ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE

ENGINEERING AND TECHNOLOGY

SMART SPREADING FACTOR ASSIGNMENT FOR LORAWANS

M.Sc. THESIS

Tuğrul YATAĞAN

Department of Computer Engineering

Computer Engineering Programme

JUNE 2019

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE

ENGINEERING AND TECHNOLOGY

SMART SPREADING FACTOR ASSIGNMENT FOR LORAWANS

M.Sc. THESIS

Tuğrul YATAĞAN
(504161551)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Sema Fatma OKTUĞ

JUNE 2019

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

LORAWAN'LAR İÇİN AKILLI YAYILMA FAKTÖRÜ ATAMASI

YÜKSEK LİSANS TEZİ

Tuğrul YATAĞAN
(504161551)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Sema Fatma OKTUĞ

HAZİRAN 2019

Tuğrul YATAĞAN, a M.Sc. student of ITU Graduate School of Science Engineering and Technology 504161551 successfully defended the thesis entitled "SMART SPREADING FACTOR ASSIGNMENT FOR LORAWANS", which he/she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Sema Fatma OKTUĞ** Istanbul Technical University	....
Jury Members :	**Doç. Dr. Berk CANBERK** Istanbul Technical University	....
	Prof. Dr. Tuna TUĞCU Boğaziçi University	....

Date of Submission : **3 May 2019**
Date of Defense : **10 June 2019**

To my parents,
For all their support.

FOREWORD

Firstly, I would like to thank to my supervisor Prof. Dr. Sema Fatma OKTUĞ for all of her support and guidance during my M.Sc. thesis.

I appreciate support of the Turkish Ministry of Development and Istanbul Technical University researcher support program under the Grant No. ITU-AYP-2017-1.

Finally, I would like to express my gratitude to my family for their continuous support through all my life.

June 2019

Tuğrul YATAĞAN
Computer Engineer

TABLE OF CONTENTS

ABBREVIATIONS

2G : Second Generation Cellular Technology
3G : Third Generation Cellular Technology
ADR : Adaptive Data Rate
B : Byte
bps : Bits per Second
BW : Bandwidth
CRC : Cyclic Redundancy Check
CSS : Chirp Spread Spectrum
dB : Decibel
dBm : Decibel Milliwatt
DTC : Decision Tree Classifier
EIRP : Equivalent Isotropically Radiated Power
ETSI : European Telecommunications Standards Institute
EN : End Node
GW : Gateway
Hz : Hertz
IEEE : Institute of Electrical and Electronics Engineers
IoT : Internet of Things
ISM : Industrial, Scientific and Medical
J : Joule
kHz : Kilohertz
km : Kilometer
LPWA : Low Power Wide Area
LPWAN : Low Power Wide Area Network
LTE : Long Term Evolution
LTE-M : Long Term Evolution Category M1
MAC : Media Access Control
MHz : Megahertz
NB-IoT : Narrowband Internet of Things
NS : Network Server
PDR : Packet Delivery Ratio
pps : Packets per Second
SF : Spreading Factor
SINR : Signal to Interference plus Noise Ratio
SNR : Signal to Noise Ratio
SVM : Support Vector Machine
TX : Transmit
RX : Receive

SYMBOLS

f	: Frequency
$\mathbf{G_{SYS}^{dB}}$	: System Gains
h	: Height
$\mathbf{L_{PATH}^{dB}}$	: Propagation Path Loss
$\mathbf{L_{SYS}^{dB}}$	: System Losses
n	: Number of Nodes
$\mathbf{P_{RX}^{dBm}}$	: Receive Power
$\mathbf{P_{TX}^{dBm}}$	: Transmit Power
r	: Radius
$\mathbf{R_b}$	: Bit Rate
$\mathbf{R_t}$	: Traffic Rate

LIST OF TABLES

Page

LIST OF FIGURES

Page

SMART SPREADING FACTOR ASSIGNMENT FOR LORAWANS

SUMMARY

Low power wide area network (LPWAN) technologies offer affordable wireless connectivity to massive number of low-power devices distributed over large geographical areas. Traditional wireless communication methods such as cellular networks, Bluetooth, WiFi cannot provide low power and long range at the same time. Cellular networks can provide long range and high data rate, but they are complex and consume too much power. Bluetooth and WiFi can provide relatively low power consumption, but their range is limited. LPWAN technologies waive data rate and latency to provide low power and long range communication. Recently, LPWAN technologies have become popular for Internet of Things (IoT) and smart city applications especially when low power consumption is critical.

Focus of this thesis is one of the most promising LPWAN technologies: LoRa. LoRa is a proprietary LPWAN technology developed by Semtech Corporation. A non-profit organization, LoRa Alliance, has developed an open standard medium access control (MAC) layer protocol called LoRaWAN to create large scale LoRa networks and interoperability between these networks. Low power embedded device in a LoRaWAN network is called end node. The device connected to the power grid that can receive from multiple channels at the same time is called gateway. Network server is a server that provides application layer processing. Network server can tweak end node communication parameters. LoRaWAN end nodes can only communicate with gateways, end nodes cannot communicate with each other. Transmission from a single end node can be received by multiple gateways. To keep the power consumption low, LoRaWAN is based on pure ALOHA medium access control, which means that end nodes do not check whether the channel is available or not before transmission. While an end node is transmitting, another end node can initiate a transmission and a collision may occur at the gateway.

LoRa is a trademarked modulation technique based on Chirp Spread Spectrum (CSS). CSS provides long range communication and resilience to interference. In LoRa modulation, the signal frequency scans the band from end to end within a particular channel. The direction of this scan determines the transmission symbol. The speed of the scan is called the spreading factor (SF). LoRa supports 6 different spreading factor options between 7 to 12. By changing the spreading factor of a LoRa transmission, it is possible to increase the communication range by sacrificing the data rate. In short, as the spreading factor increases, the data rate decreases and the power consumption increases, however the communication range increases. In addition, transmissions with different LoRa spreading factors are orthogonal to each other. In other words, LoRa transmissions with different spreading factors within the same channel can communicate up to some extent without causing any interference to each other. Therefore, spreading factor selection of the end nodes significantly affects the number of collisions, thus the overall network performance. It is difficult for end nodes to select

the best spreading factor for them, since end nodes are not aware of the transmissions around them. End nodes select the lowest spreading factor they can to communicate with the gateways to keep the power consumption low, to keep the communication duration short, and reduce the likelihood of collisions. However, when other end nodes around them begin to transmit with the same spreading factor, the probability of collisions increases. Same spreading factor transmissions can significantly reduce network performance in densely deployed networks. In some cases, assigning a higher spreading factor to the end nodes may increase the network packet delivery ratio, even if the end nodes are close to the gateway. In this work, other related academic studies on spread factor assignment issue for LoRaWANs are investigated and the current state of the art technologies are shown.

In this study, a new discrete event simulation software is developed from scratch with Python programming language in order to study the effect of spreading factor assignment on LoRaWAN network performance. LoRaWAN networks with multiple gateways and multiple end nodes can be simulated with the developed simulation tool. The simulation tool can be fed with input parameters such as spreading factor assignment method, number of gateways, number of nodes, network radius, packet size, packet generation rate, packet generation type and with these inputs the tool produces simulation results such as total number of generated packets, number of successfully received packets, number of interfered packets, number of under receive sensitivity packets, network packet delivery ratio percentage, network throughput, total transmit energy consumption.

In the simulation tool, LoRa link quality model provided by Semtech Corporation is employed. Also, interference model described in previous academic studies is employed for the interference between same and different spreading factor LoRa transmissions. Transmission results (successful, collision or under sensitivity) are calculated using these two models. Other wireless communication technology interferences are ignored. Simulation results for various number of gateways, number of nodes, network topology radius, packet size inputs are presented. Then, correctness of the simulation tool outputs is examined.

In this study, the effect of spreading factor assignment on network performance is investigated in detail. The factors that increase the number of collisions is evaluated and spreading factor increase method that can be taken to reduce the number of collisions is described. Increasing spreading factor increases transmission duration thus increases the probability of collisions with other high spreading factor transmissions. Especially, in multiple gateways scenarios, this approach may increase the collisions with the nodes in other gateways' range. Thus, extra care should be taken for nodes in the intersection area of the gateways. It is difficult to propose a single spreading factor assignment rule for every possible LoRaWAN topology since every network is different. For this reason, machine learning based spreading factor assignment approach is proposed. This network aware approach does not require any fixed spreading factor assignment rule to select the most efficient spreading factor. Support Vector Machine (SVM) and Decision Tree Classifier (DTC) machine learning methods are used for this new method called smart spreading factor assignment. In this method, network server first monitors the location of each node, spreading factor of each transmission and result of each transmission. Then, network server trains a transmission result prediction model with accumulated data by using SVM or DTC machine learning methods. With this model, the most efficient spreading factor is calculated for subsequent transmissions of the nodes. For each end node, transmission result prediction is calculated one by one from lowest spreading factor to highest spreading factor. The spreading factor of the first transmission result projected as successful is selected for the end node. Network server notifies the new spreading factors to the end nodes via gateways. The end nodes will begin to use the new spreading factors for their subsequent transmissions. This process is repeated daily basis for every end nodes and new spreading factors are assigned to end nodes once in a day.

The smart spreading factor assignment method is integrated into the developed simulation tool. First, the tool randomly assigns spreading factors to the end nodes and simulates the network. Then, the smart spreading factor assignment prediction model is generated using the accumulated transmission records and new spreading factors are calculated for each node. Finally, the tool simulates the same network again with the new spreading factors.

The simulation results show that the smart spreading factor assignment method yields better network packet delivery ratio and total transmission energy consumption than random spreading factor assignment method and lowest spreading factor assignment method. DTC machine learning method gives better packet delivery ratio results than SVM machine learning method. These two smart spreading factor assignment methods provide promising network performance improvements, especially for dense LoRaWAN networks.

LORAWAN'LAR İÇİN AKILLI YAYILMA FAKTÖRÜ ATAMASI

ÖZET

Düşük güç geniş alan ağ (DGAA) teknolojileri, geniş coğrafi alanlara yayılmış çok sayıda düşük güçlü cihaza ekonomik kablosuz haberleşme altyapısı sağlar. Hücresel ağlar, Bluetooth, WiFi gibi geleneksel kablosuz haberleşme teknolojileri düşük güç ve uzun haberleşme menzilini aynı anda sağlayamazlar. Hücresel ağlar uzun menzil ve yüksek veri hızı sağlayabilirler fakat karmaşık ve yüksek güç tüketimine sahiptirler. Bluetooth ve WiFi gibi kablosuz haberleşme teknolojileri ise düşük güç tüketimi sağlayabilirler fakat haberleşme menzilleri kısıtlıdır. DGAA teknolojileri veri hızı ve gecikme süresinden feragat edip düşük güç tüketimi ve uzun haberleşme menzili sağlamayı hedefler. Son yıllarda DGAA teknolojileri düşük güç gereksinimi isteyen nesnelerin interneti ve akıllı şehir uygulamaları için daha yoğun kullanılmaya başlanmıştır.

Bu tezin odağı yaygın olarak kullanılan DGAA teknolojilerinden biri olan LoRa üzerinedir. LoRa, Semtech şirketi tarafından geliştirilen tescilli bir DGAA teknolojisidir. Kâr amacı gütmeyen bir kuruluş olan LoRa Alliance tarafından büyük ölçekli LoRa ağları oluşturmak ve bu ağlar arasında birlikte çalışabilirlik sağlamak amacıyla LoRaWAN adı verilen açık kaynak bir ortama erişim kontrol standardı geliştirilmiştir. LoRaWAN ağlarındaki düşük güçlü, adanmış bir göreve sahip cihazlara uç düğüm adı verilir. LoRaWAN ağlarında birden fazla kanalı sürekli dinleyip güç şebekesine ve sabit bir hatta sahip cihazlara ise ağ geçidi adı verilir. Ağ sunucusu ise ağ ile ilgili uygulamaların çalıştığı, ağın beyni konumundadır. Ağ sunucusu ağın performansını arttırmak için uç düğümlerin haberleşme parametrelerini değiştirebilir. LoRaWAN uç düğümleri sadece ağ geçitleri ile haberleşebilir, kendi aralarında haberleşme yapamazlar. Bir uç düğüm birden fazla ağ geçidi tarafından dinlenebilir. Güç tüketimini düşük tutmak için uç düğümler yalın ALOHA prensibi ile haberleşme yaparlar, yani uç düğümler yayın yapmaya başlamadan önce kablosuz ortamının müsait olup olmadığını kontrol etmezler. Bir uç düğüm ağ geçidi ile haberleşirken başka bir uç düğüm haberleşmeye başlayabilir ve sinyal çakışmaya uğradığı için haberleşme ağ geçidi tarafından çözümlenemeyebilir.

LoRa, Chirp Spread Spectrum (CSS) tabanlı marka tescilli bir modülasyon tekniğidir. CSS tabanlı olması, uzun menzilli iletişim ve parazitlere karşı dayanıklılık sağlar. LoRa modülasyonu, sinyal frekansının belirli bir kanal içerisinde bandı baştan sona taramasıyla çalışır. Bu taramanın yönü haberleşme sembolünü belirler. Taramanın hızı ise yayılma faktörü (YF) olarak adlandırılır. LoRa 7'den 12'ye kadar 6 farklı yayılma faktörünü desteklemektedir. Bir LoRa haberleşmesinin yayılma faktörünü değiştirerek, veri hızından feda edip, haberleşme menzili arttırılabilir. Kısaca yayılma faktörü arttıkça veri iletim hızı azalır ve güç tüketimi artar fakat karşılığında haberleşme menzili uzar. Ayrıca farklı LoRa yayılma faktörüne sahip haberleşmeler birbirlerine karşı ortogonaldir. Yani aynı kanal içerisinde farklı yayılma faktörüne sahip LoRa iletişimleri, belirli bir seviyeye kadar birbirlerine parazit oluşturmadan

haberleşebilir. Bu sebeple uç düğümlerin yayılma faktörü seçimi gerçekleşen çakışma sayısına ve ağ performansına önemli ölçüde etki eder. Uç düğümlerin kendileri için en iyi yayılma faktörünü seçmesi zordur çünkü uç düğümler etraflarındaki yayınlardan haberdar değildir. Uç düğümler güç tüketimini düşük tutmak ve haberleşme süresini kısa tutup çakışma olasılığını düşürmek amacıyla ağ geçidi ile haberleşebilecekleri en düşük yayılma faktörünü seçerler. Fakat etraflarındaki diğer uç düğümler de aynı yayılma faktörü ile yayın yaptıklarında çakışma ihtimali artar. Özellikle yoğun uç düğüm dağılımına sahip ağlarda aynı yayılma faktörlü haberleşmeler ağ performansını ciddi şekilde düşürebilir. Bazı durumlarda uç düğümler ağ geçidine yakın olsalar bile daha yüksek bir yayılma faktörü seçmeleri ağdaki başarılı paket iletim oranını arttırabilir. Bu çalışma içerisinde, LoRaWAN ağları için yayılma faktörü atamasını iyileştirmek amacıyla yapılmış diğer akademik çalışmalar araştırılmıştır ve güncel teknolojik durum incelenmiştir.

Bu çalışmada yayılma faktörü seçiminin LoRaWAN ağ performansına etkisini gözlemlemek amacıyla Python programlama dili ile yeni bir ayrık olay simülasyon yazılımı geliştirilmiştir. Geliştirilen simülasyon aracı ile çoklu ağ geçitlerine ve uç düğümlere sahip LoRaWAN ağları simüle edilebilir. Simülasyon aracı; yayılma faktörü ataması yöntemi, ağ geçidi sayısı, uç düğüm sayısı, ağ topolojisi yarıçapı, paket büyüklüğü, paket üretilme sıklığı, paket üretilme tipi gibi girdiler alarak; üretilen paket sayısı, iletilen paket sayısı, çakışmaya uğramış paket sayısı, alıcı hassasiyetinin altında kalmış paket sayısı, ağ paket iletilme oranı, ağ veri hızı, toplam iletim enerji tüketimi gibi çıktılar üretebilir.

Simülasyon aracında, Semtech Şirketi tarafından sağlanan, LoRa haberleşmesi bağlantı kalitesi modeli kullanılmıştır. Aynı ve farklı yayılma faktörlerine sahip LoRa haberleşmelerinin girişim modeli için ise, önceki akademik çalışmalarda kullanılan modeller kullanılmıştır. Paketlerin iletim sonuçları (başarılı, çakışmaya uğramış veya alıcı hassasiyeti altında kalmış) bu iki model kullanılarak hesaplanmıştır. Farklı kablosuz haberleşme teknolojilerinin yarattığı parazitler ihmal edilmiştir. Simülasyon aracı ile çeşitli ağ geçidi sayısı, uç düğüm sayısı, ağ topolojisi yarıçapı, paket büyüklüğü, paket üretilme sıklığı ile üretilmiş çıktılar paylaşılmış ve simülasyon aracının ürettiği çıktıların doğruluğu incelenmiştir.

Bu çalışmada yayılma faktörü seçiminin ağ performansına etkisi detaylı olarak incelenmiştir. Çakışma sayısını arttıran durumlar incelenmiştir ve bu durumların değerlendirilmesi ile çakışma sayısını düşürebilecek yayılma faktörü arttırma yöntemi anlatılmıştır. Yayılma faktörünü arttırmak haberleşme süresini uzattığı için yüksek yayılma faktörüne sahip haberleşmeler ile olan çakışma ihtimalini arttırır. Özellikle birden fazla ağ geçidine sahip ağlarda, ağ geçitlerinin kapsama alanlarının kesiştiği bölgelerde kalan uç düğümlerin yayılma faktörlerinin arttırılması çok dikkatli yapılmalıdır. Her bir LoRaWAN ağı birbirinden farklı olduğu için uç düğümlerin yayılma faktörlerini yönetmek için tek bir yayılma faktörü atama kuralı türetmek zordur. Bu sebeple yayılma faktörü seçimi için makine öğrenmesi kullanan yeni bir yöntem önerilmiştir. Ağın yapısının ve güncel durumunun farkında olan makina öğrenme tabanlı bu yeni yaklaşım ile sabit bir kurala bağlı kalmadan uç düğümler için en uygun yayılma faktörü ataması yapılabilir. Akıllı yayılma faktörü ataması adını verdiğimiz bu yeni yöntem için Destek Vektör Makinesi (DVM) ve Karar Ağacı Öğrenmesi (KAÖ) makine öğrenmesi yöntemleri kullanılmıştır. Bu yöntemde ilk olarak ağ sunucusu ağdaki tüm düğümlerin konumlarını, haberleşmelerinde kullandıkları yayılma faktörlerini ve haberleşme sonuçlarını kayıt altına alır. Daha sonra ağ sunucusu biriktirilen bu veriler üzerinde DVM veya KAÖ yöntemleri ile bir haberleşme tahmin modeli eğitir. Eğitilen bu model ile uç düğümlerin sonraki haberleşmeleri için en uygun yayılma faktörü hesaplanır. Bu model ile her bir uç düğüm için en düşük yayılma faktöründen en yükseğe doğru sırayla haberleşme sonucu tahmin edilir. Haberleşme sonucu başarılı olarak tahmin edilen en düşük yayılma faktörü o uç düğüm için seçilir. Ağ sunucusu, ağ geçitleri vasıtasıyla uç düğümlere yeni yayılma faktörlerini iletir. Uç düğümler sonraki haberleşmelerinde akıllı yayılma faktörü ataması yöntemi ile bulunmuş yeni yayılma faktörlerini kullanır. Bu işlem günlük olarak tüm uç düğümler için tekrarlanır ve yeni yayılma faktörleri günde bir defa ilgili uç düğümlere iletilir.

