



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ  
ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ - WEBINTELLIGENCE

**Μελέτη της απόδοσης της τεχνολογίας ασύρματης  
δικτύωσης LoRaWAN για χρήση σε αστικό περιβάλλον**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΚΑΣΝΑΚΗ ΑΘΑΝΑΣΙΟΥ**

**Επιβλέπων :** Χατζημίσσιος Περικλής  
Καθηγητής, ΔΙΠΑΕ

Θεσσαλονίκη, Ιούνιος 2021

Η σελίδα αυτή είναι σκόπιμα λευκή.



ΔΙΕΘΝΕΣ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΤΗΣ ΕΛΛΑΔΟΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB  
INTELLIGENCE

## Μελέτη της απόδοσης της τεχνολογίας ασύρματης δικτύωσης LoRaWAN για χρήση σε αστικό περιβάλλον

### ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΚΑΣΝΑΚΗ ΑΘΑΝΑΣΙΟΥ**

**Επιβλέπων :** Χατζημίσσιος Περικλής  
Καθηγητής, ΔΙΠΑΕ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις Choose a date.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Όνομα Επώνυμο

Choose an item. ΔΙ.ΠΑ.Ε.

.....  
Όνομα Επώνυμο

Choose an item. ΔΙ.ΠΑ.Ε.

.....  
Όνομα Επώνυμο

Choose an item. ΔΙ.ΠΑ.Ε.

Θεσσαλονίκη, Choose a date

---

*(Υπογραφή)*

.....

Click here to enter text.

Click here to enter text.

© Choose a date— All rights reserved

## Περίληψη

Αντικείμενο της συγκεκριμένης διπλωματικής εργασίας είναι η μελέτη τεχνολογιών για την εγκατάσταση ενός δικτύου χαμηλής ισχύος και ευρείας περιοχής κάλυψης LPWAN (Low Power Wide Area Network). Τα τελευταία χρόνια, όλο και περισσότερες συσκευές συνδέονται στο Internet, με σκοπό την εποπτεία του φυσικού κόσμου μέσω αισθητήρων, αλλά και να αλληλεπιδράσουν μαζί του. Πολλές τεχνολογίες έχουν αναπτυχθεί με πλήθος εφαρμογών στον τομέα LPWAN με διαφορετικά χαρακτηριστικά. Το κοινό χαρακτηριστικό όλων αυτών των τεχνολογιών είναι η σύμβαση που κάνουμε, ότι θυσιάζουμε την ταχύτητα μετάδοσης και το χρόνο απόκρισης για χάρη της περιοχής κάλυψης της μετάδοσης της πληροφορίας και της μικρής κατανάλωσης ενέργειας. Το LoRaWAN είναι ένα δίκτυο ευρείας περιοχής, χαμηλής κατανάλωσης ενέργειας, χαμηλής ταχύτητας τηλεπικοινωνιακό δίκτυο. Η εργασία αυτή ασχολείται με τη μελέτη του LoRaWAN ως υλοποίηση ενός LPWAN καθώς και του φυσικού του επιπέδου LoRa (Long Range), τη μελέτη τεχνολογιών για την εγκατάσταση ενός τέτοιου δικτύου σε όλα του τα στάδια, τη μέτρηση βασικών παραμέτρων που επηρεάζουν την απόδοση σε σχέση με τη μελέτη περίπτωσης χρήσης. Περιγράφεται επίσης η εγκατάσταση ενός δικτύου LoRaWAN σε όλα τα στάδια. Εγκαταστάθηκαν διακομιστής σε επίπεδο δικτύου NS (Network Server), διακομιστής σε επίπεδο εφαρμογών AS (Application Server), διακομιστής σε επίπεδο διασύνδεσης BS (Bridge Server) και δοκιμάστηκαν τουλάχιστον δύο λύσεις για χρήση πύλης δικτύου (Gateway) και τουλάχιστον τρεις διαφορετικές επιλογές για τελικές συσκευές Eds (End Nodes). Τέλος προτάθηκε περίπτωση χρήσης για την πειραματική εποπτεία μηχανισμών σε ορυχείο της ΔΕΗ στην Πτολεμαΐδα Κοζάνης, η οποία βρίσκεται σε πειραματική φάση. Η πειραματική υλοποίηση ενός τέτοιου δικτύου, έφερε την ανάγκη για μελέτη ενός μικτού μοντέλου διαχείρισης και εποπτείας ενός περιβάλλοντος όπως είναι αυτό του ορυχείου. Σ' αυτό το μικτό μοντέλο το MQTT (Message Queuing Telemetry Transport) [1] είναι το συστατικό συγκόλλησης. Ανάλογα με τις ανάγκες για την ταχύτητα μετάδοσης δεδομένων θα μπορούσε κατά περίπτωση να χρησιμοποιηθεί είτε το LoRaWAN είτε ένα ταχύτερο ασύρματο δίκτυο (4G/5G).

**Λέξεις Κλειδιά:**IoT (Internet of Things), LoRa, LPWAN, LoRaWAN, MQTT.

Η σελίδα αυτή είναι σκόπιμα λευκή.

## **Abstract**

The subject of this specific thesis is the study of technologies for the installation of a Low-Power Wide-Area Network (LPWAN). During recent years, more and more devices are connected on the Internet, with the aim of surveilling the natural world by means of sensors, but also with the aim of interacting with the natural world. Many technologies have been developed with a variety of applications in the field of LPWAN with various properties. The characteristic that these technologies share is the convention we make; we sacrifice the speed of transmission and the time of response for the sake of the area of transmission coverage of the information and of low energy consumption. LoRaWAN is a wide area, a low power and a low speed telecommunication network. This thesis deals with the study of LoRaWAN as an implementation of LPWAN, as well as the natural level of LoRa. It also studies the technologies for the installation of this particular network during all of its stages and the measurement of principal parameters that affect its effectiveness in relation to case study use. Moreover, the study describes the installation of a LoRaWAN network at all stages. We made an installation of Network Server, Application Server, Bridge Server and we tested at least three different choices for End Nodes. Lastly, it was suggested a case use for experimental supervision of mechanisms in a mine of the Power Public Corporation (DEI) in the city of Ptolemaida, Kozani, which is in pilot phase. The experimental implementation of such a network brought forward the need for the study of a mixed model of management and surveillance of an environment, in our case, that of the mine. In this mixed model the MQTT (Message Queuing Telemetry Transport) [1] is the ingredient that unites everything. Depending on the case and on the need for data transmission speed, it could be used either the LoRaWAN or a faster wireless network (4G/5G).

**Keywords:** IoT (Internet of Things), LoRa, LPWAN, LoRaWAN, MQTT.

Η σελίδα αυτή είναι σκόπιμα λευκή.

## ***Πίνακας Εικόνων***

|                                                                           |    |
|---------------------------------------------------------------------------|----|
| Εικόνα 1- IoT .....                                                       | 15 |
| Εικόνα 2 - LPWAN .....                                                    | 16 |
| Εικόνα 3- LoRa Layers.....                                                | 17 |
| Εικόνα 4- Προδιαγραφές LoRaWAN ανά περιοχή της υφελίου. ....              | 18 |
| Εικόνα 5 - Παγκόσμιος χάρτης με τις ραδιοσυχνότητες της LoRa. ....        | 19 |
| Εικόνα 6 – Γραμμικά μεταβαλλόμενο.....                                    | 19 |
| Εικόνα 7 – Λογαριθμικά μεταβαλλόμενο.....                                 | 19 |
| Εικόνα 8- Chirp Spread Spectrum.....                                      | 20 |
| Εικόνα 9 – Κωδικοποιημένη ακολουθία bit στο LoRa .....                    | 21 |
| Εικόνα 10 - Spectrogram of generated upchirp at 125KHz using MATLAB ..... | 22 |
| Εικόνα 11 -Παράμετροι διαμόρφωσης LoRa .....                              | 23 |
| Εικόνα 12 – Πακέτο LoRa .....                                             | 26 |
| Εικόνα 13- de-chirping.....                                               | 27 |
| Εικόνα 14 – Spectrogram SF=7.....                                         | 27 |
| Εικόνα 15- Τοπολογία LoRaWAN.....                                         | 28 |
| Εικόνα 16 - Star .....                                                    | 29 |
| Εικόνα 17 - Mesh .....                                                    | 29 |
| Εικόνα 18- Bridge .....                                                   | 29 |
| Εικόνα 19 – LoRaWAN Layers .....                                          | 29 |
| Εικόνα 20 – Class A .....                                                 | 30 |
| Εικόνα 21- Class B.....                                                   | 31 |
| Εικόνα 22- Class C.....                                                   | 32 |
| Εικόνα 23- Δομή πακέτου .....                                             | 33 |
| Εικόνα 24- ΑΑρχιτεκτονική LoRa Server.....                                | 33 |
| Εικόνα 25- Chirpstack Server.....                                         | 34 |
| Εικόνα 26 – MQTT Αρχιτεκτονική.....                                       | 35 |
| Εικόνα 27- MQTT packet.....                                               | 36 |
| Εικόνα 28- MQTT packet fixed header.....                                  | 37 |
| Εικόνα 29-Πίνακας μηνυμάτων ελέγχου.....                                  | 37 |
| Εικόνα 30-TCP/IP μοντέλο 5 επιπέδων για το MQTT.....                      | 38 |
| Εικόνα 31- MQTT connect.....                                              | 39 |
| Εικόνα 32- MQTT subscribe .....                                           | 39 |
| Εικόνα 33-MQTT publish. ....                                              | 40 |
| Εικόνα 34- Δομή MQTT topics.....                                          | 41 |

|                                                                                           |    |
|-------------------------------------------------------------------------------------------|----|
| Εικόνα 35- MQTT publish με QoS 0. ....                                                    | 43 |
| Εικόνα 36. - MQTT publish με QoS 1. ....                                                  | 43 |
| Εικόνα 37. MQTT publish με QoS 2.....                                                     | 44 |
| Εικόνα 38- LoRa® CSS MATLAB simulation model.....                                         | 51 |
| Εικόνα 39- Measurement locations for dense forest experiment .....                        | 51 |
| Εικόνα 40 - Measurement locations for city experiment.....                                | 51 |
| Εικόνα 41 – Smart City with LoRa .....                                                    | 52 |
| Εικόνα 42 - Connect procedure .....                                                       | 52 |
| Εικόνα 43- Πηγή - <a href="https://www.semtech.com">https://www.semtech.com</a> [27]..... | 55 |
| Εικόνα 44- Τοπολογία δικτύου LoRa για τις μετρήσεις μας .....                             | 56 |
| Εικόνα 45 – LG01 Gateway .....                                                            | 58 |
| Εικόνα 46 – Kerlink Gateway .....                                                         | 58 |
| Εικόνα 47. Σύνδεση με WiFi της LG01-N .....                                               | 61 |
| Εικόνα 48. Log-in interface της LG01-N .....                                              | 62 |
| Εικόνα 49. Single Packet Forward v004.....                                                | 62 |
| Εικόνα 50. Upload Single Packet Forward version 004. ....                                 | 63 |
| Εικόνα 51. LoRa Radio Settings .....                                                      | 63 |
| Εικόνα 52. MAC Address .....                                                              | 64 |
| Εικόνα 53. Console TTN.....                                                               | 64 |
| Εικόνα 54. Register Gateway .....                                                         | 65 |
| Εικόνα 55. Register GateWay .....                                                         | 65 |
| Εικόνα 56. LoRa Server Settings .....                                                     | 66 |
| Εικόνα 57. Select IoT (Internet of Things) Server. ....                                   | 66 |
| Εικόνα 58 - Διαθέσιμες gateways.....                                                      | 67 |
| Εικόνα 59. Επιτυχής σύνδεση της GateWay με το TTN.....                                    | 67 |
| Εικόνα 60 - Pytrack .....                                                                 | 68 |
| Εικόνα 61 - Pycom .....                                                                   | 68 |
| Εικόνα 62 – 1 <sup>ο</sup> Πείραμα .....                                                  | 88 |
| Εικόνα 63 -1 <sup>ο</sup> Πείραμα ποιότητα σήματος.....                                   | 88 |
| Εικόνα 64 – 2 <sup>ο</sup> Πείραμα – βάδιση εντός πόλης.....                              | 89 |
| Εικόνα 65 – Ποιότητα σήματος 2 <sup>ου</sup> πειράματος.....                              | 89 |
| Εικόνα 66 - 3ο Πείραμα – Κίνηση με αυτοκίνητο .....                                       | 90 |
| Εικόνα 67 – Ποιότητα σήματος 3 <sup>ου</sup> Πειράματος.....                              | 90 |
| Εικόνα 68 – AirTime .....                                                                 | 91 |
| Εικόνα 69 - Πραγματικές μετρήσεις .....                                                   | 91 |
| Εικόνα 70 – Απασχόληση καναλιού .....                                                     | 99 |
| Εικόνα 71 – Χρόνος μετάδοσης .....                                                        | 99 |

|                                                                |     |
|----------------------------------------------------------------|-----|
| Εικόνα 72 – Κάλυψη .....                                       | 108 |
| Εικόνα 73 - Δομή δικτύου που στήθηκε στο Ορυχείο .....         | 109 |
| Εικόνα 74- Θέση των Gateways στο ορυχείο.....                  | 110 |
| Εικόνα 75- Συσκευές σε Application του ChirpStack.....         | 111 |
| Εικόνα 76- Αποστολή μετρήσεων έντασης και τάσης ρεύματος ..... | 111 |
| Εικόνα 77-Λήψη και οπτικοποίηση αναλογικών μετρήσεων.....      | 112 |
| Εικόνα 78 - Καταγραφή θέσης με GPS.....                        | 112 |

## ***Πίνακες***

|                                                                    |    |
|--------------------------------------------------------------------|----|
| Πίνακας 1 – Παράμετροι λειτουργίας Rycom .....                     | 68 |
| Πίνακας 2 - Παράγοντας Απόσταση – Μη αστικό περιβάλλον .....       | 92 |
| Πίνακας 3 - Παράγοντας Απόσταση - Αστικό Περιβάλλον .....          | 93 |
| Πίνακας 4 - Σφάλματα σε μή αστικό Περιβάλλον .....                 | 94 |
| Πίνακας 5 - Σφάλματα σε αστικό περιβάλλον .....                    | 95 |
| Πίνακας 6 – Κατανομή RSSI και SNR σε κίνηση .....                  | 96 |
| Πίνακας 7 – Κατανομή RSSI και SNR σε κίνηση μικτό περιβάλλον ..... | 97 |
| Πίνακας 8 – Απώλεια πακέτων σε κίνηση .....                        | 98 |
| Πίνακας 9 – Ρυθμός μετάδοσης με βάση την παράμετρο SF .....        | 99 |

# Πίνακας περιεχομένων

|                       |   |
|-----------------------|---|
| Πίνακας Εικόνων ..... | v |
|-----------------------|---|

|               |      |
|---------------|------|
| Πίνακες ..... | viii |
|---------------|------|

|                   |           |
|-------------------|-----------|
| <b>1 Εισαγωγή</b> | <b>12</b> |
|-------------------|-----------|

|                   |    |
|-------------------|----|
| 1.1 Κίνητρα ..... | 12 |
|-------------------|----|

|                        |    |
|------------------------|----|
| 1.1.1 Συνεισφορά ..... | 13 |
|------------------------|----|

|                             |    |
|-----------------------------|----|
| 1.2 Οργάνωση κειμένου ..... | 13 |
|-----------------------------|----|

|                             |           |
|-----------------------------|-----------|
| <b>2 Θεωρητικό υπόβαθρο</b> | <b>15</b> |
|-----------------------------|-----------|

|                |    |
|----------------|----|
| 2.1 LPWAN..... | 15 |
|----------------|----|

|                  |    |
|------------------|----|
| 2.2 LoRaWAN..... | 16 |
|------------------|----|

|                                          |    |
|------------------------------------------|----|
| 2.3 Αρχιτεκτονική του δικτύου LoRa ..... | 17 |
|------------------------------------------|----|

|                                |    |
|--------------------------------|----|
| 2.3.1 Στοιβα πρωτοκόλλων ..... | 17 |
|--------------------------------|----|

|                                              |    |
|----------------------------------------------|----|
| 2.3.2 Ραδιοσυχνότητες του Δικτύου LoRa ..... | 18 |
|----------------------------------------------|----|

|                                         |    |
|-----------------------------------------|----|
| 2.4 LoRa PHY Layer-Διαμόρφωση LoRa..... | 19 |
|-----------------------------------------|----|

|                                                               |    |
|---------------------------------------------------------------|----|
| 2.4.1 Υλοποίηση του σήματος "chirp" στη διαμόρφωση LoRa ..... | 19 |
|---------------------------------------------------------------|----|

|                                                                        |    |
|------------------------------------------------------------------------|----|
| 2.4.2 Παράγοντας Διασποράς (Spreading Factor) στη διαμόρφωση LoRa..... | 21 |
|------------------------------------------------------------------------|----|

|                                                                |    |
|----------------------------------------------------------------|----|
| 2.4.3 Θεωρητικός Υπολογισμός Ευαισθησίας και Link Budget ..... | 23 |
|----------------------------------------------------------------|----|

|                                                         |    |
|---------------------------------------------------------|----|
| 2.4.4 Τεχνικές κωδικοποίησης της διαμόρφωσης LoRa ..... | 24 |
|---------------------------------------------------------|----|

|                                   |    |
|-----------------------------------|----|
| 2.4.5 Μορφή των πακέτων LoRa..... | 25 |
|-----------------------------------|----|

|                             |    |
|-----------------------------|----|
| 2.5 Το πρότυπο LoRaWAN..... | 28 |
|-----------------------------|----|

|                                                                 |    |
|-----------------------------------------------------------------|----|
| 2.5.1 Τοπολογία και στοιβα πρωτοκόλλου του Δικτύου LoRaWAN..... | 28 |
|-----------------------------------------------------------------|----|

|                                         |    |
|-----------------------------------------|----|
| 2.5.2 Κλάσεις του δικτύου LoRaWAN ..... | 29 |
|-----------------------------------------|----|

|                                                            |    |
|------------------------------------------------------------|----|
| 2.6 Δομή Πακέτου και MAC εντολές του δικτύου LoRaWAN ..... | 32 |
|------------------------------------------------------------|----|

|                      |    |
|----------------------|----|
| 2.7 LoRa Server..... | 33 |
|----------------------|----|

|                                                      |    |
|------------------------------------------------------|----|
| 2.7.1 Επικοινωνία στο εσωτερικό του LoRa server..... | 34 |
|------------------------------------------------------|----|

|                                          |    |
|------------------------------------------|----|
| 2.7.2 User Datagram Protocol (UDP) ..... | 34 |
|------------------------------------------|----|

|                                                     |    |
|-----------------------------------------------------|----|
| 2.7.3 Message Queue Telemetry Transport (MQTT)..... | 35 |
|-----------------------------------------------------|----|

|                          |    |
|--------------------------|----|
| 2.7.4 Βασικές ιδέες..... | 36 |
|--------------------------|----|

|                        |    |
|------------------------|----|
| 2.7.5 Πακέτα MQTT..... | 36 |
|------------------------|----|

|                                  |    |
|----------------------------------|----|
| 2.7.6 Σύνδεση Client/Server..... | 37 |
|----------------------------------|----|

|                                |    |
|--------------------------------|----|
| 2.7.7 Το Pub/Sub μοντέλο ..... | 39 |
|--------------------------------|----|

|          |                                                                       |           |
|----------|-----------------------------------------------------------------------|-----------|
| 2.7.8    | Topics.....                                                           | 40        |
| 2.7.9    | Διαθήκη <i>client</i> και επιμονή μηνυμάτων.....                      | 44        |
| 2.7.10   | Google Remote Procedure Call (gRPC).....                              | 45        |
| 2.7.11   | Δομικά στοιχεία του LoRa Server .....                                 | 45        |
| 2.7.12   | Mac Controller .....                                                  | 46        |
| 2.7.13   | Gateway Bridge .....                                                  | 46        |
| 2.8      | Network Server.....                                                   | 47        |
| 2.9      | Application Server.....                                               | 49        |
| 2.10     | Σχετικές εργασίες .....                                               | 50        |
| <b>3</b> | <b>Περιπτώσεις χρήσης – παραμετροποίηση ενός δικτύου LoRa .....</b>   | <b>54</b> |
| 3.1      | Συνθήκες πραγματοποίησης της έρευνας.....                             | 54        |
| 3.2      | Στόχος της έρευνας.....                                               | 54        |
| 3.2.1    | Ερευνητικά ερωτήματα.....                                             | 55        |
| 3.3      | Μελέτη παραμέτρων που επηρεάζουν την απόδοση ενός δικτύου LoRa. ....  | 56        |
| 3.4      | Τεχνικές προδιαγραφές του δικτύου LoRa για τα πειράματα .....         | 56        |
| 3.4.1    | Εφαρμογή 1 <sup>η</sup> .....                                         | 57        |
| 3.4.2    | Περιγραφή τεχνικών χαρακτηριστικών των gateways που συμμετέχουν. .... | 57        |
| 3.5      | Παράμετροι αξιολόγησης.....                                           | 59        |
| <b>4</b> | <b>Τεχνικές λεπτομέρειες</b>                                          | <b>61</b> |
| 4.1      | Λεπτομέρειες υλοποίησης .....                                         | 61        |
| 4.1.1    | Στάδια προετοιμασίας και παραμετροποίησης της LG01-N.....             | 61        |
| 4.1.2    | Παραμετροποίηση LoRa Radio Settings.....                              | 63        |
| 4.1.3    | Σύνδεση της GateWay με το The Thnings Network (TTN).....              | 64        |
| 4.2      | LoPy – Pycom-Pytrack.....                                             | 67        |
| 4.3      | Προσομοίωση στο MatLab .....                                          | 76        |
| 4.4      | End Nodes .....                                                       | 79        |
| <b>5</b> | <b>Οργάνωση πειραμάτων</b>                                            | <b>87</b> |
| 5.1.1    | Βασική παραμετροποίηση.....                                           | 87        |
| 5.1.2    | 1 <sup>ο</sup> πείραμα .....                                          | 87        |
| 5.1.3    | 2 <sup>ο</sup> Πείραμα .....                                          | 88        |
| 5.1.4    | 3 <sup>ο</sup> πείραμα .....                                          | 89        |

|          |                                                                          |            |
|----------|--------------------------------------------------------------------------|------------|
| 5.2      | Αποτελέσματα .....                                                       | 90         |
| 5.2.1    | <i>Spreading Factor (SF)</i> .....                                       | 90         |
| 5.2.2    | <i>Received Signal Strength Indication (RSSI)</i> .....                  | 91         |
| 5.2.3    | <i>Signal-to-Noise Ratio (SNR)</i> .....                                 | 92         |
| 5.2.4    | <i>Παράγοντας απόσταση</i> .....                                         | 92         |
| 5.2.5    | <i>Παράγοντας SF</i> .....                                               | 93         |
| 5.2.6    | <i>Παράγοντας κίνηση</i> .....                                           | 95         |
| 5.3      | Παράγοντας αριθμός end-devices .....                                     | 98         |
| 5.3.1    | <i>Προσομοίωση εξυπηρέτησης πολλών end-devices</i> .....                 | 98         |
| 5.4      | Σύνοψη συμπερασμάτων αξιολόγησης .....                                   | 106        |
| 5.5      | Επιλογή Gateway .....                                                    | 106        |
| 5.5.1    | <i>Περίπτωση χρήσης που προκύπτει από το 1<sup>ο</sup> πείραμα</i> ..... | 106        |
| 5.5.2    | <i>Περίπτωση χρήσης που προκύπτει από το 2<sup>ο</sup> πείραμα</i> ..... | 106        |
| 5.5.3    | <i>Περίπτωση χρήσης που προκύπτει από το 3<sup>ο</sup> πείραμα</i> ..... | 107        |
| <b>6</b> | <b>Επίλογος</b>                                                          | <b>108</b> |
| 6.1      | Πρόταση για την περίπτωση «Ορυχεία ΔΕΗ» .....                            | 108        |
| 6.2      | Πρόταση .....                                                            | 109        |
| 6.2.1    | <i>Server</i> .....                                                      | 109        |
| 6.3      | Gateway.....                                                             | 110        |
| 6.4      | End-nodes.....                                                           | 110        |
| 6.5      | Ολοκλήρωση για Οπτικοποίηση δεδομένων .....                              | 112        |
| 6.6      | Συμπεράσματα - Μελλοντικές επεκτάσεις .....                              | 113        |
| <b>7</b> | <b>Βιβλιογραφία</b>                                                      | <b>115</b> |

# 1

## *Εισαγωγή*

### *1.1 Κίνητρα*

Η σύγχρονη ασύρματη τεχνολογία είναι ο καταλύτης για τη σύνδεση νέων αντικειμένων και συσκευών στο Διαδίκτυο. Οι άνθρωποι θέλουν να χρησιμοποιήσουν μια εφαρμογή για κινητά για να προθερμάνουν τα αυτοκίνητά τους σε ένα κρύο πρωί, να ξεκλειδώσουν το σπίτι τους από μια απομακρυσμένη τοποθεσία, ώστε ο φίλος τους να μπορεί να μπει και να ταΐσει τη γάτα, να παρακολουθεί την ενεργειακή χρήση διαφόρων συσκευών στα σπίτια τους και πολλά άλλα. Σε όλα τις περιπτώσεις συμμετέχει η χρήση ασύρματης τεχνολογίας. Τα βιομηχανικά περιβάλλοντα χρησιμοποιούν επίσης ασύρματα μέσα για τη σύνδεση αισθητήρων και ενεργοποιητών που επιτρέπουν τον αυτόματο σχεδιασμό, την παρακολούθηση, την επικοινωνία από μηχανή σε μηχανή, την ανάλυση big data και πολλές άλλες περιπτώσεις χρήσης.

Η αυξανόμενη τάση σύνδεσης διαφόρων πραγμάτων σε ασύρματο δίκτυο και έτσι η δυνατότητα απομακρυσμένης πρόσβασης σε αυτά αρχίζει να συνειδητοποιεί αυτό που συχνά αναφέρεται ως το Διαδίκτυο των πραγμάτων IoT (Internet of Things). Για να επιτευχθεί αυτό, χρησιμοποιούνται διάφορες τεχνικές, όλες ανάλογα με την περίπτωση χρήσης. Μία αναπτυσσόμενη τεχνική για επικοινωνία μεγάλων αποστάσεων είναι τα δίκτυα ευρείας περιοχής χαμηλής ισχύος (LPWANs). Τα LPWANs είναι ασύρματα πρωτόκολλα που έχουν αναπτυχθεί για περιορισμένες συσκευές, όπως αυτές που χρησιμοποιούνται συχνά σε υπηρεσίες IoT. Η αντιστάθμιση είναι ο χαμηλότερος ρυθμός bit σε αντάλλαγμα χαμηλότερης ενέργειας κατανάλωση. Ένα τέτοιο LPWAN είναι το δίκτυο ευρείας περιοχής χαμηλής ισχύος μεγάλης εμβέλειας (LoRaWAN). Το LoRaWAN είναι ένα αμφίδρομο πρωτόκολλο επικοινωνίας IoT (Internet of Things) με μεγάλη εμβέλεια αλλά χαμηλό ρυθμό δεδομένων.

Δεδομένου ότι το LoRaWAN λειτουργεί στα φάσματα χωρίς άδεια περίπου 868MHz στην Ευρώπη και 915MHz στη Βόρεια Αμερική, δεν απαιτείται άδεια ή άδεια για τη ρύθμιση ενός LoRaWAN. Ο συνδυασμός του μεγάλου εύρους και του χαμηλού ορίου για να ξεκινήσετε καθιστά το LoRaWAN έναν εξαιρετικό υποψήφιο για χρήση σε πολλές εφαρμογές IoT. Το μειονέκτημα της χρήσης του μη αδειοδοτημένου φάσματος είναι ότι υπάρχουν πολλές εφαρμογές που λειτουργούν στην ίδια ζώνη συχνοτήτων, κάτι που μπορεί να προκαλέσει παρεμβολές μεταξύ των εφαρμογών. Για να μην πλημμυρίσει μια ζώνη συχνοτήτων, το σχήμα διαμόρφωσης LoRa στην Ευρώπη επιβάλλει έναν περιορισμό ανά υποζώνη Duty Cycle (DC) για την αντιμετώπιση των κανονισμών που εκδίδονται από το Ευρωπαϊκό Ινστιτούτο Προτύπων Τηλεπικοινωνιών ETSI (European Telecommunications Standards Institute). Ο κύκλος εργασίας DC (Duty Cycle) καθορίζει πόση συνολική ώρα επιτρέπεται να μεταδίδει ένας πομπός ανά ώρα σε αυτήν την υποζώνη. Για παράδειγμα, εάν μια υποζώνη έχει DC 1%, ο συνολικός χρόνος που περνά ο πομποδέκτης για τη μετάδοση δεδομένων δεν πρέπει να υπερβαίνει τα 36 δευτερόλεπτα / ώρα. Το σχήμα διαμόρφωσης LoRa βασίζεται σε μια τεχνική φάσματος chirped spread spectrum που το καθιστά ισχυρό στο να διοχετεύει θόρυβο και επίσης ανθεκτικό σε ξεθώριασμα πολλαπλών διαδρομών. Λόγω του περιορισμού του DC (Duty cycle) και της χαμηλής απόδοσης, μεταξύ 250 και 50000 bit / s, είναι σημαντικό να διατηρηθεί ο ρυθμός σύγκρουσης όσο το δυνατόν χαμηλότερος.

### **1.1.1 Συνεισφορά**

Στα πλαίσια της εκπόνησης της διπλωματικής εργασίας πραγματοποιήθηκε η υλοποίηση και χρήση δικτύων LoRa σε διαφορετικές περιπτώσεις χρήσης και σε όλα τα στάδια υλοποίησης.

