SIMULATION BASED LORA BASE STATION PLACEMENT OPTIMIZATION FOR SMART CITY SCENARIOS

by

Görkem Karadeniz

B.S., Computer Engineering, Boğaziçi University, 2015

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2019

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor and MSc advisor Prof. Cem Ersoy for his endless energy, his motivational talks and his ability to always find time to help and share his wisdom. I am grateful to him for his guidance without which this work would not be as it is.

I would also like to thank to my thesis co-supervisor Dr. Sinan Işık for sharing his knowledge and for his support during my thesis, and during my studies towards BSc.

Finally, I would like to express my deepest gratitude to my family for their endless support and understanding through all stages of my education. Their love and faith in me help me overcome many barriers.

ABSTRACT

SIMULATION BASED LORA BASE STATION PLACEMENT OPTIMIZATION FOR SMART CITY SCENARIOS

With emerging IoT technologies and vision for smarter environments, the need for low power wide area (LPWA) communications have substantially increased. In the last decade, many LPWA technologies have been developed in order to fulfil the needs of IoT and wireless sensor network applications. Although wireless sensor network design, radio network design and antenna placement problems for cellular networks are thoroughly studied in the literature, the works on optimization of LPWA networks are scarcer. In this thesis, we mainly focus on a specific LPWA technology, LoRa and design of LoRa networks through optimal placement of LoRa gateways. We have designed and implemented a novel simulation based optimization system and proposed multiple algorithms for optimizing LoRa gateway placement. In order to achieve better placement of gateways, use of genetic algorithm and clustering algorithm based methods with multiple heuristics are proposed. Success of the proposed algorithms are evaluated via simulation and the promising results are shared with the reader.

ÖZET

AKILLI ŞEHİR SENARYOLARI İÇİN BENZETİM TABANLI LORA BAZ İSTASYONU KONUMLANDIRMA OPTİMİZASYONU

Günümüzde nesnelerin interneti (IoT) alanındaki gelişmeler ve akıllı ortamlara dair genişleyen vizyon, düşük güç tüketimi ve geniş kapsama alanı sağlayan (LPWA) teknolojilerine duyulan ihtiyacı büyük ölçüde artırmıştır. Geçtiğimiz on yılda, nesnelerin interneti ve kablosuz algılayıcı ağ uygulamalarının gelişen ihtiyaçlarını karşılamak için birçok düşük güç geniş alan teknolojisi geliştirilmiştir. Kablosuz algılayıcı ağ tasarımı, radyo ağ tasarımı ve hücresel ağlar için anten yerleştirme problemleri akademik çalışmalarda çokça kez ayrıntılı olarak incelenmesine rağmen, LPWA ağlarının optimizasyonu ile ilgili çalışmalar daha azdır. Bu tez içerisinde, temel olarak bir LPWA türü olan LoRa teknolojisi ve LoRa ağ geçitlerinin en uygun şekilde yerleştirilmesi yoluyla LoRa ağ tasarımına odaklanmaktayız. Bu amaçla, çalışmamızda benzetim tabanlı bir optimizasyon sistemi geliştirilmiş ve LoRa ağ geçidi yerleşimini eniyilemek için birden fazla algoritma önerilmiştir. Ağ geçitlerinin daha iyi yerleştirilmesini sağlamak için, genetik algoritma ve kümeleme algoritması tabanlı algoritmaların farklı sezgisel yöntemler ile birlikte kullanılması önerilmiştir. Önerilen algoritmalar gerçekleştirilen benzetimler üzerinden değerlendirilmiş ve elde edilen umut verici sonuçlar okuyucu ile paylaşılmıştır.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iii
ABSTRACT iv
ÖZET v
LIST OF FIGURES viii
LIST OF TABLES xii
LIST OF SYMBOLS xiv
LIST OF ACRONYMS/ABBREVIATIONS xvi
1. INTRODUCTION 1
2. SMART CITY AND LPWA TECHNOLOGIES IN LITERATURE 4
2.1. Possible use of IoT for Smart City Applications 4
2.2. Current State of LPWAN Protocols 7
2.3. LoRa 11
2.4. The configurable transmission options of LoRa Physical layer 13
2.4.1. Center Carrier Frequency 13
2.4.2. Other Limitations 13
2.4.3. Spreading Factor 14
2.4.4. Coding Rate 14
2.4.5. Transmission Power 15
2.4.6. Bandwidth 15
2.5. General Properties and Factors That Affect LoRa Transmission 15
2.5.1. Data Rate 15
2.5.2. LoRa Packet Structure 16
2.5.3. Packet Reception 16
2.5.4. LoRa Packet Airtime 19
2.5.5. Collision Behaviour 19
2.5.6. LoRa Gateway Capacity 21
2.6. Base Station / Sink placement Optimization on wireless Networks . . . 21
3. LORA GATEWAY PLACEMENT PROBLEM 29

3.1. Assumption for LoRa Maximum Successful Range 32
3.2. Packet Transmission Random Process 33
3.3. Packet Reception 33
3.4. Packet Collision 34
3.5. Successful Packet Reception 35
3.6. Objective Function 35
3.7. Optimization Constraints 35
4. LORA SIMULATOR BASED OPTIMIZER DESIGN 37
4.1. Implementation Architecture 37
4.1.1. Environment Generation 38
4.1.2. Simulator 43
4.1.3. Optimizer 43
4.2. Default Values and Preferred Options in Implementation 45
4.2.1. Implemented Path Loss Models 45
4.2.1.1. Okumura Hata Path Loss Model 46
4.2.1.2. Dortmund Experiment Path Loss Model 46
4.3. Heuristic and Success Metric Selection 47
4.4. Proposed Optimization Algorithms 57
5. EXPERIMENTS AND RESULTS 70
5.1. Effect of Node Count 70
5.2. Effect of Packet Rate 74
5.3. Effect of Payload Length 77
5.4. Comparison of Proposed Solution Methods 82
5.4.1. Experiment 1 83
5.4.2. Experiment 2 91
6. CONCLUSION 94
6.1. Future Work 95
REFERENCES 96

LIST OF FIGURES

Figure 2.1. LoRa packet structure. 16

Figure 2.2. Collision cases for two transmissions that are studied in the work of Haxhibeqiri *et al.*. 20

Figure 2.3. Simplified multi sink large WSN design algorithm from the work of Oyman *et al.* 24

Figure 4.1. Simulation in the loop optimizer architecture. 37

Figure 4.2. Implemented LoRa Gateway Placement Algorithm. 39

Figure 4.3. Environment Generation Algorithm. 40

Figure 4.4. An example 9 km to 9 km problem area with 40000 nodes around 1 density point. 41

Figure 4.5. An example 13.5 km to 13.5 km problem area with 30000 nodes around 2 density points. 41

Figure 4.6. An example 13.5 km to 13.5 km problem area with 20000 nodes around 3 density points. 42

Figure 4.7. An example 13.5 km to 13.5 km problem area with 20000 nodes around 4 density points. 42

Figure 4.8. Simplified LoRa Simulation Algorithm 44

Figure 4.9. Comparison of path loss models for LoRa transmission at 868 MHz. 47

Figure 4.10. Collision probability, node count plot for each spreading factor (For packet and path loss options in Table 5.1). 51

Figure 4.11. Visualization of two gateway transmission collision scenario. . . . 53

Figure 4.12. Effect of joint node count to probability of complete packet loss due to collision. 54

Figure 4.13. Effect of multiple gateways simulation result. 56

Figure 4.14. K-Means Center Finding Algorithm. 59

Figure 4.15. Algorithm for finding best gateway locations via distance based K-Means. 60

Figure 4.16. Spreading Factor Guided K-Means Center Finding Algorithm. . . 63

Figure 4.17. Algorithm for finding best gateway locations via time of air (TOA) based K-Means. 64

Figure 4.18. Chromosome design of LoRa gateway coordinates. 66

Figure 4.19. CHC algorithm for finding best gateway locations. 67

Figure 4.20. CHC Half Uniform Crossover process on an example set of LoRa gateway coordinates. 68

Figure 5.1. Effect of node count on total transmission success. 72

Figure 5.2. Effect of node count on connected node transmission failure combined with ratio of disconnected nodes to total number of nodes. . 74

Figure 5.3. Average transmission success of each node for 24 hr, 1 packet/hr simulation of 10000 nodes and one LoRa gateway. 75

Figure 5.4. Average transmission success of each node for 24 hr, 1 packet/hr simulation of 100000 nodes and one LoRa gateway. 76

Figure 5.5. Effect of packet rate on total transmission success. 77

Figure 5.6. Effect of packet rate (packet/hr) on connected node transmission failure percentage. 78

Figure 5.7. Effect of payload size on total transmission success. 79

Figure 5.8. Effect of payload size on connected node transmission failure percentage. 80

Figure 5.9. Heat map of node success for 20000 node environment with payload length 32 bytes. 81

Figure 5.10. Heat map of node success for 20000 node environment with payload length 224 bytes. 81

Figure 5.11. Distribution of end nodes in Experiment 1. 84

Figure 5.12. Comparison of algorithms using Transmission Failure Probability Scores. 85

Figure 5.13. Comparison of algorithm execution times. 86

Figure 5.14. Successful transmission percentage with different number of gateways. 87

Figure 5.15. Percentage of disconnected nodes with different number of gateways. 89

Figure 5.16. Heat map of node success for gateway locations determined by CHC-Prob optimization with 4 gateways for Experiment 1. 90

Figure 5.17. Distribution of end nodes in Experiment 2. 91

Figure 5.18. Average transmission success of CHC-Prob solution with five gateways for Experiment 2. 92

Figure 5.19. Average transmission success with proposed algorithms for Experiment 2. 93

LIST OF TABLES

Table 2.1. ETSI 868 MHz ISM Band Duty Cycle Limitations. 13

Table 2.2. The Things Network LoRa Frequency Channels 14

Table 2.3. SX1301 Module sensitivity levels for 125 kHz. 17

Table 2.4. SX1301 Module sensitivity levels for 250 kHz. 17

Table 2.5. SX1301 Module sensitivity levels for 500 kHz 18

Table 2.6. iC880A gateway sensitivity levels. 18

Table 4.1. Packet and transmission options used for collision probability examples. 50

Table 4.2. Calculated probability of collision for a transmission among 2000 nodes using denoted spreading factor with single gateway (For packet and path loss options in Table 5.1). 51

Table 4.3. Algorithm Implementation Options. 69

Table 5.1. Problem Parameters and Chosen Options. 71

Table 5.2. Transmission success ratios for each number of end nodes with 95% confidence interval for Figure 5.1. 72

Table 5.3. Transmission failure values with confidence interval for Figure 5.2. 73

Table 5.4. Transmission success values with confidence interval for Figure 5.5. 77

Table 5.5. Transmission failure values with confidence interval for Figure 5.6. 77

Table 5.6. Average transmission success rates with confidence interval for Figure 5.7. 80

Table 5.7. Average transmission failure rates with confidence interval for Figure 5.8. 80

Table 5.8. Average failure probability scores with 95% confidence intervals for Figure 5.12. 85

Table 5.9. Average execution times of proposed algorithms with 95% confidence intervals for Figure 5.13. 86

Table 5.10. Successful transmission percentage of proposed algorithms with confidence interval for Figure 5.14. 88

LIST OF SYMBOLS

BW	Bandwidth
B_i	Location of base stations
C_i	Node locations
CR	Coding rate
h	Problem area height
h_{BS}	LoRa gateway antenna height
h_{node}	LoRa node antenna height
l	Grid size for CHC algorithm
n_{base}	Number of base stations
$n_{concurrent}$	Gateway concurrent packet Reception Limit
n_{node}	Node count
$n_{preamble}$	Preamble symbols
n_{repeat}	Repeat count for random selection
W_i	LoRa uplink transmission
w	Problem area width
P_i	Probability
PL	Total payload length including the header
$Power_{TX}$	Node transmission power
Q	Receiver sensitivity
r_{node}	Traffic requirements, packet rate per hour
S_{Base}	Cost of a single LoRaWAN base station
t_{air}	Packet transmission airtime
$t_{waitMin}$	Minimum time to wait after a transmission
z	Uplink transmission frequency
α	CHC algorithm random mutation probability
δ	CHC algorithm elitist selection percentage
η	number of tiles for spatial centers algorithm

ϑ	Duty cycle limitation
κ	Number of arrivals
σ	A log-normal random variable
τ	CHC algorithm mutation decision threshold
φ	Problem duration

LIST OF ACRONYMS/ABBREVIATIONS

BPSK	Binary Phase Shift Keying
CHC	Cross generational elitist selection, Heterogeneous recombination and Cataclysmic mutation
CLMH	Candidate Location With Minimum Hop
CNP	Centroid of the Nodes in a Partition
CR	Coding Rate
CRC	Cyclic Redundancy Check
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear to Send
DE	Differential Evolution
FDMA	Frequency Division Multiple Access
GPS	Global Positioning System
GSP	Geographic Sink Placement
HUX	Half Uniform Crossover
IoT	Internet of Things
LoRa	Long Range
LPWAN	Low Power Wide Area Network
MAC	Medium Access Control
NB-IoT	Narrowband IoT
OFDMA	Orthogonal Frequency-Division Multiple Access
PBIL	Population Based Incremental Learning
ROI	Return on Investment
RPMA	Random Phase Multiple Access
RSSI	Received Signal Strength Indicator
RTS	Request to Send
SA	Simulated Annealing
SCFDMA	Single Carrier Frequency Division Multiple Access
SF	Spreading Factor

SNR	Signal to Noise Ratio
TDMA	Time Division Multiple Access
TOA	Time of Air
TTN	The Things Network
WSN	Wireless Sensor Network

1. INTRODUCTION

Internet of Things (IoT) is a radical evolution of current Internet into a ubiquitous network of interconnected objects that harvest information from their environments interact with the physical world and also uses Internet standards to provide services [1]. Small and cheap computing devices combined with various sensing capabilities become more prevalent each day, this helps IoT vision to become reality. To this date along with production of many enabling devices such as embedded sensor and actuators, different communication technologies for fulfilment of various communication requirements in different domains have become available which in turn is helping the IoT vision to be achieved. Smart city also called as "Urban IoT" mainly aims use of these diverse technologies that allow smaller connected nodes to sense, control and monitor the urban environment for business and public welfare purposes. It is forecasted that by 2030 60% of the world population will be living in urban environments [2]. Urban IoT structures could help cities accommodate such increase in their populations. There is no exact formal definition for smart cities however the main aim is to provide better and more efficient use of city resources, increase public service quality and citizen contentment. Moreover, another objective of IoT based smart cities is given as to make savings by reducing operational costs in city administration [3]. It is stated that an IoT based smart city structure can provide better management and optimization of standard public services such as public transportation, lighting, maintenance of public areas, garbage collection, security of public and touristic sites [3]. With connected devices getting more popular, making it possible for users to access everyday items like home appliances, security tools, environmental sensors and actuators remotely; IoT devices and wireless sensor networks (WSN) will become an even more integrated part of our lives. It is told that this paradigm can find applications in many different domains in a smart city such as medical aid, elderly assistance, intelligent energy management, smart grids, traffic management [3]. Moreover, the data collected from some of the IoT services such as traffic, air & noise pollution data, weather data could be directly shared with citizens of a smart city giving them the feeling of being active and in control

of their environment. This could provide an opportunity for municipal authorities to inform citizens with results of the work being done and can increase citizen satisfaction.

The terms Internet of Things (IoT) and Internet of Everything suggest the idea of everyday objects being connected become increasingly popular in the recent years. The number of connected devices are expected to reach 50 billion by the year 2020 [4]. It is stated that by the year 2024, the IoT industry together with its manifacturing, network and value added services to generate revenue of 4.3 trillion dollars [5]. In [6], shared forecasts express slightly different numbers; revenue of 7 trillion dollars and approximately 28 billion connected things. Although forecasts give different numbers, one common property is that all of the forecasts expect a large increase in both revenue and number of devices. Naturally with increasing number of devices that require being connected and accessible, importance of successful and scalable communication technologies that can accommodate the necessary communication requirements also increases. The main requirements for the IoT and simple sensor & actuator nodes, generally include low-powered communication protocols since most of the nodes are meant to be battery powered and low cost. There are numerous technologies designed for connection of wireless sensors and any other smart nodes that are capable and in need of sending information and receiving commands. Some of the wireless technologies are well known and well established in our daily lives such as Bluetooth, IEEE 802.11, GSM, UMTS, LTE and some are less common and more recent such as Bluetooth LE, Zigbee, Weightless, Sigfox, LoRa, Narrow Band IoT. These technologies have both common and disjunctive properties. One of the disjunctive properties is the designed operational range of the wireless technology. 802.11, Bluetooth, Bluetooth LE, Zigbee are mainly short range technologies whereas GSM, UMTS, LTE, Sigfox, Weightless, NB-IoT are designed for possible use for long range communications. Operation of city wide battery powered, low cost, connected smart nodes requires use of wireless technologies that consume low power and provide long range. Such networks are generally called Low Power Wide Area Networks (LPWAN).

Cellular communication technologies such as GSM, UMTS, LTE, despite their support for long range, consume high amount of energy and are not suitable for scenarios including low and limited powered devices [7]. Low power wide area (LPWA) technologies can provide low power consumption and relatively longer transmission range with a trade off in the data rate thus LPWA technologies can mostly be used with applications that does not require high amount of data transmissions or continuous communication with the network.

In this thesis, we mainly focus on a specific LPWA technology called LoRa. First, smart city use cases are discussed, then some of the smart city experiments and trials in the literature are visited. In the following chapter, LoRa physical and logical properties and operational constraints are inspected, advantages and shortcomings of various other low power wide area network technologies are comparatively discussed. Then through gathered information, first a low power LoRa network deployment problem for a smart city / an urban environment is defined and various methods for determining the optimal locations for LoRa gateways are given. Afterwards we explain the implementation of a novel simulation & optimization program for LoRa network design. Later, the simulation system is used together with the proposed optimization methods and simulation results for various cases are shared with the reader together with their analysis. Finally, conclusions on the proposed problem and its solutions are given and possible future research topics are explained.

2. SMART CITY AND LPWA TECHNOLOGIES IN LITERATURE

2.1. Possible use of IoT for Smart City Applications

Business-wise, the commonly discussed method to speed up Smart City deployments and Smart City related IoT applications is to start with IoT services that have fast return on investment (ROI) profiles. The idea is that these faster ROI services will act like catalysts and create demand for other IoT services. Some of the discussed Smart City services from the literature that could be beneficial for both citizens and municipalities with fast return of investment are given in this section.

Structural Health Monitoring of Buildings: It is explained in [3] that historical buildings of a city are continuously checked for structural problems, for early detection of any defects and to identify parts that are affected the most by environmental conditions. Use of Urban IoT could help decrease the need for human labour required for this task i.e. by installing sensors that can measure vibration, stress and atmospheric effects on historical buildings.

Waste/Garbage Management: Garbage collection is given as one of the costly processes for municipalities; it is possible that by use of smart containers, the information about garbage amount could be processed and the collector truck routes could be optimized resulting in a reduced operational cost for waste collection and reduced carbon emission.

Air Quality: Air quality plays an important role in people's health; for that reason city could monitor air quality in parks, industrial sites, neighborhoods, near hospitals, schools and crowded areas. Monitoring air pollution can be beneficial for both the citizens and the authorities that it can help in both enforcement of regulations and acceleration of policies to reduce environmental issues [8]. Moreover, this information

could also be accessible by the citizens of the smart city who can use the information differently e.g. create a jogging path in the healthiest areas [3].

Noise Monitoring: Noise pollution is also an important issue in city life; although there are rules and regulations it is not always easy to enforce such rules because of the need for periodic controls. IoT can provide noise-monitoring services for this purpose and can help generation of noise maps in different areas in different times of the day. Such an IoT application is experimented in the city of Melbourne [1]. In this experiment the sound data is collected through a multi tier, multi hop design that utilize smartphone sensors, fixed noise mesurement end nodes stationed on traffic & street lights, relay nodes and gateways. Moreover, a more controversial service is also talked upon in [3] which is using sound detection algorithms to recognize brawls, glass crashes or other noise than can point to public safety issues. However, it is also stated that this would require use of microphones that could cause privacy concerns.

Smart Parking: With use of smart parking systems, departure and arrival of cars can be tracked in the city. This would mean that better parking space planning can be done by the municipal authorities. In addition, the smart parking service is helpful in multiple ways, less time cars are searching for parking space would mean less air pollution caused by CO emissions, lesser traffic and happier citizens. This service could also generate additional revenue by allowing short-term reservation of parking spots. Moreover, it would be possible to detect parking violations such as disabled parking spot violations [3, 8].

Smart Transportation: It is said that on the long run, the traffic congestion causes big losses for a city from both environmental and citizen productivity perspectives [8]. Making use of a smart transportation IoT application can help monitor travel times and dynamically offer better routes to its residents.

Smart Lighting: This IoT service in a Smart City could optimize street light intensity according to the time of the day and presence of people. Moreover, this can

be used as a fault detection, monitoring system for street lamps by informing authorities about broken or flickering lights and removing the need for periodic maintenance. A proof of concept smart lighting deployment took place in the City of Padova Italy. In the work of Cenedese *et al.*, Padova Smart City deployment is inspected as a practical application of IoT devices. The main application in this deployment is monitoring of public street lights to check correct operation of light system using wireless nodes. The end nodes are equipped with various sensors, such as light intensity, air quality, noise, temperature and humidity. The nodes are connected to the Internet via a gateway unit. It is explained that even with a simple application such as this, there are multiple devices and different link technologies involved [9].

Emergency Use Cases: Apart from value and revenue creating or cost reducing IoT applications, there is one more possible use case in Smart Cities as emergency applications. Some examples of such applications include flood detection and early warning systems, after earthquake structural information systems, historical site safety and integrity systems. These applications could provide safety feeling among citizens. A river monitoring use case IoT application was deployed in Dublin [10]. The main objective of this project was to make a demonstration of how IoT based Smart City applications can achieve gathering of environmental data within Dublin and make use of it. The main use case is the flood zone river monitoring application. For this river monitoring application, an Intel Edison node utilizing both a low cost LPWA technology LoRa and a 3G module is used. The end node is housed inside a buoy and it collected water depth, velocity, temperature, air temperature, humidity, pressure and GPS location data. The deployment site was a previous flood zone on River Liffey. The end node is explained to get its power from batteries that are charged through solar panels installed on the buoy. In order to save power, most of the time the node is kept at the sleep state and it only wakes up periodically to collect data and make transmission. The period of transmission for LoRa is every 10 minutes whereas transmission with 3G is done every 12 hours. The LoRa transmission is done towards the LoRa gateway, on the other hand with 3G the data is directly uploaded to the server.

Apart from the explained use cases and trials, IoT have many other beneficial application areas that are not limited with the fast ROI smart city domain such as wildlife monitoring, forest fire monitoring, agriculture, industrial asset monitoring, critical infrastructure monitoring, military applications, logistics applications and border monitoring most of which also require low power consumption and longer range.

2.2. Current State of LPWAN Protocols

Low Power Wide Area technologies are developed to fulfill the needs of IoT and Wireless sensor applications which require long range connectivity, lower energy consumption than traditional wireless communication technologies such as Bluetooth & IEEE 802.11 is able to provide. As briefly explained in the introduction chapter, these two properties come at the expense of lower data rates and higher latency. Thus it is evident that LPWA technologies are not the best solution for every problem in the IoT domain such as vehicle to vehicle and vehicle to infrastructure communication applications that require high reliability, low latency and much faster response times; video or media transmission applications that require higher data rates [5]. Since the LPWA paradigm is designed for many low capacity endpoints one other requirement is lower cost of transceiver modules used in end nodes. Generally, the duty cycle of LPWA end nodes are considerably lower than traditional wireless technologies, this is for multiple reasons: Firstly the use of many ISM sub-bands are strictly dependent on government enforced duty cycle regulations of 10%, 1% and/or 0.1% [11] and secondly the end nodes need to turn off radio components as much as possible in order to save energy. These design restrictions push medium access control methods used in LPWANs towards more opportunistic and random access methods. Hence, time division multiple access methods mostly are not used because of time synchronization requirements. Methods such as CSMA/CA, used in IEEE 802.15.4 and IEEE 802.11 or RTS/CTS schemes that can be used with IEEE 802.11 are not preferred because of extra signalling overhead and messaging requirements. In the work of Raza *et al.*, it is also stated that the choice of medium access controls for LPWA technologies are mostly limited to low overhead methods such as Aloha. Nevertheless, a few other LPWA tech-

nologies that are exceptions to this generalization are shared with the reader, such as Ingenu and NB-IoT that utilize TDMA for medium access [5].

In most of LPWAN designs, the network operates in a star topology which saves energy for low powered end nodes since they are not required to listen to and relay transmissions of other end nodes, depleting their limited batteries. Moreover, most of the LPWA technologies utilize ISM bands for license-free spectrum operation which makes deployment and operating costs to be much lower than other technologies that require licensed frequencies. Additionally, usage of lower than 1 GHz sub-bands helps with longer range communication because of higher obstacle penetration property of lower frequency signals. In the following paragraphs we briefly introduce different LPWA protocols and share their distinctive properties.

Sigfox is a low power wide area communication technology that is designed by a French startup company, which also operates its own network [7]. Sigfox also utilizes a star network topology where end nodes communicate with the base station using Binary Phase-Shift Keying (BPSK) modulation and with 100 Hz bandwidth. Sigfox uses ISM bands and because of low cost low power antenna design with a narrow bandwidth, the maximum throughput is said to be 100 bps. Uplink messages can have up to 12 bytes of payload and for downlink messages, the payload is limited to 8 bytes [12].

Weightless is another LPWA technology that is designed for mainly M2M communications. It has three standards Weightless-W, Weightless-N and Weightless-P. Weightless-N and Weightless-P are designed for ISM band usage wheras Weightless-W is designed to work in frequency whitespaces such as those between TV channels. Weightless-N uses Differential Binary Phase Shift Keying modulation [13]. Weightless-P and Weightless-N uses a Time Division Duplex scheme together with frequency hopping [14]. General Weightless technology features are given as low power consumption, approximately 2 dollar end node transceiver cost, and long outdoor range up to 10 km [15]. Weightless-P is the latest standard and it is explained to support both uplink and downlink communication up to 100 kbps with combined use of TDMA and

FDMA schemes [14].

DASH7 is explained to be an open standard by DASH7 alliance. Different than most of the LPWA technologies, DASH7 proposes a hierarchical two hop standard using end nodes, gateways and middle links called sub-controllers [6]. Currently this technology is not very widely used in the real world applications. DASH7 is stated to have approximately 2 km outdoor range and nominal data rate of 27.8 kbps. In the work of Cetinkaya *et al.*, a DASH7 based power metering system in designed, its operation and performance is evaluated [16].

Ingenu Random Phase Multiple Access (RPMA) is a proprietary technology of Ingenu which uses 2.4 Ghz unlicensed band with 80MHz of bandwidth. RPMA is explained to utilize Direct Sequence Spread Spectrum and is advertised to offer approximately 19 kbps throughput per MHz of bandwidth for each message [17].

Telensa is another proprietary Ultra Narrow Band technology that utilizes unlicensed ISM bands [18]. It is stated to use Frequency shift keying with support to data rates up to 62.5 bps for uplink and 500 bps for downlink [5].

Narrowband IoT (NB-IoT) is a new cellular communication technology that was introduced by the 3GPP consortium especially for wide area and low bandwidth requirements. The key requirements targeted by NB-IoT were low device complexity, cheaper modules, wide area coverage, low power consumption and support for high number of end nodes. The main advantage of NB-IoT is that it aims to make it possible for cellular network operators to quickly deploy in its existing network, using a small part of their own frequency. Although, NB-IoT is explained to be a not completely backward compatible protocol, it is designed to coexist with the current GSM and LTE technologies [19]. For the physical layer, it requires 180 kHz minimum bandwidth for downlink and uplink. It uses OFDMA (Orthogonal Frequency Division Multiple Access) for downlink and SC-FDMA (Single Carrier Frequency Division Multiple Access) for uplink both of which are from the currently used LTE standards. The peak data

rate for downlink is given as 226.7 kb/s and the peak data rate for uplink is 250 kb/s. Although the main target of NB-IoT is latency insensitive applications, it is explained that for applications that are sensitive such as alarm applications NB-IoT can guarantee less than 10s latency [19]. On the aspect of battery life, for an example of 200 bytes a day transmission device it is stated that 10-year battery life can be achieved. Moreover, NB-IoT can receive messages with coupling loss 20 dB more than LTE by trading off data rate (Highest coupling loss support – 170 dB – is said to be achieved with a data rate of 35 b/s) [19]. The capacity of a single NB-IoT cell that utilizes only a single 180 kHz frequency block is given as 52500 end nodes and since NB-IoT supports multiple carriers, it is possible to increase the capacity by adding another frequency block.

