisms are at the ends of the spectrum of possibilities regarding routing strategies, making them very good terms of comparison further in the development of new strategies.

Additionally, probability strategies emphasise the importance of spreading only a few copies of messages by considering node capabilities, message prioritization and encounter history. Despite achieving high delivery ratios while reducing the consumption of resources, some of these mechanisms require strong storage capabilities. Also, some are based on timers, which degrades their performance in high mobile environments, resulting in inaccuracies at the encounter stage.

Geographic based forwarding uses the location of nodes to perform routing decisions, hence decreasing the usage of resources expended. Coupled with geographic based forwarding, mobility and movement prediction algorithms also make use of geolocation but in order to predict movement and anticipate events. These are good features, since this dissertation

Algorithms	Type	Single/ Multiple copy	Replication rate	Information needed	Objectives/comments
First Contact	Probabilistic	S	Very Low	N/A	Random search is used to deliver the bundle to its destination
Direct delivery	Direct	S	None	N/A	Source moves and delivers the bundle directly
Epidemic	Flooding	M	Very High	N/A	Rapid propagation of data
MaxProp	Flooding	M	High	N/A	Use of the delivery likelihood as a cost assigned to each destination
Spray and Wait	Controlled Flooding	S/M	Medium	N/A	Sets a limit on the number of copies
PRoPHET	Probabilistic	M	High	N/A	Forwards packets based on past node encounter history
NECTAR	Probabilistic	M	N/A	N/A	Spreads messages based on a calculation of a neighbourhood index
EASE	Aging	M	N/A	N/A	Computes routes purely based on last encounters
HPR	Probabilistic	M	N/A	N/A	Messages sent to increasing delivery probability nodes using mecanisms that reduce network overhead
GeoSpray	Geo	S/M	Medium	Navigation	Does not tackle mobile destination
CaD	Geo	S/M	Medium	Navigation, destination trajectory	Replicate at end to reach the destination
GPCR	Geo	M	Medium	Topology of real-world streets and junctions	Road junctions are the only places where actual routing decisions are taken
GreedyFlow	Geo	M	N/A	Transit frequencies between landmarks	A local traffic map and a global landmark map are used to perform the forwarding decision
HVR	Movement Prediction	M	Medium-High	Movement vector, velocity	Uses historical vectors of encountered nodes
VeloSent	Movement Prediction	M	N/A	Location and velocity associated with time	Tries to predict destination movement, routing accordingly
GMCRP	Movement Prediction	S	N/A	Location associated with time	Mines frequent trajectories and generate movement patterns
NTR	Movement Prediction	M	Low-Medium	Location associated with time	Mines frequent trajectories and generate trajectory trees with restricted number of packet replicas
GrAnt	Social	M	Medium	Path quality metrics	Past routes are considered as reliable routing options
SimBet	Social	M	Medium	Global contact info	Uses betweenness centrality and similarity metrics to take forwarding decisions
BUBBLE Rap	Social	M	Medium	Global contact info	Uses 2 phases: global and local forwarding
DelQue	Social	M	Medium	Spatio-temporal mobility	Receiver driven approach
CAF	Social	M	Medium	Geolocation	Combines geolocation with social characteristics
AAR	Social	M	Medium	Location associated with time	Searches for active subareas frequently used and vehicles that frequently meets each other inside each subarea
TDRP	Social	M	N/A	Historical connection list	Uses community and centrality metrics to perform forwarding decisions
Best Quality Contact	Opportunistic	M	Low-Medium	SSI, Connection Stability, Bandwidth	Discovers the best link quality to perform forward decisions
Best Quality Contact w/GW	Opportunistic	M	Medium	SSI, Connection Stability, Bandwidth	Discovers the best link quality to the gateway to perform forward decisions
Controlled Replication - Node contacts	Opportunistic	M	Low	Number of contacts, Timestamps of last connection with RSU	Uses neighbourhood rank classification based on node contact features in a recent time window
Controlled Replication - Mobile	Opportunistic	M	Medium	Mean velocity, heading angle	Uses neighbourhood rank classification based on movement properties of nodes

Table 2.1: Routing strategies summary [61].

focuses on highly mobile environments.

Undoubtedly, social-based forwarding can indeed enhance the routing decisions of a community. If social similarity metrics are able to capture the dynamic behaviour of users, connectivity graphs are going to reflect the most important social edges, translating into a very good approximation of the pretended combination of characteristics. Nevertheless, these metrics are not easy to obtain and suffer from subjectiveness.

Although the opportunistic routing strategies here presented can also fit on other taxonomies, these have the most direct correlation with the reality, since they aim to focus on connection metrics and perform accordingly.

As conclusion, a good exploitation of features according to the application at hand and the scenarios' characteristics may be the best approach, since it eliminates unnecessary variables and will focus on the final objective. This will be the subject of the next chapter.

2.3 Smart Cities Deployments

This section emphasizes on Smart Cites testbeds that perform data gathering through different types of sensors, and also use DTN alike features and/or LPWAN technologies to disseminate the data through the network. In [62], it is proposed a data-flow framework for the realization of smart cities through IoT. Various stages are emphasized, namely collection, processing, management and interpretation.

PortoLivingLab [63] is an urban-scale, multi-source sensing infrastructure deployed in the city of Porto, Portugal. This infrastructure has three monitoring platforms: SenseMyCity [64], UrbanSense [65] and BusNet [66]. SenseMyCity is a mobile crowdsensing research tool that consists of a smartphone application that can gather sensor data from the device and also considers user inputs. UrbanSense makes use of 19 strategic locations to deploy low-cost sensing stations in order to monitor environmental parameters and weather conditions. BusNet supports distributed data collection from more than 600 public transport vehicles equipped with OBUs in order to provide vehicle-to-everything communications and internet connectivity to passengers.

CityOfThings [67] is a smart city testbed located in the city of Antwerp, Belgium. It consists of a multi-wireless technology network infrastructure, with the capacity to easily perform data experiments and a living lab approach to validate the same. In terms of infrastructure, three different components are considered, namely a series of multi-technology gateways, mainly used for network experimentation; LoRaWAN data network to ensure a continuous stream of data from the City of Things sensors; and a wide variety of city sensors, such as air quality sensors, traffic monitoring sensors, parking sensors, smart parking sings, among others.

SmartSantander [68] portrays the deployment and experimentation architecture of the IoT experimentation facility being deployed at Santander city, in Spain. The SmartSantander platform has been conceived in a three-tiered architecture: IoT device tier, IoT gateway tier and server tier. IoT devices are responsible for obtaining the real-world information, are deployed at hard-to-reach locations and have dual power supplies (electric distribution network combined with batteries). The IoT gateway node tier links the IoT devices at the edges of the network to a core network infrastructure. The server tier provides more powerful server devices which are directly connected to the core network infrastructure. Hence, SmartSantander counts with more than 15.000 sensors (static and mobile) which provide a wide variety of applications, such as real-time information regarding different environmental parameters (light, temperature, noise, CO_2), and other parameters like occupancy of parking slots [69].

Oulu Smart City [70] is an ubiquitous computing testbed that has been deployed in the city of Oulu (Northern Finland) that incorporates different wireless networks, large public displays, middleware resources and monitoring tools. The network has more than 1200 access points that provide both indoor and outdoor coverage in places deemed relevant for public access. In addition, the network has 12 wireless sensor networks co-located with the displays, where the access points have IEEE 802.15.4 Medium Access Control (MAC) with 2.4 GHz radios.

Padova Smart City [71] is localized in the city of Padova, Italy. The application consists of a system for collecting environmental data and monitoring the public street lighting by means of wireless nodes, equipped with different kinds of sensors, placed on street light poles and connected to the Internet through a gateway unit. The environmental parameters collected are CO level, air temperature, air humidity, vibrations, noise, and others. Each streetlight is geographically located on the city map and uniquely associated to the IoT node attached to it, so that IoT data can be enhanced with context information. These IoT nodes form a 6LoWPAN multi-hop cloud, using IEEE 802.15.4 technology.

A survey on IoT-based Smart cities was performed in [72], where policies and strategies of practical experiences conducted in smart city related deployments were evaluated, namely in the cities of Amsterdam, Barcelona, Stockholm and Santa Cruz.

2.4 Chapter Considerations

This chapter provided an overview of the DTN concept and an architecture implementation, namely the mOVE architecture. As explained, this DTN architecture serves as a validation tool for this work, however requiring integration with new functionalities.

In order to understand the current state of the art in terms of DTN-based forwarding strategies and use some of them as baseline comparison for this work, a wide range of schemes were presented, and the trade-offs between them were summarized.

Regarding the overall context of the development of this work in current matters, data gathering, IoT and Smart Cities deployments were also addressed.

Chapter 3

Proposed Architecture

3.1 Introduction

This chapter presents the proposed architecture, where multiple technologies are explored to support data gathering in IoT. A theoretical approach of the respective contributions is outlined in this chapter, where the proposed mechanisms and protocols are described.

Section 3.2 overviews the architecture and its general features and objectives.

Section 3.3 details the network elements and their function in the network. For each element, a general description of its components is provided. Moreover, the interaction between software blocks and distinct network elements is explained.

Section 3.4 presents a mechanism of node discovery using BLE.

Section 3.5 introduces the proposed neighboring selection strategies considering GPS and social metrics.

Section 3.6 presents the proposed packet selection mechanisms.

Section 3.7 presents the chapter summary and considerations.

3.2 Architecture Overview

IoT is expected to produce an enormous amount of data, as well as a significant variety and heterogeneity of the data. In order to address these challenges, the proposed architecture, illustrated in Figure 3.1, aims to make use of opportunistic communications - making use of every device that provides some form of mobility - to disseminate the data collected from data collecting units to gateways.

The elements that support this architecture are Data Collecting Units (DCUs), Mobile Nodes or On Board Units (MNs/OBUs), Gateways or Road Side Units (GTWs/RSUs) and the Server.

In a real deployment, MNs are comprised of two parts: the network controller device and the vehicle itself. Examples of suitable transportation vehicles are: bicycles, cars, buses, aerial drones, aquatic drones, among others.

This architecture provides the following features to support the collection of data in IoT:

- Heterogeneous sensor data gathering in IoT.
- Opportunistic communications (vehicles transport the information).
- Forwarding strategies based on GPS location and social metrics to enhance the ratio of data delivery vs resource usage in the network.
- Multi-technology communication (WiFi, LoRa and BLE).
  - WiFi implements forwarding strategies in a Delay Tolerant Network (DTN).
  - Neighbourhood management uses Bluetooth Low Energy (BLE).
  - LoRa is used as a support technology for larger radio coverage.
- Software modularity to allow integration with new types of sensors, communication technologies and different types of mobile nodes.

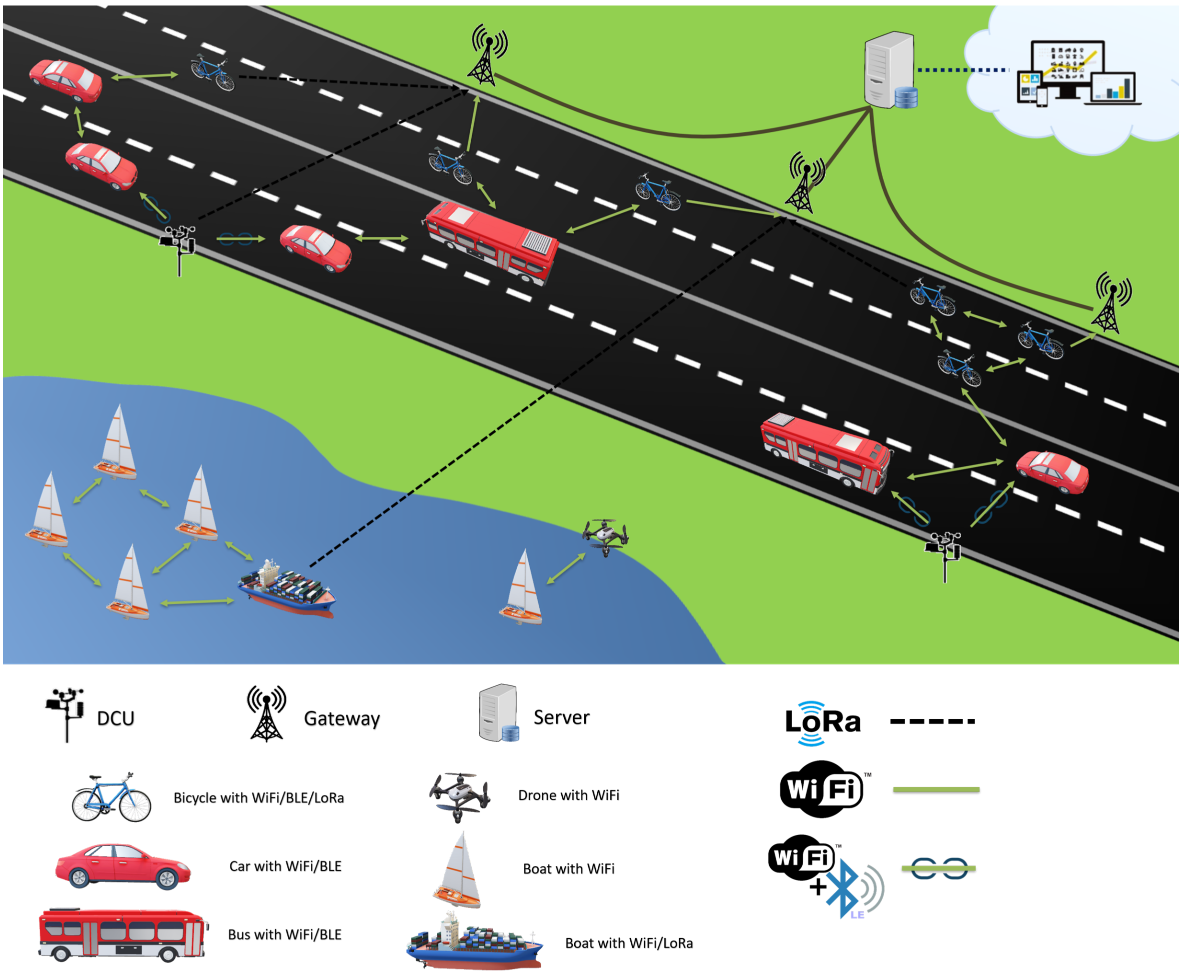


Figure 3.1: Proposed architecture.

3.3 Network Elements

All the hardware supporting the development of the several architectural elements is detailed in Annex A, which may be consulted for further details.

Prior to the description of each network element, it is relevant to mention that software modularity is achieved and mediated through Inter-Process Communication (IPC) connections such as Transmission Control Protocol (TCP) and UDP Sockets.

Furthermore, it is important to state that the elements used in this architecture were not developed from scratch [17]. All element generic figures will have references to which modules were developed from scratch or further developed. For unmodified software blocks, a brief description will be provided.

3.3.1 Data Collecting Units (DCUs)

DCUs are devices whose purpose is to detect and collect a physical property of interest. These devices usually have access to the energy network; however, the location of deployment lacks an end-to-end connection to a gateway. Therefore, these devices must be able to store their information with an associated measurement timestamp and either deliver this information to opportunistic connections or use long range communications periodically in the form of packets.

Figure 3.2 demonstrates a DCU prototype. Both hardware and software were developed taking into consideration the possibility of integrating new and different types of sensors. For instance, Inter-Integrated Circuit (I_2C) communication protocol based sensors are good options since these do not require additional wiring to connect to the board (from the Raspberry Pi point of view, no more pins are used than the first four). Nevertheless, the integration of new sensors is possible regardless of its communication interface.

Figure 3.2: DCU prototype image.

With a software and hardware breakdown based on Figure 3.3, all blocks and the interaction between them aim to address some of the challenges presented for this type of nodes. Taking a bottom-up perspective at the software, each block main function is now described:

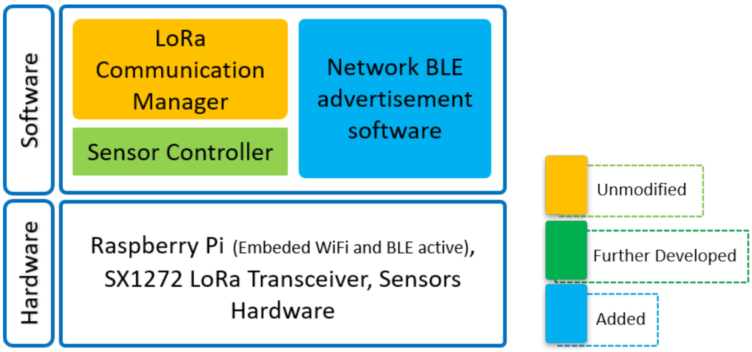


Figure 3.3: DCU breakdown.

- **Sensor Controller**: upon enumerating the specific sensors of interest, this software gathers the measured values from the sensors in a periodic manner, where different periodicities are allowed. Each set of sensed data can be used to build different types of data packets. At the moment of measurement, a timestamp is aggregated to the packet and stored in a buffer for future use. This software was designed to allow easy and fast integration of new sensors, as well as new types of packets.

- **BLE Advertisement Software**: the data packets that are received from the sensor controller software are placed in a buffer. The BLE advertisement software is continuously listening for mobile node BLE advertisements. Respectively, when an association occurs, the data packets in the buffer are forwarded (using WiFi, due to the larger transmission rate) to the mobile node.

- **LoRa Communication Manager**: when packets are available, DCUs make use of a long range communication technology (LoRa) to deliver a minimum amount of data packets, being this process managed by the LoRa communication manager software. Since LoRa has a limitation on its transmission duty cycle of 1%, a low number of packets are delivered by this technology. Even so, it is able to keep a steady stream of data delivery through the day.

3.3.2 Mobile Nodes / On Board Units (MNs/OBUs)

MNs are devices that can be aggregated to any type of vehicle and whose network function is to fill the distance/connectivity gap between DCUs and gateways through their mobility patterns. Since no end-to-end connection is possible (for most cases), this type of node must be able to store the packets received from DCUs to later deliver to a gateway. This is possible through the implementation of a DTN, resulting in a store, carry and forward mechanism. Alternatively, such as DCUs, this type of nodes can also resort to the LoRa technology for low but steady packet delivery.

Figure 3.4 demonstrates a Mobile Node prototype. Such devices can also be used as sensing units, with the addition of the board presented in Figure A.2. Generally, all the gathered information is associated with a GPS position, to associate location with the data. Moreover, GPS information can also be useful to support multi-hop data transmission decisions, as it will be later explored in this document. This type of network elements typically have higher energy constrains than other types of network elements due to their mobility.

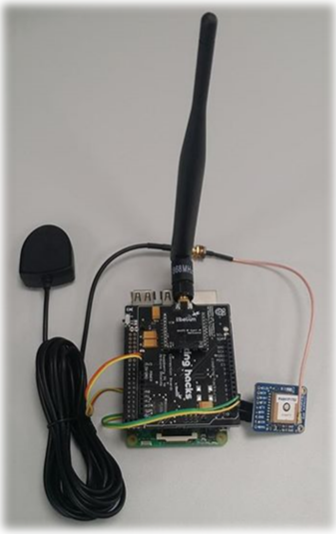


Figure 3.4: MN prototype image.

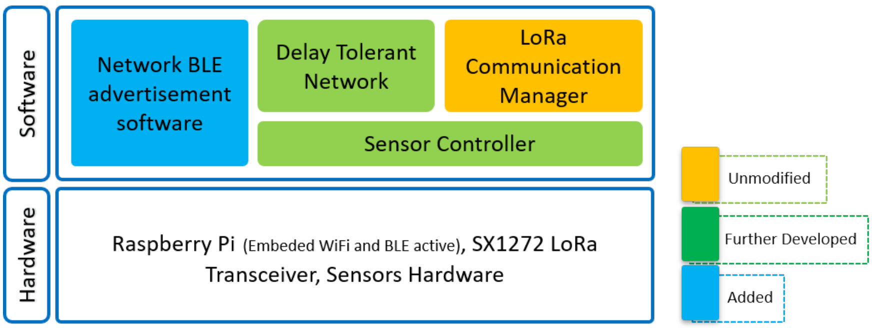


Figure 3.5: MN breakdown.

With a software and hardware breakdown based on Figure 3.5, all blocks and the interaction between them aims to address some of the challenges presented for this type of nodes. Taking a bottom-up perspective at the software, each block main function is now described:

- **Sensor Controller**: this module works similarly to what was previous presented for the DCUs. One main difference is that, when data packets are generated in a MN, these can be either sent to the DTN or to the LoRa Communication Manager.
- **DTN**: the DTN implements the mOVE architecture, using a store carry and forward mechanism to overcome intermittent connections and lack of end-to-end connectivity. The architecture was generally described in Section 2.2.2 and proposed enhancements are detailed in Section 4.3.
- **LoRa Communication Manager**: this block is used as an alternative long range communication to deliver small amounts of data, delivering, for example, data packets that exceeded their lifetime in the DTN.
- **BLE Advertisement Software**: running in parallel with all the other software, MNs announce periodically their presence using BLE advertisements to establish a connection with DCUs.

3.3.3 Gateways / Road Side Units (GTWs/RSUs)

Gateways' main function is to act as the final element in the data gathering chain. Being a non-mobile node whose function is to populate a database, generally, a Gateway has connectivity to the server. To achieve such a goal, a Gateway has seamless and transparent data swapping with the Server, using sensor identification through predefined IDs.

Figure 3.6 demonstrates a Gateway prototype. Information can arrive at a Gateway through multiple technologies, either WiFi or LoRa, which will be further explained in the software blocks analyses below. Similarly to MNs, Gateways can act as sensing units, also with additional hardware. In this case, information is directly delivered to the server.

Figure 3.6: Gateway prototype image.

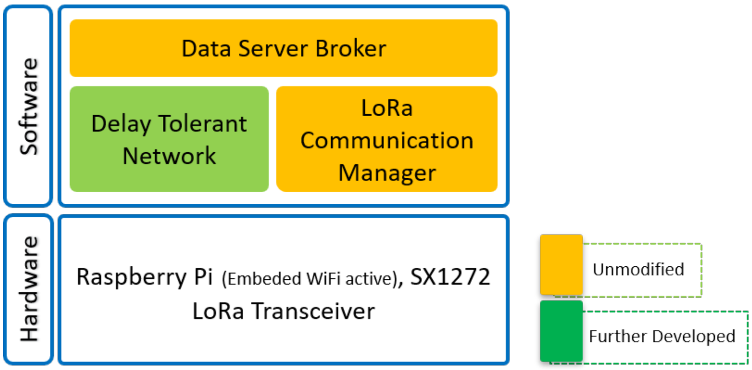


Figure 3.7: Gateway breakdown.

With a software and hardware breakdown based on Figure 3.7, all blocks and the interaction flows aim to address some of the challenges presented for this type of nodes. Taking a bottom-up perspective at the software, each block main function is now described:

- **Delay Tolerant Network**: As explained for the mobile node, this module implements the mOVE architecture. As the last element before the server, this network element aims to receive the data packets delivered by the DTN in the mobile nodes and send them to the Data Server Broker.

- **LoRa Communication Manager**: For this network element, this module manages the reception of packets through LoRa and also sends them to the Data Server Broker.

- **Data Server Broker**: Upon receiving data packets from diverse communication technologies, this module mediates the interaction between this network element and the server.

3.4 Network BLE Advertisement Manager

Prior to the development of this module, the process of neighboring discovery through WiFi required active scanning. This process incurs in high energy expenditure, which motivates the usage of BLE to enhance the energy performance of the discovery process.

Since the radio coverage of the BLE technology is theoretically close to WiFi's (around 100 meters), both technologies are compatible in this regard.

In terms of functionality, mobile nodes announce their presence in the network through BLE advertisement packets and, upon entering the coverage range of a DCU - which will be scanning for these advertisement packets - both nodes can proceed to data transmission.

Data transmission is performed through a UDP socket (DCU Socket), which is supported by WiFi because the channel's bandwidth are much higher than BLE's.

In resume, BLE deals with neighbouring discovery between DCUs and mobile nodes while WiFi supports the data packet transferring process for the same nodes.

3.5 Proposed Forwarding Strategies

Gathering data in IoT is challenging because the location of sensors may be remote, and the amount of gathered data can be large. To address such challenges, the proposed architecture allows the support of different data gathering forwarding strategies. As baselines, two simple schemes were implemented: Direct Contact and Epidemic. Both strategies were already described in Section 2.2.3, representing both ends of the spectrum in terms of delivery capacity, delay and network overhead.

The proposed strategies support multi-hop packet transmission where a good balance of delivery capacity, delay and network overhead aims to enhance network performance through mobility and social metrics. Since multi-hop transmissions - through other mobile nodes - are considered, two mechanisms were used to control the replication of packets: Loop Avoidance mechanism and Congestion Minimization mechanism. These mechanisms were proposed in [28] and were considered in this work.

In the same scope, a storage congestion management mechanism is proposed. Different than just considering a maximum packet lifetime to avoid indefinite storage occupancy, this mechanism takes into consideration the network's state in order to avoid storage congestion. Each node calculates its ranking based on different metrics and informs the network. These calculations are kept in an extended time window, which allows a medium calculation of each node's ranking for that time frame. Using *neighbor announcement* packets to spread this information in the network, nodes' neighbors are aware of their neighbors historical ranking records. This mechanism assumes that a node with a much higher historical rank will be a better forwarder than the node holding the packet, therefore removing the packet from storage after forwarding it to the neighbor. Considering this feature, the node's storage frees some valuable space taking into consideration the networks historical state, therefore minimizing the storage congestion. This mechanism is only implemented is specific cases, deferring to the following equation:

$$NeighHistoricalRanking - OwnRanking \geq HistoricalRankThreshold \qquad (3.1)$$

where the $(NeighHistoricalRanking - OwnRanking)$ represents the difference in historical ranking that a mobile node neighbor has in relation with the node, and the $HistoricalRankThreshold$ represents the threshold needed for the historical ranking difference to effectively erase the packet sent from storage. $HistoricalRankThreshold$ is parametrizable, allowing smaller or bigger differences in historical ranking to affect the storage through this mechanism.

Other than controlling the packets' replication, it is highly important to determine which are the good neighbors to forward the packets. Objectively, selected neighbors should minimize the expended resources and packet delay, and maximize the delivery ratio. By default, if a node has connection to a gateway, it delivers the maximum amount of packets it can do. On the other hand, if only mobile nodes are neighbours, it is important do address which of them are good candidates to replicate the data packets. As described before, a node announces its presence in the network and detects neighbors using *neighbor announcement* packets. Moreover, these packets are also able to inform neighbors about their rank in the network. Knowledge about neighbor ranking is then used to decide which are the good options to forward the packets. Three neighboring rank classification methods are proposed: Gateway Location Awareness (GLA), Aging Social-Aware Ranking (ASAR) and an Hybrid version of both GLA and ASAR, which will be referred in this document as HYBRID.

3.5.1 Gateway Location Awareness (GLA)

As the name suggests, mobile nodes are aware of the GPS location of gateways, and this strategy assumes that each mobile node can use that information to make forwarding decisions. Considering GPS acquaintance, each node uses a set of metrics to calculate a network rank, namely its velocity and best combination of distance and heading angle to a gateway. Each metric is then normalized between 0 and 1 in order to facilitate weighted rank computation.

Distance to the gateway and respective normalization

The distance between two GPS coordinates, respectively the mobile node location $[Lat1, Lon1]$ and the gateway location $[Lat2, Lon2]$, is given by:

$$d = \arccos(\sin(Lat1) \times \sin(Lat2) + \cos(Lat1) \times \cos(Lat2) \times \cos(Lon2 - Lon1)) \times R \quad (3.2)$$

where R is the radius of the earth (6371 kilometers). This formula uses the spherical law of cosines, which gives well-conditioned results down to distances as small as a few metres on the earths surface, and constitutes a light processing computation compared to other alternatives (such as the *haversine* formula[1]).

In order to normalize the distance considered between two nodes, a maximum distance between a mobile node and a gateway of interest ($maxD_{MG}$) needs to be outlined. Moreover, the maximum communication range of the technology (T_{maxR}) should also be considered. Thus, half of the normalization interval [0.5 to 1] is given to the technology communication range, thus decreasing linearly with the distance increase. The remaining normalization interval [0 to 0.5] refers to distances between the maximum communication range of the technology and the maximum distance between a mobile node and a gateway of interest, where an exponential decay is considered to give a higher weight to nodes closer to gateways.

The function that normalizes the distance is expressed as:

$$Normalized_{GatewayDistance}(d) = \begin{cases} 1 - \frac{2 \times d}{T_{maxR}} & , d < T_{maxR} \\ a \times (1-r)^d & , T_{maxR} < d < maxD_{MG} \\ 0 & , d > maxD_{MG} \end{cases} \quad (3.3)$$

where a represents the initial amount, and r is the decay rate. Therefore, knowing that $f(T_{maxR}) = 0.5$ and selecting a decay rate r, the parameters a and $(1-r)$ can be obtained. Also, if the distance is higher than $maxD_{MG}$, that gateway is of no interest to the mobile node, thus reducing the computational effort of processing all gateway distances in the network.

As an example, considering $T_{maxR} = 84$, $maxD_{MG} = 1000$ and $r = 0.4\%$, Figure 3.8 illustrates the correlation between distance in meters to normalized distance.

Heading Angle to the gateway and respective normalization

Calculating the heading angle to the gateway is done is several steps. First, the angle to the gateway is computed. Using the mobile node's location $[Lat1, Lon1]$ and the gateway's

[1]https://www.movable-type.co.uk/scripts/latlong.html → Accessed: 20-07-2018

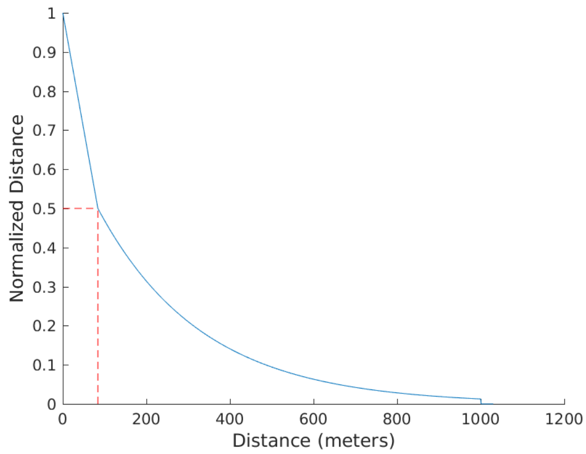


Figure 3.8: Distance normalization.