- Μελετήθηκε το απαραίτητο λογισμικό για την λειτουργία ενός τέτοιου δικτύου:
  - Network Server, Application Server, Bridge Server, Data mining – Data View
- Μελετήθηκε το απαραίτητο υλικό με διάφορους συνδυασμούς:
  - Gateways, End Nodes
- Μελετήθηκε η απόδοση ενός τέτοιου δικτύου.
- Προτάθηκε και υλοποιήθηκε ένα τέτοιο δίκτυο σε πραγματικές συνθήκες.

## **1.2 Οργάνωση κειμένου**

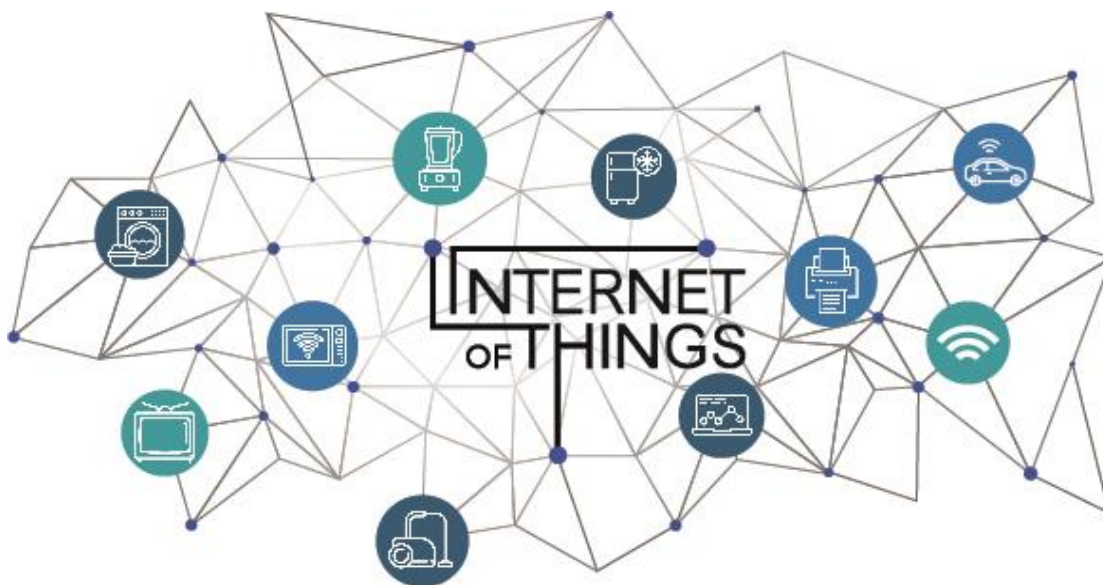
- Στο 2<sup>ο</sup> κεφάλαιο αναφέρονται ενδεικτικά σχετικές ερευνητικές εργασίες που ήταν ένας αρχικός οδηγός για την οργάνωση της εργασίας.
- Στο 3<sup>ο</sup> κεφάλαιο της εργασίας γίνεται μια αναφορά στο θεωρητικό υπόβαθρο της εργασίας και συγκεκριμένα στις έννοιες LPWAN, LoRa, LoRaWAN, MQTT, και τεχνολογίες για την ασύρματη μετάδοση δεδομένων με χαμηλή κατανάλωση.

- Στο 4<sup>ο</sup> κεφάλαιο περιγράφεται συνοπτικά η υλοποίηση ενός δικτύου LoRaWAN σε non Public περιβάλλον σε όλα τα επίπεδα. Δηλαδή περιγράφεται το στήσιμο των κατάλληλων servers, ο προγραμματισμός και παραμετροποίηση μικροσυσκευών που συμμετέχουν .
- Στο 5<sup>ο</sup> κεφάλαιο περιγράφεται το 1<sup>ο</sup> πείραμα για την μέτρηση ποιότητας και την απόδοση ενός δικτύου LoRaWAN σε τρεις διαφορετικές περιπτώσεις χρήσεις, με σκοπό την αποδοτική παραμετροποίηση ενός τέτοιου δικτύου.
- Στο ίδιο κεφάλαιο προτείνεται μια παραπέρα μελέτη για ανάπτυξη υλικολογισμικού αλλά και συνδυασμένης χρήσης τεχνολογιών, ώστε ν' αποφύγουμε τυχόν περιορισμούς που μπορεί ν' αφορά , ταχύτητα , DC (Duty cycle), ασφάλεια.
- Στο 6<sup>ο</sup> κεφάλαιο περιγράφεται το πρακτικό κομμάτι του 1<sup>ου</sup> πειράματος. Δηλαδή η παραμετροποίηση της LG01-N , του TTN (The Things Network), ο προγραμματισμός του Lopy - Pycom σαν End Node GPS και η λήψη δεδομένων για επεξεργασία από το TTN (The Things Network) με χρήση Python.
- Στο 7<sup>ο</sup> Κεφάλαιο περιγράφεται η υλοποίηση και εφαρμογή των παραμετροποιήσεων που προέκυψαν σε ένα ολοκληρωμένο δίκτυο που υλοποιήθηκε σε ορυχείο της ΔΕΗ στην Πτολεμαΐδα Κοζάνης.

# 2

## Θεωρητικό υπόβαθρο

Το Internet of Things (IoT) έχει θεωρηθεί ως η νέα επανάσταση στον τομέα των επικοινωνιών και τα τελευταία χρόνια ο αριθμός των συσκευών που έχουν πρόσβαση στο διαδίκτυο ολοένα και αυξάνεται. Γίνεται λοιπόν εφικτό μία συσκευή με χρήση αισθητήρων να παρέχει πληροφορίες στο δίκτυο, να δέχεται εντολές προς υλοποίηση κάποιας ενέργειας, ή και συνδυασμό των παραπάνω. Όπως γίνεται αντιληπτό, δίνετε να δημιουργηθούν εφαρμογές με σκοπό την αποδοτικότερη διαχείριση των ανθρώπινων, φυσικών και ενεργειακών πόρων, την βελτιστοποίηση διαδικασιών παραγωγής αλλά και την αυτοματοποίηση εργασιών. Ως αποτέλεσμα αυτού, μέχρι το 2025, αναμένεται να λειτουργούν 30.9 δισεκατομμύρια συσκευές IoT. [2]

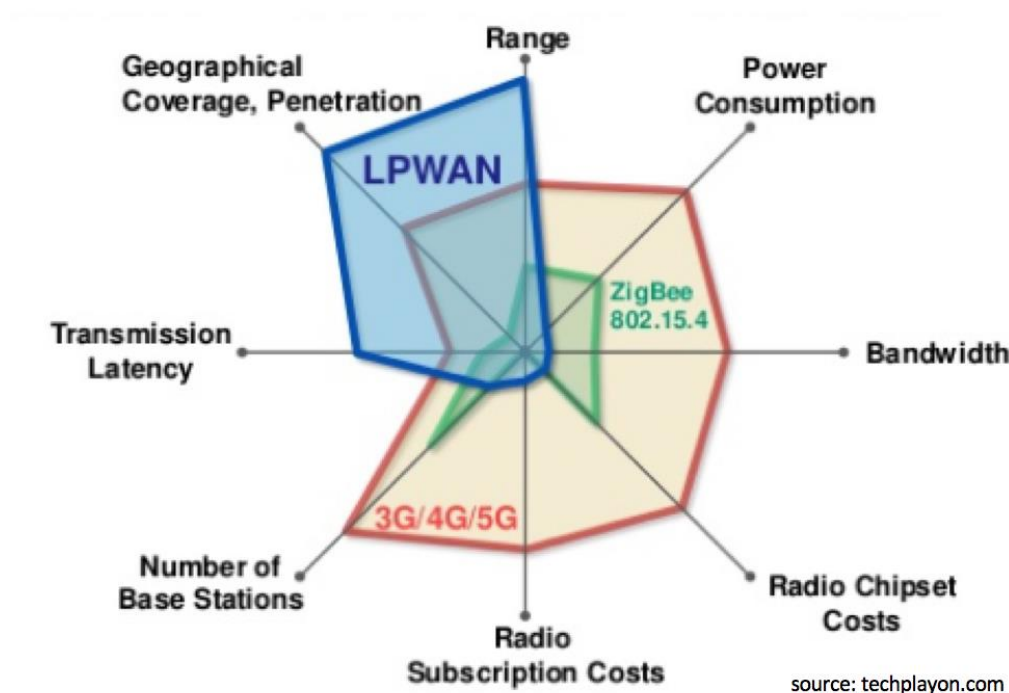


Εικόνα 1- IoT

### 2.1 LPWAN

Για την ανάπτυξη εφαρμογών IoT, υπάρχουν διαφορετικές τεχνολογίες δικτύων με σκοπό την παροχή πρόσβασης στο διαδίκτυο για τις συσκευές. Γίνεται κατανοητό ότι μία τεχνολογία δεν μπορεί να εξυπηρετήσει όλο το φάσμα των εφαρμογών που μπορεί να υπάρξουν λόγω διαφορετικότητας σε ανάγκες, στόχους λειτουργίας, διαθεσιμότητα φυσικών πόρων και άλλα

[3]. Παραδοσιακές λύσεις όπως τα τοπικά ασύρματα δίκτυα σαν το WiFi, τα ασύρματα δίκτυα μικρής εμβέλειας σαν το Bluetooth ή ZigBee αλλά και τα κυψελοειδή δίκτυα σαν το GSM, LTE και άλλα, αν και ήδη χρησιμοποιούνται για εφαρμογές IoT συνήθως έχουν υψηλό κόστος, υψηλή ενεργειακή κατανάλωση και αυξημένη πολυπλοκότητα. Ως λύση, αναπτύχθηκαν τα δίκτυα χαμηλής ισχύος και ευρείας κάλυψης (Low Power Wide Area Networks), τα οποία στοχεύουν σε εφαρμογές που ανταλλάσσουν μικρό όγκο δεδομένων. Προσφέρουν μεγάλη ενεργειακή αυτονομία, ευρεία περιοχή κάλυψης, χαμηλό κόστος και μεγάλη χωρητικότητα σε κόμβους, σε αντάλλαγμα με την χαμηλή ταχύτητα μετάδοσης και τον αυξημένο χρόνο απόκρισης.



Εικόνα 2 - LPWAN

## 2.2 LoRaWAN

Υπάρχουν περισσότερες από μία υλοποιήσεις ενός δικτύου LP WAN οι οποίες προσφέρουν διαφορετικό εύρος κάλυψης, ενεργειακή αυτονομία αλλά και συχνότητες λειτουργίας, λόγω της διαφορετικής προσέγγισης. Στην παρούσα εργασία, επιλέχθηκε η ενασχόληση με την LoRaWAN, με το φυσικό της επίπεδο να ονομάζεται LoRa το οποίο δημιουργήθηκε για να επιτευχθεί η μεγάλη απόσταση επικοινωνίας [4]. Πολλές τεχνολογίες που θέλουν να πετύχουν χαμηλή κατανάλωση ισχύος, χρησιμοποιούν την διαμόρφωση FSK στο φυσικό τους επίπεδο, ενώ η LoRa είναι η πρώτη χαμηλού κόστους εφαρμογή για εμπορική χρήση, που βασίζεται στην διαμόρφωση chirp spread spectrum, η οποία διατηρεί την χαμηλή κατανάλωση ισχύος και παράλληλα αυξάνει την περιοχή κάλυψης λόγω της μεγάλης αντοχής στον θόρυβο μέσω παρεμβολών [5]. Επιπλέον λόγοι που οδήγησαν στην επιλογή της LoRaWAN είναι ότι

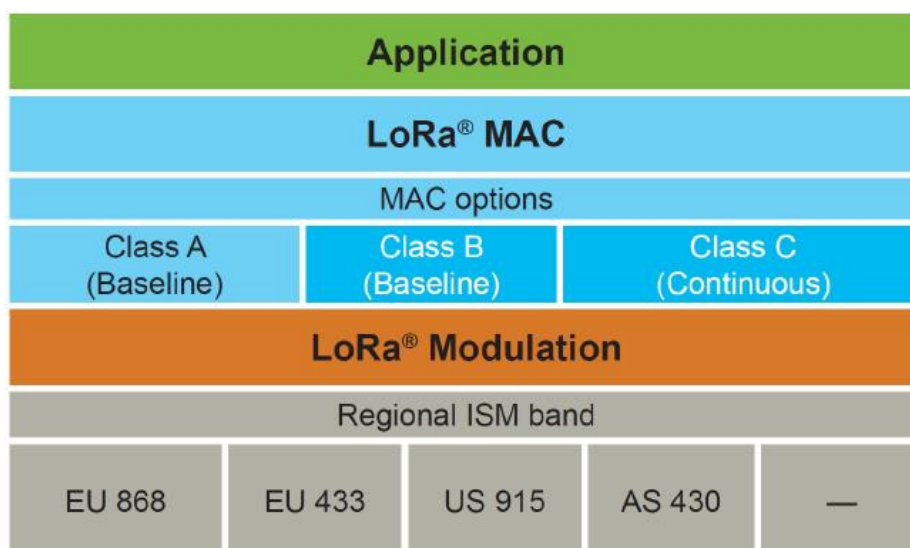
δημιουργήθηκε πρόσφατα και είναι διαρκώς εξελισσόμενη, υπάρχει μεγάλη ποικιλία στο υλικό που μπορεί να χρησιμοποιηθεί με χαμηλό κόστος, καθιστώντας την έτσι προσβάσιμη και τέλος υπάρχει μεγάλη κοινότητα υποστήριξης τόσο στο κομμάτι των λογισμικών όσο και στο κομμάτι των υλικών.

## 2.3 Αρχιτεκτονική του δικτύου LoRa

### 2.3.1 Στοιβα πρωτοκόλλων

Η στοιβα πρωτοκόλλων του δικτύου LoRa αποτελείται από 4 επίπεδα, κάθε ένα από τα οποία πραγματοποιεί συγκεκριμένες λειτουργίες χωρίς να παρεμβαίνει στις λειτουργίες του άλλου [6]. Πιο συγκεκριμένα, διαχωρίζονται ως εξής: Επίπεδο εφαρμογής (LoRa Application Layer), Επίπεδο Ελέγχου Προσπέλασης Μέσων (LoRa MAC Layer), Επίπεδο Φυσικό (LoRa PHY Layer) και τέλος Επίπεδο Ραδιοσυχνοτήτων (LoRa RF Layer).

Τα επίπεδα είναι αυστηρά διαχωρισμένα μεταξύ τους, ενώ αξιοποιούν τις υπηρεσίες του κατώτερου επιπέδου τους και προσφέρουν υπηρεσίες στο ανώτερο τους. Με αυτόν τον τρόπο, διευκολύνεται τόσο η μελέτη, όσο και η σωστή σχεδιάσή τους. Κάθε επίπεδο υλοποιεί καθορισμένα πρωτόκολλα, με κυρίαρχο στόχο την αυξημένη διάρκεια ζωής των τελικών συσκευών (End-Nodes) αλλά και την ανθεκτικότητα του σήματος. Μία τυπική μορφή της στοιβάς πρωτοκόλλων του δικτύου (LoRa) φαίνεται παρακάτω, ενώ στα επόμενα κεφάλαια θα γίνει αναλυτική περιγραφή κάθε επιπέδου.



Εικόνα 3- LoRa Layers

### 2.3.2 Ραδιοσυχνότητες του Δικτύου LoRa

Όπως έχει αναφερθεί και στην εισαγωγή, ένα πολύ σημαντικό στοιχείο του δικτύου LoRa είναι ότι χρησιμοποιεί μπάντες συχνοτήτων του Ευρωπαϊκού Ινστιτούτου Προτύπων Τηλεπικοινωνιών ISM (Industrial, Scientific and Medical). Οι συγκεκριμένες ζώνες συχνοτήτων αποτελούν μέρος του ραδιοφάσματος, στο οποίο δεν απαιτείται αδειοδότηση για τη χρήση τους. Όπως μπορεί κανείς να καταλάβει από την ονομασία τους, προορίζονται για εφαρμογές, είτε σε βιομηχανικό περιβάλλον είτε για επιστημονικούς, εκπαιδευτικούς και ιατρικούς σκοπούς. Είναι διεθνώς αναγνωρισμένες και υιοθετήθηκαν από τη Διεθνή Ένωση Τηλεπικοινωνιών ITU (International Telecommunication Union ), αλλά κάθε χώρα μπορεί να διαφοροποιείται στη χρήση τους λόγω διαφορετικών κανονισμών στην κατανομή ραδιοσυχνοτήτων. Τέλος, η χρήση τους εμπίπτει σε αυστηρούς κανονισμούς, όσων αφορά την ισχύς εκπομπής αλλά και τα όρια εκπομπής, ώστε να μη υπάρξουν τυχόν παρεμβολές με τις ζώνες συχνοτήτων που απαιτούν άδεια χρήσης επί πληρωμή.

Οι προδιαγραφές που αφορούν την τεχνολογία LoRaWAN ποικίλουν από χώρα σε χώρα, ανάλογα με την κατανομή ραδιοφάσματος αλλά και τις εκάστοτε νομοθεσίες [7]. Στην Ευρώπη, οι περιορισμοί συντάχθηκαν από τον Ευρωπαϊκό Οργανισμό Τηλεπικοινωνιακών Προτύπων ETSI (European Telecommunications Standards Institute) και ανατέθηκε η ζώνη συχνοτήτων 867-869 MHz. Αξίζει ακόμα να αναφερθεί ότι έχουν τεθεί περιορισμοί τόσο στην μέγιστη εκπεμπόμενη ισχύς , περί τα +14 dBm, όσο και στον αριθμό των καθορισμένων καναλιών.

|                       | Europe         | North America               | China                                | Korea                                | Japan                                | India                                |
|-----------------------|----------------|-----------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|
| <b>Frequency band</b> | 867-869MHz     | 902-928MHz                  | 470-510MHz                           | 920-925MHz                           | 920-925MHz                           | 865-867MHz                           |
| <b>Channels</b>       | 10             | 64 + 8 +8                   | In definition by Technical Committee | In definition by Technical Committee | In definition by Technical Committee | In definition by Technical Committee |
| <b>Channel BW Up</b>  | 125/250kHz     | 125/500kHz                  |                                      |                                      |                                      |                                      |
| <b>Channel BW Dn</b>  | 125kHz         | 500kHz                      |                                      |                                      |                                      |                                      |
| <b>TX Power Up</b>    | +14dBm         | +20dBm typ (+30dBm allowed) |                                      |                                      |                                      |                                      |
| <b>TX Power Dn</b>    | +14dBm         | +27dBm                      |                                      |                                      |                                      |                                      |
| <b>SF Up</b>          | 7-12           | 7-10                        |                                      |                                      |                                      |                                      |
| <b>Data rate</b>      | 250bps- 50kbps | 980bps-21.9kbps             |                                      |                                      |                                      |                                      |
| <b>Link Budget Up</b> | 155dB          | 154dB                       |                                      |                                      |                                      |                                      |
| <b>Link Budget Dn</b> | 155dB          | 157dB                       |                                      |                                      |                                      |                                      |

Εικόνα 4- Προδιαγραφές LoRaWAN ανά περιοχή της υφελίου.



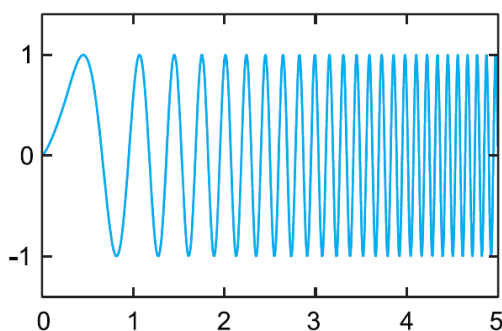
Εικόνα 5 - Παγκόσμιος χάρτης με τις ραδιοσυχνότητες της LoRa.

## 2.4 LoRa PHY Layer-Διαμόρφωση LoRa

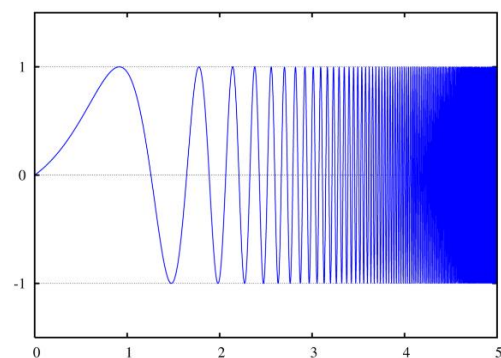
Η διαμόρφωση LoRa βασίζεται στην τεχνική διαμόρφωσης διασποράς φάσματος Chirp Spread Spectrum (CSS) και υπάρχει αντίστοιχο δίπλωμα ευρεσιτεχνίας, το οποίο κατοχυρώνει την "πατέντα". Γι' αυτό τον λόγο δεν υπάρχει σαφής περιγραφή της τεχνικής διαμόρφωσης, παρά μερικά έγγραφα που δίνουν ορισμένες πληροφορίες. Ωστόσο, έχει γίνει εξαιρετική δουλειά για την "αποκρυπτογράφηση" της από πολλούς, όπως η δουλειά του Matthew Knight [8].

### 2.4.1 Υλοποίηση του σήματος "chirp" στη διαμόρφωση LoRa

Ένα chirp αποτελεί ένα σήμα, του οποίου η συχνότητα αυξάνεται ή μειώνεται με την πάροδο του χρόνου. Η συχνότητα μπορεί να μεταβάλλεται είτε γραμμικά είτε εκθετικά. Τυπικά παραδείγματα τέτοιου σήματος φαίνονται στις παρακάτω εικόνες:



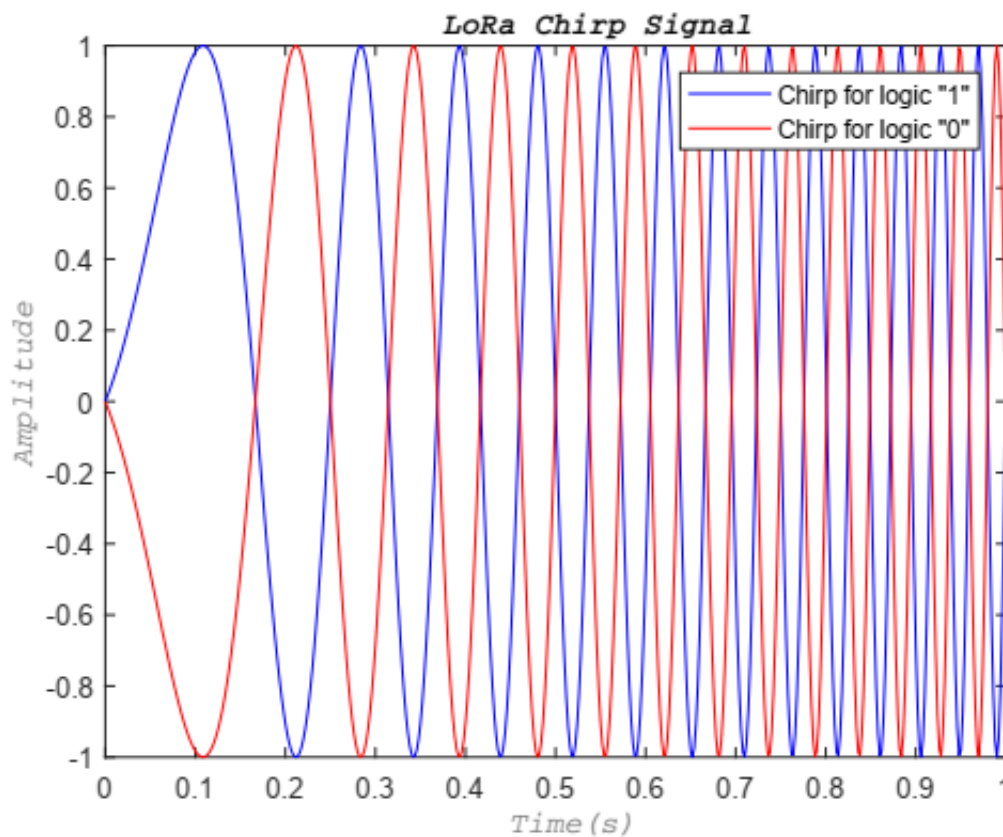
Εικόνα 6 – Γραμμικά μεταβαλλόμενο



Εικόνα 7 – Λογαριθμικά μεταβαλλόμενο

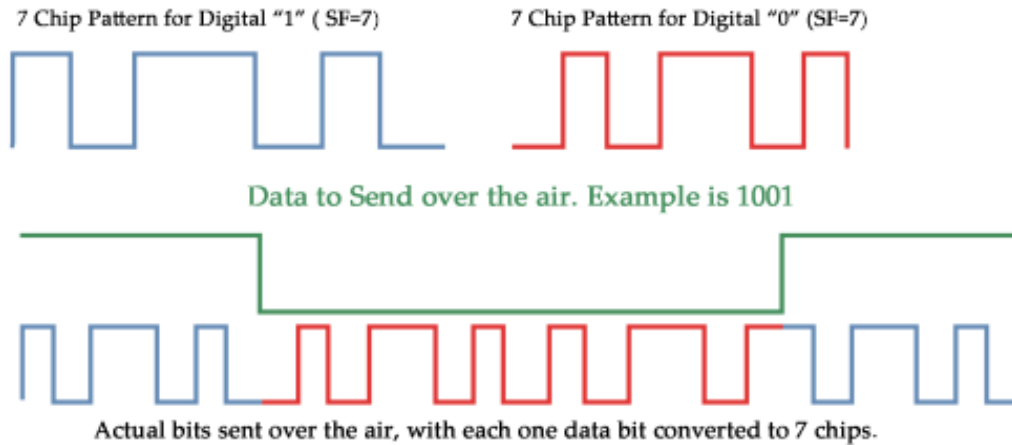
Τα δεδομένα κωδικοποιούνται σε ημιτονοειδές γραμμικά μεταβαλλόμενης συχνότητας παλμούς στην περίπτωση της διαμόρφωσης LoRa, το λογικό '1' και το λογικό '0' παριστάνονται από τον ίδιο σήμα με αντίθετη φάση. Ο γραμμικά μεταβαλλόμενης συχνότητας παλμός, σταθερής διάρκειας, μπορεί για τη διάδοση της πληροφορίας να καταλάβει ευρύτερο φάσμα

από ότι θα καταλάμβανε κανονικά. Η ομοιόμορφη κατανομή οποιουδήποτε συμβόλου σε μεγαλύτερο εύρος ζώνης παρέχει διάφορα πλεονεκτήματα. Πιο συγκεκριμένα, αφού χρησιμοποιεί όλο το διαθέσιμο εύρος ζώνης για τη μετάδοση του σήματος, το κάνει πιο ανθεκτικό στο θόρυβο του καναλιού. Επίσης, παρέχει αντίσταση σε συχνοτικά επιλεκτικό θόρυβο αλλά και σε κάθε είδους παρεμβολή, καθώς η περίοδος συμβόλων LoRa μπορεί να είναι μεγαλύτερη από αυτή άλλων τεχνικών διαμόρφωσης άλματος συχνότητας (frequency-hopping). Εν κατακλείδι, αποτελεί μία καλή λύση σε σχέση με τις τυπικές διαμορφώσεις διασποράς φάσματος (όπως Direct-sequence spread spectrum), καθώς επιτυγχάνει καλές επιδόσεις με χαμηλό κόστος και πολυπλοκότητα [9].



Εικόνα 8- Chirp Spread Spectrum

Σε επίπεδο "bit", η δυαδική ακολουθία πληροφορίας πολλαπλασιάζεται με μία ακολουθία που ονομάζεται chirp. Ο ρυθμός bit της ακολουθίας chirp είναι πολύ μεγαλύτερος από τον ρυθμό της ακολουθίας πληροφορίας, άρα κάθε σύμβολο πληροφορίας συνίσταται από πολλούς παλμούς chirp (Εικόνα 9). Ουσιαστικά, ο παλμός chirp αποτελεί την ακολουθία bit που "βγαίνει" από τον κωδικοποιητή [10].