LoRa is another LPWA wireless communication technique, which will be discussed more in depth in Sections 2.4 and 2.3. Mainly, LoRa utilizes ISM bands and it is able to use bandwidths of 500, 250 and 125 kHz. LoRa uses ALOHA for medium access in order to decrease complexity and extra overhead which in turn can cause an increase in message loss due to collision. The key properties of LoRa are given as long range, high signal robustness, multipath resistance, doppler resistance and low power consumption [20]. In Europe, LoRa can use the unlicensed ISM bands 868 Mhz and 433 Mhz. Although LoRa can be used with any MAC layer protocol there is currently a proposed LoRaWAN MAC layer protocol which is based on a star of stars topology and mainly promoted by the LoRa Alliance and The Things Network.

For many smart city applications, long battery life and long distance communication is a strict requirement whereas low bandwidth and communication delay are acceptable. These IoT applications could benefit from the discussed LPWAN Technologies. From a network operator's perspective, one hop star topology is a business-wise better model and it is utilized in many of the LPWANs. Additionally, the single hop star topology allows lower energy consumption as the end nodes do not have to actively listen or relay each other's transmissions.

2.3. LoRa

LoRa is a proprietary physical layer wireless communication system which is designed for low power, spread spectrum, long range operation by Semtech [21]. It is explained to use Chirp (Compressed High Intensity Radar Pulse) Spread Spectrum Modulation [22]. Among other similar technologies LoRa has an advantage of using unlicensed ISM bands. This makes it possible for anyone to set up a LoRa network as long as the regulative restrictions are met. LoRa is upper layer protocol agnostic but is mainly being used with the MAC layer protocol LoRaWAN. LoRaWAN enforces a star topology operation where end nodes are connecting to LoRaWAN gateways. The wireless technology LoRa and the protocol LoRaWAN are both tailored for long range and low power application. However, the low range, unlicensed ISM band usage and lower power comes at a price of lower bandwidth. The theoretical upper bandwidth limit of LoRaWAN is given as 50 kbps. LoRaWAN defines 3 types of operating classes for end nodes that could be used for smart applications; A, B and C. Class A is for nodes that are to be used in the lowest possible power applications with limited downlink (from gateway to node) transmission. Class B is designed for nodes that could need periodic downlink traffic, meaning they could periodically wake up and listen for the gateway. Class C nodes listen for downlink traffic when they are not transmitting. In LoRaWAN the end devices directly communicate with a sink node called gateway and this eliminates the need for a multi-hop design and reduces the network complexity. However, this would also mean that only a few sink nodes (gateways) are to receive packets from a lot of end nodes and many nodes have to share the same unlicensed medium and gateway for communication. For this reason LoRaWAN utilizes configurable LoRa physical layer options such as the centre carrier frequency, spreading factor, bandwidth and coding rate. In the LoRa physical layer, six different chirp spread spectrum modulations are defined. Depending on these modulations the ratio between chip rate and symbol rate changes. For LoRa, these ratios are generally defined as spreading factors (SF) and there are 6 corresponding spreading factors from SF7 to SF12. There is also an SF6 with higher data rates which requires special operations and is mostly not included in LoRaWAN standards [23, 24]. The signals transmitting with different

SF, same frequency and same bandwidth are explained to be orthogonal to each other, meaning they do not interfere. According to the protocol specification document [25] the spreading factors change depending on the range requirements thus affecting the data rates. In the work of Augustin *et al.* [26], the LoRa physical layer is inspected in a suburb of Paris, the first spreading factor (SF7) providing the highest data rate achieves more than 80% packet delivery ratio (without any retransmission) for a distance of 650 m and achieves a packet delivery ratio just below 60% for a distance of 1400 m. Usage of only six spreading factors that are designed for different data rates could mean all nodes with similar distance to the gateway would potentially use the same spreading factor and could interfere with each other. The effect of collision that could be caused by the spreading factor schema is one of the areas that is in need of further investigation. Some of the main applications that LoRaWAN and other low power, long range solutions can be useful are given as Smart City / Smart Grid applications such as Smart Metering and Distribution Automation/control. In addition to these new requirements in the energy sector, there are other possible usages such as asset management that could help increasing the efficiency in logistics and inventory optimizations for public and private institutions [27]. Moreover, the technology could also prove useful for usage in rural areas such as for farming, irrigation automation and control. The deployment requirements for an IoT network backbone based on the LoRaWAN technology depend on the characteristics of the corresponding applications. The requirements for smart meters, irrigation equipment and asset management applications could be different in means of installation strategies of LoRaWAN gateways (base stations). The gateway positioning strategies could depend on the number of nodes to serve, geographical constraints, demographics and classes of the nodes served, mobility of the nodes, the criticality of the application, uplink and downlink needs. In the work of Schroder Filho *et al.*, it is advised that usage of LoRaWAN should be further investigated for advanced distribution automation applications since the latency requirements for those applications are given as less than 160ms. Generally, the LoRaWAN protocol was found suitable for applications with less strict latency and bandwidth requirements [27].

2.4. The configurable transmission options of LoRa Physical layer

2.4.1. Center Carrier Frequency

In Europe, LoRa can use the unlicensed ISM bands 868 Mhz and 433 Mhz. In the 868 Mhz band, there are duty cycle limitations as per ETSI EN300.220 standard [11]. The central frequencies and corresponding duty cycle limitations are shared with the reader in Table 2.1.

Table 2.1. ETSI 868 MHz ISM Band Duty Cycle Limitations.

ETSI Ref.	Frequency Band	Duty Cycle Limitation
g	863.0 – 868.0 MHz	1%
g1	868.0 – 868.6 MHz	1%
g2	868.7 – 869.2 MHz	0.1%
g3	869.4 – 869.65 MHz	10%
g4	869.7 – 870.0 MHz	1%

Additionally, in parts g, g1 and footnote 7 of the regional documents of Information and Communication Technologies Authority in Turkey [28], the frequency plan and duty cycle limitations are stated the same as the ETSI regulations for the frequencies given in Table 2.1. Furthermore, The Things Network (TTN) proposed using the following sub channels with central frequencies in 868 Mhz band in Europe for its public LoRaWAN network [29] for uplink (from node to gateway) traffic: 868.1 MHz, 868.3 MHz, 868.5 MHz, 867.1 MHz, 867.3 MHz, 867.5 MHz, 867.7 MHz, 867.9 MHz

2.4.2. Other Limitations

According to TTN, the possible uplink transmission central frequencies for LoRaWAN, corresponding spreading factors and bandwidths are given in Table 2.2. Apart from the governmental regulatory duty cycle limitations of LoRa, TTN LoRaWAN network also has its own limitations for their public community LoRaWAN network as

a fair access policy [30]. They limited the uplink node airtime in 24 hours to 30 seconds in total and 10 transmissions per day. Private LoRaWAN networks do not have to be compliant with the fair access policy as long as they are compliant with the governmental regulations.

Table 2.2. The Things Network LoRa Frequency Channels

Channel Central Frequency	**Spreading Factor and Bandwidth**
868.1 MHz	SF7BW125 to SF12BW125
868.3 MHz	SF7BW125 to SF12BW125 and SF7BW250
868.5 MHz	SF7BW125 to SF12BW125
867.1 MHz	SF7BW125 to SF12BW125
867.3 MHz	SF7BW125 to SF12BW125
867.5 MHz	SF7BW125 to SF12BW125
867.7 MHz	SF7BW125 to SF12BW125
867.9 MHz	SF7BW125 to SF12BW125

2.4.3. Spreading Factor

In the Semtech documentation, the relationship between the bit rate and the chip rate for LoRa modulation is given as $R_c = 2^{SF} R_b$ [31]. Higher spreading factor would mean better Signal to Noise ratio (SNR) at the cost of lower bandwidth. There are six Spreading factors being used with LoRa and LoRaWAN from SF7 to SF12. Each increment on the spreading factor would mean doubling the chips to signify a single symbol and thus would mean doubling the transmission duration.

2.4.4. Coding Rate

Coding Rate (CR) is explained as the Forward Error Correction Rate that protects the transmission against bursts of interference. Its value corresponds to the number of transmission bits that are going to be used in encoding of 4 information bits. 4 different CR values are defined for LoRa. $CR = \frac{4}{4+n}$ where for values of n from 1 to 4

the CR is 4/5, 4/6, 4/7 and 4/8 respectively. For a Coding Rate of 4/7, 7 bits are used for transmitting 4 bits. Thus as CR value gets lower, the chance of successful packet reception increases. It is stated that devices with different CR configurations can still communicate with each other as long as explicit headers state the CR information.

2.4.5. Transmission Power

Transmission power of LoRa radio is often limited between 2 dBm to 20 dBm. Regarding packet transmission, the transmission range is only affected by the transmission power; it is not affected by Spreading Factor (SF), Bandwidth (BW) and Coding Rate (CR). In Europe, ETSI regulations limit the transmission power for a node to 14 dBm.

2.4.6. Bandwidth

As bandwidth increases, data rate also increases thus air time of a packet decreases. It is stated that the LoRa physical layer can operate at 125 kHz, 250 kHz or 500 kHz bandwidths.

2.5. General Properties and Factors That Affect LoRa Transmission

2.5.1. Data Rate

The data rate for LoRa is dependent on the bandwidth, spreading factor, and the coding rate that is being used. The following formula from Semtech Lora documentation sheet [31] summarizes the data rate dependencies for LoRa:

$$DR = SF \times \left(\frac{Bandwidth}{2^{SF}} \right) \times CR$$

$$CR = \frac{4}{4+n}$$

2.5.2. LoRa Packet Structure

Bor *et al.* [20] summarized the packet structure as follows: There is a preamble in the packet structure which is defined to be between 6 to 65535 symbols. There exists an optional header that includes the information about the payload length, the forward error correction rate and if there is an optional CRC or not. The payload can be 1 to 255 bytes long and at the end of the payload, there can be an optional payload CRC. The preamble size is denoted by $n_{preamble} + 4.25$ and it is explained to be able to take values ranging from 10.25 to 65539.25 symbols [26].

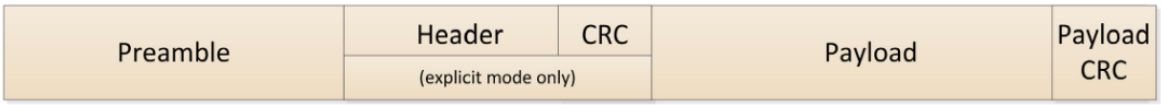

Figure 2.1. LoRa packet structure [31].

2.5.3. Packet Reception

For a transmission to successfully be received, the signal power at the receiving node must be above the receiver sensitivity. The radio receiver sensitivity can be calculated by using the following equation from Semtech LoRa Modem Design Guide documentation [31]:

$$S = -174 + 10\log_{10}(BW) + NF + SNR$$

Here NF is fixed for each hardware implementation, SNR is the signal to noise ratio. The signal strength at the receiver is calculated with the following formula where P_{rx} power of the signal at the receiver, P_{tx} is the transmission power, L_{pl} is the path loss and GL denotes the sum of all general gains and losses such as receiver and transmitter antenna gain, receiver and transmitter circuit and cabling losses and any

other miscellaneous losses.

$$P_{rx} = P_{tx} + GL – L_{pl}$$

A packet is successfully received if ($P_{rx} > S_{rx}$) meaning that the power of signal at the receiver, P_{rx} needs to be higher than the sensitivity, S_{rx} of the receiver module. A common LoRa gateway is based on the SX1301 chip and the sensitivity levels for the corresponding bandwidth and spreading factor usage are given in Tables 2.3, 2.4 and 2.5.

Table 2.3. SX1301 Module sensitivity levels for 125 kHz [32].

SF	**Data rate (bit/sec)**	**Sensitivity (dBm)**
7	5469	-126.5
8	3125	-129.0
9	1758	-131.5
10	977	-134.0
11	537	-136.5
12	293	-139.5

Table 2.4. SX1301 Module sensitivity levels for 250 kHz [32].

SF	**Data rate (bit/sec)**	**Sensitivity (dBm)**
7	10938	-123.5
8	6250	-126.0
9	3516	-128.5
10	1953	-131.0
11	1074	-133.5
12	586	-136.5

Table 2.5. SX1301 Module sensitivity levels for 500 kHz [32]

SF	**Data rate (bit/sec)**	**Sensitivity (dBm)**
7	21875	-120.5
8	12500	-123.0
9	7031	-125.5
10	3906	-128.0
11	2148	-130.5
12	1172	-133.5

The sensitivity for IC880A LoRa Gateway based on SX1257 and SX1301 modules is given as follows in the iC880A datasheet which are only slightly different than SX1301 sensitivity level which may be a result of the signal attenuation due to extra circuitry in the base station itself:

Table 2.6. iC880A gateway sensitivity levels [33].

Signal Bandwidth (kHz)	**Spreading Factor**	**Sensitivity (dBm)**
126	12	-137
125	7	-126
250	12	-126
250	7	-123
500	12	-134
500	7	-120

Additionally, according to the documents of TTN, the LoRa gateways are not able to receive any transmission while they are transmitting downlink [34].

2.5.4. LoRa Packet Airtime

Semtech documentation presents detailed equations and a computer program to calculate the airtime of a LoRa transmission [31].

$$T_{total} = T_{preamble} + T_{payload} \tag{2.1}$$

$$T_{preamble} = (n_{preamble} + 4.25)T_{symbol} \tag{2.2}$$

$$T_{symbol} = \frac{2^{SF}}{BW} \tag{2.3}$$

$$T_{payload} = \left(8 + max\left(ceil\left(\frac{8PL - 4SF + 28 + 16 - 20H}{4(SF - 2DE)}\right)(CR + 4), 0\right)\right) T_{symbol} \tag{2.4}$$

where

PL is payload bytes

SF denotes the spreading factor

H shows if header is enabled or not

DE shows if Low data rate optimization feature is enabled

CR denotes the Coding rate

2.5.5. Collision Behaviour

When two or more transmissions arrive at a LoRa receiver, there are multiple conditions that determine what can be decoded by the receiver or if anything can be decoded at all. Firstly, signals being transmitted in the same frequency and bandwidth but different spreading factors in LoRa modulation are orthogonal meaning that, they do not cause interference to each other. Secondly, two transmissions may not cause interference to each other at the receiver and can be decoded successfully if their carrier frequencies are not the same or difference in frequencies is above a defined (hardware dependent) threshold. Thus if ($|f_x - f_y| > f_{threshold}$) then the transmissions would not cause interference (It is stated that for Semtech SX1272 $f_{threshold}$ is 60 kHz for 125kHz bandwidth, 120 kHz for 250 kHz bandwidth and 240 kHz for 500 kHz bandwidth)

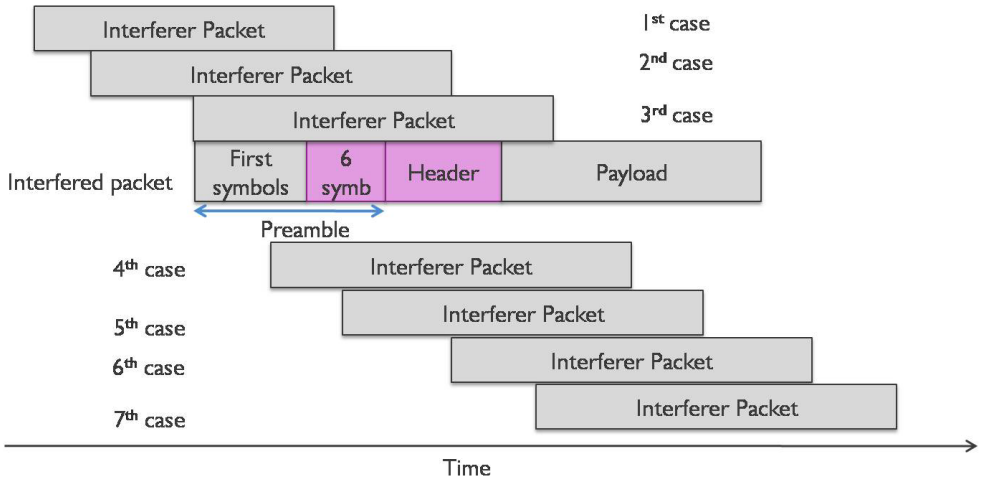

Figure 2.2. Collision Cases for Two Transmissions that are studied in the work of Haxhibeqiri *et al.* [35].

Finally Bor and Roedig explained that the amount of transmission overlap plays an important role in the successful reception. They indicated the following equation where for a transmision starting at time a and ending at time b the midpoint of transmission time is $m_i = \frac{a+b}{2}$, and half of the transmission duration is $d_i = \frac{(b-a)}{2}$ There will be interference if $(|mx - my| < dx + dy)$ [20].

In addition to above stated factors, there is a "capture effect" that can affect the success of a packet reception depending on received signal strengths of transmissions at the receiver which is not considered in our simulation system. Some packet collisions can be recoverable depending on the power and timing of overlapping signals and this situation is also studied in [20]. Similarly, the authors in [35] investigated the collision conditions in LoRa. They have also conducted experiments and concluded with the following rules regarding occurrence of a collision due to overlapping transmissions at same channel and spreading factors. The rules are explained with respect to seven collision cases denoted in Figure 2.2.

In summary, according to their experiments, depending on the power difference of two overlapping transmissions following rules are stated:

- The lower powered transmission cannot be received correctly in any case.
- If the two signals have equal RSSI or if the interfering signal has lower RSSI at the gateway, then for the cases 4,5,6 from Figure 2.2, there is approximately 90% chance of correct reception of the interfered signal.
- If the two signals have equal RSSI or if the interfering signal has lower RSSI at the gateway, then for the 7th case from Figure 2.2 the interfered signal is received correctly [35].
- For cases 1,2 and 3 from Figure 2.2 both packets are lost.

2.5.6. LoRa Gateway Capacity

LoRa gateways usually can successfully receive eight different concurrent transmissions as long as the concurrent transmissions do not cause interference to each other. An example to such gateways is IC880A LoRa gateway which utilizes SX1301 that can at most handle eight concurrent transmissions.

2.6. Base Station / Sink placement Optimization on wireless Networks

In this section literary works on base station/gateway/sink placement optimization for various types of wireless networks are surveyed. Additionally, since the problems declared in literature as "Radio Network Design" and "Antenna Placement Problem" have many similarities with our LoRa gateway placement problem; therefore, such relevant works are also inspected in this section.

Optimal Base Station positioning for wireless sensor networks (WSN) is stated to be an NP complete problem [36]. There are various proposed heuristics in the literature for WSN base station placement each focusing on different criteria such as battery life, lower delay, highest connectivity. The approaches which try optimization on a fixed topology are generally called static approaches. In this thesis, the researcher studies strategies to optimally place single and multiple LoRa gateways in a fixed network topology.

There are many different static and dynamic approaches for base station positioning in the literature with differing network models, environment types, protocols and assumptions. Akkaya *et al.* surveyed multiple Static Base station positioning approaches for wireless sensor networks in the literature [36]. For a single base station positioning, a proposed approach is finding the minimum enclosing circle and then positioning the base station at the center of the enclosing circle [37]. For maximum lifetime of all end nodes optimization problem with homogeneous nodes, it is stated that the problem of maximizing the minimum node lifetime turns to minimizing the maximum distance to the base station. In the work of Akkaya *et al.*, it is stated that the complexity of multiple base station placement problem is NP-Complete for a flat network or a clustered network topology where some sensors become cluster heads. If the nodes are grouped into individual clusters that are assigned to individual base stations, meaning the clusters are predefined in the given problem, the problem becomes similar to single BS positioning. However if clusters are not predefined, the problem is stated to be NP-complete again [36]. In order to lower the complexity of the problem, the available approaches in the literature try minimizing the candidate base station locations or try hybrid approaches such as partitioning the network into clusters first and then finding the optimal base station locations for each cluster. Two approaches for multiple base station positioning are aproximation algorithms such as greedy and local search and integer linear programming [36].

Additionally, we believe that it is possible to consider the Lora Gateway positioning problem as a facility location problem by designating the possible gateway locations and solving it as a mixed integer quadratic programming problem. On the subject of Base station placement, M.H.Wright from Bell Laboratories [38] stated in 1998 that Nelder-Mead "simplex" method is reliable at finding local optima and direct search methods can work more efficiently instead of finding an exact solution via multiple nonlinear equations for optimal placement of base stations. It is also stated in the literature that the facility location problem is NP hard [39]. Similarly, in their work on multi objective Antenna Placement Problem for Cellular networks, Raisanen *et al.* stated that with its many conflicting objectives the Antenna Placement Problem can

be classified as an extension of facility location problem where facilities are capacitated and demands are unsplittable [40].

In the work of Ilker Oyman and Cem Ersoy on the topic of multi sink large WSN design problem [41], a method for finding the optimal locations of multiple WSN sinks is investigated. There are both some similarities and differences between our LoRa gateway placement optimization problem and optimizing sink placement in a WSN. In both problem scenarios the alternatives of the sink/gateway locations are not enumerated, and sinks can be placed anywhere. However, in [41] multi hop communication of nodes are also a possibility whereas in our problem the LoRa nodes will only be communicating with the gateway. In the paper, several approaches are given for optimization of multi sink locations. One of them is finding the Best Sink Locations (BSL) when the number of gateways are known prior to the problem solution. The solution uses a method called k-means clustering. The idea is that if the Euclidean distance is used as a clustering metric the center of mass in a cluster would give the sink location for each cluster/partition. The second approach explained in the paper is for determining both the number of sink nodes and their locations (MSPOP) in order to achieve a predetermined network lifetime constraint. For this purpose, a brute force technique is used by increasing number of sink nodes and evaluating the network lifetime one by one until the required network lifetime is reached. The simplified algorithm from [41] is shared with the reader in Figure 2.3.

Oyman *et al.* used a network simulator to estimate network lifetime [41]; similarly, in our implemented LoRa Base Station placement optimization system in order to calculate viability of our current gateway location for each iteration, we developed a Simulation module. The details of our design will be explained in Chapter 4.

Taysi *et al.* also focused on the need of a good heuristic approach with the rationale that most variations of base station placement optimization problems, depending on assumptions and network models, are NP-Complete [42]. The authors proposed K-means Local+ algorithm as a solution for the problem of designing an energy efficient

1: Deploy the sensor nodes
2: Collect the location information from the nodes
3: $k = 0$
4: **repeat**
5: $k = k + 1$
6: Find the best location for k sink nodes
7: Estimate the network lifetime via Simulation
8: **until** Required network lifetime is reached
9: Output the sink locations, and the estimated lifetime

Figure 2.3. Simplified multi sink large WSN design algorithm from the work of Oyman *et al.* [41].

wireless sensor network.

As an alternative to clustering based approaches, Rehena *et al.* [43] have authored a paper on topology partitioning in wireless sensor networks, using an algorithm that is based on the K-Nearest Neighbours Graph. However, the proposed algorithm requires sink positions to be known beforehand and the main topic of the paper is to partition the graph using the given gateway locations whereas our optimization problem requires determining either the location or both the number and location of gateways. Thus in the following chapters, we mainly consider using graph partitioning techniques as a heuristic for the solution of the optimal gateway placement problem.

In the paper Multiple-Sink Placement Strategies in Wireless Sensor Networks, Das *et al.* [44] introduce two different algorithms for sink placement in wireless sensor networks and compare them to the existing Geographic Sink Placement (GSP) algorithm proposed by Poe *et al.* [45] . The proposed algorithms are called Candidate Location with Minimum Hop (CLMH) and Centroid of the Nodes in a Partition (CNP). They are designed for use in multi-hop wireless sensor networks. Both algorithms propose starting with a partitioned network and the paper does not state which ones should be chosen, the partitioning algorithm selection is left to the reader. For CLMH, after

partitioning is done, the next step is given as creation of equal size virtual square grids on the entire area and choosing candidate grids for placing the sinks/gateway. The square grid diagonal size is chosen as the range of single hop communication in WSN. Within a partition, initial candidate sink locations are chosen to be the centroids of each grid. In the results, authors stated that CLMH algorithm showed better battery life for the WSN; however, its execution time is longer. The initial location of the sink is determined as the center of the candidate grid in the CNP algorithm. Then the centroid of the 1 hop neighbours of the current sink location is chosen as a new location and its number of 1 hop neighbours are calculated if it is higher than the previous sink location, the new location is accepted and the process is repeated. If the number is the same or less, the current location is determined as the candidate sink location and iteration stops. To decide on the final location among candidate locations, hop counts for the farthest nodes to the candidate locations are calculated. The candidate location with the lowest hop count from the farthest node is chosen as the optimum sink location.

In [46] the authors inspected LoRa and LoRaWAN properties, summarized results of their survey on LoRa collision conditions, receiver sensitivities and medium access mechanisms. They also stated that the optimal gateway location for a number of nodes is the center point of a circle with minimum radius to cover all of the nodes which is also known as Chebyshev center. However the optimal placement strategy is only given for the case of a single gateway, cases with multiple gateways are not considered.

In [47], authors evaluated the LoRA performance both theoretically and by simulation. In the paper, detailed information on Link measurement and link performance models for LoRa are given. The authors state that they demonstrate via simulation that a gateway can serve nodes in the order of 10000 with 95% packet success rate. The authors applied the idea of static initial assignment of spreading factors to LoRa nodes; once nodes are given an SF depending on their best signal strength on one of the gateways; the SF values do not change. Although they assumed SX1301 based gateways in their

simulation implementation, they have taken older version of Semtech datasheets into account and the sensitivity values are slightly different. Authors used the sensitivity values for SX1301 and the following equation to calculate whether an uplink transmission is successfully received at the gateway or not: $P_{rx}^{dB} = P_{tx}^{dB} + G_{tx}^{dB} + G_{rx}^{dB} - L^{dB} + 4.34\sigma$. In the equation, the received signal power is equal to the transmission signal power plus the gateway and sender node antenna gains minus path loss plus 4.34 times a log-normal random variable. The path loss variable also consists of propagation path loss plus building path loss. The authors also used a heuristic model to simulate shadowing effects. They also utilized a SINR matrix to calculate the interference from transmissions on different spreading factors.

Genetic algorithms are bio-inspired evolutionary algorithms that can be used for the solution of optimization problems instead of traditional optimization algorithms. These algorithms utilize the natural evolutionary processes such as mutation, natural selection, recombination. For use with the base station placement problems, the base station locations are analogous to chromosomes and an evaluation function is used for evaluating the success of a solution which is analogous to fitness of a population member which affects its probability of survival. As explained in [48], the genetic algorithms usually start with a population which consists of a group of generally random solutions to the problem. Then, genetic processes are run iteratively on the population such as mutation and selection. With mutation, diversity is provided in the population. Using survival of the fittest and selection approaches support better solutions to continue their "bloodlines" to the subsequent generations (solution sets). At each iteration, modifications to the solution set make the population get closer towards the optimal solution. Tabu search is explained as an intelligent meta heuristic algorithm that discourages searches for a solution in certain spaces called tabu lists [48]. These tabu lists are generally the close proximities of each base station for a base station placement problem and at each iteration.

As Lora network uses Star topology for handling application end nodes, the structure of network resembles that of cellular networks and it is beneficial to consider base

station optimization techniques used in cellular networks. Regulna *et al.* worked on use of genetic algorithms for base station optimization of mobile networks. Their work focused on genetic and hybrid algorithms [48]. Their hybrid algorithm design combined both the genetic and tabu search approaches. In their work, they compared a direct genetic approach to a hybrid approach. For the design of hybrid algorithm, it is explained that the genetic algorithm performs better at initial phases and because of that the initial solutions are given by the genetic algorithm and from that given solution tabu search algorithm is ran. Another difference is that their hybrid algorithm tabu search includes a mutation probability that adds randomness saves searches from getting stuck in a small search space. Moreover the authors stated that the mutation probability for tabu search is also dynamic such that while the algorithm is obtaining better solutions, the mutation probability is lower in order not to disturb the optimization process and if the algorithm is not able to find better solutions the mutation probability is set to higher values allowing for a broader search space. Regulna *et al.* concluded by stating that the hybrid algorithm performs better in calculating optimal base station locations and also runs faster than the regular genetic algorithms [48].