---

location $[Lat2, Lon2]$, the angle to the gateway is given by:

$$HeadingAngleToGateway_{[0\to360]} = \left| \left[360 - \left[180 + atan2\left(\frac{\phi}{\psi}\right) \%360 \right] \right] - MN_{HeadingAngle} \right| \tag{3.4}$$

where,

$$\phi = \sin(Lon2 - Lon1) \times \cos(Lat1) \tag{3.5}$$

$$\psi = \cos(Lat2) \times \sin(Lat1) - \sin(Lat2) \times \cos(Lat1) \times \cos(Lon2 - Lon1) \tag{3.6}$$

The $MN_{HeadingAngle}$ is the heading angle measured from the GPS module in the mobile node. The $HeadingAngleToGateway$ represents the angular difference from the direction that a mobile is moving towards and the direction where the gateway is located. The resultant value is comprised between 0 and 360 degrees ($HeadingAngleToGateway = \theta_{[0\to360]}$). From the mobile nodes' perspective, it is only important if it is moving towards or in the opposite direction of the gateway, therefore the obtained value is converted to an interval between 0 and 180 degrees, using:

$$\theta_{[0\to180]} = \begin{cases} \theta_{[0\to360]} & , \theta_{[0\to360]} < 180^\circ \\ 360 - \theta_{[0\to360]} & , \theta_{[0\to360]} \geq 180^\circ \end{cases} \tag{3.7}$$

In order to normalize the heading angle between a mobile node and a gateway, an exponential decay is applied to the increasing heading angle to the gateway. Furthermore, three distinct direction intervals are considered: aligned with the gateway $\theta \in [0 \to 45]$; perpendicular direction to the gateway $\theta \in [45 \to 90]$; opposite direction to the gateway $\theta \in [90 \to 180]$. Respectively, increasing decay rates are set for each interval, accordingly to the function:

$$Normalized_{GatewayHeadingAngle}(\theta) = \begin{cases} \alpha_1^\theta & , 0^\circ < \theta \leq 45^\circ \\ \alpha_2^\theta & , 45^\circ < \theta \leq 90^\circ \\ \alpha_3^\theta & , 90^\circ < \theta \leq 180^\circ \end{cases} \tag{3.8}$$

where $\alpha_1 > \alpha_2 > \alpha_3$. As result, the direction's range where the mobile node is found greatly influences the normalized gateway heading angle. Copping with this, the exponential factor also contributes largely to such calculation.

For instance, using the following parameterization: $\alpha_1 = 0.990$, $\alpha_2 = 0.988$ and $\alpha_3 = 0.986$, Figure 3.9 depicts the normalization function, both in rectangular and polar graphs.

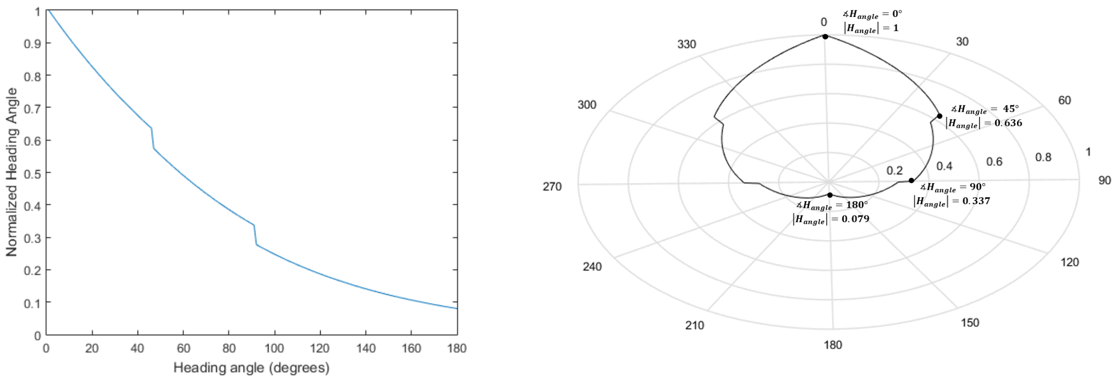


Figure 3.9: Heading angle normalization.

Velocity and respective normalization

The velocity of a mobile node is directly obtained from the GPS module. Using velocity as a metric, not only nodes that cover higher number of meters in lower time instances are prioritized, but it also prevents to some extend the rank viability of static nodes.

The normalized velocity is obtained using the following function:

$$Normalized_{Velocity}(v) = \begin{cases} 0 & , v \leq 0 \\ \frac{v}{V_{avg}} & , 0 < v < V_{avg} \\ 1 & , v \geq V_{avg} \end{cases}, \tag{3.9}$$

where v represents the mean velocity of the mobile node and V_{avg} an acceptable value of mean velocity that a mobile node should have to be granted a good neighbor ranking in terms of velocity. Furthermore, Figure 3.10 illustrates the velocity normalization.

Ranking computation

Since all metrics are normalized between 0 and 1, rank computations are made in a weighted way, described as follows:

$$Rank = W_{GatewayDistance} \times Normalized_{GatewayDistance} + W_{GatewayHeadingAngle} \times Normalized_{GatewayHeadingAngle} + W_{Velocity} \times Normalized_{Velocity} \tag{3.10}$$

where $W_{GatewayDistance}$, $W_{GatewayHeadingAngle}$ and $W_{Velocity}$ represent, respectively, the distance to the gateway weight, the heading angle to the gateway weight and the velocity weight.

Coping with the presented features, this forwarding strategy is conceived taking into consideration three novel network features: i) heterogeneous node mobility characteristics; ii) gateway selection from the multiple gateways available in the network; and iii) adaptability

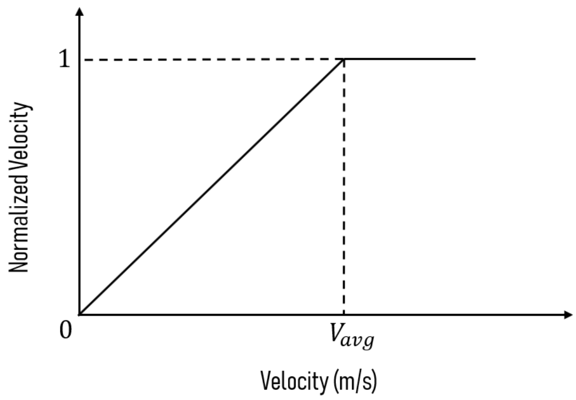


Figure 3.10: Velocity normalization.

to the connectivity technology. Regarding the first point, the model can be calibrated to the mobility characteristics of each type of vehicle. E.g., a bicycle can change its heading angle much faster than a car, but cannot travel as fast as a car. Regarding the second point, both distance and heading angle are calculated as a combination for each gateway in the network in order to provide the best network ranking classification for each node. Finally, for distinct connectivity technologies, the distance to the gateway metric parameters can be adapted accordingly, thus providing algorithmic adaptability to the technology used.

3.5.2 Aging Social-Aware Ranking (ASAR)

Usually, humans are creatures of habits. To take advantage of such behaviors, this forwarding strategy considers the distinct neighbor connections to network elements in a temporal sliding window in order to perform forwarding decisions. A ranking classification is placed and performed in several phases, resulting in a normalized value between 0 and 1.

Sliding time window phase

This phase constitutes the realization of two objectives: determine the number of different neighbors in the sliding window; and the last moment of contact with each one of those nodes. For a practical example, refer to Figure 3.11.

A 5 minute sliding window ($T_{window} = 5 \times 60$) is considered. Different colors represent distinct contacts with mobile nodes, and the time that the connection lasts is represented in the time axis. For instance, when the current timestamp is at 5 minutes, the considered time window is from $[0 \rightarrow 5]$ minutes. In that window, three different contacts are verified: gray, green and blue. The last moment of connection is, respectively, 3 minutes ago, 1 minute ago, and still connected.

When the time window advances, conditions may change. For example, at timestamp 6, four distinct contacts are verified: gray, green, blue and yellow. Respectively, the last moment of connection can also change: 4 minutes ago, 2 minutes ago, still connected, and still connected.

The objective of the sliding time window characterizes the historical connection model of a node, thus helping to make a forwarding decision. To facilitate model calibration, the

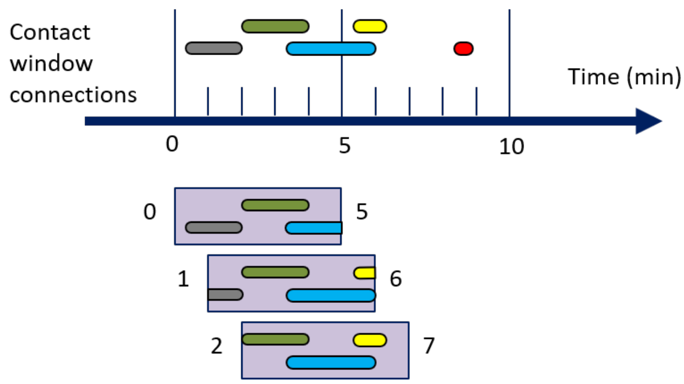


Figure 3.11: Sliding time window example.

---

sliding window interval is parameterizable.

Ranking quantitative classification

In the network, not all neighbors have the same importance in terms of forwarding decisions. For instance, a connection to a gateway is far much important than a connection to a mobile node. Therefore, from the connections detected in the sliding time window, the number of connections to different network elements is used to attribute a ranking to each mobile node.

By default, five ranking intervals are considered, as illustrated in Table 3.1. Two thresholds are defined to set the difference between *low* and *high* number of contacts with mobile nodes and gateways, τ_{MN} and τ_{GW}, respectively. s_{MN} and s_{GW} are used to denote the number of occurences in the observed time window.

$R_{interval}$	**Quantitative Classification**
1	No Gateway contacts, low number of Mobile Node contacts ($s_{GW} = 0 \wedge s_{MN} < \tau_{MN}$)
2	No Gateway contacts, high number of Mobile Node contacts ($s_{GW} = 0 \wedge s_{MN} \geq \tau_{MN}$)
3	Low number of Gateway contacts, no Mobile Node contacts ($s_{GW} < \tau_{GW} \wedge s_{MN} = 0$)
4	Low number of Gateway contacts, Mobile Node contacts ($s_{GW} < \tau_{GW} \wedge s_{MN} > 0$)
5	High number of Gateway contacts, Mobile Node contacts ($s_{GW} \geq \tau_{GW} \wedge s_{MN} > 0$)

Table 3.1: Ranking quantitative classification.

Ranking qualitative classification

After determining in which ranking interval each mobile node belongs, the aging factor is associated with each connection. Furthermore, a distinct aging constant per type of network element is considered, where a connection with a mobile node deteriorates faster than a gateway in terms of ranking. The aging factors are represented by γ_{MN} and γ_{RSU}, respectively.

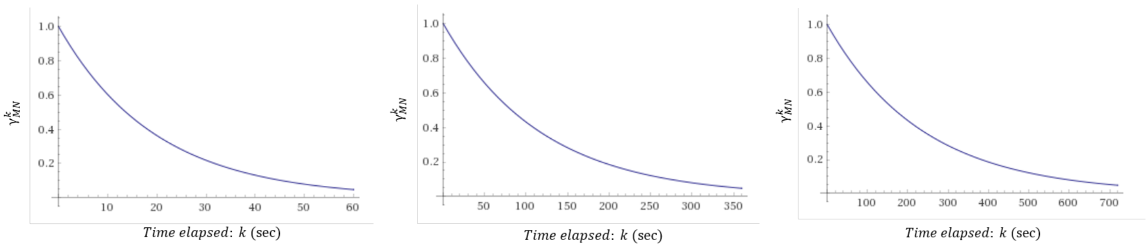
These values are directly related to the sliding window interval. The calculation of both γ_{MN} and γ_{RSU} are interpolated and validated for sliding windows of $[1 \rightarrow 12]$ minutes, so that distinct time intervals are normalized with the same aging dependency, e.g., a T_{window} = 6 minutes must have the same aging dependency with the time elapsed than a T_{window} = 3 minutes, which is obtained through distinct aging constants for each respective T_{window}. This process withholds the calculation of these constants from the network size and, instead, it makes it dependent on the time window (T_{window}) considered. The following expression translates the sliding time window into the aging constant for the mobile node:

$$\begin{aligned}\gamma_{MN} = 0.902209 + 0.00079096 \times T_{window} - 2.6633 \times 10^{-6} \times T^2_{window} \\ + 3.92739 \times 10^{-9} \times T^3_{window} - 2.09139 \times 10^{-12} \times T^4_{window}\end{aligned} \tag{3.11}$$

Coping with this, the following expression translates the sliding time window into the aging constant for the gateway:

$$\begin{aligned}\gamma_{RSU} = 0.916609 + 0.000698413 \times T_{window} - 2.40751 \times 10^{-6} \times T^2_{window} \\ + 3.60303 \times 10^{-9} \times T^3_{window} - 1.93659 \times 10^{-12} \times T^4_{window}\end{aligned} \tag{3.12}$$

The practical benefit of the aforementioned functions is to standardize the relation between time windows and normalized weight of these aging constants. Figure 3.12 demonstrates that feature.



(a) Aging function with T_{window} = 1 min ($\gamma_{MN} = 0.9506$).

(b) Aging function with T_{window} = 6 min ($\gamma_{MN} = 0.9916$).

(c) Aging function with T_{window} = 12 min ($\gamma_{MN} = 0.9958$).

Figure 3.12: MN aging functions based on different sliding time windows.

Finally, each connection needs to be associated with an aging function, a process that occurs using a normalization equation based on the five ranking intervals previously presented, and represented by (3.13), where k represents the time elapsed since the last contact with a Gateway or a MN. This equation merges both qualitative and quantitative parts of the ASAR ranking computation: inside the brackets, it refers to the qualitative part, where an increase in the number of contacts translates into an increase in the ranking computation, considering multiple node types and the time elapsed for each connection (k); the remaining formula outside the brackets represents the quantitative classification, where the five intervals in Table 3.1 classify the social viability of the node and 0.2 normalizes the classification.

$$Rank_{ASAR} = \begin{cases} \left(1 - \frac{1}{1+\sum_{i=1}^{NumMNs} \gamma^k_{MN(i)}}\right) \times 0.2 \times \frac{R_{interval}-1}{5} & , R_{interval} = \{1,2\} \\ \left(1 - \frac{1}{1+\sum_{i=1}^{NumRSUs} \gamma^k_{RSU(i)}}\right) \times 0.2 \times \frac{R_{interval}-1}{5} & , R_{interval} = 3 \\ \left(1 - \frac{1}{1+\sum_{i=1}^{NumMNs} \gamma^k_{MN(i)} + \sum_{i=1}^{NumRSUs} \gamma^k_{RSU(i)}}\right) \times 0.2 \times \frac{R_{interval}-1}{5} & , R_{interval} = \{4,5\} \end{cases} \tag{3.13}$$

Taking into consideration all configurable parameters (T_{window}, τ_{MN} and τ_{GW}), this forwarding strategy is highly dependent on the size of the network. Knowing the connectivity model, and how many of each type of nodes the network portrays, considerably facilitates the choice of such parameters. Even so, a good performance is expected for networks where nodes typically connect constantly, such as in smart city environments.

3.5.3 Hybrid between GLA and ASAR (HYBRID)

This forwarding strategy aims to combine a mobility contribution with the social behaviours, thus resulting in a smarter protocol. So, in terms of characteristics, all the previously presented features for GLA and ASAR are applied in this model. Therefore, mobile nodes calculate their rank in the network resorting to the expression:

$$Rank = W_{Mobility} \times Mobility_{Rank} + W_{Social} \times Social_{Rank} \tag{3.14}$$

where $Mobility_{Rank}$ represents the ranking obtained with GLA, and $Social_{Rank}$ represents the ranking for ASAR. $W_{Mobility}$ and W_{Social} represent the weights given to $Mobility_{Rank}$ and $Social_{Rank}$, respectively. This easy conversion is possible since both GLA and ASAR rankings are normalized between 0 and 1. Finally, even though this strategy is computational heavier, the improved intelligence level aims to reduce other resources expenditure such as diminishing the cases of unnecessary packet replication.

3.6 Proposed QoS Packet Selection

Since this architecture has a storage module, it is possible to chose among the packets in memory when forwarding situations occur. In this document two types of data packets will be considered: environmental data packets; and Gas data packets. In addition, all packets have a lifetime in the network to avoid indefinite storage congestion for nodes that do not contact frequently with gateways.

Choosing the packet to forward can greatly affect the networks' performance. The choice can be simple, such as choosing the first packet in storage (referred in this document as First Packet Selection) or involve further considerations. For instance, it enables the development of strategies that consider distinct bandwidth percentages for different types of packets (referred in this document as Distributed Packet Selection); allow higher fairness and network performance (referred in this document as Equalized Packet Selection).

3.6.1 First Packet Selection

This strategy implements a First In First Out (FIFO) scheme (Figure 3.13). When a packet is required, the first packet previously stored in the storage module will be elected. This implementation is a base approach and allows comparison with more elaborated strategies. Since no packet fields are used to make decisions, improvements are expected for more intelligent approaches.

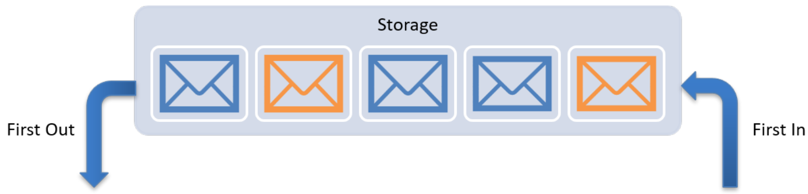
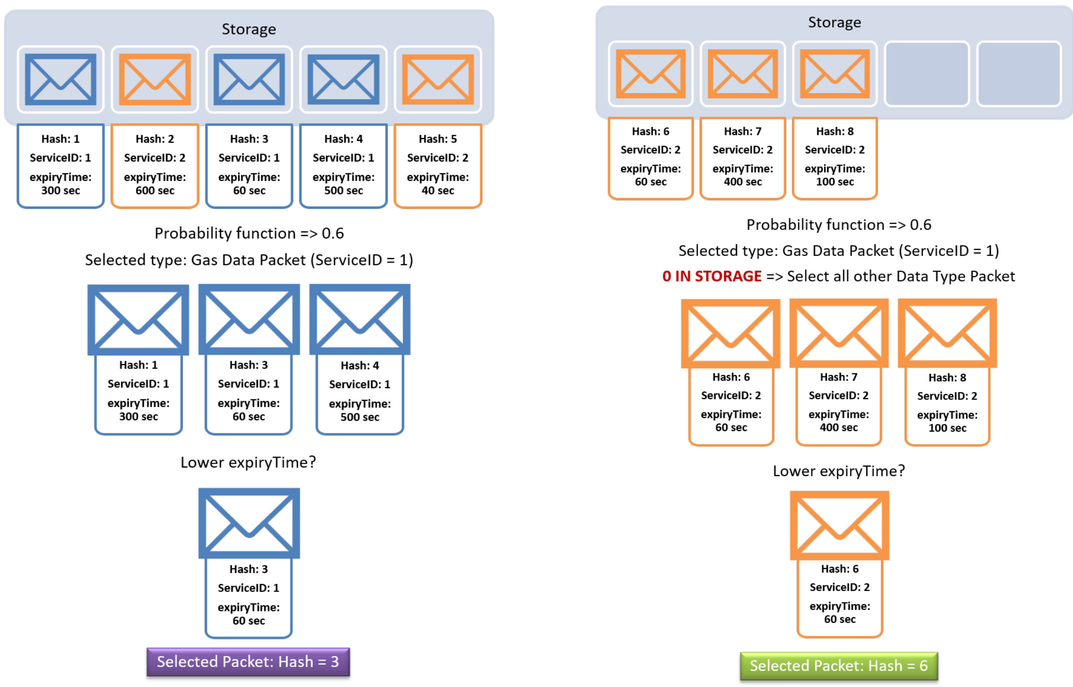


Figure 3.13: First Packet Selection example.

3.6.2 Distributed Packet Selection

In real applications, some types of information are more important than others. Taking this into consideration, this approach aims to allocate distinct percentages of bandwidth based on the packet type (*ServiceID*) and packet lifetime (*expiryTime*). In order to allocate the traffic percentages, two steps are required. First, a probability value is used to determine the type of packet. To allow different percentages, the probability should benefit those packets. Second, from the list of packets with that *ServiceID*, the one with the lowest network lifetime is chosen, thus granting those packets a higher probability of meeting their destination. The described procedure is the optimum case for this strategy, displayed in Figure 3.14a. However, if no packets with that *ServiceID* exist in storage and other types do, their selection is still safeguarded, where the tiebreaker rule is only the packets lifetime as illustrated in Figure 3.14b.



(a) Storage with multiple packet types.

(b) Storage with single packet type.

Figure 3.14: Distributed Packet Selection example.

3.6.3 Equalized Packet Selection

This strategy focuses on giving a fair weight to diverse network metrics and also improve the network performance. The packet tiebreaker metrics are the *ServiceID*, *expiryTime* and number of hops. The *ServiceID* field will allow some level of traffic percentage control. The *expiryTime* field will allow that packets with lower lifetimes can maximize their chances to reach their destination. The number of hops will allow low spread packets to maximize their chances of transmission and well spread packets to be contained. The computations are made in an weighted manner, where each packet will have a rank in the storage given by:

$$
\begin{aligned}
Rank =& \alpha \times \left(1 - \frac{expiryTime - currentTime}{maxTimeToLive}\right) + \\
& \beta \times \left(1 - \frac{packetHops}{maxNumberOfHops}\right) + \gamma \times packetTypePriority
\end{aligned}
\tag{3.15}
$$

where the $currentTime$ represents the instant when the packet is picked from storage ($[expiryTime - currentTime]$ represents the packets lifetime); the $maxTimeToLive$ represents the maximum lifetime of that type of packet in the network; the $maxNumberOfHops$ represents the maximum number of hops that a packet can travel in the network; the $packetHops$ represents the current number of hops of the packet; the $packetTypePriority$ represents the traffic type priority in the network; and α, β and γ represent the weights given to each of the components. Figure 3.15 demonstrates the described procedure, where a ranking is calculated for the packets in storage considering the described fields and the one with higher ranking selected.

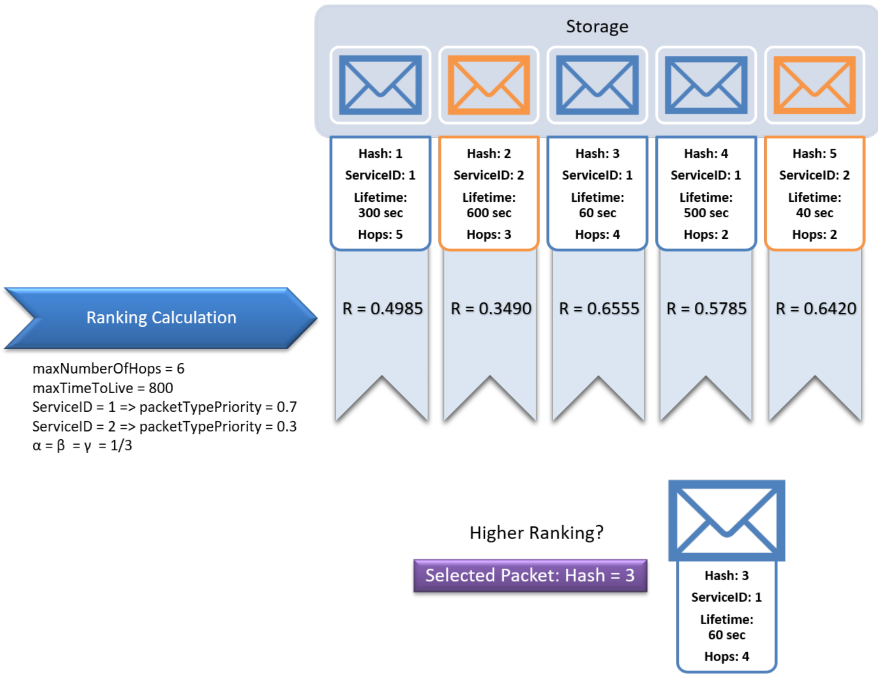


Figure 3.15: Equalized Packet Selection example.

3.7 Chapter Considerations

This chapter summarized the proposals addressed in this thesis. Firstly, an overview of the architecture was presented, with the respective characteristics. From the point of view of data gathering, while LoRa provides a low but steady traffic data gathering, DTN gathers and collects the data in bursts. These two technologies complete each other, one providing a constant data flow and the other granularity. The various network elements were presented and the interaction between them was explained. Finally, the proposed mechanisms to improve the forwarding decisions were presented, as well as QoS differentiation.

Chapter 4

Implementation and Integration

4.1 Introduction

This chapter aims to explain how the work proposed in this dissertation was implemented and integrated.

Section 4.2 presents the neighborhood mechanism that makes use of BLE to support communications between DCUs and mobile nodes.

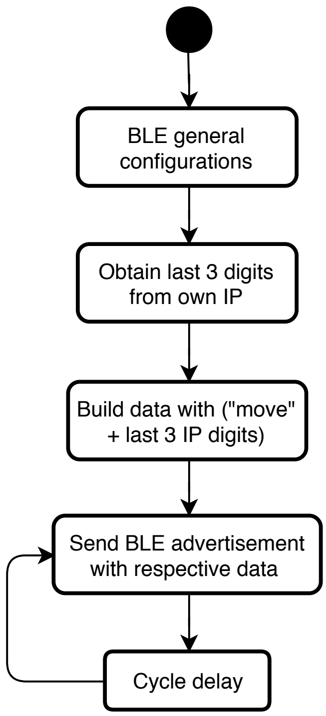
Section 4.3 describes the implementation of features in the mOVE architecture required to support the proposed forwarding strategies. Although the interaction between modules allows the overall forwarding decision in the network, this section is divided into several parts, providing a direct analysis over the distinct functionalities addressed.

Section 4.4 overviews the chapter summary and considerations.

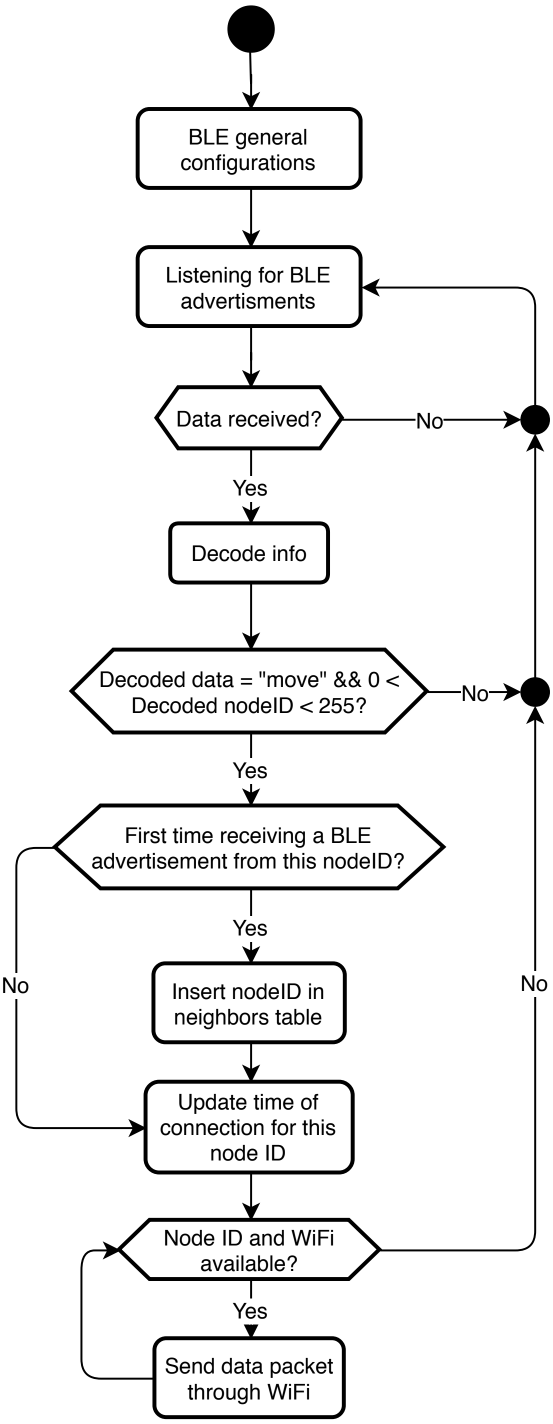
4.2 Network BLE advertisement software

To establish a connection between DCUs and mobile nodes in such a way that other network elements can still interact with DCUs, this type of network element is not inserted into the mOVE architecture. Thus, any other type of node that aims to receive data collected by DCUs only needs to support the network BLE advertisement software. Mobile nodes advertise their presence in a repetitive manner, as illustrated in Figure 4.1a. The information contained in the BLE advertisement characterizes the node in the network. For instance, a string with "move" and the last three digits of the mobile node Internet Protocol (IP) are part of the information. "move" identifies the mobile node as a member of the DTN, and the last three digits of the mobile node IP identify the node where the information is to be sent. As of now, the network address has a mask /24, where only the last three elements are required for the DCU to deliver the packets to that mobile node. Scalability is dependent on the network size, which means that more digits may need to be considered in the future.

On the other hand, Figure 4.1b represents the reception flow of a BLE advertisement. Copping with the mobile node side, the received information is decoded to understand if the node makes part of the network. In affirmative case, a neighborhood table is kept in order to forward the data through WiFi. While the connection is still viable, the DCU keeps delivering data packets to the mobile node neighbor. However, if the connection is interrupted, the DCU returns to the listen state waiting for BLE advertisements.



(a) send BLE advertisement flow in a mobile node.



(b) BLE advertisement handler flow in a DCU.

Figure 4.1: BLE advertisement flow.

4.3 DTN Features Integration

As presented and described in Section 2.2.2, mOVE is the selected DTN-based architecture to support delay tolerant communications in the platform. Data gathering through mOVE, designed to be versatile and modular, uses opportunistic communications that can be used to develop, rehearse and evaluate forwarding schemes.

A brief overview of new features added to mOVE are presented in Section 4.3.1. These features aim to support the development of new data gathering forwarding strategies, which are the core development of this thesis.

4.3.1 mOVE Features Overview

Figure 4.2 details the blocks that were further developed in order to support the new forwarding strategies. A general description of the development is provided below:

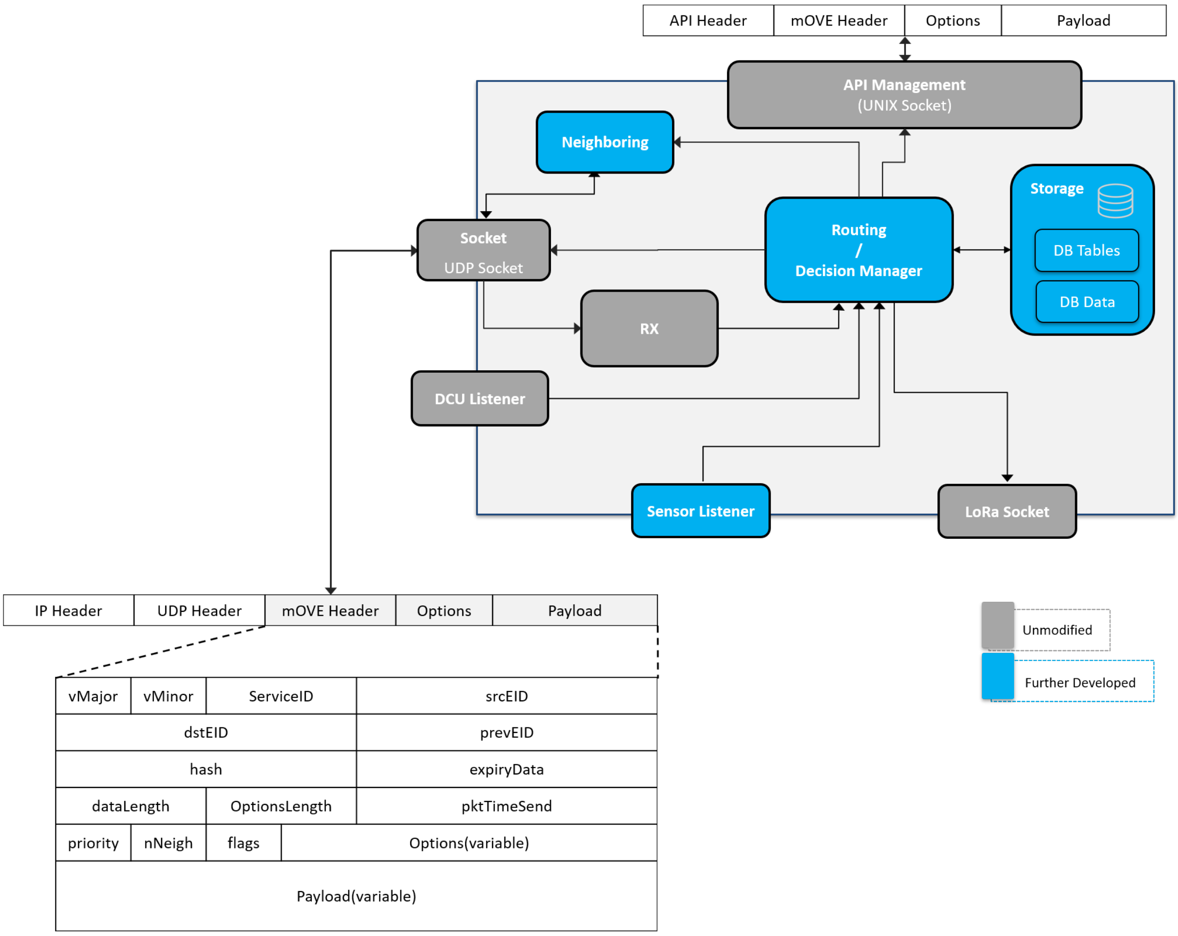


Figure 4.2: mOVE architecture.