Εικόνα 9 – Κωδικοποιημένη ακολουθία bit στο LoRa

#### 2.4.2 Παράγοντας Διασποράς (Spreading Factor) στη διαμόρφωση LoRa

Ο λόγος:

$$SF = \frac{R_{chip}}{R_s}$$

όπου το  $R_{chip}$  ορίζει τον ρυθμό του σήματος chip, ενώ το  $R_s$  τον ρυθμό του συμβόλου ονομάζεται Spreading Factor (SF). Η παράμετρος SF ονομάζεται παράγοντας διασποράς και αποτελεί πολύ σημαντικό στοιχείο στη διαμόρφωση LoRa. Ουσιαστικά αποτελεί τον αριθμό των κωδικοποιημένων bits (chip signal) ανά σύμβολο και παίρνει τιμές μεταξύ 7 μέχρι 12. [11]

Ο ρυθμός του σήματος chip είναι η πρώτη παράγωγος της συχνότητας του chirp παλμού και αποτελεί συνάρτηση τόσο του παράγοντα διασποράς όσο και του εύρους ζώνης. Πιο συγκεκριμένα δίνεται από την παρακάτω σχέση:

$$R_s = \frac{BW}{2^{SF}} \text{ symbols per sec}$$

Επομένως ο παράγοντας διασποράς είναι υπεύθυνος και για την διάρκεια του συμβόλου LoRa σύμφωνα με την αντιστροφή της παραπάνω σχέσης:

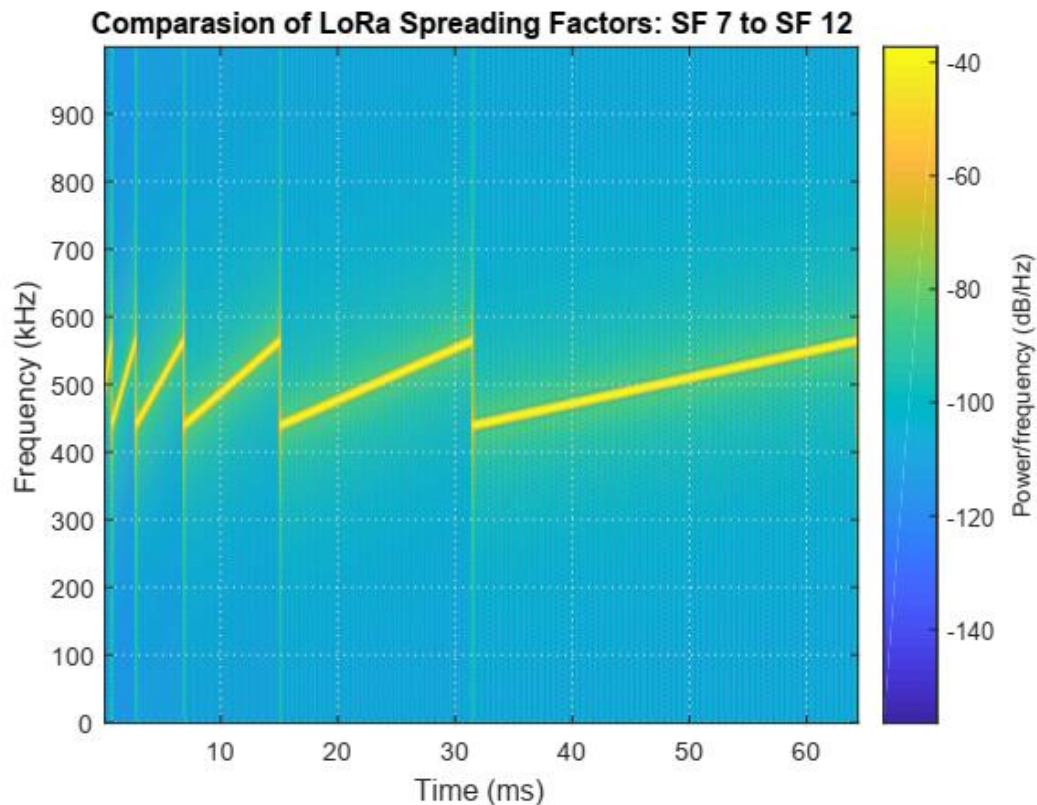
$$T_s = \frac{1}{R_{chirp}} = \frac{2^{SF}}{BW} \text{ secs}$$

Μπορούμε να ορίσουμε τελικά τον ρυθμό μετάδοσης chip:

$$R_c = R_s * 2^{SF} \text{ chips/sec}$$

Αυτό σημαίνει, υποθέτοντας ότι χρησιμοποιείται καθορισμένο εύρος ζώνης, κάθε αύξηση του παράγοντα διασποράς κατά 1 μονάδα έχει ως αποτέλεσμα διπλάσια αύξηση της διάρκειας

συμβόλου. Το παρακάτω σχήμα συγκρίνει τους διάφορους παράγοντες διασποράς στη διαμόρφωση LoRa:



Εικόνα 10 - Spectrogram of generated upchirp at 125KHz using MATLAB

Προκύπτει ο ορισμός « ...one chip is sent per second per Hz of bandwidth...»

$$R_c = R_s * 2^{SF} \Rightarrow R_c = \frac{BW}{2^{SF}} * 2^{SF} = BW$$

Η διαμόρφωση LoRa περιλαμβάνει επίσης ένα σχήμα μεταβλητής διόρθωσης σφαλμάτων που βελτιώνει την ευρωστία του μεταδιδόμενου σήματος σε βάρος του πλεονασμού.

Έτσι μπορούμε να ορίσουμε τον ονομαστικό ρυθμό μετάδοσης του σήματος δεδομένων ως

$$R_b = \frac{\frac{4}{4+CR}}{\frac{2^{SF}}{BW}} \text{ bits/sec}$$

Όπου :

- SF = spreading factor (7 .. 12)
- CR = code rate (1 ..4)
- BW = modulation bandwidth (Hz)

Όσο μεγαλύτερη είναι η διάρκεια συμβόλου, τόσο πιθανότερη είναι η αξιόπιστη μεταφορά της πληροφορίας, μιας και το διαμορφωμένο σήμα έχει υψηλότερη ανθεκτικότητα σε παρεμβολές και θόρυβο. Κατά συνέπεια, αύξηση του παράγοντα διασποράς αυξάνει την ευαισθησία του

δέκτη και μεγαλώνει την ακτίνα κάλυψης του τηλεπικοινωνιακού συστήματος. Παρ' όλα αυτά, θα πρέπει να αναλογιστούμε ότι όσο περισσότερα σύμβολα στέλνουμε, τόσο μεγαλύτερη η πιθανότητα να μεταδώσουμε λανθασμένη πληροφορία (αυξάνεται η πιθανότητα για λανθασμένη μετάδοση συμβόλων), αλλά και μεγαλύτερη πιθανότητα για να υπάρξουν τυχόν συγκρούσεις μηνυμάτων. Οπότε, γενικά προτείνεται η αποστολή μικρής έκτασης μηνυμάτων ακόμα και όταν επιλέγεται μεγάλος παράγοντας διασποράς. Ενδεικτικά παρουσιάζεται ένας πίνακας με τις διάφορες παραμέτρους της διαμόρφωσης LoRa που επηρεάζονται από τον παράγοντα διασποράς.

| Spreading Factor | Chips/symbol | SNR limit | Time-on-air (10 byte packet) | Bitrate  |
|------------------|--------------|-----------|------------------------------|----------|
| 7                | 128          | -7.5      | 56 ms                        | 5469 bps |
| 8                | 256          | -10       | 103 ms                       | 3125 bps |
| 9                | 512          | -12.5     | 205 ms                       | 1758 bps |
| 10               | 1024         | -15       | 371 ms                       | 977 bps  |
| 11               | 2048         | -17.5     | 741 ms                       | 537 bps  |
| 12               | 4096         | -20       | 1483 ms                      | 293 bps  |

Εικόνα 11 -Παράμετροι διαμόρφωσης LoRa

Αξίζει να σημειωθεί, ότι ο παράγοντας διασποράς πρέπει να είναι γνωστός τόσο στο δέκτη, όσο και στον πομπό, διότι διαφορετικοί παράγοντες διασποράς είναι ορθογώνιοι μεταξύ τους. Επίσης, η σηματοθορυβική σχέση SNR(Signal to noise) που φαίνεται στον πίνακα είναι η ελάχιστη δυνατή, από την οποία ο δέκτης θα μπορούσε να λάβει ικανοποιητικά το σήμα, ώστε να προχωρήσει στην αποδιαμόρφωσή του.

Τέλος, ιδιαίτερα σημαντικό στοιχείο μεταξύ πομπού και δέκτη είναι ο σωστός συγχρονισμός των δύο. Εξαιτίας της γραμμικότητας των chirp παλμών, συχνοτικές αντισταθμίσεις (frequency offsets) μεταξύ του πομπού και του δέκτη είναι ισοδύναμες με αντίστοιχες χρονικές αντισταθμίσεις (timing offsets), οι οποίες εύκολα εξαλείφονται στον αποδιαμορφωτή. Επομένως, η διαμόρφωση είναι εξίσου καλή στην αντιμετώπιση του φαινομένου Doppler.

### 2.4.3 Θεωρητικός Υπολογισμός Ευαισθησίας και Link Budget

Η ευαισθησία (sensitivity) του δέκτη ορίζεται από την παρακάτω σχέση:

$$S = N_t + 10\log_{10}(BW) + NF + SNR \text{ dBm}$$

Ο όρος  $N_t$  αποτελεί το θερμικό θόρυβο στον δέκτη και δίνεται από τη σχέση:

$$N_t = 10\log_{10}(1000kT) + 10\log_{10}(\Delta_f) \text{ dBm}$$

Όπου :

$K$  – σταθερά Boltzmann

$T$  – θερμοκρασία δωματίου σε βαθμούς Kelvin

$\Delta f$  – εύρος ζώνης θορύβου

Τυπική τιμή που χρησιμοποιείται συχνά στα ηλεκτρονικά του δέκτη είναι για εύρος ζώνης 1 Hz με  $N_f = -174$  dBm. Ο δεύτερος όρος στην εξίσωση  $10\log_{10}(BW)$  εμπεριέχει την μείωση της ευαισθησίας λόγω του απαιτούμενου εύρους ζώνης για τη μετάδοση της πληροφορίας. Ο όρος  $N_f$  αποτελεί την εικόνα θορύβου (Noise Figure) η οποία είναι συγκεκριμένη για κάθε "ηλεκτρονικό" σύστημα. Τέλος, ο όρος SNR αποτελεί το απαιτούμενο SNR στον αποδιαμορφωτή για αξιόπιστη λήψη. Αν αναλογιστεί κανείς ότι συνήθεις τιμές ευαισθησίας για δέκτες είναι περί τα -90 έως -100 dBm, η διαμόρφωση LoRa πετυχαίνει έως και -148 dBm. Θέλοντας να εντάξουμε και το Link Budget στην ανάλυσή μας, το οποίο αποτελεί μία μέτρηση που λαμβάνει υπόψη της όλα τα κέρδη και όλες τις απώλειες της τηλεπικοινωνιακής ζεύξης, θα έχουμε μία σχέση της μορφής:

$$P_{RX}(dBm) = P_{TX}(dBm) + G_{system}(dB) - l_{system}(dB) - l_{channel}(dB) - M(dB)$$

Όπου  $P_{RX}$  συμβολίζει την αναμενόμενη ισχύ στο Δέκτη,  $P_{TX}$  την ισχύ του πομπο συστήματος, το  $G_{system}(dB)$  διάφορα κέρδη του συστήματος που σχετίζονται με κατευθυντικές κεραίες κ.λπ., το  $l_{system}(dB)$  λοιπές απώλειες του συστήματος λόγω απωλειών στις γραμμές μεταφοράς,  $l_{channel}(dB)$  απώλειες καναλιού και τέλος το  $M$  αποτελεί το Fading Margin.

Θα μπορούσαμε να εντάξουμε όλα τα κέρδη και τις απώλειες της προηγούμενης σχέσης που αφορούν τον δέκτη στον όρο της ευαισθησίας και να έχουμε μία πιο βολική σχέση της μορφής:

$$Link\ budget = P_{TX} - (Sensitivity)$$

Λαμβάνοντας υπόψη, ότι η μέγιστη επιτρεπόμενη ισχύς μετάδοσης είναι στα +20 dBm, τότε επιτυγχάνεται 168 dB μέγιστο Link budget.

Στην πραγματικότητα όμως, για τεχνολογίες όπως η LoRa, η ευαισθησία του δέκτη και κατ' επέκταση το Link Budget έχουν μικρότερη επίπτωση στην ποιότητα υπηρεσιών Quality of Service (QoS) σε σχέση με τις πιθανές παρεμβολές. Δεν πρέπει να ξεχνάμε ότι για συχνότητες στις οποίες λειτουργούν τα LPWANs δεν απαιτείται αδειοδότηση για την χρήση τους, με αποτέλεσμα να μην υπάρχει εγγυημένο QoS.

#### 2.4.4 Τεχνικές κωδικοποίησης της διαμόρφωσης LoRa

Για την μετάδοση της πληροφορίας τα δεδομένα υπόκεινται σε μια σειρά από τεχνικές κωδικοποίησης, ώστε να διασφαλιστεί η αξιόπιστη μεταφορά τους. [8] Αυτές μπορούν να συμπτυχθούν παρακάτω:

**Data Whitening:** είναι ένας γραμμικός μετασχηματισμός που εφαρμόζεται στα δεδομένα με σκοπό την τυχαιοποίηση αυτών. Τα δεδομένα (σ.σ. σύμβολα) έχουν στην αρχική τους

μορφή αυτοσυσχέτιση μεταξύ τους, την οποία άρει αυτός ο μετασχηματισμός μετατρέποντάς τα σε ασυσχέτιστα. Επομένως, με τη βοήθεια μίας whitening ακολουθίας (τυχαία ακολουθία), η οποία είναι γνωστή τόσο στον δέκτη όσο και στον πομπό, η πληροφορία κατανέμεται ομοιόμορφα σε όλο το εύρος ζώνης του καναλιού.

**Gray Indexing:** τα κωδικοποιημένα σύμβολα διατάσσονται κατά τέτοιον τρόπο, ώστε δύο διαδοχικά σύμβολα να διαφέρουν κατά μέγιστο 1 bit. Με αυτόν τον τρόπο, αποτρέπονται off-by-one σφάλματα κατά την ανάκτηση των συμβόλων, συνεπώς αυξάνονται οι πιθανότητες να διορθωθούν σφάλματα από τον κώδικα του καναλιού.

**Forward Error Correction (FEC):** η προς τα εμπρός διόρθωση σφαλμάτων αποτελεί από τις πιο διαδεδομένες τεχνικές για τον έλεγχο σφαλμάτων κατά την ψηφιακή μετάδοση της πληροφορίας σε ένα θορυβώδες τηλεπικοινωνιακό κανάλι. Στη διαμόρφωση LoRa εφαρμόζεται με τη μορφή κώδικα Hamming(M,N). Για την ακρίβεια, το μήκος της λέξης του κώδικα πληροφορίας (μήκους N) είναι συγκεκριμένο και ίσο με 4 bits, ενώ η κωδική λέξη (μήκους M) αποτελεί παράμετρος που ρυθμίζεται από το σύστημα με μήκος [5:8] bits. Οπότε, λόγω της συγκεκριμένης απαραίτητης υλοποίησης επηρεάζεται ο ρυθμός bit του σήματος πληροφορίας και δίνεται σύμφωνα με τη σχέση:

$$R_b = \frac{4}{\frac{4+CR}{2^{SF}} BW} \text{ bits/sec}$$

**Interleaving:** Η χρήση αναδιατάκτη στις ψηφιακές τηλεπικοινωνίες είναι συχνή, διότι βελτιώνει την επίδοση των FEC κωδίκων. Ουσιαστικά, αναδιατάσσει την έξοδο του FEC κωδικοποιητή, καθιστώντας τον κώδικα πιο ανθεκτικό σε 'λάθη κατά ριπές (burst errors). Τα σφάλματα αυτά εμφανίζονται κατά ομάδες σε μεμονωμένα χρονικά διαστήματα και δεν είναι ίσο-κατανεμημένα με τυχαίο τρόπο σε όλη τη διάρκεια του μηνύματος. Άρα η χρήση ενός αναδιατάκτη που αναδιατάσσει τα bits πληροφορίας μπορεί να αποσυσχετίσει το μήνυμα από το χρόνο, δηλαδή τα λάθη που προκύπτουν σε χρονικά συνεχόμενα bits να μην είναι συνεχόμενα μέσα στο πραγματικό σήμα. Στη διαμόρφωση LoRa χρησιμοποιείται διαγώνιος αναδιατάκτης με τα δύο πιο σημαντικά δυαδικά ψηφία να αντιστρέφονται. Κάθε διαγώνια λέξη πληροφορίας είναι μετατοπισμένη ή έχει περιστραφεί από έναν αυθαίρετο αριθμό bits, ενώ τα bits μέσα σε κάθε κωδική λέξη είναι ανεστραμμένα.

#### 2.4.5 Μορφή των πακέτων LoRa

Η δομή του πακέτου LoRa έχει δύο βασικές μορφές: την explicit και την implicit [11]. Κυρίαρχη διαφορά αυτών των δύο, είναι ότι στην explicit μορφή, το πακέτο συμπεριλαμβάνει μία μικρή κεφαλίδα (header), που περιέχει πληροφορίες σχετικά με τον αριθμό των bytes, το coding rate αλλά και την ύπαρξη ή όχι ενός 16-bits CRC (Cyclic redundancy check) για το Payload. Αντίθετα, στην implicit μορφή, στην οποία είναι γνωστά όλα τα παραπάνω εκ των

προτέρων στον δέκτη (έχει φροντίσει γι' αυτό ο σχεδιαστής), η κεφαλίδα δεν είναι απαραίτητη και αφαιρείται. Με αυτόν τον τρόπο μειώνεται ο χρόνος μετάδοσης, δηλαδή η διάρκεια του συμβόλου.

| Preamble          | Mandatory Preamble | if Implicit Header disabled    |                | Payload       | if Uplink   |
|-------------------|--------------------|--------------------------------|----------------|---------------|-------------|
|                   |                    | PHY Header                     | PHY Header CRC |               | CRC         |
| 6 - 65535 symbols | 4.25 symbols       | 0 – 3 bytes (encoded in CR4/8) |                | 0 – 255 bytes | 0 – 2 bytes |

Εικόνα 12 – Πακέτο LoRa

Η κύρια χρήση του προοιμίου (Preamble) είναι ο σωστός συγχρονισμός του δέκτη με την εισερχόμενη ροή δεδομένων. Το payload είναι ένα πεδίο μεταβλητού μήκους, το οποίο περιέχει τα κωδικοποιημένα δεδομένα, όπως αυτά έχουν καθοριστεί στην κεφαλίδα. Μπορούμε να καταλάβουμε ότι ο χρόνος διάρκειας της μετάδοσης (time on air) ενός LoRa πακέτου, συνδέεται άμεσα με τον συντελεστή διασποράς SF, το ρυθμό του κώδικα CR (coding rate) και το εύρος ζώνης. Ο χρόνος μετάδοσης είναι ένα ενδιαφέρον στοιχείο στη διαμόρφωση LoRa, διότι όπως θα δούμε υπάρχουν ορισμένοι περιορισμοί που εντάσσει το πρωτόκολλο LoRaWAN. Η διάρκεια ενός LoRa πακέτου υπολογίζεται ως εξής:

$$t_{packet} = t_{preamble} + t_{payload}$$

$$t_{preamble} = (n_{preamble} + 4) \cdot T_s$$

$$t_{payload} = n_{payload} \cdot T_s$$

όπου το  $n_{preamble}$  ορίζει το μήκος του preamble, αποτελεί παράμετρος που μπορεί να ρυθμιστεί από τον σχεδιαστή και επηρεάζει τον αριθμό των συμβόλων στο preamble ενώ το  $n_{payload}$  εξαρτάται από την μορφή της κεφαλίδας και αποτελεί τον αριθμό των payload συμβόλων. Ο υπολογισμός του payload είναι αρκετά περίπλοκος και δίνεται από τη σχέση:

$$n_{payload} = 8 + \max \left( \text{ceil} \left[ \frac{8PL - 4SF + 44 - 20H}{4(SF - 2DE)} \right] (CR + 4), 0 \right)$$

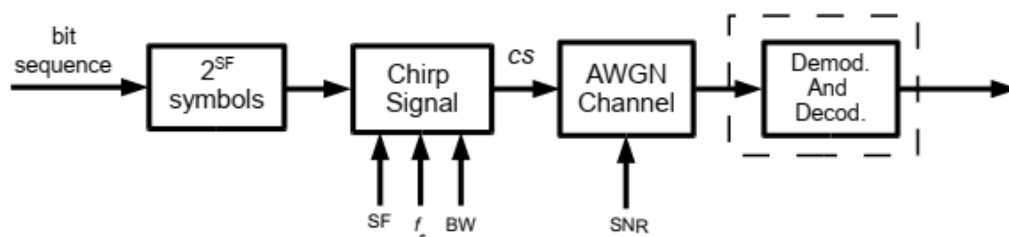
με το PL να συμβολίζει τον αριθμό των payload bytes, το SF τον παράγοντα διασποράς, το H λαμβάνει είτε την τιμή 0 όταν η δομή του πακέτου είναι σε explicit μορφή είτε την τιμή 1 όταν είναι σε implicit μορφή, το DE λαμβάνει είτε την τιμή 0 όταν ο αλγόριθμος βελτιστοποίησης αργού data rate είναι ενεργοποιημένος (LowDataRateOptimize=0) είτε την τιμή 1 διαφορετικά και τέλος CR τον αριθμό των bit ισοτιμίας (parity bits) που προστίθενται. Η βελτιστοποίηση Low Data Rate είναι μία μέτρηση που γίνεται για τον ορθό συγχρονισμό μεταξύ πομπού και δέκτη κατά την αποστολή μεγάλης διάρκειας συμβόλων.

Ένα τυπικό παράδειγμα ενός LoRa πακέτου φαίνεται στην εικόνα (Εικόνα 9), η οποία δείχνει ένα spectrogram με τον χρόνο στον οριζόντιο και την συχνότητα στον κατακόρυφο άξονα. Μπορεί κανείς να παρατηρήσει ότι το μήνυμα LoRa στο φυσικό επίπεδο, αποτελείται από

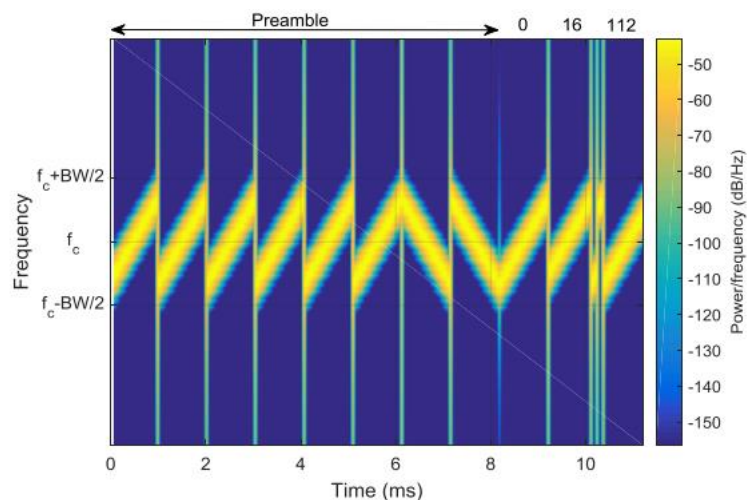
σήματα chirp, τα οποία σαρώνουν την μπάνα συχνοτήτων. Μετά από μερικές επαναλήψεις της συγκεκριμένης συχνότητας (που σαρώνεται), η οποία αποτελεί ουσιαστικά το preamble του μηνύματος πληροφορίας, τα δεδομένα κωδικοποιούνται στο σήμα ως απότομες αλλαγές στη συχνότητα του σήματος chirp ή έλλειψης αυτής. Η πιο ευρέως διαδεδομένη διαδικασία αποδιαμόρφωσης ακολουθεί την εξής λογική:

1. Προσδιορισμός της αρχής του πακέτου, δηλαδή της αρχής του preamble.
2. Προσδιορισμός της αρχής της κωδικοποιημένης πληροφορίας
3. Εξαγωγή της πληροφορίας από τις διαφορετικές μεταβάσεις της φάση των σημάτων chirp

Πιο συγκεκριμένα, αρχικά το σήμα υπόκειται σε μία de-chirping διαδικασία και στη συνέχεια εφαρμόζεται FFT στο de-chirped σήμα (). Ο αριθμός των frequency bins ισούται με τον αριθμό των συμβόλων  $M$  που αντιστοιχεί στον παράγοντα διασποράς (SF) που έχει χρησιμοποιηθεί. Το de-chirped σήμα μπορεί πλέον να αποδιαμορφωθεί σύμφωνα με τις γνωστές τεχνικές αποδιαμόρφωσης Multiple frequency-shift keying (MFSK). Η διαδικασία δεν είναι τόσο εύκολη όσο φαίνεται, διότι χρειάζεται να εφαρμοστούν πολλαπλοί overlapping FFTs προκειμένου να είναι ξεκάθαρη η ισχύς των bins με αποτέλεσμα την καθαρή ανίχνευση των συμβόλων.



Εικόνα 13- de-chirping



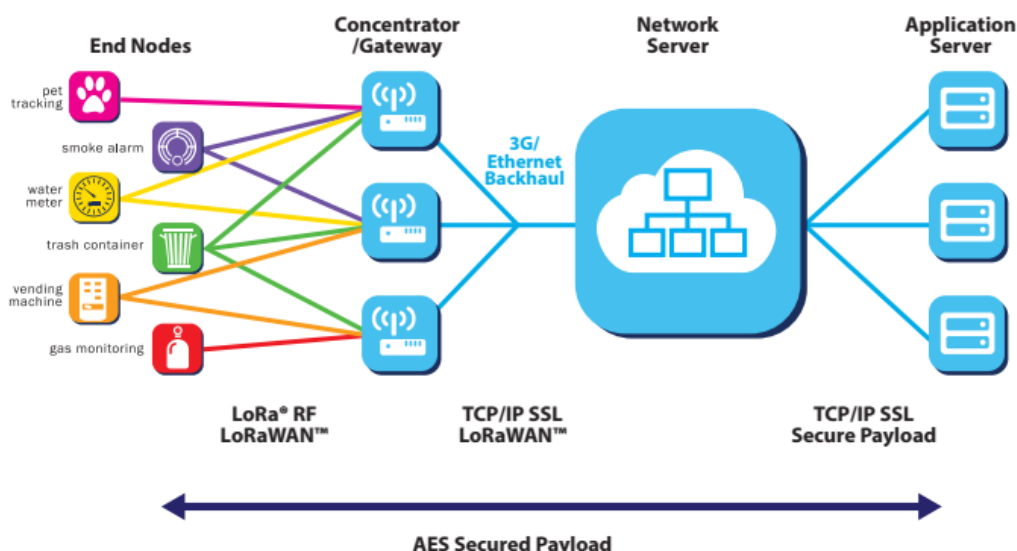
Εικόνα 14 – Spectrogram SF=7

## 2.5 Το πρότυπο *LoRaWAN*

Το πρότυπο LoRaWAN αποτελεί πρωτόκολλο δικτύου για την βέλτιστη ενεργειακή κατανάλωση τελικών κόμβων End-Nodes (ENs). Η βελτιστοποίηση γίνεται, σε τερματικούς κόμβους που τροφοδοτούνται από μπαταρία και έχει ως σκοπό την χαμηλότερη δυνατή κατανάλωση ενέργειας. Σε αντίθεση με την τεχνολογία LoRa-PHY, το πρότυπο LoRaWAN είναι ανοιχτό και έχει συνταχθεί από μία ομάδα εταιριών και ερευνητικών κέντρων, την LoRa Alliance [6]. Στη συνέχεια, θα περιγράψουμε σύντομα βασικές πτυχές του πρωτοκόλλου και πως αυτές επηρεάζουν τη συνολική λειτουργία του δικτύου.

### 2.5.1 Τοπολογία και στοίβα πρωτοκόλλου του Δικτύου *LoRaWAN*

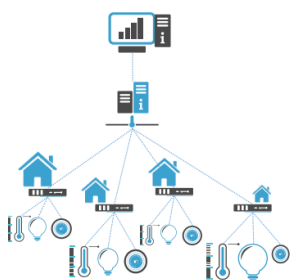
Τα δίκτυα LoRa WAN βρίσκονται συνήθως σε τοπολογία αστέρα ( stars-of-stars topology), διότι αποτελεί εύκολα διαχειρίσιμη λειτουργικά τοπολογία και καθότι καθίσταται δυνατή η επέκταση του δικτύου με πολύ απλό τρόπο. Αυτό δεν σημαίνει ότι δεν υπάρχουν περιπτώσεις LoRaWANs που βρίσκονται σε διαφορετική τοπολογία δικτύου, όπως πλέγματος (mesh topology), με την λύση αυτή να είναι εξαιρετικά πολύπλοκη και αρκετά ασύμφορη. Η τυπική μορφή του δικτύου αποτελείται από τερματικούς κόμβους, οι οποίοι στέλνουν/λαμβάνουν μηνύματα από ένα ή περισσότερα Gateways (GWs) (γνωστά και ως Σταθμοί Βάσης, Κεντρικοί Κόμβοι ή και Concentrators), τα οποία με τη σειρά τους μεταφέρουν αυτά τα μηνύματα σε έναν κεντρικό Network Server (NS). Στην εικόνα ( Εικόνα 15) μπορούμε να δούμε σε πολύ απλή μορφή την τοπολογία του δικτύου.



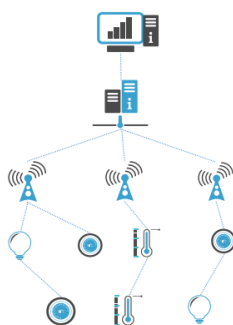
Εικόνα 15- Τοπολογία LoRaWAN

Παράλληλα, η ύπαρξη ενδιάμεσων σταθμών αναμετάδοσης αποφεύγεται, διότι επιβάλλει πολυπλοκότητα και κόστος. Οι τερματικές συσκευές δεν συνδέονται αποκλειστικά με έναν

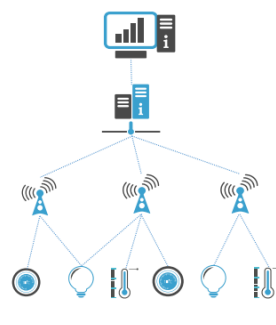
κεντρικό κόμβο, αντ' αυτού τα μηνύματα στέλνονται σε κάποιο τηλεπικοινωνιακό κανάλι, υποθέτοντας ότι τουλάχιστον ένας θα τα παραλάβει και θα τα προωθήσει στον server. Το κεντρικό σύστημα (ή αλλιώς Network Server στην περίπτωση μας), είναι υπεύθυνο για την αξιοποίηση των μηνυμάτων, τυχόν πολλαπλές αφίξεις μηνυμάτων από διαφορετικούς σταθμούς βάσης ή από ποιον σταθμό θα σταλεί κάποιο downlink μήνυμα κ.ο.κ. Τέλος, διαφορετικά τηλεπικοινωνιακά λογικά κανάλια ορίζονται σε όλο το δίκτυο, η επιλογή των οποίων γίνεται από τους ENs με έναν ψευδοτυχαίο (pseudo-random) τρόπο σε κάθε μετάδοση πακέτου.