In the work of Raisanen *et al.*, multi objective optimization methods with different genetic algorithms are inspected and comparatively evaluated. The performance of four genetic algorithms SPEA2, NSGA-II, PESA and SEAMO are evaluated in the areas of speed of execution, speed of convergence to solutions in final population, spread of solutions and average performance in terms of defined criteria [40].

Vega *et al.* inspected multiple different meta-heuristics for solution of radio network design problem in their work [49]. They focused on bio-inspired algorithms such as Population Based Incremental Learning (PBIL), Differential Evolution (DE), Simulated Annealing (SA) and Cross Generational Elitist selection, Heterogeneous recombination and Cataclysmic mutation (CHC). It is stated that best solutions simultaneously came from the CHC and PBIL algorithms.

In their work on Optimal Antenna Placement, Nebro *et al.* surveyed the literature on optimal radio network design with different techniques which utilizes meta-heuristics such as genetic algorithms, simulated annealing and differential evolution [50]. It is stated that it becomes necessary to use faster techniques (heuristics) for solving complex problems that arise with radio network design and the authors explained that the best results have been obtained using a genetic algorithm called Cross Generational Elitist selction, Heterogeneous recombination and Cataclyssmic mutation (CHC). This algorithim starts with a set of solutions, at each iteration two of these solutions are selected as parents and combined with a procedure called HUX, where common information from parents are directly copied to offsprings and half of the disjunctive properties from each parent are transferred to the resultig two offsprings [51]. It is further explained that pairing of too similar solutions are not allowed and when the population convergence is achieved, most of the population except the best ones are modified and the process is restarted. In [50], authors proposed a multi-objective version of CHC the main difference is that at the time of elitist selection more than one criteria is considered and the best members of the population are conserved accordingly to the next run (restart).

3. LORA GATEWAY PLACEMENT PROBLEM

Many Smart City applications are discussed in Section 2.1. Most of the inspected applications require low power and long range communications and are tolerant of higher latency and lower bandwidth. Thus most of these applications can benefit from a well designed LoRa network. To the best of our knowledge, in a generic LoRa network there are no predetermined allocations or special signalling schemes that make network provide better service to specific set of end nodes. The general idea with LoRaWAN is to have an autonomous network with the least signalling overhead, energy consumption and human interaction. Thus, the success of such generic LoRa network is mostly dependent on the location of end nodes and the gateways as the distance between an end node and a LoRa gateway is of major effect to transmission success. In many of the discussed use cases, the end nodes are either already placed or the locations where the end nodes are to be placed are known. An example use case is deployment of LoRa communicating smart meters where the smart meters are installed in prearranged buildings known by the network operator. For a given number of end nodes with known positions, determining the optimal number and location of LoRa gateways are crucial for the success and efficiency of the network. Our aim is to find the optimal number and location of gateways that fulfil requested success criteria, given the location of end nodes. In this section we analyse the LoRa gateway placement optimization problem and the various factors effecting the success of the LoRa network.

- Node Count: n_{node}
- Problem Area Width: w
- Problem Area Height: h
- Node Locations: $C_i = (x_i, y_i)$
- Cost of a single LoRaWAN Base Station: S_{Base}
- Traffic requirements per Node (Packet rate per hour): r_{node}
- CRC Enabled / Disabled: CRC
- Coding Rate: CR

- Total Payload Size (Payload Length): PL
- Low DR optimization (For SF11 and SF12): DE
- Preamble Symbols: $n_{preamble}$
- Transmission Power: $Power_{TX}$
- Bandwidth: BW
- Location of Base Stations: $B_j = (x_j, y_j)$
- Number of Base Stations: n_{base}
- Duty Cycle Limitation: ϑ
- Maximum reporting Delay: t_{delay}
- Uplink Transmission Frequencies, z_i: For each node the frequency to be used for transmission is going to be selected randomly between 8 possible sub bands from 867.1 to 868.5 as given in Table 2.2.
- Distance of each node to base station, d_{ij}:

$$d_{ij} = distance(C_i, B_j) = \sqrt{(x_i - p_j)^2 + (y_i - q_j)^2}$$
$$0 \leq i < n_{node}, \quad 0 \leq j < n_{base}$$

- Receiver Sensitivity, Q: LoRa gateway sensitivity is explained in detail in Section 2.5.3. Depending on the type and chip set of chosen gateway modules, used bandwidth and utilized spreading factor, the sensitivity of LoRa gateways differ. For a single type of LoRa gateway and static bandwidth of BW, the sensitivity levels at each spreading factor is given as follows:

$$Q = \begin{cases} Q_{SF7} & \text{if } SF7 \\ Q_{SF8} & \text{if } SF8 \\ Q_{SF9} & \text{if } SF9 \\ Q_{SF10} & \text{if } SF10 \\ Q_{SF11} & \text{if } SF11 \\ Q_{SF12} & \text{if } SF12 \end{cases} \tag{3.1}$$

- Packet Transmission airtime, t_{air}: Data transmission over the air duration is dependent on BW, CR, PL, $n_{preamble}$, CRC and Header information. Air time can be calculated using the equation from Semtech documentation that is given in Section 2.5.4. Thus, for transmissions with the same packet properties and bandwidth; it is possible to write a function $transmissionTime(SF)$ that is only dependent on the spreading factor.

$$t_{air} = \begin{cases} transmissionTime(SF7) & \text{if } SF7 \\ transmissionTime(SF8) & \text{if } SF8 \\ transmissionTime(SF9) & \text{if } SF9 \\ transmissionTime(SF10) & \text{if } SF10 \\ transmissionTime(SF11) & \text{if } SF11 \\ transmissionTime(SF12) & \text{if } SF12 \end{cases} \tag{3.2}$$

- Minimum time to wait after a transmission, $t_{waitMin}$: Due to duty cycle limitations in the ISM bands, there is a minimum amount of time for a node to wait after making a transmission.

$$t_{waitMin} = \frac{t_{air}}{\vartheta}$$

- Path Loss, L_{pl}: There are many path loss models in the literature. With use of a module that assumes a homogeneous interference environment, the path loss function becomes mainly dependent on the distance between the sender and the receiver. For an uplink transmission from an end node, if packet properties, node and gateway properties and bandwidth are predetermined and static, then the path loss can be explained as a function of distance. Multiple path loss models implemented in our program are further discussed in Section 4.2.1.

3.1. Assumption for LoRa Maximum Successful Range

The power of the received signal at the receiver is:

$$P_{rx} = P_{tx} + Gain_{TXantenna} + Gain_{RXantenna} + Loss_{TX} + Loss_{RX} + Loss_{Misc} – L_{pl}$$

In order for a transmission w to be successfully received, the signal power at the receiving side should be higher than or equal to the sensitivity level :

$$P_{rx} >= Q$$
$$Q <= P_{tx} + Gain_{TXantenna} + Gain_{RXantenna} + Loss_{TX} + Loss_{RX} + Loss_{Misc} – L_{pl}$$

$(Gain_{TXantenna} + Gain_{RXantenna} + Loss_{TX} + Loss_{RX} + Loss_{Misc})$ is all gains and losses combined and simply shown as GL

Since the sensitivity values are defined in the data sheets and change only depending on the transmission spreading factor and bandwidth, the same equation can be rewritten as follows:

For Spreading factor $w.SF$ and Bandwidth $w.BW$, the path loss should be lower than the following:

$$L_{pl} <= P_{tx} + GL - Q$$

Then, for successful reception of transmission w at each spreading factor $= SF$ for predetermined bandwidth BW, the path loss limit is given as follows :

$$PathLoss_{SF} <= P_{tx} + GL - Q_{SF} = MaximumPathLoss_{SF}$$

As path loss can be defined as a function of distance, for all transmissions W with the same static packet properties, bandwidth, frequency, transmission power, base station and node heights, it is possible to define a maximum range calculation function, $maxRange(SF)$ that is dependent on only the transmission spreading factor, SF pro-

vided that the path loss is accepted as deterministic and there is no other loss involved:

$$maxRange(SF) = \begin{cases} d_{MAXSF7} & \text{if } SF7 \\ d_{MAXSF8} & \text{if } SF8 \\ d_{MAXSF9} & \text{if } SF9 \\ d_{MAXSF10} & \text{if } SF10 \\ d_{MAXSF11} & \text{if } SF11 \\ d_{MAXSF12} & \text{if } SF12 \end{cases} \tag{3.3}$$

3.2. Packet Transmission Random Process

For a node N_i with assigned spreading factor SF_i , every hour r_{node} random transmission start times are generated from uniform random function with 1 millisecond intervals. Packet start time generation needs to take minimum wait time into consideration as well.

$$t_{start} = U\left(0, \left(\frac{3600000}{r_{node}}\right) - minWaitTime(SF_i)\right)$$
$$t_{duration} = transmissionDuration(SF_i)$$
$$t_{end} = t_{start} + t_{duration}$$

The center carrier frequency Z_i of transmission is randomly chosen before each transmission between Channel 1 to Channel 8 as U(1,8) the transmission $W_i = (t_{start}, t_{end}, z)$ is added to transmission list of corresponding node N_i.

3.3. Packet Reception

For a transmission,W_i from node,N_i, $(0 \leq i < n_{node})$; with spreading factor, $SF = SF_i$, with start time $t_{start} = t_a$ and end time $t_{end} = t_b$:

For each base station B_j, $(0 \leq j < n_{base})$; if $(d_{ij} < maxRange(SF_i))$ then Base Station B_j is a candidate receiver for transmissions originating from N_i.

$$E_{inrange} = \begin{cases} 1 & \text{if } d_{ij} < maxRange(SF_i)) \\ 0 & \text{else} \end{cases} \tag{3.4}$$

3.4. Packet Collision

There is a possibility of packet collision if packet transmissions from two or more nodes arrive at the same gateway on overlapping times using the same frequency channel and spreading factor. For a transmission W_i from node N_i with frequency z_i, SF SF_i, start time t_a, end time t_b and euclidean distance to base station B_j d_i; if there is another transmission W_k from a node N_k, $(0 \leq k < n_{node})$, $k \neq i$ with spreading factor SF_k, frequency z_k, start time t_m, end time t_n and distance to same base station B_j d_k; that fulfils the below condition, collision occurs.

$$\begin{aligned} (d_{kj}, d_{ij} < maxRange(SF)) \wedge (SF_i = SF_k) \wedge (z_i = z_k) \\ \wedge((t_b > t_m > t_a) \vee (t_b > t_n > t_a)) \end{aligned} \tag{3.5}$$

The transmission overlap of two transmissions W_i and W_k with start times t_a, t_m and end times t_b, t_n respectively can be redefined with mid-points of two transmissions being away from each other at least the sum of both half durations as also suggested in [20], then the collision decision formula becomes:

$$\begin{aligned} E_{collision} = &(d_{kj}, d_{ij} < maxRange(SF)) \wedge (SF_i = SF_k) \wedge (z_i = z_k) \\ &\wedge \left(|\frac{(t_a + t_b)}{2} - \frac{(t_m + t_n)}{2}| < (\frac{(t_b - t_a)}{2} + \frac{(t_n - t_m)}{2}) \right) \end{aligned} \tag{3.6}$$

According to [20], not every transmission overlap is enough to cause a collision. In their work, they have made multiple experiments and determined that there are two more

factors called the capture effect and timing which affects the transmission collision. These factors are explained in more detail in Section 2.5.5. Capture and timing effects are not included in our simulation for simplicity purposes. Thus compared to their design for transmission collision, our simulation implementation may over-calculate some of the collisions.

3.5. Successful Packet Reception

A transmission w from a node with location C_i is successfully received if the transmission is in range of at least one base station where the transmission do not collide with another signal and is not lost due to congestion at the gateway.

$$E_{success} = E_{inrange} \wedge (\neg E_{collision}) \wedge (\neg E_{congestion}) \tag{3.7}$$

3.6. Objective Function

For the success of a network, base station placement is crucial. It is expected that the number of required base stations to be highly dependent on the fitness of base station placement. Simply the goal of the objective function is to minimize the total cost which can be defined as the number of LoRa gateways times the cost of each gateway, S = (n_{base} x s_{base}). However, since the cost of each base station can be accepted as constant; the objective function simply becomes the minimization of number of base stations, n_{base}.

3.7. Optimization Constraints

In the literature, the focus for optimization is for lowering the cost while maximizing the coverage, however for a network with opportunistic, intermittent and short period medium access requirements such as LoRa, simply covering nodes in range of a gateway may not mean a successful network. Transmissions from a node in the range of

a LoRa gateway may still be completely lost. Thus, simple coverage optimization may not be the only concern and we have multiple definitions for a successful LoRa network depending on the application requirements. One of the success criteria that we define is the total ratio of successfully received transmissions to the total number of transmissions as success criteria R_s, $0 \leq R_s \leq 1$. Thus the ratio of successful transmissions should not be lower than R_s.

$$\frac{numberOfSuccessfullTransmissions}{n_{node} \times r_{node} \times problemDurationHrs} \geq R_s \tag{3.8}$$

However, this ratio by itself may still not be enough to truly express the success in the whole network since the total transmission success percentage for each node in the network may drastically change and this ratio would not show it. In order to fix this possible unfairness, we propose the use of success ratio together with a minimum acceptance criterion, which defines the minimum acceptable ratio of nodes, R_n, $0 \leq R_n \leq 1$ that should fulfil the success criteria. Thus, the worst performing $((1 - R_n) \times n_{nodes})$ nodes are excluded from calculations and the average transmission success is recalculated for the remaining nodes. The equation for meeting success criteria is given below. In the equation, φ denotes the problem duration in hours.

$$\frac{\sum_{i=1}^{R_n \times n_{nodes}} \left(\sum_{j=1}^{\varphi \times r_{node}} E_{success} \right)}{R_n \times n_{nodes} \times r_{node} \times \varphi} \geq R_s \tag{3.9}$$

4. LORA SIMULATOR BASED OPTIMIZER DESIGN

4.1. Implementation Architecture

In order to find the lowest possible count of gateways that can successfully accommodate the network for given successful transmission requirements, we implemented an iterative approach that is combined with simulation capabilities. The general design of the system is inspired by the work of Oyman *et al.* [41]. From our problem definition, it is apparent that increasing the number of gateways directly increases the cost while possibly increasing coverage and total successful transmission ratio while lowering the total failed transmission ratio. The amount of increase in coverage and success of transmissions depends on the location of gateways. In order to determine the minimum number of base stations and their locations that fulfil the given requirements, an optimization procedure is run for a given initial number of gateways, the results are compared with the success criteria, if the criteria are fulfilled then results and gateway locations are returned if not then the number of gateways is increased and the process is repeated.

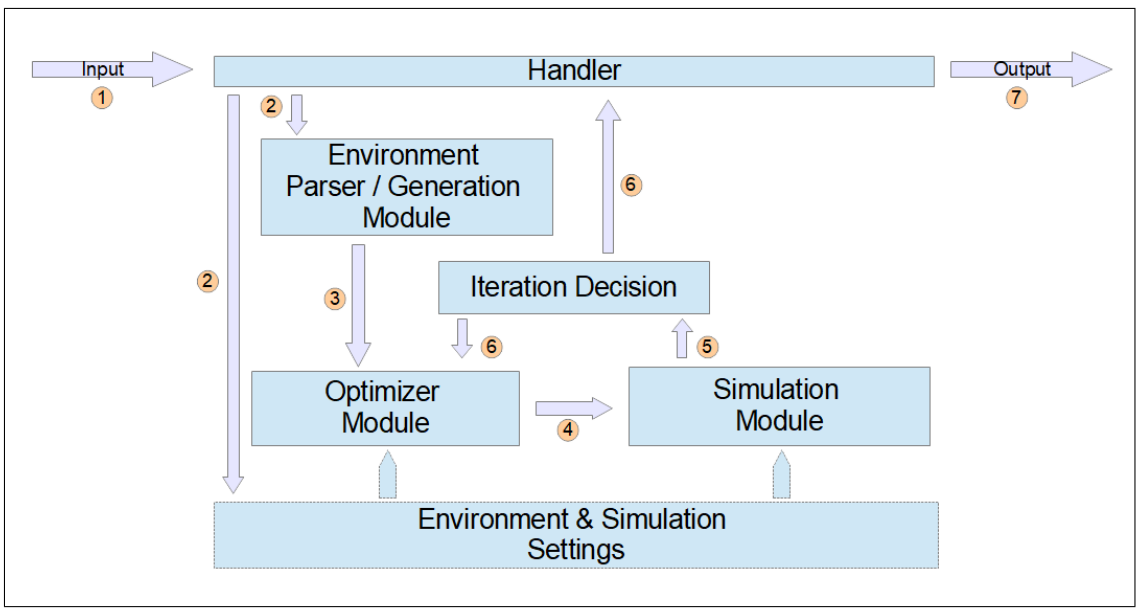

Figure 4.1. Simulation in the loop optimizer architecture.

The simulation in the loop optimization system implementation consists of Environment Generation Module, Simulator Module, Optimizer Module and multiple helper modules. The Environment Generation Module is responsible for generating environments to test according to the given parameters. The Simulator Module is mainly responsible for packet generation, packet transmission, path loss, packet reception simulation of nodes and gateways as well as returning simulation's analytic results. Optimizer Module tries to calculate the optimal placement locations for gateways given the node locations. The helper modules are responsible for writing, reading of data, drawing graphs and charts. These modules draw all node and base station locations and also colours the nodes according to their failed transmission counts. Moreover, they convert returned analytic values into charts. The architecture is visualized in Figure 4.1.

Implemented program can be used both for simulation of a LoRa network and for optimization of LoRa gateway placement in a confined LoRa network. The static initial spreading factor assignment criteria depends on the maximum successful distance calculations given in Equation 3.3. After the initial SF assignment, the program creates randomly generated transmissions from each node depending on the packet rate and simulation duration, calculates collisions and congestions on gateways and determines if a transmission can be received on another gateway without collision or if it is completely lost. After calculations and decision making, the application outputs simple result statistics such as the number of nodes that have a hundred percent transmission success, number of nodes that are unreachable, failure and success rates for different number of nodes. Finally, the program shows collided and disconnected (not in range of any gateway) node information on a graphical interface. The simplified algorithm of the combined optimizer and simulation mechanism is given in Figure 4.2.

4.1.1. Environment Generation

This module takes the simulation width and height, the total count of nodes in the area, resolution (pixel to meter scale) and the number of distribution centre points.

1: Success criteria, chosen optimization method, simulation duration and packet transmission rate and environment options are taken as input
2: Environment Generation Module creates an environment with given parameters or reads a given environment as input
3: Accordingly the created environment object and the given success criteria are passed to the Optimizer Module
4: Starting from a predetermined number of gateways N_G, the best gateway locations are calculated by the Optimizer Module with the given method and passed to the Simulation Module
5: Simulation Module runs a discrete event simulation with the given parameters and outputs statistics back to the Optimizer Module
6: If returned statistics fulfil success criteria, the program ends and returns the gateway locations
7: Else number of gateways N_G is incremented and Optimizer Module is re-run.

Figure 4.2. Implemented LoRa Gateway Placement Algorithm.

The resolution represents the real distance in meters that a pixel in the simulation area corresponds to. The Environment Generation module uses a Gaussian distribution to create an area filled with nodes In order to create uneven and challenging problem environments such as urban cities, the program takes an integer as input to determine the number of center points around which the generated nodes will be placed. Initially, the given number of random points are generated. These are the center points for nodes to be spread around with different Gaussian distributions. The system equally distributes the given number of nodes around the previously created center points using Gaussian distribution with randomly selected mean and variance on both x and y axes. The program can also save the generated environment information to the disk for reuse. The simplified algorithm for the environment generation is given in Figure 4.3.

The idea behind the implemented Environment Generation algorithm is that; in an urban environment, there are points of higher density around which more buildings/people are located in varying shapes such as popular districts and there are points

of less density with less people and/or buildings. By utilizing the random creation of multiple of these density points, using a Gaussian distribution and randomly deciding on their sizes, shapes and variances, we believe that more diverse and realistic urban environments can be generated.

function generateEnvironment$(nodeCount, nc, w, h)$
with 2D problem area width and height as w, h
with number of distribution clusters as nc
$distributionClusters[nc]$ is list of nc distribution clusters
$nodeList$ is an empty list with size $nodeCount$
$numberOfNodesToGeneratePerCenter = (nodeCount \div nc)$
for all clusters in $distributionClusters[nc]$ **do**
  Choose nc uniformly random x_i, y_i value pairs between $0.1(w, h)$ and $0.9(w, h)$
  Choose nc uniformly random $stdevX_i$, $stdevY_i$ values between $0.05(w, h)$ and $0.5(w, h)$
  Create $numberOfNodesToGeneratePerCenter$ (x, y) coordinate values using Gaussian distribution with mean x_i, y_i and standard deviation $stdevX_i$, $stdevY_i$
  (If created random point is not in the area $0, 0$ to w, h the process is repeated)
  insert nodes with generated (x, y) coordinates into $nodeList$
end for
return $(nodeList, w, h)$ as $environment$

Figure 4.3. Environment Generation Algorithm.

Figures 4.4, 4.5, 4.6 and 4.7 present example environments with different node distributions that are generated by the module.

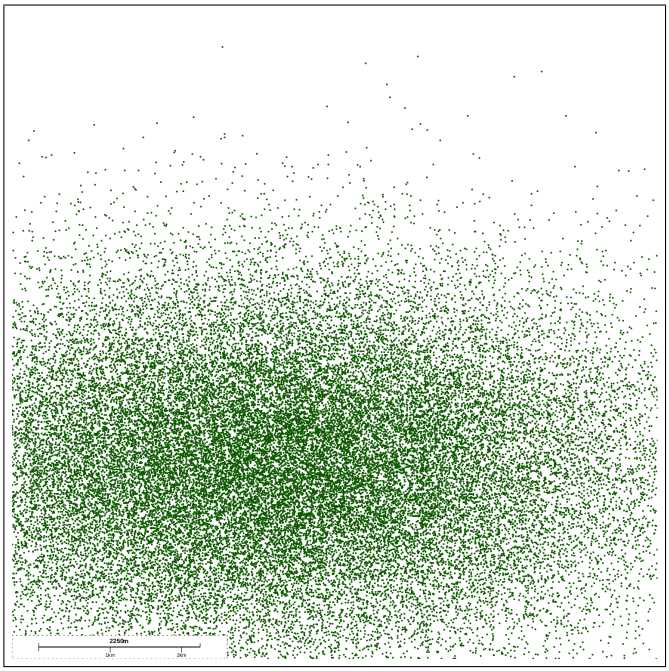

Figure 4.4. An example 9 km to 9 km problem area with 40000 nodes around 1 density point.

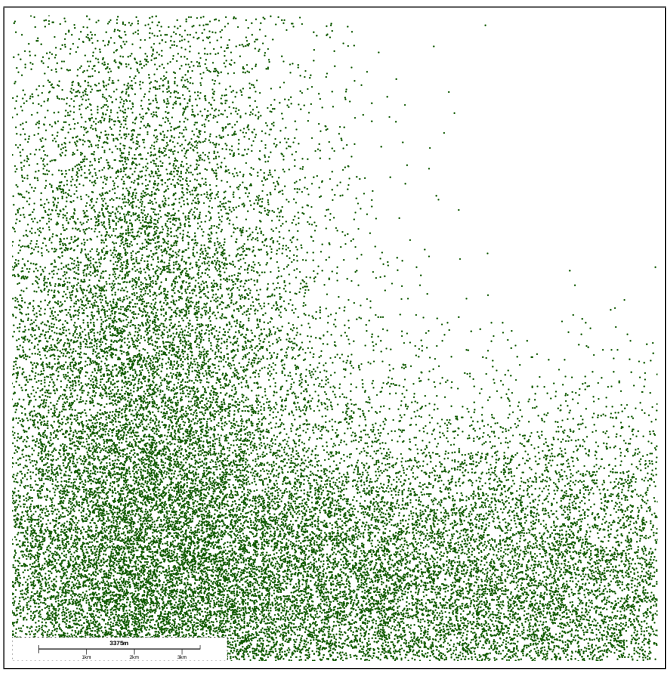

Figure 4.5. An example 13.5 km to 13.5 km problem area with 30000 nodes around 2 density points.

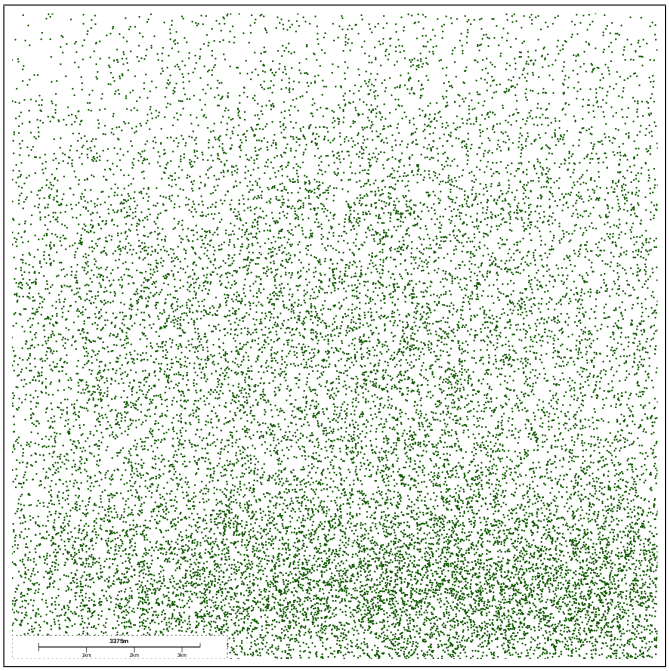

Figure 4.6. An example 13.5 km to 13.5 km problem area with 20000 nodes around 3 density points.

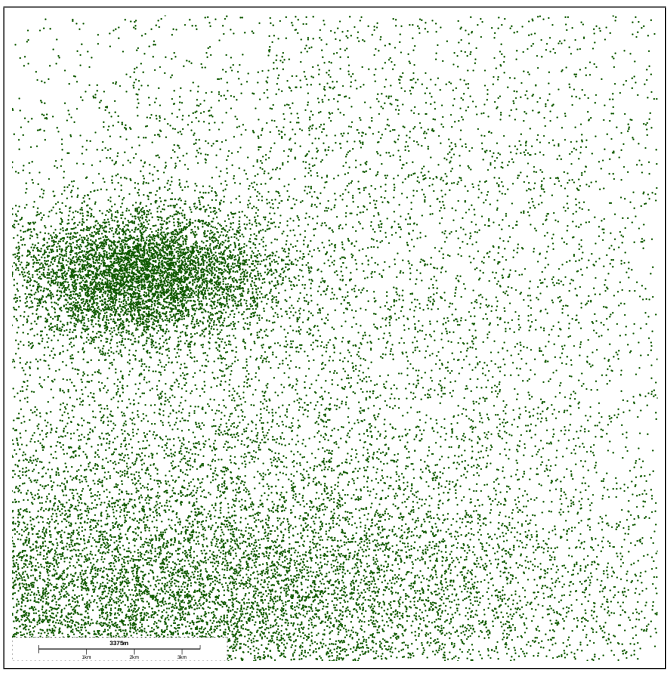

Figure 4.7. An example 13.5 km to 13.5 km problem area with 20000 nodes around 4 density points.

4.1.2. Simulator

Simulation Module is the main module for simulating LoRa node and base station actions in a given duration. This module gets the following arguments: simulation duration, packet generation rate of LoRa nodes, simulation environment, base station locations, LoRa packet options, LoRa Gateway settings and chosen Path loss model. There are two path loss models supported in the simulation module: Okumura Hata model for Large City [52] and Dortmund experiment model [53]. Either of them can be chosen for simulating a given problem.

Interference is checked by comparing power of each transmission occurring at the same time at the place of each gateway. If there is more than one transmission at the same frequency and at the same spreading factor with powers higher than the sensitivity of LoRa base station for the used spreading factor at any time, then both transmissions are marked as collision at that base station. If one of the transmissions can still be received without issues by another base station, then that transmission is accepted as success. For the congestion calculations, depending on used LoRa gateway model, the LoRa base stations are able to serve $n_{concurrent}$ distinct transmissions which does not interfere with each other at the same time. For congestion calculations if a gateway already has been listening $n_{concurrent}$ different transmissions, the following transmissions are marked as congestion and are not processed until a slot is opened for processing (one of the being listened transmissions ends) in the gateway. The simplified algorithm for the simulation given an environment input is in Figure 4.8.