- **DCU Listener**: when data packets are received from a DCU, it gets a packet type (ServiceID field) in the aggregating process between the sensed data and the mOVE header in order to create a mOVE packet.

- **Sensor Listener**: upon receiving data packets from the sensor controller, this module is now capable of giving a packet type (ServiceID field) when aggregating the sensed data with the mOVE header in order to create a mOVE packet, which is the same behavior as the module before, although the source of the data packets is different.

- **Routing/Decision Manager**: this module is responsible for obtaining all the information required to make the forwarding decision. Usually, the Storage module provides a packet, and the Neighboring module provides the neighbour. Moreover, extended mechanisms are considered in this module (further description in Section 3.5).

- **Neighboring**: it is responsible for all the knowledge about neighbors. This module provides all the algorithms and performs all the calculations to decide which is the best neighbor to forward the data based on different metrics (further description in Section 3.5).

- **Storage**: extensions to this module are made in the process of choosing which packet should be sent to the Routing module, providing differentiated quality of services (further description in Section 3.6).

mOVE Packet Structure

All packets exchanged in the DTN are mOVE packets, which are composed by a mOVE Header, an Options field and the Payload. Each field is illustrated in Figure 4.2, where the aforementioned fields have the following components:

→ **mOVE Header:**

  - vMajor and vMinor: mOVE Version;
  - ServiceID: Payload content type identification;
  - srcEID: Source node identifier;
  - dstEID: Destination node identifier;
  - prevEID: Previous custodian node identifier;
  - hash: packet unique identifier;
  - expiryData: time of creation + lifetime;
  - dataLength: Payload Length;
  - OptionsLength: Options Length;
  - pktTimeSend: Packet last temporal reference in which this packet was forwarded;
  - priority: Packet level of priority;
  - nNeigh: Current number of neighbors with this packet;
  - flags: Packet type identification (Data, ACK, Control);

→ **Options:** optional field, for extended options required such as the list of EIDs where the packets were received.

→ **Payload:** with a maximum size of 32KB, this field is where the raw data of interest is concatenated. If the field of Options is being used, the data length (payload size) is decreased by the *OptionsLength* size.

Distinct types of packets are considered in this architecture, being then differentiated by their respective *flags*. The packet types are as follows:

Data Packets: collected sensor data together with a mOVE Header and the Options field. These packets can cruise through all the network elements (in a multi-hop manner), or just be directly delivered from a DCU to a RSU.

Acknowledgement (ACK) Packets: when a RSU receives a data packet, it informs the network through ACKs packets. These packets' main function is to advertise mobile nodes that the RSU already has the packets, so they may free that data packet from their storage. However, ACKs are only transmitted in one hop, not advertising the whole network to avoid overhead enlargement.

Neighbor Announcement Packets: architectural support for communication between mobile nodes is mediated by these packets. Being transmitted in broadcast, these allow neighboring discovery. Coping with this, the Payload field can be used to transmit information between neighbors.

4.3.2 Data Type Classification

Taking into consideration that multiple types of packets are considered in this implementation, packet classification is required when entering the mOVE architecture. Two moments are key for this classification: when a mobile node meets a DCU, therefore receiving its data; and when the sensor controller module in the mobile node generates data packets. Respectively, it is also true for a RSU, if it contacts directly with DCUs, and if it acts as data sensing unit itself.

Figure 4.3 represents the operation flow of the DCU listener module, which is continuously listing for data packets forthcoming from DCUs. When a packet is received, a type classification is done depending on the data received. For instance, two types of data are considered: environmental and gas data. Respectively, distinct values for the *ServiceID* field are allocated, granting data type classification. Moreover, the maximum packet lifetime in the network is also parametrizable, which allows some differentiation in the network traffic types.

Figure 4.4 represents the operation flow of the sensor listener module, which is continuously listening for data packets forthcoming from the sensor controller module. If a mobile node has data collecting capabilities, this module takes care of the data type classification process, working similarly to the DCU listener module in terms of classification, allocating different *ServiceID*'s to distinct data types and allowing parametrizable packet type lifetime in the network.

Furthermore, this module is also responsible for updating the GPS information of the mobile node, such as Latitude, Longitude, Heading Angle and Velocity. This procedure only takes place when GLA or HYBRID forwarding strategies are running, since they require such information to make the respective forwarding decisions. Since the heading angle and the instant velocity are volatile measurements to some degree, an arbitrary number of measurements is considered to provide the mean value for such measurements.

As the aforementioned modules are extensible, more packet types can be easily integrated. In this dissertation, the two data types considered provide differentiation for the proposed packet selection algorithms, which will be described later.

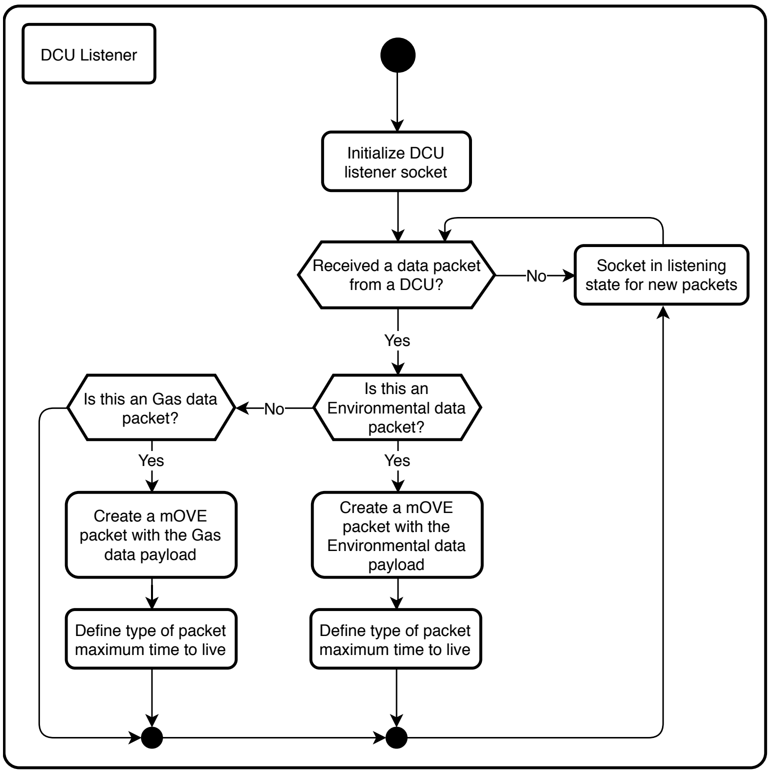


Figure 4.3: DCU Listener module flow.

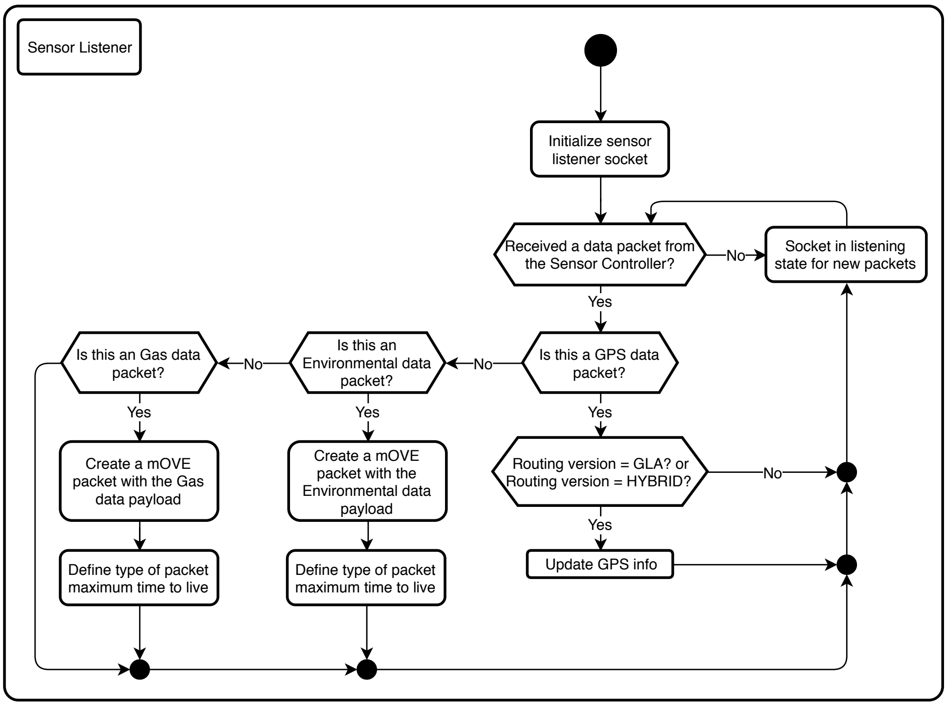


Figure 4.4: Sensor Listener module flow.

4.3.3 DTN Packet Reception Management

Every mOVE packet in the network is mediated through the UDP Socket module. mOVE packets are sent to other network elements through this module; however, the decision is taken in the Routing module, which will be addressed later in this document. The receiving end of packets in the DTN is done through the RX module.

Figure 4.5 represents the working flow of the RX module, whose main function is to deal with received mOVE packets. This module filters packets that were sent by the own node, using the *prevEID* field. Moreover, this module uses the *dstEID* field to understand if the packet was aimed for the respective node or to be forwarded again in the network. After these initial verifications, a packet classification mechanism - using the *flags* field - is used to deal with the different types of packets, namely: ACK packets, Neighbor announcement packets and Data packets. Respectively, it forwards each type of packet to its respective handler. The numbers in Figure 4.5 represent the flowcharts that further describe how each type of information is dealt.

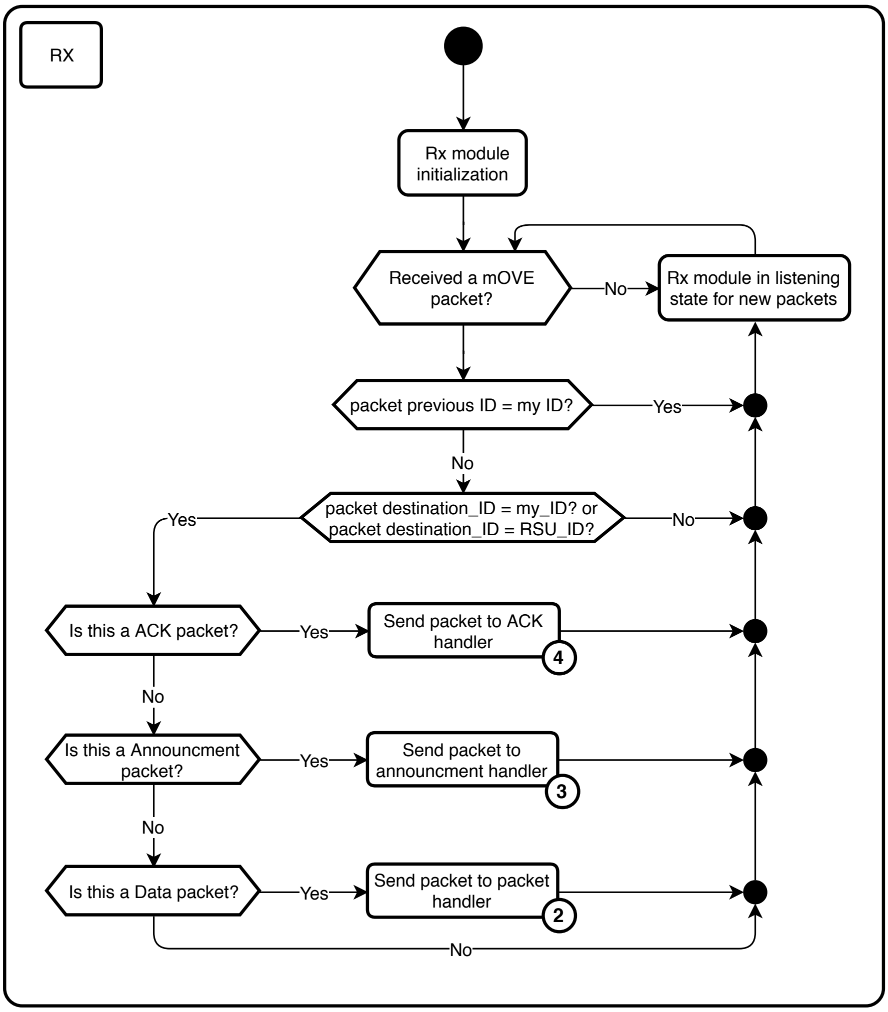


Figure 4.5: RX module flow.

The data packet handler execution depends on the type of network element. Mobile nodes want to store and carry to later forward these packets. On the other side, RSUs want to notify the network of their arrival (using ACK packets) and deliver them to the server.

For instance, Figure 4.6 represents how a mobile node deals with the reception of a data packet. By checking the packet *hash* field with the list of *hash*'s in storage, duplicated packets are eliminated. If it does not exist in the storage already, the list of hops table for that packet is updated based on the *Options* field and the packet is stored.

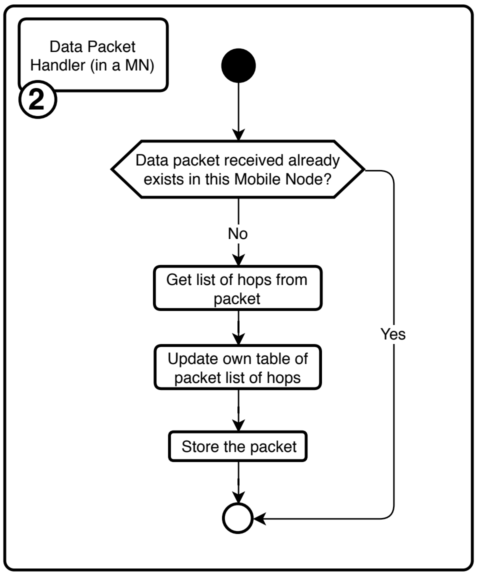


Figure 4.6: Data packet handler in a mobile node.

Figure 4.7 illustrates how a RSU handles the reception of a data packet. Similarly to a mobile node, this network element also checks if the received packet does not already exist in storage. The packet is stored, but in this case, an ACK packet is broadcasted in response, so that receiving nodes are aware of the packets arrival to the final element of the data gathering process. Since a network element was responsible for sending the data packet, the network is always notified - with an ACK packet - even when a RSU already has the packet in storage.

When a mobile node receives an ACK packet, the Figure 4.8 working flow is executed. The objective is to free the packets from storage that were already received by a gateway. To do so, the *hash* field is used to verify if the packet exists in storage and the packet is removed permanently. This process is only done in one hop, which means that only nodes in contact with the RSU will be notified with ACK (propagating the ACK further in the network would inflict undesired overhead). For those not in range, other mechanisms are considered, such as: finite packet lifetime and the historical ranking storage congestion management.

Despite *neighbour announcement* packet handler constituting an integrating part of this subsection, Section 4.3.4 will address the implementation and behaviour of these special messages.

4.3.4 Control Packets Management

Control packets are exchanged between nodes regardless of the forwarding strategy running. These have the name of *neighboring announcement* packets and are used primarily for one reason: network nodes announce their presence in the network so that neighbors are

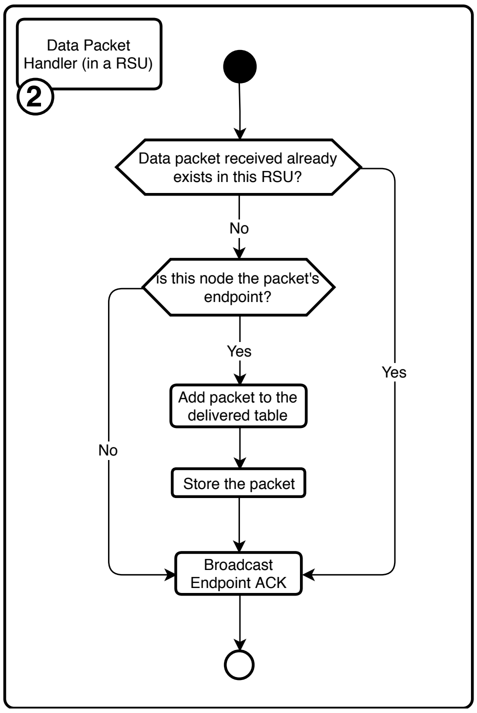


Figure 4.7: Data packet handler in a RSU.

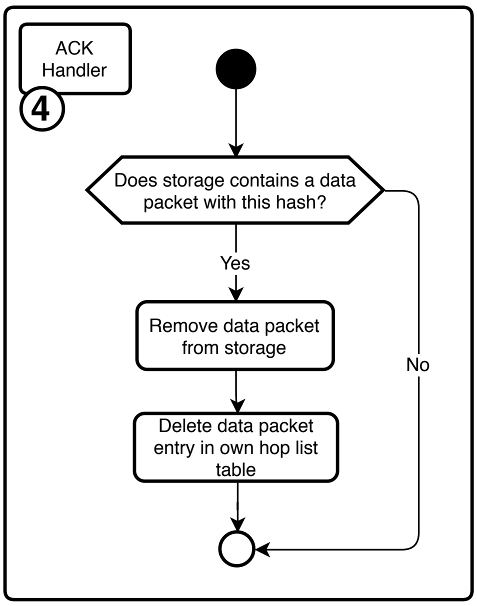


Figure 4.8: ACK handler in a mobile node.

aware of their presence. Since these packets are mOVE packets, the payload is used in some forwarding strategies (GLA, ASAR and HYBRID between GLA and ASAR (HYBRID)) to transmit information between nodes, namely: the node's respective rank and historical rank. Figure 4.9 illustrates the neighbor announcement flow from the sender side. The algorithms computing the ranks will be presented in Section 4.3.6.

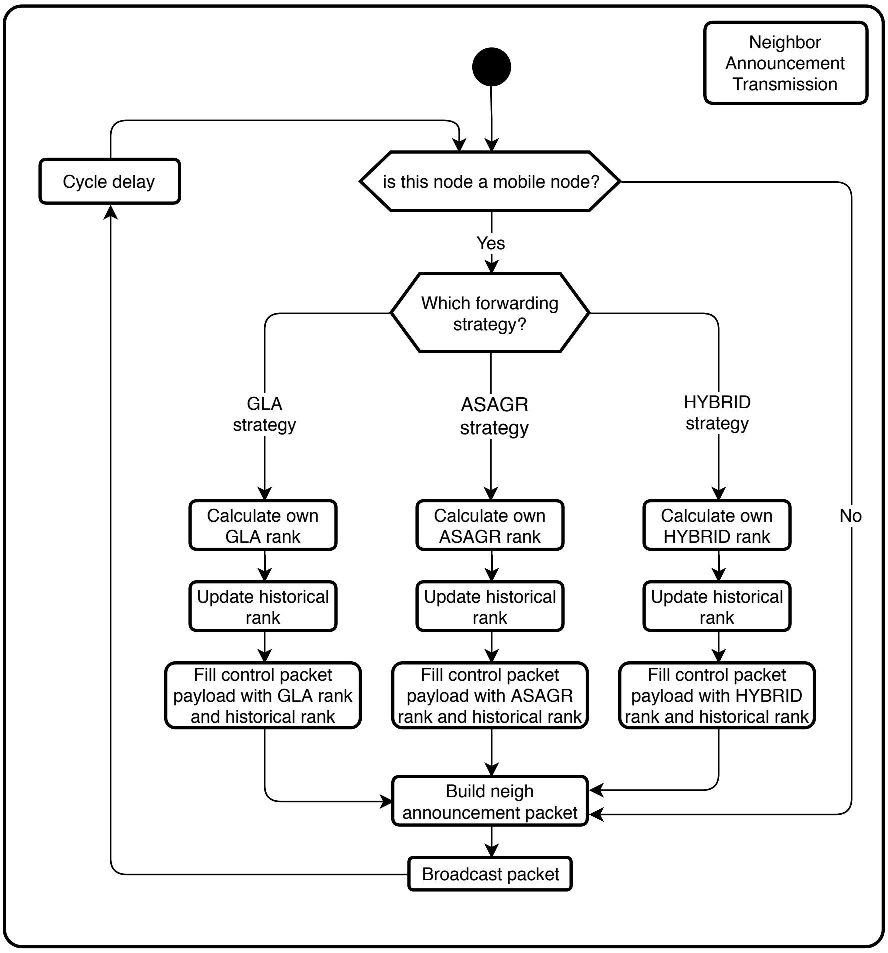


Figure 4.9: Neighbor announcement transmission flow.

Figure 4.10 illustrates the neighbor announcement flow from the receiver side, where a neighborhood table is updated and node classification type is ensured. When the information is received from a mobile node, depending on the forwarding strategy, the *neighrank* and historical rank are updated. If the forwarding strategy differs from GLA, ASAR or HYBRID, only the neighborhood information is updated.

4.3.5 Packet Selection Algorithms

This dissertation proposes several packet selection algorithms, as described in Section 3.6. The packet selection mechanism aims to enhance the network performance when dealing with distinct data types, which is now possible in this architecture as explored in Section 4.3.2. Therefore, three distinct packet selection mechanisms were implemented, of which one was

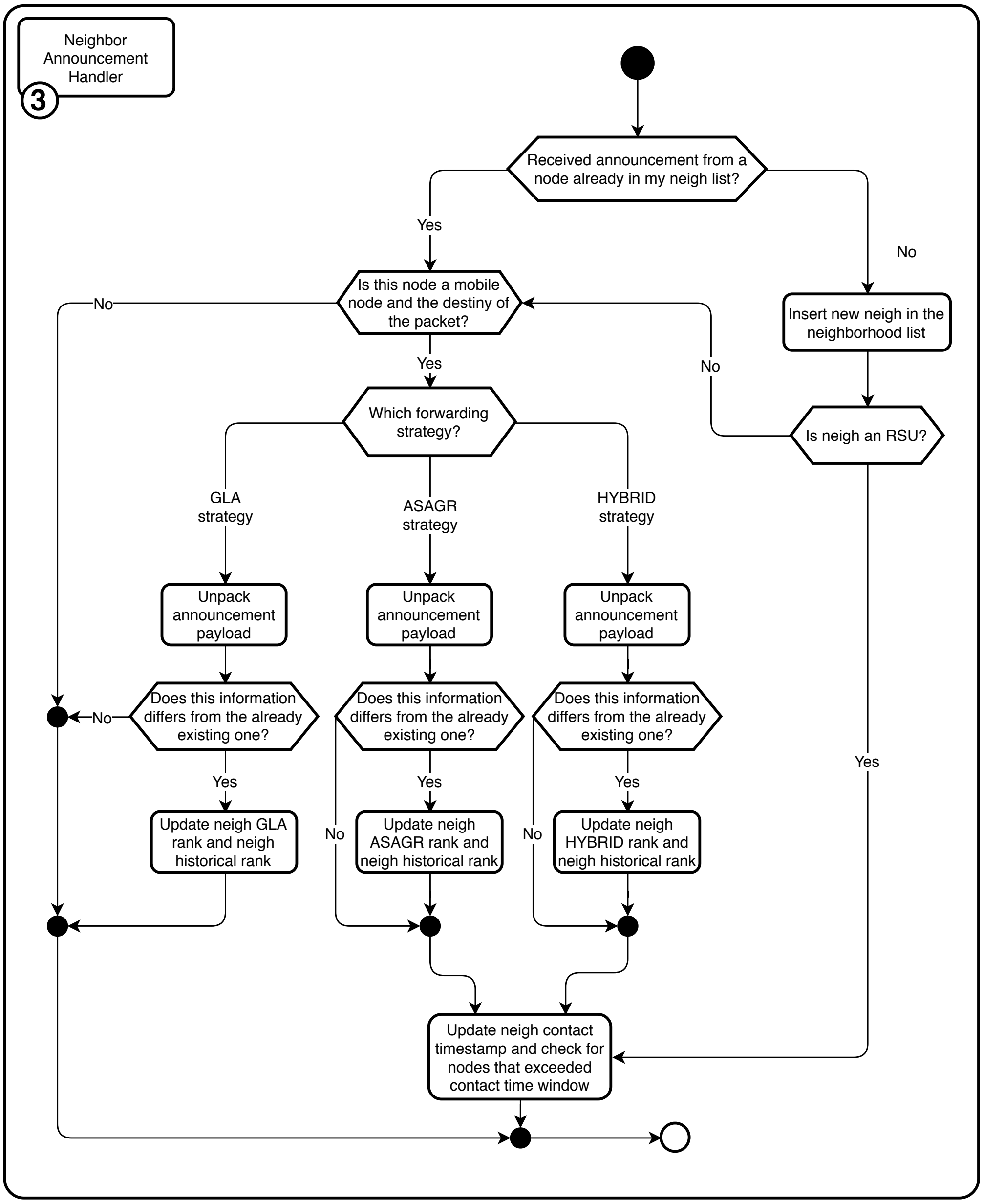


Figure 4.10: Neighbor announcement handler flow.

implemented as baseline: First Packet Selection. The two remaining algorithms enforce packet selection based on the packet type, where the Distributed Packet Selection algorithm is more strict on the data type prioritization, and the Equalized Packet Selection algorithm performs some degree of data type selection while keeping a level of fairness using a packet ranking classification.

It is relevant to mention that all algorithms will first obtain the packet's list and check if its size is higher than 0, i.e., if there are any packets to be forwarded.

4.3.5.1 First Packet Selection

Implementing a FIFO, the packet selection is quite simple. Algorithm 1 illustrates the implementation of this technique. After understanding if any packets exist in storage (line 3), the selected packet is always the first packet that entered the node's storage (line 4).

Algorithm 1 First packet selection algorithm

1: **procedure** GETFIRST($Packet^*$)
2: $PacketList \leftarrow getPacketsList()$
3: **if** $PacketList_{Size} \neq NULL$ **then**
4: $Packet^* \leftarrow PacketList^*_{Begin}$
5: **return** $True$
6: **end if**
7: **return** $False$
8: **end procedure**

4.3.5.2 Distributed Packet Selection

This packet selection strategy, which is implemented through Algorithm 2, aims to provide higher probability of choice to one data type over the other. To implement this algorithm, two packet fields are used, namely: $ServiceID$ for data type differentiation; and $expiryTime$ to determine the packets' lifetime in the network.

Before the list of packets starts to be exploited, a random value between 0 and 1 is generated. If the respective value is lower than the $DataProbThreshold$, then the selected packet is an Environmental data packet (line 8). Otherwise, the system picks a Gas data packet (line 14). With the random calculation, a probability heuristic is generated for the data type selection.

After such mechanism, the decision factor relies on the network lifetime for each packet (lines 9 and 15), where packets with lower lifetimes are prioritized, giving them higher probability to eventually being delivered to a gateway.

There are cases where the random value selected a data type that does not exist in storage (line 27), and the other packet types can still be selected where the only decision metric is the packet's lifetime (line 28).

4.3.5.3 Equalized Packet Selection

In order to provide higher levels of fairness in packet selection and still provide some sort of data type priority, a packet ranking system was implemented. Also, this strategy aims to improve the networks' delivery ratio since the selection will be based also on the network

Algorithm 2 Distributed packet selection algorithm

```
procedure GETDISTRIBUTED(Packet*)
    PacketList ← getPacketsList()
    foundFlag ← False
    if PacketList_Size ≠ NULL then
        Packet* ← PacketList*_Begin
        random ← rand()
        while packet P ≠ PacketList_End do
            if P_ServiceID = EnvironmentDataType ∧ random ≤ DataProbThreshold then
                if P_ExpiryTime ≤ LowerPacket_ExpiryTime then
                    LowerPacket_ExpiryTime ← P_ExpiryTime
                    Packet* ← P*
                    foundFlag ← True
                end if
            else if P_ServiceID = GasDataType ∧ random > DataProbThreshold then
                if P_ExpiryTime ≤ LowerPacket_ExpiryTime then
                    LowerPacket_ExpiryTime ← P_ExpiryTime
                    Packet* ← P*
                    foundFlag ← True
                end if
            end if
            NextPacket
        end while
        if foundFlag = True then
            return True
        end if
        Packet* ← PacketList*_Begin
        while Packet ≠ PacketList_End do
            if P_ExpiryTime ≤ LowerPacket_ExpiryTime then
                LowerPacket_ExpiryTime ← P_ExpiryTime
                Packet* ← P*
                foundFlag ← True
            end if
            NextPacket
        end while
        if foundFlag = True then
            return True
        end if
    end if
    return False
end procedure
```

deepness of each packet. Algorithm 3 illustrates the implementation of this selection mechanism, where three packet fields are used, namely: *ServiceID* for data type differentiation, *Options* field, to determine the packets number of hops, and *expiryTime* to determine the packets lifetime in the network.

Data type prioritization is done directly through the interpretation of each packet type. The number of hops of a packet describes how distributed a packet is in the network. So, packets with lower number of hops have higher probability over packets with high number of hops. Furthermore, the network lifetime of a packet is considered, to enhance the probability of a packet with lower lifetime to reach a gateway.

For each packet in the packet list, a rank is calculated based on the three aforementioned metrics (line 7). The rank classification system is based on (3.15), which is done in a weighted manner, which allows balancing the contribution of each parameter.

Algorithm 3 Equalized packet selection algorithm

1: **procedure** GETEQUALIZED($Packet^*$)
2: $HighestPacketRank \leftarrow 0$
3: $PacketList \leftarrow getPacketsList()$
4: **if** $PacketList_{Size} \neq NULL$ **then**
5: $P^* \leftarrow PacketList^*_{Begin}$
6: **while** packet $P \neq PacketList_{End}$ **do**
7: $P_{Rank} = getPacketRank(P_{ServiceID}, P_{ExpiryTime}, P_{Hops})$ ▷ Equation 3.15
8: **if** $P_{Rank} > HighestPacketRank$ **then**
9: $HighestPacketRank \leftarrow P_{Rank}$
10: $Packet^* \leftarrow P^*$
11: **end if**
12: NextPacket
13: **end while**
14: **return** $True$
15: **end if**
16: **return** $False$
17: **end procedure**

4.3.6 Delay Tolerant Network Forwarding Strategies

In the development course of this dissertation, several forwarding strategies were proposed, as presented in Section 3.5. To evaluate their performance, they were implemented and integrated in the mOVE architecture. Two extra base forwarding strategies were also considered, namely Epidemic and Direct Contact. These two strategies represent both ends of the spectrum in terms of delivery ratio and network delay versus network resources expenditure (network overhead).