Εικόνα 16 - Star

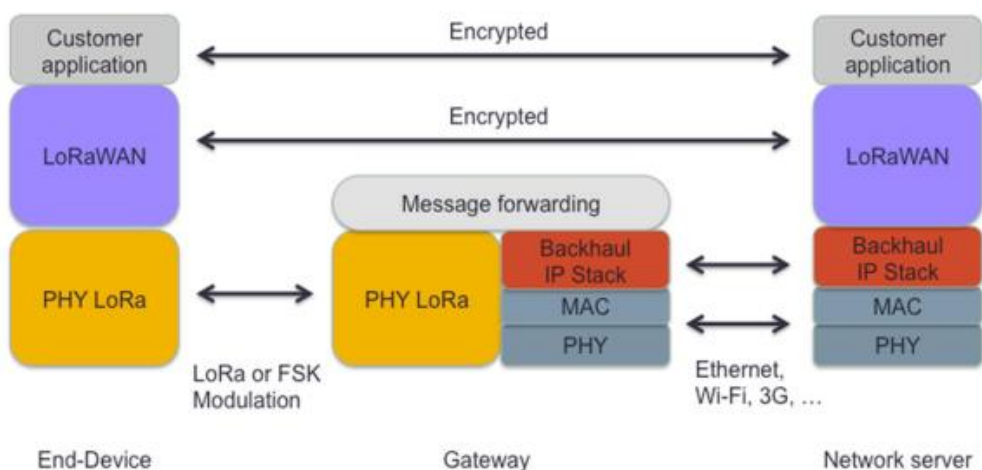


Εικόνα 17 - Mesh



Εικόνα 18- Bridge

Η στοιβά πρωτοκόλλου του δικτύου LoRaWAN φαίνεται στην εικόνα (Εικόνα 19). Αξίζει να αναφέρουμε ότι παρόλο που τόσο οι τερματικοί κόμβοι, όσο και ο Network Server έχουν στα υψηλότερα επίπεδα κάποιου είδους Application layer, οι σταθμοί βάσης ασχολούνται αποκλειστικά με την προώθηση των μηνυμάτων μεταξύ των δύο παραπάνω, το οποίο υλοποιείται κυρίως σε επίπεδο firmware. Επομένως, υπάρχει μία στενή σύνδεση των τερματικών συσκευών/ και του server, κάτι το οποίο θα παρατηρήσουμε και παρακάτω.



Εικόνα 19 – LoRaWAN Layers

## 2.5.2 Κλάσεις του δικτύου LoRaWAN

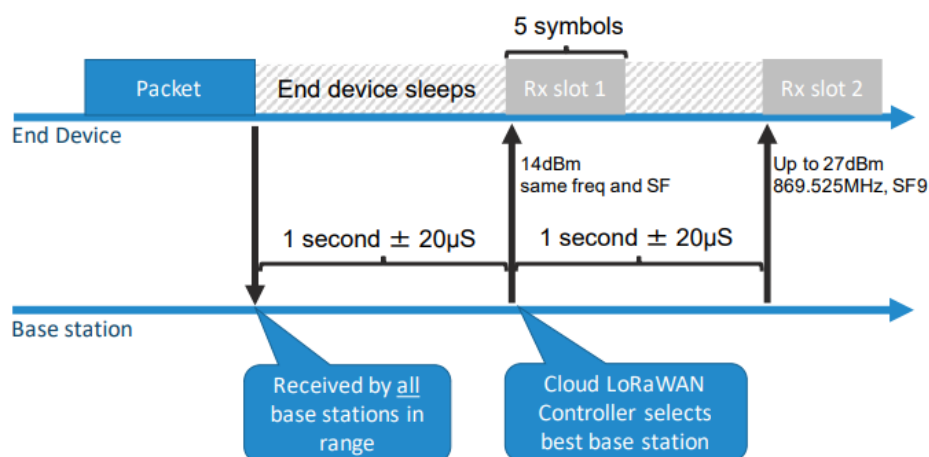
Για να ακολουθούν οι ENs το πρωτόκολλο LoRaWAN, θα πρέπει να εντάσσονται σε κάποιες Κλάσεις (Classes). Πιο συγκεκριμένα όλες οι συσκευές ENs είναι συμβατές με την Κλάση A

και ανάλογα με τη λειτουργικότητά τους, μπορούν να ενταχθούν σε διαφορετικές Κλάσεις, την B ή την C. Κάθε κλάση επιβάλλει αμφίδρομη επικοινωνία μεταξύ GW και EN, ενώ διαφορετική ενεργειακή αποδοτικότητα επιτυγχάνεται ανάλογα με την επιλογή Κλάσης. Αναλυτικότερη περιγραφή των κλάσεων ακολουθεί παρακάτω:

- **Class A:** Η βασική Κλάση A αποτελεί τον κατεξοχήν τρόπο λειτουργίας των ENs. Στην συγκεκριμένη λειτουργία κάθε EN στέλνει την πληροφορία με έναν τυχαίο και ασύγχρονο τρόπο όταν αυτή υπάρχει, υλοποιώντας έναν Aloha τρόπο πρόσβασης σε κοινό κανάλι. Με το πέρας κάθε uplink μετάδοσης (άνω ζεύξης), "ανοίγουν" δύο μικρά παράθυρα λήψεως (με χρονική διάρκεια περί τα 20 ms), περιμένοντας οποιαδήποτε απόκριση από τον NS (downlink, κάτω ζεύξη), είτε κάποια εντολή αποδοχής του αιτήματος για αποστολή μηνύματος, είτε κάποιο πακέτο που θα θελήσει να επιστρέψει ο NS. Το πρώτο παράθυρο ανοίγει στο ίδιο ακριβώς κανάλι της uplink μετάδοσης, ενώ το δεύτερο σε κάποιο διαφορετικό, το οποίο είναι στην ευχέρεια του NS να το ορίσει κατάλληλα. Ο λόγος του δεύτερου παραθύρου είναι κυρίως να αυξηθεί η αξιοπιστία της μετάδοσης, απέναντι στις διάφορες διακυμάνσεις του καναλιού. Αξίζει να σημειωθεί ότι δεν υπάρχει κάποιος μηχανισμός αποστολής της πληροφορίας, αν δεν πραγματοποιηθεί το downlink, δηλαδή αν δεν επιτύχει η αμφίδρομη επικοινωνία σε κάποιο από τα παράθυρα λήψεως. Η συγκεκριμένη Κλάση A συνιστά την πιο ενεργειακά αποδοτική κλάση, τον EN με την χαμηλότερη ισχύ, επομένως θα πρέπει να κρατά ανενεργό (να μη καταναλώνει ενέργεια με κάποιο τρόπο, είτε βρίσκεται σε sleep mode, είτε σε κάποια μορφή off mode) τον δέκτη (συσκευή EN) όσο το δυνατό περισσότερο.

**Class A** (available since LoRaWAN 1.0)

The below figure illustrates default configuration in LoRaWAN standard in SF12. Values can be adjusted.

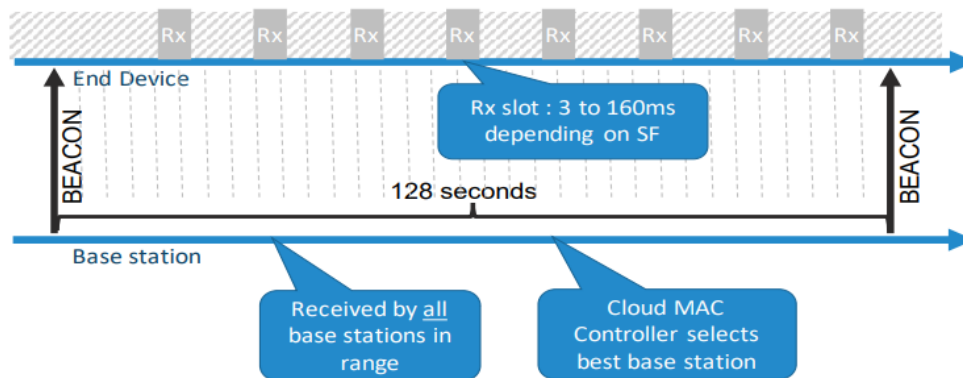


Εικόνα 20 – Class A

- **Class B:** Στην Κλάση B, οι ENs ανοίγουν επιπλέον παράθυρα λήψεως σε προκαθορισμένες-προγραμματισμένες χρονικές στιγμές, σε αντίθεση με την Κλάση A

η οποία έχει τυχαία παράθυρα λήψεως. Πιο συγκεκριμένα, οι ENs συγχρονίζονται με τον NS, όταν ο NS αποστέλλει μέσω του GW πιλοτικά σήματα (beacons), με σκοπό να γίνει η αμφίδρομη επικοινωνία και να ανοίξουν τα αντίστοιχα παράθυρα λήψεως. Ο μηχανισμός του beacon λειτουργεί ανεξάρτητα από το uplink, το οποίο καθιστά τις συσκευές που χρησιμοποιούν την Κλάση B ιδανικές για λειτουργίες όπως ενεργοποιητές (actuators) ή διακόπτες (switches). Εύκολα μπορεί κανείς να καταλάβει ότι οι ENs της Κλάσης B καταναλώνουν περισσότερη ισχύ από αυτή της Κλάσης A και αποτελούν καλή λύση για ενεργοποιητές που χρησιμοποιούν κάποιου είδους μπαταρία.

The below figure illustrates default configuration in LoRaWAN standard in SF12. Values can be adjusted.

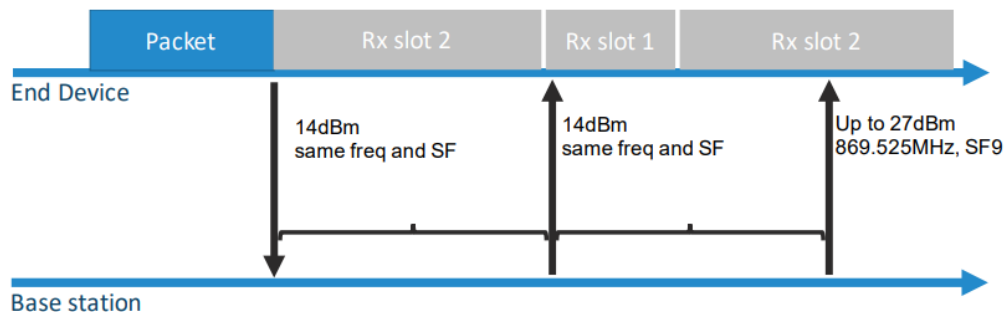


Εικόνα 21- Class B

- Class C: Τέλος στην Κλάση C, οι ENs βρίσκονται σε συνεχή επαγρύπνηση για να δεχτούν κάποιο πακέτο από τον NS, μέσω του GW, εκτός αν εκείνη τη στιγμή μεταδίδουν κάποια μορφή πακέτου. Στη συγκεκριμένη υλοποίηση των EN, δεν υπάρχουν αυστηροί περιορισμοί όσον αφορά την ενεργειακή κατανάλωση, μιας και οι συσκευές αυτού του τύπου δεν μπορούν να βρεθούν σε κάποιου είδους sleep mode. Συνεπώς, έχουν σχεδόν συνέχεια ανοιχτά παράθυρα λήψεως με αποτέλεσμα να υπάρχει μικρότερη χρονική καθυστέρηση (latency) στην επικοινωνία μεταξύ του NS και των ENs. Τέτοιου είδους ENs χρησιμοποιούνται σε οποιαδήποτε εφαρμογή χρειάζεται πολύ μικρό latency και η ενεργειακή κατανάλωση δεν αποτελεί σημαντικό παράγοντα.

**Class C** (available since LoRaWAN 1.0)

The below figure illustrates default configuration in LoRaWAN standard in SF12. Values can be adjusted.



Εικόνα 22- Class C

## 2.6 Δομή Πακέτου και MAC εντολές του δικτύου LoRaWAN

Μεγάλο μέρος του πρωτοκόλλου LoRaWAN αποτελούν οι MAC εντολές. Ουσιαστικά, η δομή ενός πακέτου συνίσταται από τη μορφή του PHY που εξηγήθηκε προηγουμένως και ένα σύνολο από παραμέτρους του δικτύου LoRaWAN (όπως η επιλογή του SF), που αποτελούν τις εντολές MAC. Η τυπική δομή του πακέτου (Frame) καθώς και το πεδίο με τις εντολές MAC δίνονται στην εικόνα (Εικόνα 23). Οι πληροφορίες που περιέχουν οι MAC εντολές σχετίζονται με τον τύπο του μηνύματος, αλλά και τον ίδιο τον EN, ενώ οι περισσότερες αποτελούν μεταβλητές παράμετροι του συστήματος. Η επεξεργασία των παραμέτρων MAC γίνεται στο κομμάτι του NS (και κατ' επέκταση πολλές αλλαγές σε αυτές), γι' αυτό θα ακολουθήσει συνοπτική περιγραφή, με βάση την εικόνα (Εικόνα 23). Το μέρος τους πακέτου που περιέχει το MAC payload περιέχει τα εξής subframes: Frame header, Frame port και Frame payload. Στο Frame header περιέχονται διάφορα πεδία που σχετίζονται με διαφορετικές πτυχές του δικτύου LoRa. Πιο συγκεκριμένα, αρχικά υπάρχει ένα subframe: DevAddr που αποτελεί μία 4-bytes αλληλουχία, για την αναγνώριση του NS στο οποίο ανήκει ο EN. Στη συνέχεια, παρατηρούμε τα πεδία του Frame Control (1-byte), Frame Counter (2-bytes) και Frame Options(16-bytes) τα οποία εμπεριέχουν διάφορες εντολές MAC, που συνεχώς ανταλλάσσονται μεταξύ του NS και του EN. Ακόμα, το πεδίο Frame payload περιέχει δεδομένα που προέρχονται από το Application Layer, ενώ το πεδίο Frame port είναι προαιρετικό και χρησιμοποιείται αποκλειστικά για την εφαρμογή που προορίζεται το μήνυμα.

Οι εντολές MAC που ανταλλάσσονται μεταξύ EN και NS αφορούν διάφορες λειτουργίες του δικτύου. Οι πιο σημαντικές λειτουργίες έχουν να κάνουν με:

- Προσδιορισμό της απόστασης GW-EN. Κάθε EN μπορεί να ζητήσει από τον NS το Link margin, δηλαδή την ληφθείσα ισχύ στον GW.

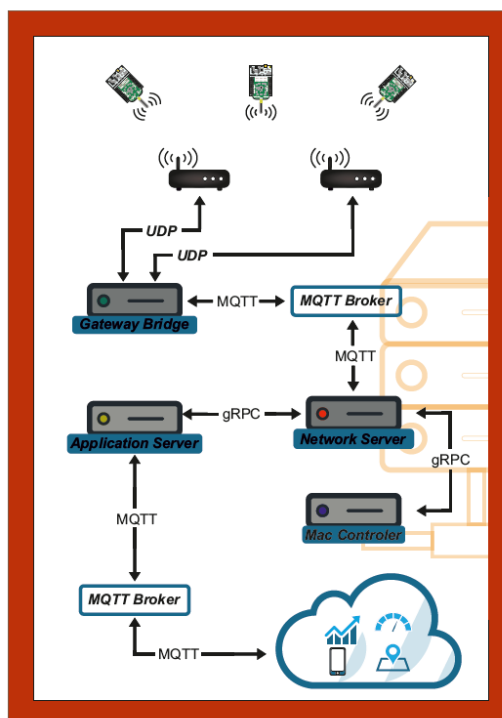
- Δεδομένα για την εφαρμογή αλγορίθμων βελτιστοποίησης, όπως ο ADR algorithm (Adaptive Data Rate).
- Περιορισμούς που θέτει ο NS που αφορούν το Duty Cycle, το οποίο παίζει πολύ σημαντικό ρόλο στην τεχνολογία.
- Ρυθμίσεις παραμέτρων που σχετίζονται με τα παράθυρα λήψεως.
- Δεδομένα, τα οποία δίνουν πληροφορίες για την κατάσταση του EN σε πραγματικό χρόνο, όπως στάθμη μπαταρίας και demodulation margin.
- Τυχόν δημιουργία νέων καναλιών επικοινωνίας, με σκοπό μία πιο αξιόπιστη μετάδοση πληροφορίας.

|          |      |          |                          |            |  |  |                  |  |     |     |
|----------|------|----------|--------------------------|------------|--|--|------------------|--|-----|-----|
|          |      |          | DevAddr FCtrl FCnt FOpts |            |  |  |                  |  |     |     |
| Preamble | PHDR | PHDR_CRC |                          | FHDR       |  |  | FPort FRMPayload |  | CRC |     |
| Preamble | PHDR | PHDR_CRC | MHDR                     | MACPayload |  |  |                  |  | MIC | CRC |
| Preamble | PHDR | PHDR_CRC | PHYPayload               |            |  |  |                  |  | CRC |     |

Εικόνα 23- Δομή πακέτου

Από όλα τα παραπάνω φαίνεται η στενή σχέση που υπάρχει μεταξύ των MAC εντολών και του NS. Οποιαδήποτε εντολή υπάρχει στο firmware/software του EN, προϋποθέτει αντίστοιχες εντολές στο κομμάτι του NS, ώστε να μπορούν να λειτουργήσουν σωστά.

## 2.7 LoRa Server

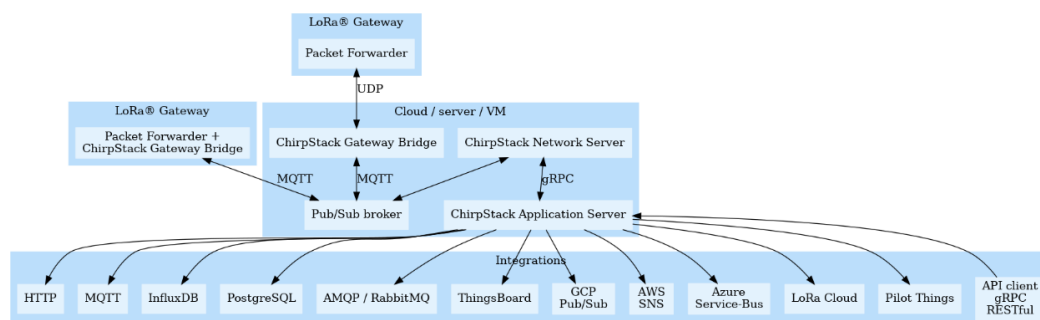


Εικόνα 24- Αρχιτεκτονική LoRa Server

Ο LoRa server αποτελείται από ένα πακέτο εφαρμογών, με σκοπό τον έλεγχο και την εποπτεία των δεδομένων σε όλη την διαδρομή από το gateway μέχρι την τελική εφαρμογή που τα δέχεται, αλλά και την εξυπηρέτηση του χρήστη σε ότι αφορά την πρόσβαση στα δεδομένα είτε στην uplink είτε στην downlink επικοινωνία. Παρέχει μηχανισμούς για την διαχείριση των gateways, των εφαρμογών που αναπτύσσονται στο δίκτυο, αλλά και των τερματικών συσκευών που σχετίζονται με αυτές. Στην εικόνα (Εικόνα 19), φαίνεται η αρχιτεκτονική του LoRa server με τα δομικά στοιχεία που το αποτελούν, καθώς και της μεθόδους επικοινωνίας μεταξύ αυτών. [12]

### 2.7.1 Επικοινωνία στο εσωτερικό του LoRa server

Στη συνέχεια, κρίνεται αναγκαία η περιγραφή των πρωτοκόλλων επικοινωνίας που χρησιμοποιούνται στο εσωτερικό του LoRa server. Όπως φαίνεται και στην παρακάτω εικόνα (Εικόνα 25), δεν χρησιμοποιείται σε όλες τις συνδέσεις κοινό πρωτόκολλο μεταφοράς δεδομένων, λόγω των διαφορετικών απαιτήσεων ανάμεσα στις περιπτώσεις.



Εικόνα 25- Chirpstack Server

### 2.7.2 User Datagram Protocol (UDP)

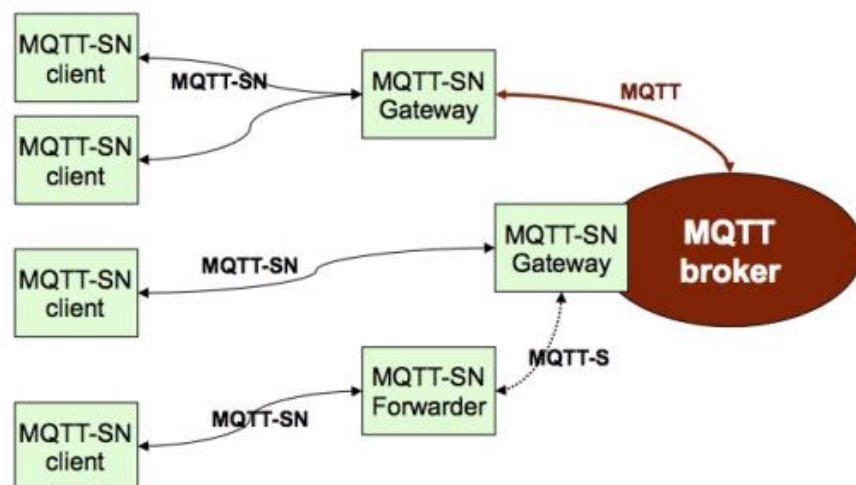
Είναι ένα πρωτόκολλο για μεταφορά δεδομένων ανάμεσα σε έναν server και έναν client σε εφαρμογές δικτύων. Λειτουργεί πάνω στο Internet Protocol (IP) και είναι μία εναλλακτική του TCP/IP [13]. Η επικοινωνία γίνεται μέσω μηνυμάτων που ονομάζονται Datagrams. Σε αυτά τα μηνύματα, στα πρώτα 8 bytes βρίσκεται μια επικεφαλίδα όπου περιέχει πληροφορίες σχετικές με την αφετηρία, τον προορισμό, το μέγεθος του μηνύματος, αλλά και μια ειδική μεταβλητή που χρησιμοποιείται για την προστασία από την αλλοίωση της πληροφορίας μέσω της διαδικασίας αποστολής της (checksum). Στα υπόλοιπα bytes των Datagrams βρίσκεται η επιθυμητή ανταλλάξιμη πληροφορία. Ο λόγος που επιλέγεται αυτός ο τρόπος επικοινωνίας είναι η γρήγορη απόκριση που έχει και ο χαμηλός χρόνος καθυστέρησης (latency) στην μετάδοση πληροφορίας. Αυτό επιτυγχάνεται λόγω της απουσίας αρκετών βαθμίδων ελέγχου που υπάρχουν για παράδειγμα στο TCP πρωτόκολλο. Η παραπάνω επιτάχυνση είναι

απαραίτητη για να προωθηθούν γρήγορα τα πακέτα από τα gateways κατευθείαν στον server. Σε αντίθετη περίπτωση, θα έπρεπε να γίνονται αναμεταδόσεις των χαμένων πακέτων, προκαλώντας μεγαλύτερη κίνηση στο κανάλι και αυξάνοντας τον συνολικό χρόνο μετάδοσης. Το tradeoff αυτής της επιλογής βρίσκεται στο γεγονός της αναξιόπιστης μερικές φορές επικοινωνίας, αφού όπως αναφέρθηκε δεν υπάρχουν πολλοί έλεγχοι κατά την διάρκεια της διαδικασίας. Βέβαια, λόγω της φύσης της πληροφορίας και της συχνότητας της, αποτελεί συμφέρουσα επιλογή, αφού και με τον ορισμό μικρού μέγιστου αριθμού πακέτων προς αποστολή κάθε φορά [14] περιορίζεται η αναξιοπιστία. Αρκετές διαδικτυακές εφαρμογές που επιθυμούν γρήγορη και μικρής πολυπλοκότητας επικοινωνία χρησιμοποιούν το UDP/IP, όπως η μετάδοση ήχου και εικόνας σε πραγματικό χρόνο (video and audio live streaming). Σε αυτήν την περίπτωση προτιμάται η απώλεια κάποιων μηνυμάτων προκαλώντας μια μικρή πτώση της ποιότητας επικοινωνίας χωρίς όμως να επηρεάζει σημαντικά την συνολική επίδοση του συστήματος, παρά η προσθήκη καθυστέρησης προς επίτευξη καλύτερης ζεύξης μέσω αναμεταδόσεων.

### 2.7.3 Message Queue Telemetry Transport (MQTT)

Το MQTT (Message Queuing Telemetry Transport) είναι ένα ελαφρύ και ελεύθερο πρωτόκολλο μεταφοράς μηνυμάτων βασισμένο σε αρχιτεκτονική έκδοσης/συνδρομής (publish/subscribe) μεταξύ πελάτη και εξυπηρετητή (client/server) [15]. Χρησιμοποιείται σε περιβάλλοντα περιορισμένων δυνατοτήτων με μικρό εύρος ζώνης δικτύου (network bandwidth), μεγάλες καθυστερήσεις, αναξιόπιστες συνδέσεις και συσκευές με μικρή υπολογιστική και αποθηκευτική δυνατότητα [15].

#### MQTT-SN Architecture



Εικόνα 26 – MQTT Αρχιτεκτονική

#### 2.7.4 Βασικές ιδέες

Με σκοπό την διατήρηση του μικρού υπολογιστικού και δικτυακού κόστους το MQTT υλοποιεί τις εξής βασικές ιδέες:

**Publish/subscribe:** Το μοντέλο αυτό επιτρέπει στους clients να λαμβάνουν τα μηνύματα που τους αφορούν και μόνο αυτά, ενώ παράλληλα στέλνουν μηνύματα σε θέματα (topics) που γίνονται διαθέσιμα μόνο σε όσους clients έχουν κάνει subscribe στα θέματα αυτά. Εδώ είναι σημαντικό να αναφέρουμε ότι το μοντέλο αυτό μας δίνει την δυνατότητα επικοινωνίας μεταξύ clients χωρίς να γνωρίζει το ένα την ύπαρξη του άλλου, που είναι βασική απαίτηση ενός επεκτάσιμου (scalable) περιβάλλοντος [1].

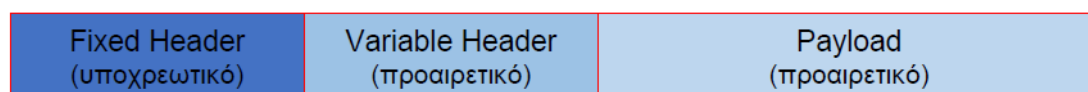
**Θέματα και συνδρομές (topics/subscriptions):** Τα θέματα είναι ουσιαστικά ο τρόπος διευθυνσιοδότησης των μηνυμάτων αγνοώντας την ύπαρξη των clients. Αποτελούν θεματικές ενότητες στις οποίες κάνουν subscribe οι ενδιαφερόμενοι clients και έτσι έχουν πρόσβαση στα μηνύματα που αφορούν αυτές τις θεματικές ενότητες [1].

**Επίπεδα ποιότητας υπηρεσιών (Quality of Service - QoS):** Το MQTT μας παρέχει 3 επίπεδα QoS για την παράδοση μηνυμάτων. Κάθε επίπεδο προσφέρει μεγαλύτερη αξιοπιστία παράδοσης με μεγαλύτερο κόστος αντιστοίχα [1].

**Διατήρηση μηνυμάτων και συνδέσεων (retained messages and connections):** Υπάρχει η δυνατότητα να ληφθούν μηνύματα σε ένα topic ακόμη και αν ο client έκανε subscribe μετά την αποστολή αυτών. Επίσης, μπορούμε να διατηρήσουμε τα subscriptions ενός client ακόμη και σε περίπτωση ξαφνικής αποσύνδεσης, ώστε να συνεχίσει απρόσκοπτα μετά την επανασύνδεση του. Σε περιπτώσεις ξαφνικών αποσυνδέσεων έχουμε και την δυνατότητα διαθήκης (will), η οποία εκτελεί κάποια συγκεκριμένη εργασία με την αποσύνδεση [1].

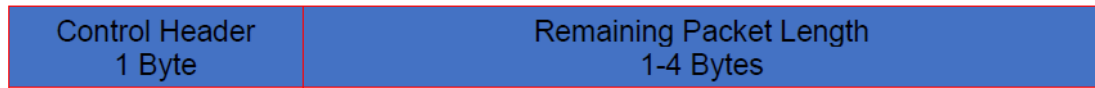
#### 2.7.5 Πακέτα MQTT

Ένα μήνυμα σε MQTT αποτελείται από ένα MQTT πακέτο (packet) το οποίο ενθυλακώνεται από ένα TCP/IP packet. Στην εικόνα (Εικόνα 27) διακρίνουμε σε τρία μέρη την δομή ενός MQTT packet [1]: Fixed Header, Variable Header και Payload, εκ των οποίων το πρώτο είναι υποχρεωτικό, ενώ τα δύο τελευταία προαιρετικά.



Εικόνα 27- MQTT packet.

Το MQTT packet ξεκινάει πάντα από μια σταθερή επικεφαλίδα (fixed header), η οποία αποτελείται από τα Control Header και Remaining Packet Length [1] όπως βλέπουμε στην Εικόνα 28- MQTT packet fixed header.:



Εικόνα 28- MQTT packet fixed header.