Akıllı yayılma faktörü ataması yöntemi, geliştirilen simülasyon aracına entegre edilmiştir. Geliştirilen araç öncelikle uç düğümlere rastgele olarak yayılma faktörü atar ve ağı simüle eder. Daha sonra biriktirilen haberleşme kayıtları kullanılarak akıllı yayılma faktörü ataması tahmin modeli oluşturulur ve hesaplanan yeni yayılma faktörleri ile aynı ağ yeniden simüle edilir.

Simülasyon sonuçları göstermektedir ki, akıllı yayılma faktörü ataması yöntemi, rastgele yayılma faktörü ataması ve en düşük yayılma faktörü ataması yöntemlerine göre daha iyi ağ paket iletilme oranı ve toplam iletim enerji tüketimi çıktısı vermektedir. KAÖ makine öğrenmesi yöntemi, DVM makine öğrenmesi yöntemine göre daha iyi ağ performansı sağlamaktadır. Simülasyon sonuçlarına göre kullanılan bu iki akıllı yayılma faktörü ataması yöntemi özellikle yoğun uç düğüm dağılımına sahip LoRaWAN ağlarında, umut vadeden ağ performans iyileştirmesi sağlamaktadır.

1. INTRODUCTION

Number of Internet of Things (IoT) applications increased exponentially in last few years [8]. Total IoT industry is expected to generate 4.3 trillion dollar revenue by 2024 [9]. Recent developments on low power wide area network (LPWAN) technologies has great impact on growth of number of IoT applications such as smart city, health care, asset monitoring, transportation, agriculture and logistic applications. LPWAN technologies address some of the well-known wireless communication challenges. Key challenges of LPWAN technologies are:

- **Communication range:** LPWAN end devices are expected to communicate over tens of kilometers. LPWAN radio technology should be easy to decode and resilient to interference.
- **Device battery life:** Almost all of the LPWAN end devices run on batteries. End nodes should not be power hungry. End nodes are expected to achieve up to 10 years of battery life with a single AA battery [10].
- **Scalability:** LPWAN technologies should support densely populated wireless networks. Network performance should not be affected dramatically as number of end devices increases.
- **Device cost:** Radio chip of the end nodes should be as low as possible (under $10). End devices should not require any expensive calculation or processing units. If any subscription cost exists, then it should be low too.

Traditional wireless communication methods such as cellular networks (e.g., 2G, 3G, LTE) and short-range communication technologies (e.g., NFC, Bluetooth, WiFi, Zigbee) cannot provide low power and long range at the same time [11]. Cellular networks can provide long range and high data rate, but they are complex and consume too much power. They are optimized for low latency communication such as voice and data, however, most of the IoT applications do not require high data rate or low latency.

Short-range communication methods can provide relatively low power consumption, but their range is limited to a few hundred meters at best [10]. LPWAN technologies fill the technology gap between short range and cellular technologies by providing low power and long-range communication. The technology gap is visualized in Figure 1.1. LPWAN technologies basically sacrifice data rate and latency to provide low power consumption. Some of the key features and weakness of LPWAN applications are: long range (a few to tens of kilometers), low power (up to ten years battery time), low data rate (in orders of tens of kbps), low cost and high latency (in orders of seconds or minutes). Characteristics of LPWAN technologies are visualized in Figure 1.2.

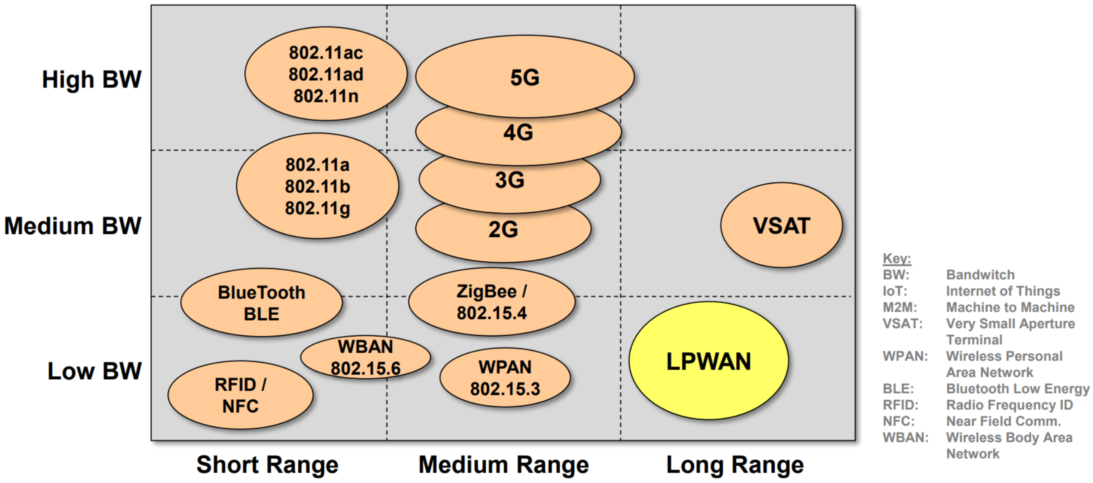


Figure 1.1 : Target of LPWAN technologies [4].

There are several emerging and competing LPWAN technologies. LoRa, Sigfox, NB-IoT and LTE-M are commonly used, well-known LPWAN technologies [10]. LoRa and Sigfox use license free ISM frequency bands while NB-IoT and LTE-M use licensed frequency bands which brings extra cost [10]. Both LoRa and Sigfox are known for ultra-low power consumption and resilience to interference. While NB-IoT and LTE-M are promoted for higher data rate. LoRa has open standard MAC protocol called LoRaWAN. LoRaWAN and Sigfox MAC protocols are based on pure ALOHA medium access [12]. LoRaWAN networks can be deployed as a private network like WiFi. However, Sigfox and NB-IoT are only available with operator contract [10]. Number of messages that Sigfox end device can send in a day is limited to 140 packets for uplink and just 4 packets for downlink [13]. Also, Sigfox packet payload size is limited to 12 bytes for uplink and 8 bytes for downlink. However, LoRa supports up to 243 bytes payload size and NB-IoT supports up to 1600 bytes payload size. Sigfox maximum data rate is 100 bps, on the other hand maximum data rate for LoRa and NB-IoT are 50 kbps and 200 kbps respectively [10].

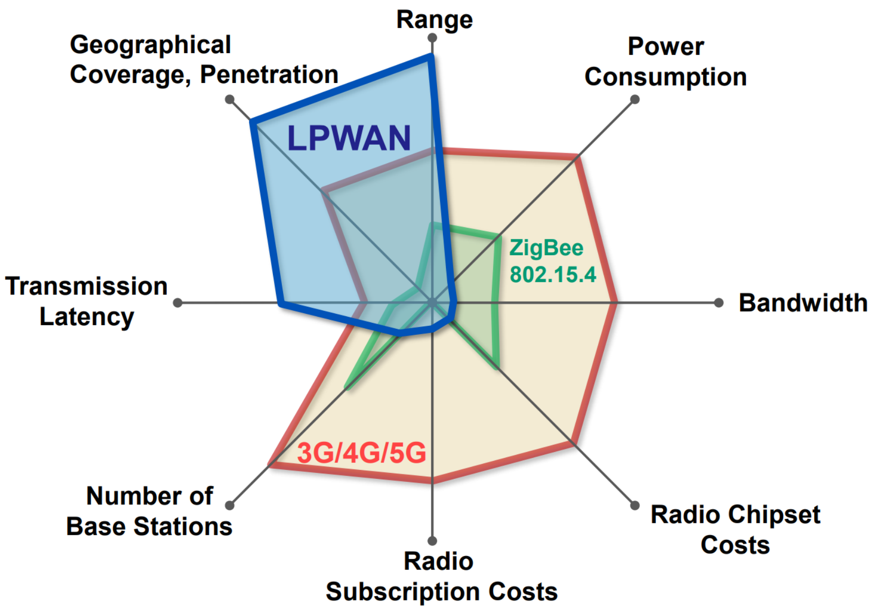


Figure 1.2 : Characteristics of LPWAN technologies [4].

LoRa can adjust data rate by spreading symbols within a fixed channel bandwidth. This enables tradeoff between receive sensitivity and air time of transmission [14]. LoRa supports 6 different spreading factor (SF) options. Simultaneous same spreading factor transmissions are prone to collision, however, different spreading factor transmissions in the same channel are orthogonal to each other. Thus, spreading factor assignment is crucial for overall network performance.

1.1 Contribution

In this work, a LoRa discrete event simulator is developed from scratch to study the performance of various LoRa spreading factor assignment strategies. The tool is capable of simulating multiple node and multiple gateway LoRaWAN networks. The simulation tool can be fed with input parameters such as number of gateways, number of nodes, network radius, packet size, packet generation rate and with these inputs the tool produces simulation results such as total number of generated packets, number of successfully received packets, number of interfered packets, number of under sensitivity packets, network packet delivery ratio percentage, network throughput, total transmit energy consumption.

In this work, first, spreading factor assignment challenge for both single gateway and multiple gateway networks are described. It is shown how spreading factor assignment effects collisions and network performance. Then, an avoidance mechanism is proposed to decrease number of collisions. A machine learning based spreading factor assignment approach is proposed. Support Vector Machine (SVM) and Decision Tree Classifier (DTC) machine learning methods are employed and the introduced schemes are called as smart spreading factor assignment schemes. Smart spreading factor schemes are integrated into the simulation tool. The performance of the smart schemes is compared with the performance of the lowest spreading factor assignment scheme. It is shown that the proposed smart spreading factor assignment schemes give promising results.

1.2 Organization of the Thesis

This thesis is organized as follows:

Chapter 2 provides background information about LoRa and LoRaWAN. First, LoRa spreading factor and spreading factor assignment issue is covered. Then, LoRaWAN architecture and LoRaWAN network entities are explained.

Other recent related works are summarized in Chapter 3.

Chapter 4 describes the proposed smart spreading factor assignment techniques. Two machine learning method is utilized to solve spreading factor assignment issue described in Chapter 2.

Simulation environment is described in Chapter 5. First, employed link model and interference model are explained. Then, software structure of the simulation tool is described.

Simulation results are shown and discussed in Chapter 6. Foremost, single gateway network topology simulation results are shown. Then, multiple gateway network topology simulation results alongside with smart spreading factor schemes simulation results are shown.

Finally, Chapter 7 concludes the thesis by giving future directions.

2. LORA

LoRa is a proprietary physical layer radio technology that provides wireless link solution for low power wide area networks developed by Semtech Corporation [15].

2.1 LoRa Modulation

LoRa uses proprietary spread spectrum modulation technique that is the derivative of Chirp Spread Spectrum (CSS). CSS was originally developed for radar applications in the 1940's. A chirp is a sinusoidal signal of which frequency increases over time [16]. Chirp frequency increases linearly and sweeps the entire bandwidth [17]. An example chirp is visualized in Figure 2.1.

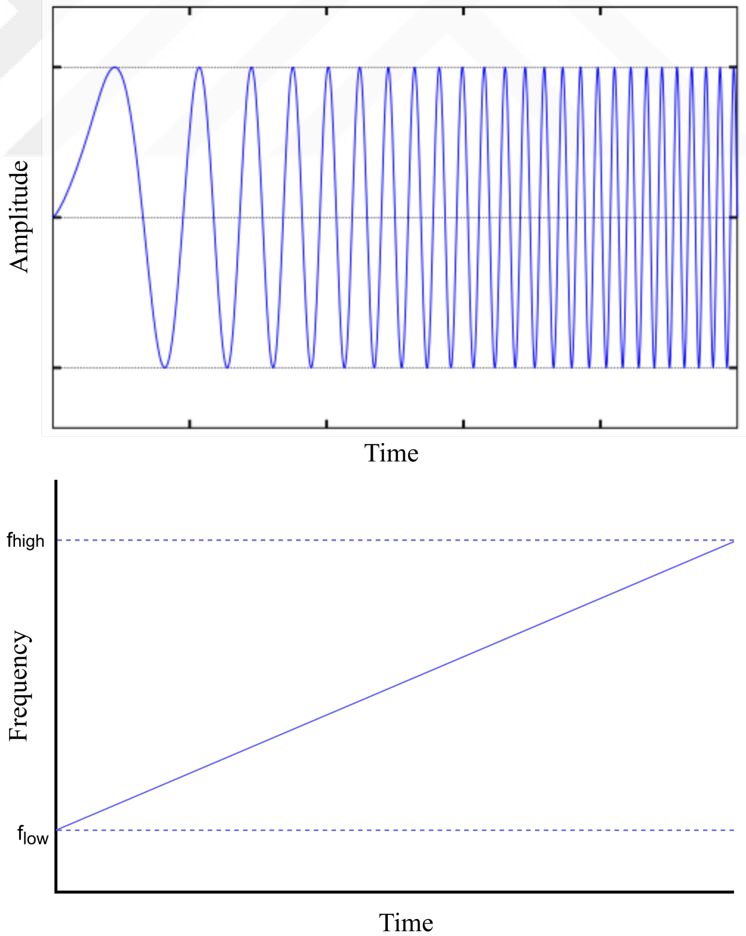


Figure 2.1 : A CSS up-chirp. [5]

Table 2.1 : Bit rate (bps) for various spreading factors (CR = 1).

		SF					
		7	8	9	10	11	12
BW (kHz)	125	5469	3125	1758	977	537	293
	250	10938	6250	3516	1953	1074	586
	500	21875	12500	7031	3906	2148	1172

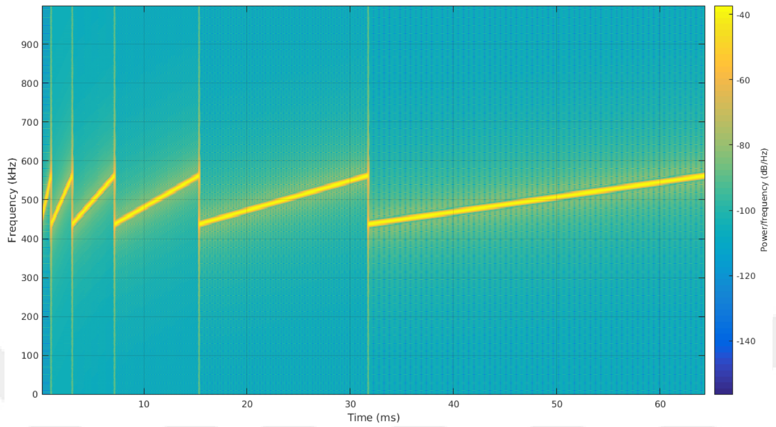


Figure 2.2 : Spectrogram of various spreading factors (SF7 to SF12). [5]

2.1.1 Spreading factor

The ratio between symbol and chirp rate is equal to 2^{SF}. Spreading factor can take values between 7 to 12. Spreading factor also determines data rate of a LoRa transmission [17]. Comparison between various spreading factors can be seen in Figure 2.2. Spectrogram of an example LoRa transmission with various symbols can be found in Figure 2.3. Bit rate of a LoRa transmission can be calculated as:

$$R_b = SF * \frac{\left[\dfrac{4}{4+CR}\right]}{\left[\dfrac{2^{SF}}{BW|_{Hz}}\right]} * 1000\ bps \tag{2.1}$$

Where, R_b is bit rate in bps, SF is spreading factor $SF \in \{7,..,12\}$, CR is error correction code rate $CR \in \{1,..,4\}$ and BW is bandwidth in Hertz [17]. Bit rate for various spreading factors can be found in Table 2.1.

When bandwidth and code rate are constant, as the spreading factor increases, the data rate decreases. Increasing the spreading factor makes the signal more resilient to noise thus increases the transmission range. Increasing the spreading factor also increases the transmission duration which increases the power consumption. Therefore, it is

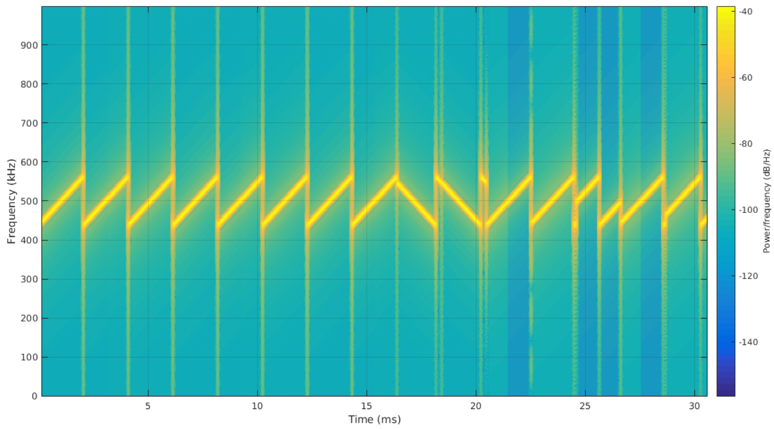


Figure 2.3 : Spectrogram of an example LoRa transmission. [5]

possible to trade between range and power consumption by changing spreading factor. Communication range of various spreading factors is roughly visualized in Figure 2.4.