4.1.3. Optimizer

LoRa optimizer module gets the optimization criteria and percentage of nodes that are required to fit these criteria. The design is a “Simulation in the loop Optimization”, meaning each optimized base station setup is sent to the simulation module and results are returned to the optimizer module. It iteratively calculates the optimal base station locations using the chosen algorithm, by increasing the number of base

function simulate($gwList, environment$)
Get number of Gateways and their locations from *gwlist*
Get number of nodes and their locations from environment as *nodelist*
for all node n in *nodeList* **do**
  Calculate distance of each node n to all gw_i in *gwlist*
  Determine if the node n is out of range for all gw_i in *gwlist*
  if node n cannot reach any gateway **then**
    continue
  end if
  if node can reach a gateway **then**
    Determine spreading factor SF_i property of a node depending on its distance to the closest gateway.
    Create transmissions with uniform randomly generated start times for the node depending on the settings: *hourlyPacketRate*, *simulationDuration*, *packetSize*
    for all transmission t in generated transmissions **do**
      uniform randomly assign one of 8 frequency channels to $t.freq$
      calculate transmission duration, $t.duration$
      insert transmission t into the $plannedTransmissionList$
    end for
  end if
end for
for all transmission t in the $plannedTransmissionList$ **do**
  Find other transmissions to_i with same Spreading factor $t.sf$, same bandwidth $t.bw$ and start time between $t.start$ and $t.start + t.duration$
  Add these transmissions to *candidateCollisionList*
  for all transmission tc_i in *candidateCollisionList* **do**
    Calculate the list of gateways transmission tc_i can propagate as $gwlist_{tci}$
    Calculate the list of gateways transmission t can propagate as $gwlist_t$
    if $gwlist_t == gwlist_{tci}$ **then**
      both t and tc_i can only propagate to the same gateways thus both transmissions are recorded as failure due to collision
    end if
    if transmission t can propagate to a gateway gw_j in $gwlist_t$ that is not in $gwlist_{tci}$ **then**
      Transmission t is not recorded as failure
    end if
    if transmission tc_i can propagate to a gateway gw_k in $gwlist_{tci}$ that is not in $gwlist_t$ **then**
      Transmission tc_i is not recorded as failure
    end if
  end for
end for
return all collision, congestion and successful transmission information

Figure 4.8. Simplified LoRa Simulation Algorithm.

stations at each iteration until the optimization criteria are met. After each iteration, the found optimal base station locations are passed to the simulation module which returns certain analytic values according to the simulation it had run. The returned analytic values are then compared against given criteria and the decision to stop or move on to the next iteration is done by the optimizer module.

4.2. Default Values and Preferred Options in Implementation

The implementation used the same frequency channels given in problem the definition which are also shown in Table 2.2. The spreading factor assignments are done at the beginning of the simulation and assumed to be static, not changing. LoRa transmission interference at the gateways are calculated using Equation 3.5. At the beginning of the simulation, all nodes are assumed to be starting their packet generation processes at the same time at t_0 and for the duration of the simulation the end nodes are assumed to be immune to clock drift or any other effect that can change their own reference clocks. The congestion logic is implemented according to the IC880A module and the SX1301 chip that it utilizes. The sensitivity levels and other parameters used in the implementation for gateways and end nodes are shared in Section 3.

4.2.1. Implemented Path Loss Models

The two different urban environment path loss models that are used in our implementation are Okumura Hata and Dortmund Experiment models. Both Okumura Hata and Dortmund Experiment Path Loss models require the base stations to be at least 30m above from the ground and also Okumura Hata Model requires all end nodes to be at least 1 meter above the ground [52]. When packet, node and gateway properties are predetermined and static, both models can be simplified to functions dependent on the distance.

4.2.1.1. Okumura Hata Path Loss Model. Okumura hata has multiple options for urban environments [52], in our implementation we have chosen the Large City version of the model as it is more suitable for a smart city application.

$$\begin{aligned} PathLoss = 69.55 + 26.16\log_{10}(freq) - 13.82\log_{10}(h_{BS}) \\ -C_H + (44.9 - (6.55\log_{10}(h_{BS})))\log_{10}(d) \end{aligned} \tag{4.1}$$

Where;
$C_H = (3.2(\log_{10}(11.75h_{node})^2) - 4.97)$ for large cities
$freq$ is the transmission frequency (MHz),
d is the distance between transmitter and receiver (km),
h_{BS} is the height of the Base station antenna (m) and
h_{node} is the height of the node antenna (m).
With the equation reversed, for a given amount of path loss it is possible to approximately determine the distance of the transmitting node:

$$log_{10}d = PathLoss - \frac{(69.55 + 26.16\log_{10}(freq) - 13.82\log_{10}(h_{BS}) - C_H)}{(44.9 - (6.55\log_{10}(h_{BS})))} \tag{4.2}$$

$$d = 10^{\left(PathLoss - \frac{(69.55+26.16\log_{10}(freq)-13.82\log_{10}(h_{BS})-(3.2(\log_{10}(11.75h_{node})^2)-4.97))}{(44.9-(6.55\log_{10}(h_{BS})))}\right)} \tag{4.3}$$

4.2.1.2. Dortmund Experiment Path Loss Model. The Dortmund model depends on the empirical measurements in the work of Jörke *et al.* [53]. According to their measurements the regression curve in Equation 4.4 is determined for mean Path Loss.

$$PL_{mean} = B + 10nlog10(d) \tag{4.4}$$

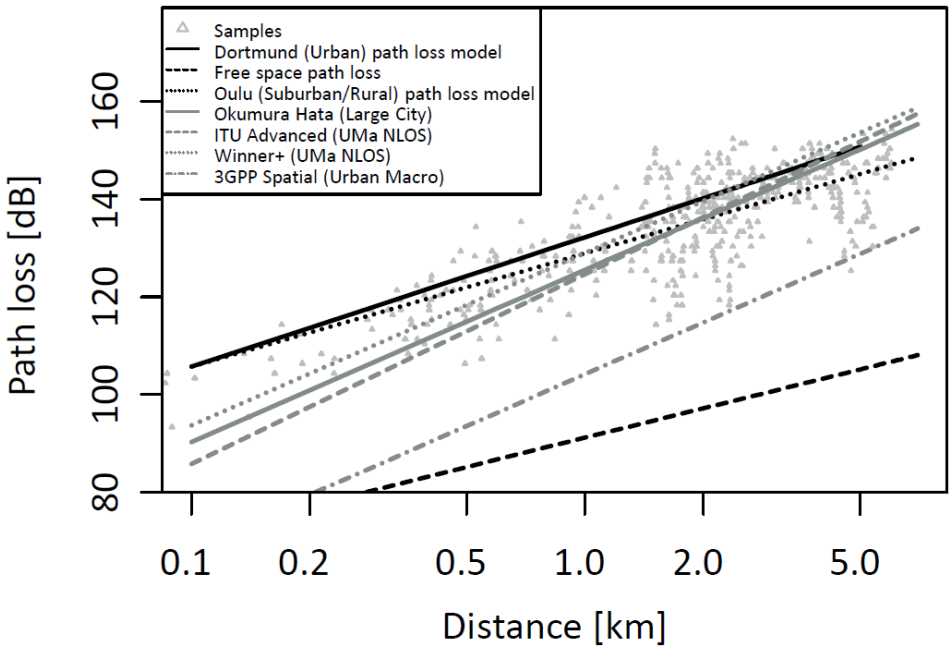

Figure 4.9. Comparison of path loss models for LoRa transmission at 868 MHz, Jörke *et al.* [53].

Again, with the equation reversed, it is possible to approximately determine the distance for a given path loss with Dortmund model.

$$pathLoss = (B + 10n \log_{10}(d)) \tag{4.5}$$

$$d = 10^{\left(\frac{pathLossValue - B}{10n}\right)} \tag{4.6}$$

B is 132.25 dBm Path Loss Intercept for transmission at 868MHz and in a large urban environment(e.g. Dortmund)
n is 2.65 Path Loss Exponent for transmission at 868MHz and in a large urban environment (e.g. Dortmund)

4.3. Heuristic and Success Metric Selection

As seen previously in the literature [48–50], heuristics and meta-heuristics are widely used for solving optimization problems in a reasonable time. Instead of directly

optimizing the whole problem which can get very complex, certain heuristics are optimized. In this section we would like to propose certain metrics and heuristics that can be used for optimization of LoRa gateway placement problem.

Firstly, the distance of each node to the closest gateway directly affects the path loss. It is safe to assume that a LoRa node would be using the lowest possible spreading factor for transmission and as shown in Equation 3.3, distance is the only variable factor which can affect spreading factor decision in our problem. Hence one of the experimented heuristics is the distance of a node to the nearest gateway. Additionally, the prediction of the assigned spreading factor to each node is chosen as another possible source of heuristics for our experiments. The rationale behind the choice of the spreading factor based heuristics is that, when distance is used for evaluation, nodes with different distances to gateways are evaluated differently. Even though the distance differences are a few meters to hundreds of meters, all of these nodes can still utilize the same spreading factor and hence generate transmissions with the same air time. By using spreading factors as an evaluation method, instead of making a decisive choice between two nodes, one of which is only one meter closer to the gateway, both of the nodes may receive the same evaluation score which in turn can help getting better optimization results.

Furthermore, we argue that the time of air cost of a node's transmissions is another very useful heuristic. This is because in our problem, the payload sizes, packet properties and bandwidth are same for all of the nodes and predetermined at the beginning of the problem. The time of air cost of a transmission is dependent on the spreading factor which is assigned to the node depending on its distance to the closest gateway. Moreover, the air time of a transmission directly affects the probability of both collision and congestion, which are the two factors in problem definition that mainly affect transmission success. The mathematical calculations are given in Equation 4.11. Further explanation of the Time of Air Cost heuristic is given in Figure4.17.

For the problem defined in Section 3, for an uplink transmission W_i with time of air t milliseconds, since only transmissions with the same spreading factors cause interference with each other at the gateway as defined in Equation 3.5, any other transmission W_k it can collide would also have a time of air of t milliseconds. If there are N nodes that are in the range of a single gateway, with transmission properties that causes interference with transmission W_i and if the nodes have an hourly packet rate of r then probability of collision for a transmission W_i is explained as follows:
The medium access scheme for LoRa is pure Aloha and the probability of κ arrivals in a duration of T_{frame} is a Poisson process denoted as follows:

$$P(T_{frame}) = \frac{(\lambda T_{frame})^{\kappa} e^{-\lambda T_{frame}}}{\kappa!} \tag{4.7}$$

Where;
T_{frame} is the vulnerable time frame equal to the time of air duration $2 \times t$,
λ is the packet rate.
For a transmission to be successful, there should be $\kappa = 0$ arrivals in T_{frame} to the same gateway.
Then, the equation becomes:

$$P(2t) = \frac{(2G)^0 e^{-2G}}{0!} \tag{4.8}$$

Where G is total number of attempted transmissions per transmission time $G = \lambda t$

It is important to note that as described in Equation 3.5, the number of nodes that are to cause collision must have same the spreading factor and should be using the same frequency channels. As each node is assumed to make a uniform random decision between eight available uplink frequency channels at each transmission, approximately only $\frac{1}{8}$ of the same spreading factor assigned nodes can cause interference and only $\frac{1}{8}$ of the number of nodes should be included in the probability function. r is the hourly packet rate of a node and t is transmission duration in milliseconds. In order to use milliseconds as time unit, the packet rate is divided by 3600000 and G is calculated as

follows:

$$G = \frac{\frac{N}{8} \times r}{3600000} \times t \tag{4.9}$$

As a result, the probability of transmission without any collision for a LoRa network that utilizes Pure Aloha is defined as follows:

$$= e^{-2G} \tag{4.10}$$

$$= e^{-2t \times \frac{N \times r}{8 \times 3600000}} \tag{4.11}$$

With probability P for a successful transmission defined in Equation 4.11, it can be seen that as the transmission duration, t is increased, the failure probability due to collision also increases. Thus, utilization of higher spreading factors increases probability of failure caused by collision.

$$P_{collision} = 1 - e^{-2t \times \frac{N \times r}{8 \times 3600000}} \tag{4.12}$$

In order to show the effect of spreading factors to collision probability, Equation 4.12 is calculated for different spreading factors with different number of end nodes and the results are shared in Table 4.2 and Figure 4.3. Calculations are done using the packet options given in Table 4.1.

Table 4.1. Packet and transmission options used for collision probability examples.

Packet Rate, r_{node}	1 packet/hr
Total Payload Size, PL	32 Bytes (Header Included)
Coding Rate, CR	4/5
CRC	Enabled
Preamble Symbols, $n_{preamble}$	8 symbols
Low DR Optimization (For SF11 and SF12)	Enabled

Table 4.2. Calculated probability of collision for 2000 nodes using denoted spreading factor with single gateway.

Spreading Factor	**Air Time**	**Calculated Probability of Collision**
SF 7	71.936 ms	0.010
SF 8	133.632 ms	0.018
SF 9	246.784 ms	0.034
SF 10	452.608 ms	0.061
SF 11	987.136 ms	0.128
SF 12	1810.432 ms	0.222

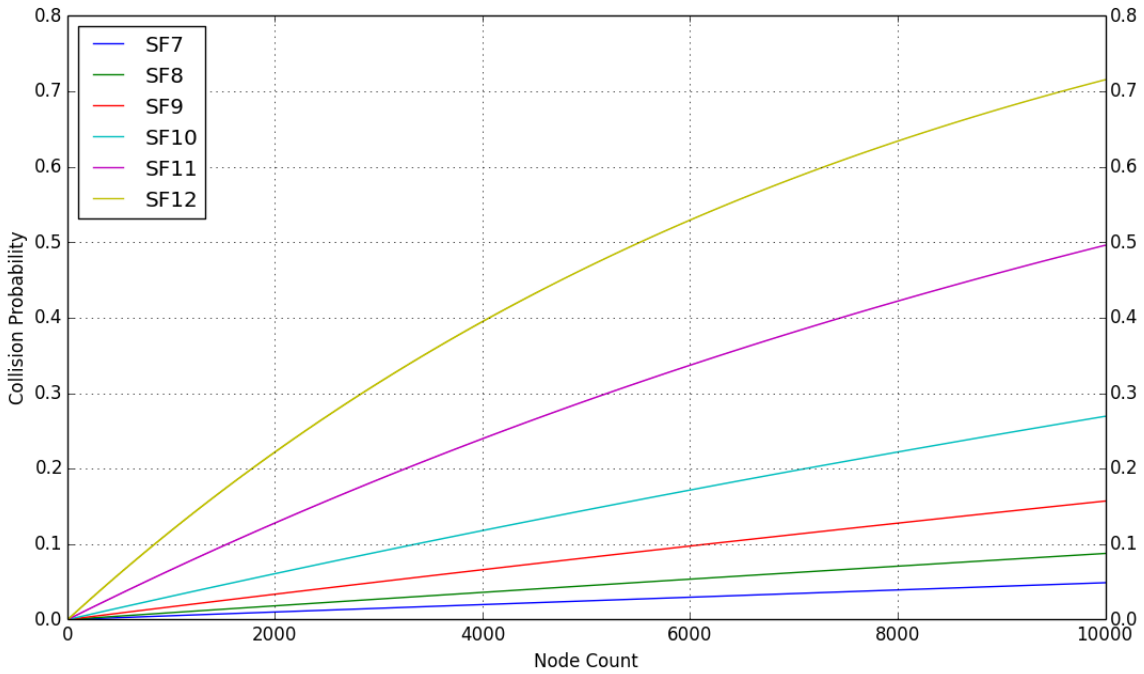

Figure 4.10. Collision probability, node count plot for each spreading factor.

Another proposed method is to make simplified calculation of collision probabilities for each node in the network using the Equation 4.11 to calculate approximate collision probabilities of Nodes. Since the idea is to simplify the calculations, number of nodes in the range, N is calculated by making an approximation and dividing the number of all nodes to the total number of gateways. This method is named naive

probability of collision and is abbreviated as NProb in the rest of the thesis.

Evidently, it can be seen that all of the discussed heuristics up to here, directly or implicitly, only consider the distance to the closest gateway, however there is another effect that plays a significant role in the transmission success which is the existence of another gateway with which the transmissions can also be received. In a pure LoRa network, there can be more than one gateway that is able to successfully receive a transmission and in our base station placement problem there can be nodes that are in the range of multiple gateways for the assigned spreading factor. The increase in the chance of success is calculated in the following mathematical equations.

For a single transmission W_i from a node $node_x$ with a predetermined SF=sf and in the reception range of n gateways; P_{ij} denotes the probability of transmission W_i to have a collision with another transmission on gateway j. If transmission W_i is in the range of two gateways j and k, then the probability of a complete reception failure due to collision can be calculated as $P_{ij} \times P_{ik}$. This calculation gives the correct probability, if the intersection set of nodes that use the same spreading factor in the range of the gateway j, S_j and in the range of gateway k, S_k has only one element $node_x$ and no other element.

Let $P(x)$ be the collision probability function stated in Equation 4.12 and x is the total number of LoRa Nodes. If the intersection set of S_j and S_k has more than one node, the calculations are as follows:

$$N_{intersection} = |S_j \cap S_k| \tag{4.13}$$

$$P(x) = 1 - e^{-2t \cdot \frac{x \cdot r}{8 \cdot 3600000}} \tag{4.14}$$

$$P_{intersection} = P(N_{intersection}) = P(|S_j \cap S_k|) \tag{4.15}$$

$$P_{ij} = P(|S_j|) \tag{4.16}$$

$$P_{ik} = P(|S_k|) \tag{4.17}$$

$$P = (P(|S_j|) * P(|S_k|)) + P_{intersection} \tag{4.18}$$

$$P = (P(|S_j \setminus (|S_j \cap S_k|)|) \cdot P(|S_k \setminus (|S_j \cap S_k|)|)) + P(|S_j \cap S_k|) \tag{4.19}$$

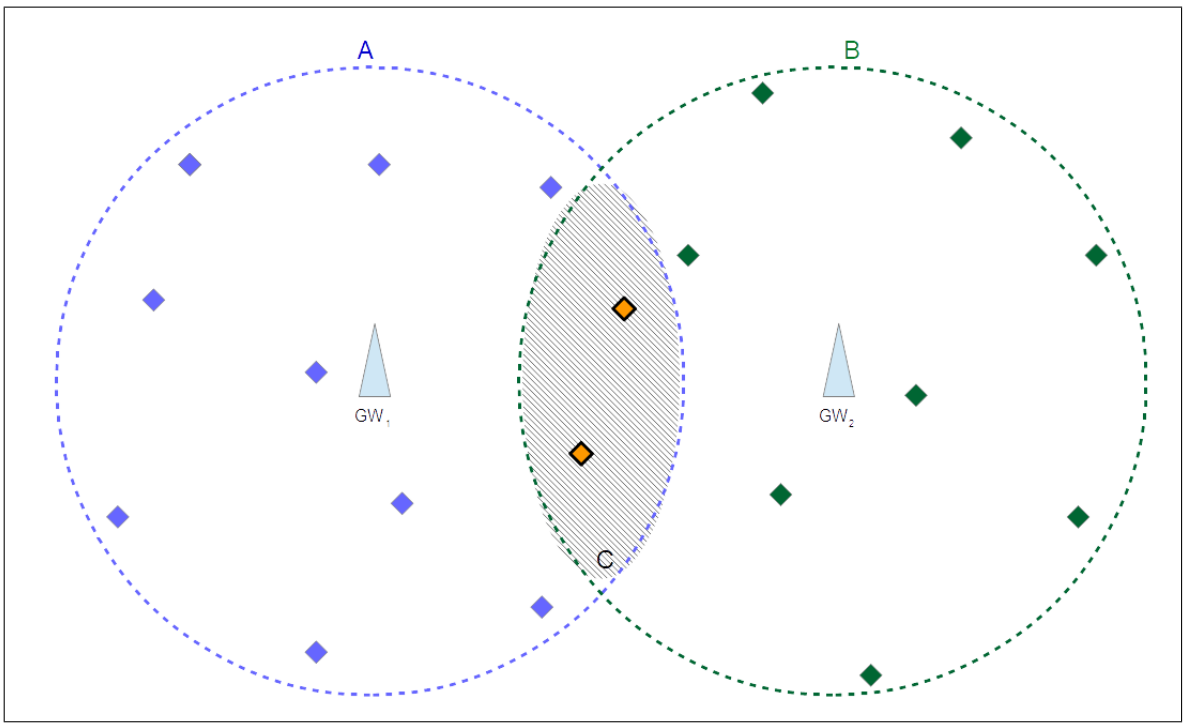

Figure 4.11. Visualization of two gateway transmission collision scenario.

The logic behind Equation 4.19 is visualized for the reader in Figure 4.11. In this figure, GW_1 covers the same spreading factor nodes in A and GW_2 covers the same spreading factor nodes in B. The nodes in area C are in the range of both GW_1 and GW_2. Transmissions from nodes in the shaded region C does not fail as long as it can

be received by one of the Gateways. Thus, the probability of collision is calculated as collision caused by a transmission from a node that is in A but not C at the same time with a collision caused by a transmission from a node that is in B but not C, or a collision caused by a transmission from a node that is in area C.

Generalizing the probability equation for "transmission in range of two gateways" to "transmission in the range of n gateways":

$$S_{intersection} = \bigcap_{j=1}^{n} S_j \tag{4.20}$$

$$N_{intersection} = |S_{intersection}| \tag{4.21}$$

$$P_{loss} = \left(\prod_{j=1}^{n} P\left(|S_j \setminus S_{intersection}| \right) \right) + P(N_{intersection}) \tag{4.22}$$

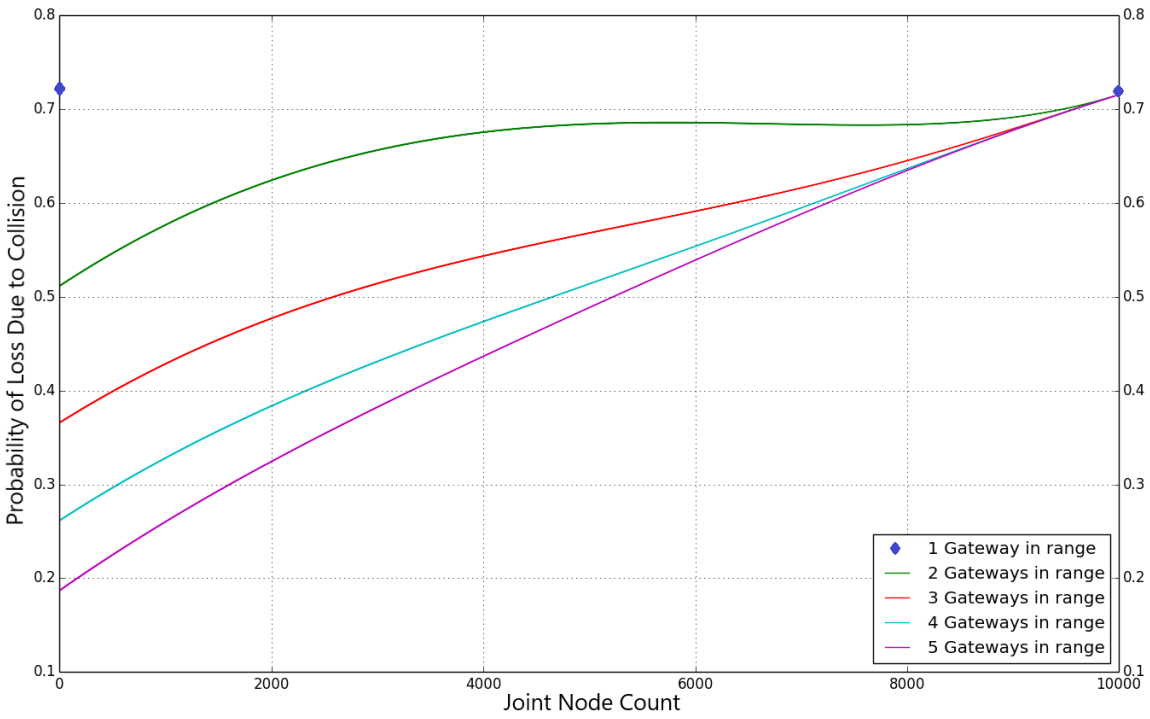

Figure 4.12. Effect of joint node count to probability of complete packet loss due to collision, each gateway has only 10K nodes with SF12 in range.

In Figure 4.12, the change of collision probability with respect to the number of intersecting nodes is shown. The graph is drawn using Equation 4.22. In the graph, each gateway is assumed to have 10000 Nodes using Spreading Factor 12 and the packet options are as given in Table 4.1. The effect of multiple gateways in range is visualized in Figure 4.13 with three gateways. For the simulation of a total of 15129 uniformly placed nodes that only utilize SF12 in an area sized 13.5 km to 13.5 km, the shared heat map shows the success percentage for nodes denoted with color codes. The nodes that reach two gateways show higher success ratio (yellowish green and green) than the nodes in the range of single gateways (mixed orange red and yellow). Furthermore, the nodes that reach all three gateways also have higher success ratio (denoted with greener colors).

In the light of explained collision and packet loss effects, we propose a final metric for the determination of a LoRa network's success as the transmission failure probability score which is calculated for each node in the network using Equation 4.22. Lower the total calculated failure probability score, more successful is the network. In our implementation, we assign a score value bigger than 1 to nodes that are out of range of all gateways in order for the algorithms to prefer covering more disconnected (not in range of any gateway) nodes instead of making small improvements with the connected nodes.

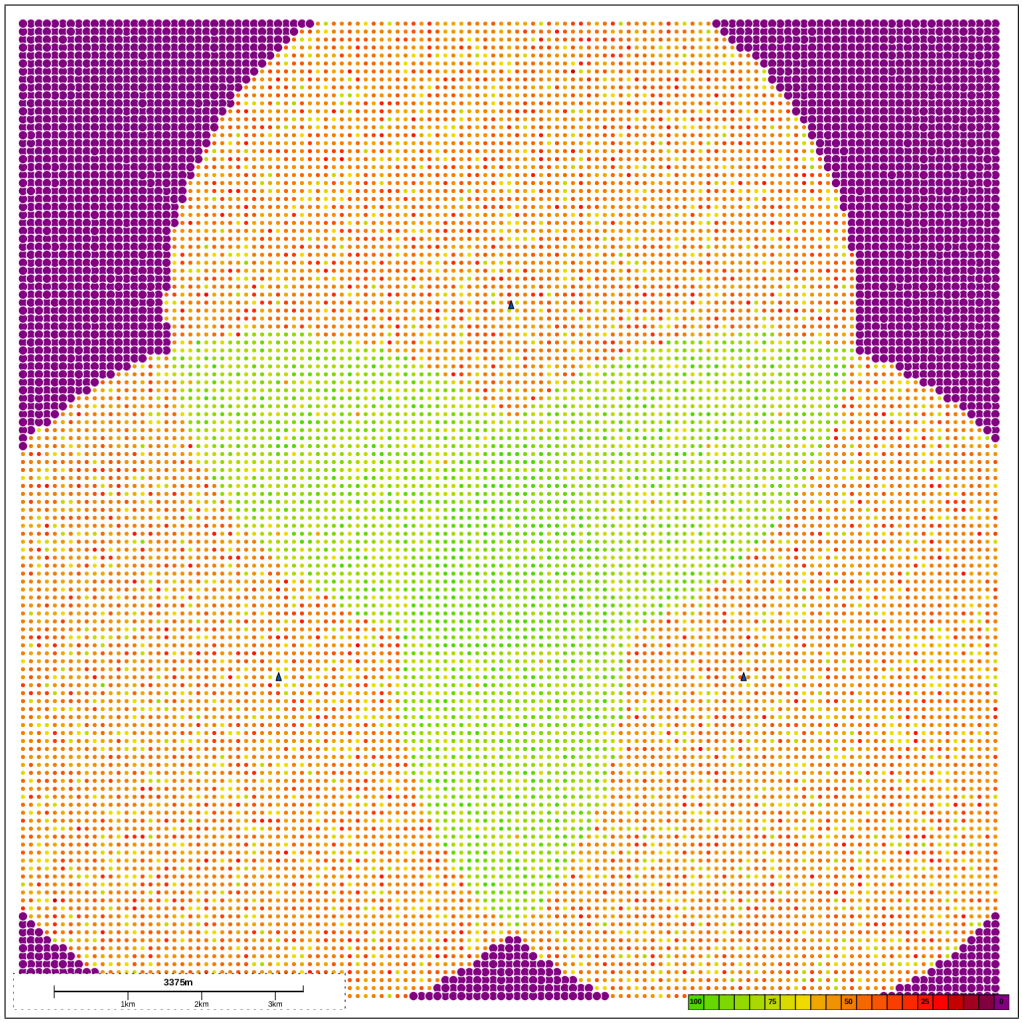

Figure 4.13. Effect of multiple gateways shown via simulation of 15129 nodes which use only SF12 around 3 gateways, duration of 1 day with packet rate of 1 packet/hour.