Το **control header**, μήκους 1 byte, ελέγχει το είδος του μηνύματος με τα 4 πιο σημαντικά bit, ενώ τα υπόλοιπα 4 bit περιέχουν σημαίες (flags) σχετικές με κάθε είδος μηνύματος [12].

| MsgType Field Value | MsgType      | MsgType Field Value | MsgType              |
|---------------------|--------------|---------------------|----------------------|
| 0x00                | ADVERTISE    | 0x01                | SEARCHGW             |
| 0x02                | GWINFO       | 0x03                | reserved             |
| 0x04                | CONNECT      | 0x05                | CONNACK              |
| 0x06                | WILLTOPICREQ | 0x07                | WILLTOPIC            |
| 0x08                | WILLMSGREQ   | 0x09                | WILLMSG              |
| 0x0A                | REGISTER     | 0x0B                | REGACK               |
| 0x0C                | PUBLISH      | 0x0D                | PUBACK               |
| 0x0E                | PUBCOMP      | 0x0F                | PUBREC               |
| 0x10                | PUBREL       | 0x11                | reserved             |
| 0x12                | SUBSCRIBE    | 0x13                | SUBACK               |
| 0x14                | UNSUBSCRIBE  | 0x15                | UNSUBACK             |
| 0x16                | PINGREQ      | 0x17                | PINGRESP             |
| 0x18                | DISCONNECT   | 0x19                | reserved             |
| 0x1A                | WILLTOPICUPD | 0x1B                | WILLTOPICRESP        |
| 0x1C                | WILLMSGUPD   | 0x1D                | WILLMSGRESP          |
| 0x1E-0xFD           | reserved     | 0xFE                | Encapsulated message |
| 0xFF                | reserved     |                     |                      |

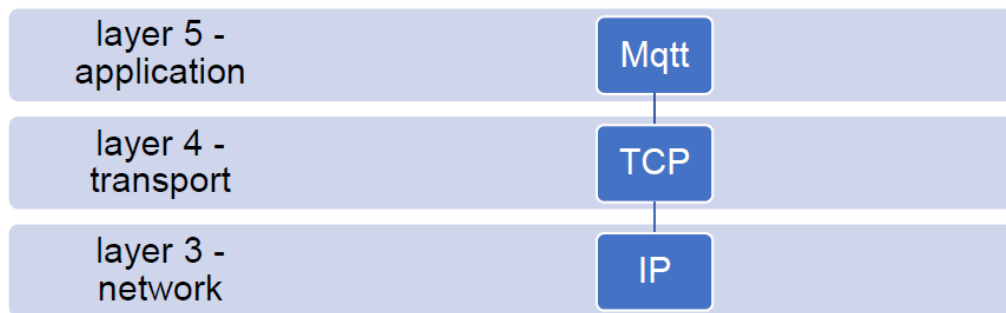
Εικόνα 29-Πίνακας μηνυμάτων ελέγχου.

Το **remaining packet length** header δηλώνει το υπολειπόμενο μέγεθος του πακέτου του μηνύματος συμπεριλαμβανομένου του variable header και payload. Αναλόγως του μήκους μπορεί να είναι από 1 έως 4 byte [1]. Κάποιοι τύποι πακέτων περιέχουν και variable header όταν αυτό έχει αυστηρό QoS (QoS 1 ή QoS 2). Τα byte του header χρησιμοποιούνται για την ταυτοποίηση του επιπέδου του πρωτοκόλλου και των πακέτων δίνοντάς τους ένα μοναδικό αναγνωριστή (identifier) [1]. Τέλος, στο payload βρίσκουμε το περιεχόμενο του μηνύματος που αποστέλλουμε. Όπως είναι προφανές, δεν έχουν όλα τα είδη πακέτων payload. Εκτός από το publish, payload έχουν τα connect, subscribe, suback και unsubscribe. Στην περίπτωση τους, το payload χρησιμοποιείται για να δώσει επιπλέον πληροφορία στην ενέργεια του μηνύματος [1].

### 2.7.6 Σύνδεση Client/Server

Πριν αναλύσουμε την pub/sub αρχιτεκτονική του MQTT αναλύουμε τον τρόπο σύνδεσης μεταξύ client και server. Στο μοντέλο αναφοράς πέντε επιπέδων TCP/IP [13], το MQTT

πρωτόκολλο βρίσκεται στο επίπεδο εφαρμογής (που περιλαμβάνει τα επίπεδα 5-7 του OSI μοντέλου), δηλαδή σε παραπάνω επίπεδο από τα πρωτοκόλλα TCP/IP, επομένως κάθε client και server απαιτούν TCP/IP stack για να μπορούν να συνδεθούν όπως παρατηρούμε στην εικόνα (Εικόνα 30-TCP/IP μοντέλο 5 επιπέδων για το MQTT.).

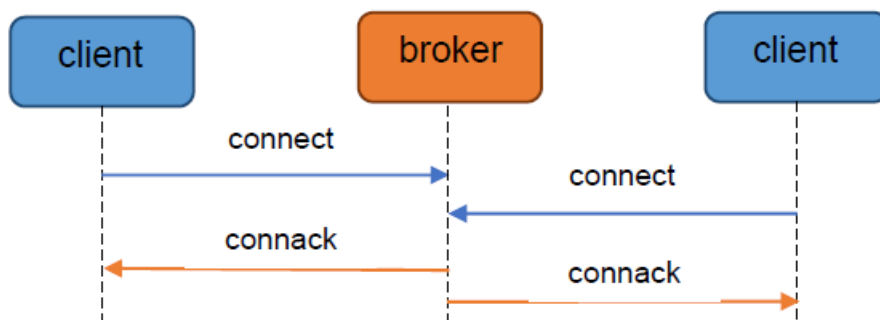


Εικόνα 30-TCP/IP μοντέλο 5 επιπέδων για το MQTT.

Ο client που χρησιμοποιεί το πρωτόκολλο MQTT για να επικοινωνήσει τον ονομάζουμε από εδώ και πέρα MQTT client. Αντίστοιχα τον server που μεσολαβεί μεταξύ των MQTT clients, λαμβάνει όλα τα μηνύματα και τα προωθεί κατάλληλα τον αναφέρουμε ως broker.

Ο MQTT client έχει τις εξής δυνατότητες [1]: Σύνδεση (connect), Δημοσίευση (publish), Συνδρομή (subscribe), Διαγραφή συνδρομής (unsubscribe) , Αποσύνδεση (disconnect)

Ο MQTT broker βρίσκεται στο κέντρο κάθε MQTT συστήματος. Η πρωταρχική του δουλειά είναι να λαμβάνει και να φιλτράρει όλα τα μηνύματα στέλνοντας τα στους κατάλληλους (subscribed) clients. Επίσης είναι υπεύθυνος για την εξασφάλιση της ορθής αποστολής και αποθήκευσης μηνυμάτων σε περίπτωση αποτυχίας, την αποθήκευση των persisted clients, και την εφαρμογή των απαιτούμενων security features. Καταλαβαίνουμε ότι σε ένα scalable δίκτυο έχει μεγάλη σημασία η ικανότητα του broker να διαχειρίζεται τον όγκο των δεδομένων με αξιοπιστία. Σημειώνουμε εδώ ότι όπως βλέπουμε στην εικόνα (Εικόνα 31) όλοι οι clients συνδέονται αποκλειστικά και μόνο με τον broker χωρίς την δυνατότητα να δουν ή να συνδεθούν με άλλους clients. Αυτό είναι πρωταρχικής σημασίας για την απλότητα των clients σε υπολογιστικό επίπεδο και για το scalability του συστήματος [15] [1].

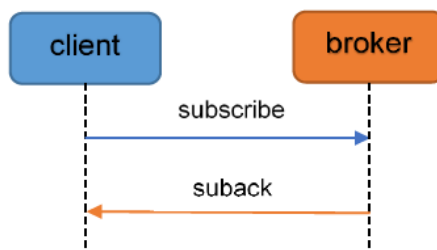


Εικόνα 31- MQTT connect.

### 2.7.7 Το Pub/Sub μοντέλο

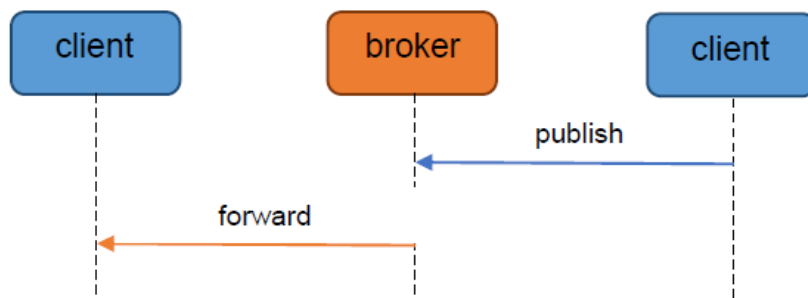
Το pub/sub είναι μοντέλο επικοινωνίας όπου οι αποστολείς των μηνυμάτων (εκδότες/publishers) δεν τα αποστέλλουν απευθείας στους παραλήπτες συνδρομητές (subscribers) αλλά τα μηνύματα διακρίνονται σε κατηγορίες, χωρίς να γνωρίζουν οι εκδότες ποιοι είναι οι παραλήπτες. Αντίστοιχα, οι παραλήπτες εκδηλώνουν το ενδιαφέρον τους σε μια κατηγορία μηνυμάτων, τα οποία και λαμβάνουν χωρίς να ξέρουν ποιοι είναι οι αποστολείς. Η παραπάνω κατηγοριοποίηση επιτυγχάνετε με την χρήση θεμάτων (topics) τα οποία αποτελούν χαρακτηριστικό κάθε μηνύματος, αντικαθιστώντας την διεύθυνση ενός παραδοσιακού client/server μοντέλου. Η χρήση του pub/sub μοντέλου ενδείκνυται για την επεκτασιμότητα που παρέχει στο σύστημά μας, όπως και για την ανοχή σε αλλαγές του δικτύου μας.

Με το subscribe ο client δηλώνει στον broker τα ονόματα των θεμάτων από τα οποία θέλει να λαμβάνει μηνύματα (Εικόνα 32Εικόνα 32- MQTT subscribe). Έτσι ο broker γνωρίζει ποιοι client έχουν κάνει subscribe σε κάθε topic, φιλτράρει κατάλληλα τα εισερχόμενα μηνύματα και τα προωθεί στους subscribed clients. Η αποστολή μηνύματος με topic που δεν έχει subscribers, δεν έχει και παραλήπτη για τον broker [1] [15].



Εικόνα 32- MQTT subscribe

Με το publish ένας client μπορεί να στείλει μήνυμα σε όλους τους subscribers ενός topic, χωρίς να γνωρίζει ποιοι είναι και που βρίσκονται (Εικόνα 33). Τα βασικά γνωρίσματα του publish είναι το περιεχόμενο του μηνύματος (payload), το όνομα του θέματος (topic name) και το απαιτούμενο QoS. Σημειώνουμε ότι ο client που κάνει publish δεν λαμβάνει καμία πληροφορία για τους παραλήπτες, μιας και αυτό είναι δουλειά του broker [1] [15].



Εικόνα 33-MQTT publish.

### 2.7.8 Topics

Το topic είναι ένα UTF-8 string που χρησιμοποιείται από τον broker στο φιλτράρισμα των μηνυμάτων για τους κατάλληλους clients. Μπορούμε να έχουμε πολλαπλά θεματικά επίπεδα σε ένα topic για τη διευκόλυνσή μας, με κάθε επίπεδο να ξεχωρίζει στο string από ένα διαχωριστικό «/» [1].

#### Parent\_topic / child\_topic

Γενικά είναι καλή πρακτική να ξεκινάμε με ένα super topic «home». Συνίσταται να μην ξεκινάμε ποτέ με διαχωριστική γραμμή γιατί αυτό δημιουργεί ένα super topic χωρίς όνομα. Παρατηρούμε ότι κάθε parent topic μπορεί να έχει πολλαπλά child topics, ανάλογα με τον σχεδιασμό του συστήματός μας. Έτσι δημιουργούμε ένα δέντρο θεμάτων (topic tree) το οποίο όμως βρίσκεται μόνο στον σχεδιασμό μας και δεν αποθηκεύεται στο σύστημα. Ένας broker βλέπει μόνο το παραπάνω format, δηλαδή ένα string διαχωρισμένο με slashes.

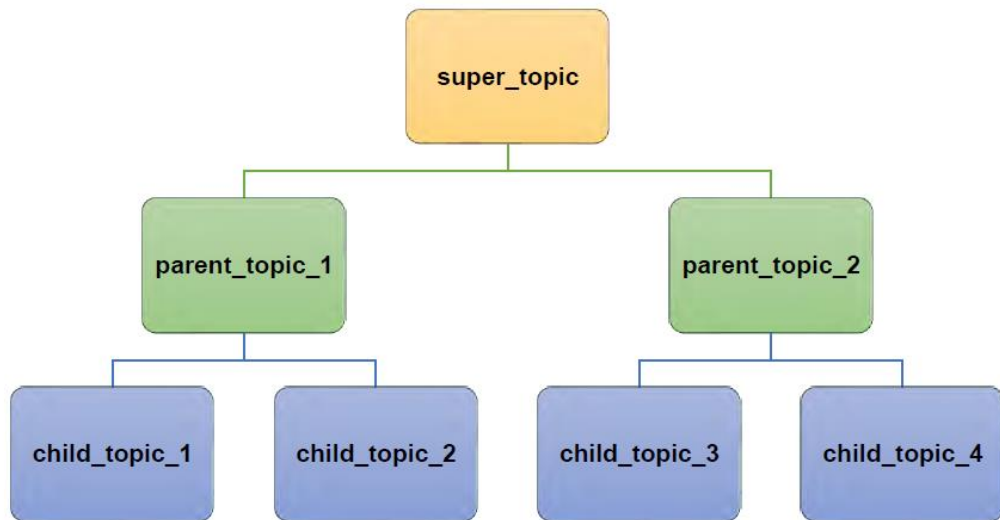
Στην εικόνα (Εικόνα 34) δείχνουμε την ιεραρχία των θεμάτων σε ένα δέντρο. Το super\_topic είναι ένα topic που χρησιμοποιούμε για την ολοκληρωμένη ιεράρχηση του δέντρου μας. Δεν το χρησιμοποιούμε αλλά η βιβλιογραφία προτείνει την χρήση του για το scalability του δέντρου μας [15]. Ονόματα που μπορούμε να χρησιμοποιήσουμε για αυτό το topic είναι home ή root ή και τον χώρο στον οποίο βρίσκεται η εφαρμογή μας όπως university, factory. Για να αναφερθούμε στα 4 child topics γράψουμε:

**Super\_topic/parent\_topic\_1/child\_topic\_1**

**Super\_topic/parent\_topic\_1/child\_topic\_2**

**Super\_topic/parent\_topic\_2/child\_topic\_3**

**Super\_topic/parent\_topic\_2/child\_topic\_4**



Εικόνα 34- Δομή MQTT topics

Πολλές φορές θέλουμε να αναφερθούμε σε πολλαπλά topics. Το MQTT μας δίνει την δυνατότητα να χρησιμοποιήσουμε τρία σύμβολα (+, #, \$), τα οποία η βιβλιογραφία ονομάζει wildcards [1] για να αναφερθούμε σε πολλά topic με μία έκφραση. Το σύμβολο της πρόσθεσης + λέγεται μονοεπίπεδο wildcard γιατί μας επιτρέπει να αναφερθούμε σε ένα ολόκληρο θεματικό επίπεδο, για παράδειγμα:

**Super\_topic/+/child\_topic\_1**

Στο παραπάνω παράδειγμα αναφερόμαστε σε όλα τα parent\_topics των θεμάτων με όνομα child\_topic\_1, δηλαδή στα:

**Super\_topic/parent\_topic\_1/child\_topic\_1**

**Super\_topic/parent\_topic\_2/child\_topic\_1**

**Super\_topic/parent\_topic\_3/child\_topic\_1**

Επίσης, το σύμβολο της δίεσης # λέγεται πολυεπίπεδο wildcard και μας δίνει την δυνατότητα να αναφερθούμε σε ένα όλα τα επίπεδα που βρίσκονται στο ίδιο επίπεδο και χαμηλότερα, για παράδειγμα:

**Super\_topic/#**

Στο παραπάνω παράδειγμα αναφερόμαστε στα εξής topics:

**Super\_topic/parent\_topic\_1/child\_topic\_1**

**Super\_topic/parent\_topic\_1/child\_topic\_2**

**Super\_topic/parent\_topic\_2/child\_topic\_3**

#### **Super\_topic/parent\_topic\_2/child\_topic\_4**

Παρατηρούμε ότι με την δέση αναφερόμαστε σε όλα τα topics που βρίσκονται στο επίπεδο 2 και χαμηλότερα [1].

Εδώ σημειώνουμε την ύπαρξη του συμβόλου \$ που χρησιμοποιείται μόνο για topics προς εσωτερική χρήση του broker και debugging και όχι για αποστολή μηνυμάτων .

Από τα παραπάνω καταλαβαίνουμε την τεράστια σημασία του σχεδιασμού του topic tree σε ένα MQTT project. Συνήθως χρησιμοποιούμε πρακτικές που διευκολύνουν το σωστό σχεδιασμό. Οι κυριότερες είναι:

- Δεν χρησιμοποιούμε κενά στα ονόματα των θεμάτων γιατί μεγαλώνει η πιθανότητα λάθους στην αναγραφή αυτών στον κώδικά μας.
- Χρησιμοποιούμε σύντομα και περιεκτικά ονόματα για να διευκολύνουμε την ανάγνωσή και κατανόηση τους.
- Δεν ξεχνάμε την πιθανότητα της ανάγκης πρόσθεσης νέων topic στο δέντρο. Άρα, σχεδιάζουμε πάντα με αυτό υπόψιν. Η χρήση super topic όπως πχ. home/ μας διευκολύνει σε αυτό και γ' αυτό την προτιμούμε.
- Συνδυάζουμε τα αντικείμενα του project με τις θεματικές ενότητες.

**Sup\_e\_topic/device\_1/sensor\_1**

**Sup\_e\_topic/device\_1/sensor\_2**

- Συνδυάζουμε λειτουργίες των αντικειμένων.

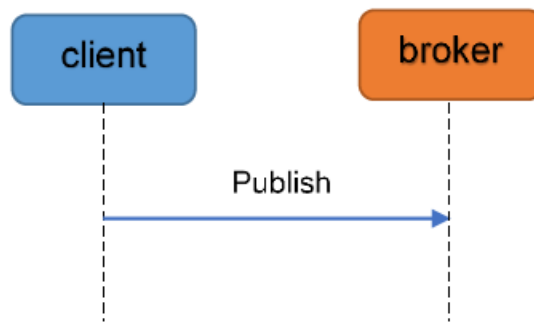
**Sup\_e\_topic/door\_1/open**

**Sup\_e\_topic/door\_1/close**

## **QoS**

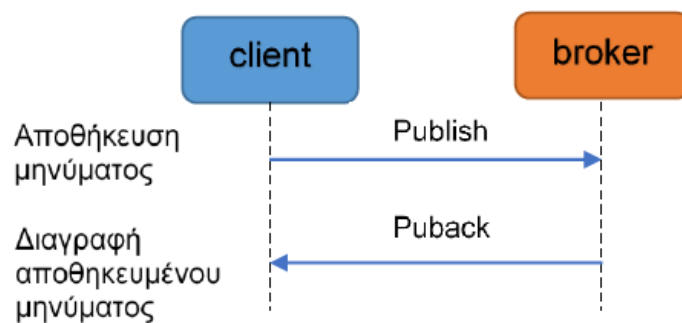
Τα επίπεδα ποιότητας υπηρεσιών (QoS levels) καθορίζουν την διαδικασία αποστολής ενός μηνύματος. Κάθε φορά που κάνουμε publish ορίζουμε το επίπεδο QoS που θέλουμε για το συγκεκριμένο μήνυμα. Το επίπεδο QoS το επιλέγουμε με βάση τις ανάγκες μας και την αξιοπιστία του δικτύου που χρησιμοποιούμε. Τα επίπεδα QoS που μας είναι διαθέσιμα είναι:

**QoS 0 (Αποστολή το πολύ μία φορά).** Στο επίπεδο αυτό δεν ελέγχεται η ορθή αποστολή του μηνύματος. Το μήνυμα αποστέλλεται χωρίς να αναμένεται απάντηση ή νέα αποστολή σε περίπτωση σφάλματος. Προφανώς σε αυτήν την περίπτωση έχουμε την ταχύτερη αποστολή. Το χρησιμοποιούμε όταν το δίκτυο μας είναι αξιόπιστο και οι client μας δεν αναμένουμε να έχουν απρόβλεπτες αποσυνδέσεις. Στην εικόνα (Εικόνα 35) βλέπουμε μια απλή αποστολή με QoS 0.



Εικόνα 35- MQTT publish με QoS 0.

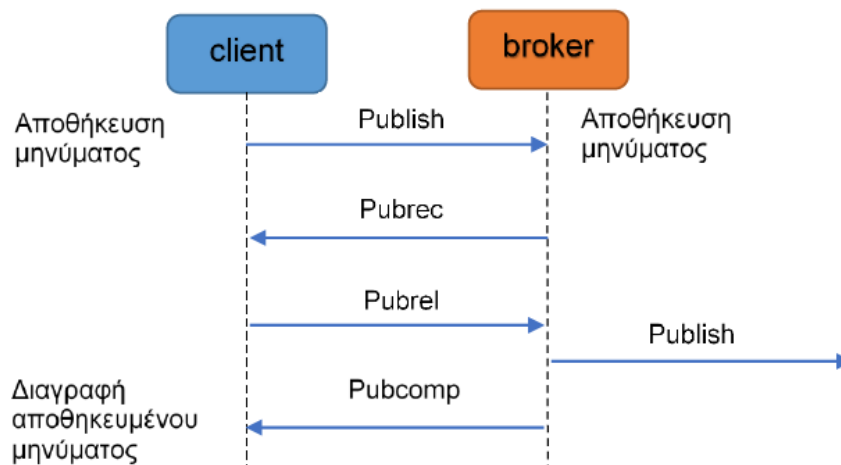
**QoS 1 (Αποστολή τουλάχιστον μια φορά).** Σε αυτό το επίπεδο ο αποστολέας αναμένει αναγνώριση της αποστολής (PUBACK) από τον broker (Εικόνα 36). Σε περίπτωση που δεν ληφθεί το PUBACK μέσα σε ένα προκαθορισμένο χρόνο το μήνυμα ξαναστέλνεται με το flag DUP που καταδεικνύει ότι είναι αντίγραφο. Ο broker όταν λάβει το αντίγραφο το επαναπροωθεί στους σωστούς αποδέκτες και στέλνει στον αποστολέα PUBACK. Όπως καταλαβαίνουμε όταν υπάρχει πρόβλημα στην αποστολή του PUBACK αλλά όχι του PUBLISH οι αποδέκτες λαμβάνουν πολλαπλά αντίγραφα του ίδιου μηνύματος. Τέλος, από την πλευρά των clients εφαρμόζουμε μεθόδους αποθήκευσης των μηνυμάτων (MQTT persistence) σε περίπτωση διακοπής της λειτουργίας τους, πριν ληφθεί η αναγνώριση της αποστολής, ώστε να είναι δυνατή η συνέχεια της διαδικασίας της αποστολής με την επαναλειτουργία τους.



Εικόνα 36. - MQTT publish με QoS 1.

**QoS 2 (Αποστολή μόνο μια φορά).** Το υψηλότερο επίπεδο του QoS, επιπλέον από τις λειτουργίες του QoS 1, εξασφαλίζει ότι οι αποδέκτες δεν λαμβάνουν αντίγραφα του ίδιου μηνύματος. Για το επιτύχει αυτό ο αποστολέας και ο broker αποθηκεύουν το μήνυμα μέχρι να ολοκληρωθεί η διαδικασία αποστολής (Εικόνα 37). Αφού ο client στείλει PUBLISH και αποθηκεύσει το μήνυμα, ο broker απαντάει στον αποστολέα με PUBREC και αποθηκεύει και αυτός το μήνυμα. Ο client επιβεβαιώνει την αποστολή

με την εντολή PUBREL, που μόλις ληφθεί από τον broker αρχίζει την διαδικασία αποστολής του μηνύματος στους αποδέκτες και απαντάει στον αποστολέα με PUBCOMP το οποίο ολοκληρώνει την διαδικασία και επιτρέπει στον αποστολέα να διαγράψει το αποθηκευμένο μήνυμα. Σε περίπτωση σφάλματος η παραπάνω διαδικασία επαναλαμβάνετε με το flag DUP. Όπως καταλαβαίνουμε ότι η παραπάνω διαδικασία είναι η πιο αργή από τις τρεις, και για αυτό την χρησιμοποιούμε μόνο όταν θέλουμε να σιγουρευτούμε ότι οι αποδέκτες δεν λαμβάνουν αντίγραφα του ίδιου μηνύματος.



Εικόνα 37. MQTT publish με QoS 2.

### 2.7.9 Διαθήκη client και επιμονή μηνυμάτων

Σε περίπτωση αποσύνδεσης ενός client μπορούμε να έχουμε ορίσει από πριν ένα μήνυμα που θα σταλεί σε αυτή την περίπτωση. Αυτό μας διευκολύνει στο να εντοπίσουμε πότε και πού έγινε η αποσύνδεση αλλά και να φροντίσουμε την συνέχεια της λειτουργίας του συστήματος μετά την αποσύνδεση. Επίσης μπορούμε να διευκολύνουμε την αυτόματη επανασύνδεση και συνέχιση του αποσυνδεδεμένου client.

Ο broker έχει την ευθύνη αυτομάτως να αποσυνδέει όλους τους clients μετά από μια περίοδο απραξίας. Ο client έχει την ευθύνη να καταγράφει τον χρόνο από την τελευταία φορά που αλληλεπίδρασε με τον broker. Όταν ξεπεραστεί ένα όριο χρόνου που ορίζουμε εμείς, τότε ο client στέλνει PINGREQ στον broker (ο οποίος απαντά με PINGRESP) για να αποφύγει την αυτόματη αποσύνδεσή του από τον broker. Στην περίπτωση αποσύνδεσης τότε ο broker στέλνει το προκαθορισμένο μήνυμα διαθήκης του client.

Παράλληλα, μπορούμε να σιγουρευτούμε ότι μηνύματα με αποδέκτη τον αποσυνδεδεμένο client θα γίνουν αποδεκτά ετεροχρονισμένα με την επανασύνδεση του. Με το flag RETAIN ένα μήνυμα χαρακτηρίζεται ως επίμονο και αποθηκεύεται από τον broker μέχρι να ληφθεί από όλους τους αποδέκτες ακόμη και αν αυτοί είναι αποσυνδεδεμένοι. Επομένως, με την

επανασύνδεση ενός client μπορεί να λάβει μηνύματα που είχαν αποσταλεί όσο ήταν εκτός λειτουργίας. Αυτή η λειτουργία μπορεί να χρησιμοποιηθεί και για την αρχικοποίηση ενός νέου client μέσω μηνυμάτων .

#### **2.7.10 Google Remote Procedure Call (gRPC)**

Το gRPC [16] είναι ένα open source framework για RPC που έχει αναπτύξει η Google. Αυτή η μέθοδος επικοινωνίας χρησιμοποιείται ανάμεσα στον Network Server, τον Mac Controller και τον Application Server. Πολλές λειτουργίες απαιτούν την συνεργασία αυτών των στοιχείων και για αυτόν τον λόγο επιλέγεται η απομακρυσμένη διαδικασία κλήσης. Στα διανεμημένα συστήματα, αυτή η μέθοδος επιτρέπει σε ένα πρόγραμμα να εκτελέσει μία υπορουτίνα του σε διαφορετική θέση διεύθυνσης και να αναφέρεται σε αυτή σαν να ήταν τοπική διαδικασία κλήσης. Η θέση διεύθυνσης μπορεί να ανήκει ακόμη και σε διαφορετικό υπολογιστή που βρίσκεται σε κοινό δίκτυο. Με αυτόν τον τρόπο δεν είναι αναγκαία η κωδικοποίηση των εντολών για απομακρυσμένη λειτουργία, μειώνοντας την πολυπλοκότητα και επιταχύνοντας την διαδικασία εκτέλεσής τους.

#### **2.7.11 Δομικά στοιχεία του LoRa Server**

Παρατηρώντας την αρχιτεκτονική του Server, γίνεται φανερό ότι αποτελείται από τέσσερις υπομονάδες, οι οποίες δρουν ξεχωριστά αλλά και σε συνεργασία ώστε να πραγματοποιήσουν εντολές και διαδικασίες απαραίτητες για την σωστή λειτουργία του συστήματος. Ο Gateway Bridge φέρεται ως γέφυρα ανάμεσα στα Gateways και τον Network Server. Λαμβάνει τα πακέτα πληροφορίας που στέλνει ο κάθε κεντρικός κόμβος μέσω UDP και τα δημοσιεύει σε μορφή Json μέσω MQTT στον Network Server. Ο Network Server είναι ο υπεύθυνος για την γενική λειτουργία του δικτύου. Αφού δέχεται τα δεδομένα των κόμβων από τον Gateway Bridge και τις απαραίτητες πληροφορίες για αυτά από τον Application Server, προχωράει σε μια σειρά από ενέργειες, όπως το να ελέγχει τα δεδομένα σε περίπτωση που έχουν προωθηθεί από παραπάνω από έναν κεντρικό κόμβο. Ακόμη, προωθεί τις πληροφορίες στον Application Server αλλά και τις αντίστοιχες από τον Server προς τους κόμβους (downlink data). Τέλος, όταν υπάρχει Mac εντολή θα ενεργοποιήσει τον Mac Controller για να την επεξεργαστεί. Ο Application Server είναι η μονάδα εκείνη που επιτρέπει την διαχείριση των κόμβων και την κατηγοριοποίηση τους ανά εφαρμογή και χρήστη. Είναι συμβατή με τον Network Server και είναι εκείνη από την οποία θα παραλάβει τελικά η εφαρμογή τις πληροφορίες που προορίζονται για τον χρήστη αλλά και εκείνη στην οποία θα δοθούν οι αντίστοιχες εντολές που πρέπει να μεταβούν σε οποιονδήποτε κόμβο. Ακόμη, αναφέρεται ότι μπορεί να λειτουργήσει παραπάνω από ένας Application Server για τον ίδιο Network Server. Έτσι, χρήστες θα μπορούν να

χρησιμοποιούν το ίδιο δίκτυο για διαφορετικό σκοπό με ξεχωριστά επίπεδα εφαρμογής. Τέλος, ο Mac Controller ασχολείται με τις Mac εντολές που ανταλλάσσονται στο δίκτυο.