4.3.6.1 Epidemic Strategy

In Epidemic routing, nodes replicate packets in a flooding manner, maximizing the delivery ratio and minimizing the network delay, but expending an excessive amount of network resources.

Figure 4.11 presents the working flow implementation for this strategy. Since this scheme aims to maximize the delivery ratio of the network and minimize the delay of each packet, the packets are broadcasted to the entire neighborhood excluding the case where a gateway is found in the neighborhood. Since some nodes are not good neighbors, it is expected that an excessive amount of network resources are expended with no real contribution for the networks performance.

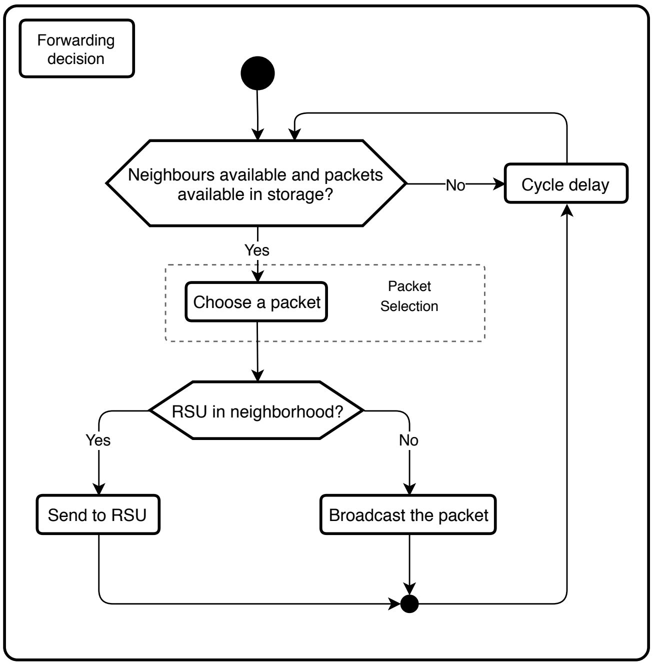


Figure 4.11: Epidemic flow from the sender side.

4.3.6.2 Direct Contact Strategy

In this forwarding scheme, there is no replication of packets between mobile nodes, i.e., packets are directly delivered to a gateway in a single hop. Obviously, the resource expenditure will be minimum, but the delivery ratio is low and the packet delay is high for this strategy.

Figure 4.12 represents the working flow for this strategy. As expected, only mobile nodes contacting with gateways can effectively deliver data packets. Nevertheless, for scenarios with high density of gateways and limited storage size, it may constitute a good option.

4.3.6.3 Neighborhood Selection Strategies

All the proposed forwarding strategies share the same vision, where each node has a rank that classifies how *good* a neighbor is. Thereby, all of the strategies exchange their respective ranks and historical ranks through *neighbor announcement* packets. Furthermore, all of them also share the same forwarding decision flow, illustrated in Figure 4.13, where the several mechanisms are implemented. The main focus of these mechanisms is solving some inherent problems associated with replication strategies, diminishing the network resources while maintaining high delivery ratio and low packet delay in the network.

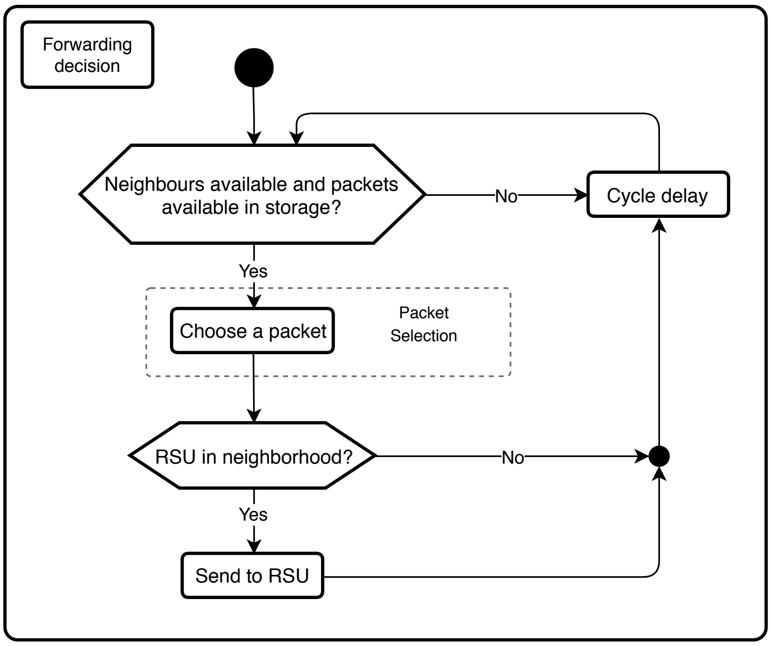


Figure 4.12: Direct Contact flow from the sender side.

By default, if a gateway is in connectivity range, packets are directly delivered to it. Otherwise, the Congestion Minimization mechanism is applied.

- **Congestion Minimization mechanism**: This mechanism makes use of a probability function, which determines the probability of a packet being transmitted according to the number of packet hops. Effectively, the number of hops characterize how spread a packet is in the network, where packets widely spread present a higher probability of being delivered. On the other hand, packet with low number of hops will not be found in many nodes. The probability value obtained from the number of packet hops is compared to a pseudo-random value (between 0 and 1), being the packet selected if it has a higher probability than the pseudo-random value.

Afterwards, a neighboring mobile node is selected, which in Figure 4.13 is presented with a respective number. This number represents the neighboring selection flow, represented in Figure 4.14.

In the process of selecting the neighbor, neighbors are sorted by its rank, from the highest to the lowest. This occurs because neighbors will be inspected iteratively, from the highest to the lowest rank. Coping with this, the Loop Avoidance mechanism is considered.

- **Loop Avoidance mechanism**: This mechanism makes use of the *Options* and *Options lenght* fields to store the packet's hops list, i.e., the previous nodes of this packet. With this information, when a forwarding decision is required, the nodes that already received this packet will not be selected. To support this mechanism, when a node receives a data packet, it updates the *Options* field with its own ID and the *Option Length* field is incremented by one. For instance, the *Options Length* field will directly inform the number of hops of the packet and the *Options* field the sequence of hop IDs. Since

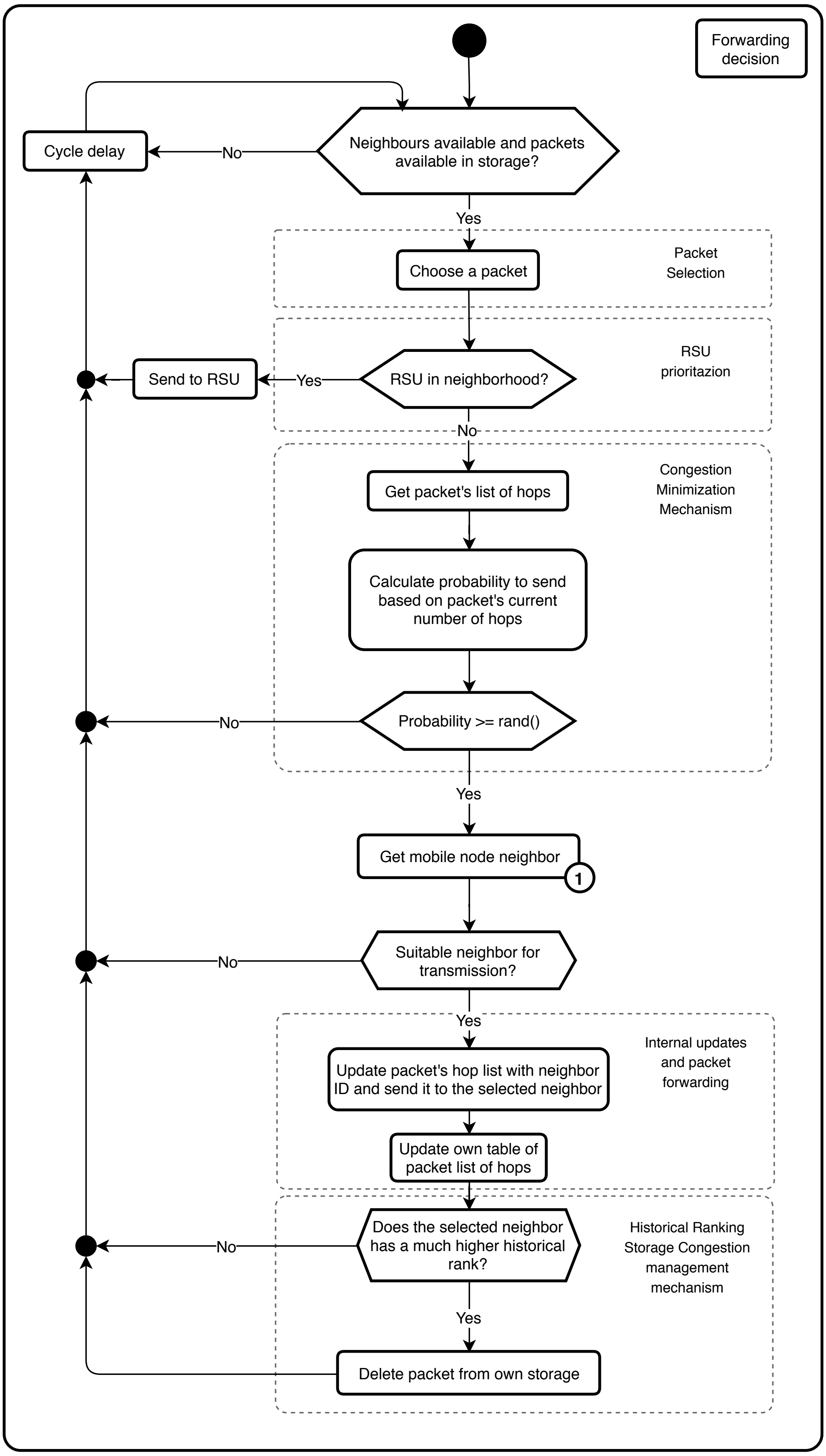


Figure 4.13: Proposed strategies forward decision flow from the sender side.

the Congestion Minimization mechanism already limits the number of hops of each packet, these fields' maximum growth is already limited. Furthermore, this mechanism requires a complementary table with the information of neighbors to which the packet was forwarded.

Two moments require the rank computation, when a *neighbor announcement* is being sent and when a node needs to compare its own rank to the neighbors. The three ranking algorithms will be presented after the general forwarding flow is fully described.

After the neighbor to forward a packet is selected, forwarding decision flow continues, deferring back to Figure 4.13. The packet and the neighbor are already selected at this point, which results in a packet replication. After the packet is sent, the receiver neighbor historical rank is compared to the forwarding node's one through (3.1), implementing the Historical Ranking Storage Congestion Management.

- **Historical Ranking Storage Congestion Management**: This mechanism aims to reduce the storage congestion, using the networks neighbourhood history. To do so, each node keeps track of its historical rank and informs its neighbors through the *neighbor announcement* packets. Previous to the development of this mechanism, packets could only be liberated from storage if: nodes receive the respective ACK packet from a gateway, or the packets lifetime expires. By enabling the deletion of packets after being received by nodes with higher network historical ranks, nodes can free space in their storage and trust the selected neighbor.

The implementation process of the three variants of network rank classification presented in Section 3.5 are now explained through the following algorithms:

Gateway Location Awareness (GLA): Algorithm 4 represents the practical implementation of this ranking calculation mechanism. This computation is done in a weighted manner and considers the best combination of distance and heading angle to the gateway, and the node's velocity. This algorithm considers all gateways' locations in order to compute the network ranking, as expressed in line 4. By default, only gateways distanced below the $maxD_{MG}$ are considered, resulting in lower computation effort since no time is lost in calculations for unrealistic gateways to deliver packets (lines 5 and 6). For the gateways of interest, the best combination of metrics determines the gateway considered for the ranking, which is the procedure done in lines 9, 10 and 11. If the distance to all gateways exceeds their maximum range, only the nodes' velocity is considered as a metric in the ranking calculation.

Aging Social-Aware Ranking (ASAR): Algorithm 5 represents the practical implementation of this ranking mechanism. This calculation considers how many contacts each node had in the predefined sliding time window (lines 9 and 12), the types of those contacts (lines 8 and 11) and the moment when both neighbors lost contact (lines 10 and 13). Nodes with a high number of contacts with gateways will have higher rankings. Also, having a high number of neighbors is also valuable (even if these are not gateways), since a node with more connections enhances the ramification delivery process. In line 17, the contacts in the sliding time window have their aging factor considered and the ranking is normalized.

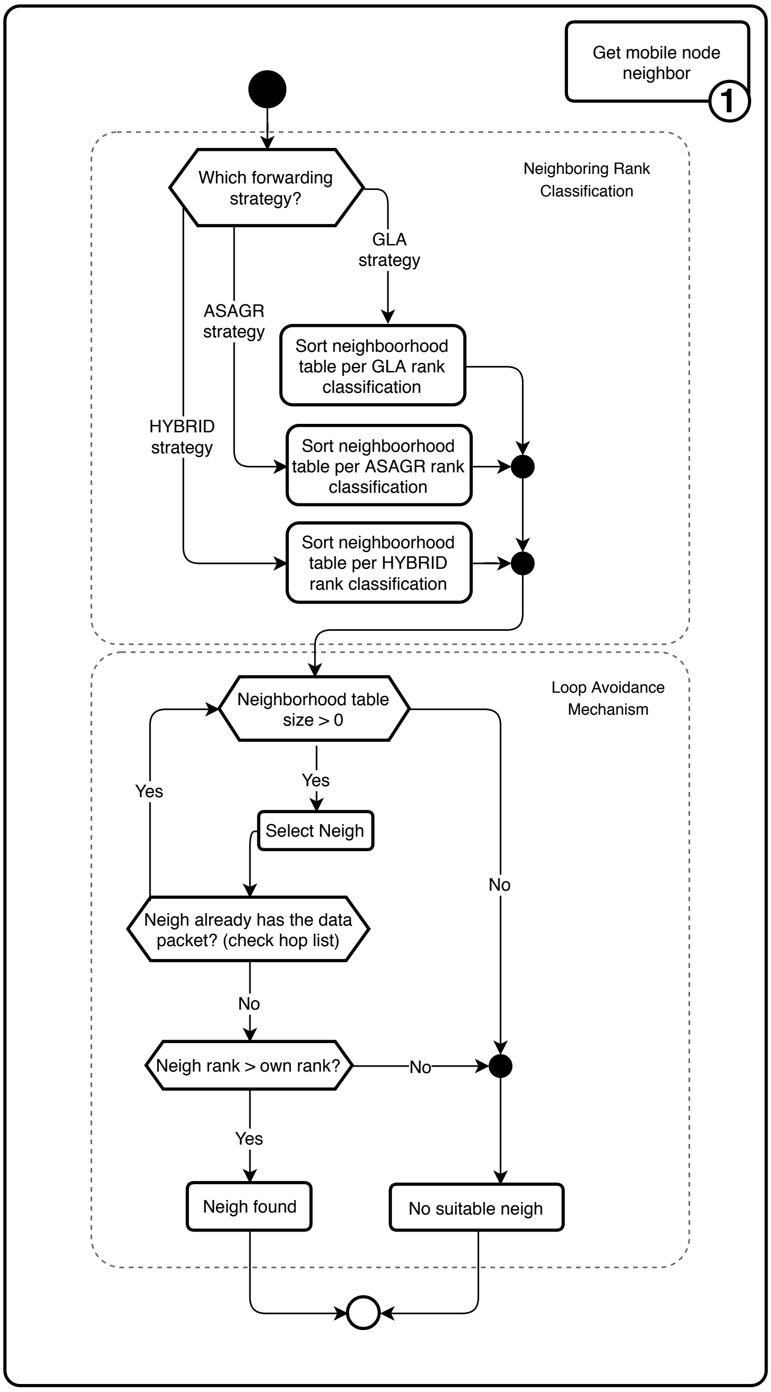


Figure 4.14: Select mobile node neighbor flow.

Algorithm 4 GLA rank calculation algorithm

procedure GETRANKGLA()
  $Normalized_{Velocity} \leftarrow getVelocityNormalized()$ ▷ Equation 3.9
  $\text{GLA}_{Rank} \leftarrow W_{Velocity} \times Normalized_{Velocity}$
  for GatewayLocations **do**
    $GW_{distance} \leftarrow calculateDistance(GW_{Latitude}, GW_{Longitude})$ ▷ Equation 3.2
    if $GW_{distance} < maxD_{MG}$ **then**
      $Normalized_{GatewayDistance} \leftarrow getDistanceNorm(GW_{distance})$ ▷ Equation 3.3
      $Normalized_{GatewayHeadingAngle} \leftarrow getHeadingAngleNorm()$ ▷ Equation 3.8
      $Current_{GatewayRank} \leftarrow W_{GatewayDistance} \times Normalized_{GatewayDistance} + W_{GatewayHeadingAngle} \times Normalized_{GatewayHeadingAngle} + W_{Velocity} \times Normalized_{Velocity}$
      if $Current_{GatewayRank} > \text{GLA}_{Rank}$ **then**
        $\text{GLA}_{Rank} \leftarrow Current_{GatewayRank}$
      end if
    end if
    NextGateway
  end for
  return GLA_{Rank}
end procedure

Algorithm 5 ASAR rank calculation algorithm

procedure GETRANKASAR()
  $sumRSU \leftarrow 0$
  $sumMN \leftarrow 0$
  $s_{GW} \leftarrow 0$
  $s_{MN} \leftarrow 0$
  $UpdateTimeWindow()$
  for DifferentContacts **do**
    if $ContactisRSU$ **then**
      $s_{GW} \leftarrow s_{GW} + 1$
      $sumRSU \leftarrow sumRSU + \gamma^{k}_{RSU}$
    else
      $s_{MN} \leftarrow s_{MN} + 1$
      $sumMN \leftarrow sumMN + \gamma^{k}_{MN}$
    end if
    NextContact
  end for
  $\text{ASAR}_{Rank} \leftarrow getContactsRankNorm(s_{GW}, sumRSU, s_{MN}, sumMN)$ ▷ Equation 3.13
  return ASAR_{Rank}
end procedure

Hybrid between GLA and ASAR (HYBRID): Algorithm 6 represents the practical implementation of this ranking calculation mechanism. This computation combines both mobility (line 27) and social metrics (line 19), therefore following the same algorithmic implementation for each individual computation. The contribution of each component is done in a weighted manner, as expressed in line 32. With a higher level of complexity and inherent intelligence, this algorithm is expected to minimize the number of network resources expended, while maintaining good levels of delivery ratio and packet delay.

Algorithm 6 HYBRID rank calculation algorithm

1: **procedure** GETRANKHYBRID()
2:     $Normalized_{Velocity} \leftarrow getVelocityNormalized()$ ▷ Equation 3.9
3:     $Mobility_{rank} \leftarrow W_{Velocity} \times Normalized_{Velocity}$
4:     $sumRSU \leftarrow 0$
5:     $sumMN \leftarrow 0$
6:     $s_{GW} \leftarrow 0$
7:     $s_{MN} \leftarrow 0$
8:     $UpdateTimeWindow()$
9:     **for** DifferentContacts **do**
10:         **if** $ContactisRSU$ **then**
11:             $s_{GW} \leftarrow s_{GW} + 1$
12:             $sumRSU \leftarrow sumRSU + \gamma^{k}_{RSU}$
13:         **else**
14:             $s_{MN} \leftarrow s_{MN} + 1$
15:             $sumMN \leftarrow sumMN + \gamma^{k}_{MN}$
16:         **end if**
17:         NextContact
18:     **end for**
19:     $Social_{Rank} \leftarrow getContactsRankNorm(s_{GW}, sumRSU, s_{MN}, sumMN)$ ▷ Equation 3.13
20:     **for** GatewayLocations **do**
21:         $GW_{distance} \leftarrow calculateDistance(GW_{Latitude}, GW_{Longitude})$ ▷ Equation 3.2
22:         **if** $GW_{distance} < maxD_{MG}$ **then**
23:             $Normalized_{GatewayDistance} \leftarrow getDistanceNorm(GW_{distance})$ ▷ Equation 3.3
24:             $Normalized_{GatewayHeadingAngle} \leftarrow getHeadingAngleNorm()$ ▷ Equation 3.8
25:             $Current_{GatewayRank} \leftarrow W_{GatewayDistance} \times Normalized_{GatewayDistance} + W_{GatewayHeadingAngle} \times Normalized_{GatewayHeadingAngle} + W_{Velocity} \times Normalized_{Velocity}$
26:             **if** $Current_{GatewayRank} > Mobility_{rank}$ **then**
27:                 $Mobility_{rank} \leftarrow Current_{GatewayRank}$
28:             **end if**
29:         **end if**
30:         NextGateway
31:     **end for**
32:     $\text{HYBRID}_{Rank} \leftarrow W_{Mobility} \times Mobility_{rank} + W_{Social} \times Social_{Rank}$
33:     **return** HYBRID_{Rank}
34: **end procedure**

4.4 Chapter Considerations

This chapter presented the implementation of the fundamental blocks required to support the work proposed for this dissertation. Firstly, the mechanism of BLE advertisement has presented, where the sender and receiver sides were described. Afterwards, the blocks of the DTN were explained, where the proposed packet selection techniques and forwarding strategies were implemented.

Chapter 5

Evaluation

5.1 Introduction

This chapter aims to validate the proposed solution, present the test scenarios and discuss the results. Both emulated and real scenarios were used to perform the evaluation.

Section 5.2 evaluates the energy consumption for the proposed BLE scanning mechanism.

Section 5.3 presents the evaluation results through emulations. First, the integration and complementary additions to support extensive protocol analysis done in the mobile Opportunistic Vehicular Emulator for Real Scenarios (mOVERS) emulator are outlined, followed by scenarios definitions and considerations. Finally, the evaluation done in several steps to evaluate the proposed forwarding strategies and packets selection mechanisms is addressed.

Section 5.4 validates the proposed forwarding strategies through a real use case deployment in the "Ria" of Aveiro.

Section 5.5 presents the chapter summary and considerations.

5.2 BLE Scanning Energy Evaluation

This section evaluates the energy expenditure of the neighboring discovery mechanism between a mobile node and a DCU. Two implementations are compared for DCU and mobile node discovery mechanisms: WiFi scanning, and the proposed BLE advertisement mechanism.

5.2.1 General Test Considerations

Since the evaluated base device is a Raspberry Pi 3, the voltage supply is 5 volts. To evaluate the power consumption, a HM8112-2 multimeter [73] in amperemeter mode was setup in series with the Raspberry's power supply port. This multimeter was configured with IEEE.–.8 Program (Talk Only Mode), which means that the current is measured every 100 ms.

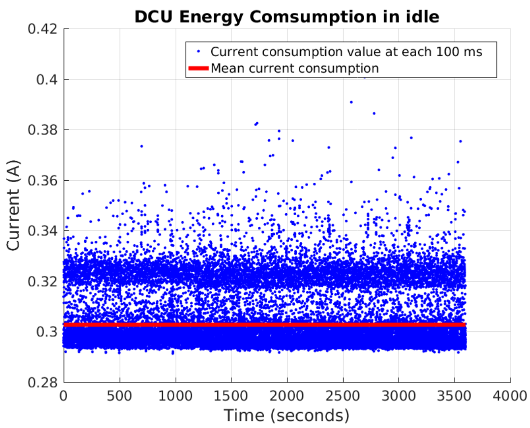
Effectively, in a real scenario, mobile nodes do not contact constantly with DCUs. Therefore, a mechanism was considered where mobile nodes and DCUs contact with a pseudo-random exponential distribution interval, meaning that both will contact in burst mode, with small contact windows followed by bigger periods of no connectivity. In practice, the tests were conducted with the mobile node and DCU in connectivity range, but the scanning mechanism was turned on or off according to the contact window or no connectivity, respectively.

However, even though the contact window and no connectivity intervals are pseudo-random, the same distribution was used in all evaluations providing a fair and seamless evaluation. The energy consumption tests were performed for 1 hour.

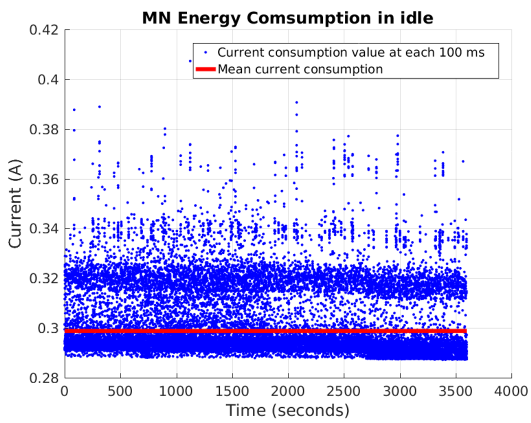
5.2.2 Obtained Results

As a ground rule and to better evaluate the energy consumption of each mechanism, both DCU and mobile node energy consumption's were measured in idle, i.e., with no contact mechanism software running.

Figure 5.1a and Figure 5.1b present the energy consumption (measured in amperes) of the DCU and mobile node, respectively. The mean consumption of each element is depicted in both figures and summarized in Table 5.1. For both network elements, the energy consumption is similar, with a small difference of around 1%. This can be explained through diverse reasons, such as: temperature of the processor, processes running in background, small differences in the Raspeberry's hardware components, among others.



(a) DCU energy consumption in idle.



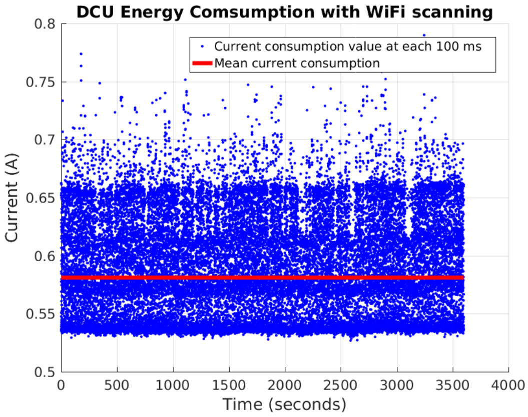
(b) MN energy consumption in idle.

Figure 5.1: RPi's energy consumption in idle.

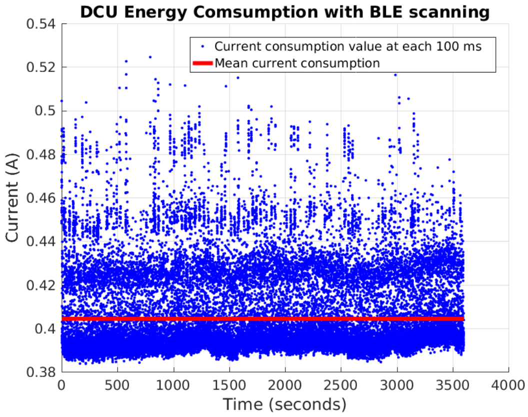
Node Type	Mean consumption (mA)
DCU (idle)	303
MN (idle)	299

Table 5.1: Idle mean energy consumption.

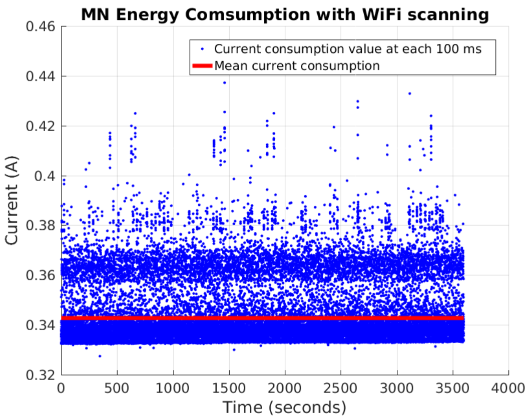
Figure 5.2a and Figure 5.2c illustrate the power consumption when the neighborhood scanning is done through the WiFi technology. On the other hand, Figure 5.2b and Figure 5.2d represent the power consumption of such mechanism using the BLE technology. Table 5.2 summarizes the mean energy consumption of each individual test. Looking at the mean consumption for the DCU element, the mean current differs by 177 mA, which is an improvement of around 30% in relation to the WiFi mean value. However, since the mobile node will also have the BLE active, an increase of 8 mA is observed for this element, which represents an increment of around 2% in relation to the WiFi mean value.



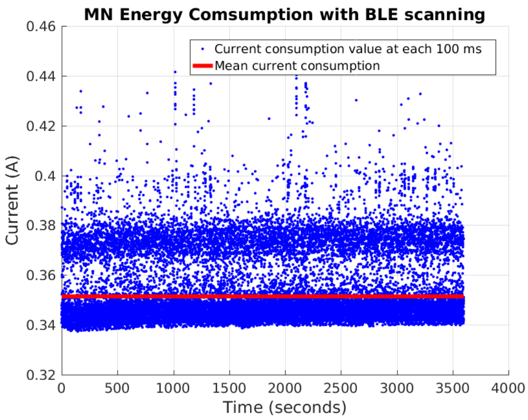
(a) DCU energy consumption running WiFi scanning.



(b) DCU energy consumption running BLE scanning.



(c) MN energy consumption running WiFi scanning.



(d) MN energy consumption running BLE scanning.

Figure 5.2: RPi's energy consumption with neighborhood scanning.

Node Type	Mean consumption (mA)
DCU (WiFi)	581
DCU (BLE)	404
MN (WiFi)	343
MN (BLE)	351

Table 5.2: Neighborhood scanning mean energy consumption.

This evaluation allows to understand an important aspect: the best neighborhood mechanism for the DCUs relies on the relation between the number of DCUs and the number of mobile nodes. If the relation between them is lower than 22 mobile nodes to 1 DCU, then the BLE scanning mechanism is better in terms of energy consumption. On the other hand, if it is bigger, the WiFi scanning mechanism becomes more energy efficient.

5.3 DTN Emulator Evaluation

This section focuses on evaluating the proposed forwarding strategies using the mOVERS emulator. An evaluation study of each parameter was made for each forwarding strategy to better understand its influence on the overall network performance.

5.3.1 mOVERS Emulator Integration

The evaluation process resorts to a vehicular network emulator, mOVERS [26], developed in *Instituto de Telecomunicações*. The emulator runs a DTN, which allows the creation of multiple processes that run DTN software, and is capable of scaling for larger networks with the same software as mOVE. Beyond using the same software as in real OBUs and RSUs, the emulator uses the extracted datasets, with real traces from vehicle's behavior and real connectivity maps (considering IEEE 802.11p/WAVE communication), reproducing exactly real events and real communication from vehicular networks. Therefore, this emulator provides a tool that allows to compare distinct strategies in the same scenario, which is very difficult to do in a real experiment.

The need to consider an emulator emerged due to the fact that the proposed forwarding strategies are constituted of a big number of parameters. Thus, to evaluate the behavior and influence of the proposed strategies and respective parameters, all the modules considered in Section 4.3 were also updated in the emulator, with the exception of the DCU listener and Sensor Listener modules. The referred modules do not exist in this emulator since it does not support DCU node types. Therefore, the packet generation and type classification were implemented using static node locations, as will be further described in this chapter.

Effectively, it is of high importance to mention that this emulator is only capable of evaluating the DTN network, not all the proposed architecture. Separating these contributions is also important, since it enables direct evaluation of this component with no external influence.

The access to the database is mediated by timestamp, where the information for each node in each time instance is obtained and the DTN work flow completed until the new timestamp information being required. During the experiment, each node collects its own mobility data and respective neighboring information per timestamp, already covering the radio-electric connectivity of each node and its neighbors. In order to integrate the proposed strategies and the emulator database, two requirements need to match: i) each node needs to update its own location, velocity and heading angle; ii) neighbor nodes need to exchange their ranking information.