In a problem scenario where only a single SF is being used, if each gateway has approximately the same number of nodes, a variable P can denote the probability of a transmission to collide with another transmission on any of the gateways. Furthermore, since a single successful reception of a transmission is enough for it to be deemed successful, the probability of successful transmission for any node with n gateways in range is calculated as: $1 - (P^n + P_{intersection})$. As P is smaller than 1, as number of gateways in range of a transmission n increases, it can be trivially seen from both the graph in Figure 4.12 that the probability of successful reception increases and

the probability of transmission loss due to collision decreases. Thus, we state that using a metric that takes the effect of multiple gateways into consideration can provide better evaluation of candidate gateway locations with a probable trade-off of increased computational complexity. The discussed metrics: comparing number of spreading factors, use of transmission airtime costs for each node, use of approximate and accurate collision probabilities are used in the design and implementation of proposed solutions. Most of the proposed algorithms require multiple iterations in search of the optimum solution by design and simulation of the whole scenario at each iteration would simply take too long time. Thus, the usage of defined success metrics will be described in Section 4.4.

4.4. Proposed Optimization Algorithms

We propose multiple different algorithms for LoRa gateway placement optimization and share their details in this Section. The evaluation and detailed comparison of the proposed algorithms are given in Chapter 5. The simulation in the loop optimizer design used in our implementation is explained in Section 4.1 and visualized in Figure 4.1. Our proposed solution can iteratively search for the minimum number of gateways necessary that fulfil a given success criteria, at each iteration, the optimization algorithm is run and the result is used in the simulation. The proposed algorithms in this chapter are for use in the optimizer module of our system.

One of the initial considerations was a naive local search approach which consists of dividing the network into C grids and exhaustively trying all combinations for the optimal placement of k base stations. However this would require $\binom{C}{k}$ trials. It can be trivially seen that this approach requires extensive number of trials and execution may take very long time even for low number of grids.

Another similar solution, which is inspired from the GSP [44] algorithm, is as follows. The problem area is divided into equally sized grids that satisfy the following conditions: the grid diagonal size should not exceed the probable successful reception

range of a transmission with SF7 and the number of grids should not be less than the number of gateways. After the initial division into grids, the centroid of each grid is designated as a possible gateway location. For all combinations of k gateways choosing n possible gateway locations, a heuristic algorithm is ran and best combination is determined. From the chosen grids each grid is divided into four smaller grids and the same heuristic algorithm is ran for all possible 4^k combinations. The four grid division and decision is repeated until the grid diagonal size becomes smaller than a given limit. The heuristic calculations would approximately run $\binom{n}{k}4^{kj}$ where j is the repeat count until minimum size limit l is reached. Although this algorithm could successfully help identify efficient gateway locations, the algorithm's time complexity makes it less feasible with increasing number of gateways and for bigger problem areas even for heuristic calculations.

For geographical clustering of nodes, k-means is a useful algorithm which mainly uses the distances of nodes from each other to create geographical partitions. For LoRa, the factor that effects a transmission's time over air is the spreading factor which in short mainly depends on the distance of a node to the gateway. Thus, an algorithm which takes geographical locations of nodes into account is considered to be the most suitable for partitioning of LoRa endpoints into groups. Our k-means based approach first partitions the nodes into k groups and then determines the gateway locations as center of gravity of each partition. Since homogeneous placement of nodes inside clusters is not guaranteed, the center of gravity approach which helps gateways to have a closer position to denser areas in a partition where more nodes are present is preferred over the center of enclosing circle approach.

In our basic k-means implementation, the graph is partitioned into k clusters using geographical Euclidean distances. The initial cluster centers are chosen randomly. The psuedo code for an algorithm which returns the center of gravity locations for each partition determined by k-means for the given node locations. The algorithm for finding k-means centers is explained in Figure 4.4.

function findKmeansCenters($nodes, k$)

Require k =number of Groups/Centers

Require nodes = $N_1, N_2, \ldots, N_n$ are all given nodes with different locations x_i, y_i in a 2D plane

Initially **k** random nodes are selected as centers from given nodes

centers = $[C_1, C_2, \ldots, C_k]$

k node groups $clusters[k]$ are determined by assigning each node N to the closest C in $centers[k]$

for all C in $centers$ **do**

C_i is recalculated as center of gravity of $clusters[k]$

end for

while $repeatcount < k \times 10$ **do**

for all N in $nodes$ **do**

N_i is assigned to the $cluster$ with center C_i in $centers[k]$ where Euclidean(N,C_i) is smaller than all other Euclidean(N,C_j)

end for

for all C_{new} in $newCenters[k]$ **do**

C_{new} is recalculated as center of gravity of each $cluster$ in $clusters[k]$

end for

if $newCenters[k]$ equals $centers[k]$ **then**

return $centers[k]$

else

$centers[k] = newCenters[k]$

end if

end while

return $centers[k]$

Figure 4.14. K-Means Center Finding Algorithm.

The result of the k-means algorithm can depend on the initial random locations of the gateways and the algorithm's results can be local optima thus, in order to increase the chance of finding better local optima or the optimum solution for LoRa gateways we proposed using k-means clustering with multiple iterations with different random start locations at each iteration. Multi iteration, distance based k-means calculation design for finding better LoRa gateway, locations is explained with pseudo code given in Figure 4.4. For our basic k-means algorithm, at each iteration the sum of node to closest gateway distances are compared and gateway locations with the lowest sum is returned at the end of iterations.

function findBestGWlocationsKmeansDistanceBased$(nodes, k)$
Require k =number of Groups/Centers
Require nodes $= N_1, N_2, \ldots, N_n$ are all given nodes with different locations x_i, y_i in a 2D plane
$centers[k]$ = findKmeansCenters$(nodes, k)$
for all n in $nodes$ **do**
  $sum+=$ (calculate sum of node distances to their cluster center in $centers[k]$)
end for
while $repeatcount < X$ **do**
  $newCenters[k]$ = findKmeansCenters$(nodes, k)$
  for all n in $nodes$ **do**
    $newSum+=$ (calculate sum of node distances to their cluster center in $newCenters[k]$)
  end for
  if $newSum < sum$ **then**
    $centers[k] = newCenters[k]$
    $sum = newSum$
  end if
end while
return $centers[k]$

Figure 4.15. Algorithm for finding best gateway locations via distance based K-means.

In addition to distance based k-means calculations, we propose using other heuristics defined in Section 4.3 together with the k-means algorithm. One of the proposed heuristics was prediction of node counts utilizing each spreading factor depending on gateway locations. The algorithm for using Spreading Factor properties of nodes together with k-means is given in Figure 4.4. In this algorithm, instead of using regular distances, we chose to use spreading factor values that are determined depending on the distance and path loss function. During the cluster assignments in k-means, the decision of cluster assignment for the same spreading factor nodes are done randomly i.e. if there is a node that can use the same spreading factor to reach multiple k-means centers then the node is assigned to a cluster uniform randomly. Since we accept that same spreading factor nodes have equal success percentage, this way the results of each iteration becomes more diverse and chance of escaping local optima and finding better results may be increased. As explained earlier, since the result of k-means algorithms may depend on the initial random center locations, multiple runs of this algorithm is also necessary for reaching a better local optima or the optimum result. The best selection depending on the total spreading factor counts algorithm is not included in the pseudo codes. Briefly, it works as follows; at each iteration the result with the highest SF7 node count is chosen, in case of equality the result with the highest SF8 node count is chosen an so on. Another alternative for the best result selection between multiple runs is to compare approximate time of air costs for each node. Time of air costs are calculated by the algorithm in Figure 4.17 and the selection function between multiple results using air time cost is also given in the same figure. In the approximate time cost calculation function, the SF7 nodes has half the cost of SF8 nodes and SF8 nodes has half the cost of SF9 nodes and so on. The disconnected nodes are assumed to have twice the cost of SF12 nodes in order for the algorithm to prefer connected nodes during calculation.

For use as a baseline we define a simple spatial tiling algorithm. This algorithm only considers the problem area and does not take location or distribution of nodes into account. For any rectangular space this algorithm divides the space into given number of gateway η tiles and determines the centre of each tile as a gateway location.

The tiles do not have to be all equal sized or do not have to cover equal areas. The implemented algorithm works as follows: For a given number of gateways η, its factors x and y ($\eta = x \cdot y$) are calculated for which $\frac{min(x,y)}{max(x,y)}$ is closest to 1. The problem area is divided into $max(x, y) \times min(x, y)$ tiles. If the problem area is not an exact square, the longer edge of the rectangle is divided into $max(x, y)$ parts and the shorter edge is divided into $min(x, y)$ parts. For prime numbers, since x or y would be 1 and the divided tiles may become thinner or longer with increasing number of gateways η. In order to fix the issue, another naive feature is added such that both $(x1, y1)$ for $\eta - 1 = x1 \cdot y1$ and $(x2, y2)$ for $\eta + 1 = x2 \cdot y2$ are calculated, then the x, y value for which $\frac{min(x,y)}{max(x,y)}$ is closest to 1 is selected again. However, this time one tile is added to or subtracted (depending on which x, y pair is selected) from a single row or column (depending on which side of the rectangular area is longer).

In order to use as another baseline and show the reader average success of randomly selected gateway locations for a given problem, a method that makes random selection for possible gateway locations is created. The random selection algorithm takes the number of gateways η, as input and selects η different random locations. Then it evaluates the resulting network by calculating the sum of time of air (TOA) costs for all nodes. The approximate TOA cost calculation algorithm is shown in Figure 4.17. The random η gateway location selection and evaluation process is repeated n_{repeat} times. The resulting TOA costs for each sample are sorted and finally, the sample whose calculated cost is the median of all n_{repeat} samples is returned. Since this is a complete random selection process, selection of minimum or maximum could potentially give excessively good or bad gateway locations. Furthermore, the mean cost of all samples could be affected by outliers such as low number of very good or very bad results. Thus, as a baseline method, we believe that median of all random samples could better show the approximate success of random selection method.

K-means based proposed algorithms may consider the best for each cluster in itself however they do not consider nodes that are adjacent to other clusters and thus, do not consider the other base stations that a node may be in contact, discarding the fact that

function findKmeansCentersSFGuided($nodes, k$)
Require k = number of Groups
Require $nodes = N_1, N_2, \ldots, N_n$ are all given nodes with different locations x_i, y_i in a 2D plane
Require determineSpreadingFactor(distance): A simple function that returns probable spreading factor for a node given the distance to closest gateway
Initially k random nodes are selected as centers from given nodes
$centers = [C_1, C_2, \ldots, C_k]$
k node groups $clusters[k]$ are determined by assigning each node N to the closest C in $centers[k]$
for all C in $centers$ **do**
  C_i is recalculated as center of gravity of $clusters[i]$
end for
while $repeatcount < k \times 10$ **do**
  for all N in $nodes$ **do**
    $mindist_i = min$(Euclidian(N,C_i in $centers[k]$)) is calculated for N_i
    $SF_i = determineSpreadingFactor(mindist_i)$
    N_i is assigned to the $cluster$ with center C_i in $centers[k]$ where SF_i is smaller than all other SF_j
  end for
  for all C_{new} in $newCenters[k]$ **do**
    C_{new} is recalculated as center of gravity of each $cluster$ in $clusters[k]$
  end for
  if $newCenters[k]$ equals $centers[k]$ **then**
    return $centers[k]$
  else
    $centers[k] = newCenters[k]$
  end if
end while
return $centers[k]$

Figure 4.16. Spreading Factor Guided K-Means Center Finding Algorithm.

function calculateApproxLoraToaIndicator($SFcountMap$)

$toa = 0$

for SF from 7 to 12 **do**

  $toa+ = SFcountMap[sf] \times 2^{(SF-6)}$

end for

$toa+ = SFcountMap$[Out-Of-Range] $\times 2^7$

function findBestGwLocationsKmeansTOABased($nodes, k$)

Require k =number of Groups/Centers

Require nodes = $N_1, N_2, \ldots, N_n$ are all given nodes with different locations x_i, y_i in a 2D plane

$centers[k]$ = findKmeansCentersSFGuided($nodes, k$)

$sfCountDict$ = (calculate the number of nodes that are predicted to use each spreading factor depending on cluster center locations in $centers[k]$)

$sumTOA = calculateApproxLoraToa(sfCountDict)$

while $repeatcount < X$ **do**

  $newCenters[k]$ = findKmeansCentersSFGuided($nodes, k$)

  $newSfCountDict$ = (calculate the number of nodes that are predicted to use each spreading factor depending on cluster center locations in $centers[k]$)

  $newSumTOA = calculateApproxLoraToa(newSfCountDict)$

  if $newSumTOA < sumTOA$ **then**

    $centers[k] = newCenters[k]$

    $sumTOA = newSumTOA$

  end if

end while

return $centers[k]$

Figure 4.17. Algorithm for finding best gateway locations via time of air (TOA) based K-Means.

nodes with multiple gateways in the range have lower collision probabilities for their transmissions. A LoRa node that can reach multiple gateways with the same spreading factor is luckier and has higher probability of making a successful transmission since they have a lower probability of being victims of collision and congestion.

The detailed mathematical explanation for probability of collision with multiple gateways was given in Section 4.3 with Equation 4.22 and Figure 4.12. For a node in the range of multiple gateways to have its transmission fail on multiple gateways because of congestion can be given as $P_1 \times P_2 \times ...P_n$ where $1, 2, ...n$ denote each gateway and no other node is in the range of all n gateways. For two gateways, (assuming no other node is in the range of both of the gateways) the probability of it not failing on both gateways because of congestion is $1 - P_1 \times P_2$. Thus, the probability of a node not failing a transmission is higher with multiple gateways since $P_n < 1$.

Furthermore, as they are utilized for radio network design problems for cellular networks in the literature multiple times before and since these networks have also star topologies, another method we propose for the optimization is genetic algorithms. Cross generational elitist selection, heterogeneous recombination and cataclysmic mutation (CHC) algorithm by Eshelman [54] is a type of genetic algorithm that is used for solving network optimization and design problems in the literature. The use of genetic algorithms and CHC in particular is discussed in [50, 51]. We have customized the general CHC algorithm and utilized it for solving the problem of designing an optimal LoRa network, achieving promising results that are shared in detail in Chapter 5. The general CHC algorithm works as follows. An initial population is randomly chosen by selecting L random solution sets among all possible solutions. Then iteratively the population is paired while preventing any two similar members of the population from pairing and each pair are processed via half uniform crossover (HUX) algorithm. After all population members are processed, before the next iteration an elitist selection occurs where the original population and the offspring population are evaluated and the best N performers form the next population in the iteration. If almost all members of the population become too similar, then cataclysmic mutation is performed on

the result set with a α probability while keeping the best δ percent of the population exactly the same. The implemented algorithm for CHC is shared with the reader in Figure4.19.

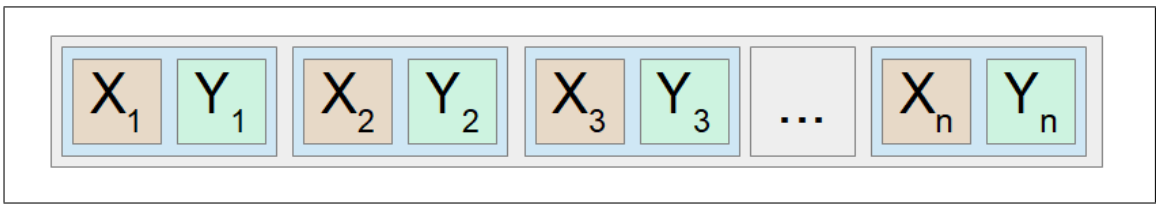

Figure 4.18. Chromosome design of LoRa gateway coordinates.

In order to decrease the size of the solution set and provide efficiency to the genetic algorithm, initially the given problem area is divided into sufficiently small grids. Then x and y coordinates of centroid of the grids are inserted into the two lists of possible solutions, listX and listY. An initial population of $L = 50$ solution sets are generated by randomly selecting x and y coordinate values. $L/2$ times randomly selected pairs from the population are processed and two new children solutions (gateway locations) are created. The created children solutions are inserted into a new list. In order to generate a child solution set from parents a CHC process called Half Uniform Crossover (HUX) is run. If two of the parents are too similar then they are not paired. At each iteration the original and child populations are combined into a temporary set, evaluated and sorted, the best L members of the combined population form the next original population and the rest is discarded. If no child population can be generated with the current similarity threshold, then the threshold is decreased. Whenever the threshold becomes negative, a cataclysmic mutation takes place where each x and y value in every member of the population is randomly changed with a probability of α. Before cataclysmic mutation, top δ of the population is saved and the top δ of the population directly proceed to the next generation untouched. The evaluation function is very important because the natural selection process of the genetic algorithm depends on the evaluation function that is used with the algorithm. We proposed using the Time of Air (TOA), Naive Probability of Collision (NProb) and Probability of Collision (Prob) heuristics as evaluation functions in our CHC implementation and shared the results of CHC algorithm with each heuristic in Chapter 5.

Require: Number of Gateways:**N**, Size of Population:**L**, Elitist Selection Ratio:δ, Mutation Probability:α, Grid length:**l**, Problem Area Width:**w**, Problem Area Height:**h**, Fitness Evaluation Function: **EvaluationFunction**

The Problem area is divided into $\frac{w}{l} \times \frac{h}{l}$ grids

repeat

Initial Threshold $\tau = \alpha \times (1 - \alpha) \times L$.

Centroid of each grid is a possible gateway location

$L \times N$ random gateway locations are inserted into the population list **poplist**

for L/2 times **do**

Randomly Make 2 selections from poplist

if distance between the 2 selections are bigger than **t then**

Calculate Half Uniform Crossover

Add Resulting Children to Cpoplist

end if

end for

if CpopList is empty **then**

Decrement threshold τ by 1

end if

Combine poplist and Cpoplist into one list, cobinedList

Clear popList and CPopList

Calculate toa score on each member of combined List

Sort combinedList according to the **EvaluationFunction** results

Keep first L elements in combinedList and Remove rest of its members

if $\tau < 0$ **then**

Keep top δ percent of combinedList untouched

Mutate axis values from remaining items with probability P

end if

$popList = combinedList$

until Enough Iterations are run (iteration limit is reached)

Return Top element in combinedList

Figure 4.19. CHC algorithm for finding best gateway locations.

The gateway locations are represented as a set of x,y coordinates, thus in a Genetic algorithm these can be treated as a chromosome structure as shown in Figure 4.18. The coordinates can be regarded as genes and a complete set of coordinates can be considered as chromosomes. For genetic algorithms, there are works in the literature which define the chromosome completely binary however, in our work we chose to use genes as x,y coordinates. The process of Half Uniform Crossover between two solution sets is visualized in Figure 4.20. In this Figure, the (x_1, y_1) and (x_4, y_4) coordinates, (A, B) and (I, J) respectively are a complete match and they will be preserved during the crossover. Additionally y_2 and x_3 (D and E) are also matching genes and will be preserved. For Half Uniform Crossover, only exactly half of the possible genes are randomly chosen and switched between chromosomes. Thus, the final pair in Figure 4.20 shows the resulting children from the Half Uniform Crossover process.

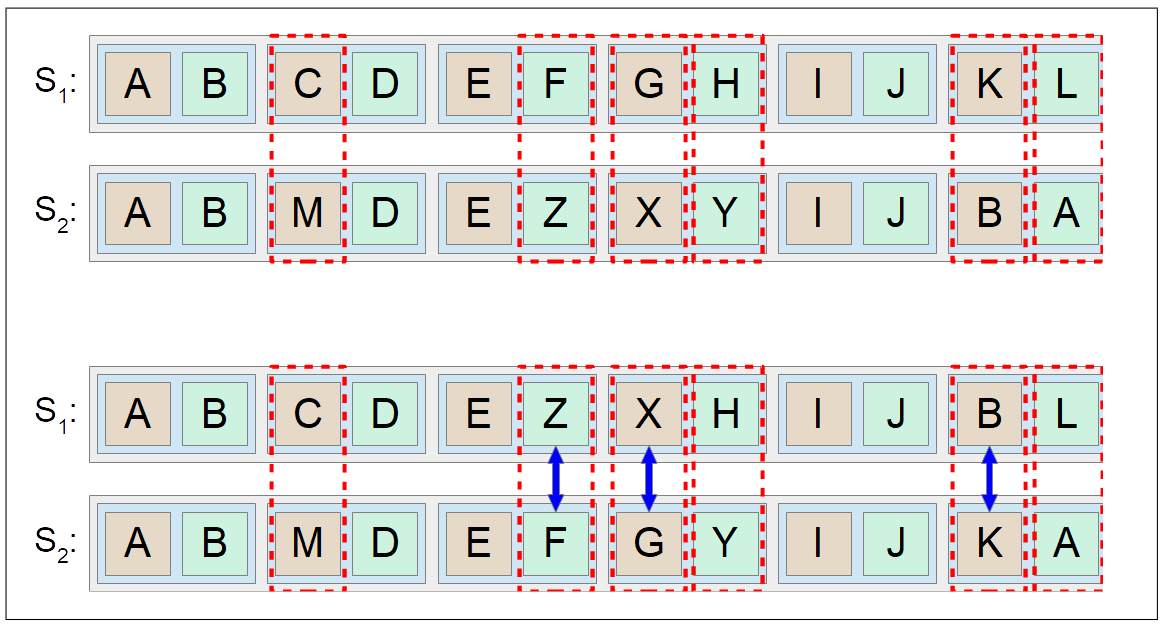

Figure 4.20. CHC Half Uniform Crossover process on an example set of LoRa gateway coordinates.

Table 4.3. Algorithm Implementation Options.

Parameter	**Default Value**
α(CHC)	35%
δ(CHC)	5%
L(CHC)	50
n_{repeat}(Random)	1000
l(CHC)	50m
Iteration Limit(CHC)	50

5. EXPERIMENTS AND RESULTS

In this chapter we share the results of multiple experiments using our simulation in the loop optimization system. First, we show the effects of node density, packet rate and payload length on average transmission success. Later, we share two scenarios, and use optimization algorithms from the previous chapter to determine the optimal LoRa gateway locations for different number of gateways. As a note to the reader; although LoRa communications are not connection oriented, we talk about the nodes that are not in the range of any gateway and cannot have any successful transmission as "disconnected nodes" in this chapter. Chosen simulation parameters and static problem options are given in Table 5.1.

5.1. Effect of Node Count

In order to show the effect of node count, multiple environments with different number of nodes are generated using our environment generation module. By using the same random seeds, the distribution of nodes in the generated environments are kept the same while increasing the number of end nodes. Repeating simulations are run on the generated environments. The effect of the node count in the reception area of a single LoRa gateway are inspected using the simulation results. Simulation results of an example scenario with different node densities are shared with the reader in this section. This scenario simulates a smart city application that utilizes LoRa for communication between the gateway and end nodes. End nodes are required to periodically transmit their application data to a single gateway positioned in the center of the problem area. The simulation results are consistent with earlier transmission failure probability calculations, and it can be seen from the results that the number of nodes that transmit in the range of a LoRa gateway directly affects the collision probability as shown in Equation 4.11 and Figure 4.3. The problem area is 9km to 9km. For the simulation of the scenario, the options and problem parameters in Table 5.1 are used.

Table 5.1. Problem Parameters and Chosen Options.

Parameter	**Default Value**
Simulation Duration	24 Hrs
Packet Rate, r_{node}	1 packet/hr
Total Payload Size, PL	32 Bytes (Header Included)
Coding Rate, CR	4/5
CRC	Enabled
Preamble Symbols, $n_{preamble}$	8 symbols
Low DR Optimization (For SF11 and SF12)	Enabled
Gateway Height (Antenna), h_{BS}	30 m
Node Height (Antenna), h_{node}	1 m
Gateway Model	iC880A (SX1301 based)
Gateway Concurrent Packet Reception, $n_{concurrent}$	8
Node Transmission Power, $Power_{TX}$	14 dBm
Used Frequency Channels, z	TTN Channels given in Table 2.2
Bandwidth, BW	125 kHz only
Duty Cycle, ϑ	1%
Path Loss Model	Dortmund Model [53]

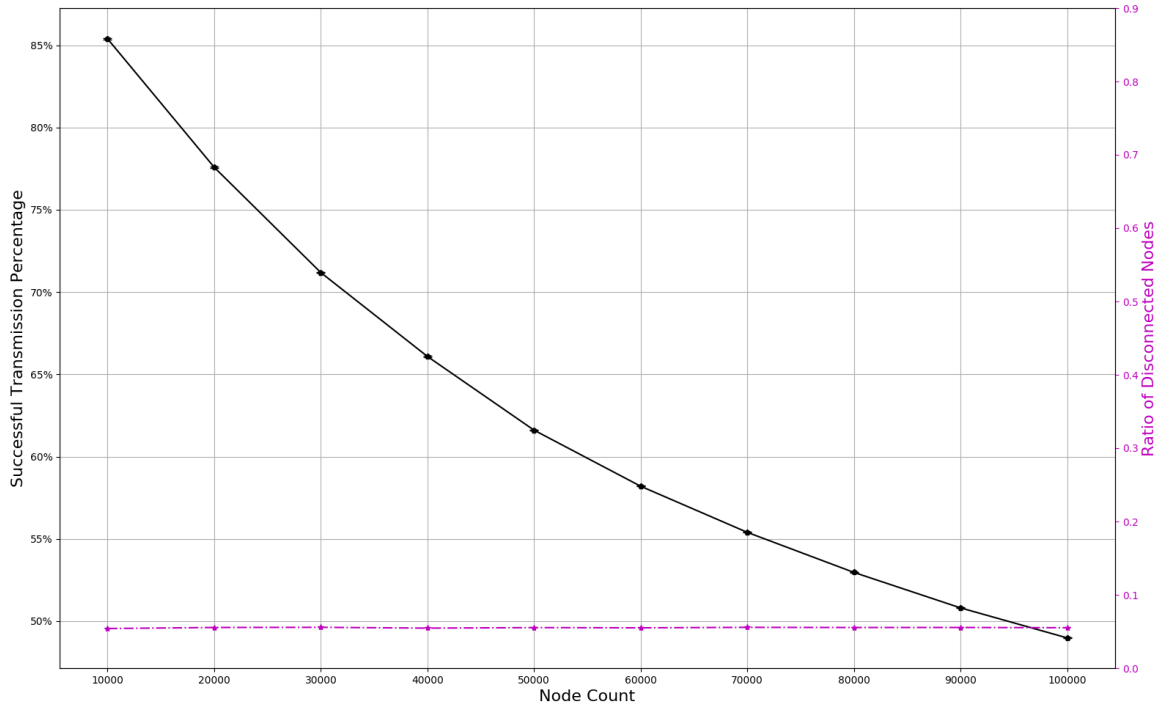

Figure 5.1. Effect of node count on total transmission success (Single Gateway).

Table 5.2. Transmission success ratios for each number of end nodes with 95% confidence interval for Figure 5.1.

Node Count	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
Transmission Success	0.85387 ±0.00044	0.77582 ±0.00038	0.71184 ±0.00019	0.66086 ±0.00029	0.61606 ±0.00017	0.58198 ±0.00023	0.55397 ±0.00017	0.52962 ±0.0002	0.50806 ±0.00017	0.48983 ±0.00016

In Figure 5.1, the total transmission success percentage is drawn for increasing number of nodes from 10000 to 100000 nodes in the given environment. All disconnected nodes are assumed to make completely failed transmissions in the successful transmission percentage calculations given in Figure 5.1. The total failed transmission percentage versus the node count graph is drawn in Figure 5.2, in which only the connected nodes (nodes that are in range of a gateway) are taken into account and disconnected nodes are excluded from the percentage calculation. Additionally, the ratio of disconnected nodes to the total number of nodes is also included in the same figure, denoted with a purple dashed line, showing that the distribution of nodes are kept the same between iterations while the density of end nodes is increased.

We would like to note for the reader that in [35], simulation results for another scenario that evaluated packet loss ratio for varying number of nodes are shared. However, in their simulations the nodes try to transmit packets as soon as possible while obeying the duty cycle limitations whereas in our scenario, we assumed an hourly packet rate depending on the application needs and ran simulations for the duration given in Table 5.1. We believe that with this method it is easier to see the capacity of a LoRa gateway and the effect of the node count for a real world application.