### **2.7.12 Mac Controller**

Η επικοινωνία ανάμεσα στον Server και τις τερματικές συσκευές δεν περιορίζεται στην ανταλλαγή πληροφορίας αξιοποιήσιμης από την εφαρμογή αλλά και στην αποστολή MAC εντολών που σκοπό έχουν την αποτελεσματικότερη διαχείριση του δικτύου [17]. Ως παράδειγμα MAC εντολής, είναι η διαταγή του Server προς την τερματική συσκευή, να αλλάξει την ισχύ αποστολής που χρησιμοποιεί ή η ενημέρωση για κάποια τροποποίηση των καναλιών μετάδοσης. Η παραλαβή, σύνθεσή και επεξεργασία αυτών των εντολών γίνεται από την υποδομή του Server που ονομάζεται Mac Controller. Λόγο της φύσης των εντολών, πολλές φορές χρειάζεται η συνεργασία και με άλλες υποδομές όπως ο Network Server ή και ο Application Server και για αυτόν τον λόγο η εσωτερική επικοινωνία βασίζεται στο gRPC. [16]

### **2.7.13 Gateway Bridge**

Όπως αναφέρθηκε προηγουμένως, ο Gateway Bridge βρίσκεται ανάμεσα στους κεντρικούς κόμβους και στον Network Server. [18] Δέχεται τα πακέτα δεδομένων από τους πρώτους και τα μεταφορτώνει στον δεύτερο. Ο λόγος που χρησιμοποιείται και δεν στέλνονται απευθείας οι πληροφορίες στον Network Server [19], είναι επειδή προσφέρει πρόσθετες δυνατότητες που διευκολύνουν συνολικά το σύστημα. Δίνει άμεση πρόσβαση στα δεδομένα καθ' όλη την διαδρομή τους καθώς τα δημοσιεύει μέσω του MQTT Broker. [15] Με αυτόν τον τρόπο διευκολύνεται η παρακολούθηση των πακέτων που στέλνονται ή λαμβάνονται από κάθε κεντρικό κόμβο και απλοποιείται η αποσφαλμάτωση (debugging). Αυξάνεται λοιπόν η ορατότητα του διαχειριστή του δικτύου (visibility). Επιπλέον, αν ο Gateway Bridge εγκατασταθεί στους κεντρικούς κόμβους αντί στο σύστημα του LoRa Server, στην επικοινωνία μεταξύ αυτού και του Network Server μπορεί να χρησιμοποιηθεί επιπλέον επίπεδο ασφάλειας με χρήση Transport Layer Security (TLS) [20] στο MQTT, κρυπτογραφώντας έτσι τα πακέτα πληροφορίας. Ακόμη, κάνοντας εγγραφή στα ζητήματα που σχετίζονται με τον εκάστοτε κεντρικό κόμβο, ο MQTT Broker αναλαμβάνει την δρομολόγηση των μηνυμάτων προς τους κόμβους (downlink frames), στέλνοντας τα σωστά πακέτα στα διαφορετικά Gateways. Τέλος, το πιο σημαντικό ίσως πλεονέκτημα είναι ότι μέσω του Gateway Bridge, ανεξαρτητοποιείται ο LoRa Server από το πρωτόκολλο επικοινωνίας που χρησιμοποιεί ο packet forwarder του κεντρικού κόμβου. Διαφορετικές τεχνολογίες κόμβων δύναται να χρησιμοποιούν διαφορετικές μορφές στα δεδομένα που στέλνουν προς τον Server, μιας και αυτή δεν είναι μια αυστηρά τυποποιημένη διαδικασία. Έτσι, με την ανάπτυξη ενός νέου Gateway Bridge προσδίδεται η

ανεξαρτησία που χρειάζεται ώστε να μην επηρεαστεί ο Network Server και το υπόλοιπο σύστημα.

## **2.8 Network Server**

Μέχρι τώρα είδαμε ότι ο Network Server είναι υπεύθυνος για την γενική διαχείριση του δικτύου LoRaWAN. Στη συνέχεια θα αναλυθούν τα χαρακτηριστικά του, με σκοπό να γίνει σαφής ο τρόπος λειτουργίας του και η χρησιμότητά του. Το πακέτο πληροφορίας που στέλνεται από κάποιον κόμβο, λαμβάνεται από όλους τους κεντρικούς κόμβους που βρίσκονται εντός της εμβέλειας του (RF link). Στη συνέχεια προωθείται στον LoRa Server, μη γνωρίζοντας αν έχει ήδη λάβει από κάποιο άλλο αντίγραφο του, μειώνοντας έτσι ακόμα την ανάγκη για υπολογιστική ισχύ στους κεντρικούς κόμβους και συνεπώς το κόστος κατασκευής τους, αφού δεν χρειάζεται να ανταλλάσσουν μηνύματα μεταξύ τους. Έτσι ο LoRa Server θα πρέπει να εξασφαλίσει την μοναδικότητα της πληροφορίας που θα λάβει η εφαρμογή. Την δουλειά αυτή την αναλαμβάνει ο Network Server [19]. Φαίνεται ότι η κύρια λειτουργία του είναι η διαχείριση της ροής πακέτων αλλά και η επεξεργασία της πληροφορίας από τους κόμβους στον Application Server [21] και το αντίστροφο, αφού στην ροή ανόδου (uplink) όταν από-πολυπλέξει τα δεδομένα, θα τα προωθήσει στον Application Server όπου θα τα παρέχει με την σειρά του στην εφαρμογή. Αντίστοιχα στην ροή καθόδου (downlink), ο Network Server θα ζητήσει από τον Application Server τυχόν δεδομένα προς αποστολή αντί να του τα στείλει ο δεύτερος. Σε αυτήν την μονάδα γίνεται η επιλογή του ρυθμού μετάδοσης των δεδομένων και έτσι, ζητώντας τα δεδομένα (polling) από τον Application Server αντί να τα προωθεί ο δεύτερος (push), μπορεί να πετύχει το μέγιστο μέγεθος πληροφορίας (maximum payload size) με τον υπάρχον ρυθμό μετάδοσης σε συνδυασμό με τον κύκλο λειτουργίας που επιβάλλεται (duty cycle). Με αυτόν τον τρόπο οργανώνεται και αξιοποιείται καλύτερα το κανάλι μετάδοσης της πληροφορίας. Μετά την αποστολή τους, τα δεδομένα προς τους κόμβους θα διατηρηθούν σε αναμονή μέχρι να λάβει ο Server την αναγνώριση (acknowledge) από τον κόμβο ότι η πληροφορία έφτασε στον τελικό προορισμό της. Όπως αναφέρθηκε, τα μηνύματα που στέλνουν οι κόμβοι μπορούν να τα λάβουν όλοι οι κεντρικοί κόμβοι που βρίσκονται εντός εμβέλειας και το αντίστροφο, ακόμα και αν ανήκουν σε διαφορετικό δίκτυο. Για τον λόγο αυτό πρέπει να υπάρχει ένα σύστημα πιστοποίησης στον Server το οποίο θα επιτυγχάνεται με την καταχώρηση ειδικών κλειδιών τόσο στον ίδιο, όσο και στους κόμβους αντίστοιχα. Αυτή η διαδικασία αναλαμβάνεται από τον Network Server ο οποίος υποστηρίζει δύο διαφορετικές μεθόδους και έτσι, οι τερματικές συσκευές κατηγοριοποιούνται σε αυτές που χρησιμοποιούν μια στατική διαδικασία εκ των προτέρων πιστοποίησης Activation By Personalization(ABP) και σε αυτές που χρησιμοποιούν μια δυναμική διαδικασία πιστοποίησης κατά την διάρκεια αποστολής των μηνυμάτων [22]. Με χρήση του OTAA, [22] ο κόμβος πρέπει να περάσει από

μια διαδικασία αιτήματος-αποδοχής κάθε φορά που θέλει να ανταλλάξει πληροφορίες με τον Server. Δεν διαθέτει κλειδί για το δίκτυο όπως στην περίπτωση του ABP, αλλά κάποιο άλλο σχετικό με την εφαρμογή, το οποίο στέλνει σε ένα πρώτο μήνυμα αιτήματος στον Server (join request). Ο Network Server θα προωθήσει αυτήν την πληροφορία στον Application Server και αν δεχτεί θετική απάντηση, θα στείλει στον κόμβο ένα μήνυμα αποδοχής (join accept) επιτρέποντάς του με αυτόν τον τρόπο να ξεκινήσει την μετάδοση της πληροφορίας. Τα μηνύματα join request και join accept, στέλνονται με την μορφή MAC εντολών σε συνεργασία με τον Mac Controller. Σε αυτό το μήνυμα θα εμπεριέχεται ένα ειδικό κλειδί για την κρυπτογράφηση αλλά και την πιστοποίηση της επικοινωνίας. Η εκ των προτέρων καταχώρηση του κόμβου δηλαδή, γίνεται στην εφαρμογή και όχι στον Network Server, όπου σε αυτόν γίνεται δυναμικά κατά την διάρκεια της επικοινωνίας. Με αυτόν τον τρόπο διευκολύνεται η περιήγηση των κόμβων μεταξύ διαφορετικών δικτύων επιτρέποντας την δημιουργία εφαρμογών μεγαλύτερης κλίμακας. Αντίθετα, με χρήση ABP κάθε κόμβος (end device) έχει αποθηκευμένα κάποια κλειδιά τα οποία στέλνει στον Server στα πακέτα πληροφορίας και αν αυτά είναι όμοια με εκείνα που γνωρίζει ο Network Server, τότε συνδέεται αυτόματα. Κάθε τερματική συσκευή διαθέτει μοναδικά κλειδιά έτσι ώστε να μην θέτει σε κίνδυνο την ασφάλεια της επικοινωνίας με τις υπόλοιπες συσκευές τα οποία πρέπει να επιλέγονται με τέτοιον τρόπο ώστε να μην μπορούν να εξαχθούν από τρίτους μέσα από τις ήδη υπάρχουσες γνωστές πληροφορίες του node. Πρόκειται για μια στατική διαδικασία, η οποία δεσμεύει την τερματική συσκευή με ένα συγκεκριμένο δίκτυο παρακάμπτοντας τις MAC εντολές αίτηση-αποδοχή. Μια ακόμη λειτουργία που παρέχει και διαχειρίζεται ο Network Server είναι αυτή του προσαρμοσμένου ρυθμού μετάδοσης δεδομένων (adaptive data rate-ADR). Η λειτουργία αυτή εφαρμόζεται σε συσκευές που δεν μετατοπίζονται στον χώρο. Ο Network Server συλλέγει πληροφορίες από τα τελευταία μηνύματα που δέχτηκε από ένα συγκεκριμένο node, όπως την ισχύ λήψης του μηνύματος, τον αριθμό των gateways που το έλαβαν και το υπάρχων data rate. Έτσι ο Network Server υπολογίζει και επαναπροσδιορίζει τις συνθήκες αποστολής των μηνυμάτων από την μεριά του node, για παράδειγμα μειώνοντας την ισχύ αποστολής ή αυξάνοντας το data rate, πετυχαίνοντας έτσι μεγαλύτερη αποδοτικότητα μειώνοντας την ενεργειακή κατανάλωση ή αξιοποιώντας καλύτερα τον διαθέσιμο χρόνο αποστολής (airtime) [23]. Για την πραγματοποίηση αυτής της λειτουργίας, στο MAC Layer της LoRaWAN έχουν φτιαχτεί τέσσερις εντολές εκ των οποίων οι δύο είναι εντολές MAC. Αυτές είναι:

**ADR:** Ενεργοποιείται στην μεριά της τερματικής συσκευής και είναι μεγέθους 1 bit. Όταν τεθεί  $ADR = 1$ , τότε ενημερώνει τον Server ότι επιθυμεί να ενεργοποιήσει το ADR και έτσι να ελέγχει τις ιδιότητες των λαμβανόμενων μηνυμάτων. Σε αντίθετη περίπτωση, δηλώνει ότι επιθυμεί απενεργοποιημένη την λειτουργία.

**ADRACKReq:** Μεγέθους 1 bit. Όταν είναι ίσο με μονάδα, ο κόμβος ζητάει από τον server να επιβεβαιώσει σε ADR-ACK-DELAY αριθμό frames ότι λαμβάνει τα μηνύματα που του στέλνει.

**LinkADRReq:** Είναι MAC εντολή μεγέθους 4 bytes. Το στέλνει ο Server στον κόμβο για να τον ενημερώσει για τις αλλαγές που πρέπει να κάνει στις παραμέτρους αποστολής. Σε αυτήν την εντολή ρυθμίζεται ο ρυθμός μετάδοσης δεδομένων, η ισχύς αποστολής αλλά και το κανάλι που θα πρέπει να χρησιμοποιεί ο κόμβος στη συνέχεια.

**LinkADRAns:** Είναι MAC εντολή μεγέθους 1 byte και το στέλνει ο κόμβος στον server, προς απάντηση της προηγούμενης εντολής όπου κάθε παράμετρος αποστολής αντιστοιχίζεται σε μια συγκεκριμένη θέση μνήμης της εντολής. Αν σε κάποια θέση καταχωρηθεί η τιμή μηδέν, τότε σημαίνει ότι για κάποιον λόγο δεν έγινε η αλλαγή των παραμέτρων. Αλλιώς όλα άλλαξαν κανονικά.

## 2.9 Application Server

Ως υποδομή του LoRa Server, είναι υπεύθυνο για τα μηνύματα των κόμβων. Διαχειρίζεται τα λαμβανόμενα πακέτα πληροφορίας από της τερματικές συσκευές που θα χρησιμοποιήσει η εφαρμογή (application payloads), καθώς επίσης και αυτά που βρίσκονται σε αναμονή προς αποστολή από την εφαρμογή προς τους κόμβους (application downlink queue). Ένα μεγάλο πλεονέκτημα του Application Server, είναι ότι διαθέτει διεπαφή διαδικτυακής χρήσης (web interface). Μέσω αυτής, γίνεται ευκολότερη η διαχείριση κάποιων τμημάτων του δικτύου όπως των τερματικών συσκευών και των κεντρικών κόμβων. Έτσι πλέον, σε γραφικό περιβάλλον μπορεί να γίνει η προσθήκη, η διαγραφή αλλά και η επεξεργασία αυτών των στοιχείων. Επιπλέον, δημιουργούνται ενότητες για τις διαφορετικές εφαρμογές που χρησιμοποιούν το δίκτυο, αλλά υπάρχει και η δυνατότητα δημιουργίας διαφορετικών χρηστών. Σύμφωνα με τα παραπάνω, οι πληροφορίες από τους κόμβους οδηγούνται σε συγκεκριμένες εφαρμογές, στις οποίες έχουν πρόσβαση κάποιοι χρήστες. Επιτυγχάνεται μια κατηγοριοποίηση της πληροφορίας δηλαδή, ανά κόμβο, εφαρμογή και χρήστη. Αυτό διευκολύνει την δρομολόγηση των μηνυμάτων με σκοπό την τελική τους χρησιμοποίηση από τον χρήστη μέσω μιας εφαρμογής. Τα πακέτα πληροφορίας που μεταδίδονται σε ένα δίκτυο LoRaWAN, χρησιμοποιούν κρυπτογράφηση με χρήση AES 128 και σε αυτήν την μονάδα γίνεται η αποκρυπτογράφηση τους. Το τελικό μήνυμα θα εξακολουθεί να είναι κωδικοποιημένο με χρήση base64. Το κλειδί για την αποκρυπτογράφηση (application key) καταχωρείται κοινά και στον Application Server αλλά και σε κάθε κόμβο. Αφού αποκρυπτογραφηθούν τα μηνύματα, δημοσιεύονται (publish) σε ειδικά ζητήματα (topics) μέσω MQTT στα οποία έχει κάνει εγγραφή (subscribe) η εφαρμογή για να τα λάβει. Για τα δεδομένα προς τους κόμβους, η

διαδικασία είναι αντίστροφη, με την εφαρμογή να δημοσιεύει το μήνυμα στο ζήτημα μέσω MQTT από όπου λαμβάνεται από τον Application Server, κρυπτογραφείται και στη συνέχεια όταν ζητηθεί από τον Network Server, προωθείται για να σταλεί τελικά στους κεντρικούς κόμβους μέσω του Gateway Bridge και από εκεί στις τερματικές συσκευές. [21]

## 2.10 Σχετικές εργασίες

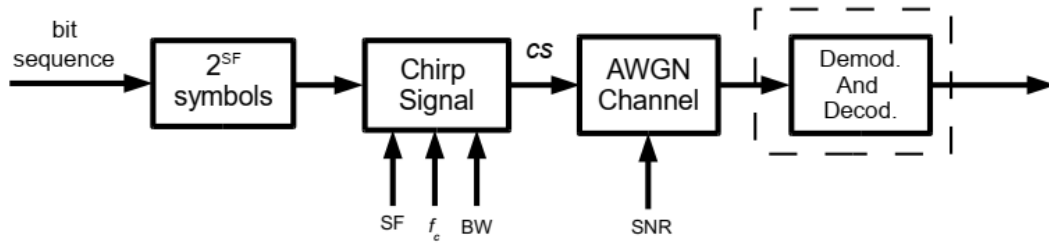
Στην εργασία «Investigating the practical performance of the LoRaWAN» [24] γίνεται μια μελέτη της απόδοσης που σχετίζεται με παρεμβολές μεταξύ συσκευών LoRaWAN που χρησιμοποιούν την ίδια συχνότητα.

Η απόδοση έχει αποδειχθεί ότι ποικίλλει ανάλογα με την ποσότητα των δεδομένων που αποστέλλονται και ο μέγιστος ρυθμός μετάδοσης είναι κοντά σ' αυτόν που καθορίζεται στο πρότυπο LoRaWAN. Έχει επίσης αποδειχθεί ότι όταν η χρήση καναλιού είναι πάνω από 54% για τον υψηλότερο ρυθμό μετάδοσης, μπορεί κανείς να περιμένει πολύ υψηλό ποσοστό απώλειας πακέτων. Στα πειράματα που πραγματοποιήθηκαν διαπιστώθηκε ότι κατά την αποστολή 51 byte δεδομένων εφαρμογής σε ένα τέτοιο κανάλι, μπορεί κανείς να λάβει απώλεια πακέτου ως έως και 92%. Αυτό που είναι ενδιαφέρον είναι ότι οι εκπομπές στο ίδιο κανάλι αλλά με διαφορετικούς ρυθμούς δεδομένων θα μπορούσαν να συνυπάρχουν χωρίς σημαντική απώλεια πακέτου.

*Κάνουμε χρήση των συμπερασμάτων της παραπάνω εργασίας και για την αποφυγή συγκρούσεων χρησιμοποιούμε κυλιόμενο ρυθμό μετάδοσης. Το αποτέλεσμα όπως θα δούμε και στα πειραματικά δεδομένα ήταν να έχουμε πολύ μικρές απώλειες.*

Μια μελέτη των σημάτων chirp για ασύρματες επικοινωνίες μεταξύ του Διαδικτύου των συσκευών που χρησιμοποιούνται σε δίκτυα χαμηλής ισχύος παρουσιάζεται στην εργασία «Wireless Communication Based on Chirp Signals for LoRa IoT Devices» [25]. Σ' αυτό το άρθρο παρουσιάζεται η μελέτη των σημάτων chirp για ασύρματες επικοινωνίες μεταξύ του Διαδικτύου των συσκευών που χρησιμοποιούνται σε δίκτυα χαμηλής ισχύος. Παρουσιάζεται η έννοια up/down chirp, καθώς και η έννοια φάσματος chirp spread. Μια υπολογιστικά αποδοτική μέθοδος αποκωδικοποίησης συμβόλων παρουσιάζεται με βάση διακριτό μετασχηματισμό Fourier ως εναλλακτική λύση στην τυπική συνεκτική ανίχνευση.

*Δανειστήκαμε το προτεινόμενο μοντέλο προσομοίωσης LoRa που υλοποιείται στο MATLAB επιτρέποντας την αξιολόγηση του συστήματος επικοινωνίας με βάση το ρυθμό σφάλματος bit και το ποσοστό σφάλματος πακέτου.*



Εικόνα 38- LoRa® CSS MATLAB simulation model.

**Τα πειράματά μας βασίστηκαν σε αντίστοιχο πείραμα** που παρουσιάζεται στην εργασία «Measuring a LoRa Network Performance, Possibilities and Limitations των Robin Franksson Alexander Liljegren» [26] . Στο παραπάνω πείραμα γίνεται μια αξιολόγηση στο κατά πόσο εμπόδια που αφορούν το περιβάλλον και τη θέση εφαρμογής επηρεάζουν την απόδοση ενός δικτύου LoRa.



Εικόνα 39- Measurement locations for dense forest experiment



Εικόνα 40 - Measurement locations for city experiment

Η εξέλιξη της τεχνολογίας LoRaWAN τα τελευταία χρόνια είναι ραγδαία. Κάτι τέτοιο φαίνεται στην εργασία « Frame Loss Characterisation and Data Erasure Coding for LoRaWAN» του Paul Marcelis, ο οποίος υποστηρίζει ότι οι μετρήσεις για την απόδοση του δικτύου βελτιώθηκαν αισθητά κατά την διάρκεια της συγκεκριμένης μελέτης. Παρόμοια παρατήρηση έχουμε κάνει και

εμείς, ως όσο αναφορά το υλικό και λογισμικό που έχει αναπτυχθεί και είναι διαθέσιμο στην αγορά κατά τη διάρκεια της δικής μας μελέτης. Στην εκκίνηση της μελέτης της εργασίας μας, οι πηγές και οι διαθέσιμες υλοποιήσεις ήταν περιορισμένες. Σήμερα υπάρχει πληθώρα λύσεων και σε επίπεδο λογισμικού και σε επίπεδο υλικού.

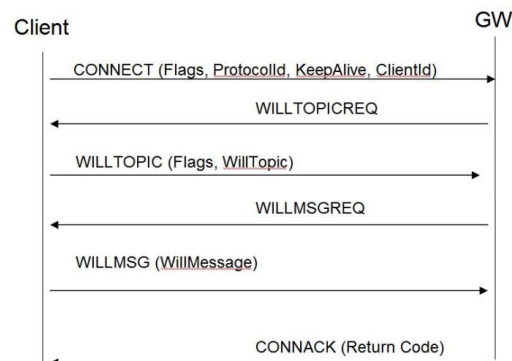


Εικόνα 41 – Smart City with LoRa

Τεχνικές λεπτομέρειες για τη τεχνολογία Spread Spectrum Communications αντλήθηκαν από το δικτυακό τόπο της Semtech Corporation [27] , « LoRa™ Modulation Basics» [11]

Για το πρωτόκολλο MQTT χρησιμοποιήθηκαν οι προδιαγραφές έτσι όπως περιγράφονται στο «MQTT For Sensor Networks (MQTT-SN) Protocol Specification Version 1»

και MQTT Version 5 από την OASIS. [1]



Εικόνα 42 - Connect procedure

Για την εγκατάσταση , και προγραμματισμό των συσκευών που χρησιμοποιήθηκαν στην μελέτη μας αντλήθηκε υλικό από τους αντίστοιχους δικτυακούς τόπους. Έτσι αντλήθηκαν πληροφορίες που αφορούν τεχνικές λεπτομέρειες της λειτουργίας του LoRaWAN από τον αντίστοιχο δικτυακό τόπο και ενδεικτικά αναφέρουμε τεκμήρια που χρησιμοποιήθηκαν από τον οργανισμό LoRa Alliance

- LoRaWAN® Regional Parameters
- LoRaWAN® Protocol with LoRa Radio
- LoRaWAN Specification 1.0.4

Από τον οργανισμό ChirpStack, open-source LoRaWAN® Network Server stack αντλήθηκαν πληροφορίες για την εγκατάσταση μιας ιδιωτικής στοίβας πρωτοκόλλων εξυπηρέτησης gateways και end devices.

Χρήσιμη ήταν και η τεκμηρίωση στις διάφορες φάσεις της εργασίας μας που αντλήθηκε από το δικτυακό τόπο του The Things Networks(TTN).

Για την παραμετροποίηση της gateway αντλήθηκαν πληροφορίες από την περιοχή υποστήριξης της Dragino

- LG01-N Single Channel LoRa IoT (Internet of Things) Gateway ,
- «LG308 LoRaWAN Gateway User Manual,» Dragino, 2020 [28]
- LPS8 LoRaWAN Gateway User Manual, Dragino, 2021. [29]

Για τα end devices αντλήθηκαν πληροφορίες από την υποστήριξη των:

- pycom (<https://pycom.io/wp-content/uploads/2018/08/lopy-specsheet.pdf>) [30]
- dragino shield ([https://wiki.dragino.com/index.php?title=Lora\\_Shield](https://wiki.dragino.com/index.php?title=Lora_Shield)).

# 3

## *Περιπτώσεις χρήσης – παραμετροποίηση ενός δικτύου LoRa*

Ο σκοπός της παρούσης διπλωματικής εργασίας είναι να προσδιορίσει πως η απόδοση ενός δικτύου LoRa επηρεάζεται από την θέση στην οποία υλοποιείται με παράλληλη αξιολόγηση της πολυπλοκότητας και του κόστους εγκατάστασης. Ένας άλλος στόχος είναι να ερευνήσουμε τις αντισταθμίσεις της χρήσης ενός τέτοιου δικτύου σε διαφορετικά περιβάλλοντα.

Με ανάλυση της απόδοσης των τελικών συσκευών (end-devices) και της gateway μετά από διαφορετικές παραμετροποιήσεις σε διαφορετικά περιβάλλοντα ελπίζουμε ότι θα μπορέσουμε να προτείνουμε κατάλληλους συνδυασμούς παραμετροποίησης ανάλογα με την περίπτωση χρήσης. Τέλος να κάνουμε προτάσεις χρήσης και εφαρμογή ενός τέτοιου συνδυασμού εγκατάστασης και παραμετροποίησης .

### **3.1 Συνθήκες πραγματοποίησης της έρευνας**

Η απόδοση θα μελετηθεί ως προς την περιοχή που εφαρμόζεται, αστικό ή μη αστικό περιβάλλον, ως προς το αν υπάρχει κίνηση ή όχι και τέλος ως προς την απόδοση του εξοπλισμού, κυρίως της gateway. Δεν θα λάβουμε υπόψη παραμέτρους καιρικών συνθηκών υποθέτοντας ότι οι απώλειες είναι σχετικά μικρές για τις περιπτώσεις χρήσης που θα εξετάσουμε.

### **3.2 Στόχος της έρευνας**

Το τελικό αποτέλεσμα της έρευνας θα είναι η καλύτερη κατανόηση για το κατά πόσο περιβαλλοντικοί παράγοντες επηρεάζουν την απόδοση ενός LoRa δικτύου. Επιπρόσθετα θα προκύψει μια καλά τεκμηριωμένη πρόταση για την εγκατάσταση ενός δικτύου IoT ανάλογη με την περίπτωση χρήσης που θα προταθεί.

### 3.2.1 Ερευνητικά ερωτήματα

**RQ1:** Ποιοι είναι οι περιορισμοί χρήσης ενός τέτοιου δικτύου σε ένα πυκνό αστικό περιβάλλον πόλης ή σε μια αγροτική περιοχή που μπορεί διάφορα εμπόδια να προκαλέσουν απώλειες. Καθώς το LoRa είναι σχετικά νέα τεχνολογία και δεν έχουν μελετηθεί διεξοδικά τρέχοντες περιορισμοί που επηρεάζουν την απόδοση σε διαφορετικές εφαρμογές και περιπτώσεις χρήσης, θα πρέπει να ερευνηθούν διάφοροι παράγοντες που σχετίζονται με το περιβάλλον εγκατάστασης και επηρεάζουν την απόδοση ενός δικτύου LoRa.

**RQ2:** Πως επηρεάζει η κίνηση των end-nodes την απόδοση του δικτύου. Δεδομένου ότι τα ραδιοκύματα επηρεάζονται σε μεγάλο βαθμό από το μέσο διέλευσης, μια έρευνα για το πως το μέσο μεταξύ του hardware μπορεί να επηρεάσει ένα δίκτυο LoRa μπορεί να μας οδηγήσει σε προτάσεις κατάλληλης τοποθέτησης ώστε να παρακαμφθούν τέτοιου είδους εμπόδια.

**RQ3:** Πόσο επηρεάζει η ποιότητα του Hardware κυρίως της gateway την απόδοση του συστήματος. Καθώς η πληροφορία που μεταδίδεται μέσω ραδιοκυμάτων μπορεί να είναι πολύ κυμαινόμενη στην απόδοσή της και επηρεάζεται από πλήθος περιβαλλοντικών μεταβλητών. Κάθε απώλεια πακέτου που προκύπτει πρέπει να ληφθεί υπόψη και να αναλυθεί. Η σημασία κάθε πακέτου που φτάνει στον προορισμό του αυξάνεται για τα δίκτυα LoRa καθώς οι κανονισμοί κύκλου λειτουργίας περιορίζουν σε μεγάλο βαθμό την ποσότητα των πακέτων που μπορούν να σταλούν.