2.1.2 Spreading factor assignment issue

Simultaneous different spreading factor transmissions are orthogonal to each other up to some extent, which means that a LoRa gateway can simultaneously receive multiple transmissions with different spreading factors. However, simultaneous transmissions with the same spreading factor may not be received by the gateway due to collision. For this reason, spreading factor assignment of nodes is crucial for network performance [18].

In a LoRaWAN network, initially, a node is not aware of how far away it is from a gateway. However, a node can guess the distance from a gateway by observing received signal power of a downlink transmission. If received signal power of a downlink transmission is too high, then the node can decrease its next transmission spreading factor to decrease power consumption. This spreading factor assignment method is called lowest possible spreading factor assignment scheme for the rest of the thesis. Lowest possible spreading factor assignment scheme is commonly used in LoRaWAN deployments. Besides, gateway can request from a node to decrease its spreading factor or transmit power.

In Figure 2.5, a LoRaWAN network deployed with a single gateway is illustrated. Different color rings represent achievable range of different spreading factors from the gateway and different color circles represent selected spreading factor of the nodes. The end devices close to the gateway will fall into the lowest spreading factor (SF7)

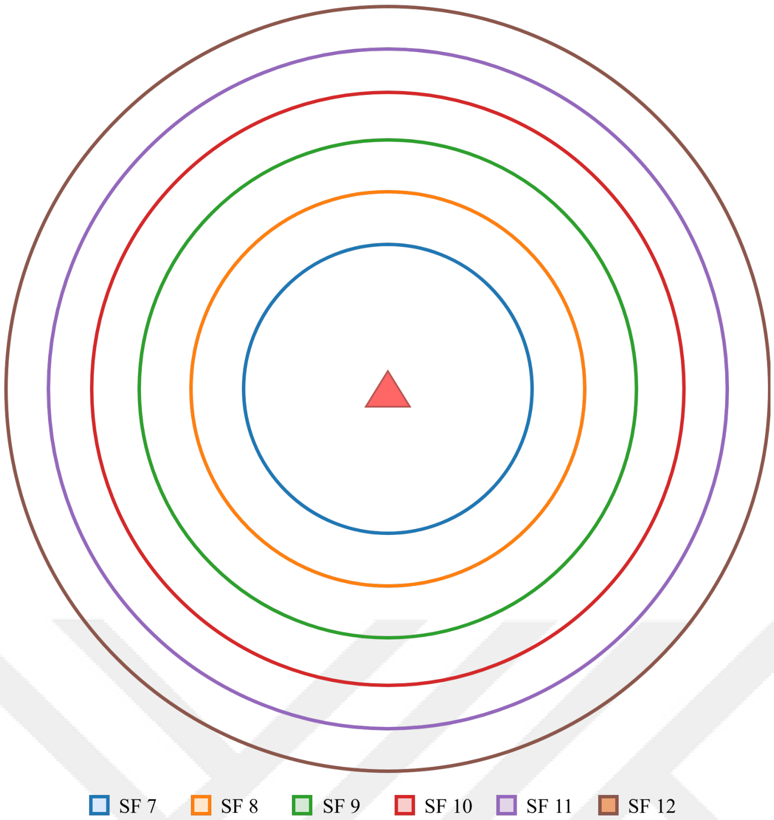


Figure 2.4 : Various spreading factor communication (disproportionate) ranges.

area section. The end devices close to the gateway will probably select the lowest spreading factor most of the time. This causes a lot of collisions between same spreading factor transmissions. Hence the number of collisions will increase as the number of end devices close to the gateway increases.

2.2 LoRaWAN

LoRa has an open standard medium access control (MAC) layer protocol called LoRaWAN which is designed for large scale LoRa networks considering well known LPWAN challenges and their best practice solutions. LoRaWAN provides lightweight but powerful standard for wide range of LoRa IoT applications. LoRaWAN is developed and maintained by LoRa Alliance [19]. LoRa Alliance is an open, non-profit organization dedicated to standardization of LoRaWAN. LoRaWAN provides inter-operability between different LoRa networks. LoRa can be used as a wireless link technology without complying LoRaWAN, however this would break inter-operability between different LoRa networks. LoRaWAN is based on pure

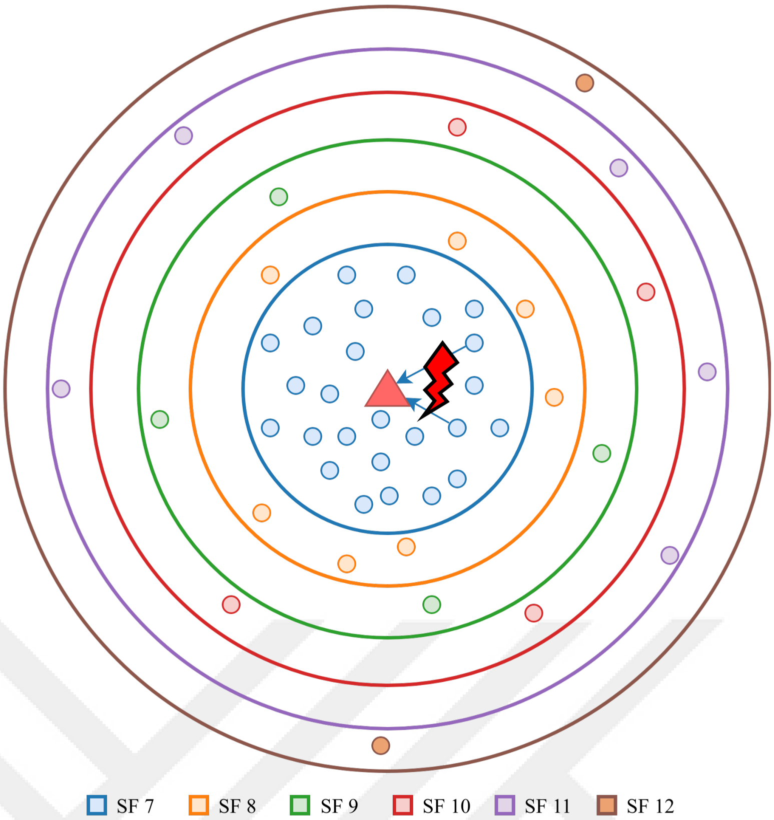


Figure 2.5 : Collision between nodes close to the gateway.

ALOHA medium access, which implies the end nodes do not check whether the channel is free or not before transmission, accepting the possibility of a collision.

2.2.1 Network structure

A typical LoRaWAN network consists of three network entities, which are end node, gateway, and network server.

2.2.1.1 End node

LoRaWAN end node (EN) is a low power embedded device that only communicates with gateways. Single end node can communicate with multiple gateways. This communication architecture is called star of stars network topology. LoRaWAN star of stars network architecture is visualized in Figure 2.7. LoRaWAN standard defines three classes for end devices which are Class A, Class B, and Class C. Different classes provide LPWAN solutions to different applications and deployments.

- **Class A** end nodes generate uplink transmission at any time and only receive a period of time after uplink transmission (pure ALOHA manner). First receive

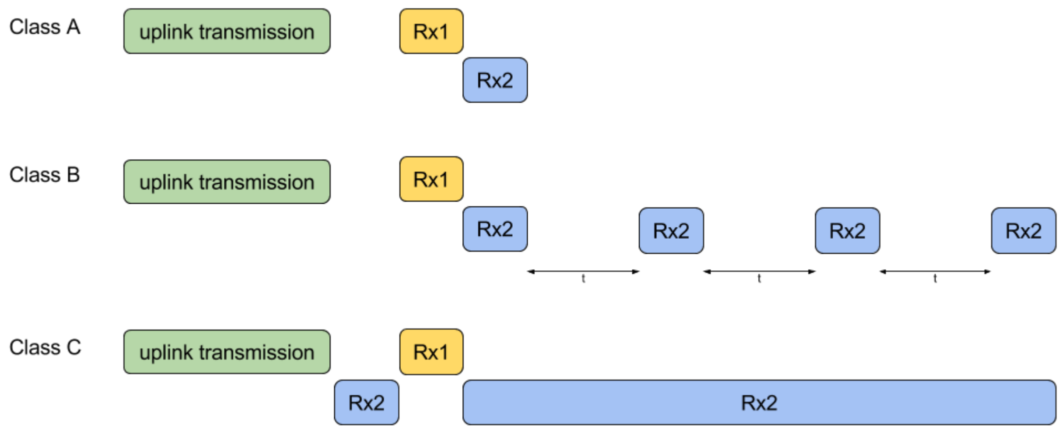


Figure 2.6 : LoRaWAN class A, B and C TX/RX timing diagram [6].

window (Rx1) is opened after an uplink transmission at the same channel with uplink transmission. Then, second receive window (Rx2) is opened at different predefined channel. Transmit and receive window timing diagram can be found in Figure 2.6. Class A offers lowest energy consumption. Class A behavior is the default operation mode of LoRaWAN end devices.

- **Class B** end nodes extend Class A behavior by adding scheduled periodic receive windows (Rx2). Receive window is synchronized using a beacon packet transmitted by gateways.

- **Class C** end nodes extend Class A behavior by keeping receive window (Rx2) open all the time except uplink transmission duration. This provides Class C end nodes with low latency downlink communication, which requires more power consumption. Thus, Class C is suitable for applications which are delay intolerant and end devices which are connected to power grid.

In this thesis, only Class A end devices are considered since Class A behavior leads to the lowest power consumption and Class A is the default operation mode.

2.2.1.2 Gateway

LoRaWAN gateway (GW) is a device that receive/transmit packets coming from/to end nodes. A typical gateway can receive from multiple channels at the same time. Gateways are usually connected to power grid, so power consumption of a gateway is insignificant in most of the deployments.

2.2.1.3 Network server

LoRaWAN network server (NS) is a server that provides MAC layer processing. Network server routes messages from application to end nodes and vice versa. Network server can be used for tweaking end node parameters such as channel, transmit power and spreading factor to increase network performance. NS is the brain of the LoRaWAN network. All process hungry calculations and applications can be run in NS. For example, NS can locate end node locations by triangulation if an uplink transmission is received by at least 3 gateways.

2.2.2 Packet structure

LoRaWAN specification describes the communication protocol and packaging format. LoRaWAN PHY and MAC layer packet structures can be seen in Figure 2.8. PHY layer packet consist of a preamble, a header, a header CRC, a payload and a payload CRC. The MAC payload consists of a frame header, a frame port and a frame payload. The frame header consists of a device address, a frame control field, a frame counter and a frame option field. Frame control field is used for Adaptive Data Rate (ADR) feature. NS is able to ask to an end node to modify spreading factor. If ADR bit in the frame control field is set, then spreading factor of the end node's future transmissions are controlled by the NS. NS provides new spreading factor information in frame options field [7]. The ADR algorithm is not standardized in LoRaWAN protocol. ADR implementation is left to the network operators. If ADR feature is enabled in a network, then the ADR algorithm should select a spreading factor which is high enough to provide a reliable link and low enough to minimize transmission time to avoid collisions.

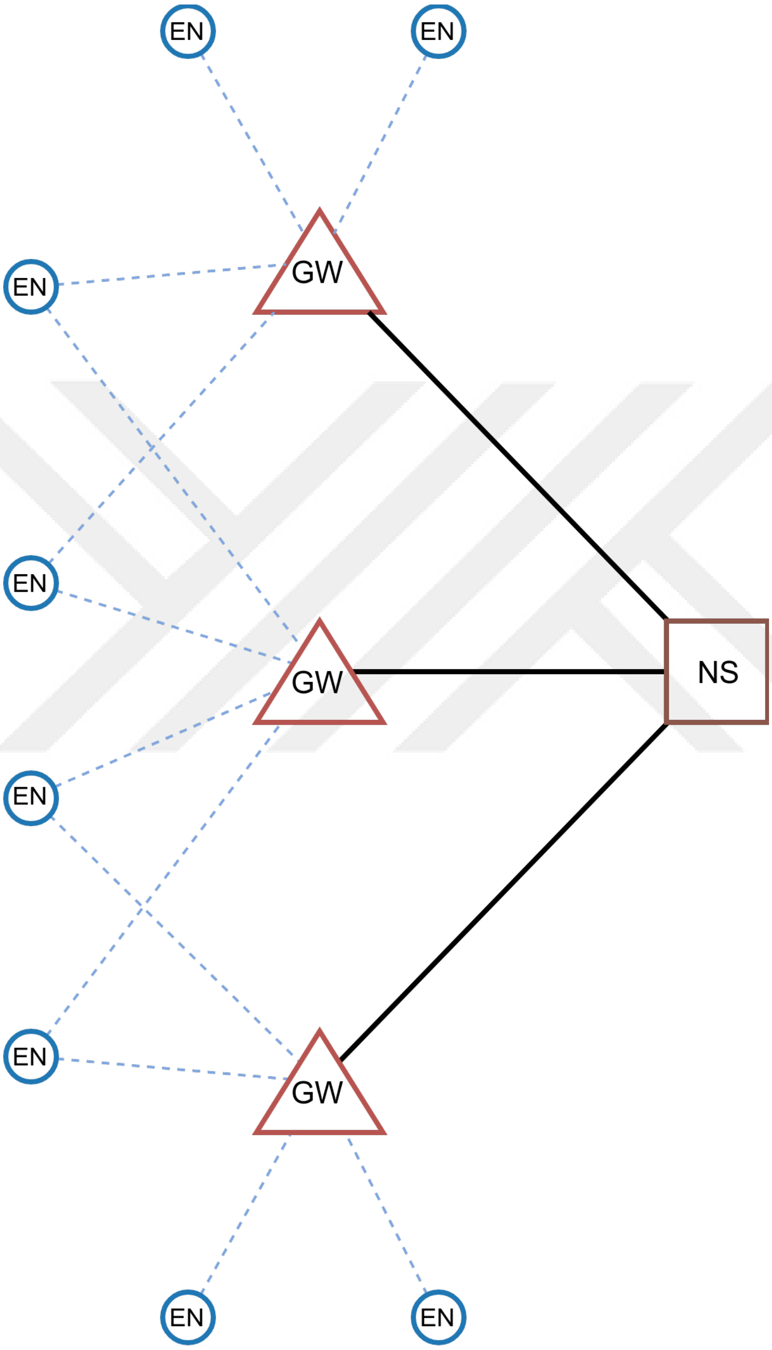


Figure 2.7 : LoRaWAN star of stars network architecture.

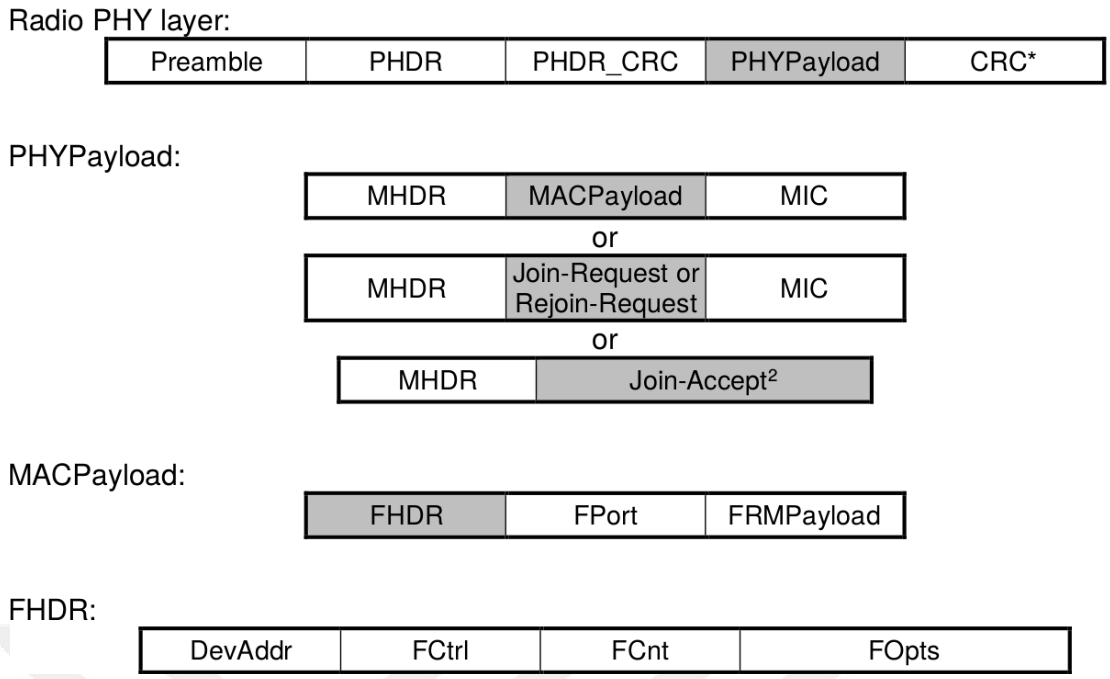


Figure 2.8 : LoRaWAN packet format [7].

2.3 ISM Band Regulations

LoRaWAN operates in license free Industrial, Scientific, and Medical (ISM) bands. It operates in 902-928 MHz in US and 863-870 MHz in Europe [2]. ISM bands are subject to radio transmission time regulations to prevent any device to occupy the channel too long. So, transmissions are required to either adopt listen before talk policy or duty cycle restrictions. Most of the LPWAN technologies adopt duty cycle restrictions since listen before talk policy requires much more power consumption. LoRaWAN obeys duty cycle restrictions. Also, ISM bands are subject to effective isotropic radiated power (EIRP) restrictions. For the rest of this thesis, European ISM band regulations are considered.

- **Duty Cycle Restriction** is a limitation on percentage of air time. For example, 1% duty cycle restriction enforces a device such that single transmission cannot exceed 3.6 seconds and total time on air in one hour cannot exceed 36 seconds. Duty cycle limitations for various European ISM channels can be found in Table 2.2. This limitation is applied for every radio device who don't adopt listen before talk policy. This includes all LoRaWAN end nodes and gateways.

- **Effective Isotropic Radiated Power Restriction** sets maximum allowed transmit power of a transmitter. Maximum allowed effective radiated power for various European ISM channels can be found in Table 2.2.

Table 2.2 : EU863-870 ISM band LoRa channels [1] [2].