The first requirement solution differs from the real implementation, where the Sensor Listener receives the information from a real sensor and updates the GPS information. In the emulator, the GPS information per timestamp is contained in the database. Thus, for each timestamp, each node updates its GPS information using the database information directly, i.e., it updates its own latitude, longitude, velocity and heading angle.

To address the second point, two options are possible: 1) use the neighboring information in the database to perform extended search for each neighbor information in order to calculate its rank and historical rank; 2) exchange control packets with the rank and historical rank information between the neighbors. In this work, the second solution was elected since it better resembles what happens in a real application, also constituting a more precise evaluation of the network overhead, since these packets contribute to the overhead.

As mentioned before, DCUs are not supported by this emulator. So, the solution was to keep a location file for each DCU considered. Each DCU has a storage file where the produced packets are kept until a mobile node is near its location, being able to collect its data. It is relevant to mention that this distance is configurable, being identified as $maxDCUconnectionRange$. When data is collected by a mobile node, it is erased from the storage file for that DCU, therefore not being available for other mobile nodes.

Other than collecting data from DCUs, it was also considered that each mobile node is capable of generating data. This is done in a periodic manner, which means that at periodic timestamps, a data packet is generated. The periodicity of generation for mobile nodes is identified by the configurable variable $MNgenerationPeriodicity$.

Finally, since neither the DCU Listener or Sensor Listener modules exist in this emulator, for the evaluation of the packet selection mechanisms, the packet type classification was made considering a probability heuristic. Therefore, a 50% probability was attributed to each type of packet generation, resulting in a half probabilistic type for the generated packets.

5.3.2 Scenarios definition and considerations

To evaluate the proposed forwarding strategies, seven scenarios were considered, whose properties will be described sequentially. Some properties are equal for all scenarios, namely:

- The connectivity dataset gathered from the devices installed in the bus network in Porto is updated every 2 seconds, and a full duration of 4 connectivity hours is considered in the simulation.

- The connectivity dataset uses 161 vehicles (OBUs) that perform the network function of mobile nodes.

- The connectivity model is supported by IEEE 802.11p/WAVE technology.

- The default mechanism used to choose a packet in the forwarding decision is Algorithm 1 (later substituted whilst evaluating the packet selection mechanisms).

- Environmental data type has a packet expiry time of 30 minutes and Gas data type has a packet expiry time of 15 minutes (also substituted whilst evaluating the packet selection mechanisms).

- DCUs generate data packets every 16 seconds ($DCUgenerationPeriodicity = 16$) and are detected by mobile nodes distanced under 100 meters ($maxDCUconnectionRange = 100$).

- Packets are only generated for the first 3 hours of simulation, meaning that all mobile nodes have 1 hour remaining to deliver their packets, thus avoiding the influence of those packets on the evaluation metrics.

The considered scenarios aim to implement distinct data gathering densities, where only DCUs collect data or both DCUs and mobile nodes generate data packets. Moreover, a scenario with a higher number of gateways is also evaluated.

Scenario 1: has the lowest number of generated packets from all the scenarios, where only DCUs generate data packets (mobile nodes do not generate data packets). The number of DCUs is 8, the same number of RSUs. Figure 5.3 illustrates the node's location. This scenario represents the case where higher number of hops is expected per packet, since the mean distance between DCUs and RSUs is the biggest among all the scenarios.

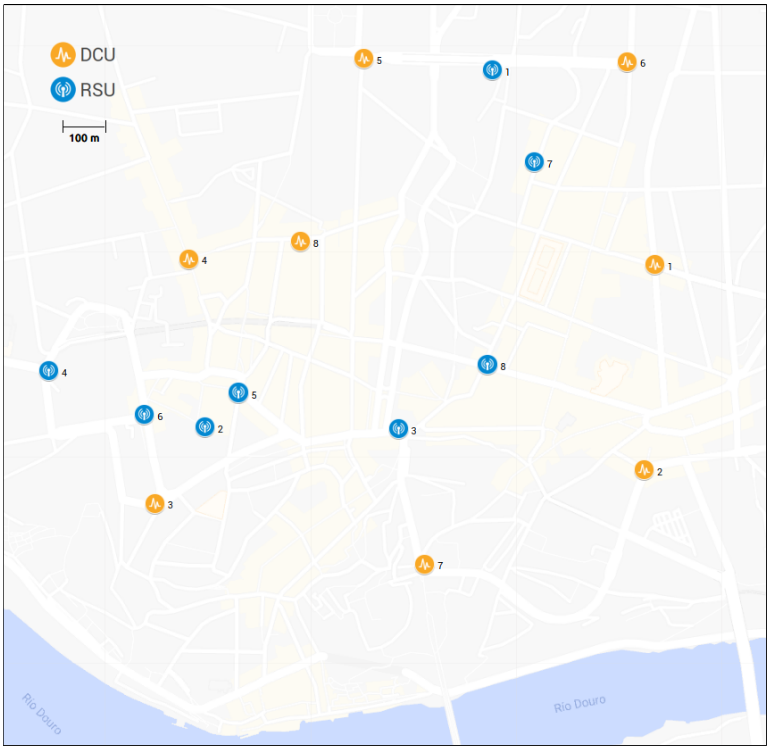


Figure 5.3: 8 RSUs and 8 DCUs locations (Scenario 1 and 4).

Scenario 2: only DCUs generate data packets (mobile nodes do not generate data packets). The number of DCUs and RSUs is 12 and 8, respectively. Figure 5.4 presents the location of these nodes. This scenario represents a mean density of data gathering considering only DCUs as collecting devices.

Scenario 3: only DCUs generate data packets (mobile nodes do not generate data packets). The number of DCUs and RSUs is 16 and 8, respectively. Figure 5.5 presents the location of these nodes. This scenario represents a high density of data gathering considering only DCUs as collecting devices.

Scenario 4: this scenario uses the same number of DCUs and RSUs as scenario 1, referring to the same node location distribution in Figure 5.3. However, for this scenario both DCUs and mobile nodes generate data packets ($MNgenerationPeriodicity = 60$ sec).

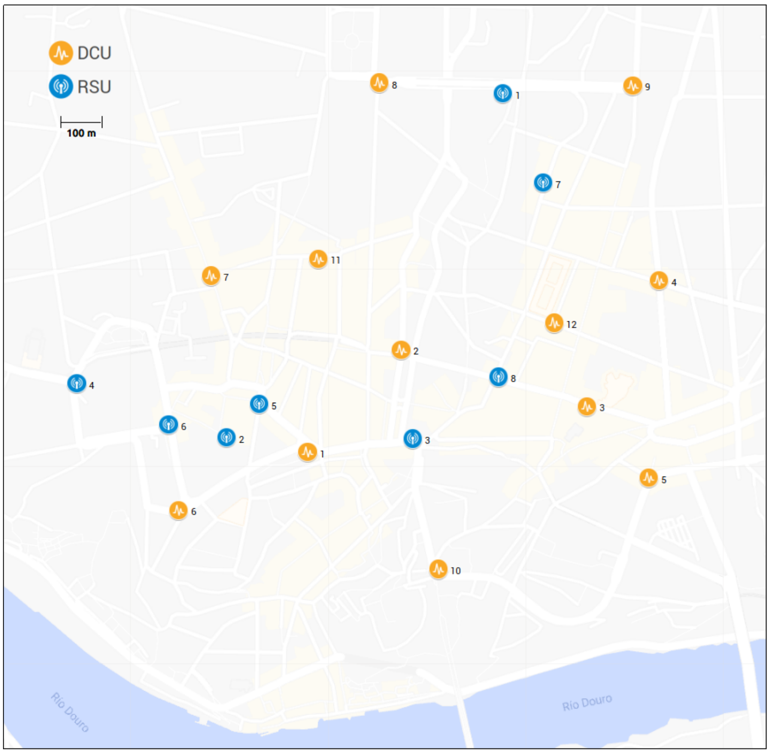


Figure 5.4: 8 RSUs and 12 DCUs locations (Scenario 2 and 5).

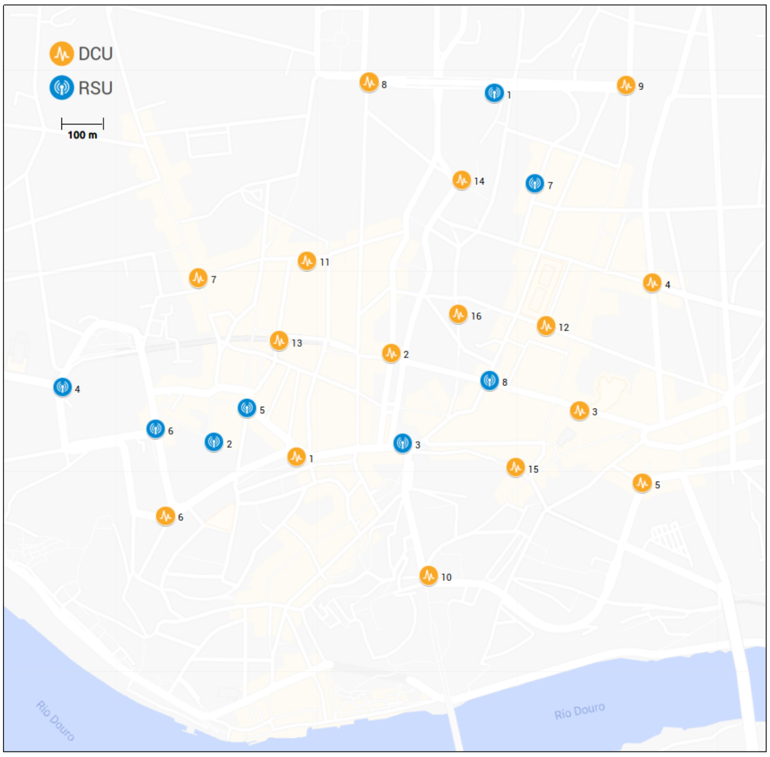


Figure 5.5: 8 RSUs and 16 DCUs locations (Scenario 3 and 6).

Scenario 5: this scenario uses the same number of DCUs and RSUs as scenario 2, referring to the same node location distribution in Figure 5.4. However, for this scenario both DCUs and mobile nodes generate data packets ($MNgenerationPeriodicity$ = 60 sec).

Scenario 6: this scenario uses the same number of DCUs and RSUs as scenario 3, referring to the same node location distribution in Figure 5.5. However, for this scenario both DCUs and mobile nodes generate data packets ($MNgenerationPeriodicity$ = 60 sec). Since the higher number of DCUs is considered in this scenario and mobile nodes generate packets periodically, this is the scenario with the most generated packets in the network.

Scenario 7: this scenario uses the same number of DCUs and the same type of generation as scenario 1. However, in this scenario, the number of RSUs is higher, a total of 12. Figure 5.6 presents the location of DCUs and RSUs. It is expected that the delivery ratio in this scenario is higher than in scenario 1, since more gateways are available to receive the data packets.

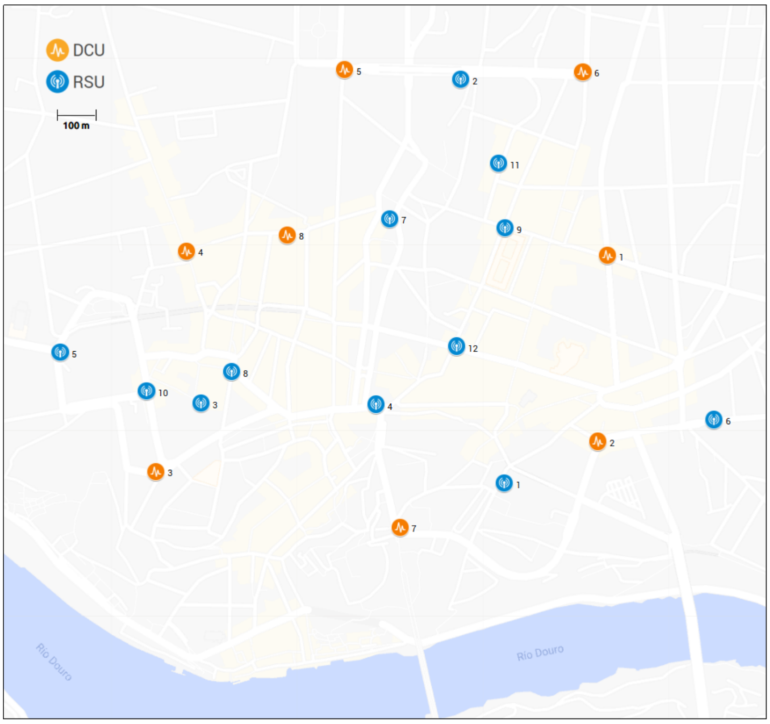


Figure 5.6: 12 RSUs and 8 DCUs locations (Scenario 7).

Table 5.3 summarizes the adjacent properties of all evaluation scenarios.

The evaluation is performed in several phases: first, considering scenario 1, an evaluation of the parameters of each forwarding strategy is done. Afterwards, for the selected parameters, each scenario is evaluated for all the forwarding strategies. Finally, for the scenario with the most generated packets (scenario 6), and using the strategy parameters with the better results, the proposed packet selection mechanisms are evaluated.

Scenario	n of RSUs	n of DCUs	MNs generate packets?
1	8	8	No
2	8	12	No
3	8	16	No
4	8	8	Yes
5	8	12	Yes
6	8	16	Yes
7	12	8	No

Table 5.3: Scenario properties summary.

5.3.3 Evaluation of Parameters of the Forwarding Strategies

In the first phase, the proposed forwarding strategies parameters are evaluated. To do so, three metrics are used: the delivery ratio, the network delay and the network overhead. The objective is to better understand which is the best combined fit between all metrics for all the experiments. In this subsection, the relevant results are shown in table format, where the best and worst metric values are highlighted in green and red, respectively. Scenario 1 is considered for the strategy parameterization.

5.3.3.1 GLA parameters evaluation

To assess the parameterization of the GLA algorithm, some configurations need to be set, namely: the maximum communication range, set to $T_{maxR} = 800$ meters (IEEE 802.11p/WAVE) and the maximum distance of interest, set to $maxD_{MG} = 2000$ meters; the mean velocity, set to $V_{avg} = 13.9$ m/s (around 50 km/h); and finally, the heading angle attenuation constants, set to $\alpha_1 = 0.990$, $\alpha_2 = 0.988$, $\alpha_3 = 0.986$.

To reach a good compromise in terms of network performance, several weights for the ranking components in are evaluated. The parameter selection is based on the delivery ratio, network overhead and network delay. Initially, experiments using a single metric were considered (e.g., $W_{GatewayDistance} = 1$ and $W_{GatewayHeadingAngle} = W_{Velocity} = 0$), allowing individual influence evaluation in the forwarding decision for each metric. These experiments allowed to understand that the velocity metric is the one with the highest preponderance for this specific connectivity model. Afterwards, distinct metric weights were explored. Table 5.4 summarizes the obtained results. Through the analysis of the network metrics (delivery ratio, network overhead and network delay), it is selected a distribution of $W_{Velocity} = 0.6$, $W_{GatewayDistance} = 0.2$ and $W_{GatewayHeadingAngle} = 0.2$ (last line in the table), which provides a good overall network performance: it has the highest delivery ratio and the lowest network delay, with a low network overhead.

5.3.3.2 ASAR parameters evaluation

In ASAR it is considered distinct sliding windows (T_{window}), and thresholds distinguishing between *low* and *high* in the number of both mobile nodes (τ_{MN}) and gateways (τ_{GW}). The influence of the several parameters in the network's performance is presented in Table 5.5: the change in the parameters is more pronounced in the network overhead metric, where higher τ_{MN} and τ_{GW} translate into higher overheads. Coping with this, higher T_{window} values

Parameters	Delivery Ratio (%)	Network Overhead (%)	Network Delay (sec)
$W_{Velocity} = 1; W_{GatewayHeadingAngle} = 0; W_{GatewayDistance} = 0$	69.41	403.95	220.69
$W_{Velocity} = 0; W_{GatewayHeadingAngle} = 1; W_{GatewayDistance} = 0$	66.83	366.16	179.11
$W_{Velocity} = 0; W_{GatewayHeadingAngle} = 0; W_{GatewayDistance} = 1$	66.82	379.34	171.74
$W_{Velocity} = 0.34; W_{GatewayHeadingAngle} = 0.33; W_{GatewayDistance} = 0.33$	69.25	357.55	214.01
$W_{Velocity} = 0.2; W_{GatewayHeadingAngle} = 0.4; W_{GatewayDistance} = 0.4$	69.44	354.49	213.58
$W_{Velocity} = 0.2; W_{GatewayHeadingAngle} = 0.6; W_{GatewayDistance} = 0.2$	70.03	351.06	216.09
$W_{Velocity} = 0.2; W_{GatewayHeadingAngle} = 0.2; W_{GatewayDistance} = 0.6$	69.22	362.59	216.20
$W_{Velocity} = 0.4; W_{GatewayHeadingAngle} = 0.2; W_{GatewayDistance} = 0.4$	69.42	363.14	211.41
$W_{Velocity} = 0.4; W_{GatewayHeadingAngle} = 0.4; W_{GatewayDistance} = 0.2$	69.29	356.79	216.14
$W_{Velocity} = 0.6; W_{GatewayHeadingAngle} = 0.2; W_{GatewayDistance} = 0.2$	70.22	360.86	208.01

Table 5.4: GLA distinct parameters experiments.

translate into lower delivery ratios. Therefore, considering the parameterization $T_{window} = 9$, $\tau_{MN} = 3$ and $\tau_{GW} = 1$ would result in the lowest overhead and delay, but would highly jeopardize the delivery ratio. With this in mind, a better compromise between all metrics is achieved by using $T_{window} = 6$, $\tau_{MN} = 3$ and $\tau_{GW} = 1$, which has one of the lowest network overheads and overall delays, although with a fair delivery ratio.

Parameters	Delivery Ratio (%)	Network Overhead (%)	Network Delay (sec)
$T_{window} = 3; \tau_{MN} = 3; \tau_{GW} = 1$	71.58	391.43	234.91
$T_{window} = 3; \tau_{MN} = 3; \tau_{GW} = 2$	71.68	413.12	236.83
$T_{window} = 3; \tau_{MN} = 6; \tau_{GW} = 1$	72.67	539.76	249.71
$T_{window} = 3; \tau_{MN} = 6; \tau_{GW} = 2$	71.77	561.13	241.16
$T_{window} = 3; \tau_{MN} = 9; \tau_{GW} = 1$	72.09	593.24	243.57
$T_{window} = 3; \tau_{MN} = 9; \tau_{GW} = 2$	71.59	618.05	243.35
$T_{window} = 6; \tau_{MN} = 3; \tau_{GW} = 1$	68.70	322.17	223.01
$T_{window} = 6; \tau_{MN} = 3; \tau_{GW} = 2$	70.87	349.86	234.99
$T_{window} = 6; \tau_{MN} = 6; \tau_{GW} = 1$	71.02	478.51	238.21
$T_{window} = 6; \tau_{MN} = 6; \tau_{GW} = 2$	71.18	495.20	248.47
$T_{window} = 6; \tau_{MN} = 9; \tau_{GW} = 1$	70.94	542.97	234.66
$T_{window} = 6; \tau_{MN} = 9; \tau_{GW} = 2$	71.97	570.06	246.88
$T_{window} = 9; \tau_{MN} = 3; \tau_{GW} = 1$	67.00	273.15	205.13
$T_{window} = 9; \tau_{MN} = 3; \tau_{GW} = 2$	68.59	318.03	214.27
$T_{window} = 9; \tau_{MN} = 6; \tau_{GW} = 1$	67.42	406.51	221.73
$T_{window} = 9; \tau_{MN} = 6; \tau_{GW} = 2$	70.24	440.11	233.38
$T_{window} = 9; \tau_{MN} = 9; \tau_{GW} = 1$	69.06	480.91	237.79
$T_{window} = 9; \tau_{MN} = 9; \tau_{GW} = 2$	69.96	508.17	237.14

Table 5.5: ASAR distinct parameters experiments.

5.3.3.3 HYBRID parameters evaluation

Since this strategy combines both GLA and ASAR strategies, the idea is to find the best combination between a location and a social based algorithm. The base parameters for the GLA and ASAR strategies are the ones that achieved a better output in the individual analysis. For the GLA strategy, the following parameters were selected $W_{Velocity} =$

0.6, $W_{GatewayDistance} = 0.2$ and $W_{GatewayHeadingAngle} = 0.2$, while for the ASAR strategy, $T_{window} = 6$, $\tau_{MN} = 3$ and $\tau_{GW} = 1$ were the selected weights.

Because the HYBRID ranking computation is obtained from two strategies already studied, the variations for the several experiments are quite smaller when compared to the two first forwarding strategies. However, from the results presented in Table 5.6, it is possible to understand that $W_{Mobility} = 0.2$ and $W_{Social} = 0.8$ provides the lowest delay, a similar delivery ratio, when compared to other parameterizations, and one of the lowest network overheads. Thus, these will be the HYBRID weights to be used in the remaining evaluations.

Parameters	**Delivery Ratio (%)**	**Network Overhead (%)**	**Network Delay (sec)**
$W_{Mobility} = 0.2; W_{Social} = 0.8$	69.02	321.87	200.02
$W_{Mobility} = 0.3; W_{Social} = 0.7$	69.18	323.76	205.59
$W_{Mobility} = 0.4; W_{Social} = 0.6$	69.37	324.97	204.70
$W_{Mobility} = 0.5; W_{Social} = 0.5$	68.98	327.06	205.03
$W_{Mobility} = 0.6; W_{Social} = 0.4$	69.05	319.33	207.06
$W_{Mobility} = 0.7; W_{Social} = 0.3$	69.10	316.64	209.99
$W_{Mobility} = 0.8; W_{Social} = 0.2$	69.29	311.83	210.77

Table 5.6: HYBRID distinct parameters experiments.

5.3.4 Forwarding strategies comparison

Considering scenario 1, the several strategies are evaluated, through the analysis of the packets received, transmitted and replicated in each node, and general network metrics. This scenario was selected for extended evaluation considering several forwarding strategies, since their parameters were previously optimized for this case.

Packets collected per mobile node

Considering the distinct forwarding strategies, the number of packets collected from DCUs per mobile node is presented in Figure 5.7. Even though 161 mobile nodes send traffic in the emulation, only 14 are represented in the figure.

Since not all mobile nodes contact with DCUs, these do not collect packets, which is the case of mobile nodes 319 and 321. However, in multi-hop forwarding strategies, these nodes can also contribute in later stages of the data gathering process, acting as a relay for data packets. In contrast, since some mobile nodes have much higher connection time with DCUs, these have much more collecting opportunities.

Since the connectivity model is the same for all forwarding strategies, the number of packets collected per forwarding strategy is expected to be the same. However, as explained before, when a mobile node collects a data packet from a DCU, it erases it from the list of packets available. Therefore, if for example, two mobile nodes are in connectivity range with a DCU, the processor scheduler in one emulation may decide that one mobile node is the first to execute and in another emulation it can be the second one, which explains the variation in packets received for some mobile nodes in distinct forwarding strategies.

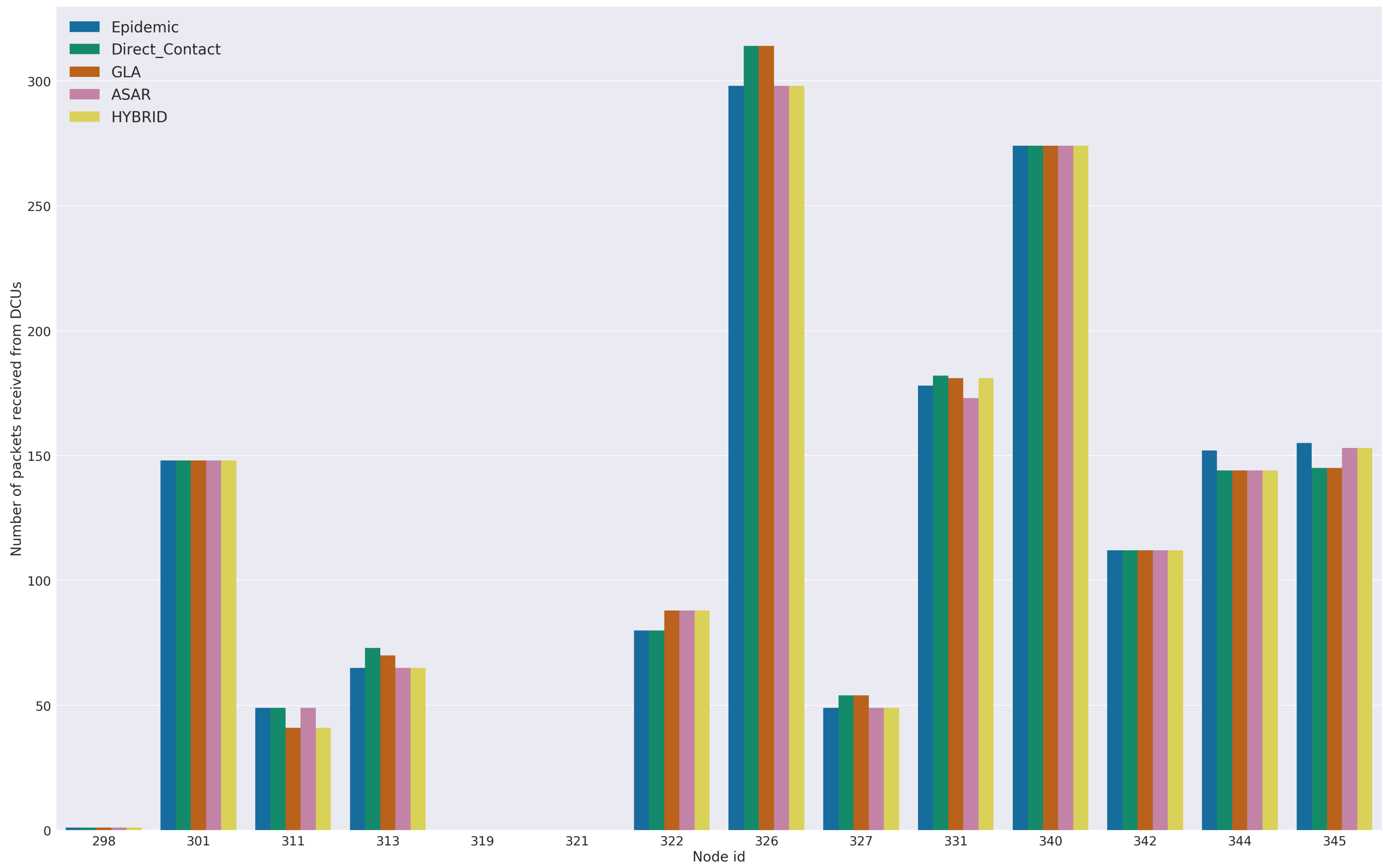


Figure 5.7: Scenario 1: Number of packets received from DCUs per mobile node (only 14 mobile nodes represented).

Packets transmitted from specific mobile nodes to other mobile nodes

Figure 5.8 illustrates the number of packets that each mobile node replicates to other mobile nodes per forwarding strategy. Following the same idea, only 14 of the 161 mobile nodes are presented.

Three conditions influence the number of packets delivered to other mobile nodes, independently of the multi-hop forwarding strategy: if the mobile nodes have many mobile neighbors; if the mobile nodes have packets in storage at the moment of connection with other mobile nodes; and the connection time with gateways, since packets delivered to gateways are erased from storage if an ACK packet is received.

Taking into account these considerations, it is observed that the number of packets transmitted varies significantly. As observed in Figure 5.8, the number of packets replicated varies also according to the forwarding strategy. Thus, for the forwarding strategies, the analysis is partitioned:

- **Epidemic:** this forwarding strategy has the highest number of packets replicated per mobile node. Since no criteria is applied in the neighborhood selection, if packets exist in storage, these are replicated to all the mobile node neighbors. This excessive amount of replication contributes directly to a high network overhead.

- **Direct Contact:** since packets are only delivered to gateways, there is no replication of packets to other mobile nodes. As a single-hop forwarding strategy, the lowest network overhead is expected for all the forwarding strategies evaluated.

- **Proposed forwarding strategies:** according to a ranking system, only good neighbors are selected for packet replication. The contrast of packet replication between Epidemic routing and the proposed forwarding strategies represents the influence in using neighboring classification metrics in the replication process. Furthermore, it is possible to observe that the number of packets delivered between Epidemic and the proposed strategies is not linear for all the mobile nodes. This is expected, since the neighborhood metrics may dictate that a mobile node with high number of neighbors should not replicate packets since these are *poor* neighbors. However, a mobile node with lower number of neighbors, but with better neighborhood classification, will replicate more often. So, this evaluation allows to understand that, according to the neighborhood metrics, some mobile nodes will have a higher importance in the replication process than others.

Packets transmitted from specific mobile nodes to gateways

Figure 5.9 presents the number of packets that each mobile node delivers to gateways per forwarding strategy. As before, only 14 of the 161 mobile nodes are presented.

Some general considerations can be made for the several mobile nodes, not considering the forwarding strategies. For instance, mobile nodes 326 and 340 collect and high amount of packets from DCUs and mainly deliver them to gateways, thus not replicating packets as much as other mobile nodes. On the other hand, mobile nodes 319 and 321 did not contact with DCUs. However, through packet replication from their neighbors, they were able to deliver packets to gateways. Furthermore, cases such as mobile nodes 301, 311 and 313 occur, where the packets collected from DCUs together with packet replication from neighbors contribute to higher delivery of packets to gateways.