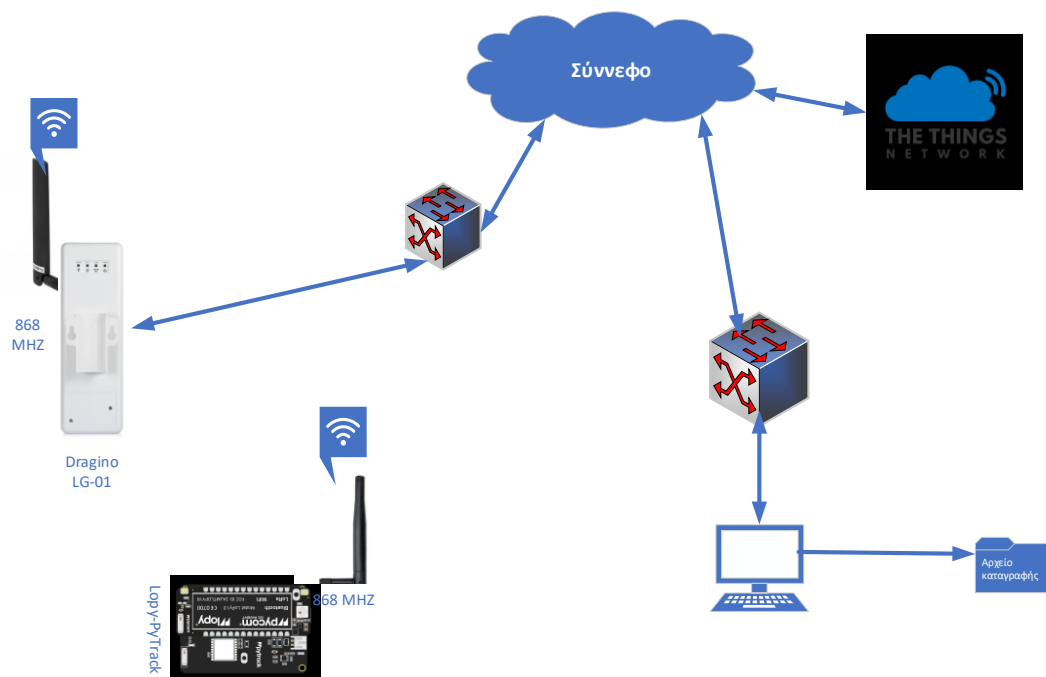
Εικόνα 43- Πηγή - <https://www.semtech.com> [27]

### 3.3 *Μελέτη παραμέτρων που επηρεάζουν την απόδοση ενός δικτύου LoRa.*

Η έρευνά μας επικεντρώνεται κυρίως στην απόδοση και τους περιορισμούς ενός δικτύου LoRa. Ο πρωταρχικός μας στόχος κατά την αναζήτηση βιβλιογραφικών αναφορών ήταν να βρούμε ερευνητικά έγγραφα και άρθρα, τα οποία περιλαμβάνουν εφαρμογή προσομοιώσεων ενός δικτύου LoRa. Επικεντρωθήκαμε στο πόσο ακριβείς μπορεί να είναι οι προσομοιώσεις σε σχέση με την εφαρμογή σε πραγματικές συνθήκες και στον αν μπορεί να υπάρξει ένα μοντέλο το οποίο θα προσομοιώνει αποτελεσματικά πραγματικά πειράματα. Σκοπός είναι να προταθούν μετά από πολύπλευρη μελέτη περιπτώσεις χρήσεις εφαρμογής ενός δικτύου LoRa.

### 3.4 *Τεχνικές προδιαγραφές του δικτύου LoRa για τα πειράματα*

Το πρώτο στάδιο της δημιουργίας ενός δικτύου είναι να διασφαλιστεί ότι τα απαραίτητα στοιχεία έχουν διαμορφωθεί σωστά. Αυτό με τη σειρά του θα εξασφαλίσει την ακρίβεια των δεδομένων που συλλέγονται κατά τη διάρκεια της έρευνας και ανάλυσης του δικτύου. Στην περίπτωση μας και με δεδομένους οικονομικούς πόρους η τοπολογία του δικτύου που αποφασίστηκε για τις μετρήσεις μας ήταν απλή.



Εικόνα 44- Τοπολογία δικτύου LoRa για τις μετρήσεις μας.

### 3.4.1 Εφαρμογή 1<sup>η</sup>

Στο σύστημα δοκιμών συμμετέχουν δύο Gateway

**Gateway τύπου Dragino LG01 outdoor** η οποία τοποθετήθηκε σε αστικό περιβάλλον στην περιοχή της Κοζάνης σε υψόμετρο 758 μέτρα και σε απόσταση 5 μέτρα πάνω από το έδαφος. Διαθέτει ένα κανάλι μετάδοσης LoRa με 49 demodulators . Η βασική συχνότητα μετάδοσης ρυθμίστηκε στα 868.3 MHz.

**Gateway τύπου Kerlink Wirnet Station** η οποία είναι τοποθετημένη στο Πανεπιστήμιο Δυτικής Μακεδονίας για ερευνητικούς σκοπούς σε κτίριο εντός πόλης. Η gateway αυτή μπορεί να υποστηρίξει 9 κανάλια μετάδοσης με 49 demodulators το κάθε κανάλι. Συχνότητα 863 – 873 Mhz.

**Σαν End – Device συμμετέχει ένας συνδυασμός Pytrack – Lopy.** Η συσκευή αυτή αναλαμβάνει την αποστολή πακέτων που γίνονται στα διάφορα πειράματά μας καταγράφοντας ταυτόχρονα και πληροφορίες θέσης.

#### 3.4.1.1 End - Device

Σαν συσκευή μετάδοσης χρησιμοποιήθηκε ένας συνδυασμός Pytrack – Lopy. Το Pytrack διαθέτει ενσωματωμένο GPS και Accelerometer, ενώ το Lopy διαθέτει ένα module LoRa στα 868 MHz . Η συσκευή προγραμματίστηκε να στέλνει πακέτα με σταθερό μέγεθος που περιείχαν τη θέση αλλά και επιπλέον πληροφορίες από το επιταχυνσιόμετρο. Έγινε χρήση External LoRa and Sigfox antenna 2.5 dBi.

Ο προγραμματισμός έγινε σε περιβάλλον Python ενώ όλες οι μετρήσεις έγιναν με βάση τις προδιαγραφές LoRaWAN που υποστηρίζεται σε επίπεδο εφαρμογών από το Lopy.

**Bandwidth: 868.1-868.5Mhz**

**Channel size: 125kHz**

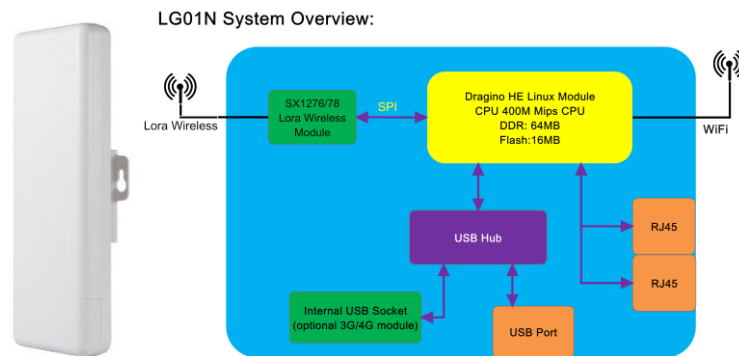
**Spreading Factor (sending and receiving), variable = 7 to 12:**

**Coding Rate (sending and receiving), fixed = 4/5**

**Transmitting Power: 14dBm**

### 3.4.2 Περιγραφή τεχνικών χαρακτηριστικών των gateways που συμμετέχουν.

#### 3.4.2.1 1<sup>η</sup> GateWay – LG01-N



Εικόνα 45 – LG01 Gateway

Η LG01-N Gateway είναι μια open source single channel Gateway. Μας επιτρέπει την σύνδεση ενός LoRa wireless network με ένα IP network με την βοήθεια σύνδεσης στο δίκτυο είτε με WiFi, είτε με Ethernet. Η LG01-N σχεδιάστηκε για να προσφέρει χαμηλού κόστους IoT wireless λύσεις υποστηρίζοντας 50~300 sensor nodes.

Όλες οι μετρήσεις έγιναν με τις παρακάτω παραμέτρους στη gateway:

**Bandwidth: 868.3Mhz**

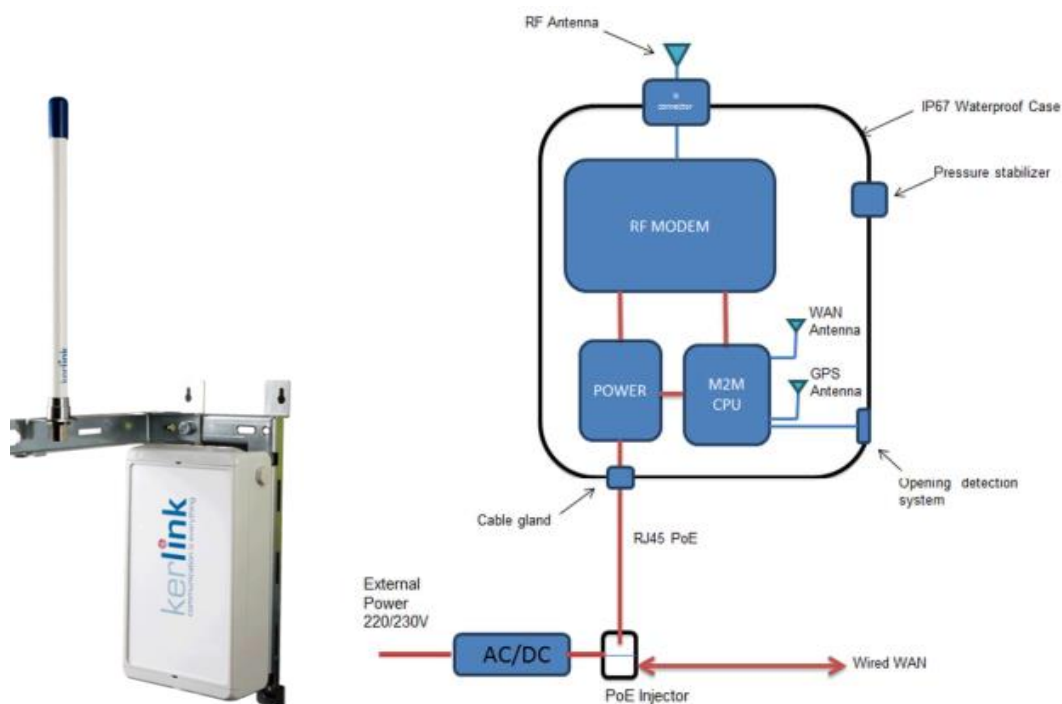
**Channel size: 125kHz**

**Spreading Factor (sending and receiving), variable = 7 to 12:**

**Coding Rate (sending and receiving), fixed = 4/5:**

**Transmitting Power: 14dBm**

### 3.4.2.2 Kerlink Wirnet Station



Εικόνα 46 – Kerlink Gateway

Η Kerlink Wirnet Station είναι μια multichannel Gateway. Μας επιτρέπει την σύνδεση ενός LoRa wireless network με ένα IP network με την βοήθεια σύνδεσης στο δίκτυο είτε με WiFi, είτε με Ethernet. Σχεδιάστηκε για να προσφέρει IoT wireless λύσεις υποστηρίζοντας 9 channels με 49 LoRa demodulators sensor nodes.

Όλες οι μετρήσεις έγιναν με τις παρακάτω παραμέτρους στη gateway:

- Bandwidth: 868.1-868.5Mhz
- Channel size: 125kHz
- Spreading Factor (sending and receiving), variable = 7 to 12:
- Coding Rate (sending and receiving), fixed = 4/5:
- Transmitting Power: 14dBm

### 3.4.2.3 Διαφοροποιήσεις μεταξύ των gateways

- Η μία είναι single channel η άλλη είναι multichannel.
- Η εξωτερική κεραία στην μία περίπτωση είναι μήκους 135mm 2.5 dBi ενώ στην άλλη 320mm – 3 dBi
- Η ισχύς μετάδοσης και στις δύο είναι 14 dBm
- Το κόστος για την αγορά της LG01N είναι στα 150 € , ενώ για την Kerlink Wirnet είναι στα 1600 € .
- Και στις δύο περιπτώσεις οι gateway δηλώθηκαν στο TTN με κοινό Intergration για το end-device (Pytrack-Lopy).

## 3.5 Παράμετροι αξιολόγησης

Στις δοκιμές που κάναμε έπρεπε να έχουμε συγκριτικά στοιχεία είτε σε σχέση με άλλες έρευνες, είτε σε πειράματα που διαφοροποιούνται ως προς τις παραμέτρους που εξετάζουμε, δηλαδή περιβαλλοντικοί παράγοντες και βασικές παράμετροι LoRa.

Έτσι η συσκευή Pytrack-Lopy ρυθμίστηκε να στέλνει:

- Αριθμημένα πακέτα - για τον έλεγχο απώλειας πακέτων.
- Δεδομένα θέση (Latitude, Longitude) – για τη γεωγραφική θέση της συσκευής καθώς και τη μέτρηση της απόστασης.
- Επιπλέον δεδομένα (accelX, accelY, accelZ) - για να έχουμε ένα payload με χρήσιμη πληροφορία

Η gateway με τη σειρά της δρομολογεί στην υπηρεσία TTN τα εξής δεδομένα:

- ID πακέτου που έχει παραλάβει

- Σετ βασικών παραμέτρων ρύθμισης δικτύου LoRa (SF,BW,F,CR)
- Μετρήσεις ποιότητας σήματος RSSI, SNR, airtime.
- Χρονική σήμανση, ταυτότητα συσκευής, ταυτότητα gateway.
- Εφαρμογή λήψης και καταγραφής δεδομένων από το TTN.
- Η εφαρμογή καταγράφει κατάλληλα δεδομένα σε αρχείο σαν εναλλακτικός τρόπος καταγραφής δεδομένων από την υπηρεσία TTN.

#### Υπηρεσία Integrations Application

Καταγραφή και παρουσίαση των πειραμάτων (10000 τελευταίες μετρήσεις) με δυνατότητα να έχουμε πρόσβαση και σε gateway που έχουν στηθεί από το Πανεπιστήμιο Δυτικής Μακεδονίας.

#### Σύστημα αξιολόγησης

Με χρήση του στατιστικού πακέτου SPSS κάναμε ανάλυση μετρήσεων ποιότητας σήματος RSSI, SNR, airtime, απώλειας πακέτων και εποπτική παρουσίαση με κατάλληλα διαγράμματα:

1. Παρουσίαση πάνω σε χάρτη
2. Συγκριτικά ιστογράμματα
3. Περιγραφικά στατιστικά και μέτρα θέσης.

Οι ανάλυση έγινε σε δεδομένα που αφορούσαν τρεις διαφορετικές καταστάσεις:

- Θέσεις εκτός πόλης με τοποθέτηση σε σταθερό σημείο χωρίς σοβαρά ενδιάμεσα εμπόδια.
- Σημεία εντός πόλης με ταχύτητα το βάδισμα ενός ανθρώπου.
- Εντός και εκτός πόλης με ταχύτητα κάτω από 60 km/h .

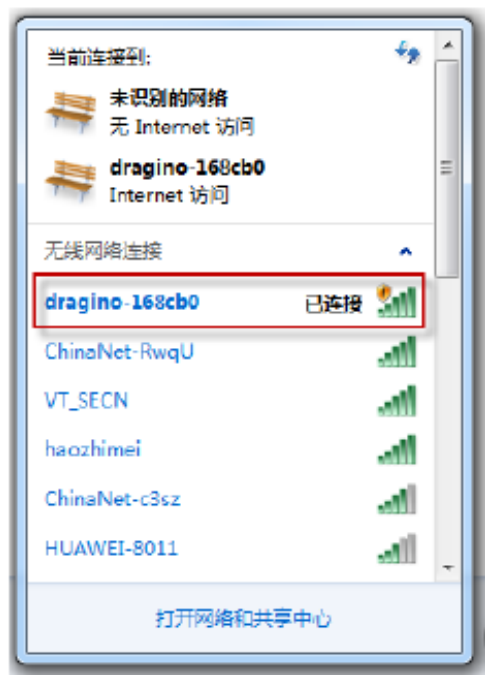
# 4

## Τεχνικές λεπτομέρειες

### 4.1 Λεπτομέρειες υλοποίησης

#### 4.1.1 Στάδια προετοιμασίας και παραμετροποίησης της LG01-N

Για την πρόσβαση και την παραμετροποίηση της gateway χρησιμοποιήθηκε το WiFi δίκτυο καθώς έτσι είναι ρυθμισμένη εργοστασιακά η gateway. Συνδέουμε την gateway με καλώδιο Ethernet στο δίκτυο και έπειτα με την βοήθεια ενός laptop ψάχνουμε να βρούμε την LG01-N στο WiFi δίκτυο όπως φαίνεται στην (Εικόνα 47).

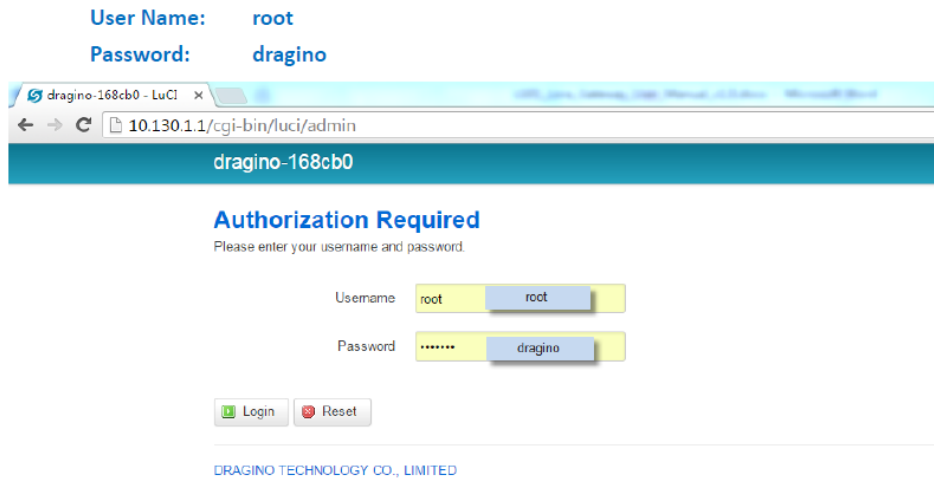


Εικόνα 47. Σύνδεση με WiFi της LG01-N

Αφότου συνδεθούμε στην gateway το laptop θα λάβει την διεύθυνση IP 10.130.1.xxx, ενώ η LG01-N από τον κατασκευαστή έχει σαν default IP την 10.130.1.1. Στην συνέχεια ανοίγουμε

έναν οποιονδήποτε browser πληκτρολογούμε στην διεύθυνση url την διεύθυνση IP 10.130.1.1. Με αυτόν τον τρόπο καταφέρνουμε να συνδεθούμε στο login interface της LG01-N.

User Name: root  
Password: dragino



Εικόνα 48. Log-in interface της LG01-N

Χρησιμοποιούμε τα defaults User Name και Password όπως φαίνεται στην εικόνα (Εικόνα 49). Το επόμενο μας βήμα είναι να κατεβάσουμε την τελευταία έκδοση του Single Packet Forward. Πηγαίνουμε στο URL:

<http://www.dragino.com/downloads/index.php?dir=motherboards/lg01/sketch/>

και κατεβάζουμε το αρχείο HEX με την τελευταία έκδοση.



Εικόνα 49. Single Packet Forward v004.

Στη συνέχεια αφού έχουμε συνδεθεί στην GateWay πηγαίνουμε στην καρτέλα **Sensor--> Flash MCU**. Κάνουμε Upload το αρχείο HEX που κατεβάσαμε προηγουμένως.

dragino-146d78
Status
Sensor
System
Network
Logout

## Micro-Controller settings

Configure correct Arduino profile will let you to able upload avr program via Arduino IDE and WiFi

### MCU Upload Profile

MCU Part

ATmega328P
Auto detected by software on boot

Profile

Arduino Uno w/ATmega328P
Auto detected by software on boot

MCU Version

single\_pkt\_fwd\_v002

Add Bootloader

☒
Add Arduino bootloader while upload

Εικόνα 50. Upload Single Packet Foward version 004.

### 4.1.2 Παραμετροποίηση LoRa Radio Settings

Στην καρτέλα Sensor--> LoRaWAN μπορούμε να παραμετροποιήσουμε τις σωστές τιμές του LoRa Radio. Ορίζουμε την συχνότητα εκπομπής TX αλλά και την συχνότητα λήψης RX ανάλογα σε ποια περιοχή βρισκόμαστε. Επίσης έχουμε την δυνατότητα να αλλάξουμε το Spreading Factor, Coding Rate και το Signal Bandwidth ανάλογα το είδος της εφαρμογής μας αλλά και τις αποστάσεις που έχει ένα Node με την Gateway.

### Radio Settings

Radio settings requires MCU side sketch support

TX Frequency

868100000
Gateway's LoRa TX Frequency

RX Frequency

868100000
Gateway's LoRa RX Frequency

Encryption Key

Encryption Key

Spreading Factor

SF7

Coding Rate

4/5

Signal Bandwidth

125 kHz

Preamble Length

8
Length range: 6 ~ 65536

Save & Apply

Save

Εικόνα 51. LoRa Radio Settings

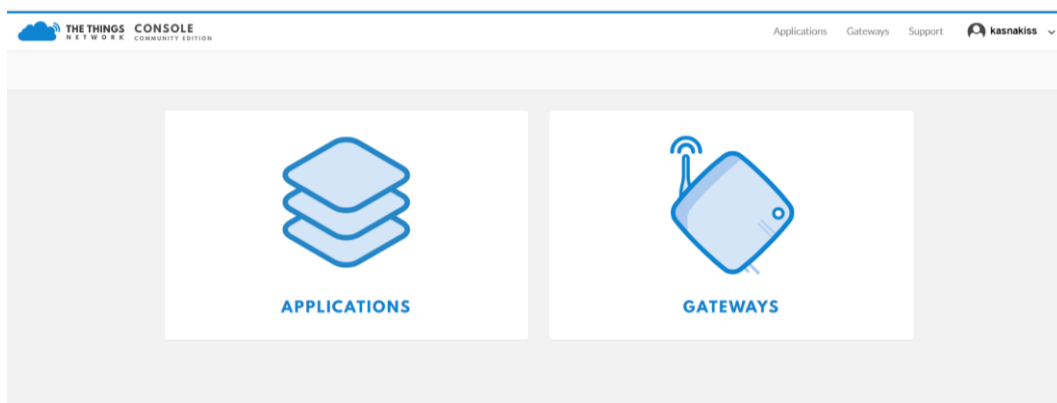
#### 4.1.3 Σύνδεση της GateWay με το The Thnings Network (TTN)

Για να κάνουμε εγγραφή την GateWay στο TTN χρειαζόμαστε την MAC address της. Η MAC address βρίσκεται στο πίσω μέρος της GateWay όπως φαίνεται στην εικόνα (Εικόνα 52).

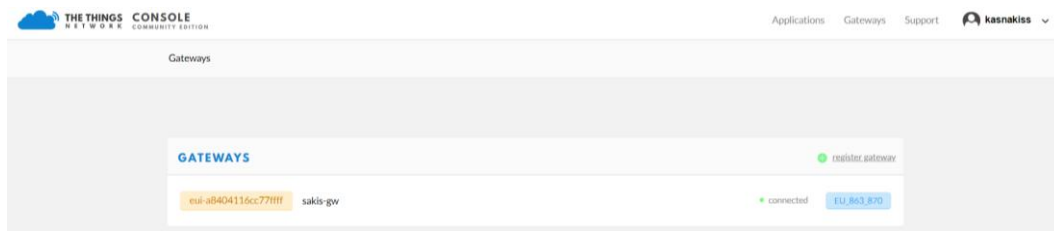


Εικόνα 52. MAC Address

Η διεύθυνση του TTN είναι: <https://www.thethingsnetwork.org/>. Αφότου κάνουμε εγγραφή στο TTN και ενεργοποιήσουμε έναν λογαριασμό πηγαίνουμε στην καρτέλα **CONSOLE--> GATEWAYS--> register gateway**.

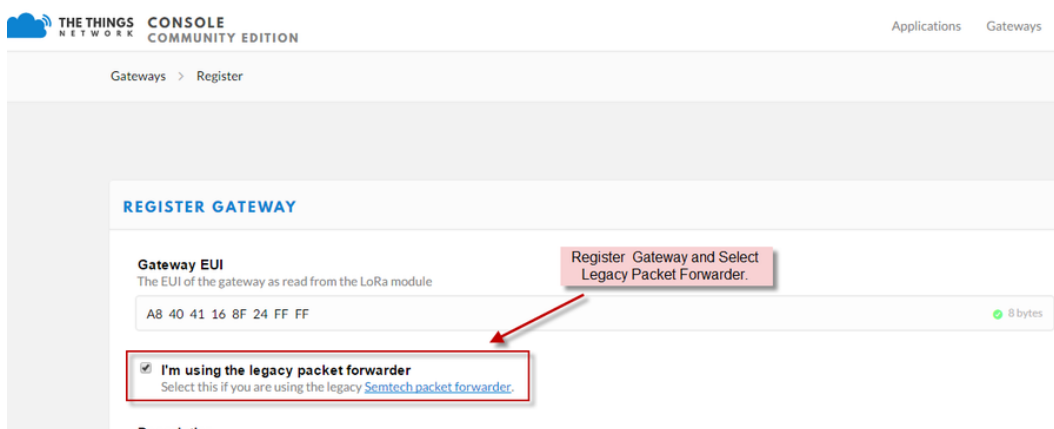


Εικόνα 53. Console TTN.



Εικόνα 54. Register Gateway

Έπειτα συμπληρώνουμε την MAC Address της GateWay και στο τέλος τοποθετούμε και τέσσερα FFFF όπως φαίνεται στην εικόνα (Εικόνα 55). Επιπλέον επιλέγουμε το κουτάκι “ I’m using the legacy packet forwarder.”



Εικόνα 55. Register GateWay

Στο επόμενο βήμα είναι να κάνουμε ρυθμίσεις στο LoRaWAN Server της GateWay ώστε να μπορεί να συνδεθεί με το The Things Network. Στην καρτέλα Sensor--> LoRaWAN εισάγουμε την **TTN Server Address** στην προκειμένη περίπτωση για Ευρώπη, **router.eu.thethings.network** με **Server Port 1700**. Στο πεδίο GateWay ID βάζουμε **την MAC Address** της Gateway και στο τέλος τέσσερα **fff**.

dragino-146d78

Status ▾Sensor ▾System ▾Network ▾Logout

## LoRa Gateway Settings

Configuration to communicate with LoRa devices and LoRaWAN server

### LoRaWAN Server Settings

Server Address

router.eu.thethings.network

Server Port

1700

Gateway ID

a84041168f24ffff

Mail Address

edwin@dragino.com

Latitude

Location Info

Longitude

Location Info

Εικόνα 56. LoRa Server Settings

Στο τελευταίο βήμα πηγαίνουμε στην καρτέλα Sensor--> IoT (Internet of Things) Server και επιλέγουμε IoT (Internet of Things) Server LoRaWAN.

dragino-1801c2

Status ▾Sensor ▾System ▾Network ▾Logout

## Select IoT Server

Select the IoT Server type to connect

**Page: Sensor --> IoT Server**

### Select IoT Server

IoT Server

LoRaWAN ▾

Select LoRaWAN as IoT Server

Log Debug Info

Disable Debug Info ▾

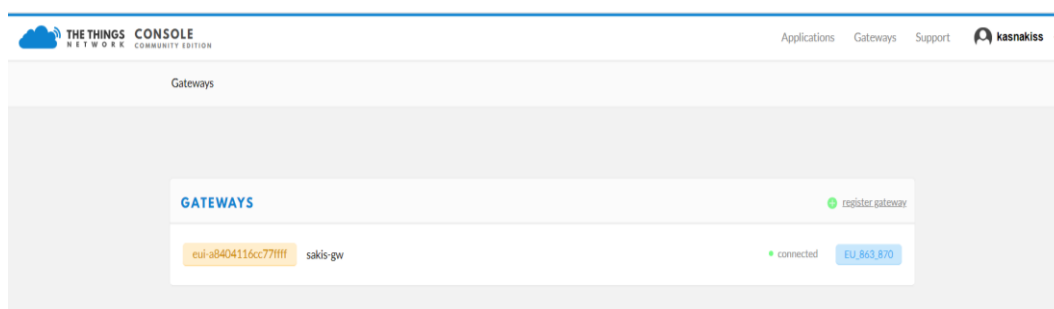
☒ Show Log in System Log

Save &

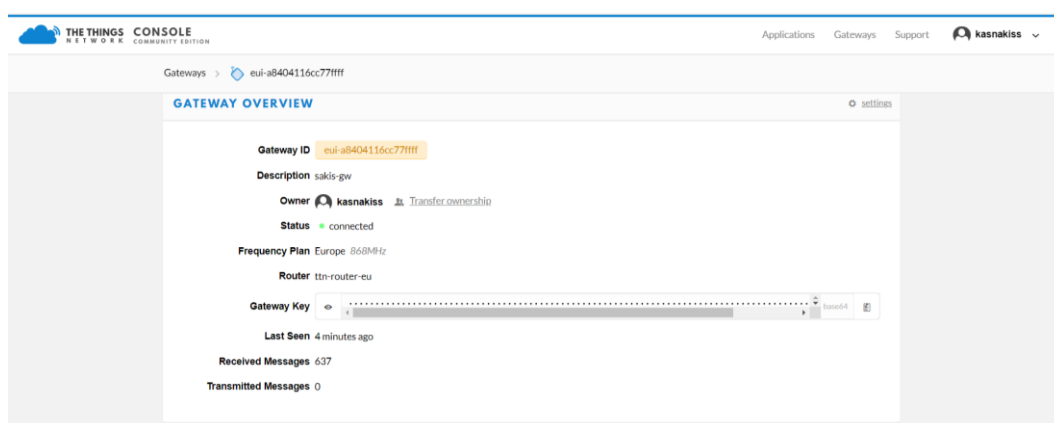
DRAGINO TECHNOLOGY CO., LIMITED

Εικόνα 57. Select IoT (Internet of Things) Server.

Αφού έχουμε ολοκληρώσει όλα αυτά τα βήματα στην σελίδα του TTN και στην καρτέλα CONSOLE μπορούμε να διακρίνουμε ότι η GateWay έχει συνδεθεί με το TTN όπως φαίνεται στην (Εικόνα 58).