Frequency (MHz)	Bandwidth (kHz)	Max Duty Cycle (%)	Max ERP (dBm)
863.0 - 868.0	125	1	14
868.0 - 868.6	125	1	14
868.7 - 869.2	125	0.1	14
869.4 - 869.65	125	10	14
869.7 - 870.0	125	1	14

3. RELATED WORKS

The literature related to the work presented in this thesis has started to grow recently. LPWAN technologies and especially LoRa attracted researchers' attentions lately. Some of these works which studies LoRa/LoRaWAN spreading factor are summarized.

In [20], the authors evaluated the performance of LoRaWAN networks in a smart city scenario. The authors proposed a link measurement and a link performance model for LoRa. The authors also proposed a SINR threshold matrix for modeling LoRa interference between simultaneous but different spreading factor LoRa transmissions. They implemented a LoRa simulator in ns-3 to study scalability and performance of LoRaWAN networks. Their results show that LoRaWAN networks scale well as the number of nodes and gateways increases. They also show that spreading factor assignment has great effect on LoRaWAN network performance.

In [21], another LoRaWAN ns-3 simulator is presented. Authors introduced an error model for determining range as well as interference between multiple simultaneous LoRa transmissions. Their simulator supports LoRaWAN Class A end devices, multiple gateways, both upstream and downstream confirmed messages. Their results show that allocating network parameters such as spreading factor is highly important for the performance of LoRaWAN networks.

In [22], the authors studied the effects of imperfect orthogonality between different LoRa spreading factor transmissions. The authors state that a LoRa transmission can be interfered by a different spreading factor transmission when power of the interfering signal is significantly greater than the reference signal. Their experimental results show that this power difference is around 16 dB. Such a power difference can be seen when an interfering signal is close to a receiver or the sum of interfering signals' energy can create this power difference.

In [23], the authors investigated the impact of the interference caused by simultaneous LoRa transmissions with the same and different spreading factors. They derived aggregated co-SF and inter-SF interference power SIR distributions to capture the coverage distance from the gateway for modeling interference in multiple gateways scenarios. Their results show that transmission among different spreading factors can cause a significant impact in high-density LoRaWAN networks.

In [24], the authors introduced two new algorithms for spreading factor assignment in LoRaWANs. First, algorithm assigns spreading factors based on the total number of connected devices, second algorithm assigns spreading factors by balancing air time of the packets in each spreading factor. Based on the simulation results, it is shown that the proposed algorithms give promising results.

In [25], the authors investigated single gateway LoRaWAN network scalability in terms of the number of end nodes using a simulation model based on real measurements. They measure the impact of two concurrent LoRa transmissions on each other by using physical LoRaWAN end devices and a gateway. Then, they created a simulation model from their measurements. Their results show that LoRaWAN has better scalability than pure ALOHA since a LoRa packet may still go through under collision if the last six symbols of preamble and header of the packet does not collide.

In [26], the authors developed an open-source framework for end-to-end LoRa simulations in OMNeT++. With this tool, the authors investigate adaptive mechanisms to configure the communication parameters of LoRa networks in dense IoT scenarios. The simulation tool is able to evaluate Adaptive Data Rate (ADR) mechanism to dynamically manage link parameters. Their results show that the performance of ADR is severely affected by the network size. Their results also show that network performance can be improved by considering collision probability and parameters of the nodes in the network. They proposed a network-aware approach which link parameters are configured based on the global knowledge of the network.

The adaptive data rate (ADR) algorithm recommended by Semtech Corporation utilizes signal to noise ratio (SNR) of the last 20 transmissions to lower the transmit power and spreading factor while ensuring successful transmissions [27]. The recommended algorithm uses a predefined spreading factor and corresponding required SNR table. The algorithm considers maximum SNR of the last 20 transmissions. The algorithm increases transmit power or decrease data rate in case of low SNR and the opposite in case of high SNR. If SNR is low, then transmit power is incremented by 3 dB until it reaches the maximum transmit power. If SNR is high, then spreading factor is decreased until it reaches the lowest SF. If SNR is still high, then transmit power is decreased by 3 dB until the minimum transmit power (2 dB) is reached. The algorithm basically selects the lowest possible spreading factor and transmit power to maintain reliable communication between end node and gateway. It can be said that the adaptive data rate algorithm recommended by Semtech Corporation is current state of the art implementation. The Things Network (TTN), one of the biggest LoRaWAN networks in the world, is utilizing this algorithm for their deployed networks [28].

4. SMART SPREADING FACTOR TECHNIQUE

The collision problem illustrated in Figure 2.5 is solved by forcing some of the close nodes to select higher spreading factors even though they are able to communicate with lower spreading factors. This has the potential to prevent collisions due to the orthogonality of the spreading factors as shown in Figure 4.1. Higher spreading factor assigned nodes are drawn with bold circle border in Figure 4.1. However, the distribution of spreading factor among nodes becomes an important problem. Increasing a node's spreading factor should be done carefully since higher spreading factor means longer air time and longer air time means increasing the probability of collisions with other high spreading factor transmissions. In multiple gateways scenarios, this approach may increase the collisions with the nodes in other gateways' range. Thus, extra care should be taken for nodes in the intersection area of the gateways illustrated in Figure 4.2.

It is difficult to propose a single spreading factor assignment rule for every possible LoRaWAN topology since every network is different and optimizing their nodes' spreading factors requires different rules. For this reason, machine learning based spreading factor assignment approach is proposed to decrease the collisions for the same spreading factor transmissions. This technique starts by learning the transmission behavior of the nodes in a network. NS can keep track of successful uplink transmissions and their spreading factors. NS can also keep track of some of the collided transmissions if header part of the packet is not interfered at the gateway. However, NS cannot keep track of transmissions with lower receive power than the sensitivity of the gateway. Using those obtained information, NS can train a classifier to predict future transmission result for a specific node and a specific spreading factor. Using this prediction model NS can assign spreading factors to nodes considering the collision probability. NS can modify spreading factor of the node using existing ADR mechanism. Thus, this technique does not require any modification on LoRaWAN protocol. It is reasonable to assume that every end node makes at least one uplink

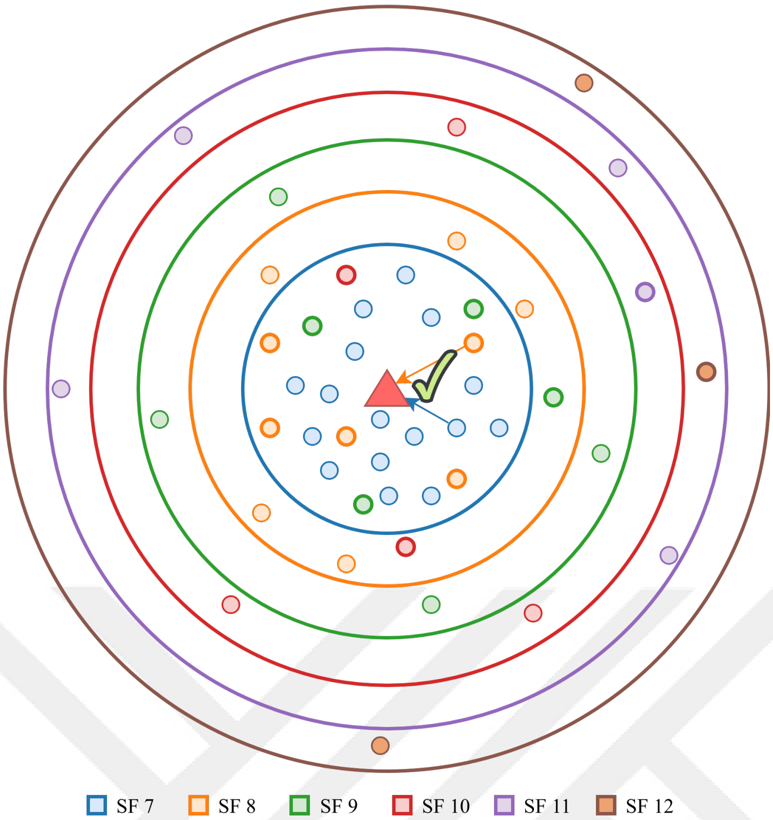


Figure 4.1 : Collision avoidance by using higher spreading factor for nodes close to the gateway.

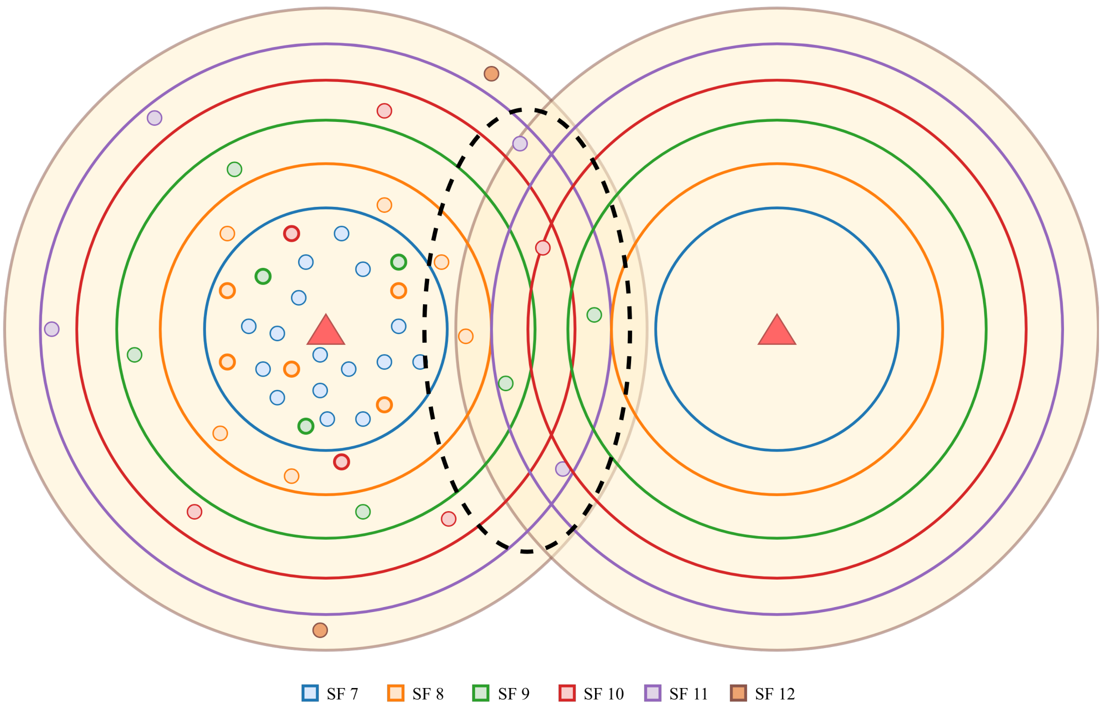


Figure 4.2 : Collision avoidance for intersecting gateways.

transmission in a day, so there is at least one open receive window every day to inform end nodes about new spreading factor.

In this work, DTC and SVM [29] schemes are employed to predict the transmission results. Class weights are balanced according to sample distributions for both methods. For DTC, Gini impurity criteria is used to measure the quality of splits. For SVM, penalty parameter is set to 1, degree is set to 3 and RBF kernel is used.

It is possible to generate mass amount of LoRaWAN transmission logs for various topologies using our simulator. Training dataset size is directly proportional to simulation duration. Thus, increasing the simulation duration, improves the prediction accuracy up to some extent. In real world deployments, NS can keep track of transmissions and it can create a classifier daily basis. Then, gateways can request from nodes to use suggested spreading factors.

There are three features in the dataset generated by the simulation tool. Features of the dataset are: X coordinate of the transmission source, Y coordinate of the transmission source and spreading factor of the transmission. X and Y coordinate feature values are continuous numbers. In this technique, it is assumed that node locations are known to NS by triangulation. Spreading factor feature values are integer numbers between 7 to 12. Class label of the dataset is the result of the transmission. Class label values are successful, interfered and under sensitivity. Example dataset can be found in Table 4.1. DTC and SVM prediction schemes are integrated into the simulation tool to study smart spreading factor assignment schemes. A classifier is trained from generated dataset and this classifier is used for selecting optimum spreading factor for the nodes in the network.

The tool first runs a simulation with random spreading factor scheme. After random spreading factor scheme simulation completed, transmission logs are combined into three feature columns and one class label column to create the training dataset. Then, the dataset is fed to Python scikit-learn DTC or SVM classifiers for training phase. After the classifier model is built, second simulation is run with the trained classifier. The tool selects optimum spreading factor for transmissions considering prediction of the transmissions. For every transmission, the classifier predicts the transmission result for the lowest possible spreading factor. If the transmission result is predicted as

Table 4.1 : Sample section of a transmission dataset.

X Coordinate	Y Coordinate	Spreading Factor	Result
-2033.1	713.0	8	successful
995.5	-2148.6	7	under sensitivity
-3738.6	-4665.3	12	interfered
-1268.9	1348.2	9	successful
-427.1	593.7	9	successful
-193.4	-4047.5	11	interfered
-2729.1	-3663.3	10	interfered
637.3	-715.3	7	successful
3843.7	1996.3	8	successful
2853.6	-2696.0	7	successful
...	...	...	...

interfered, then the tool increases the spreading factor and predicts a new transmission result. If the new transmission result is classified as successful, then simulator continues to execute with selected spreading factor. If no spreading factor transmission result is predicted as successful, then the tool selects the lowest possible spreading factor and continue to execute.

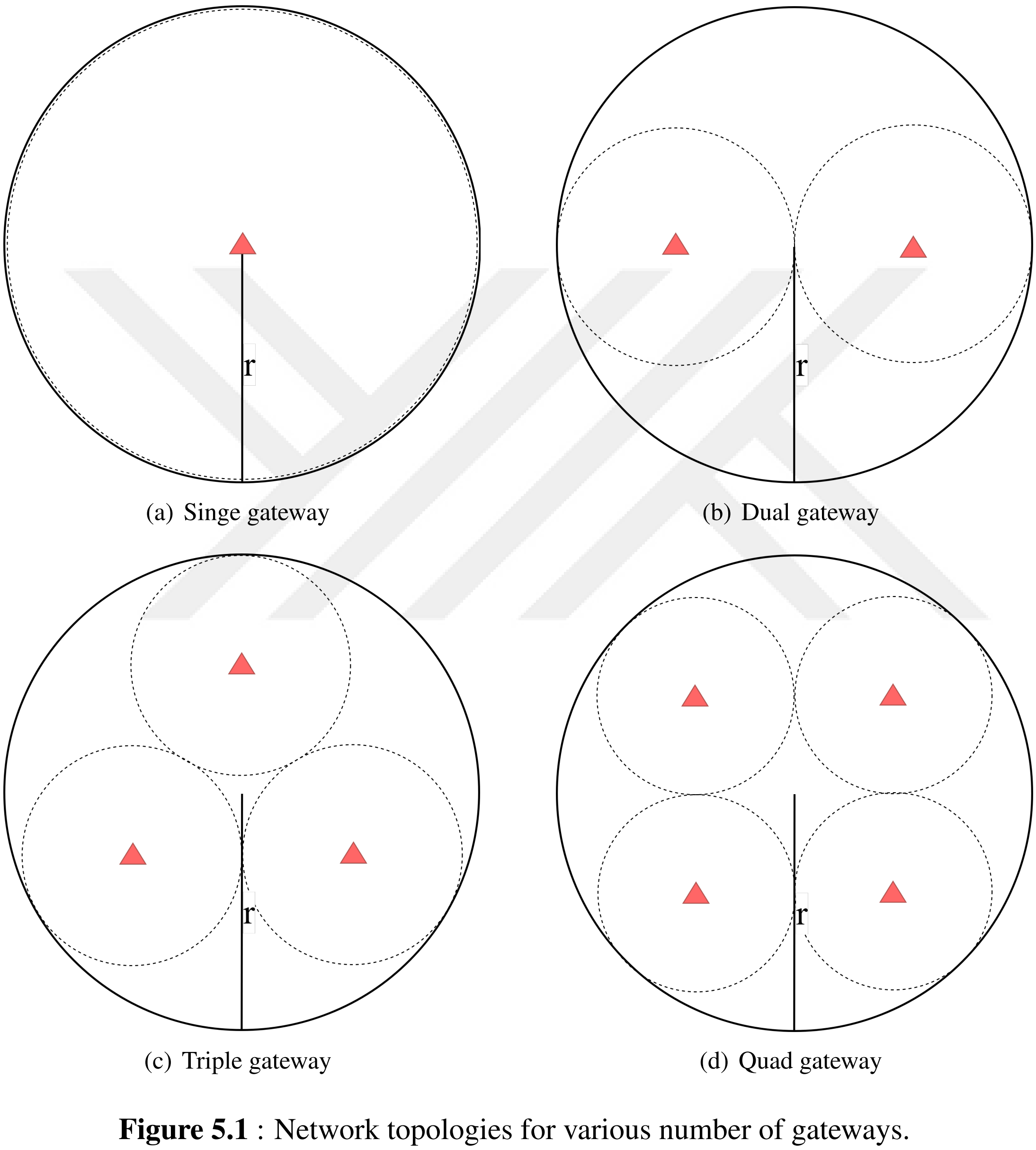
5. SIMULATION ENVIRONMENT

A discrete event simulator is developed in Python to study the effects of various spreading factor strategies in LoRaWANs. LoRaWAN spreading factor simulation tool source code is open source and available at GitHub [30]. Simulation tool supports custom LoRaWAN topologies as well as randomly generated LoRaWAN topologies. Simulator can generate uniformly distributed circular shape network topology with following input parameters: radius in meter (m), number of nodes and number of gateways. Global simulation input parameters are simulation duration in second (s), packet size in Byte (B), packet generation rate in packets per second (pps), packet generation type and spreading factor assignment method. With these inputs, the simulator produces total number of generated packets, number of successfully received packets, number of interfered packets, number of under sensitivity packets, network packet delivery ratio percentage (PDR), network throughput in bits per second (bps) and total transmit energy consumption in Joule (J). Simulator also produces prediction accuracy percentage and confusion matrix for machine learning schemes.

The simulation tool LoRaWAN network gateway placement for various number of gateways can be found in Figure in Figure 5.1. Simulation tool first places gateways by number of gateways input. The tool homogeneously distributes gateways over network by keeping them apart as possible. Then, the tool randomly places end nodes in network radius.

The simulation tool only covers the LoRaWAN Class A devices since Class A behavior is the default operation mode and Class A behavior leads to the lowest power consumption. Transmissions are always initiated by end nodes in pure ALOHA manner. Nodes generate a new packet for given packet rate parameter. Packets can be generated according to Poisson interval or periodic interval.