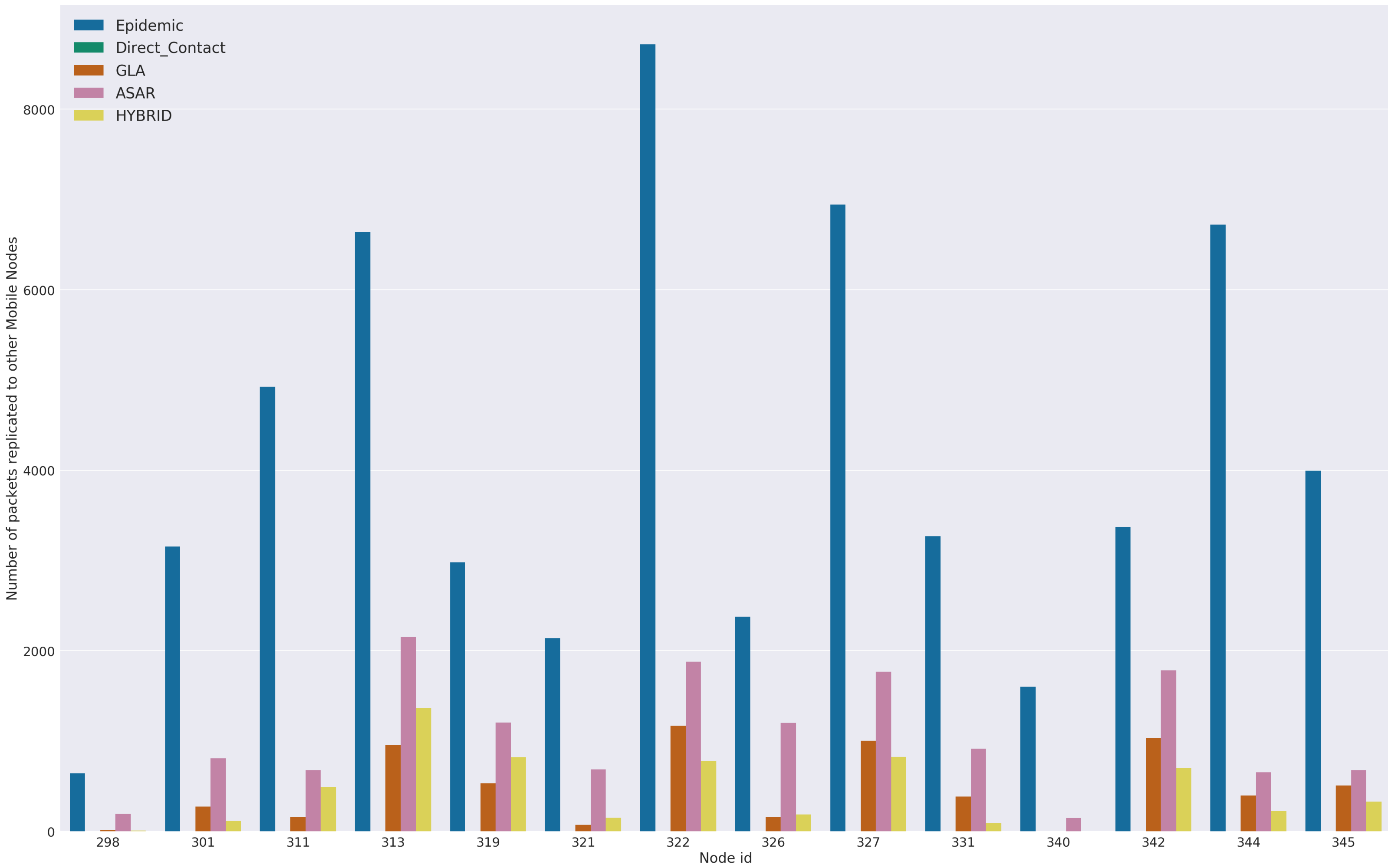


Figure 5.8: Scenario 1: Number of packets transmitted by mobile nodes to other mobile nodes (only 14 mobile nodes represented).

As observed in Figure 5.9, the number of packets delivered to gateways varies also according to the forwarding strategy. Thus, for the forwarding strategies, the analysis is partitioned:

- **Epidemic:** this strategy delivers a high amount of packets to gateways, which is expected due to the flooding of packets in the network.
- **Direct Contact:** in the best case scenario, the number of packets delivered to gateways is equal to the number of packets collected from DCUs per mobile node. However, if for some reason packets are dropped, the number of delivered packets will decrease. Being a single-hop strategy, mobile nodes that do not collect packets from DCUs will not deliver packets to gateways.
- **Proposed forwarding strategies:** despite the number of packets replicated between mobile nodes, the number of packets delivered to gateways for the proposed forwarding strategies are similar as the ones of Epidemic strategy. This is a positive sign, since it shows that with much lower replication, a similar number of packets delivered to gateways is achieved.

Packets dropped per mobile node

Figure 5.10 depicts the number of dropped packets per mobile node. As expressed before, a mechanism to prevent storage congestion and network lifetime for each packet is considered.

There is a high dependency between the number of packets in storage and the number of packets dropped per mobile node. Since the packet selection mechanism does not consider the network state (first packet selection mechanism used for these experiments), some packets may not have the chance to be selected to be forwarded, thus being dropped due to lifetime expiry. Also, it is possible to observe that not all mobile nodes drop the same number of packets, which is related to their connections with other mobile nodes or gateways.

As evidenced in Figure 5.10, the number of dropped packets per mobile node varies also according to the forwarding strategy. Thus, for the forwarding strategies, the analysis is partitioned:

- **Epidemic:** due to the unmediated replication of packets, this strategy will have the maximum number of dropped packets per mobile node.
- **Direct Contact:** only packets that exceed their respective network lifetime will be dropped in this forwarding strategy. Thus, the number of packets dropped is at the minimum when comparing with multi-hop forwarding strategies.
- **Proposed forwarding strategies:** the number of packets dropped for the proposed forwarding strategies follows the same trend as the Epidemic routing. This is expected, since no packet selection technique is also applied. However, the number of dropped packets is slightly lower when comparing to the Epidemic routing, which is explained by the lower number of packet replication for the proposed forwarding strategies.

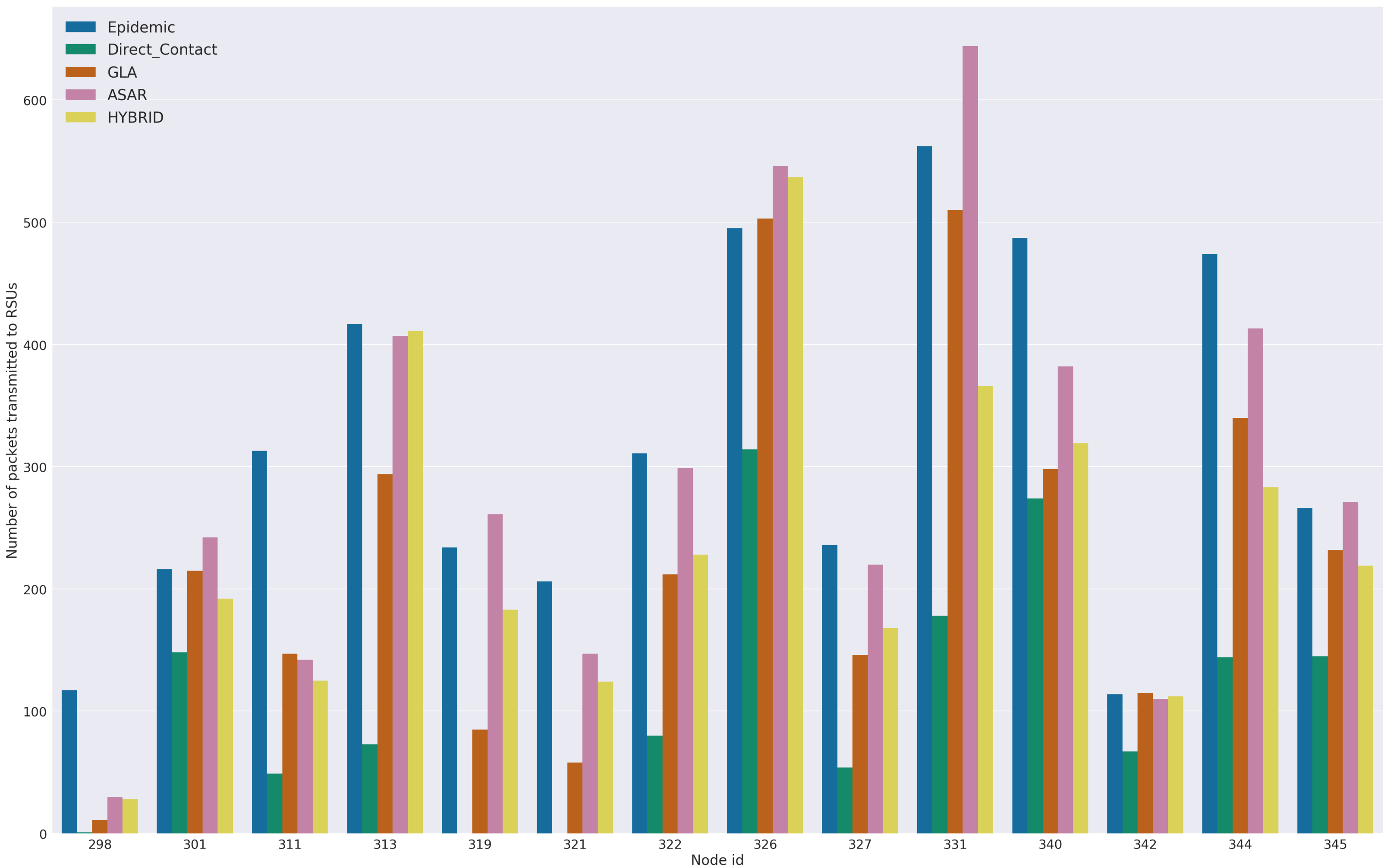


Figure 5.9: Scenario 1: Number of packets transmitted by mobile nodes to RSUs (only 14 mobile nodes represented).

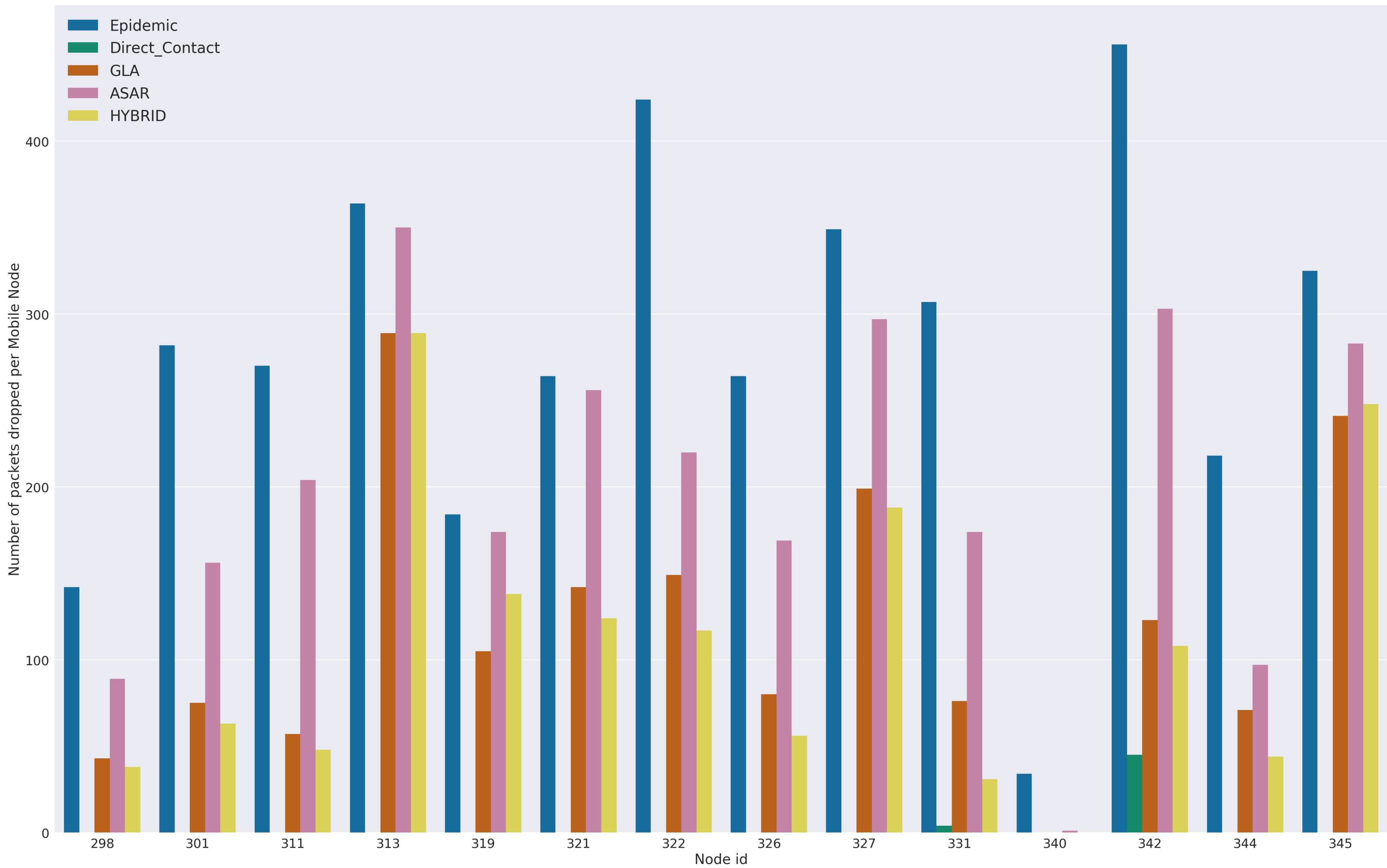


Figure 5.10: Scenario 1: Number of packets dropped per mobile node (only 14 mobile nodes represented).

Network mean number of packet hops

Figure 5.11 depicts the mean number of packet hops per forwarding strategy, which is the mean number of nodes that a packet traverses until reaching a gateway.

For the Direct Contact strategy, the average number of hops is static and 1, since packets are directly delivered to a gateway. However, for the multi-hop forwarding strategies, the mean number of hops increases. As expected, Epidemic strategy has the highest mean number of hops per packet, because the flooding mechanism will blindly forward each packet to the network. ASAR forwarding strategy has a higher mean number of packet hops when compared to GLA and HYBRID, since the ranking metrics decide the forwarding neighbors in a distinct way.

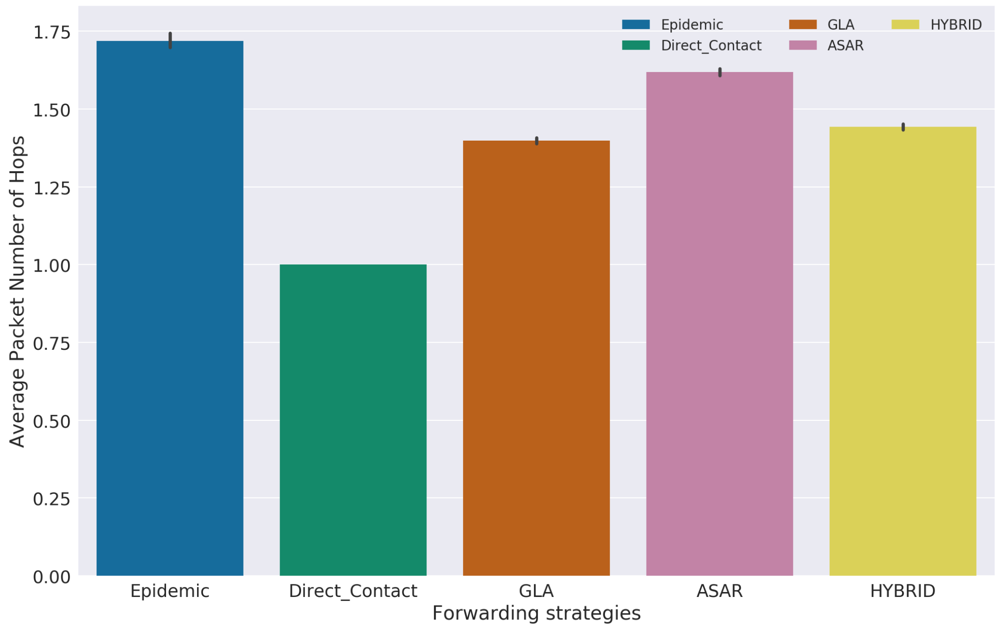


Figure 5.11: Scenario 1: network average number of hops per forwarding strategy.

Overall network delay

The results for the network delays in the distinct forwarding strategies are shown in Figure 5.12, representing the mean delay of the packets that were effectively delivered to a gateway. The delay is measured from the moment when the packet is generated until it reaches the first gateway. If the same packet reaches another gateway, the packet considered is the one with the lowest delay.

When comparing with the results obtained for the mean number of packet hops in Figure 5.11, there is a correlation between them. Effectively, cases where packets are delivered through multiple hops, since the first hop did not have connection with gateways, will introduce higher delay in the network. This happens because, for each hop, the node travelling time is added until a new forwarding opportunity arises.

However, in first approach, it is not expected that the network delay for the Epidemic routing to be higher than the Direct Contact strategy, since the packet either arrives at the same time or first, due to replication to a node which contacts faster with a gateway. However, the point stated above explains why this behaviour is verified, because packets that were not delivered, are now able to reach gateways through replication. To fairly compare the mean delay, only packets that are delivered in the Direct Contact will be compared with other forwarding strategies, which is presented in Figure 5.13.

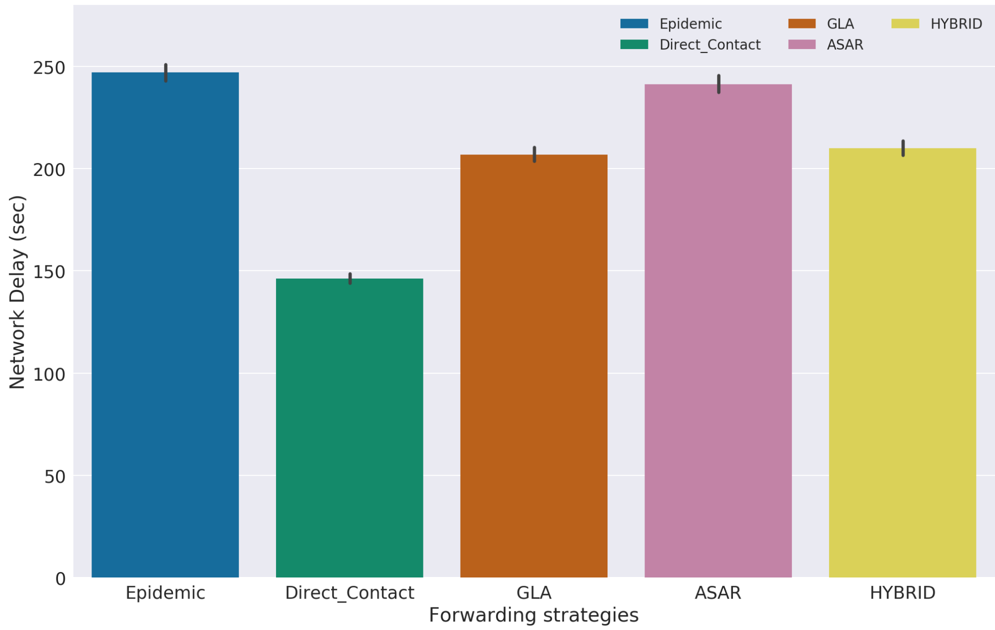


Figure 5.12: Scenario 1: overall network delay per forwarding strategy.

Figure 5.13 presents the delay of the same received packets in all strategies. For the Direct Contact strategy, the mean network delay is exactly the same as before. However, the results change drastically for the other forwarding strategies, where the mean network delay decreases significantly. It is observable that the mean delay decreases through multi-hop mechanisms, which is expected since some of the nodes that received the replicated packets delivered them faster to a gateway, in comparison with the first hop.

Overall network overhead

The results of the network overhead for the distinct forwarding strategies are shown in Figure 5.14, representing the amount of redundant information that each forwarding ranking combination introduces in the network to deliver the collected information.

The results for the network overhead are correlated with the number of packets replicated between mobile nodes, illustrated in Figure 5.8. This is evidenced in the Epidemic routing, since the huge amount of packet replication translates into the large amount of network overhead. For the Direct Contact strategy, the network overhead is very close to 100%, since there is no replication of packets. The reason why it is not exactly 100% is due to the ACK packets, which allow mobile nodes to free the delivered packets from storage. In the proposed forwarding strategies, the network overhead is below 400%, much lower than in the Epidemic, which means that making a neighborhood selection in the forwarding decision translates into a lower resource consumption. Ideally, the proposed forwarding mechanisms aim to reduce the resource expenditure to a minimum, but a multi-hop approach will always have an associated

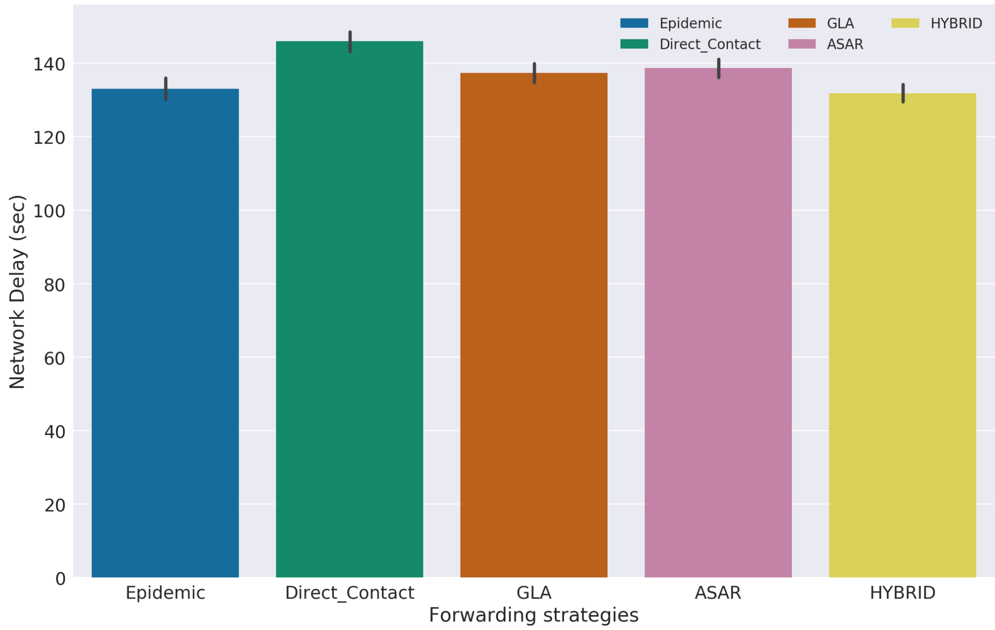


Figure 5.13: Scenario 1: overall network delay for the same packets in distinct forwarding strategies.

---

cost.

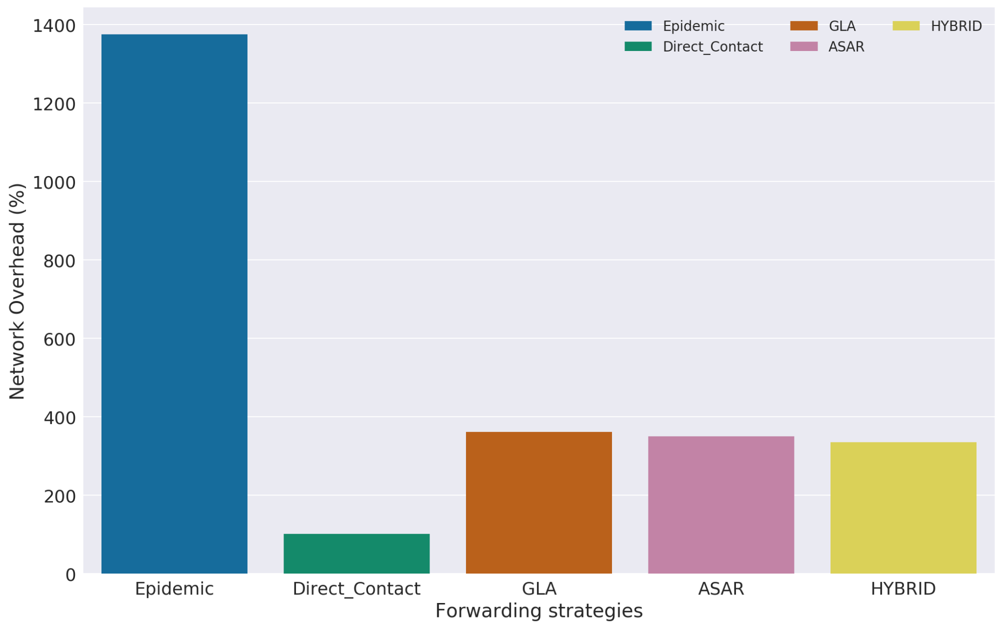


Figure 5.14: Scenario 1: overall network overhead per forwarding strategy.

Overall network delivery ratio

Presented in Figure 5.15, this metric represents the effective percentage of data packets delivered from DCUs to gateways during the experiments (duplicated packets received by the gateway do not contribute to this percentage).

In the network overhead evaluation part, it was referred the associated cost of using multi-hop approaches in the forwarding decision. In this section, the benefit of such approach is evidenced, where the replication of packets allow an increase in the network delivery ratio. Comparing the delivery ratio between Epidemic and Direct Contact, they show both extremes, with Epidemic representing the potential of delivery ratio increase through multi-hop strategies. The results achieved by the Epidemic strategy show that, for the considered time period, it is impossible to deliver 100% of the data packets using only the vehicular network. ASAR is the best overall in terms of delivery ratio, approaching the one of Epidemic. However, the differences between the 3 proposed strategies are small. As observed in Figure 5.15, the proposed strategies have a similar delivery ratio when compared to Epidemic. However, the amount of network resources expended to obtain such delivery ratio is significantly lower. For this scenario, it is possible to observe that there are time frames where some strategies are more viable than others in terms of delivery ratio. This is expected, since the metrics used to classify neighbors aim to explore distinct behaviours.

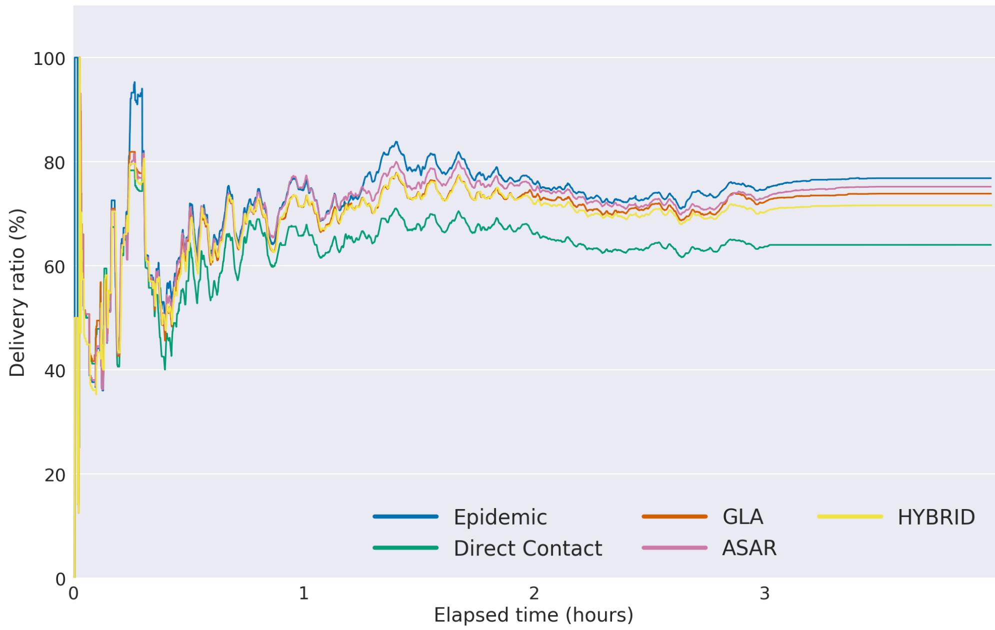


Figure 5.15: Scenario 1: overall delivery ratio per forwarding strategy.

5.3.5 Distinct scenarios evaluation

This evaluation will be conducted grouping adjacent characteristics for several scenarios. This allows a better understanding of the observed trends. For the several evaluation metrics, tables summarizing the obtained results are provided, where, for each scenario, the forwarding strategies are compared among themselves. The groups considered are:

- **Group 1:** Scenarios 1, 2 and 3 belong to this group, where the number of DCUs varies from 8, 12 and 16, respectively.
- **Group 2:** Scenarios 4, 5 and 6 belong to this group, where the number of DCUs varies from 8, 12 and 16, respectively, and all mobile nodes generate packets periodically.
- **Group 3:** Scenarios 1 and 7 are considered in this group, where the number of gateways varies from 8 to 12, respectively.

Network delivery ratio

Figure 5.15, Figure 5.16 and Figure 5.17 represent the network delivery ratio evolution considering the experiment time for scenarios 1, 4 and 7, respectively. Only these are presented in this form, since the variation observed during the time is larger in these scenarios.

The delivery ratio of scenario 1 was already evaluated before, thus serving as the baseline scenario for comparison purposes. The differences between scenarios 1 and 4 is that, for scenario 4, the mobile nodes generate data packets periodically. Since the number of packets in the network increases, the delivery ratio is lower when compared to scenario 1. Moreover, it is visible that the delivery ratio for scenarios 1 and 4 progresses in a distinct manner, which is due to the fact that mobile nodes in scenario 1 do not have packets in storage (no contacts with DCUs), while packets in scenario 4 are stored due to the periodic generation.

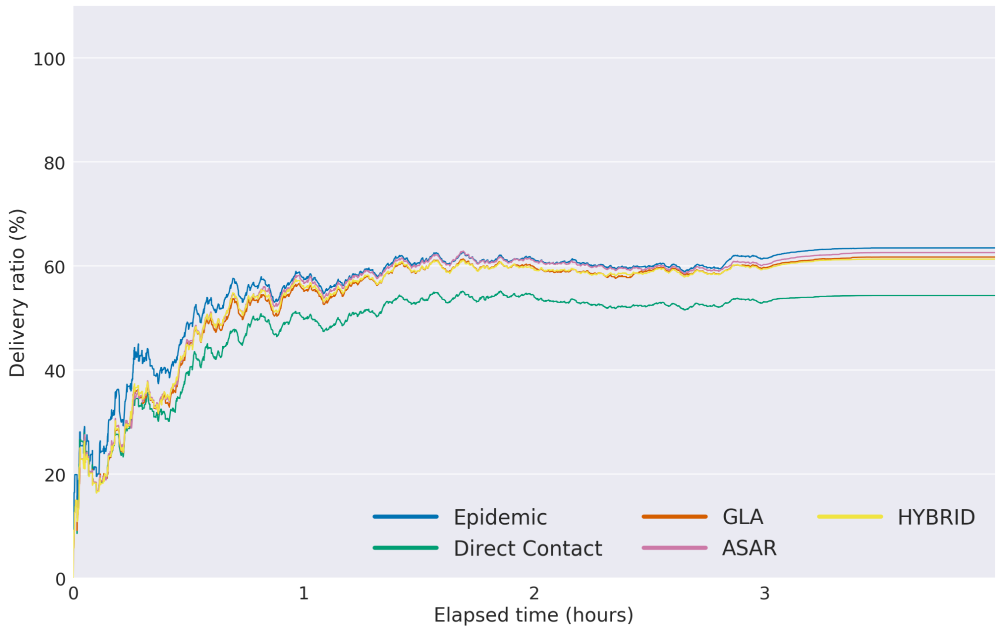


Figure 5.16: Scenario 4: overall delivery ratio per forwarding strategy.

Scenarios 1 and 7 only differ on the number of gateways considered, the first with 8 and the second with 12. As expected, since scenario 7 has more gateways, there is an increase in the delivery ratio when comparing with scenario 1. However, since the number of packets generated is equal to scenario 1, the time evolution of the delivery ratio is very similar for these scenarios.

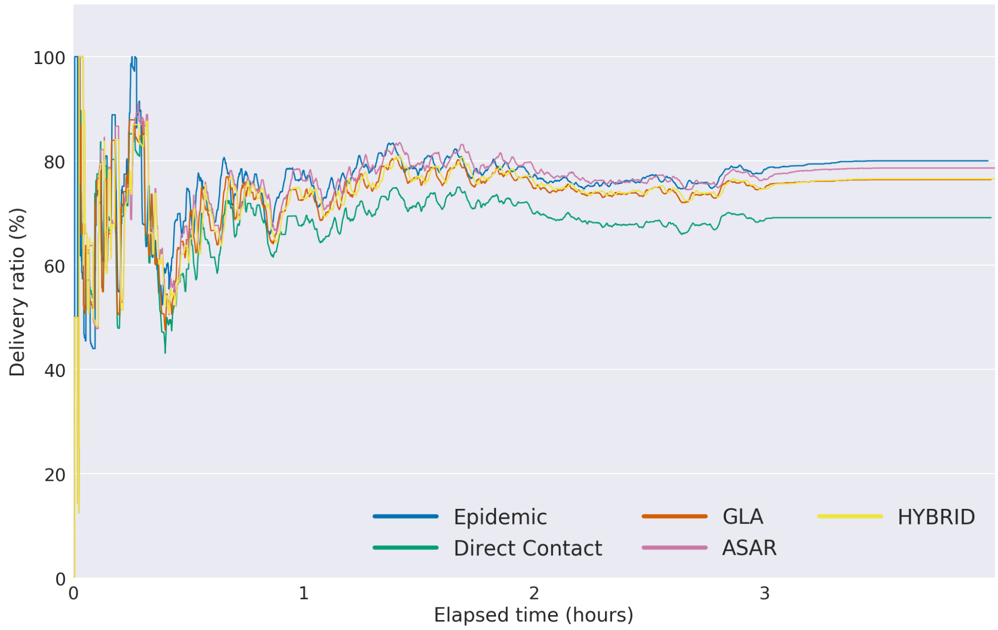


Figure 5.17: Scenario 7: overall delivery ratio per forwarding strategy.