From simulation results, it is apparent that for increasing number of nodes that are being serviced, the success percentage of total transmissions during the course of simulation decreases. Furthermore, in the failed transmission percentage graph, we show that since the environment is generated with the same node distribution and although the number of nodes are increased, the ratio of disconnected number of nodes do not change drastically between each simulation. Thus, it is safe to say that with increasing number of nodes in the range of a single gateway, the failure percentage for transmissions of connected nodes increases.

Table 5.3. Transmission failure values with confidence interval for Figure 5.2.

Node Count	10000	20000	30000	40000	50000	60000	70000	80000	90000	100000
Failed Transmission Ratio	0.09739 ±0.00047	0.17864 ±0.0004	0.2462 ±0.00021	0.30101 ±0.00031	0.3479 ±0.00018	0.38416 ±0.00025	0.4134 ±0.00018	0.4393 ±0.00021	0.46211 ±0.00018	0.4816 ±0.00017

In Figures 5.3 and 5.4, the heatmaps for the corresponding simulation are shown. In Figure 5.3, the simulation is run for 10000 Nodes in a given area. It is possible to see that for SF12 there is approximately 70% success rate, denoted by the yellow colored nodes and for lower spreading factors the success rate is higher, denoted by green and dark green colored nodes. In Figure 5.4 the total node count is 100000 and it is visible that for SF12 and SF11, almost all transmissions have a maximum of 5% success rate. For SF10 the success rate is approximately 30% whereas for SF9 the ratio is above 60% and only for nodes with SF7 and SF8, the transmission success ratio is above 70%.

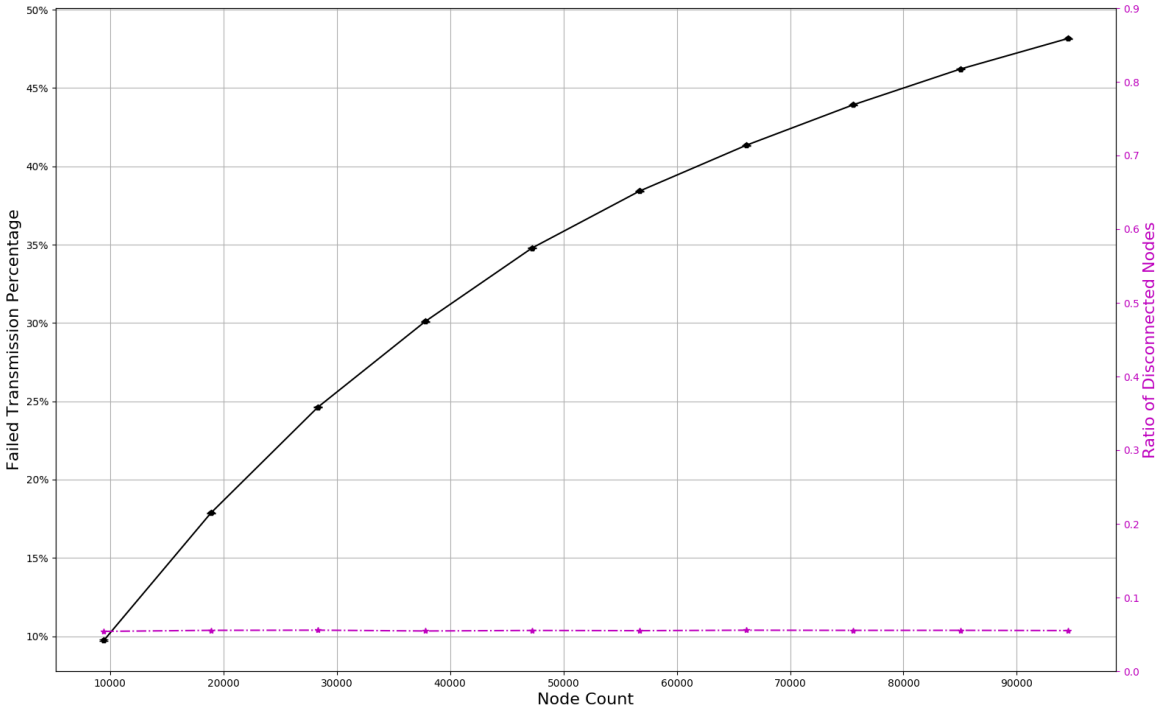

Figure 5.2. Effect of node count on connected node transmission failure, combined with ratio of disconnected nodes to total number of nodes (dashed magenta).

5.2. Effect of Packet Rate

It is explained in Section 2.5 that as the number of transmissions in a time frame increases, the chances of collisions and overload (congestion) at the gateway also increases. The charts in Figures 5.5 and 5.6 visualize these effects. The simulation and packet options are as given in Table 5.1. The simulation environment is the 10000 node 9 km to 9 km environment that is also used for the previous simulation in Section 5.1. Only the hourly packet rate is changed. Multiple simulations were run on the same environment with increasing packet rates from 1 packet per hour to 10 packets per hour. Each simulation instance is repeated 9 times with different random seeds. The disconnected node ratio is approximately 0.06, the same with the simulation in Section 5.1. From Figures 5.5 and 5.6, it can be seen that the average transmission success goes down from 85.4% to 48% and average transmission failure for connected nodes increased from 9.5% to 49% as packet rate goes up to 10 packets per hour.

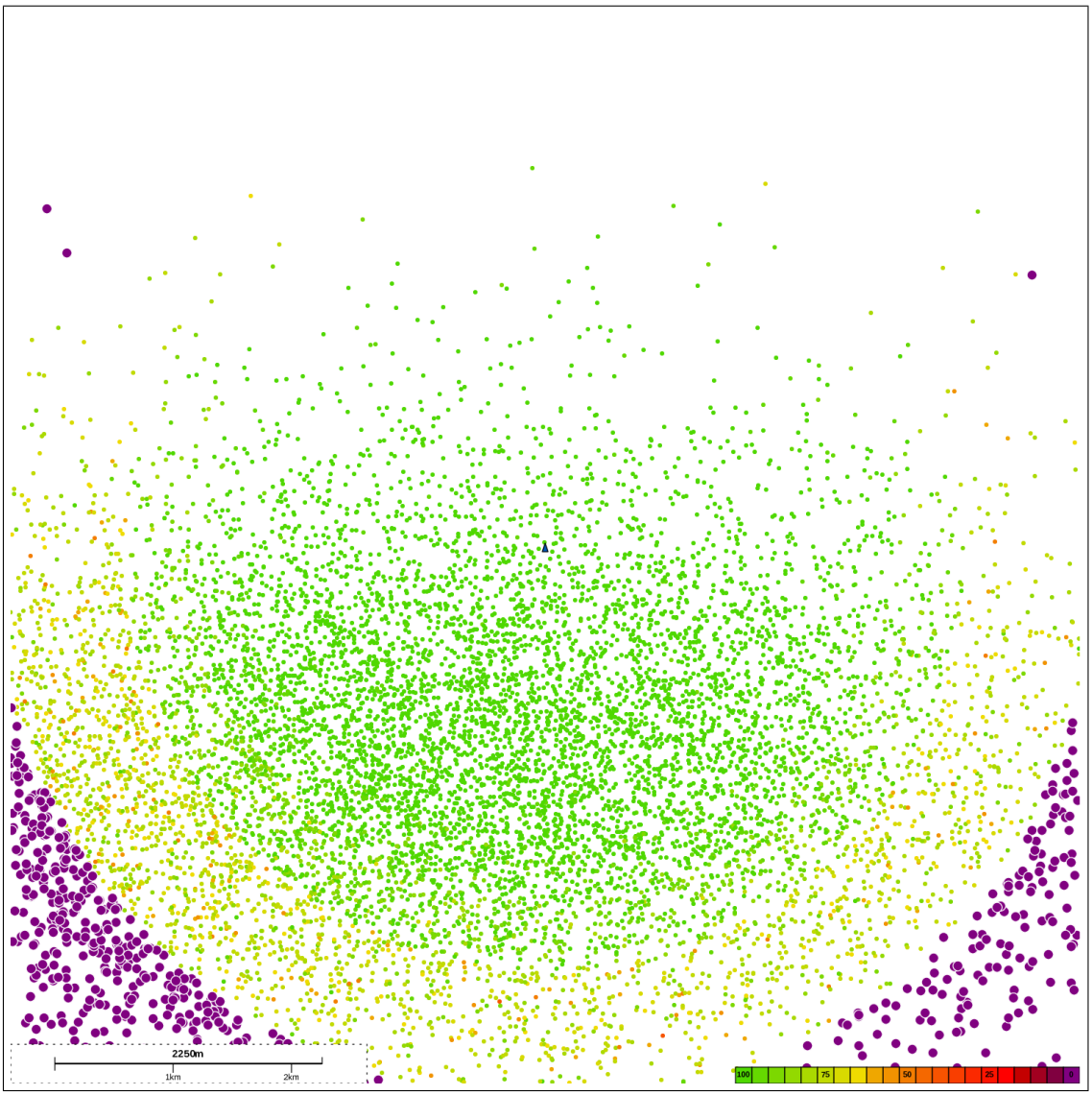

Figure 5.3. Average transmission success of each node for 24 hr, 1 packet/hr simulation of 10000 nodes and one LoRa gateway.

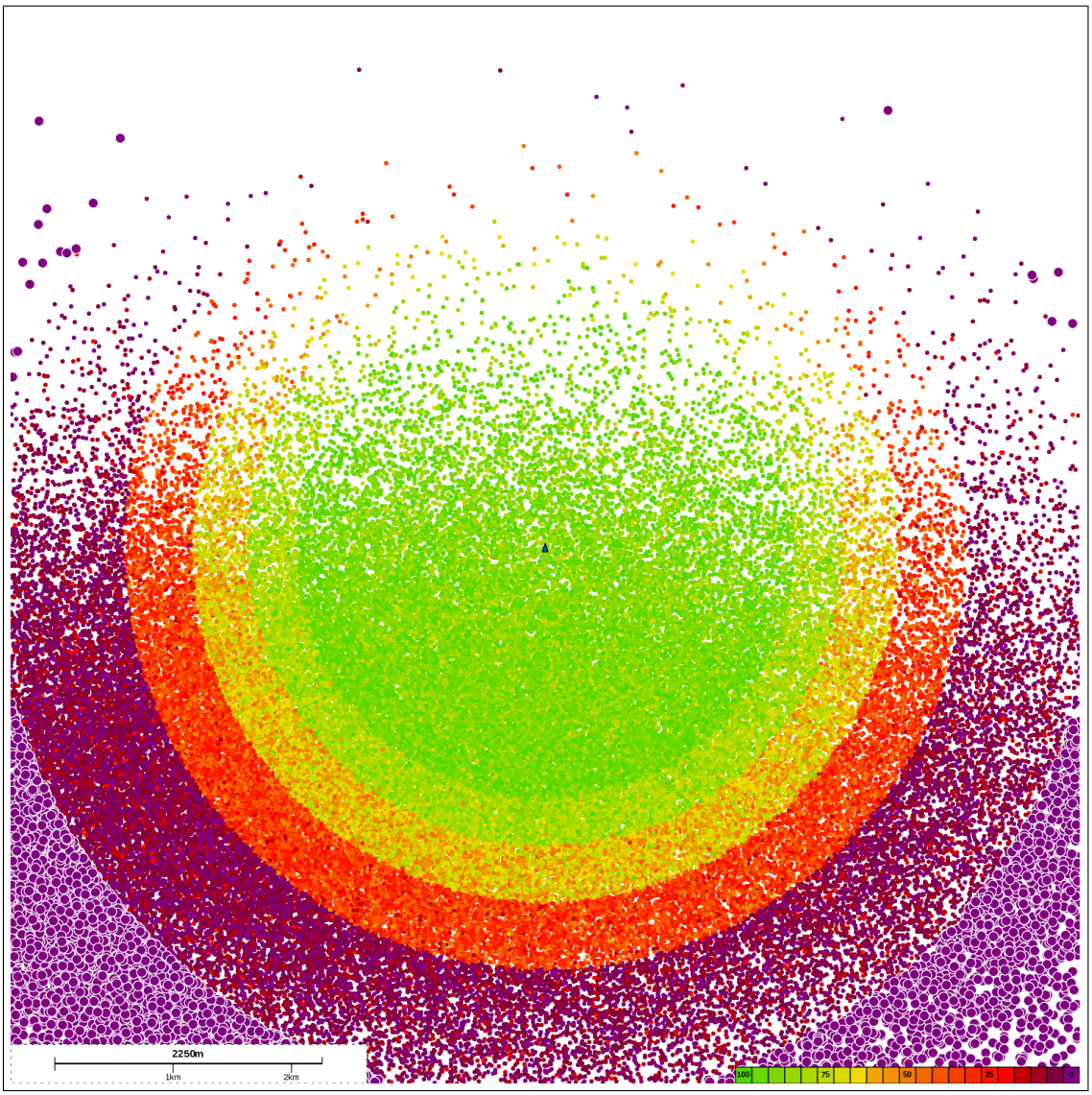

Figure 5.4. Average transmission success of each node for 24 hr, 1 packet/hr simulation of 100000 nodes and one LoRa gateway.

The statistics in Figures 5.5 and 5.6 show similar results to the results of effect of node count simulations. Hence, it can be inferred that for single gateway environments, the effect of doubling the packet rate has similar effects to that of doubling the node count for LoRa networks (if the duty cycle limitations are negligible/ignored) as long as the distribution of nodes do not change with the node count increase.

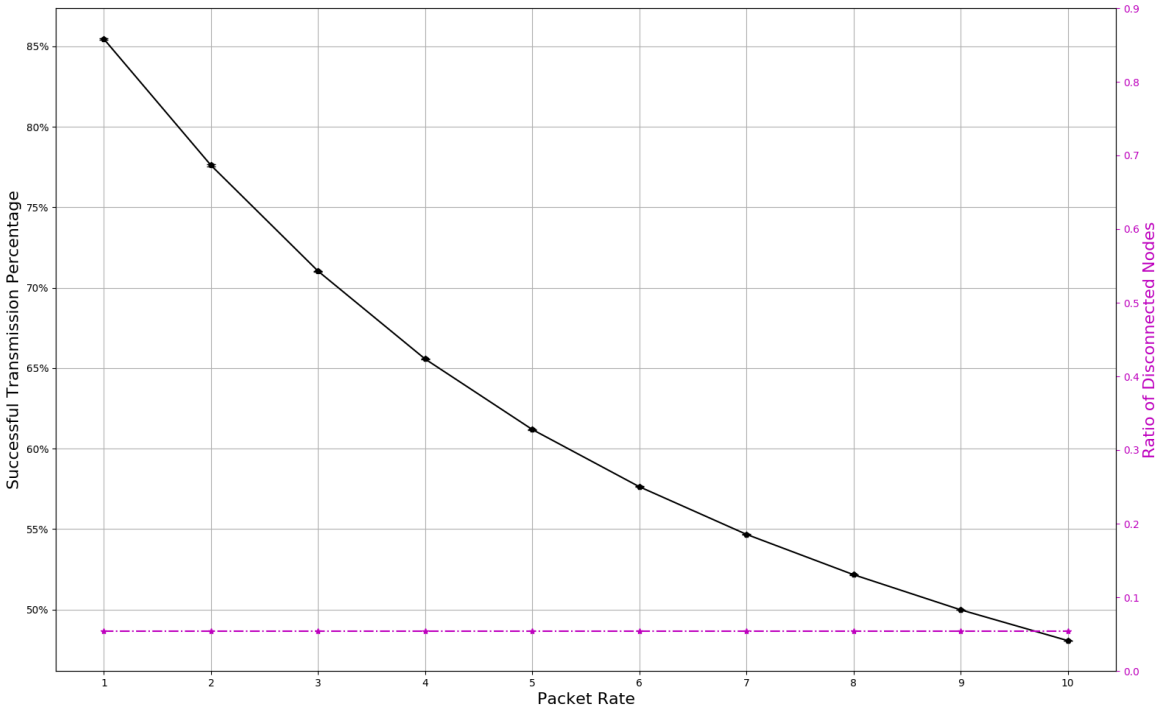

Figure 5.5. Effect of packet rate (packet/hr) on total transmission success.

Table 5.4. Transmission success values with confidence interval for Figure 5.5.

Packet Rate	1	2	3	4	5	6	7	8	9	10
Success	0.85447 ±0.00047	0.776 ±0.00048	0.71028 ±0.00026	0.65567 ±0.00024	0.6119 ±0.00022	0.57634 ±0.00021	0.54676 ±0.0002	0.52171 ±0.00022	0.49988 ±0.00013	0.48075 ±0.00011

Table 5.5. Transmission failure values with confidence interval for Figure 5.6.

Packet Rate	1	2	3	4	5	6	7	8	9	10
Failed Transmission Rate	0.09675 ±0.0005	0.17971 ±0.00051	0.24918 ±0.00027	0.30691 ±0.00025	0.35317 ±0.00023	0.39076 ±0.00022	0.42203 ±0.00021	0.44852 ±0.00023	0.47159 ±0.00013	0.49181 ±0.00012

5.3. Effect of Payload Length

It can be deduced from Section 2.5.2 that for a LoRa transmission, the length of payload drastically affects the Time of Air (TOA) and the air time of a transmission

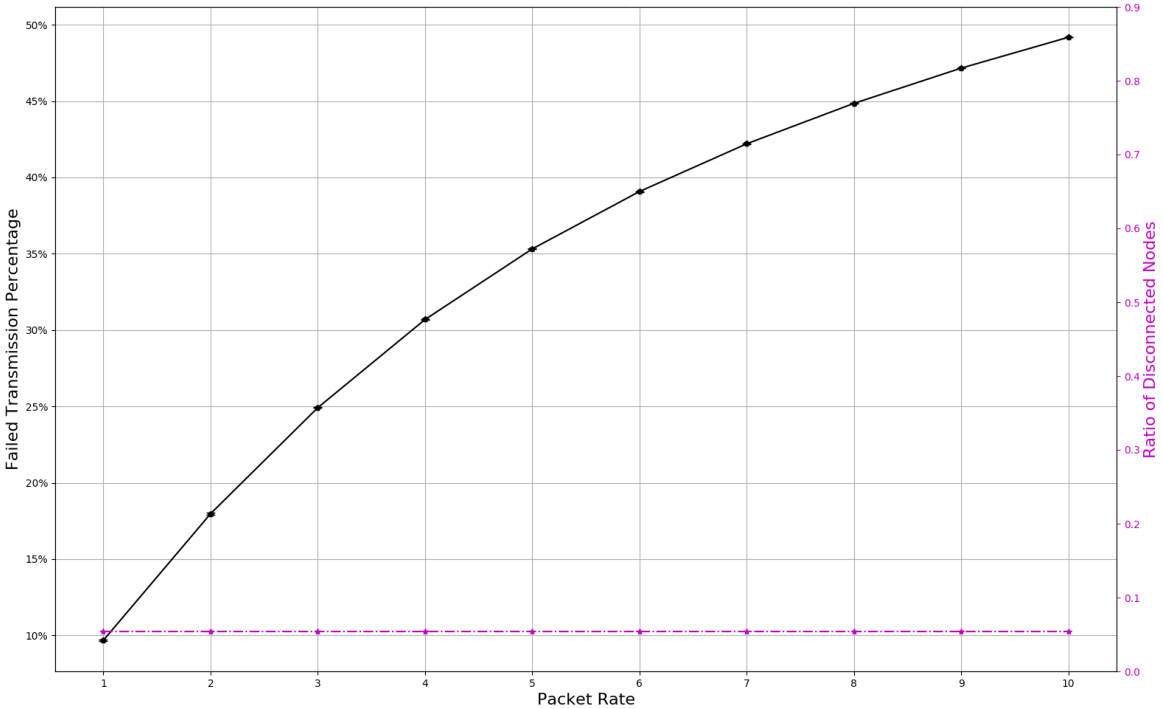

Figure 5.6. Effect of packet rate (packet/hr) on connected node transmission failure percentage.

directly affects the collision probability as seen in Equation 4.12. Multiple simulations using different payload lengths have been run for an environment with 20000 LoRa nodes. Each simulation instance is repeated 10 times with different random seeds. Simulation options are as given in Table 5.1 unless otherwise specified. The results are shared with the reader.

The chart in Figure 5.7 show average transmission success for different payload sizes for the given environment and Figure 5.8 shows the effect of payload length (including packet header) on average transmission failure percentage for connected end nodes. It can be seen from Figure 5.8 that approximately 5 percent of end nodes are not in the range of the gateway and thus not included in the calculations. The transmission failure ratio increases from 17.8% to 46.7% for the connected nodes and the overall average success decreases from 77.5% to 50% as payload length increases from 32 bytes to 224 bytes for the given environment. This effect is visualized using heat maps given in Figures 5.9 and 5.10.

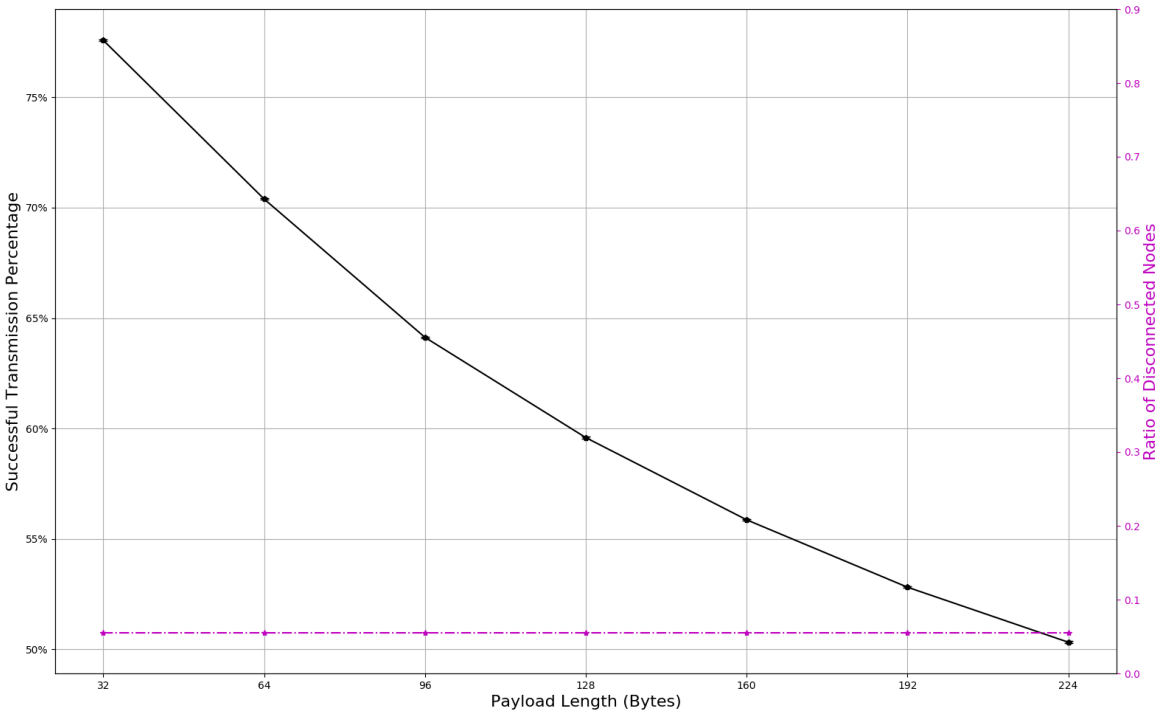

Figure 5.7. Effect of payload size on total transmission success.

Table 5.6. Average transmission success rates with confidence interval for Figure 5.7.

Payload Length	32 Bytes	64 Bytes	96 Bytes	128 Bytes	160 Bytes	192 Bytes	224 Bytes
Success Rate	0.77579 ±0.00043	0.7039 ±0.00038	0.64125 ±0.00031	0.59585 ±0.00048	0.55862 ±0.00036	0.52811 ±0.00025	0.50319 ±0.00045

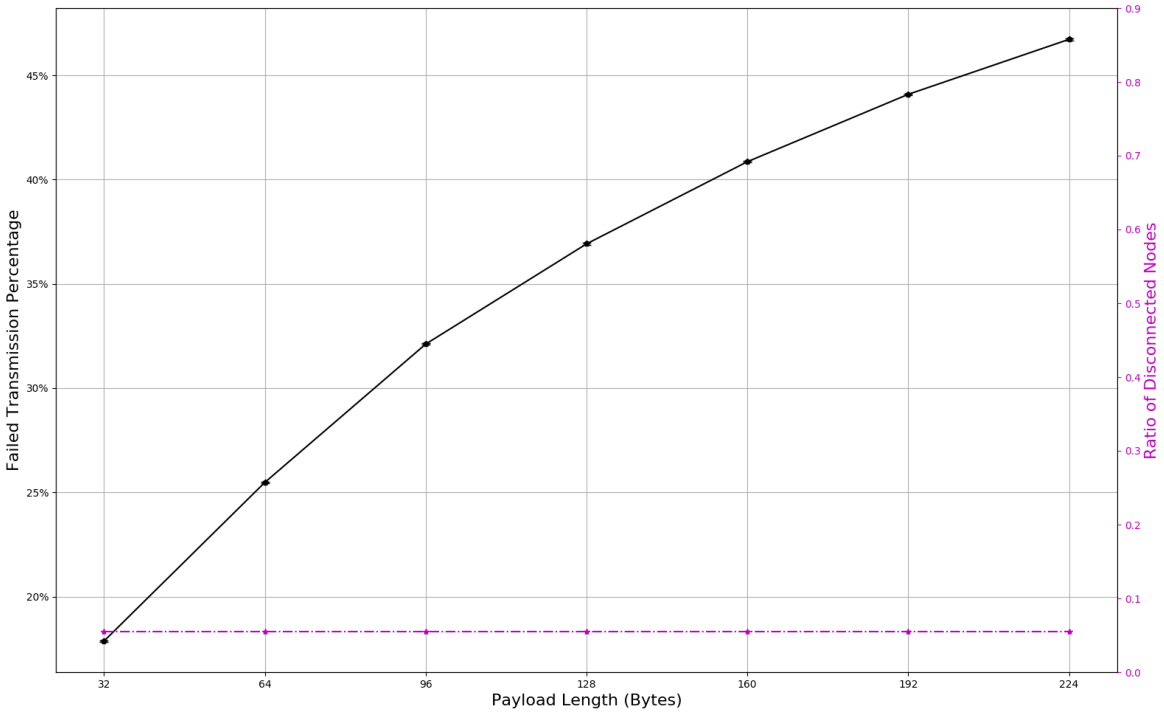

Figure 5.8. Effect of payload size on connected node transmission failure percentage.

Table 5.7. Average transmission failure rates with confidence interval for Figure 5.8.

Payload Length	32 Bytes	64 Bytes	96 Bytes	128 Bytes	160 Bytes	192 Bytes	224 Bytes
Failed Transmission Rate	0.17867 ±0.00046	0.25478 ±0.0004	0.3211 ±0.00032	0.36917 ±0.00051	0.40858 ±0.00038	0.44089 ±0.00027	0.46727 ±0.00048

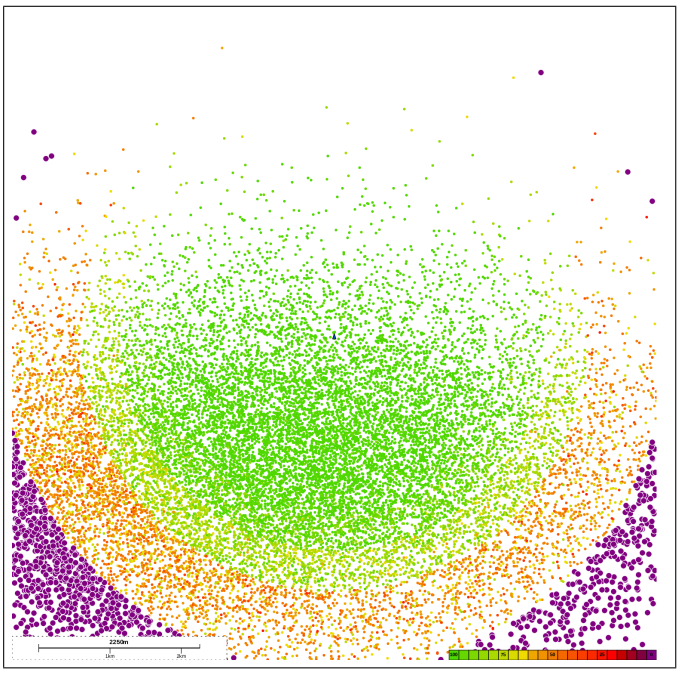

Figure 5.9. Heat map of node success for 20000 node environment with payload length 32 bytes.