Timing diagram for periodic traffic generation is shown in Figure 5.2. Initial transmission intervals are always generated by Poisson as it can be seen in first transmission in Figure 5.2. This prevents continuous collisions between periodic



(a) Singe gateway (b) Dual gateway (c) Triple gateway (d) Quad gateway

Figure 5.1 : Network topologies for various number of gateways.

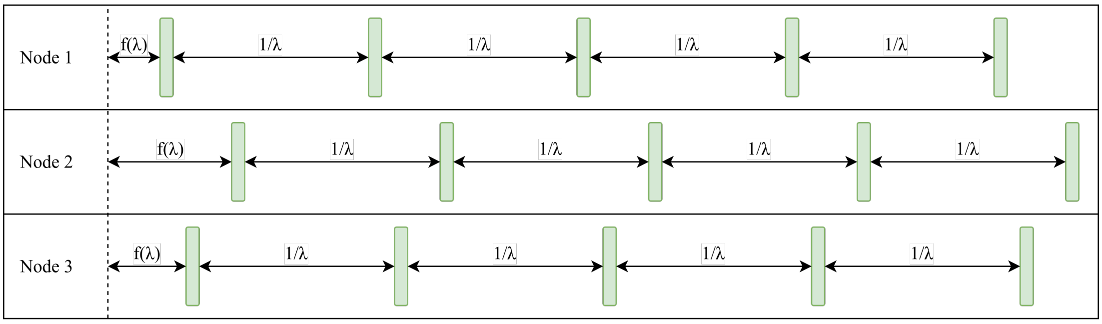


Figure 5.2 : Periodic traffic generation timing diagram.

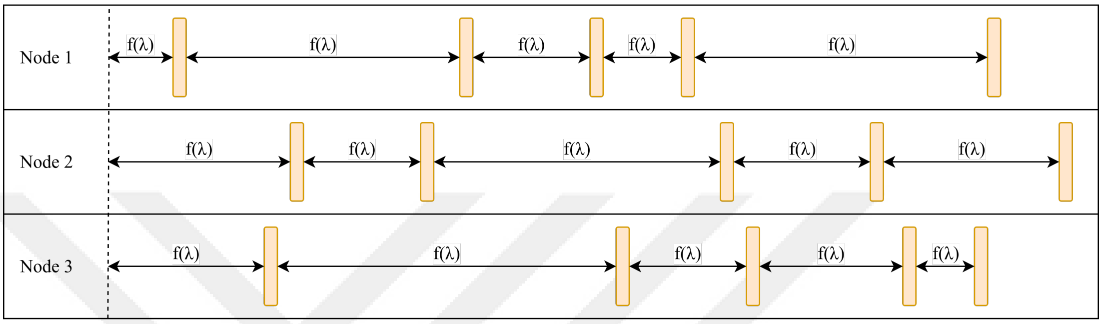


Figure 5.3 : Poisson traffic generation timing diagram.

transmissions by adding various time gap between different nodes. $1/\lambda$ represents periodic traffic generation interval between two transmission for traffic rate input (λ). $f(\lambda)$ represents Poisson traffic generation interval for traffic rate input [31].

Timing diagram for Poisson traffic generation is shown in Figure 5.3. All transmission intervals are generated according to Poisson for given packet rate parameter.

Downlink transmissions are not considered. Downlink transmissions are rare in real world deployments since ISM band regulations dictate duty cycle transmission limit for all devices including gateways.

5.1 Link Model Employed

Link quality of a wireless system can be expressed by the metric of link budget. Link budget is a measure of all gains and losses from transmitter device to receiver device. Link budget of a wireless link can be calculated as [17]:

$$P_{RX}^{dBm} = P_{TX}^{dBm} + G_{SYS}^{dB} - L_{SYS}^{dB} - L_{PATH}^{dB} \tag{5.1}$$

Table 5.1 : Gateway sensitivity for various spreading factors [3].

		SF					
		7	8	9	10	11	12
BW (kHz)	125	-123	-126	-129	-132	-133	-136
	250	-120	-123	-125	-128	-130	-133
	500	-116	-119	-122	-125	-128	-130

Where, P_{RX}^{dBm} is the expected receive power at the receiver. P_{TX}^{dBm} is the transmit power of the transmitter. G_{SYS}^{dB} is the system gains such as transmitter and receiver antenna gains. L_{SYS}^{dB} is the system losses such as transmitter and receiver line, circuit, antenna losses. L_{PATH}^{dB} is the propagation path loss between transmitter and receiver antennas in open space. In the simulator, it is assumed that sum of system gains G_{SYS}^{dB} and system losses L_{SYS}^{dB} is +7 dB.

Maximum transmit power for European ISM band LoRa default channels can be found in Table 2.2. In the simulator, it is assumed that nodes always select maximum allowed transmit power, which is 14 dBm. Different channel transmissions are independent from each other. However, this study focuses on spreading factor orthogonality. Thus, only single channel transmissions are utilized in the simulator.

Receive sensitivity of a LoRa gateway for various spreading factors and bandwidths in dBm unit can be found in Table 5.1. In the simulator, 125 kHz bandwidth receive sensitivity values are used.

Free space propagation loss is calculated as [32]:

$$\begin{aligned} P_{PATH}^{dB} &= 40(1 - 4 \times 10^{-3} \times h)\log_{10} R|_{km} \\ &\quad - 18\log_{10} h|_{m} + 21\log_{10} f|_{MHz} + 80 \end{aligned} \tag{5.2}$$

Where, h is the gateway altitude and f is the frequency of the signal. In this work, it is assumed that h = 15 m and f = 868 MHz. With these assumptions, propagation loss calculation become [20]:

$$P_{PATH}^{dB} = 120.5 + 37.6\log_{10} R|_{km} \tag{5.3}$$

If the received signal power is higher than the gateway sensitivity, then signal can be decoded by the receiver successfully when there is no interfering transmission.

5.2 Interference Model Employed

In the simulator, interference model described in [20] is adopted. They use SINR threshold matrix for modeling LoRa interference between simultaneous but different spreading factor LoRa transmissions.

In this work, it is assumed that there is no other technology interference in the network except LoRa interference. To exploit imperfect orthogonality of different spreading factor transmission, simulator should calculate the effect of different spreading factor transmissions to each other. In simulator, signal to interference plus noise ratio (SINR) threshold matrix $T_{i,j}$ from [33] is used:

$$T = \begin{bmatrix} 6 & -16 & -18 & -19 & -19 & -20 \\ -24 & 6 & -20 & -22 & -22 & -22 \\ -27 & -27 & 6 & -23 & -25 & -25 \\ -30 & -30 & -30 & 6 & -26 & -28 \\ -33 & -33 & -33 & -33 & 6 & -29 \\ -36 & -36 & -36 & -36 & -36 & 6 \end{bmatrix} \tag{5.4}$$

To decide if a referenced signal is interfered at receiver by an interfering signal, SINR threshold matrix is used. $T_{i,j}$ is SINR margin in dB unit between referenced signal with SF = i and interfering signal with SF = j to correctly decode the referenced signal. If there are more than one interfering signal, referenced signal must satisfy the margin for cumulative sum of all interfering signal received power for each spreading factor [20]. SINR threshold can be calculated as:

$$SINR_{i,j} = \frac{P_{rc,0}}{\sum_{l \in I_j} P_{rc,l}} \tag{5.5}$$

Where $P_{rc,0}$ is received signal power of referenced signal and $P_{rc,l}$ is received signal power of interfering signal for SF = j. If a packet with SF = i satisfies the following condition for every SF = j, then packet is survived from all interferences.

$$SINR_{i,j}^{dB} > T_{i,j} \tag{5.6}$$

To calculate interfering power at receiver, we should consider the case where two transmissions are not perfectly overlapping. To equalize the interfering power at receiver [20]:

$$P_{rc,y}^{interf} = \frac{P_{rc,y}(t_{interf})}{t_x} \tag{5.7}$$

Where t_x is transmission duration of referenced signal. t_{interf} is overlapping duration between referenced signal and interfering signal. Transmission duration of a packet can be calculated by bit rate R_b and packet size PS. Bit rate of a LoRa transmission is already expressed in Equation 2.1. Duration of a LoRa transmission can be calculated as:

$$t_x = \frac{8 \times PS|_B}{R_b|_{bps}} \; s \tag{5.8}$$

5.3 Software

The discrete event simulation tool is developed entirely with Python. The tool is designed to simulate smart spreading factor schemes, thus machine learning libraries are integrated to the tool. Integrating smart spreading factor schemes to the existing LoRa/LoRaWAN simulation tools such as ns-3 would require major changes. Thus, the tool is designed and developed from scratch to make it easier to integrate smart spreading factor schemes.

The tool provides framework for simulating LoRaWAN networks. The tool keeps a transmission queue. Every event in this queue is a transmission. Every transmission has time, spreading factor, source, size, duration, and status fields. Every end node generates an event according to their traffic generation rate parameter and inserts this event to simulation event queue. Initially all events are in pending state. Simulation tool iterates and executes events. Then, marks events status as transmitted, interfered or under sensitivity. After all events are executed, the tool calculates network PDR, throughput and transmit energy consumption.

A command line parser is provided for interacting with the framework, this enables users to play with it without any programming or scripting. Also, an example script is provided to demonstrate usage of the framework. Python scikit-learn machine learning library is utilized for smart spreading factor schemes operations. Also, Python matplotlib library is used to generate figures in this thesis.

Source files of the simulation tool and their primary objectives are described below:

- **main.py**
  Command line interface of the simulator. It parses simulation inputs, executes simulation and reports simulation results.

- **simulation.py**
  Core simulation methods and classes are defined. 'run' method is the core function that executes simulation steps.

- **packet.py**
  LoRa related packet information classes are defined. Methods for calculating transmission duration, receive sensitivity, propagation loss and transmission energy are in this file. Also SNIR matrix and packet status enumeration types resides in this file.

- **node.py**
  End node and gateway related classes are defined. Also, node traffic generator methods are defined.

- **topology.py**
  LoRaWAN network topology information such as node and gateway locations are defined. Also, random topology generator method is defined.

- **location.py**
  Location class that keeps x and y coordinate of nodes or gateways are defined.

- **paper.py**
  An example application code for utilizing LoRa spreading factor simulation Python framework. This example script generates figures and results in this thesis.

Relationships between these classes can be seen in UML class diagram shown in Figure 5.4.

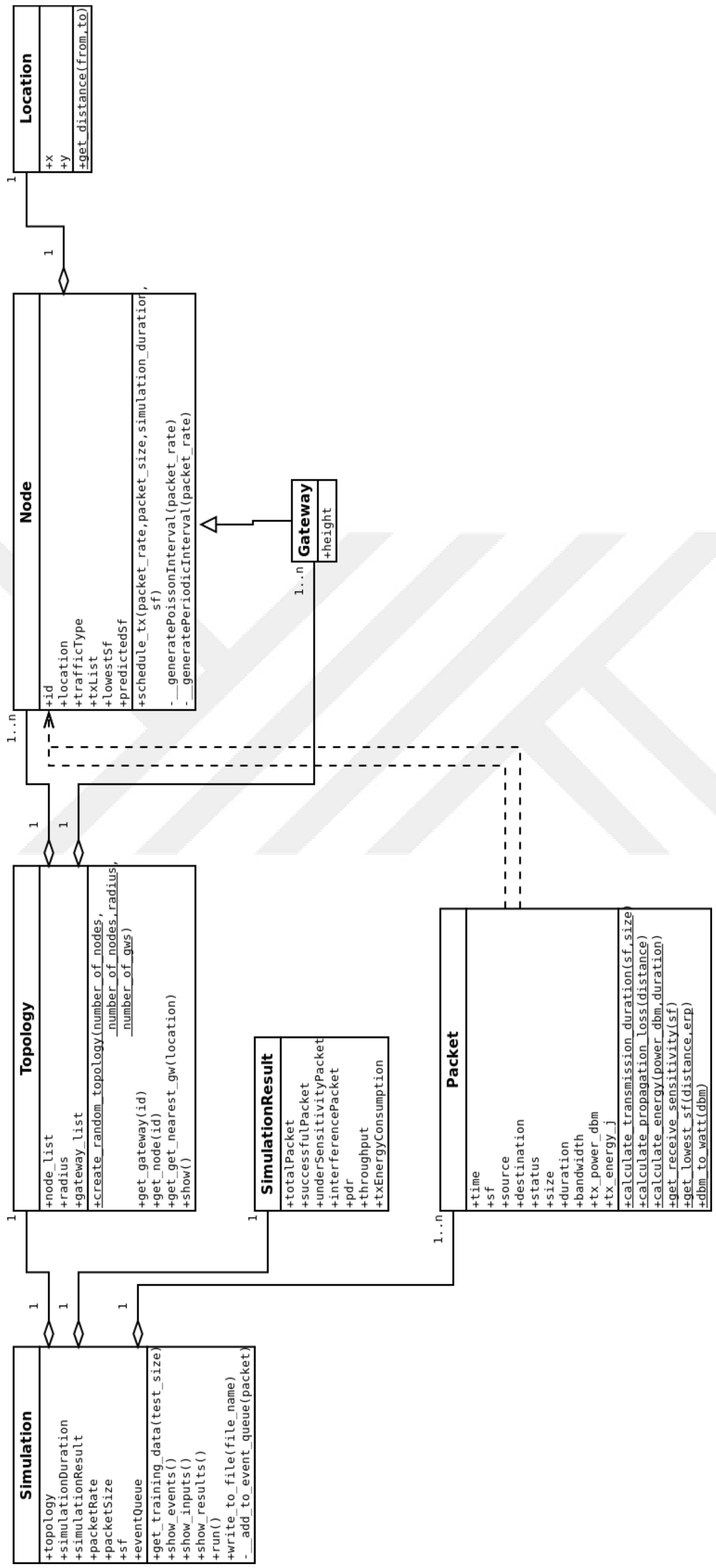


Figure 5.4 : UML class diagram.

5.3.1 Installation

The simulation tool is developed and tested with Python 3.x [34].

The simulation tool requires specific Python modules to run. All of these modules can be installed by "pip" Python package manager. A new Python module can be installed with "pip install" command. Required Python modules for the simulation tool are:

- matplotlib
- numpy
- scipy
- scikit-learn

5.3.2 Command line interface

Command line interface examples of the simulation tool are given below:

- Show help:

  ```
  python3 main.py -h
  ```

- Topology radius in meter:

  ```
  python3 main.py -r 5000
  ```

- Number of gateways:

  ```
  python3 main.py -g 3
  ```

- Number of nodes:

  ```
  python3 main.py -n 300
  ```

- Spreading factor assignment method:

  ```
  python3 main.py -s SF_Lowest
  ```

- Smart spreading factor assignment classifier:

  ```
  python3 main.py -s SF_Smart -c DTC
  ```

- Simulation duration in second:

```
python3 main.py -d 3600
```

- Packet rate in packet per second:

```
python3 main.py -p 0.02
```

- Packet size in byte:

```
python3 main.py -z 80
```

- Proportions of different traffic generator type nodes:

```
python3 main.py -o 0.8 0.2
```

- Random number generator seed:

```
python3 main.py -e 42
```

- Events log path:

```
python3 main.py -l events.txt
```

- Verbose level:

```
python3 main.py -v INFO
```

- Complex example:

```
python3 main.py -r 7000 -g 3 -n 300 -s SF_Lowest -d 3600
```

- Get figures and results in this thesis:

```
python3 paper.py
```

6. SIMULATION RESULTS

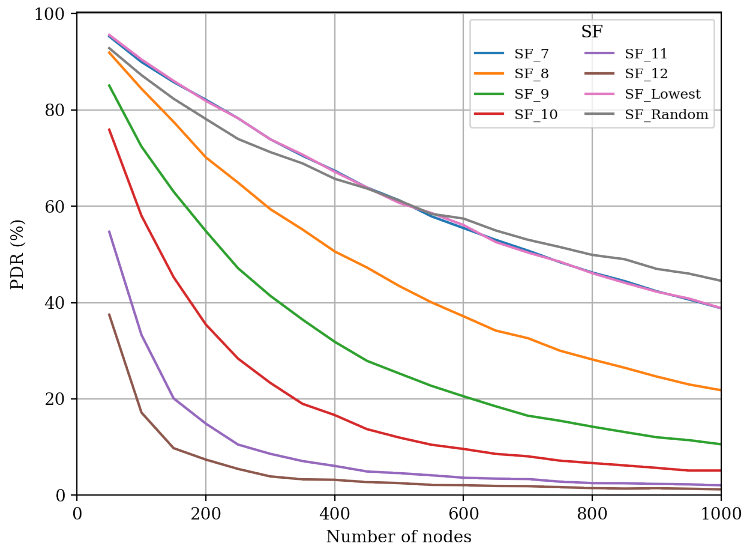
For simulation results in this thesis, global simulation parameters are set as follows: packet size is set to 60 Bytes, simulation duration is set to 3600 seconds and traffic generation type is set as Poisson. First, single gateway and multiple gateway LoRaWAN network simulation results are presented to show correctness of the simulator. Then, smart spreading factor schemes simulation results are presented.

6.1 Single Gateway

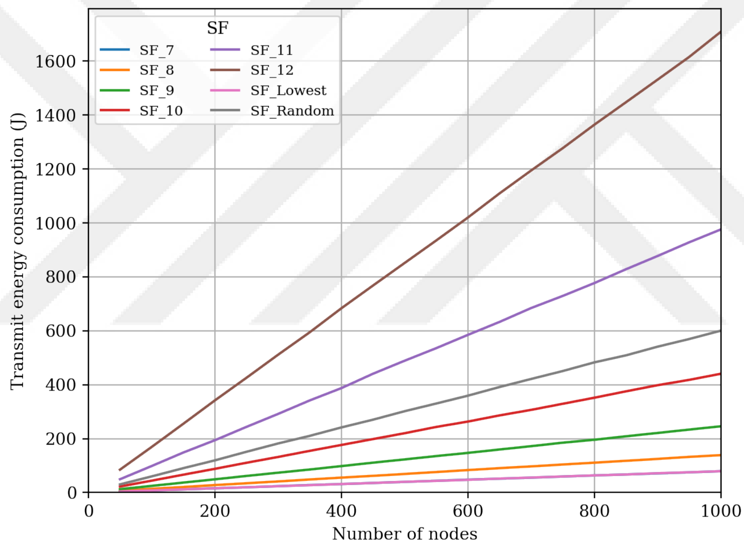
In Figure 6.1, PDR and transmit energy consumption plots of various spreading factor assignment schemes are shown. Randomly generated network topology radius is set to 3000 meters, number of gateways is set to 1 and packet generation rate is set to 0.01 pps. Increasing spreading factors increases air time. This increases the number of collisions thus decreases the PDR. Also, increasing spreading factors increases total transmit energy consumption as expected. High spreading factor schemes gives poor PDR results as the number of nodes increases. Since network topology radius is quite small, all spreading factors can reach to gateway.