Allied with the time evolution of the delivery ratio, there are also patterns observed when looking at the mean delivery ratio for all the scenarios, presented in Table 5.7.

Considering group 1, the average delivery ratio increases with the increase of the number of DCUs. Even though more packets are generated, the location of the DCUs is closer to the location of the gateways (as illustrated in Figures 5.3, 5.4 and 5.5), thus contributing to this increase.

In group 2, an increment in the number of DCUs located closer to the location of the gateways also contributes to an increment in the delivery ratio. However, when comparing with group 1, the delivery ratio for scenarios 4, 5 and 6 is lower, because the number of packets per mobile node is higher due to the periodic mobile packet generation.

In group 3, an increase in the number of RSUs translates into an increase in the delivery ratio. Since the number of gateways available to deliver the packets increases, the delivery ratio also increases.

Group(s)	Scenario	Epidemic	Direct Contact	GLA	ASAR	HYBRID
1 and 3	Scenario 1	73.20	63.35	70.22	71.49	69.20
1	Scenario 2	76.67	70.59	75.12	74.66	75.44
1	Scenario 3	76.96	72.29	75.05	75.59	76.05
2	Scenario 4	57.01	49.15	54.75	55.65	54.80
2	Scenario 5	60.59	56.07	60.57	60.13	60.65
2	Scenario 6	62.30	59.72	63.52	62.92	64.14
3	Scenario 7	76.43	68.40	73.60	75.93	73.96

Table 5.7: Average delivery ratio in percentage per scenario and forwarding strategy.

Network mean number of packet hops

Table 5.8 summarizes the mean number of packet hops per forwarding strategy and per scenario. It is important to mention that only multi-hop forwarding strategies have variations on the mean number of hops per packet, which means that for the Direct Contact strategy the results are static.

A decrease in the number of packet hops is observed as the number of DCUs increases, which is again related to the location of the added DCUs (scenarios in groups 1 and 2). Since the added DCUs are closer to the gateways, the mean number of packet hops decreases.

Comparing groups 1 and 2, the number of hops decreases with a lower rate, which denotes that packets generated periodically in mobile nodes for group 2 have a lower mean number of hops than packets generated at DCUs. This is observed for this connectivity model, but it is highly dependent on the considered network.

In group 3, the mean number of hops per packet decreases with the increase in the number of RSUs, which is also related with the location of the gateways both in scenario 1 and 7 (Figures 5.3 and 5.6).

Group(s)	Scenario	Epidemic	Direct Contact	GLA	ASAR	HYBRID
1 and 3	Scenario 1	1.72	1	1.40	1.62	1.44
1	Scenario 2	1.54	1	1.35	1.36	1.36
1	Scenario 3	1.45	1	1.30	1.31	1.31
2	Scenario 4	1.69	1	1.46	1.63	1.48
2	Scenario 5	1.56	1	1.38	1.41	1.41
2	Scenario 6	1.45	1	1.34	1.34	1.35
3	Scenario 7	1.65	1	1.36	1.58	1.42

Table 5.8: Average number of traversed hops per scenario and forwarding strategy.

Overall network delay

Table 5.9 shows the mean network delay per forwarding strategy and per scenario. This metric depicts the amount of time elapsed since a packet is created until it reaches a gateway.

For group 1, it is observed that scenario 1 has the highest delay, followed by scenario 3 and finally, scenario 2. Since scenario's 1 DCUs and gateways have the most distance between them, it is expected that this scenario is effectively the one with the highest delay for this group. The added DCUs in scenario 2, in comparison with scenario 1 are much closer in distance with the gateways, which greatly contributes for the decrease in the network delay. Scenario 3 added DCUs in comparison with scenario 2 are a compromise in distance between the two scenarios referred before, thus translating into a network delay comprised between the ones verified for scenarios 1 and 2. Since the locations for the DCUs in group 2 are the same, the same pattern is observed.

Comparing group 1 and 2, there is an increase in the network delay caused by the packets generated periodically in mobile nodes. Since all nodes now have packets, those that do not contact frequently with gateways, or do not have too many forwarding opportunities, will increase the overall network delay.

Finally, for group 3, a decrease in the mean delay with the increase of the number of gateways is observed. Having more points to deliver information, and the same packets to deliver, it is expected that the delivered packets arrive faster.

Group(s)	Scenario	Epidemic	Direct Contact	GLA	ASAR	HYBRID
1 and 3	Scenario 1	247.45	146.27	208.01	242.71	210.56
1	Scenario 2	182.77	119.61	166.39	157.08	155.67
1	Scenario 3	210.40	147.35	187.93	181.73	188.86
2	Scenario 4	351.63	254.76	333.12	337.92	326.28
2	Scenario 5	267.23	198.02	251.89	249.97	262.13
2	Scenario 6	273.14	203.78	260.50	251.77	260.11
3	Scenario 7	216.28	139.15	196.22	215.31	200.93

Table 5.9: Average network delay in seconds per scenario and forwarding strategy.

Overall network overhead

Table 5.10 depicts the mean network overhead per forwarding strategy and per scenario. This metric translates the percentage of overall data used to deliver the effective amount of data to a gateway.

In group 1, the overall network overhead decreases with the increment of DCUs, which is expected since additional DCUs are closer in distance to the gateways, and then additional packets do not need to be replicated so many times in order to reach the gateway.

When comparing groups 1 and 2, the network overhead is higher for group 2. Since the difference relies on packets generated by mobile nodes in group 2, some of these packets require a larger number in replications to reach a gateway, causing a higher network overhead.

In group 3 the network overhead is lower than in scenario 1. Since the number of gateways is higher in scenario 7, a lower number of packet replications is needed for a packet to reach a gateway, resulting in a lower network overhead.

Group(s)	Scenario	Epidemic	Direct Contact	GLA	ASAR	HYBRID
1 and 3	Scenario 1	1375.06	101.20	361.11	350.14	335.30
1	Scenario 2	982.21	101.08	291.18	271.11	261.92
1	Scenario 3	726.52	100.92	246.40	232.80	224.20
2	Scenario 4	1945.42	101.22	679.63	801.82	551.74
2	Scenario 5	1276.44	101.05	491.67	412.35	402.93
2	Scenario 6	1057.00	100.87	414.44	343.02	339.33
3	Scenario 7	1311.87	101.15	338.19	325.57	307.32

Table 5.10: Average network overhead in percentage per scenario and forwarding strategy.

5.3.6 Packet selection mechanisms evaluation

From the considered scenarios, scenario 6 has the most generated packets. For that reason, it was elected as base scenario for the evaluation of the packet selection mechanisms.

As described earlier, two packet types are considered in this work: environmental data packets and gas data packets. This evaluation addresses distinct priorities and network lifetimes for these data types. Both types are generated with a 50% probability, which means that the network contains close to half of packets of each type.

Table 5.11 summarizes the distinct experiments used to evaluate the proposed packet selection techniques. Since each experiment has a duration of 4 hours, the considered packets' lifetimes comprises time intervals which allows a good delivery opportunity per packet and avoids nodes' storage congestion.

Packet selection mechanism	**Environmental data network lifetime**	**Gas data network lifetime**	**Observations (algorithm constants)**
First packet selection	15 min	15 min	None
Distributed packet selection	15 min	15 min	Environmental probability = 30% Gas probability = 70%
Equalized packet selection	15 min	15 min	Environmental probability = 30% Gas probability = 70% $\alpha = \beta = \gamma = 1/3$

Table 5.11: Packet selection experiments summary.

Packet type delivery percentage to gateways

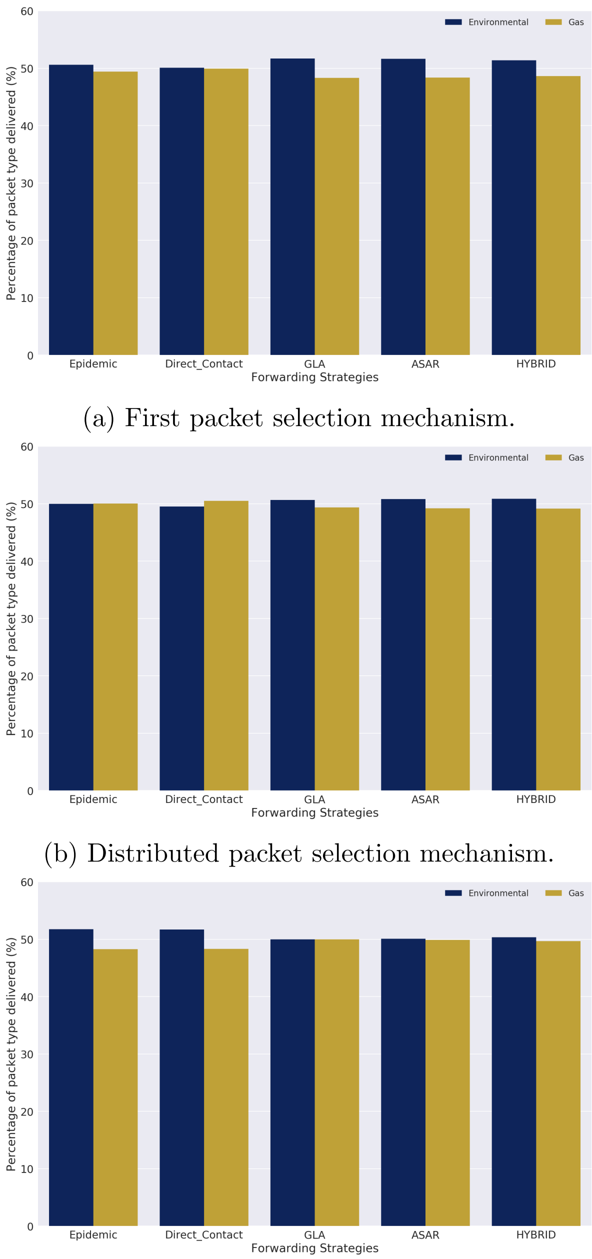
Figure 5.18 presents the percentage of packet types delivered from mobile nodes to gateways for all the distinct experiments and per forwarding strategy.

For all tests, a rate of around 50% for each data type is observed. Since some packet selection mechanisms were specially designed for prioritizing data types, these results seem contradictory. Effectively, the reason why this happens is related to the way this network operates, where data packets are transmitted to gateways in bursts. In other words, when a data packet is received by a gateway, an ACK packet is sent by the same and received by the mobile nodes, and the respective packet is deleted from the storage. If the connection time with gateways is long enough, all packets in storage are delivered, meaning that the packet selection mechanisms only influence the order of delivery. Taking this into consideration, this evaluation relies in other metrics to validate the packet selection mechanisms: dropped packets.

Dropped packets delivered to gateways through replication

Figure 5.19 depicts the percentage of packets that mobile nodes dropped but which were still delivered through multi-hop replication in all the experiments and per forwarding strategy. Since the proposed packet selection mechanisms aim to address inherent problems associated with the high volumes of data collected in IoT, it is important to consider this percentage since it contributes to the overall delivery ratio of distinct packets to gateways.

For the Direct Contact strategy, the percentage is null since no packets are replicated. On the other hand, the Epidemic routing has the highest delivery ratio of packets dropped through replication, which is expected since mobile nodes flood the network with packet replications. For the proposed strategies, the percentage is slightly lower, which is related with the lower network overhead in comparison with the Epidemic strategy.



(a) First packet selection mechanism.

(b) Distributed packet selection mechanism.

(c) Equalized packet selection mechanism.

Figure 5.18: Packet type delivery percentage per packet selection mechanism.

When comparing the distributed packet selection mechanism in Figure 5.19b with the first packet selection mechanism in Figure 5.19a, there is a slight decay on the delivery ratio of dropped packets through multi-hop replication, with the exception of the Epidemic routing. This happens because this packet selection algorithm gives priority to one type of data in deterioration of another, unbalancing the chances of some packets to reach a gateway. However, this also portrays advantages, which will be addressed later in this packet selection evaluation. For the Epidemic case in specific, the delivery ratio rises slightly.

Considering the equalized packet selection mechanism in Figure 5.19c, since it uses the network state in order to choose a packet, the delivery ratio of dropped packets through multi-hop replication is the highest when compared to the first packet selection. For cases where a good network performance is aimed from the delivery ratio point of view, this packet selection mechanism is the better choice.

Dropped packet types delivered to gateways through replication

Figure 5.20 illustrates the percentage of packet types dropped by mobile nodes but still were delivered through multi-hop replication in all experiments and per forwarding strategy. Since these packets are replicated to other mobile nodes and not delivered to gateways, they are not erased from storage since no ACK packet is received. Therefore, contrary to the percentages observed for the data packets delivered to gateways, the effects of the packets' selection mechanisms will be more evident on a data type analysis.

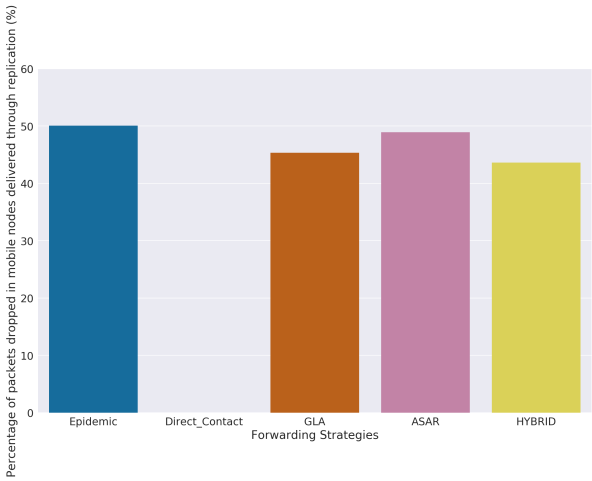
For the first packet selection mechanism in Figure 5.20a, the percentages of dropped packets per type are close to 50%. Since no data type filter is applied and each data type is generated with 50% probability, these percentages are as expected.

Figure 5.20b shows the results for the distributed packet selection mechanism. Since this packet selection algorithm is based primarily in data type selection, it is visible that the percentages for each data type are very close to the packet type probabilities used in these experiments. These values will never match for two reasons: the packet selection technique is probabilistic; and there are cases where the selected data type does not exist in storage but other types do, thus the one with lower lifetime will be the one to be selected.

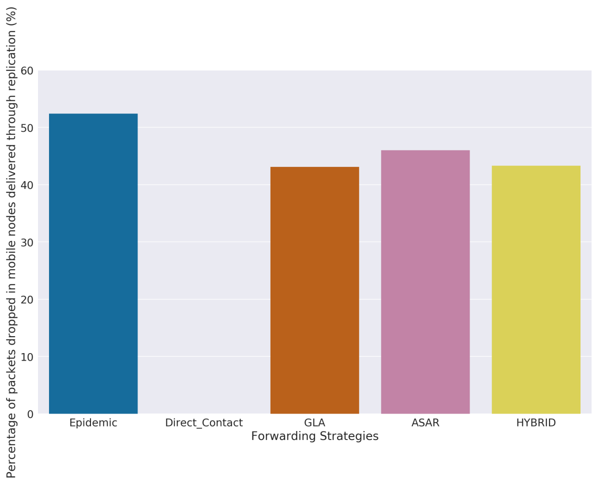
For the equalized packet selection algorithm, it was already observed that it contributes to a better network performance. However, when compared to the distributed packet selection technique, it has a cost in the packet type selection (comparison between Figure 5.20c and Figure 5.20b). Since a rank is calculated based on various parameters, the packet type selection does not portray the same efficiency when compared with the distributed algorithm. However, the equalized packet selection mechanism still offers a significant degree of packet type priority, which is a good sign when compared with the first packet selection algorithmn.

5.4 Network Practical Use Case

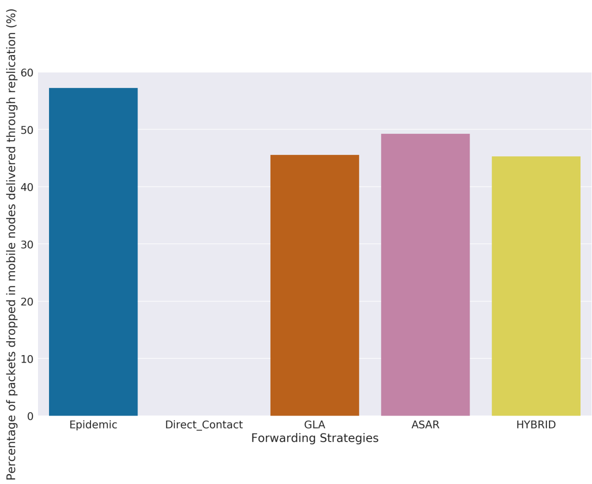
This section focuses on validating the proposed algorithms in a real scenario. Since the network conditions are in constant change in a real scenario, it is harder to compare the proposed algorithms amongst them (already addressed through emulation). However, it is important to validate the robustness and functionality of each approach, as well as the architecture as a whole.



(a) First packet selection mechanism.

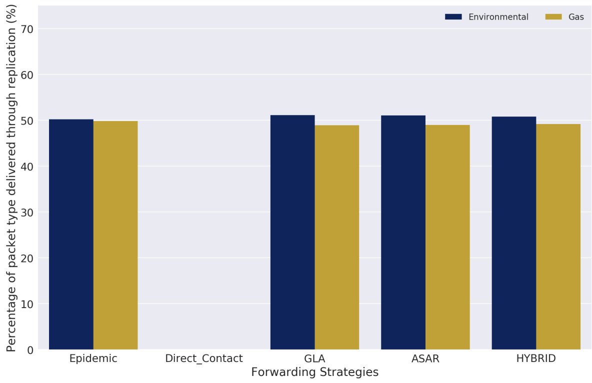


(b) Distributed packet selection mechanism.

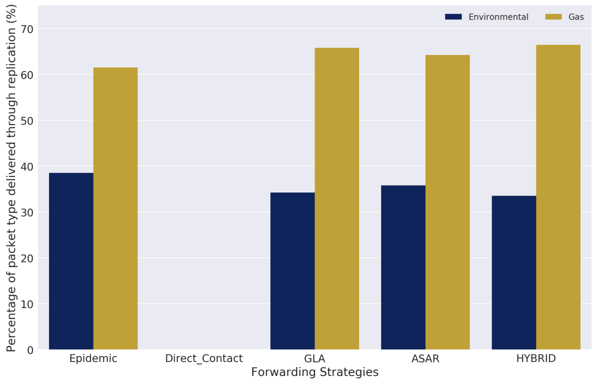


(c) Equalized packet selection mechanism.

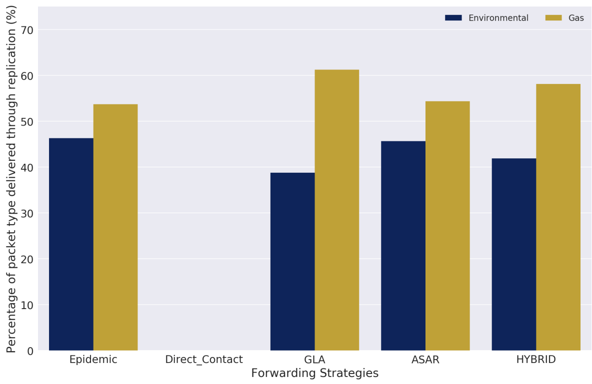
Figure 5.19: Dropped packets delivery percentage through multi-hop replication per packet selection mechanism.



(a) First packet selection mechanism.



(b) Distributed packet selection mechanism.



(c) Equalized packet selection mechanism.

Figure 5.20: Packet type percentages for dropped packets delivered through multi-hop replication per packet selection mechanism.

5.4.1 Scenarios definition and considerations

In order to evaluate the proposed forwarding strategies in a real scenario, a use case was formulated making use of the mobility of typical boats ("moliceiros") in the "Ria" of Aveiro. These boats are used as touristic attraction, making trips of around 45 minutes through the canals of the "Ria" of Aveiro. Therefore, the mobility is used to gather information in specific places (through the connection of boats with DCUs) and deliver it to another boat or a gateway. Figure 5.21 illustrates the location of the network nodes, where the mobile nodes' location shows the resting point between trips of each boat.

Figure 5.21: Network nodes locations.

Some definitions and considerations were taken into account for this specific scenario, namely:

- Each forwarding strategy was evaluated for $\simeq$ 8 hours, starting at 9am and ending at 5pm at all days.
- In order to not introduce entropy, the LoRa part of the network was disabled. With this consideration, the practical results will be coherent with the experiments performed before.
- DCU 204 is distanced 165 meters away from RSU 10, and DCU 239 is distanced 132 meters from RSU 10. Even considering line of sight, which is not the case, these nodes are not in connectivity range, thus justifying the need of the DTN architecture.
- Both DCUs generate data packets every 5 seconds, resulting in 5760 total generated packets per DCU.
- The equalized packet selection mechanism (Algorithm 3) was used to choose a packet in the forwarding decision.

- Both environmental and gas data types have a packet expiry time of 90 minutes, which was chosen considering the typical 45 minutes boat trips and interval of time between trips. If this time is exceeded, the packet is dropped to avoid storage congestion.

- Only DCUs are generating data packets for these experiments, even though it is possible for mobile nodes to do the same, with extended hardware.

- The GPS position was tracked for each mobile node every 5 seconds in order to better evaluate the network performance between mobile nodes and forwarding strategies.

- GLA parameters:
  - Connectivity range distance = 84 meters (WiFi practical experiment) and the maximum distance of interest $maxD_{MG} = 1000$ meters, both in (3.3);
  - Heading angle attenuation constants: $\alpha_1 = 0.990$; $\alpha_2 = 0.988$; $\alpha_3 = 0.986$ in (3.8);
  - Nodes' mean velocity: $V_{avg} = 6.94$ m/s (around 25 Km/h) in (3.9);
  - $W_{GatewayDistance} = W_{GatewayHeadingAngle} = W_{Velocity} = 1/3$.

- ASAR parameters:
  - $T_{window} = 10$ minutes;
  - $\tau_{MN} = 2$, since the rest location for boats between trips generally has more than 1 boat;
  - $\tau_{GW} = 1$, since there is only one gateway in this scenario.

- HYBRID parameters:
  - $W_{Mobility} = W_{Social} = 0.5$;
  - The aforementioned parameters for GLA and ASAR ranking computations were replicated for this strategy.

5.4.2 Obtained results

This scenario has a high level of randomness. Also, the network performance depends highly on the boats' number of travels (assuming the same course at each travel).

To address these inherent scenario properties, plots with location heatmaps per forwarding strategy and per mobile node were produced.

Figures 5.22, 5.23, 5.24, 5.25, 5.26 represent the mobile nodes location heatmaps for the practical tests regarding the Epidemic, Direct Contact, GLA, ASAR and HYBRID forwarding strategies, respectively. In these maps, the more vivid the color, the more trips boats performed.

Mobile node 102 had a logging problem, which is verified for all the forwarding strategies. This translates into a inconclusive representation of this node's location heatmap. However, since it was verified through observation that this node effectively traveled during the evaluation days, it contributed to the network performance.

(a) Mobile Node 101.

(b) Mobile Node 102.

(c) Mobile Node 103.

(d) Mobile Node 104.

Figure 5.22: Epidemic routing: Mobile nodes location heatmaps.

(a) Mobile Node 101.

(b) Mobile Node 102.

(c) Mobile Node 103.

(d) Mobile Node 104.

Figure 5.23: Direct Contact: Mobile nodes location heatmaps.

(a) Mobile Node 101.

(b) Mobile Node 102.

(c) Mobile Node 103.

(d) Mobile Node 104.

Figure 5.24: GLA: Mobile nodes location heatmaps.

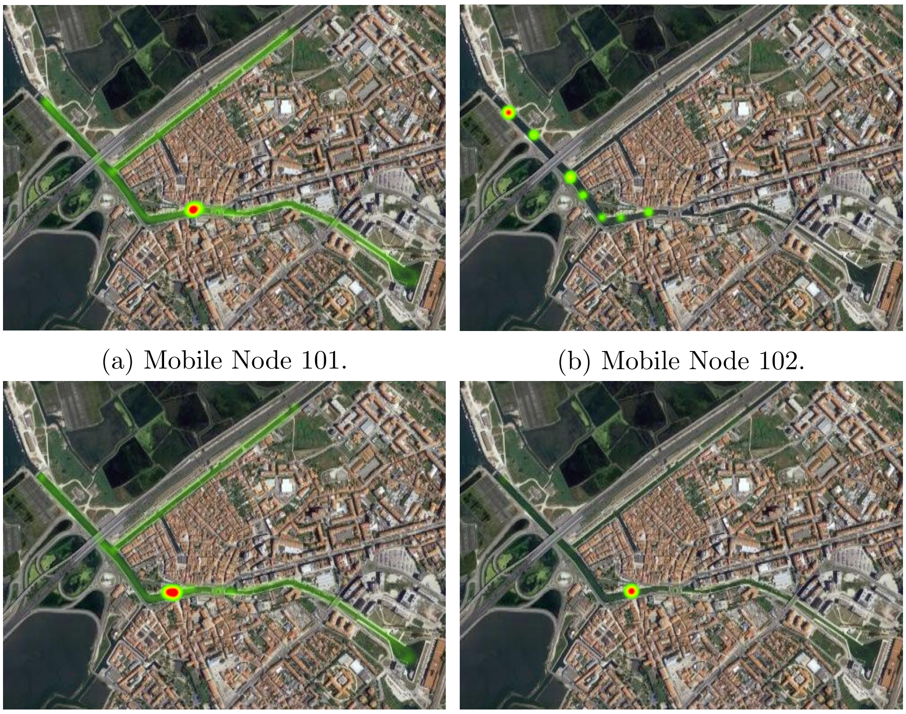
(a) Mobile Node 101.

(b) Mobile Node 102.

(c) Mobile Node 103.

(d) Mobile Node 104.

Figure 5.25: ASAR: Mobile nodes location heatmaps.



(a) Mobile Node 101. (b) Mobile Node 102.

(c) Mobile Node 103. (d) Mobile Node 104.

Figure 5.26: HYBRID: Mobile nodes location heatmaps.

Overall delivery ratio

Figure 5.27 illustrates the delivery ratio that represents the effective percentage of data packets delivered from DCUs to gateways. Considering all previously tests, these follow the same pattern, where multi-hop approaches outperform single-hop strategies, as it can be seen through the comparison of Direct Contact with the remaining strategies. For the multi-hop strategies, Epidemic has the highest delivery ratio, due to its flooding mechanism. GLA, ASAR and HYBRID reach similar delivery ratios, where ASAR delivery ratio is slightly higher than GLA and HYBRID.

The delivery ratio can reach higher values if the packet expiry time is higher and if more trips are performed. Regarding the first point, the packet network lifetime is 90 minutes, being dropped afterwards. If this network parameter is increased, packets will have more time to reach a gateway, thus increasing the delivery ratio. However, this increases the number of packets in storage, causing nodes' storage congestion. Regarding the second point, higher node mobility translates into higher delivery ratio, since nodes contact more frequently, being however out of control in this network.

Overall network overhead

The results of the network overhead are presented in Figure 5.28, representing the amount of redundant information that each ranking combination introduced in the network. As expected, the flooding mechanism of the Epidemic strategy has the highest network overhead. Coping with this, the direct delivery approach from the Direct Contact strategy translates

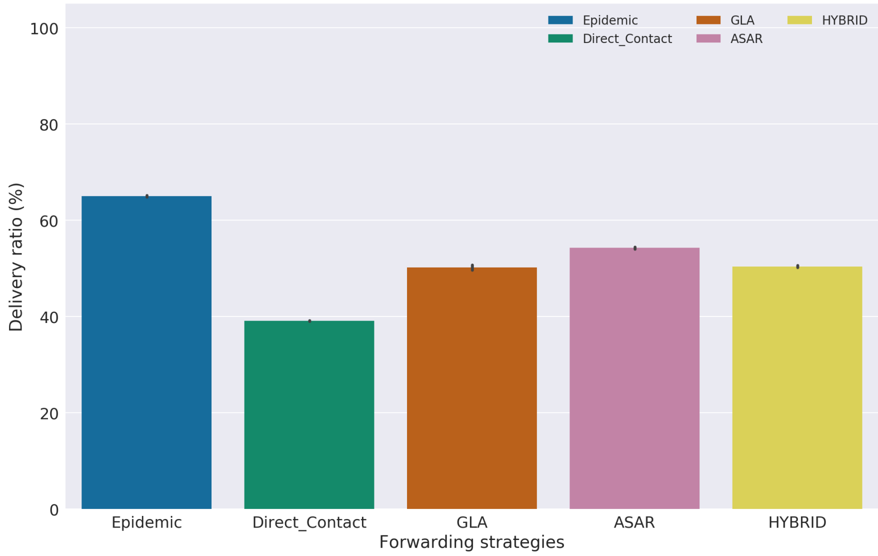


Figure 5.27: Overall delivery ratio per forwarding strategy, real network.

into almost no overhead. The proposed strategies have values slightly higher than the Direct Contact mechanism; however having a higher delivery ratio as previously observed in Figure 5.27. Among the proposed forwarding strategies, ASAR has the highest network overhead, which is related with the trade-off from the higher delivery ratio obtained.

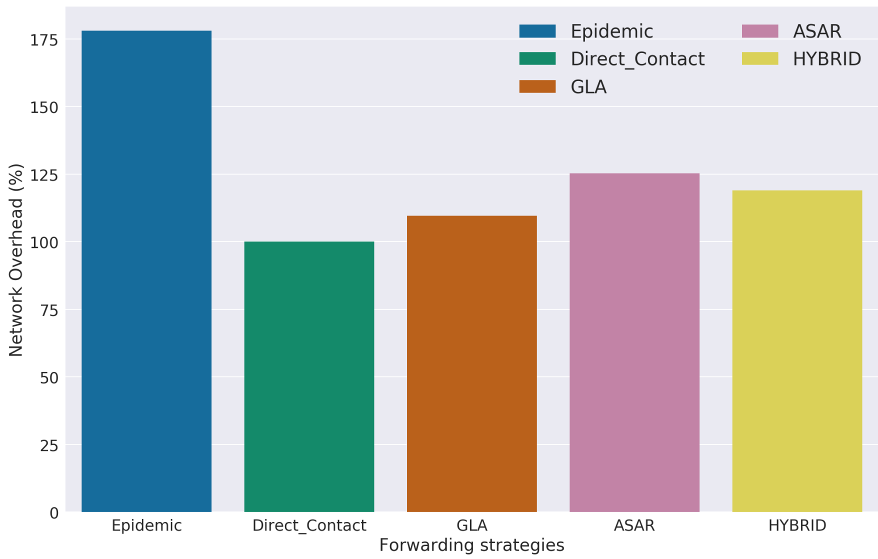


Figure 5.28: Overall network overhead per forwarding strategy, real network.