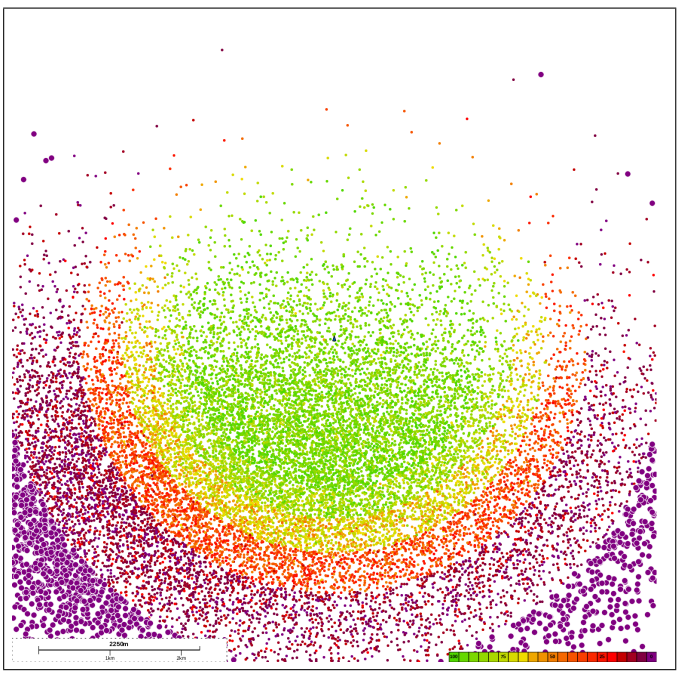

Figure 5.10. Heat map of node success for 20000 node environment with payload length 224 bytes.

5.4. Comparison of Proposed Solution Methods

In this section, we share simulation results of two LoRa network scenarios in which the LoRa gateway locations are determined by the optimization methods proposed in the earlier chapter. Three problem scenarios are shared with the reader. For each scenario, optimization algorithms were run multiple times and optimal gateway locations were found for different number of gateways. The list of used optimization algorithms are briefly summarized below for the convenience of the reader:

- Spatial: This method divides the problem area into k rectangular grids where k is the number of gateways. The gateway locations are determined as centroid of each grid.
- Random-median: This method generates k random gateway locations and calculates TOA scores for the generated gateway set. The process is repeated 1000 times. After repetitions are completed, the median air time scored solution among set of generated 1000 solutions is returned as the result.
- K-means-Euclidean: K-means partitioning is done for given end node locations based on Euclidean distances. Center points of each partition is accepted as a gateway location. The process is repeated multiple times and the result set with the sum of minimum distance to gateway is returned.
- K-means-SF: K-means partitioning is done based on the predicted spreading factors of end nodes which are derived from Euclidean distances. After multiple repetitions the result that provides highest node count that uses lower SF (e.g. SF7, SF8) are returned.
- K-means-TOA: K-means partitioning is done according to the predicted spreading factors. Repeated multiple times and the result with the lowest TOA score is returned.
- CHC-TOA: CHC algorithm using TOA cost score algorithm as evaluation function.
- CHC-NProb: CHC algorithm with naive probability algorithm (NProb) as evaluation function

- CHC-Prob: CHC algorithm with collision probability algorithm (Prob) as evaluation function

5.4.1. Experiment 1

In the first scenario, 80187 LoRa end nodes in an urban area need to periodically send information to the application server for a smart city application. The problem area is 13.5 km by 13.5 km rectangular area and the distribution of the nodes is shown in Figure 5.11. The problem parameters are as given in Table 5.1. Multiple optimization methods are compared for a problem environment consisting of 80187 LoRa end nodes.

The evaluation results are shared with the reader in Figure 5.12. In this graph, the success of algorithms are evaluated without simulation, using the metric of transmission failure probability scores which are calculated by summing up the collision probability of each node depending on the determined locations of gateways by the stated algorithm. The exact method for calculation of collision probability for a single node is explained in Equation 4.22. The algorithm with the lowest failure probability score is accepted to be the most successful in finding gateway locations. Each algorithm is run 10 times with different random seeds and their average scores are shared. In Figure 5.12, logarithmic scale is used to better visualize the predicted success of different methods using our failure probability score metric. According to the metric of failure probability score, it can bee seen that our proposed method CHC-Prob gives the result with the lowest (best) score within a small confidence interval for each number of gateways. CHC-NProb shows the second best scores with a larger confidence interval as the number of LoRa gateways increases. All of the k-means based algorithms show similar scores with slight differences. The detailed scores for each algorithm can be seen in Table 5.8. For Experiment 1, it can be seen that k-means-SF method results in slightly better (lower) scores than other two k-means based methods.

In Figure 5.13, the average algorithm execution times of the proposed algorithms on a system with Intel 2600K i7 CPU with 12GB RAM using Ubuntu 14.04 with

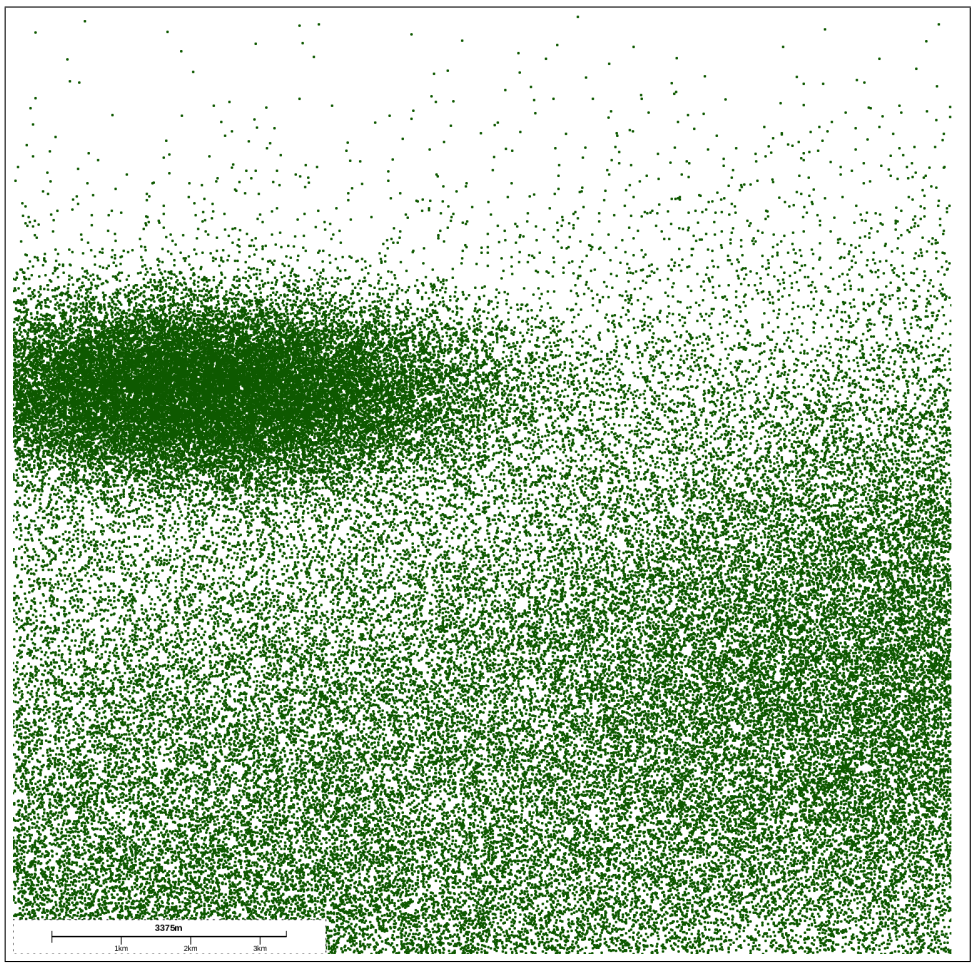

Figure 5.11. Distribution of end nodes in Experiment 1.

Python 2.7 are shown. These run times are for the same scenario in Experiment 1 that consists of 80187 nodes. The numerical values for the average execution times that are shown in Figure 5.13 for different number of gateways with 95% confidence are given in detail in Table 5.9. It is evident that the execution time for CHC-Prob is the biggest among all proposed algorithms for any number of gateways whereas CHC-NProb and CHC-TOA show the second largest execution time. As expected, for the spatial centres baseline method, the execution time is negligibly low.

Simulations of this environment that are optimized with the proposed algorithms for different number of gateways are executed and success rates for each algorithm is calculated. Additionally, the random method results are included in the same graph in order to show the average success of randomly selected gateway locations as a base-

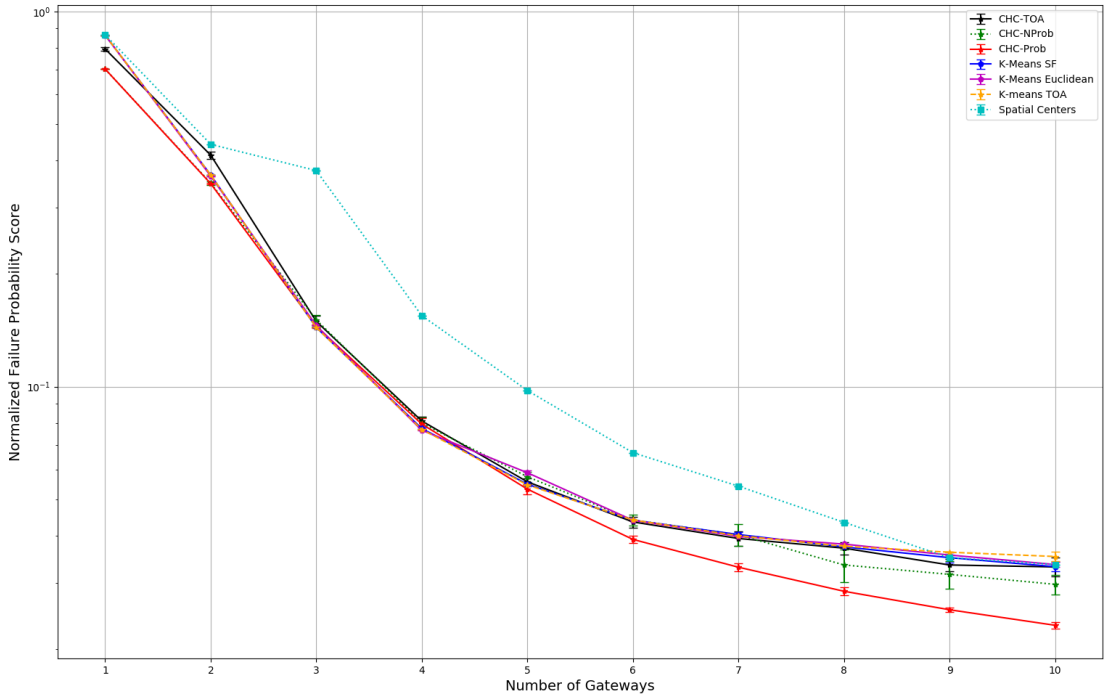

Figure 5.12. Comparison of algorithms using Transmission Failure Probability Scores (Normalized).

Table 5.8. Average failure probability scores (normalized) with 95% confidence intervals for Figure 5.12.

	1	2	3	4	5	6	7	8	9	10
CHC-TOA	0.79388 ±0.00751	0.41383 ±0.00958	0.14923 ±0.0051	0.08095 ±0.00223	0.05574 ±0.00111	0.04357 ±0.00145	0.03936 ±0.00174	0.03709 ±0.00137	0.03345 ±0.00126	0.03305 ±0.0019
CHC-NProb	0.70192 ±0.00051	0.34865 ±0.00412	0.15063 ±0.00475	0.07983 ±0.00306	0.05752 ±0.00175	0.04409 ±0.00152	0.04027 ±0.00262	0.03346 ±0.00333	0.03158 ±0.00267	0.0297 ±0.00181
CHC-Prob	0.70192 ±0.00051	0.34744 ±0.00211	0.14475 ±0.00147	0.07943 ±0.00279	0.0533 ±0.00163	0.03914 ±0.00088	0.033 ±0.00076	0.02847 ±0.00066	0.02542 ±0.00037	0.0231 ±0.00048
K-Means SF	0.86392 ±0.0	0.36654 ±0.00026	0.14373 ±6e-05	0.07732 ±0.00037	0.05493 ±0.00064	0.04404 ±4e-05	0.04032 ±0.00055	0.03739 ±0.00028	0.03499 ±0.00107	0.0331 ±0.00097
K-Means Euclidean	0.86392 ±0.0	0.36384 ±0.0	0.14548 ±0.0	0.07646 ±0.00015	0.05894 ±0.00084	0.04403 ±0.0	0.03985 ±0.0	0.03803 ±8e-05	0.03556 ±0.00011	0.03355 ±0.00052
K-means TOA	0.86392 ±0.0	0.36651 ±0.00015	0.14373 ±5e-05	0.07686 ±0.00028	0.05469 ±5e-05	0.04408 ±3e-05	0.04004 ±0.00012	0.03753 ±0.00034	0.03618 ±7e-05	0.03524 ±0.00102
Spatial Centers	0.86642 ±0.0	0.44222 ±0.0	0.37643 ±0.0	0.15459 ±0.0	0.09761 ±0.0	0.06665 ±0.0	0.05426 ±0.0	0.04346 ±0.0	0.03499 ±0.0	0.03354 ±0.0

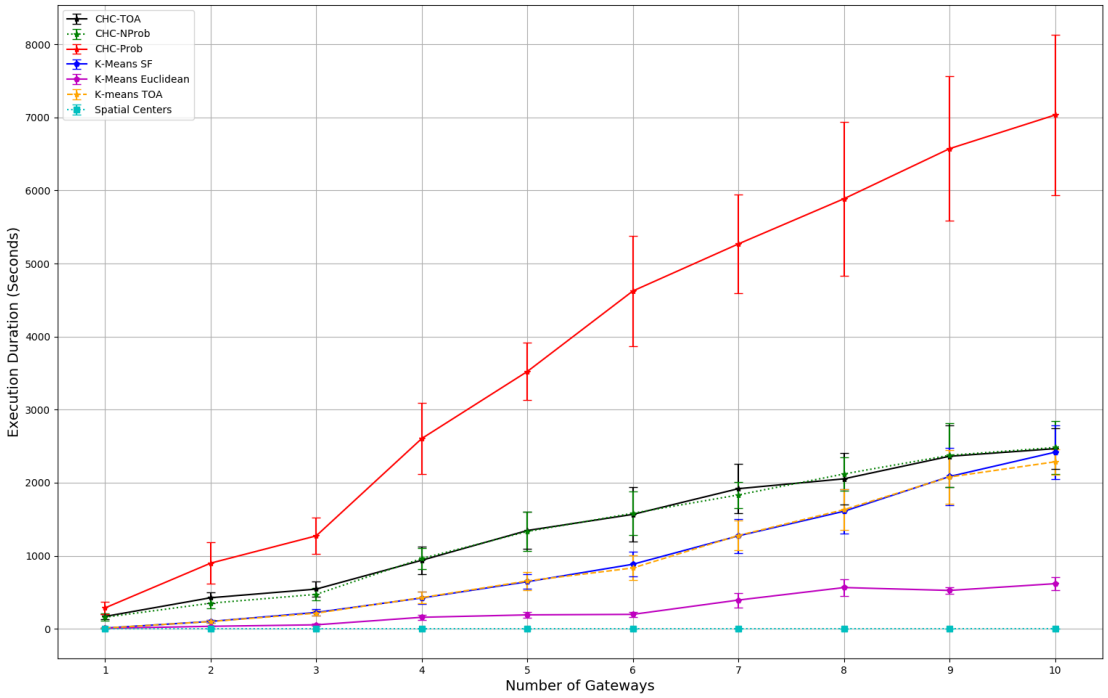

Figure 5.13. Comparison of algorithm execution times.

Table 5.9. Average execution times of proposed algorithms with 95% confidence intervals for Figure 5.13.

	1	2	3	4	5	6	7	8	9	10
CHC-TOA	168.89875 ±35.2735	424.70875 ±70.96781	543.01625 ±108.09351	937.72 ±189.90737	1345.14125 ±255.39047	1565.65375 ±369.24041	1917.845 ±334.57901	2052.69875 ±353.22055	2361.41625 ±420.86139	2465.59625 ±278.52694
CHC-NProb	157.8975 ±49.43203	350.1125 ±75.5459	470.9575 ±85.92593	961.7975 ±142.57129	1331.30375 ±271.51674	1581.67 ±298.03412	1832.5025 ±178.541	2119.02375 ±232.40429	2375.0325 ±439.10693	2482.085 ±360.39178
CHC-Prob	286.265 ±81.84755	898.52875 ±283.66455	1272.0775 ±248.77427	2604.6825 ±487.20839	3522.0775 ±392.72028	4624.42625 ±754.79987	5268.7625 ±676.51405	5887.175 ±1053.79102	6572.53 ±988.36926	7033.6125 ±1098.75015
K-Means SF	9.48875 ±1.89844	100.485 ±20.49863	223.2675 ±47.25048	421.4725 ±83.89505	643.825 ±100.33403	883.105 ±166.88372	1271.64625 ±234.51058	1608.58125 ±300.94686	2084.11625 ±389.3519	2417.21 ±371.59769
K-Means Euclidean	6.585 ±1.115	33.525 ±5.19408	53.9475 ±12.01054	157.09375 ±33.23639	190.0975 ±41.36258	196.965 ±34.25913	392.9725 ±99.80327	564.2225 ±117.10068	524.6075 ±42.53305	617.195 ±87.96439
K-means TOA	9.53 ±1.75728	98.83375 ±19.78352	213.68 ±34.9775	426.0725 ±79.59442	654.41 ±125.86526	832.20125 ±168.15178	1277.065 ±201.66369	1630.27375 ±279.95954	2078.04875 ±363.72608	2285.76125 ±174.8373
Spatial Centers	0.28 ±0.05898	0.35857 ±0.07578	0.43714 ±0.11004	0.56286 ±0.13526	0.62857 ±0.12786	0.72571 ±0.14065	0.78857 ±0.16032	0.87429 ±0.18083	1.0 ±0.16371	1.05286 ±0.20281

line. In Figure 5.14 and Table 5.10 average success and confidence intervals are shared for each algorithm for different number of gateways. The results show similar success order of algorithms (from best to worst) to the order of algorithms in the failure prob-

ability score graph in Figure 5.12. The average transmission success percentage graph shows that CHC-Prob method finds gateway locations that provide the highest average transmission success for each number of gateways. CHC-NProb mostly gives the second best average transmission success with a larger confidence interval as number of LoRa gateways increases. All of the k-means based algorithms show similar average transmission success and for Experiment 1, it can be seen that k-means-SF method results in slightly higher average transmission success than the other two k-means based methods. The average transmission success with each algorithm can be seen in detail in Table 5.10.

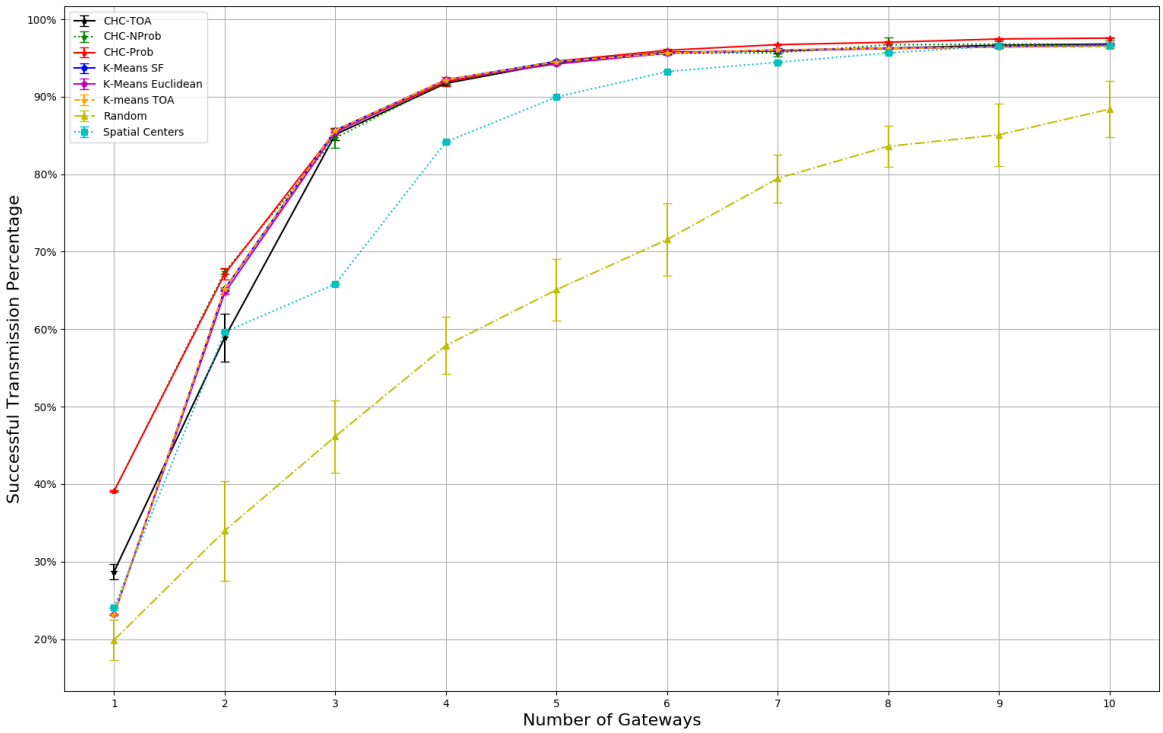

Figure 5.14. Successful transmission percentage with different number of gateways.

Furthermore, the percentage of disconnected nodes for different number of gateways with each algorithm are also shown in Figure 5.15. It can be inferred from the graphs that in order to achieve better average transmission success, the CHC-Prob and CHC-NProb algorithms may prefer improving success of connected nodes instead of directly decreasing number of disconnected nodes (increasing coverage). This decision mainly depends on the evaluation function for CHC algorithm.

Table 5.10. Successful transmission percentage of proposed algorithms with confidence interval for Figure 5.14.

	1	2	3	4	5	6	7	8	9	10
CHC-TOA	0.28711 ±0.01001	0.58913 ±0.03106	0.85129 ±0.00734	0.91734 ±0.00298	0.944 ±0.00138	0.95742 ±0.00236	0.95922 ±0.00375	0.96263 ±0.00129	0.96646 ±0.0025	0.96805 ±0.00184
CHC-NProb	0.39102 ±0.00123	0.67399 ±0.00343	0.84686 ±0.01317	0.91894 ±0.00215	0.94359 ±0.00173	0.95591 ±0.00236	0.9569 ±0.00515	0.96701 ±0.00958	0.96803 ±0.0031	0.96786 ±0.00493
CHC-Prob	0.39102 ±0.00123	0.67139 ±0.00721	0.85501 ±0.00442	0.91938 ±0.0056	0.94586 ±0.00126	0.96002 ±0.00194	0.96728 ±0.00078	0.97032 ±0.00132	0.97456 ±0.00056	0.97573 ±0.00067
K-Means SF	0.23162 ±0.00032	0.65201 ±0.00149	0.85685 ±0.00041	0.92189 ±0.00046	0.94562 ±0.00025	0.95642 ±0.00014	0.95999 ±0.00099	0.9628 ±0.00046	0.96435 ±0.0016	0.96671 ±0.00065
K-Means Euclidean	0.23162 ±0.00032	0.64697 ±0.00196	0.85408 ±0.00064	0.92222 ±0.0006	0.94228 ±0.00097	0.95629 ±0.00014	0.96039 ±0.00011	0.96204 ±0.00059	0.96471 ±0.00028	0.96645 ±0.00022
K-means TOA	0.23162 ±0.00032	0.65256 ±0.00145	0.85674 ±0.00042	0.92195 ±0.00015	0.94527 ±0.00036	0.95642 ±0.00012	0.96044 ±8e-05	0.96267 ±0.00021	0.9639 ±0.00018	0.96465 ±0.00215
Random	0.19892 ±0.02601	0.33981 ±0.06436	0.46158 ±0.04668	0.57892 ±0.03694	0.65089 ±0.03969	0.71572 ±0.04672	0.79455 ±0.03084	0.836 ±0.02652	0.85086 ±0.04037	0.88407 ±0.03649
Spatial Centers	0.24098 ±0.00073	0.59621 ±0.00103	0.65822 ±0.00045	0.84173 ±0.0003	0.89956 ±0.00046	0.93247 ±0.00032	0.94444 ±0.00037	0.95652 ±0.00023	0.96586 ±0.0008	0.96598 ±0.00041

In Figure 5.16, the simulation results for the gateway locations determined by CHC-Prob algorithm is visualized. It can be seen that as the result of CHC-Prob algorithm the gateways are placed closer to each other in the right (east) side of the area instead of covering more area. The genetic algorithm implicitly calculated that according to its evaluation function, covering more of the uncovered nodes in the north-east side would result in worse average transmission success than providing better quality coverage (lower spreading factor coverage) to many nodes in the south-east corner of the area. Thus the algorithm determined that two of the gateways should be placed close to each other in the south-east side of the problem area.

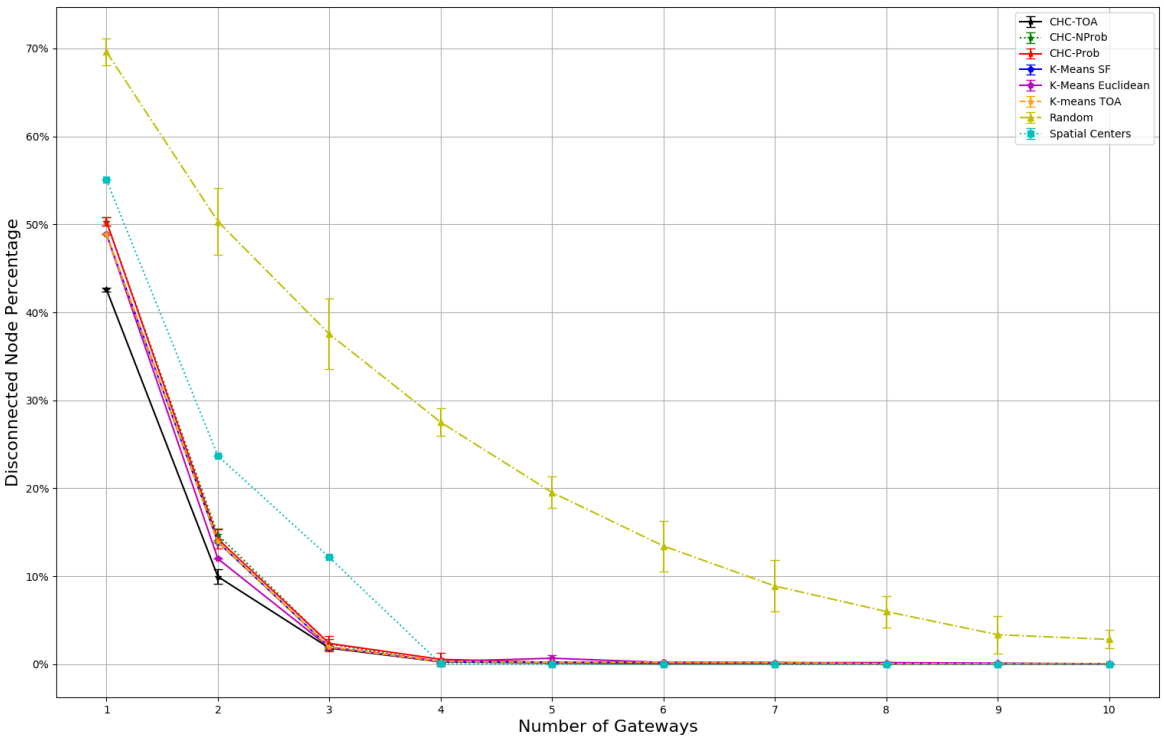

Figure 5.15. Percentage of disconnected nodes with different number of gateways.