In Figure 6.2, PDR and transmit energy consumption plots of various network radii are shown. Number of gateways is set to 1, lowest spreading factor assignment scheme is used and packet generation rate is set to 0.01 pps. Increasing network radius, increases the number of under sensitivity transmissions thus decreases the PDR of network. Nodes select higher spreading factors when they are far away from the gateway. Thus, higher spreading factor leads to longer air time, higher number of collisions and higher total transmit power consumption.

In Figure 6.3, PDR and transmit energy consumption plots of various packet generation rate are shown. Randomly generated topology radius is set to 3000 meters, number of gateways is set to 1 and lowest spreading factor assignment scheme is used. Increasing packet generation rate, increases air time thus increases number of collisions, decreases



(a) PDR



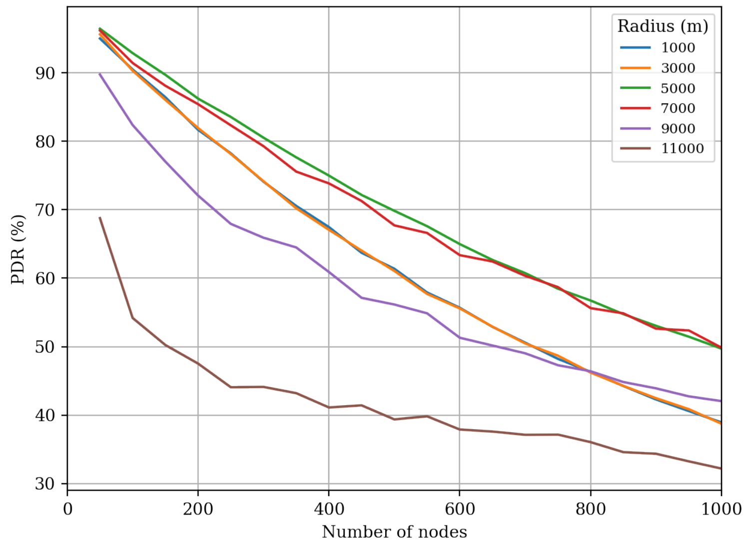
(b) Transmit energy

Figure 6.1 : PDR and transmit energy plots for various spreading factors. (r = 3000 m, GW = 1, R_t = 0.01 pps)

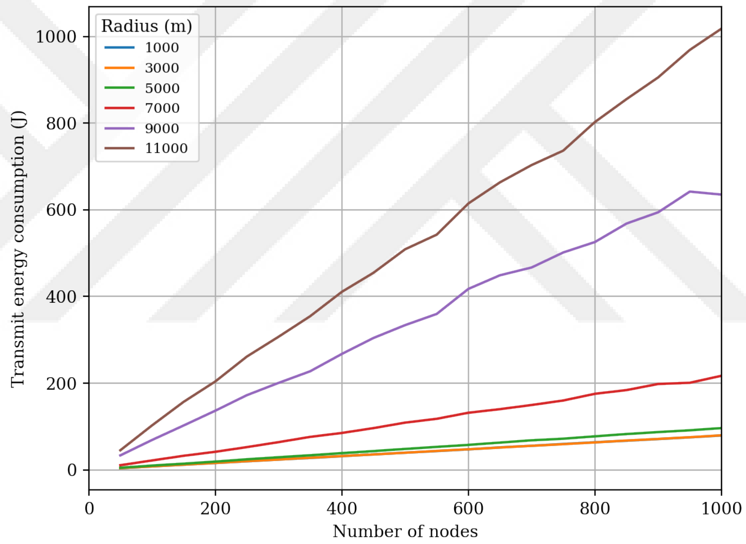
PDR of the network and increases transmit power consumption. Doubling packet generation rate doubles total transmit power consumption.

6.2 Multiple Gateway

In Figure 6.4, PDR and transmit energy consumption plots for various number of gateways are shown. Randomly generated topology radius is set to 3000 meters, the lowest spreading factor assignment scheme is used and packet generation rate is set to 0.01 pps. Increasing number of gateways in constant network topology radius,



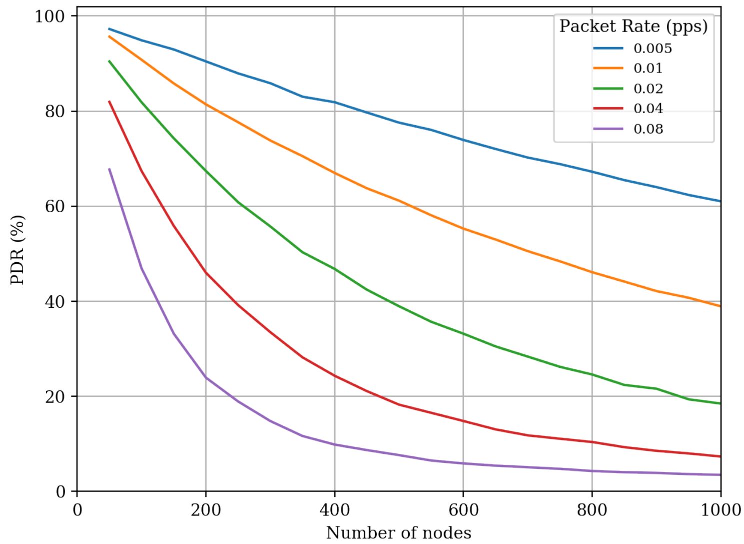
(a) PDR



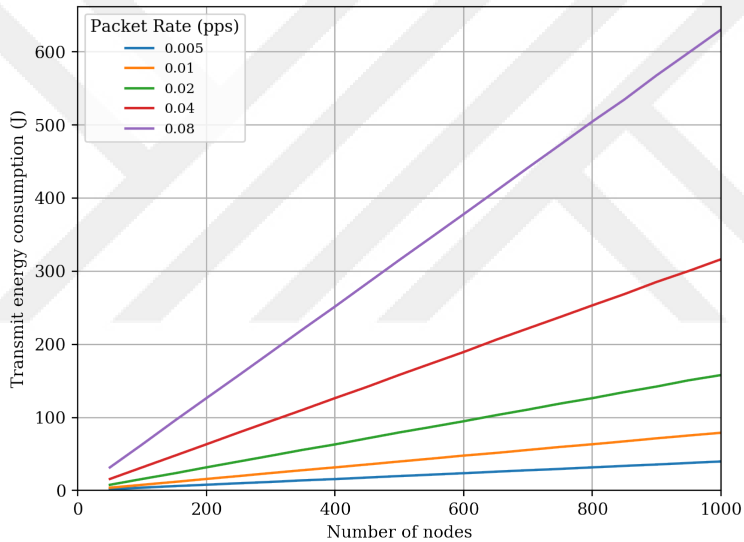
(b) Transmit energy

Figure 6.2 : PDR and transmit energy plots for various network radii. (GW = 1, SF = SF_Lowest, R_t = 0.01 pps)

decreases the distance between nodes and gateways. Thus, decreases the number of collisions at receivers (gateways). Hence, increases the PDR of network. Since topology radius is too small, most of the end nodes selects lowest spreading factor all the time. Thus air time is same and total transmit power consumption is same for all plots.



(a) PDR

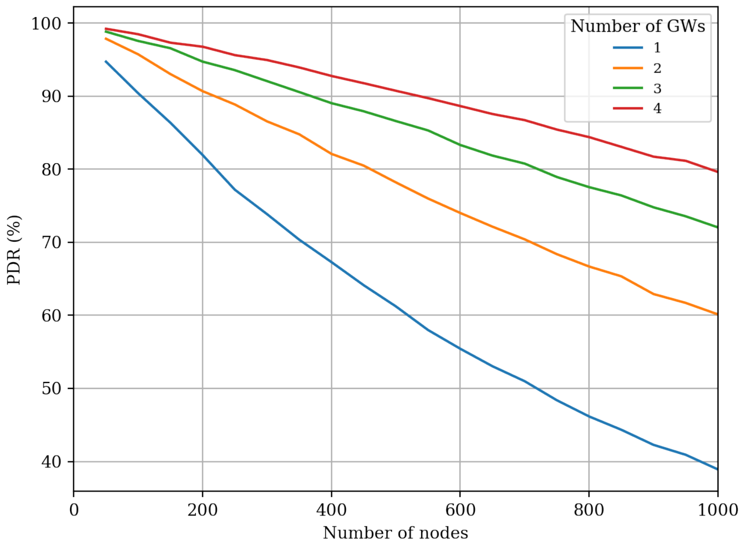


(b) Transmit energy

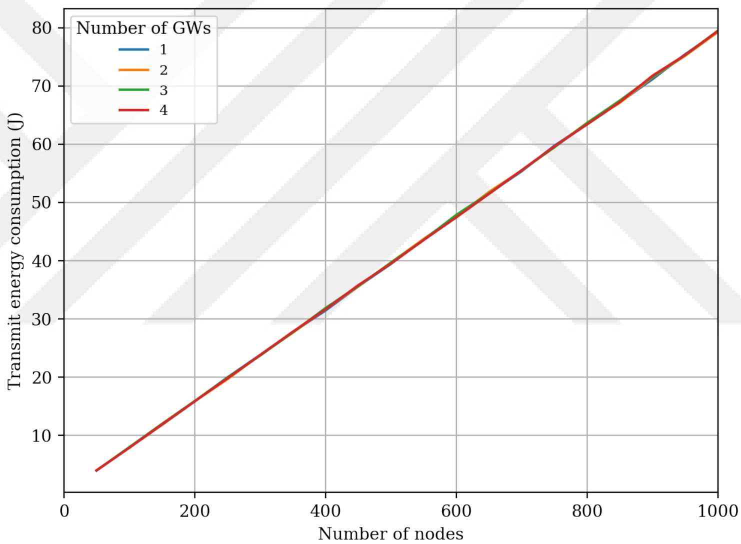
Figure 6.3 : PDR and transmit energy plots for various packet generation rates. (r = 3000 m, GW = 1, SF = SF_Lowest)

6.3 Smart Spreading Factor Schemes

PDR and transmit energy plots of the lowest spreading factor, random spreading factor and the smart prediction schemes are shown in Figure 6.5. Randomly generated network radius is set to 5000 meters, number of gateways is set to 3 and packet generation rate is set to 0.01 pps. Prediction model needs nodes' locations and three gateways are enough to locate position of nodes by triangulation. In Table 6.1, PDR



(a) PDR



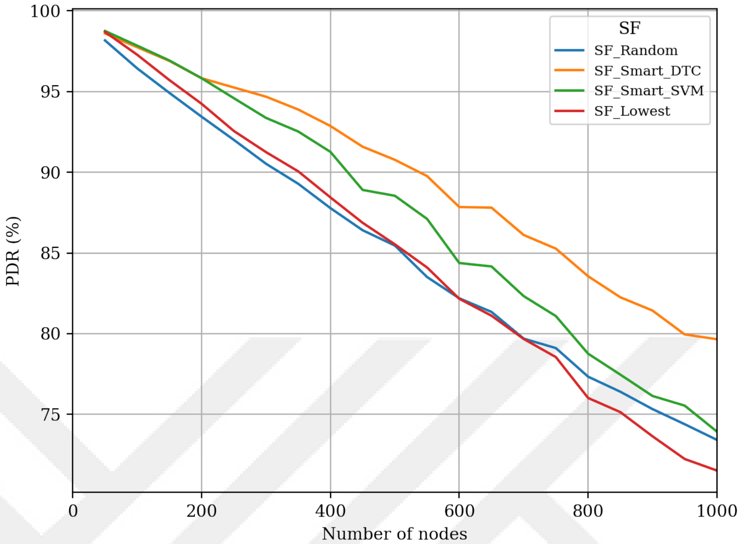
(b) Transmit energy

Figure 6.4 : PDR and transmit energy plots for various number of gateways. (r = 3000 m, SF = SF_Lowest, R_t = 0.01 pps)

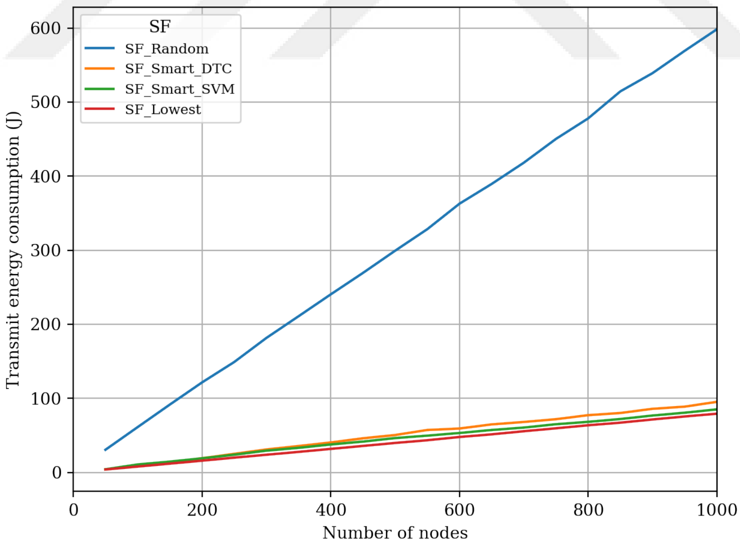
values for various network radii are presented. Both smart SVM and smart DTC schemes give better PDR than lowest spreading factor schemes when number of nodes increases. Increasing number of nodes, increases number of interferences. Smart schemes improve network performance when LoRa interference is high. Moreover, smart schemes give better results when nodes are deployed closer to the gateway, since nodes have margin to increase their spreading factors when they are deployed closer to the gateway. If a node is far away from gateway, then smart schemes cannot increase the spreading factor to avoid interference since the assigned spreading factor is already high. As number of nodes increases, both smart SVM and smart DTC schemes consume slightly more energy than lowest SF scheme, however they give better PDR results in return. Random spreading factor scheme consumes much higher transmit power than the rest.

In Table 6.2, prediction accuracy of smart SVM and smart DTC schemes are presented for various network radii and number of nodes. Prediction accuracy is not directly proportional to network PDR. Correct prediction of an interfered transmission may not increase the PDR but increases the prediction accuracy. Smart DTC gives better network PDR results than smart SVM, even overall smart SVM prediction accuracy is higher. The distribution of different labeled data points in the dataset is strongly related to simulation parameters. If simulation is run with small topology radius and high number of nodes, then number of interfered transmissions labeled data points increases. On the other hand, using a large topology radius in the simulation causes the number of under sensitivity transmission labeled data points to increase. Moderately dense network parameters are chosen to make it closer to real world deployments. With the simulation parameters utilized in this section, simulation tool usually produces imbalanced dataset. Number of under sensitivity transmission labeled data points are less than number of successful transmissions labeled data points. Besides, number of interfered transmissions labeled data points are even less than number of under sensitivity transmission labeled data points. In this case, smart DTC predicts interfered transmission labeled data points more accurately. Correct classification of interfered transmissions yields better network PDR results, thus smart DTC scheme gives better network PDR results.

PDR and transmit energy plots for single gateway, multiple gateway and smart spreading factor schemes with various simulation parameters are given in Appendix A and Appendix B. Also, readers are kindly invited to experiment the simulation tool with the parameters they like [30].



(a) PDR



(b) Transmit energy

Figure 6.5 : PDR and transmit energy plots for lowest and smart spreading factor schemes. (r = 5000 m, GW = 3, R_t = 0.01 pps)

Table 6.1 : PDR for lowest and smart spreading factor schemes. (GW = 3, R_t = 0.01 pps)

			Number of Nodes		
			100	500	1000
r (m)	3000	Lowest	97.8	86.0	72.3
		SVM	98.0	88.2	75.2
		DTC	97.8	89.8	78.7
	5000	Lowest	96.8	85.5	71.2
		SVM	98.0	87.8	74.8
		DTC	97.7	90.2	79.8
	7000	Lowest	97.2	87.5	76.8
		SVM	98.2	88.8	78.6
		DTC	97.8	90.7	81.6
	10000	Lowest	98.2	90.3	81.5
		SVM	98.3	90.3	81.9
		DTC	98.3	90.6	81.9

Table 6.2 : Prediction accuracy for SVM and DTC. (GW = 3, R_t = 0.01 pps))

			Number of Nodes		
			100	500	1000
r (m)	3000	SVM	82.4	70.4	71.7
		DTC	86.0	67.3	70.4
	5000	SVM	79.5	69.0	71.1
		DTC	84.5	67.3	69.5
	7000	SVM	79.5	70.6	71.2
		DTC	84.5	67.7	69.2
	10000	SVM	79.2	74.4	76.1
		DTC	83.8	70.7	74.3

7. CONCLUSION

In this thesis, after a brief introduction about LPWAN technologies, LoRa modulation basics and spreading factor assignment issue is discussed. An open source discrete event simulator is presented, which is developed from scratch to study network performance of LoRaWAN and evaluate various spreading factor assignment schemes. Moreover, it is shown how same spreading factor collisions can be avoided, hence, machine learning based solutions called smart DTC scheme and smart SVM scheme are proposed. Simulation results for the lowest spreading factor assignment scheme and smart spreading factor assignment scheme are presented. Simulation results show that, smart spreading factor assignment scheme can increase network performance for LoRaWAN networks, especially, when the nodes are deployed close to gateways.

LPWAN technologies sacrifice data rate and latency to provide low power and long range communication. Focus of this thesis is on LoRa which is one of the most popular LPWAN technologies. In LoRa modulation, the signal frequency scans the band from end to end within a particular channel. The speed of the scan is called the spreading factor. As the spreading factor increases, the data rate decreases and the power consumption increases, however the communication range increases. LoRa transmissions with different spreading factors within the same channel can communicate without causing interference to each other. Therefore, spreading factor selection of the end nodes significantly affects the number of collisions thus the network performance. It is difficult for end nodes to select the best spreading factor for them, since the end nodes are not aware of the transmissions around them. End nodes select the lowest spreading factor they can to communicate with the gateways to keep the power consumption low, to keep the communication duration short and reduce the likelihood of collisions. However, when other end nodes around them begin to transmit with the same spreading factor, the probability of collisions increases. Same spreading factor transmissions can significantly reduce network performance in densely deployed networks. Assignment of a higher spreading factor may increase the successful packet delivery ratio, even if the end nodes are close to the gateway.