Overall network delay

Figure 5.29 shows the results for the network delay, representing the mean delay of the packets that were effectively delivered to a gateway. It should be remembered that boat trips have a typical duration of 45 minutes, and the time interval between trips is highly dependent on the presence of costumers. Therefore, it is expected that delays for this specific scenario to be around 1 hour.

This metric is the most influenced by the entropy associated with evaluating the strategies in distinct days. However, it is possible to observe that multi-hop approaches usually translate in lower mean packet delays when compared to the single-hop one. From the proposed forwarding strategies, ASAR has the highest delivery ratio, illustrated in Figure 5.27. However, the network delay is lower for GLA and HYBRID when compared to ASAR. This is expected, due to the lack of social behavior between boats. However, their trips are always confined between the same canals, where location metrics are able to perform better delay-aware decisions.

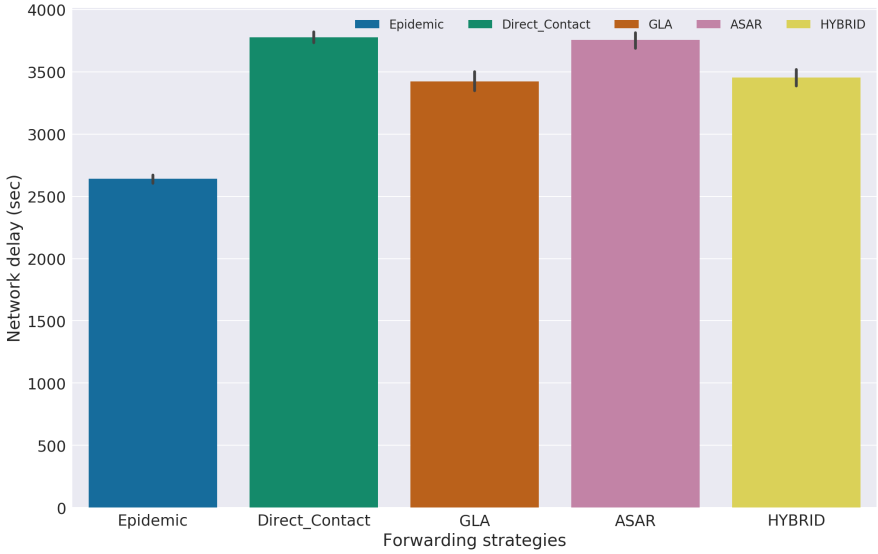


Figure 5.29: Overall network delay per forwarding strategy, real network.

Transmitted packets between mobile nodes

Figure 5.30 illustrates the percentage of packets transmitted from each specific mobile node to other mobile nodes in the network per strategy. For the Direct Contact, this value is 0 for all mobile nodes since they only transmit packets to gateways.

This metric shows important information: not all the same nodes had the same importance in the data forwarding mechanism for each strategy; the day where each test was performed has influence in the percentage of packets being forwarded per mobile node, since distinct number of travels were done per boat in distinct days. This is supported by the mobile nodes heatmaps figures.

Network dropped packets

Figure 5.31 illustrates the percentage of dropped packets per mobile node and per forwarding strategy. Packets that exceed their network lifetime are the main contributers to this percentage.

Mobile nodes 101 and 102 have a higher percentage of dropped packets when compared to mobile nodes 103 and 104, which happens due to their respective location. As can be seen

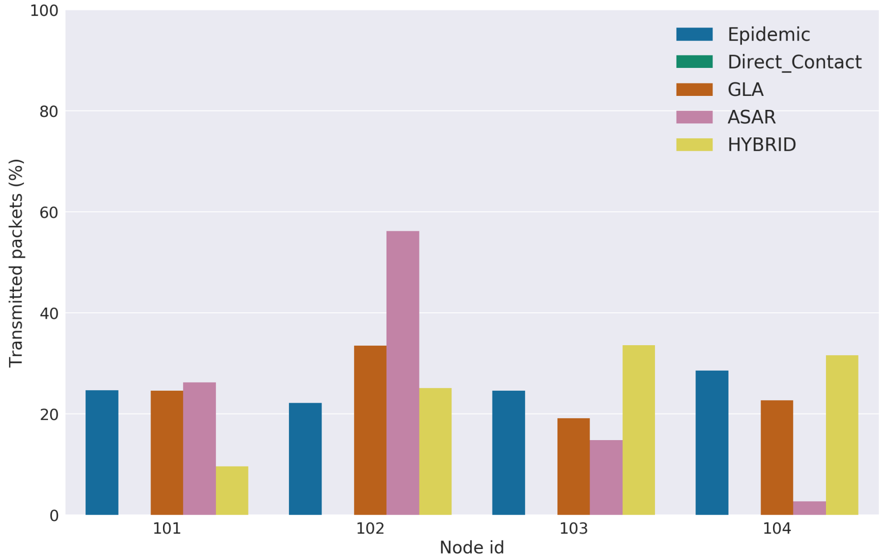


Figure 5.30: Transmitted packets per forwarding strategy, real network.

in Figure 5.21, nodes 101 and 102 rest between trips close to DCU 204, therefore receiving much more packets than mobile nodes 103 and 104. Due to the higher number of packets in storage, it is expected that mobile nodes 101 and 102 have a higher percentage of dropped packets.

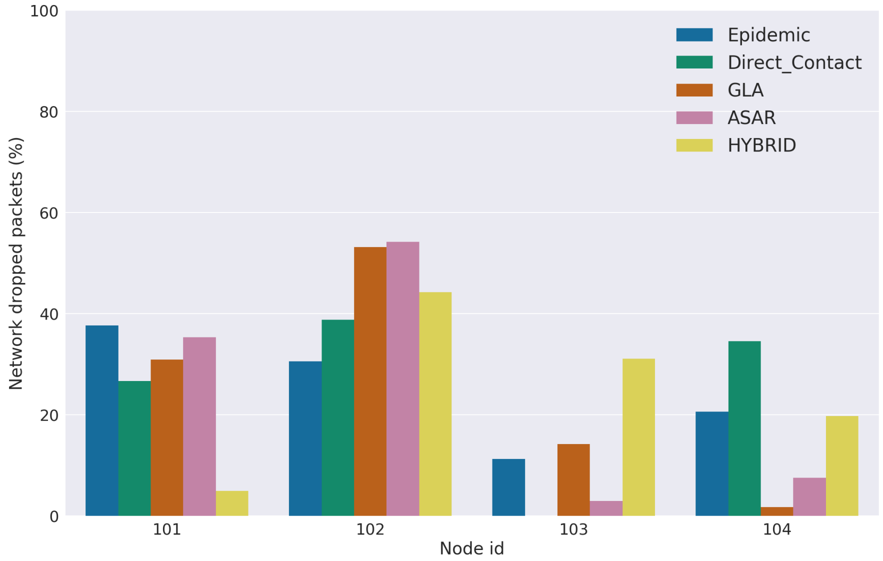


Figure 5.31: Dropped packets per forwarding strategy, real network.

5.5 Chapter Considerations

This chapter presented the overall methods used to evaluate the work proposed in this dissertation. To do so, the experiment scenarios were presented for each evaluation followed by the obtained results.

First, the BLE scanning energy evaluation was performed. The hardware and experimental setup was outlined, and results evidence that this scanning mechanism outperforms the WiFi scanning mechanism in energy consumption depending on the relation between number of mobile nodes and DCUs.

Afterwards, a partitioned evaluation through experiments was performed to evaluate the proposed forwarding strategies. First, a parameter impact analysis was performed for each strategy, followed by a comparison between forwarding strategies in the same scenario, and finally, considering several scenarios. This approach allowed to understand singular patterns as well as overall network changes for the evaluated strategies. The obtained results show that multi-hop strategies outperform single-hop strategies in terms of delivery ratio and network delay. Between the multi-hop strategies, GLA, ASAR and HYBRID schemes reach similar delivery ratios and network delays as Epidemic. However, the proposed strategies use much less network resources when compared to Epidemic, which is supported by the network overhead analysis. Direct Contact outperforms the proposed strategies in terms of network overhead (since it is a single-hop scheme), but the improvement observed in terms of delivery ratio and network delay makes these approaches a better trade-off in terms of overall network performance.

Also using the emulator, the packet selection mechanisms were evaluated, with focus given in the applicability of each method. The results show that the Distributed packet selection mechanism is able to prioritize certain data types, maintaining a good network performance. Considering the Equalized packet selection, the network performance increases since it uses the network state in order to select packets. This scheme is also capable of prioritize data types, however, not being so effective as the Distributed packet selection mechanism.

Finally, to validate the application in a real environment, a use case was defined and the considered forwarding strategies evaluated during several days. The randomness present in a practical experiment allows algorithmic adaptability validation. However, entropy is introduced due to the experiment being performed in distinct days and conditions. More than validating each strategy, the real experiments allowed validation for the architecture as a whole, where the software and hardware pieces interact in such a way that allows data gathering using a set of rules.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This dissertation aimed to enhance the data gathering in moving IoT networks through novel forwarding strategies and packet selection mechanisms. Also, the first steps into a energy efficient architecture were given, where DCUs use the BLE technology to discover neighbors.

The BLE scanning mechanism energy analysis shows that, when compared to the WiFi scan, the DCU node becomes more energy efficient. However, mobile nodes now need to use the BLE technology, which translates into a very small energy cost. The performed experiments show that the BLE mechanism will be more energy efficient when the relation between the number of mobile nodes and DCUs does not exceed 22 to 1.

In this work, several forwarding strategies for urban vehicular networks were proposed and evaluated. GLA, a location-aware ranking classification, makes use of velocity, heading angle and distance to the gateway, to select the vehicles that have higher chance to deliver the information in a shorter period of time, thus differentiating nodes through their movement patterns. On the other hand, a social-based algorithm, ASAR, exploits the social behaviours of each vehicle, where nodes are ranked based on a historical contact table, differentiating vehicles with a high number of contacts from those who barely contact with other vehicles. Finally, a compromise between the aforementioned strategies was evaluated (HYBRID), where the improvement in the classification criterion resulted in a better resource management, with a lower network overhead, while maintaining the same level of delivery ratio. The proposed strategies are able to closely follow the two extreme strategies, in terms of minimizing overhead (close to Direct contact overhead) and maximizing delivery ratio (close to Epidemic delivery), thus comprising good tradeoffs for the data gathering through vehicular networks in urban scenarios.

Moreover, packet selection mechanisms were proposed to provide differentiated levels of QoS, considering distinct applications. Distributed packet selection exhibits good packet type prioritization capacity, since it performs the packet selection firstly based on the packet type and secondly based on the packet network lifetime. This mechanism serves applications where some data types have more priority than others. Equalized packet selection provides a significant network performance improvement, while still maintaining a slight level of packet type selection (lower than distributed packet selection). The relative comparisons were made with a FIFO algorithm, which served as ground truth.

Finally, the experiments in a real scenario allowed not only architecture validation, but also individual protocol feasibility in the data gathering stages, where vehicles combined with network elements are able to support smart city applications.

6.2 Future Work

In such dynamic environment, there is always space for further improvements, such as:

Energy consumption Energy constrains are always a concern when developing battery devices, especially in IoT applications. Therefore, it is important to evaluate the current energy consumption for all network elements. Coping with this, improvements not only from the distinct nodes point of view, but also from the proposed protocol must be taken into account.

Arquitectural validation The experiments performed in this thesis were focused on validating the proposed protocols. It remains to test the architecture as a whole, considering all technologies. In doing so, an analysis through the same metrics would help to understand the improvements obtained through the multi-technology approach.

Multi-technology Integrate new technologies, such as IEEE 802.11p/WAVE, and build a traffic management controller, which provides data flow management according to the network conditions and/or application.

Scalability Expand the number of elements in the network, enabling city-scale validation in a real environment, understanding the current limitations and, if necessary, proposing a new scalable approach.

Packet Selection Techniques Compare the current proposed packet selection mechanisms in real scenarios and propose new ones.

Applications Create end user applications using this network architecture as a tool, thus expanding its capabilities from a environmental monitoring tool to other use cases.

Forwarding Strategies Automate the forwarding strategy performance evaluation and increase the number of state of art comparisons, thus providing a better understanding about the viability of the proposed algorithms. Also, extend the number of use case experiments in real scenarios.

Appendices

Appendix A

General Hardware Equipment

Each network node is based on a Raspberry Pi 3 Model B+ [1], a board whose hardware specifications are presented in Table A.1.

Proccessor	1.4 GHz 64-bit quad-core ARMv8 CPU
RAM	1 GB
Networking	2.4 GHz 802.11n Wireless LAN and BLE
Operating System	64-bit Raspian GNU

Table A.1: Raspberry Pi 3 Model B+ specifications.

Moreover, to achieve long range communication, a SX1272 LoRa module from Libelium is used. A representative image and respective specifications can be consulted in Figure A.1. A Multiprotocol Radio Shield (also manufactured by Libelium) integrates both the SX1272 LoRa module and the Raspberry Pi, working as connection bridge between both components.

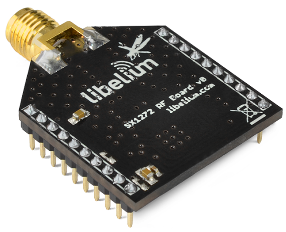
Finally, a board provides the connection to all sensors of interest. Displayed in Figure A.2, this board has an easy to repair structure and effortless plug and play connection to all sensors.

A list of possible sensors with developed and tested drivers is presented in Table A.2.

Sensor	**Acquired data**	**Communication**	**Observations**
DS18B20	Temperature	1-wire	Contains waterproof probe
BME280	Humidity, Barometric pressure, Temperature	I_2C	Capable of estimating altitude through pressure correlation
SI1145	Ultraviolet, Ambient Light	I_2C	None
MH-Z16 - NDIR	CO_2	I_2C	Requires callibration
SEN-12642	Sound Detector	ADC	None
MTK3339	GPS: Latitude, Longitude, Velocity, Altitude	Universal Asynchronous Receiver/Transmitter (UART)	Position accuracy under 3 meters

Table A.2: Sensors List.

[1] https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/ → Accessed:2018-07-11



(a) SX1272 LoRa module hardware.

LoRa	
Module	SX1272
Dual Frequency Band	863-870 MHz (Europe)
	902-928 MHz (US)
Transmission Power	25 mW
Sensitivity	-134 dBm
Channels	8 (868MHz)
	13 (900MHz)
Range	LOS = 21km (13.4miles)
	NLOS = +2km (1.2miles)

(b) SX1272 LoRa module specifications.

Figure A.1: SX1272 LoRa module [74].

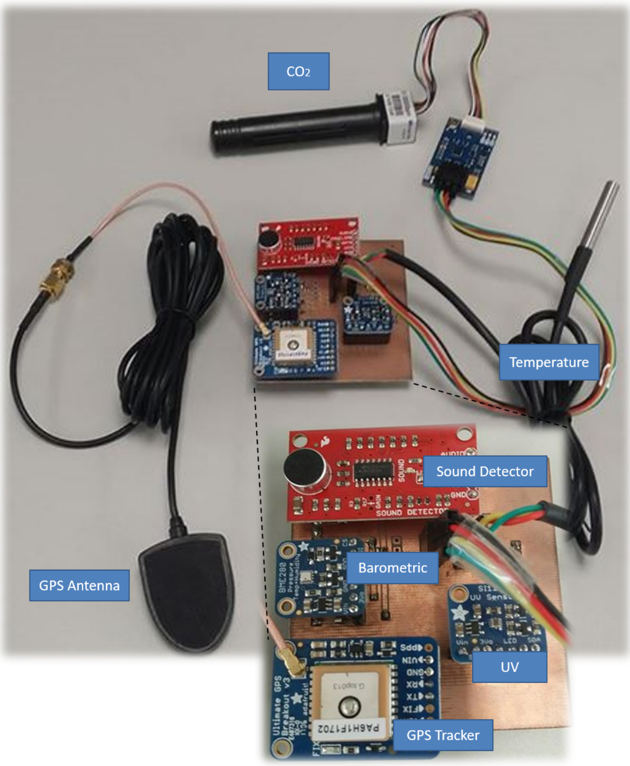


Figure A.2: PCB sensor board.

Bibliography

[1] Carles Anton-Haro and Mischa Dohler. *Machine-to-machine (M2M) communications: architecture, performance and applications*. Elsevier, 2014.

[2] Ericsson. White paper: Cellular networks for massive IoT - enabling low power wide area applications. `https://www.ericsson.com/assets/local/publications/white-papers/wp_iot.pdf`, 01 2016. Accessed: 2018-01-25.

[3] Nokia. White paper: Optimizing LTE for the Internet of Things. `https://novotech.com/docs/default-source/default-document-library/lte-m-optimizing-lte-for-the-internet-of-things.pdf?sfvrsn=0`, 05 2015. Accessed: 2018-01-25.

[4] Gartner. Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020. `https://www.gartner.com/newsroom/id/2636073`, 12 2013. Accessed: 2018-01-25.

[5] Ericsson. White paper: Ericsson mobility report - on the pulse of the networked society. `https://www.ericsson.com/assets/local/news/2016/03/ericsson-mobility-report-nov-2015.pdf`, 11 2015. Accessed: 2018-01-25.

[6] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2):855–873, 2017.

[7] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67, 2016.

[8] Mats Andersson. Short range low power wireless devices and internet of things (iot). In *Wireless Congress*, 2013.

[9] Alexandros-Apostolos A. Boulogeorgos, Panagiotis D. Diamantoulakis, and George K. Karagiannidis. Low power wide area networks (lpwans) for internet of things (iot) applications: Research challenges and future trends. *CoRR*, abs/1611.07449, 2016.

[10] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015.

[11] M. Li, P. Si, and Y. Zhang. Delay-Tolerant Data Traffic to Software-Defined Vehicular Networks with Mobile Edge Computing in Smart City. *IEEE Transactions on Vehicular Technology (Early Access)*, pages 1–1, 2018.

[12] P. Pereira, A. Casaca, JPC Rodrigues, V. Soares, J. Triay, and C. Cervello-Pastor. From delay-tolerant networks to vehicular delay-tolerant networks. *IEEE Communications Surveys & Tutorials*, 14(4):1166–1182, 2012.

[13] H. Ding, C. Zhang, Y. Cai, and Y. Fang. Smart cities on wheels: a newly emerging vehicular cognitive capability harvesting network for data transportation. *IEEE Wireless Communications*, 25(2):160–169, 2018.

[14] Y. R. B. Al-Mayouf, N. F. Abdullah, O. A. Mahdi, S. Khan, M. Ismail, M. Guizani, and S. H. Ahmed. Real-Time Intersection-Based Segment Aware Routing Algorithm for Urban Vehicular Networks. *IEEE Transactions on Intelligent Transportation Systems*, 19(7):2125–2141, July 2018.

[15] A. Fonseca and T. Vazão. Applicability of position-based routing for VANET in highways and urban environment. *Journal of Network and Computer Applications*, 36(3):961–973, 2013.

[16] K. Lin, J. Luo, L. Hu, M. Hossain, and A. Ghoneim. Localization based on social big data analysis in the vehicular networks. *IEEE Transactions on Industrial Informatics*, 13(4):1932–1940, 2017.

[17] Rodrigo Almeida, Rui Oliveira, Miguel Luís, Carlos Senna, and Susana Sargento. A multi-technology communication platform for urban mobile sensing. *Sensors*, 18(4):1184, 2018.

[18] Jacqueline Hom, Lauren Good, and Shuhui Yang. A survey of social-based routing protocols in delay tolerant networks. In *Computing, Networking and Communications (ICNC), 2017 International Conference on*, pages 788–792. IEEE, 2017.

[19] Dhanesh Raj, Maneesha Vinodini Ramesh, and Subhasri Duttagupta. Delay tolerant routing protocol for heterogeneous marine vehicular mobile ad-hoc network. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*, pages 461–466. IEEE, 2017.

[20] Fatima Zohra Benhamida, Abdelmadjid Bouabdellah, and Yacine Challal. Using delay tolerant network for the internet of things: Opportunities and challenges. In *Information and Communication Systems (ICICS), 2017 8th International Conference on*, pages 252–257. IEEE, 2017.

[21] Forrest Warthman et al. Delay-and disruption-tolerant networks (dtns). *A Tutorial. Interplanetary Internet Special Interest Group*, 2012.

[22] Xiao Du, Wuwen Lai, Jie Zhang, and Hua Wang. A motion and counting based-routing algorithm for delay tolerant network. *International Journal of Communications, Network and System Sciences*, 10(05):14, 2017.

[23] Melissa Ho Jain and Robin Patra. Implementing delay tolerant networking. 2003.

[24] Sebastian Schildt, Johannes Morgenroth, Wolf-Bastian Pöttner, and Lars Wolf. Ibr-dtn: A lightweight, modular and highly portable bundle protocol implementation. *Electronic Communications of the EASST*, 37, 2011.

[25] K. Scott and S. Burleigh. Bundle Protocol Specification, RFC 5050. `http://www.ietf.org/rfc/rfc5050.txt`. Accessed:29-10-2018.

[26] G. Pessoa, R. Dias, T. Condeixa, J. Azevedo, L. Guardalben, and S. Sargento. Content distribution emulation for vehicular networks. In *2017 Wireless Days*, pages 208–211, March 2017.

[27] Network Architectures and Protocols (NAP). `http://www.nap.av.it.pt/`, 01 2018.

[28] Rodrigo Almeida, Ruben Oliveira, Miguel Luis, Carlos Senna, and Susana Sargento. Forwarding strategies for future mobile smart city networks. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–7. IEEE, 2018.

[29] Sushant Jain, Kevin Fall, and Rabin Patra. *Routing in a delay tolerant network*, volume 34. ACM, 2004.

[30] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S Raghavendra. Single-copy routing in intermittently connected mobile networks. In *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pages 235–244. IEEE, 2004.

[31] Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. http://issg.cs.duke.edu/epidemic/epidemic.pdf, 2000.

[32] John Burgess, Brian Gallagher, David Jensen, and Brian Neil Levine. Maxprop: Routing for vehicle-based disruption-tolerant networks. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–11. IEEE, 2006.

[33] 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*, pages 252–259. ACM, 2005.

[34] Anders Lindgren, Avri Doria, and Olov Schelen. Probabilistic routing in intermittently connected networks. In *Service assurance with partial and intermittent resources*, pages 239–254. Springer, 2004.

[35] Chen Yu, Zhongqiu Tu, Dezhong Yao, Feng Lu, and Hai Jin. Probabilistic routing algorithm based on contact duration and message redundancy in delay tolerant network. *International Journal of Communication Systems*, 29(16):2416–2426, 2016.

[36] Etienne CR De Oliveira and Celio VN De Albuquerque. Nectar: a dtn routing protocol based on neighborhood contact history. In *Proceedings of the 2009 ACM symposium on Applied Computing*, pages 40–46. ACM, 2009.

[37] Matthias Grossglauser and Martin Vetterli. Locating mobile nodes with ease: learning efficient routes from encounter histories alone. *IEEE/ACM Transactions on Networking*, 14(3):457–469, 2006.

[38] Vasco NGJ Soares, Joel JPC Rodrigues, and Farid Farahmand. Geospray: A geographic routing protocol for vehicular delay-tolerant networks. *Information Fusion*, 15:102–113, 2014.

[39] Vasco NGJ Soares, Joel JPC Rodrigues, and Farid Farahmand. Performance assessment of a geographic routing protocol for vehicular delay-tolerant networks. In *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, pages 2526–2531. IEEE, 2012.

[40] Yue Cao, Zhili Sun, Ning Wang, Fang Yao, and Haitham Cruickshank. Converge-and-diverge: A geographic routing for delay/disruption-tolerant networks using a delegation replication approach. *IEEE Transactions on Vehicular Technology*, 62(5):2339–2343, 2013.

[41] Vijay Erramilli, Mark Crovella, Augustin Chaintreau, and Christophe Diot. Delegation forwarding. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 251–260. ACM, 2008.

[42] Christian Lochert, Martin Mauve, Holger Füßler, and Hannes Hartenstein. Geographic routing in city scenarios. *ACM SIGMOBILE mobile computing and communications review*, 9(1):69–72, 2005.

[43] Xiao-Tao Liu, Bin-Jie Hu, Zong-Heng Wei, and Zhi-Xiao Zhu. A congestion-aware gpcr routing protocol for vehicular ad-hoc network in urban scenarios. In *Communication Software and Networks (ICCSN), 2017 IEEE 9th International Conference on*, pages 166–170. IEEE, 2017.

[44] Kang Chen and Haiying Shen. Greedyflow: Distributed greedy packet routing between landmarks in dtns. *Ad Hoc Networks*, 2018.

[45] Hyunwoo Kang and Dongkyun Kim. Hvr: History-based vector routing for delay tolerant networks. In *Computer Communications and Networks, 2009. ICCCN 2009. Proceedings of 18th Internatonal Conference on*, pages 1–6. IEEE, 2009.

[46] Hyunwoo Kang and Dongkyun Kim. Vector routing for delay tolerant networks. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5. IEEE, 2008.

[47] Gil Eduardo de Andrade, Luiz Augusto de Paula Lima, Alcides Calsavara, and Carlos Alberto Maziero. Routing protocol based on the position, velocity, and direction of the nodes. In *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, pages 971–976. IEEE, 2013.

[48] W. Qi, Q. Song, X. Wang, and L. Guo. Trajectory Data Mining-Based Routing in DTN-Enabled Vehicular Ad Hoc Networks. *IEEE Access*, 5:24128–24138, 2017.

[49] Chau Chang, Ming-Han Hung, Ching-Ru Chang, and Chin-En Yen. Ntr: An efficient trajectory-based routing protocol for the vehicular delay tolerant networks. In *Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on*, pages 389–394. IEEE, 2017.

[50] Ana Cristina Kochem Vendramin, Anelise Munaretto, Myriam Regattieri Delgado, and Aline Carneiro Viana. Grant: Inferring best forwarders from complex networks dynamics through a greedy ant colony optimization. *Computer Networks*, 56(3):997–1015, 2012.

[51] Elizabeth M Daly and Mads Haahr. Social network analysis for routing in disconnected delay-tolerant manets. In *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing*, pages 32–40. ACM, 2007.

[52] Elizabeth M Daly and Mads Haahr. Social network analysis for information flow in disconnected delay-tolerant manets. *IEEE Transactions on Mobile Computing*, 8(5):606–621, 2009.

[53] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: Social-based forwarding in delay-tolerant networks. *IEEE Transactions on Mobile Computing*, 10(11):1576–1589, 2011.

[54] Mary R Schurgot, Cristina Comaniciu, and Katia Jaffres-Runser. Beyond traditional dtn routing: social networks for opportunistic communication. *IEEE Communications Magazine*, 50(7), 2012.

[55] Jialu Fan, Jiming Chen, Yuan Du, Ping Wang, and Youxian Sun. Delque: A socially aware delegation query scheme in delay-tolerant networks. *IEEE Transactions on Vehicular Technology*, 60(5):2181–2193, 2011.

[56] Abderrahmen Mtibaa and Khaled A Harras. Caf: Community aware framework for large scale mobile opportunistic networks. *Computer Communications*, 36(2):180–190, 2013.

[57] B. Wu, H. Shen, and K. Chen. Exploiting Active Subareas for Multicopy Routing in VDTNs. *IEEE Transactions on Vehicular Technology*, 67(5):4374–4388, May 2018.

[58] Zhixiao Zhu, Binjie Hu, Haina Song, and Xiaotao Liu. A social-aware routing protocol for two different scenarios in vehicular ad-hoc network. In *International Conference on Communicatins and Networking in China*, pages 563–575. Springer, 2017.

[59] Pan Hui, Eiko Yoneki, Shu Yan Chan, and Jon Crowcroft. Distributed community detection in delay tolerant networks. In *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture*, page 7. ACM, 2007.

[60] D. Moura, L. Guardalben, M. Lus, and S. Sargento. A drone-quality delay tolerant routing approach for aquatic drones scenarios. In *IEEE IEEE Globecom 2017 - 8th International Workshop on Wireless Networking and Control for Unmanned Autonomous Vehicles*, pages –, December 2017.

[61] Nabil Benamar, Kamal D Singh, Maria Benamar, Driss El Ouadghiri, and Jean-Marie Bonnin. Routing protocols in vehicular delay tolerant networks: A comprehensive survey. *Computer Communications*, 48:141–158, 2014.

[62] Jiong Jin, Jayavardhana Gubbi, Slaven Marusic, and Marimuthu Palaniswami. An information framework for creating a smart city through internet of things. *IEEE Internet of Things Journal*, 1(2):112–121, 2014.

[63] Pedro M Santos, João GP Rodrigues, Susana B Cruz, Tiago Lourenço, Pedro M dOrey, Yunior Luis, Cecília Rocha, Sofia Sousa, Sérgio Crisóstomo, Cristina Queirós, et al. Portolivinglab: an iot-based sensing platform for smart cities. *IEEE Internet of Things Journal*, 2018.

[64] João GP Rodrigues, Ana Aguiar, and Cristina Queirós. Opportunistic mobile crowdsensing for gathering mobility information: Lessons learned. In *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*, pages 1654–1660. IEEE, 2016.

[65] Yunior Luis, Pedro M Santos, Tiago Lourenco, Carlos Pérez-Penichet, Tânia Calcada, and Ana Aguiar. Urbansense: An urban-scale sensing platform for the internet of things. In *Smart Cities Conference (ISC2), 2016 IEEE International*, pages 1–6. IEEE, 2016.

[66] Carlos Ameixieira, André Cardote, Filipe Neves, Rui Meireles, Susana Sargento, Luís Coelho, Joao Afonso, Bruno Areias, Eduardo Mota, Rui Costa, et al. Harbornet: a real-world testbed for vehicular networks. *IEEE Communications magazine*, 52(9):108–114, 2014.

[67] Steven Latre, Philip Leroux, Tanguy Coenen, Bart Braem, Pieter Ballon, and Piet Demeester. City of things: An integrated and multi-technology testbed for iot smart city experiments. In *Smart Cities Conference (ISC2), 2016 IEEE International*, pages 1–8. IEEE, 2016.

[68] Luis Sanchez, Luis Muñoz, Jose Antonio Galache, Pablo Sotres, Juan R Santana, Veronica Gutierrez, Rajiv Ramdhany, Alex Gluhak, Srdjan Krco, Evangelos Theodoridis, et al. Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks*, 61:217–238, 2014.

[69] Bin Cheng, Salvatore Longo, Flavio Cirillo, Martin Bauer, and Ernoe Kovacs. Building a big data platform for smart cities: Experience and lessons from santander. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 592–599. IEEE, 2015.

[70] Timo Ojala. Open urban testbed for ubiquitous computing. In *Communications and Mobile Computing (CMC), 2010 International Conference on*, volume 1, pages 442–447. IEEE, 2010.

[71] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

[72] H Arasteh, V Hosseinnezhad, V Loia, A Tommasetti, O Troisi, M Shafie-Khah, and P Siano. Iot-based smart cities: a survey. In *Environment and Electrical Engineering (EEEIC), 2016 IEEE 16th International Conference on*, pages 1–6. IEEE, 2016.

[73] HM8112 Datasheet. `https://cdn.rohde-schwarz.com/hameg-archive/HM8112_Englisch.pdf`. Accessed:14-8-2018.

[74] Cooking Hacks LoRa 868/900 MHz SX1272 LoRa module for Arduino Waspmote and Raspeberry Pi. `https://www.cooking-hacks.com/documentation/tutorials/extreme-range-lora-sx1272-module-shield-arduino-raspberry-pi-intel-galileo`. Accessed: 20-08-2018.