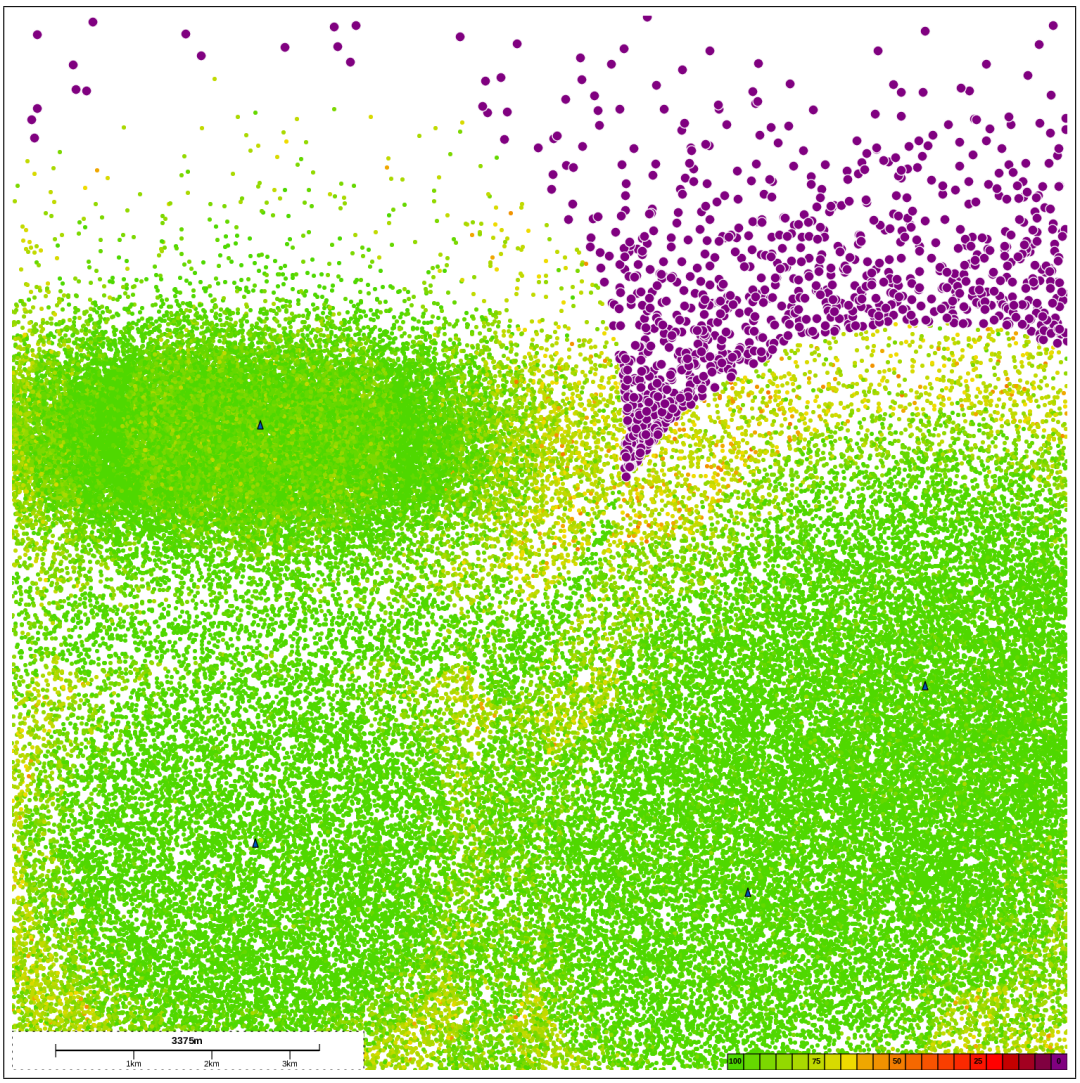

Figure 5.16. Heat map of node success for gateway locations determined by CHC-Prob optimization with 4 gateways for Experiment 1.

5.4.2. Experiment 2

In this experiment we use a more populated environment. The average density of nodes in this environment is used to simulate the building density from Kadikoy, where the acreage of Kadikoy is given as approximately 25 km^2 and number of buildings are given as 26669 [55, 56]. The building density of approximately 1100 buildings per km^2 is used to determine the number of nodes in the rectangular problem area of 13.5 km to 13.5 km and an environment with 200468 nodes is generated. The node distribution in the problem area is shown in Figure 5.17

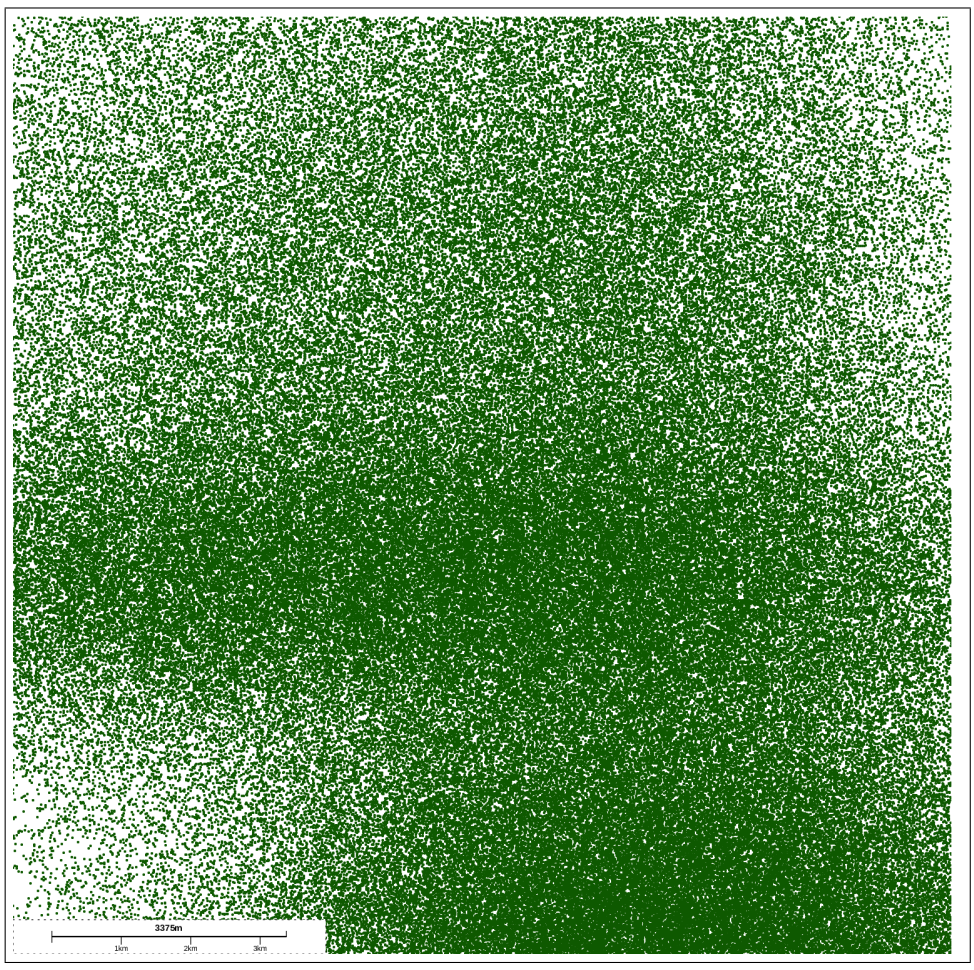

Figure 5.17. Distribution of end nodes in Experiment 2.

In order to evaluate the success of the proposed algorithms in a densely populated area, LoRa gateway locations are determined for the generated environment using

k-means-Euclidean, k-means-SF, k-means-TOA, CHC-TOA, CHC-NProb and CHC-Prob algorithms for different number of gateways. Simulations are executed for each optimization result. The simulation and packet options are as given in Table 5.1. In Figure 5.18, the success of CHC-Prob solution for five gateways is visualized as an example.

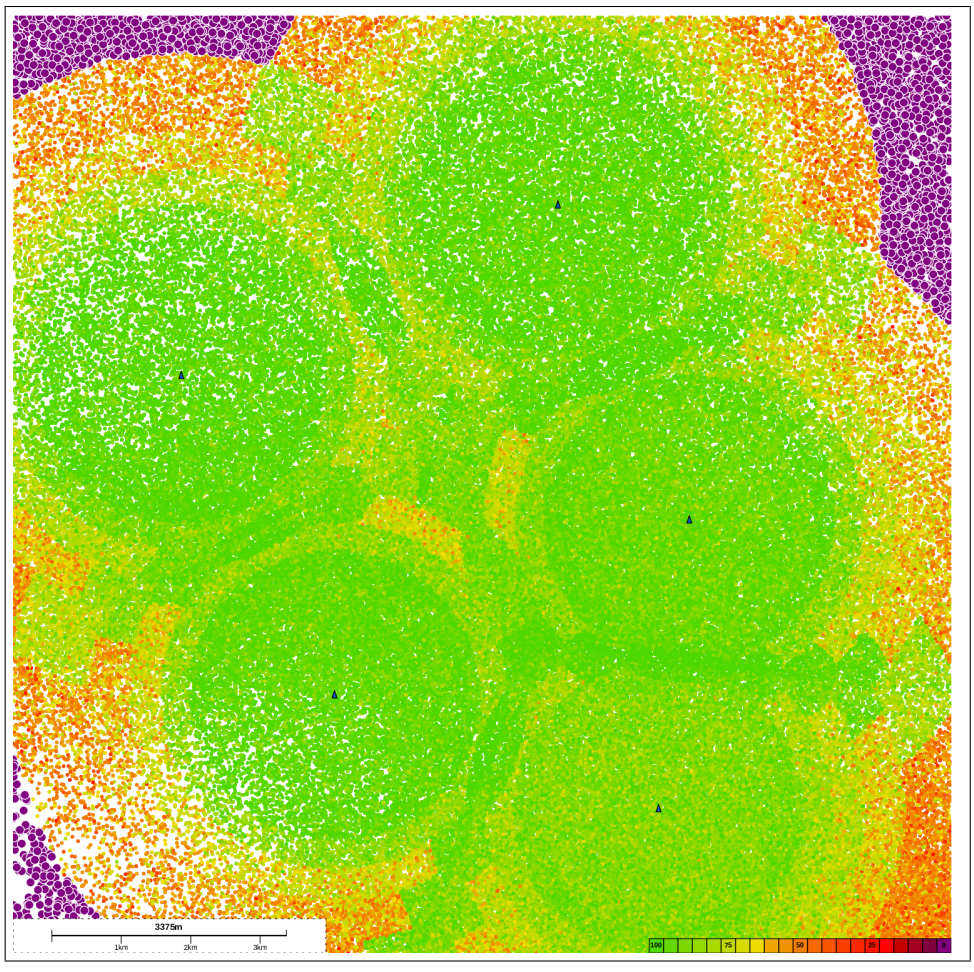

Figure 5.18. Average transmission success of CHC-Prob solution with five gateways for Experiment 2.

Success of all six algorithms are compared in Figure 5.19 where average transmission success percentages are shown for solutions with one to ten LoRa gateways. Because of very high execution times, this scenario was repeated two times with different random seeds. It can be seen from Figure 5.19 that up to five gateways, the best results are obtained using CHC-Prob algorithm, same as Experiment 1 results.

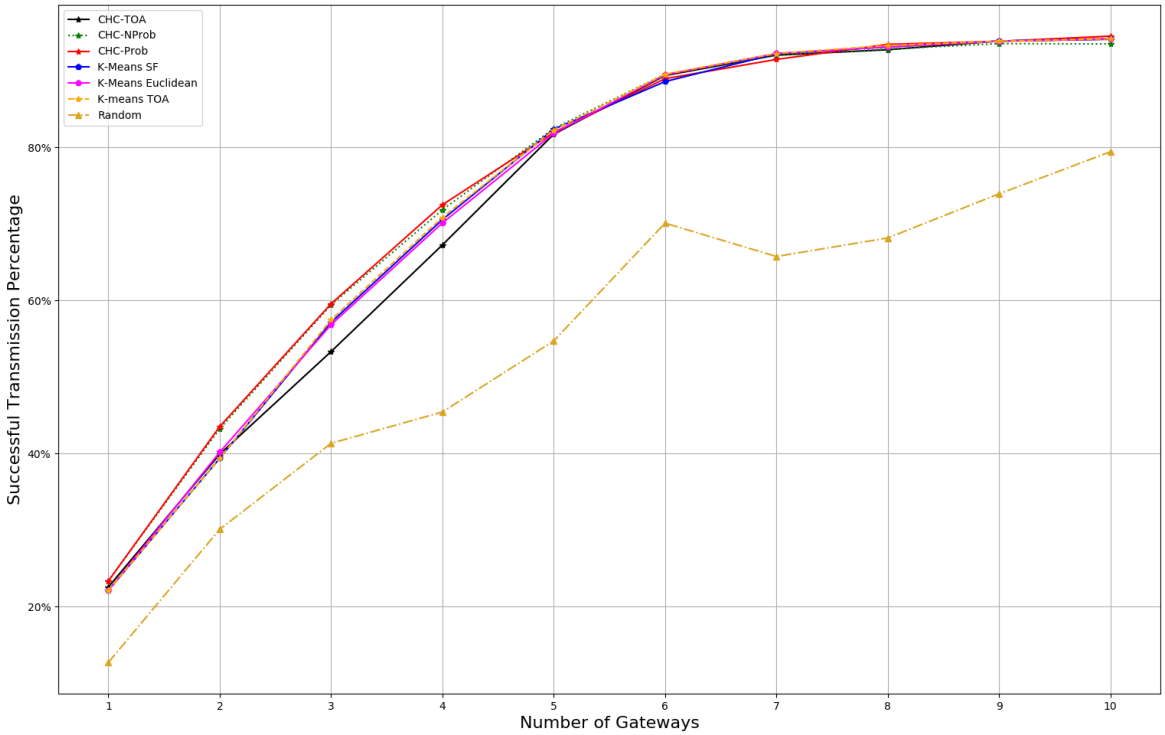

Figure 5.19. Average transmission success with proposed algorithms for Experiment 2.

However, after five gateways, as the number of gateways are increased, the success difference between k-means and CHC algorithms is decreased and all six algorithms provided solutions with similar average transmission success. All proposed algorithms performed better than the average of 1000 random gateway selection method. It is important to note that for this experiment, the iteration limits of CHC algorithms (CHC-TOA, CHC-NProb and CHC-Prob) were increased to 100 since with the default iteration count of 50, the CHC algorithms were still improving their solutions when they reached iteration limits.

6. CONCLUSION

In this thesis, we focused on LoRa and design of LoRa networks through optimal placement of LoRa gateways. We have designed and implemented a novel simulation based optimization system and proposed multiple algorithms for optimizing LoRa gateway placement. In order to achieve better placement of gateways, use of genetic algorithm and clustering algorithm based methods with multiple different metrics are proposed. The suggested methods include customized CHC and k-means algorithms with different metrics for optimal placement of LoRa gateways. The main success metrics that are used with our customized k-means based optimization algorithm are Euclidean distance of end nodes to gateway locations, predicted spreading factors of end nodes and predicted TOA of end node transmissions. Moreover, three different evaluation methods are proposed for use with our CHC implementation: CHC-TOA, CHC-NProb and CHC-Prob. Success of the all proposed methods were evaluated in multiple scenarios via our simulation system. All of our proposed algorithms performed better than the baseline methods. Among all proposed algorithms, the most successful optimization results for the medium density scenario come from CHC-Prob method which in turn has a much higher execution time compared to all other methods. Although CHC-NProb and CHC-TOA methods are in the second place and show similar success and execution times, CHC-NProb resulted in slightly better placement of gateways and slightly higher execution times. For the densely populated environment scenario, CHC-Prob algorithm performed better with lower number of gateways. However, as the number of gateways are increased, all of our k-means and CHC based algorithms showed similar success for the higher density environment.

In conclusion, we believe that our work would be beneficial for LoRa networks in urban environments and could speed up deployment of new LoRa based smart city applications by lowering costs and increasing efficiency. When locations and transmission requirements of LoRa end nodes are known or can be predicted, our simulation based optimization system can be used to determine the optimal location and opti-

mal number of LoRa gateways. Thus, with the help of proposed system, a network planner could decide on the most efficient LoRa network setup that fulfils the design requirements which in turn lowers the initial and operational costs of LoRa networks.

6.1. Future Work

We believe that the tabu list method could be combined with genetic algorithms in order to achieve shorter execution times. Furthermore, it would be beneficial if a simplified metric for congestion could be designed and failure rates due to congestion could be predicted. By using such a metric, the process of LoRa gateway placement optimization could be improved. Also, the current design of our system is for static networks only, meaning the end nodes do not move and their places are predetermined. Further research could be done for LoRa network design for dynamic environments and moving end nodes. Energy consumption of end nodes could be further researched with both optimization methods and simulations. Currently our implementation does not include the phenomenon called the capture effect. In the future this feature should also be included in the simulation in order to better simulate collisions. Additionally, optimization tools such as Gurobi can be used for determining the optimal gateway locations and the optimal number of gateways after the main problem is converted into a mixed integer programming problem and success of the resulting network may be compared with our proposed algorithms.

REFERENCES

1. Jin, J., J. Gubbi, S. Marusic and M. Palaniswami, "An information framework for creating a smart city through internet of things", *IEEE Internet of Things journal*, Vol. 1, No. 2, pp. 112–121, 2014.

2. Petrolo, R., V. Loscri and N. Mitton, "Towards a Smart City Based on Cloud of Things", *Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities*, pp. 61–66, ACM, 2014.

3. Zanella, A., N. Bui, A. Castellani, L. Vangelista and M. Zorzi, "Internet of Things for Smart Cities", *IEEE Internet of Things journal*, Vol. 1, No. 1, pp. 22–32, 2014.

4. Evans, D., "The Internet of Things: How the next evolution of the internet is changing everything", *CISCO white paper*, Vol. 1, pp. 1–11, 2011.

5. Raza, U., P. Kulkarni and M. Sooriyabandara, "Low Power Wide Area Networks: An Overview", *IEEE Communications Surveys & Tutorials*, Vol. 19, No. 2, pp. 855–873, 2017.

6. Sanchez-Iborra, R. and M.-D. Cano, "State of the Art in LPWAN Solutions for Industrial IoT Services", *Sensors*, Vol. 16, No. 5, p. 708, 2016.

7. Mekki, K., E. Bajic, F. Chaxel and F. Meyer, "A Comparative Study of LPWAN Technologies for Large-scale IoT Deployment", *ICT Express*, 2018.

8. Talari, S., M. Shafie-khah, P. Siano, V. Loia, A. Tommasetti and J. P. Catalão, "A Review of Smart Cities Based on the Internet of Things Concept", *Energies*, Vol. 10, No. 4, p. 421, 2017.

9. Cenedese, A., A. Zanella, L. Vangelista and M. Zorzi, "Padova Smart City: An Urban Internet of Things Experimentation", *Proceeding of IEEE International Sym-*

posium on a World of Wireless, Mobile and Multimedia Networks, pp. 1–6, IEEE, 2014.

10. Guibene, W., J. Nowack, N. Chalikias, K. Fitzgibbon, M. Kelly and D. Prendergast, "Evaluation of LPWAN Technologies for Smart Cities: River Monitoring Use-Case", *Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 1–5, IEEE, 2017.

11. European Telecommunications Standard Institute, *EN 300 220-2-V3.1.1 Short Range Devices operating in the frequency range 25 MHz to 1000 MHz*, 2016, `http://www.etsi.org/deliver/etsi%5fen/300200%5300299/30022002/03.01.01%530/`, accessed at October 2018.

12. Ismail, N. L., M. Kassim, M. Ismail and R. Mohamad, "A Review of Low Power Wide Area Technology in Licensed and Unlicensed Spectrum for IoT Use Cases", *Bulletin of Electrical Engineering and Informatics*, Vol. 7, No. 2, pp. 183–190, 2018.

13. Weightless SIG, *Weightless-N Open Standard IoT Networks*, 2015, `http://www.weightless.org`, accessed at October 2018.

14. Weightless SIG, *Weightless-P System specification Version 1.03*, 2017, `http://www.weightless.org`, accessed at October 2018.

15. Anupriya, K., J. Yomas and S. Jubin, "A Review on IoT Protocols for Long Distance and Low Power", *Engineering Science and Technology: An International Journal (ESTIJ)*, Vol. 5, No. 6, pp. 344–347, 2015.

16. Cetinkaya, O. and O. B. Akan, "A DASH7-Based Power Metering System", *Consumer Communications and Networking Conference*, pp. 406–411, IEEE, 2015.

17. Ingenu, *An Educational Guide: How RPMA Works*, 2018, `https://www.ingenu.com/portfolio/how-rpma-works-white-paper/`, accessed at October 2018.

18. Telensa, *Ultra Narrow Band Smart City Network*, 2018, `https://www.telensa.com/resources/data-sheet-unb-smart-city-network`, accessed at October 2018.

19. Wang, Y. P. E., X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman and H. S. Razaghi, "A Primer on 3GPP Narrowband Internet of Things", *IEEE Communications Magazine*, Vol. 55, No. 3, pp. 117–123, 2017.

20. Bor, M. C., U. Roedig, T. Voigt and J. M. Alonso, "Do LoRa Low-Power Wide-Area Networks Scale?", *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 59–67, ACM, 2016.

21. Adelantado, F., X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui and T. Watteyne, "Understanding the Limits of LoRaWAN", *IEEE Communications Magazine*, Vol. 55, No. 9, pp. 34–40, 2017.

22. Semtech Corporation, *LoRa Modulation Basics AN1200.22*, 2015, `https://www.semtech.com/uploads/documents/an1200.22.pdf`, accessed at October 2018.

23. Bor, M., J. Vidler and U. Roedig, "LoRa for the Internet of Things", *Proceedings of International Conference on Embedded Wireless Systems and Networks*, pp. 361–366, Junction Publishing, 2016.

24. Adelantado, F., X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui and T. Watteyne, "Understanding the Limits of LoRaWAN", *IEEE Communications magazine*, Vol. 55, No. 9, pp. 34–40, 2017.

25. LoRa Alliance, "White paper: A Technical Overview of LoRa and LoRaWAN", *The LoRa Alliance: San Ramon*, 2015.

26. Augustin, A., J. Yi, T. Clausen and W. M. Townsley, "A Study of LoRa: Long Range and Low Power Networks for the Internet of Things", *Sensors*, Vol. 16,

No. 9, 2016.

27. Filho, H. G. S., J. P. Filho and V. L. Moreli, "The Adequacy of LoRaWAN on Smart Grids: A Comparison with RF Mesh Technology", *2016 IEEE International Smart Cities Conference*, pp. 1–6, Sept 2016.

28. Bilgi Teknolojileri ve İletişim Kurumu, *Kısa Mesafe Erişimli Telsiz Cihazları Hakkında Yönetmelik*, 2018, `http://www.mevzuat.gov.tr`, accessed at October 2018.

29. The Things Network, *LoRaWAN Frequencies Overview*, 2015, `https://www.thethingsnetwork.org/docs/lorawan/frequency-plans.html`, accessed at October 2018.

30. The Things Network, *Duty Cycle for LoRaWAN Devices*, 2015, `https://www.thethingsnetwork.org/docs/lorawan/duty-cycle.html`, accessed at October 2018.

31. Semtech Corporation, *SX1272/3/6/7/8: LoRa Modem Design Guide AN1200.13 Revision 1*, 2013, `https://www.semtech.com/uploads/documents/LoraDesignGuide-STD.pdf`, accessed at October 2018.

32. Semtech Corporation, *Semtech Datasheet SX1301 v2.4 June 2017*, 2017, `http://www.semtech.com/images/datasheet/sx1301.pdf`, accessed at October 2018.

33. IMST GmbH, *WiMOD iC880A Datasheet v1*, 2018, `https://wireless-solutions.de`, accessed at October 2018.

34. The Things Network, *Limitations of LoRaWAN*, 2018, `https://www.thethingsnetwork.org/docs/lorawan/limitations.html`, accessed at October 2018.

35. Haxhibeqiri, J., F. Van den Abeele, I. Moerman and J. Hoebeke, "Lora Scalability: A Simulation Model Based on Interference Measurements", *Sensors*, Vol. 17, No. 6, p. 1193, 2017.

36. Akkaya, K., M. Younis and W. Youssef, "Positioning of Base Stations in Wireless Sensor Networks", *IEEE Communications Magazine*, Vol. 45, No. 4, 2007.

37. Pan, J., L. Cai, Y. T. Hou, Y. Shi and S. X. Shen, "Optimal Base-station Locations in Two-tiered Wireless Sensor Networks", *IEEE Transactions on Mobile Computing*, Vol. 4, No. 5, pp. 458–473, 2005.

38. Wright, M. H., "Optimization Methods for Base Station Placement in Wireless Applications", *48th Vehicular Technology Conference*, Vol. 1, pp. 387–391, IEEE, 1998.

39. Aspnes, J., T. Eren, D. K. Goldenberg, A. S. Morse, W. Whiteley, Y. R. Yang, B. D. Anderson and P. N. Belhumeur, "A Theory of Network Localization", *IEEE Transactions on Mobile Computing*, Vol. 5, No. 12, pp. 1663–1678, 2006.

40. Raisanen, L. and R. M. Whitaker, "Comparison and Evaluation of Multiple Objective Genetic Algorithms for the Antenna Placement Problem", *Mobile Networks and Applications*, Vol. 10, No. 1-2, pp. 79–88, 2005.

41. Oyman, E. I. and C. Ersoy, "Multiple sink network design problem in large scale wireless sensor networks", *Communications, 2004 IEEE International Conference on*, Vol. 6, pp. 3663–3667, IEEE, 2004.

42. Taysi, Z. C., M. A. Guvensan and A. G. Yavuz, "Multiple Base Station Placement with K-means Local+", *17th Signal Processing and Communications Applications Conference*, pp. 896–899, IEEE, 2009.

43. Rehena, Z., S. Roy and N. Mukherjee, "Topology Partitioning in Wireless Sensor Networks Using Multiple Sinks", *Computer and Information Technology (ICCIT),*

2011 14th International Conference on, pp. 251–256, IEEE, 2011.

44. Das, D., Z. Rehena, S. Roy and N. Mukherjee, "Multiple-sink Placement Strategies in Wireless Sensor Networks", *Fifth International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–7, IEEE, 2013.

45. Poe, W. Y. and J. B. Schmitt, *Minimizing the Maximum Delay in Wireless Sensor Networks by Intelligent Sink Placement*, 2007, `https://disco.cs.uni-kl.de/discofiles/publicationsfiles/SP07-1.pdf`, accessed at December 2018.

46. Harwahyu, R., A. Presekal and R. F. Sari, "LoRaWAN Performance Evaluation with Optimized Configuration", *International Journal of Future Generation Communication and Networking*, Vol. 11, No. 4, pp. 51–68, 2018.

47. Magrin, D., M. Centenaro and L. Vangelista, "Performance Evaluation of LoRa Networks in a Smart City Scenario", *International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2017.

48. Regula, P., I. Pozniak-Koszalka, L. Koszalka and A. Kasprzak, "Evolutionary Algorithms for Base Station Placement in Mobile Networks", *Asian Conference on Intelligent Information and Database Systems*, pp. 1–10, Springer, 2011.

49. Vega-Rodríguez, M. A., J. A. Gómez-Pulido, E. Alba, D. Vega-Pérez, S. Priem-Mendes and G. Molina, "Evaluation of Different Metaheuristics Solving the RND Problem", *Workshops on Applications of Evolutionary Computation*, pp. 101–110, Springer, 2007.

50. Nebro, A. J., E. Alba, G. Molina, F. Chicano, F. Luna and J. J. Durillo, "Optimal Antenna Placement Using a New Multi-objective CHC Algorithm", *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pp. 876–883, ACM, 2007.

51. Alba, E., G. Molina and F. Chicano, "Optimal Placement of Antennae Using Metaheuristics", *International Conference on Numerical Methods and Applications*, pp. 214–222, Springer, 2006.

52. Hata, M., "Empirical Formula for Propagation Loss in Land Mobile Radio Services", *Transactions on Vehicular Technology*, Vol. 29, No. 3, pp. 317–325, 1980.

53. Jörke, P., S. Böcker, F. Liedmann and C. Wietfeld, "Urban Channel Models for Smart City IoT-networks Based on Empirical Measurements of LoRa-links at 433 and 868 MHz", *28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1–6, IEEE, 2017.

54. Eshelman, L. J., "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination", *Foundations of Genetic Algorithms*, Vol. 1, pp. 265–283, Elsevier, 1991.

55. Kadikoy Belediyesi, *Kadikoy Belediyesi 2015-2019 Stratejik Plan 2017-2019 Dönemi için Güncellenmiş Plan*, 2017, `http://www.kadikoy.bel.tr/Documents/file/dosya/2017-2019%20D%C3%96NEM%C4%B0%20%C4%B0%C3%87%C4%B0N%20G%C3%9CNCELLENM%C4%B0%C5%9E%20PLAN.pdf`, accessed at November 2018.

56. Harita Genel Müdürlüğü, *İl ve İlçe Yüz Ölçümleri*, 2014, `https://www.hgk.msb.gov.tr/images/urun/il_ilce_alanlari.pdf`, accessed at November 2018.