In this study, the effect of spreading factor assignment on network performance is investigated in detail. The factors that increase the number of collisions is evaluated and measures that can be taken to reduce the number of collisions is described. A novel method which utilizes machine learning techniques is proposed to select the most efficient spreading factor. Support Vector Machine and Decision Tree Classifier machine learning methods are used for this new method called smart spreading factor assignment. In this method, gateways first monitor the location of each node, spreading factor of each transmission and result of each transmission. Then, gateways train a transmission prediction model with accumulated data by using SVM or DTC machine learning methods. With this model, the most efficient speeding factor is calculated for subsequent transmissions of the nodes. For each end node, transmission result prediction is calculated one by one from lowest speeding factor to highest spreading factor. The spreading factor of the first transmission projected as successful is selected for the end node. Gateways notify the new spreading factors to the end nodes. The end nodes will begin to use the new spreading factors for their subsequent transmissions. The simulation results show that the smart spreading factor assignment method yields better network packet delivery ratio and total transmission energy consumption than random spreading factor assignment method and lowest spreading factor assignment method. DTC machine learning method gives better packet delivery ratio results than SVM machine learning method. These two smart spreading factor assignment methods provide promising network performance improvements, especially for dense LoRaWAN networks.

As for future work, transmit power optimization can be included to the proposed smart spreading factor schemes. In this thesis, it is assumed that nodes always use maximum transmit power for uplink transmission, however nodes close to gateways can decrease transmit power to save energy. This will make transmissions more vulnerable to interference thus requires extra care. Also other spreading factor assignment techniques described in Chapter 3 can be integrated to simulation tool for comparision. In this thesis, it is also assumed that only LoRaWAN Class A end nodes are present, however Class B and Class C end node behavior can be integrated to check how extra downlink communication effects smart spreading factor schemes. Moreover, other machine learning methods can be investigated for spreading factor assignment enhancement. Reinforcement learning could be a good candidate. Also, other transmission parameters such as node id and transmission time can be included to the proposed smart spreading factor schemes in order to improve prediction performance.

REFERENCES

[1] **European Telecommunications Standards Institute** (2018). Electromagnetic compatibility and Radio spectrum Matters (ERM); Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1 000 MHz frequency range with power levels ranging up to 500 mW EN 300 220-1 V2.4.1, **Technical Report**, European Telecommunications Standards Institute, Sophia Antipolis, France. Retrieved May 3, 2019, from `https://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.02.01_60/en_30022002v030201p.pdf`

[2] **LoRa Alliance Technical Committee Regional Parameters Workgroup** (2018). LoRaWAN 1.1 Regional Parameters v1.1rB, **Technical Report**, LoRa Alliance, San Ramon, California, USA. Retrieved May 3, 2019, from `https://lora-alliance.org/sites/default/files/2018-04/lorawantm_regional_parameters_v1.1rb_-_final.pdf`

[3] **Semtech Corporation Wireless Sensing and Timing Products Division** (2019). SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver Datasheet, **Technical Report**, Semtech Corporation, Camarillo, California, USA. Retrieved May 3, 2019, from `https://www.semtech.com/uploads/documents/DS_SX1276-7-8-9_W_APP_V6.pdf`

[4] **Egli, P.**, (2017), LPWAN - Low Power Wide Area Networks, Retrieved May 3, 2019, from `http://peteregli.net/content/mobile-wireless/LPWAN/LPWAN.pdf`

[5] **Ghoslya, S.**, (2017), LoRa: Symbol Generation, Retrieved May 3, 2019, from `https://www.sghoslya.com/p/lora-is-chirp-spread-spectrum.html`

[6] **Witekio**, (2018), LoRaWan, a dedicated IoT network, Retrieved May 3, 2019, from `https://witekio.com/blog/lorawan-dedicated-iot-network`

[7] **Sornin, N. and Yegin, A.** (2017). LoRaWAN 1.1 Specification, **Technical Report**, LoRa Alliance, Beaverton, Oregon, USA. Retrieved May 3, 2019, from `https://lora-alliance.org/sites/default/files/2018-04/lorawantm_specification_-v1.1.pdf`

[8] **Centenaro, M.**, **Vangelista, L.**, **Zanella, A. and Zorzi, M.** (2016). Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios, *IEEE Wireless Communications*, *23*(5), 60–67.

[9] **Al-Fuqaha, A.**, **Guizani, M.**, **Mohammadi, M.**, **Aledhari, M. and Ayyash, M.** (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, *IEEE Communications Surveys Tutorials*, *17*(4), 2347–2376.

[10] **Raza, U.**, **Kulkarni, P. and Sooriyabandara, M.** (2017). Low Power Wide Area Networks: An Overview, *IEEE Communications Surveys Tutorials*, *19*(2), 855–873.

[11] **Finnegan, J. and Brown, S.** (2018). A Comparative Survey of LPWA Networking, *CoRR*, *abs/1802.04222*.

[12] **Abramson, N.** (1970). THE ALOHA SYSTEM: Another Alternative for Computer Communications, *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, AFIPS '70 (Fall), ACM, New York, NY, USA, pp.281–285, `http://doi.acm.org/10.1145/1478462.1478502`.

[13] **Sigfox** (2017). Sigfox Technical Overview, **Technical Report**, Sigfox, Labège, France. Retrieved May 3, 2019, from `https://storage.sbg1.cloud.ovh.net/v1/AUTH_669d7dfced0b44518cb186841d7cbd75/dev_medias/build_technicalOverview.pdf`

[14] **Georgiou, O. and Raza, U.** (2017). Low Power Wide Area Network Analysis: Can LoRa Scale?, *IEEE Wireless Communications Letters*, *6*(2), 162–165.

[15] **Semtech Corporation**, *<https://www.semtech.com>*, date retrieved May 3, 2019

[16] **Berni, A. and Gregg, W.** (1973). On the Utility of Chirp Modulation for Digital Signaling, *IEEE Transactions on Communications*, *21*(6), 748–751.

[17] **Semtech Corporation Wireless Sensing and Timing Products Division** (2015). LoRa™ Modulation Basics AN1200.22, **Technical Report**, Semtech Corporation, Camarillo, California, USA. Retrieved May 3, 2019, from `https://www.semtech.com/uploads/documents/an1200.22.pdf`

[18] **Adelantado, F.**, **Vilajosana, X.**, **Tuset-Peiro, P.**, **Martinez, B.**, **Melia-Segui, J. and Watteyne, T.** (2017). Understanding the Limits of LoRaWAN, *IEEE Communications Magazine*, *55*(9), 34–40.

[19] **LoRa Alliance**, *<https://lora-alliance.org>*, date retrieved May 3, 2019

[20] **Magrin, D.**, **Centenaro, M. and Vangelista, L.** (2017). Performance evaluation of LoRa networks in a smart city scenario, *2017 IEEE International Conference on Communications (ICC)*, pp.1–7.

[21] **Van den Abeele, F.**, **Haxhibeqiri, J.**, **Moerman, I. and Hoebeke, J.** (2017). Scalability Analysis of Large-Scale LoRaWAN Networks in ns-3, *IEEE Internet of Things Journal*, *4*(6), 2186–2198.

[22] **Croce, D.**, **Gucciardo, M.**, **Mangione, S.**, **Santaromita, G. and Tinnirello, I.** (2018). Impact of LoRa Imperfect Orthogonality: Analysis of Link-Level Performance, *IEEE Communications Letters*, *22*(4), 796–799.

[23] **Mahmood, A.**, **Sisinni, E.**, **Guntupalli, L.**, **Rondón, R.**, **Hassan, S.A. and Gidlund, M.** (2018). Scalability Analysis of a LoRa Network under Imperfect Orthogonality, *IEEE Transactions on Industrial Informatics*, 1–1.

[24] **Cuomo, F.**, **Campo, M.**, **Caponi, A.**, **Bianchi, G.**, **Rossini, G. and Pisani, P.** (2017). EXPLoRa: Extending the performance of LoRa by suitable spreading factor allocations, *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp.1–8.

[25] **Haxhibeqiri, J.**, **Van den Abeele, F.**, **Moerman, I. and Hoebeke, J.** (2017). LoRa Scalability: A Simulation Model Based on Interference Measurements, *Sensors*, *17*(6), `http://www.mdpi.com/1424-8220/17/6/1193`.

[26] **Slabicki, M.**, **Premsankar, G. and Di Francesco, M.** (2018). Adaptive configuration of lora networks for dense IoT deployments, *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*, pp.1–9.

[27] **Semtech Corporation** (2016). LoRaWAN – simple rate adaptation recommended algorithm, **Technical Reportv1.0**, Semtech Corporation, Camarillo, California, USA.

[28] **The Things Network**, (2018), LoRaWAN Adaptive Data Rate, Retrieved May 3, 2019, from `https://www.thethingsnetwork.org/docs/lorawan/adr.html`

[29] **Alpaydin, E.** (2010). *Introduction to Machine Learning*, The MIT Press, 2nd edition.

[30] **Yatagan, T.**, (2019), tugrulyatagan/simlorasf, Retrieved May 3, 2019, from `https://doi.org/10.5281/zenodo.3072925`

[31] **Yates, R.D. and Goodman, D.J.** (2014). *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*, Wiley, 3rd edition.

[32] **European Telecommunications Standards Institute** (2016). Radio Frequency (RF) system scenarios TR 136 942 V13.0.0, **Technical Report**, European Telecommunications Standards Institute, Sophia Antipolis, France. Retrieved May 3, 2019, from `https://www.etsi.org/deliver/etsi_tr/136900_136999/136942/13.00.00_60/tr_136942v130000p.pdf`

[33] **Goursaud, C. and Gorce, J.M.** (2015). Dedicated networks for IoT : PHY / MAC state of the art and challenges, *EAI endorsed transactions*

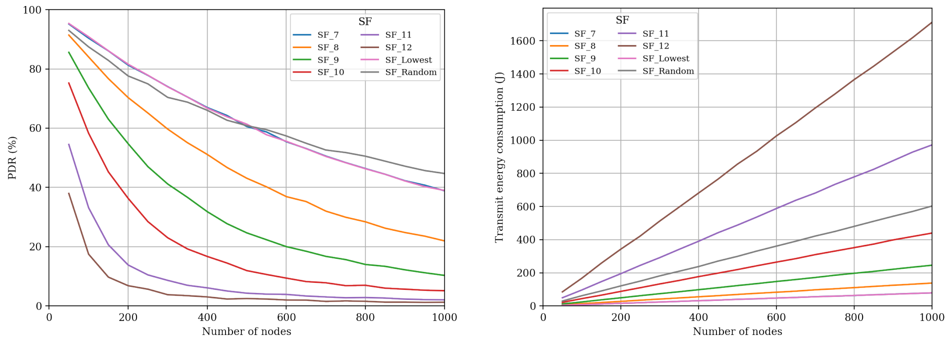
on Internet of Things, https://hal.archives-ouvertes.fr/hal-01231221.

[34] **Python Software Foundation**, <*https://www.python.org*>, date retrieved May 3, 2019

APPENDICES

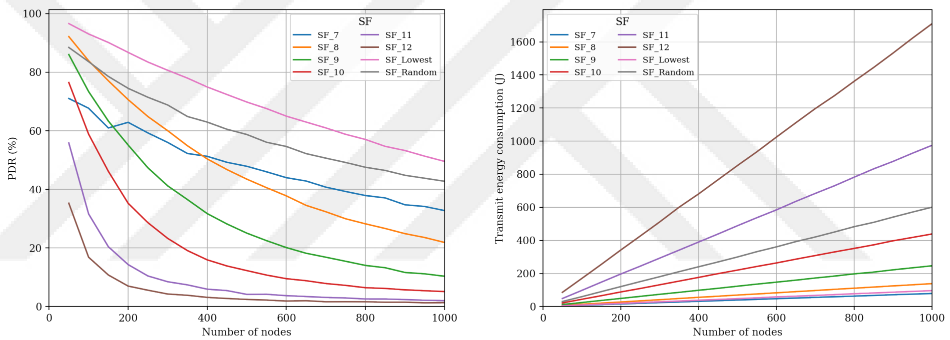
APPENDIX A : PDR and transmit energy plots for various spreading factors, network radii and number of gateways. (R_t = 0.01 pps)
APPENDIX B : PDR and transmit energy plots for various smart spreading factors schemes, network radii and number of gateways. (R_t = 0.01 pps)

APPENDIX A



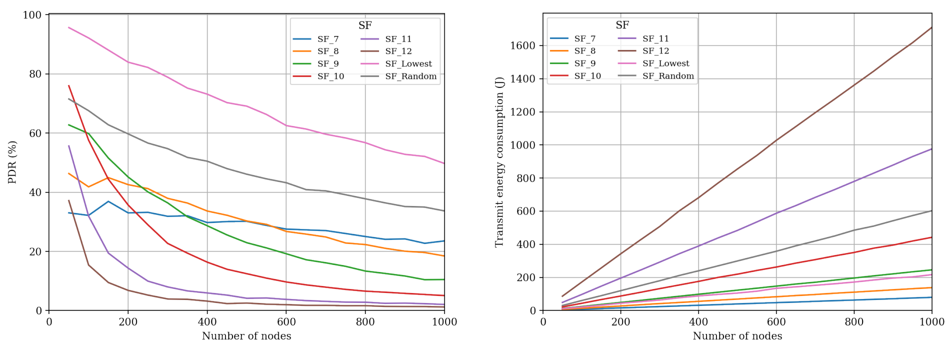
(a) PDR (b) Transmit energy

Figure A.1 : Plots for various spreading factors. (r = 3000 m, GW = 1)



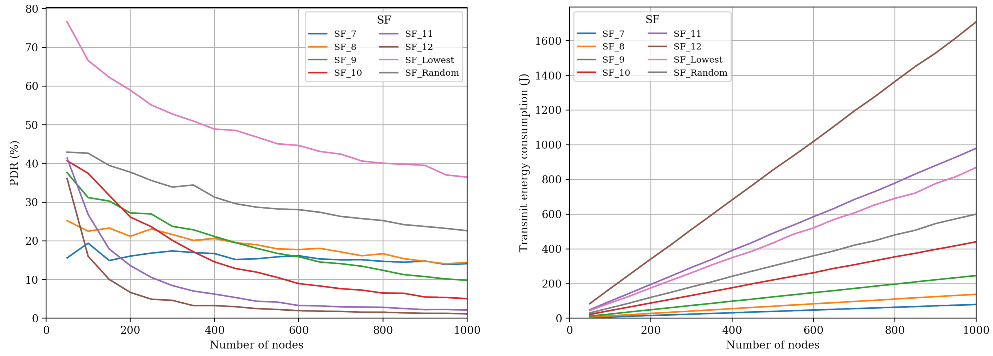
(a) PDR (b) Transmit energy

Figure A.2 : Plots for various spreading factors. (r = 5000 m, GW = 1)



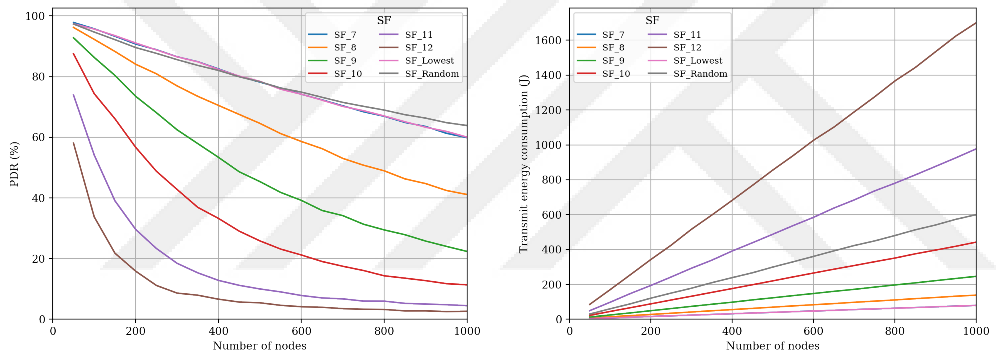
(a) PDR (b) Transmit energy

Figure A.3 : Plots for various spreading factors. (r = 7000 m, GW = 1)



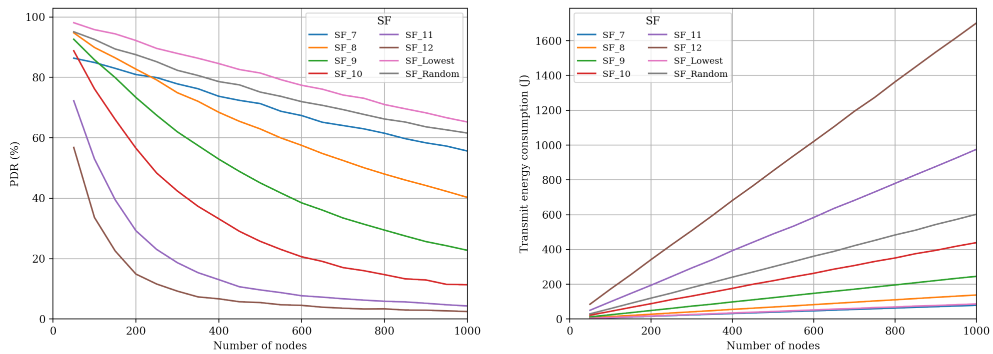
(a) PDR (b) Transmit energy

Figure A.4 : Plots for various spreading factors. (r = 10000 m, GW = 1)



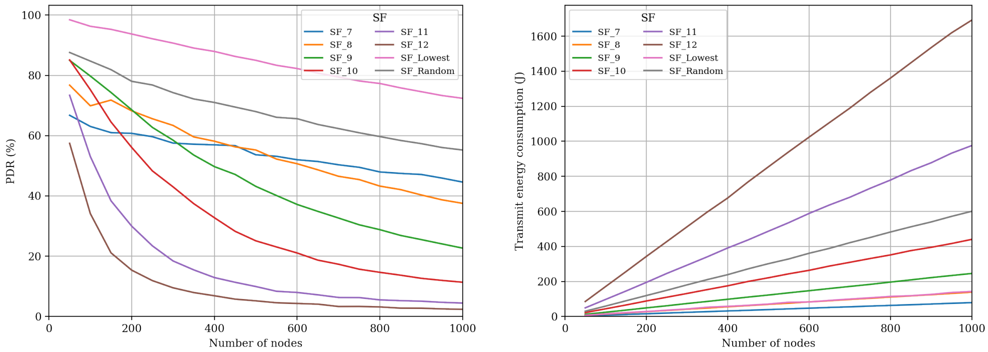
(a) PDR (b) Transmit energy

Figure A.5 : Plots for various spreading factors. (r = 3000 m, GW = 2)



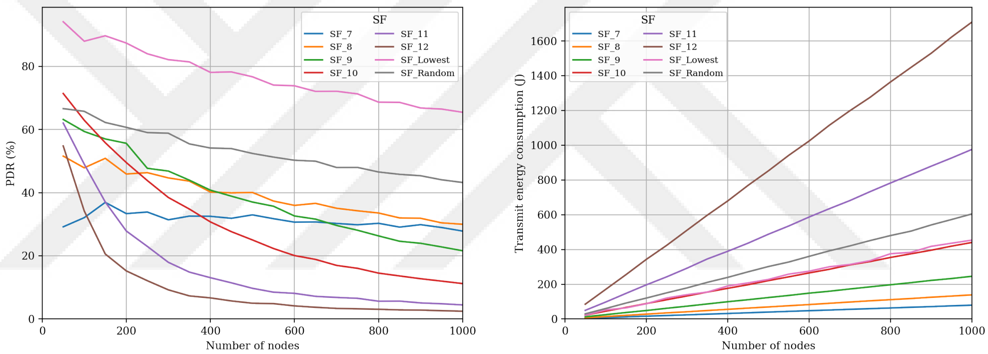
(a) PDR (b) Transmit energy

Figure A.6 : Plots for various spreading factors. (r = 5000 m, GW = 2)



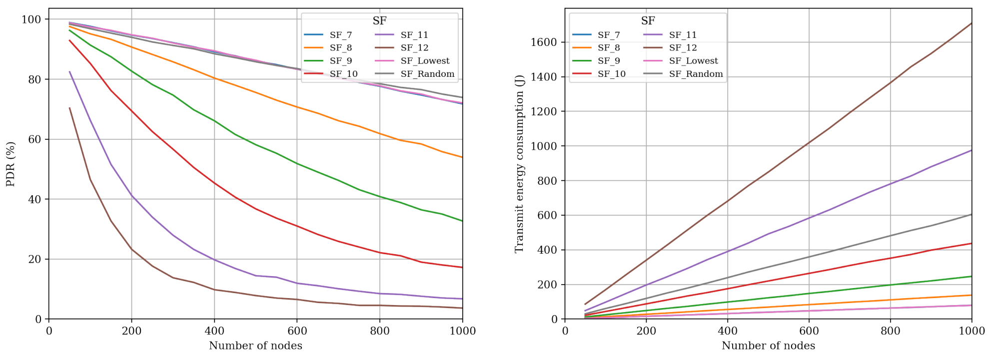
(a) PDR (b) Transmit energy

Figure A.7 : Plots for various spreading factors. (r = 7000 m, GW = 2)



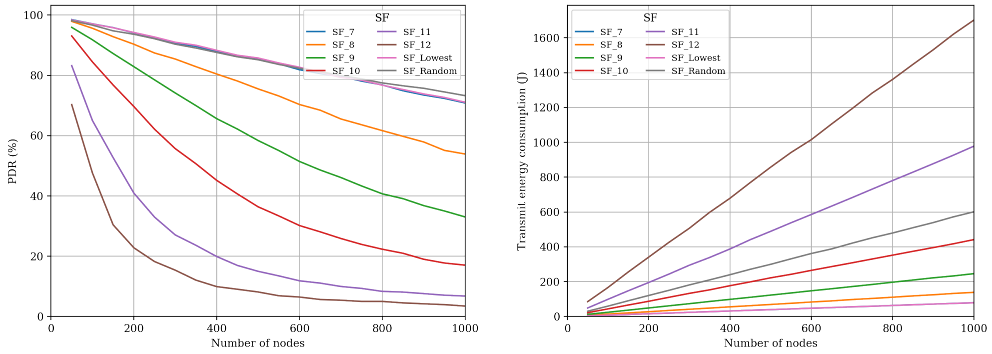
(a) PDR (b) Transmit energy

Figure A.8 : Plots for various spreading factors. (r = 10000 m, GW = 2)



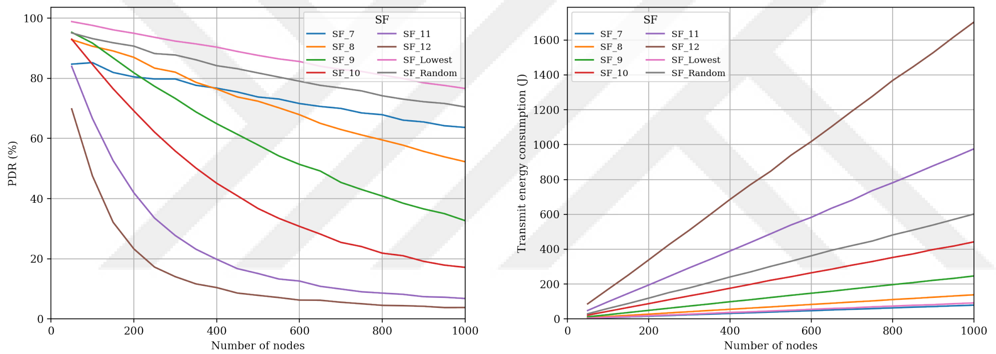
(a) PDR (b) Transmit energy

Figure A.9 : Plots for various spreading factors. (r = 3000 m, GW = 3)



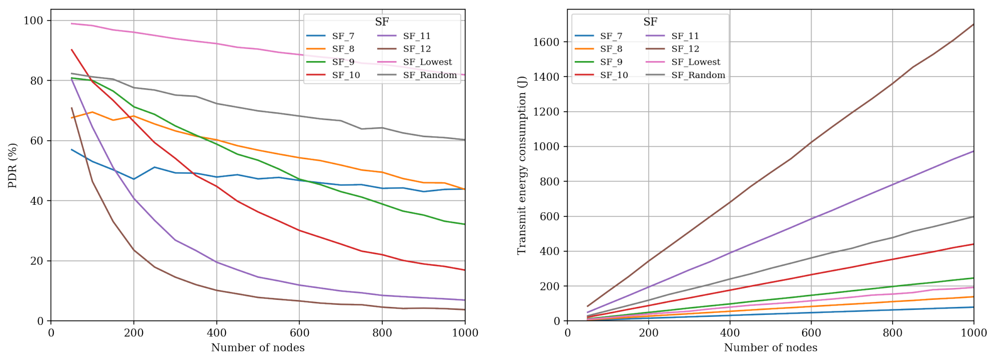
(a) PDR (b) Transmit energy

Figure A.10 : Plots for various spreading factors. (r = 5000 m, GW = 3)



(a) PDR (b) Transmit energy

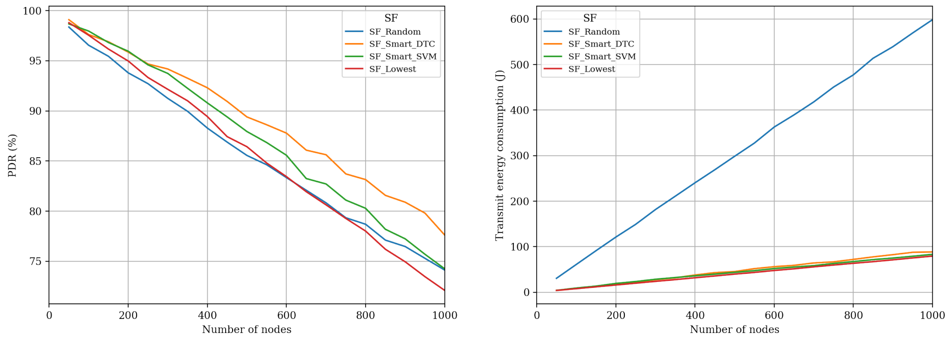
Figure A.11 : Plots for various spreading factors. (r = 7000 m, GW = 3)



(a) PDR (b) Transmit energy

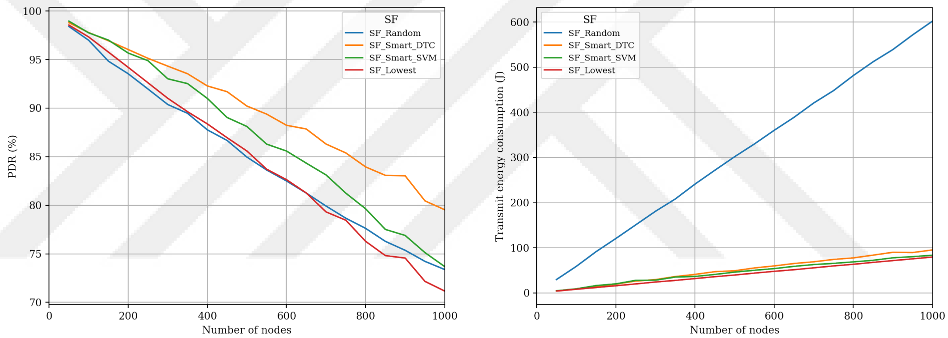
Figure A.12 : Plots for various spreading factors. (r = 10000 m, GW = 3)

APPENDIX B



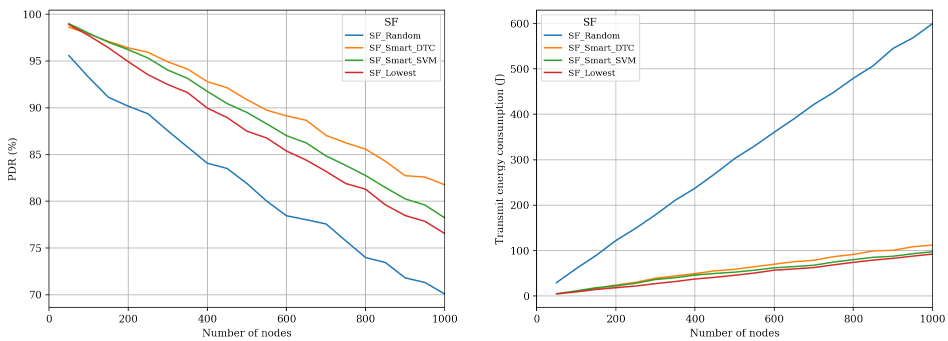
(a) PDR (b) Transmit energy

Figure B.1 : Plots for smart spreading factor schemes. (r = 3000 m, GW = 3)



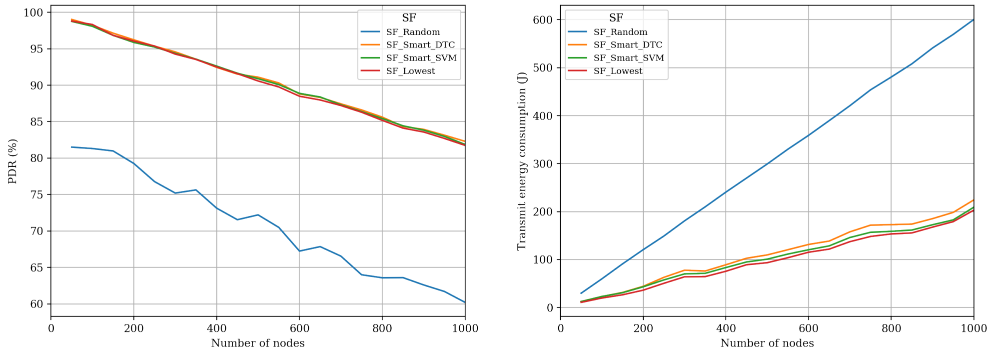
(a) PDR (b) Transmit energy

Figure B.2 : Plots for smart spreading factor schemes. (r = 5000 m, GW = 3)



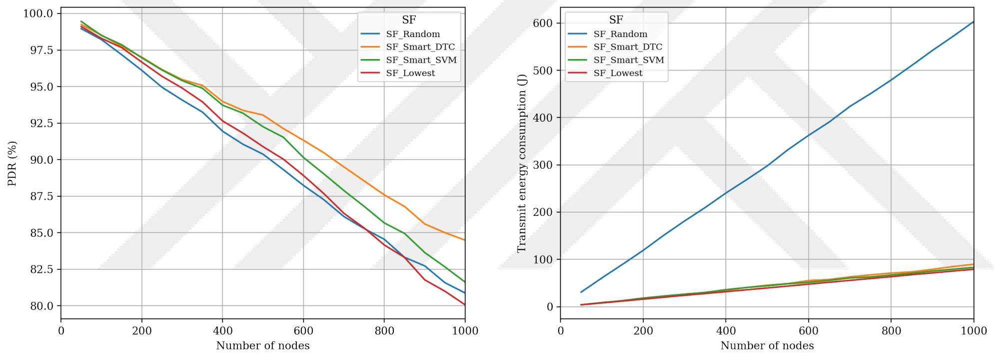
(a) PDR (b) Transmit energy

Figure B.3 : Plots for smart spreading factor schemes. (r = 7000 m, GW = 3)



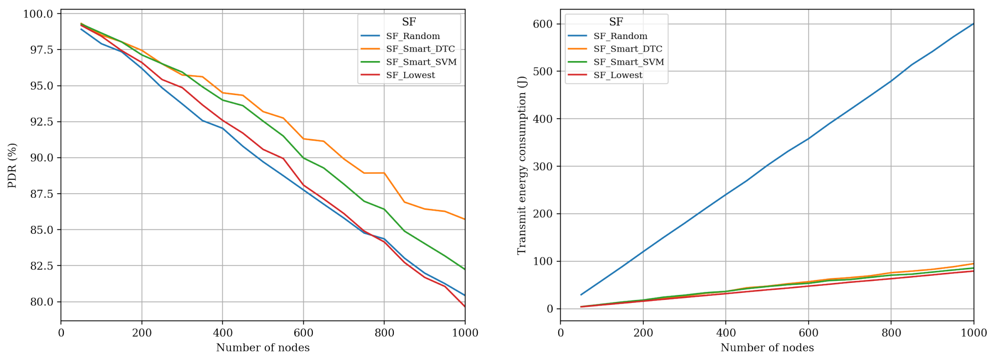
(a) PDR (b) Transmit energy

Figure B.4 : Plots for smart spreading factor schemes. (r = 10000 m, GW = 3)



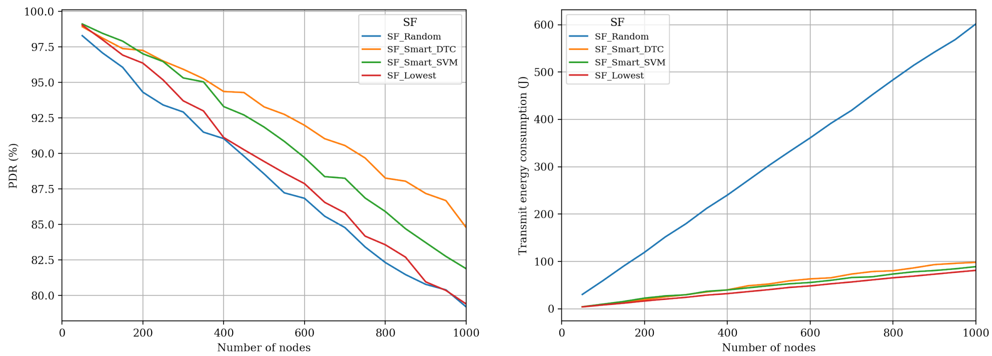
(a) PDR (b) Transmit energy

Figure B.5 : Plots for smart spreading factor schemes. (r = 3000 m, GW = 4)



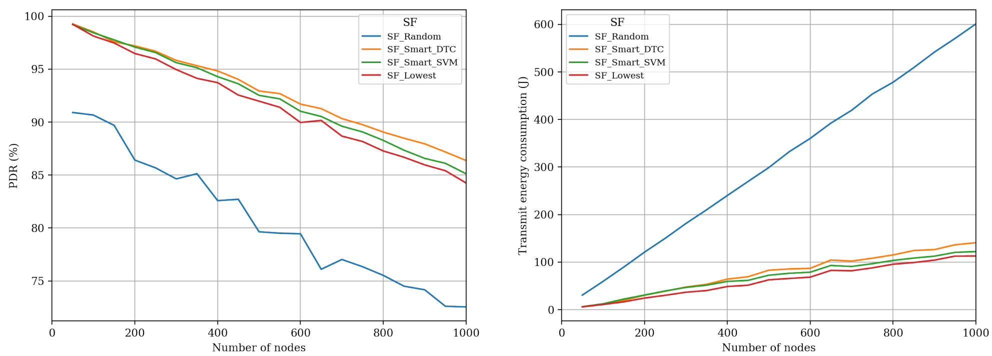
(a) PDR (b) Transmit energy

Figure B.6 : Plots for smart spreading factor schemes. (r = 5000 m, GW = 4)



(a) PDR (b) Transmit enery

Figure B.7 : Plots for smart spreading factor schemes. (r = 7000 m, GW = 4)



(a) PDR (b) Transmit energy

Figure B.8 : Plots for smart spreading factor schemes. (r = 10000 m, GW = 4)

CURRICULUM VITAE

Name Surname: Tuğrul Yatağan

Place and Date of Birth: Denizli / Turkey, 1992

E-Mail: yatagan@itu.edu.tr

EDUCATION:

- **B.Sc.:** 2015, Istanbul Technical University, Computer and Informatics Faculty, Computer Engineering Department
- **High School:** 2010, Denizli Nalan Kaynak Anatolian High School

PROFESSIONAL EXPERIENCE:

- 2017-Present Embedded Software Engineer at Maxim Integrated
- 2015-2016 Software Engineer at AirTies Wireless Networks
- 2014 Software Development Summer Intern at Yeni Hayat IT Corp.
- 2013 Software Development Summer Intern at Kartaca IT Corp.

PUBLICATIONS:

- Yatagan, T. and Oktug S., 2019, Smart Spreading Factor Assignment for LoRaWANs, *2019 IEEE Symposium on Computers and Communications (IEEE ISCC 2019)*