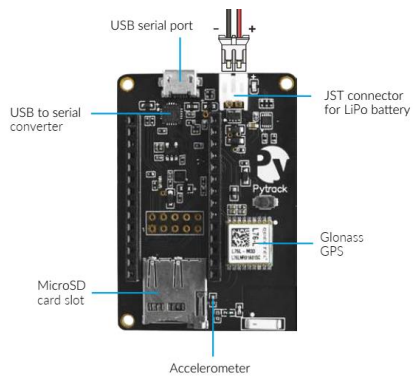
Εικόνα 58 - Διαθέσιμες gateways



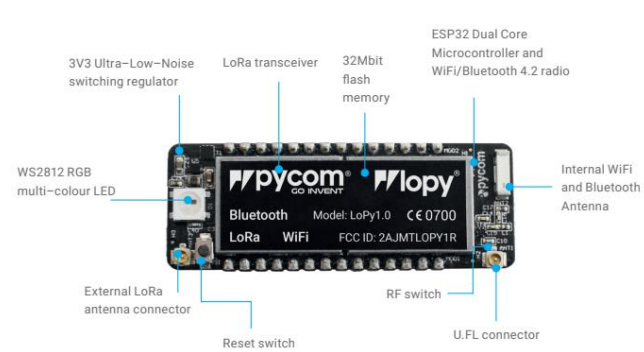
Εικόνα 59. Επιτυχής σύνδεση της GateWay με το TTN

## 4.2 LoPy – Pycom-Pytrack

Συσκευή αποστολής δεδομένων είναι ένας συνδυασμός ενός LoRa module (Lopy) και ενός GPS module (Pytrack).



Εικόνα 60 - Pytrack



Εικόνα 61 - Pycom

Το Lopy module που χρησιμοποιήθηκε για τις δοκιμές μας μπορεί τυπικά να δεχτεί τις παραμετροποιήσεις που φαίνονται στον πίνακα( Πίνακας 1 ).

Πίνακας 1 – Παράμετροι λειτουργίας Pycom

| Bandwidth (KHz) | Spreading Factor | Nominal Rb (bps) | Sensitivity (dBm) |
|-----------------|------------------|------------------|-------------------|
| 125             | 6                | 9380             | -122              |
| 125             | 12               | 293              | -137              |
| 250             | 6                | 18750            | -119              |
| 250             | 12               | 588              | -134              |
| 500             | 6                | 37500            | -116              |
| 500             | 12               | 1172             | -131              |

Η υλοποίηση του κώδικα έγινε σε Python και αυτό που αναλαμβάνει είναι να στέλνει πακέτα με δεδομένα του GPS και τους Accelerometer με διαφορετικές παραμετροποιήσεις στη gateway.

Παράλληλα για λόγους ελέγχου μετάδοσης δεδομένων μπορεί να στέλνει τα δεδομένα σε έναν τοπικό MQTT server στον οποίο γίνεται και καταγραφή των μετρήσεων κατευθείαν από το Lopy.

Ο προγραμματισμός έγινε με χρήση του προγραμματιστικού περιβάλλοντος Atom με χρήση Python. Επίσης τα δεδομένα μορφοποιήθηκαν με χρήση της βιβλιοθήκης CayenneLPP.

Οι βιβλιοθήκες που επιπλέον συμπεριλήφθηκαν στο έργο μας ήταν :

LoRa για τη διαχείριση του LoRa module , η CayenneLPP για την κωδικοποίηση των δεδομένων, οι LIS2HH12 και L76GNSS για την επικοινωνία με τους αισθητήρες accelerometer και gps του pytrack, την MQTTClient για την αποστολή πακέτων με χρήση WiFi, και η Wlan

για την επικοινωνία του συστήματος με WLAN. Η επικοινωνία μέσω Wlan χρησιμοποιήθηκε κυρίως για εκσφαλμάτωση.

```
from network import LoRa
import struct
import binascii
import socket
import time
import uos
import array
import machine
import pycom
from CayenneLPP import CayenneLPP
from LIS2HH12 import LIS2HH12
from L76GNSS import L76GNSS
from pytrack import Pytrack
from mqtt import MQTTClient
from network import WLAN
```

**Σε πρώτη φάση δημιουργούμε κατάλληλα αντικείμενα που κληρονομούν τις ιδιότητες των κλάσεων PyTrack, LIS2HH12, CayenneLPP, L76GNSS**

```
py = Pytrack()
li = LIS2HH12(py)
lpp = CayenneLPP()
l76 = L76GNSS(py, timeout=30)
```

**Για το LoRa ορίζουμε ότι θα χρησιμοποιήσουμε LORAWAN στην συχνότητα που είναι προκαθορισμένη για Ευρώπη , δηλαδή EU868.**

```
lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868)
```

**Για την επικοινωνία με ABP (Activation By Personalized) χρειάζονται τρεις παράμετροι. Η Device Address , η Network Key και η Application Key.**

**Επειδή μπορούμε σε ένα application να έχουμε παραπάνω Devices η μεταβλητή dev-addr έχει οριστεί σαν πίνακας.**

```
dev_addr = struct.unpack(">I", binascii.unhexlify('26011CF5'))[0]
print("device address")
```

```
print(dev_addr)

nwk_swkey = binascii.unhexlify('A1E45D3DA759BB2CD568765E972DFB32')

app_swkey = binascii.unhexlify('7462D500FA761F2B9B59CA6F885CF96C')
```

**Παραμετροποιούμε επίσης το ασύρματο δίκτυο για αποστολή των δεδομένων και στο τοπικό δίκτυο κυρίως σε επίπεδο εκσφαλμάτωσης.**

**Επίσης χρησιμοποιούμε ένα χρωματικό κώδικα για τις διάφορες καταστάσεις του συστήματος με την `rgbled()`.**

```
wlan = WLAN(mode=WLAN.STA)

wlan.connect("COSMOTE-346020", auth=(WLAN.WPA2, "vfD5kyZHDRpzu4RP"),
timeout=5000)

pycom.heartbeat(False)

#pycom.rgbled(0x0000ff)

#while not wlan.isconnected():

#  machine.idle()

#  pycom.rgbled(0xff0000)

#print("Connected to WiFi\n")

#pycom.rgbled(0x0ff00)

#time.sleep(1)

pycom.rgbled(0x000ff)
```

**Γίνεται σύνδεση σε ένα τοπικό MQTT server κυρίως για λόγους εκσφαλμάτωσης.**

```
#client = MQTTClient("pycom", "192.168.1.2", port=1883)

#try:

#  client.connect()

#  ok=True

#  pycom.rgbled(0x00ff00)

#except:

#  ok=False

#  pycom.rgbled(0x000ff)

#  time.sleep(1)

#  pycom.rgbled(0x00ff00)
```

**Δημιουργούμε μια λίστα ζευγών που αφορούν την παραμετροποίηση data rate και SF γιατί στο LoRaWAN έχουμε τυποποιημένη παραμετροποίηση.**

```
sflist=[[0,12],[1,11],[2,10],[3,9],[4,8],[5,7]]
```

**Για σταθερή τιμή των παραμέτρων Bandwidth και Frequency**

```
lora.bandwidth(LoRa.BW_125KHZ)
```

```
lora.frequency(868300000)
```

**και για κάθε ζευγάρι τις λίστες παραμέτρων γίνεται join στο TTN και προετοιμάζεται η αποστολή δεδομένων μέσω ενός Socket που δημιουργείται.**

```
for sf in sflist:
```

```
    lora.sf(sf[1])
```

```
    lora.join(activation=LoRa.ABP, auth=(dev_addr, nwk_swkey, app_swkey))
```

```
    print(lora.has_joined())
```

```
    print(lora.sf())
```

```
    print(lora.bandwidth())
```

```
    print(lora.frequency())
```

```
    time.sleep(1)
```

```
    print("create a LoRa socket ")
```

```
    s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)
```

```
    s.setsockopt(socket.SOL_LORA, socket.SO_DR, sf[0])
```

```
    s.setblocking(False)
```

**Στέλνουμε 10 πακέτα που το καθένα περιλαμβάνει δεδομένα που προέρχονται από τους αισθητήρες τοποθετώντας πρώτα σε ένα buffer του Cayenne LPP , το οποίο αναλαμβάνει και την κωδικοποίηση σε byte . Στην περίπτωση που το gps δεν λαμβάνει σήμα τότε στέλνονται μηδενικές τιμές.**

```
for x in range(1, 10):
```

```
    pycom.rgbled(0xff0000)
```

```
    time.sleep(1)
```

```
    s.setblocking(True)
```

```
    print("\n\n** 3-Axis Accelerometer (LIS2HH12)")
```

```
    print('Acceleration', li.acceleration())
```

```
    print('Roll', li.roll())
```

```
    print('Pitch', li.pitch())
```

**Προσθήκη δεδομένων επιταχυνσιόμετρου**

```
lpp.add_accelerometer(1, li.acceleration()[0], li.acceleration()[1], li.acceleration()[2])
```

```
lpp.add_gyrometer(1, li.roll(), li.pitch(), 0)

print("\n\n** GNSS/GPS module')
```

### **Προσθήκη δεδομένων GPS**

```
gpsResult = l76.coordinates()

print('Coordinates', gpsResult)

if gpsResult == (None, None):
    gpsResult=(0,0)

if gpsResult == (None, None):
    print("No GPS fix")
    pycom.rgbled(0xff0000)

    lpp.reset()

else:
    print("GPS lat: " + str(gpsResult[0]) + " long: " + str(gpsResult[1]))
    lpp.add_gps(1, gpsResult[0], gpsResult[1],2)
    pycom.rgbled(0x00ff00)
    print('Sending data (uplink)...')
```

### **Αποστολή πακέτου μέσω LoRaWAN Socket**

```
s.send(bytes(lpp.get_buffer()))

# stats=lora.stats()

# print(stats)
```

### **Αποστολή στατιστικών πακέτου τοπικά για εκσφαλμάτωση**

```
# if ok==True:

# client.publish(topic="/pycom",msg="{:d}".format(stats[7]))

time.sleep(2)

s.setblocking(False)

lpp.reset()

#data = s.recv(64)

#print('Received data (downlink)', data)

# print("stats")

pycom.rgbled(0x0000ff)

time.sleep(1)

pycom.heartbeat(True)
```

Για την καταγραφή των δεδομένων από το TTN δημιουργήθηκε μια εφαρμογή με υλοποίηση σε python , η οποία μπορεί να καταγράφει τα δεδομένα σε αρχείο, να δημιουργεί ένα διάγραμμα με βοήθεια μιας βιβλιοθήκης matLab και επίσης να τοποθετεί σημεία πάνω σ' ένα χάρτη Google. Ακόμη υπολογίζει την απόσταση της συσκευής από τη gateway με γεωτοποθέτηση.

**Απαραίτητες βιβλιοθήκες για Python:** Η paho.mqtt.client για την επικοινωνία με το TTN μέσω του πρωτοκόλλου MQTT.

Οι Json και base64 χρησιμοποιήθηκαν για την αποκωδικοποίηση των πακέτων που λαμβάνονται από το TTN μέσω MQTT.

Η pygmaps χρησιμοποιήθηκε για την απεικόνιση δεδομένων πάνω σε χάρτη Google. Λόγω περιορισμού της χρήσης του API Maps της Google απενεργοποιήθηκε.

Η geopy.distance από τη βιβλιοθήκη geodesic χρησιμοποιήθηκε για τη μέτρηση της απόστασης μεταξύ gateway και end node.

Επίσης χρησιμοποιήθηκε η matplotlib μόνο δοκιμαστικά για την online γραφική απεικόνιση δεδομένων.

```
import paho.mqtt.client as mqtt
import json
import base64
#import pygmaps
from geopy.distance import geodesic
APPEUI = '70B3D57ED001033A'
APPID = 'sakis-app'
PSW = 'ttn-account-v2.5LMhecPeJ1eJJ3-NCMO8Ruekp6odzfi1asoY7nYjQTE'
#import matplotlib.pyplot as plt
#import DataPlot and RealtimePlot from the file plot_data.py
#from plot_data import DataPlot, RealtimePlot
#fig, axes = plt.subplots()
#plt.title('Data from TTN console')

#data = DataPlot()
#dataPlotting= RealtimePlot(axes)
#mymap = pygmaps.maps(40.29,21.801, 15)
```

```
count=0
```

```
airtime=0
```

**Συνάρτηση για τη μετατροπή από byte σε δεκαδικό.**

```
def bytes_to_decimal(i,d):
```

```
    xx = i - 127
```

```
    dec = (-d if xx < 0 else d)/100
```

```
    return xx + dec
```

**Ορισμός ενέργειας συμβάντος κατά τη σύνδεση στο TTN : Επιστρέφει τη λίστα συσκευών που έχουν εγγραφή στο application με ταυτότητα την APPEUI**

```
def on_connect(client, userdata, flags, rc):
```

```
    client.subscribe('+/devices/+/up'.format(APPEUI))
```

**Ορισμός ενέργειας συμβάντος κατά τη λήψη μηνύματος: Το μήνυμα αποκωδικοποιείται από ακολουθία bytes σε json format utf8. Αποσπώνται τα πεδία που μας ενδιαφέρουν και τοποθετούνται σε κατάλληλα πεδία.**

```
def on_message(client, userdata, msg):
```

```
    j_msg = json.loads(msg.payload.decode('utf-8'))
```

```
    mcounter = j_msg['counter']
```

```
    dev_eui = j_msg['hardware_serial']
```

```
    mtime=j_msg['metadata']['time']
```

```
    mfreq=j_msg['metadata']['frequency']
```

```
    mdrate=j_msg['metadata']['data_rate']
```

```
    mairtime=j_msg['metadata']['airtime']
```

```
    lat=j_msg['payload_fields']['gps_1']['latitude']
```

```
    lon=j_msg['payload_fields']['gps_1']['longitude']
```

```
    snr=j_msg['metadata']['gateways'][0]['snr']
```

```
    rssi=j_msg['metadata']['gateways'][0]['rssi']
```

**Τοποθετούνται σημεία πάνω στο χάρτη**

```
    #mymap.addradpoint(lat,lon,1, "#0000FF")
```

**Υπολογίζεται η απόσταση μεταξύ gateway και pycom**

```
    gateway = (40.3050998,21.7964435)
```

```
    pycom = (lat,lon)
```

```
    apostasi=geodesic(gateway, pycom).meters
```

```
print(apostasi)
```

```
print('---')
```

**Γίνεται εγγραφή των δεδομένων που μας ενδιαφέρουν σε αρχείο (ASCII)**

```
f = open("myfile.txt", "a+")
```

```
f.write("%d,%7.5f,%7.5f,%7.2f,%5.2f,%5.2f,%9.2f\n" %
```

```
(mcounter,lat,lon,apostasi,snr,rssi,mairtime))
```

```
f.close()
```

```
print('latitude:', lat, ' longitude:', lon, 'snr:',snr,'rssi:',rssi,'airtime:',mairtime)
```

```
print('dev eui: ', dev_eui)
```

```
# plot data
```

```
#global count
```

```
#count+=1
```

```
#data.add(count,snr,rssi)
```

```
#dataPlotting.plot(data)
```

```
#plt.pause(0.001)
```

```
print('ok')
```

```
# set paho.mqtt callback
```

**Προετοιμάζουμε τη σύνδεση με το TTN μέσω MQTT : Ορίζουμε τι θα εκτελείται για τα συμβάντα on\_connect, on\_message , ορίζουμε username και password και κάνουμε σύνδεση στην προκαθορισμένη θύρα για την υπηρεσία MQTT του TTN.**

```
ttn_client = mqtt.Client()
```

```
print('test2')
```

```
ttn_client.on_connect = on_connect
```

```
print('test3')
```

```
ttn_client.on_message = on_message
```

```
print('test4')
```

```
ttn_client.username_pw_set(APPID, PSW)
```

```
ttn_client.connect("eu.thethings.network", 1883, 60) #MQTT port over TLS
```

```
try:
```

**Επαναληπτικά διαβάζουμε δεδομένα μέχρι να γίνει τερματισμός με χρήση του πηκτρολογίου. Κατά την ολοκλήρωση γίνεται παρουσίαση του χάρτη. (έχει απενεργοποιηθεί).**

```

ttn_client.loop_forever()
except KeyboardInterrupt:
    print('disconnect')
    #mymap.draw('./mymap.html')
    ttn_client.disconnect()

```

### 4.3 Προσομοίωση στο MatLab

#### Συγκρούσεις για SF7 – SF12

```

timespan = 60*1000; %ms
timeinterval = 10; %ms
nrofslots = timespan/timeinterval;

ορίζουμε ότι το εύρος κάθε καναλιού θα είναι 125KH και θα χρησιμοποιούμε 6 κανάλια  
στη συχνότητα 868.7 – 869.2. Η παρουσίαση στο γράφημα θα γίνεται με βήμα 100HZ

freqspan = 125e3; %hz : 125khz -> 868.7 - 869.2 MHZ
freqinterval = 100; %hz
nrofchannels = 6;
ft = zeros(nrofslots, nrofchannels);
nrofdevices = 1000;
nrofpackets = 1;
packetduration = 2000;

Ορίζουμε έναν πίνακα παραμέτρων στον οποίο για κάθε SF ορίζεται και ο αντίστοιχος  
ρυθμός μετάδοσης αλλά και ο χρόνος απασχόλησης του καναλιού για ένα πακέτο  
συνολικού μεγέθους 38 bytes(13 head + 25 payload)

%LORA
% SF bitrate(bps) duration (for 38 bytes 13+25)
lora_duration = [12 293 1974.3;
    11 547 1069.1;
    10 976 534.5;
    09 1757 287.7;
    08 3125 154.1;
    07 5478 82.2;
    06 9375 41.1];

```

```
packetduration = lora_duration(1,3);
```

**Δημιουργούμε αν διάνυσμα στο οποίο καταγράφουμε αν υπάρχει σύγκρουση ή όχι(1,0). Αρχική τιμή 0. Ύπαρξη σύγκρουσης τροποποιεί την τιμή σε 1.**

```
collision = zeros(nrofdevices, nrofpackets);
```

**Παράγουμε ψευδοτυχαίες τιμές που ακολουθούν ομοιόμορφη κατανομή με ελάχιστη τιμή το κανάλι 1 και μέγιστη τιμή nrofchannels και τις αποθηκεύουμε σε πίνακα μεγέθους nrofdevices γραμμών και nrofpackets στηλών.**

```
sf = randi([1 nrofchannels], [nrofdevices nrofpackets]);
```

**Παράγουμε ψευδοτυχαίες τιμές που ακολουθούν ομοιόμορφη κατανομή με ελάχιστη τιμή το 1 και μέγιστη τιμή τη στρογγυλοποίηση του της διαίρεσης της διάρκειας μετάδοσης προς το διάστημα χρόνου, αν αυτό αφαιρεθεί από τον αριθμό των slots και τις αποθηκεύουμε σε πίνακα μεγέθους nrofdevices γραμμών και nrofpackets στηλών.**

```
time = randi([1 floor(nrofslots - (packetduration*nrofpackets)/timeinterval)], [nrofdevices 1]);
```

**Για κάθε συσκευή και με βάση τον τυχαίο χρόνο απασχόλησης του καναλιού ελέγχουμε αν μπορεί να εξυπηρετηθεί. Σε κάθε περίπτωση η κατάσταση 0,1,2 αποθηκεύεται στον πίνακα ft ο οποίος θα μετατραπεί στη συνέχεια ένα γράφημα.**

**Το σκεπτικό είναι απλό. Αρχικά το κανάλι είναι ελεύθερο (0) , αν ζητηθεί εξυπηρέτηση από κάποιο πακέτο και μπορεί να εξυπηρετηθεί τότε μπαίνει σε κατάσταση απασχολημένο (1) . Αν είναι απασχολημένο και ζητηθεί εξυπηρέτηση από κάποιο πακέτο τότε έχουμε σύγκρουση(2).**

```
for i = 1:nrofdevices
```

```
    time_offset = time(i, 1);
```

```
    for p = 1:nrofpackets
```

```
        for k = -1:1:1
```

```
            if (sf(i, p)+k<1) | (sf(i, p)+k>nrofchannels)
```

```
                continue
```

```
            end
```

```
            duration = lora_duration(sf(i, p),3);
```

```
            for j = 1:ceil(duration/timeinterval)
```

```
                %freq(i, 1)
```

```
                if j+time_offset > nrofslots
```

```
                    continue
```

```
                end
```

```

        if ft(j+time_offset, sf(i, p)+k) == 0
            ft(j+time_offset, sf(i, p)+k) = 1;
        else
            ft(j+time_offset, sf(i, p)+k) = 2;
            collision(i, p) = 1;
        end
    end
end

time_offset = time_offset + duration/timeinterval;
end

fail = sum(collision, 2) == nrofpackets;
sum(fail)
ft(1,1) = 1;
ft(1,2) = 2;
ft = [ft ones(6000,1)];
Εκκίνηση γραφήματος
figure1 = figure();

Δημιουργούμε ένα χρωματικό κώδικα
map2 = [0 0 0; 0 1 0; 0 0.9 0; 0.9 0 0];
colormap(map2);

Με βάση τις τιμές του πίνακα ft και το χρωματικό κώδικα γίνεται απεικόνιση του ft.
pcolor(ft);
shading flat;

Στη συνέχεια γίνεται προσθήκη των τίτλων και ετικετών στους άξονες.
titlestring = sprintf('LoRa packet collision simulation withing 125 kHz with %d devices
\ntransmitting randomly within 60 seconds', ...
    nrofdevices);
title(titlestring);
xlabel('Logical channel (Spreading Factor)' ) % x-axis label
ylabel('Time (10 ms)' ) % y-axis label
ax = gca;

```

```
set(gca,'XTickLabel',{'12','11','10','9','8','7',''});
```

**Το αποτέλεσμα το αποθηκεύουμε σε αρχεία τύπου png και fig.**

```
saveas(figure1, sprintf('lora_spectrum_%d_dev.fig',nrofdevices));
```

```
saveas(figure1, sprintf('lora_spectrum_%d_dev.png',nrofdevices))
```

## **4.4 End Nodes**

**Περίπτωση Arduino Mega – Dragino Gps ή ESP**

Έγινε χρήση των βιβλιοθηκών HAL (hardware abstraction layer) και LMIC (LoRaMAC-in-C) της IBM για το LoRa Shield, καθώς και της TinyGPS για το GPS module. Για την κωδικοποίηση των δεδομένων χρησιμοποιήθηκε η βιβλιοθήκη Cayenne LPP.

```
#include <lmic.h>
```

```
#include <hal/hal.h>
```

```
#include <SoftwareSerial.h>
```

```
#include <TinyGPS.h>
```

```
#include <SPI.h>
```

```
#include <CayenneLPP.h>
```

```
TinyGPS gps;
```

**Το GPS συνδέθηκε μέσω σειριακής**

```
SoftwareSerial ss(50,51 ); // 34,12 για ESP32
```

**Κάνουμε χρήση ενός buffer μεγέθους 51 bytes**

```
CayenneLPP lpp(51);
```

**Χρησιμοποιούμε ABP (Activation By Personalize) οπότε χρειαζόμαστε τρεις παραμέτρους για τη σύνδεση στο TTN.( NwkSKey, AppSKey, DevAddr)**

```
// LoRaWAN NwkSKey, network session key
```

```
// This is the default Semtech key, which is used by the early prototype TTN
```

```
// network.
```

```
static const PROGMEM u1_t NWKSKEY[16] =
```

```
{0xEA,0x9A,0x27,0xC5,0x84,0x20,0xFE,0x89,0x93,0x93,0xAC,0x0F,0x14,0x3F,0x11,0x74};
```

```
// LoRaWAN AppSKey, application session key
```

```
// This is the default Semtech key, which is used by the early prototype TTN
```

```

// network.

static const u1_t PROGMEM APPSKEY[16] =
{0x15,0x80,0x8A,0x3F,0x6E,0xB3,0xB3,0x00,0xA9,0x60,0x64,0x01,0xFC,0x18,0xFB,0xD
8};

// LoRaWAN end-device address (DevAddr)

static const u4_t DEVADDR = 0x26013DE8 ; // <-- Change this address for every node!

// These callbacks are only used in over-the-air activation, so they are
// left empty here (we cannot leave them out completely unless
// DISABLE_JOIN is set in config.h, otherwise the linker will complain).

void os_getArtEui (u1_t* buf) { }

void os_getDevEui (u1_t* buf) { }

void os_getDevKey (u1_t* buf) { }

static osjob_t sendjob;

```

**Καθορίζουμε την ανανέωση για αποστολή δεδομένων τα 30 sec**

```

// Schedule TX every this many seconds (might become longer due to duty
// cycle limitations).

const unsigned TX_INTERVAL = 30;

```

**Προσδιορίζουμε τη σύνδεση του LoRa module με το Arduino Mega ή το ESP32 ή το Arduino Uno.**

```

// Pin mapping

const lmic_pinmap lmic_pins = {
    .nss = 10,//10
    .rxtx = LMIC_UNUSED_PIN,
    .rst = 9,//9
    .dio = {2,6,7},//2,6,7
};

```

**Χειριζόμαστε τα συμβάντα για το επίπεδο MAC**

```

void onEvent (ev_t ev) {
    Serial.print(os_getTime());
    Serial.print(": ");
    Serial.println(ev);
    switch(ev) {

```

```

case EV_SCAN_TIMEOUT:
    Serial.println(F("EV_SCAN_TIMEOUT"));
    break;
case EV_BEACON_FOUND:
    Serial.println(F("EV_BEACON_FOUND"));
    break;
case EV_BEACON_MISSED:
    Serial.println(F("EV_BEACON_MISSED"));
    break;
case EV_BEACON_TRACKED:
    Serial.println(F("EV_BEACON_TRACKED"));
    break;
case EV_JOINING:
    Serial.println(F("EV_JOINING"));
    break;
case EV_JOINED:
    Serial.println(F("EV_JOINED"));
    break;
case EV_RFU1:
    Serial.println(F("EV_RFU1"));
    break;
case EV_JOIN_FAILED:
    Serial.println(F("EV_JOIN_FAILED"));
    break;
case EV_REJOIN_FAILED:
    Serial.println(F("EV_REJOIN_FAILED"));
    break;
case EV_TXCOMPLETE:

```

**με την ολοκλήρωση μιας αποστολής εκτυπώνονται κατάλληλα μηνύματα για debug και προγραμματίζεται η επόμενη αποστολή.**

```

    Serial.println(F("EV_TXCOMPLETE (includes waiting for RX windows)"));
    if (LMIC.dataLen) {

```

```

        Serial.println(F("Received "));
        Serial.println(LMIC.dataLen);
        Serial.println(F(" bytes of payload"));
    }
    // Schedule next transmission
    os_setTimedCallback(&sendjob,os_getTime()+sec2osticks(TX_INTERVAL),
do_send);

    break;
case EV_LOST_TSYNC:
    Serial.println(F("EV_LOST_TSYNC"));
    break;
case EV_RESET:
    Serial.println(F("EV_RESET"));
    break;
case EV_RXCOMPLETE:
    // data received in ping slot
    Serial.println(F("EV_RXCOMPLETE"));
    break;
case EV_LINK_DEAD:
    Serial.println(F("EV_LINK_DEAD"));
    break;
case EV_LINK_ALIVE:
    Serial.println(F("EV_LINK_ALIVE"));
    break;
default:
    Serial.println(F("Unknown event"));
    break;
}
}

```

**Βασική μέθοδος που τροποποιήθηκε είναι η do\_send()**

```

void do_send(osjob_t* j){
    // Check if there is not a current TX/RX job running

```

```
float flat, flon,falt;
```

```
unsigned long age;
```

**Γίνεται λήψη δεδομένων από το GPS**

```
gps.f_get_position(&flat, &flon, &age);
```

```
falt=gps.f_altitude(); //get altitude
```

```
smartdelay(1000);
```

**Ομαλοποιούνται οι τιμές στην περίπτωση που δεν υπάρχουν διαθέσιμες τιμές**

```
flon == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flon, 6;//save six decimal places
```

```
flat == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : flat, 6;
```

```
falt == TinyGPS::GPS_INVALID_F_ANGLE ? 0.0 : falt, 2;//save two decimal places
```

```
Serial.print(flon);
```

**Κάνουμε reset στον buffer του αντικειμένου lpp**

```
lpp.reset();
```

**Προσθέτουμε τις συντεταγμένες στο αντικείμενο lpp**

```
lpp.addGPS(3, flat, flon, falt);
```

**Κάνουμε αποστολή των δεδομένων μέσω LoRaWAN.**

```
// send data
```

```
if (LMIC.opmode & OP_TXRXPEND ) {
```

```
    Serial.println(F("OP_TXRXPEND, not sending"));
```

```
} else {
```

```
    // Prepare upstream data transmission at the next possible time.
```

```
    LMIC_setTxData2(1, lpp.getBuffer(), lpp.getSize(), 0);
```

```
    Serial.println(F("Packet queued"));
```

```
}
```

```
// Next TX is scheduled after TX_COMPLETE event.
```

```
}
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    ss.begin(9600);
```

```
    Serial.println(F("Starting"));
```

```
    // LMIC init
```

```

os_init();

// Reset the MAC state. Session and pending data transfers will be discarded.
LMIC_reset();

// Set static session parameters. Instead of dynamically establishing a session
// by joining the network, precomputed session parameters are be provided.

```

### **Σύνδεση στο TTN με τις παραμέτρους της συσκευής**

```

#ifdef PROGMEM

// On AVR, these values are stored in flash and only copied to RAM
// once. Copy them to a temporary buffer here, LMIC_setSession will
// copy them into a buffer of its own again.
uint8_t appskey[sizeof(APPSKEY)];
uint8_t nwkskey[sizeof(NWKSKEY)];
memcpy_P(appskey, APPSKEY, sizeof(APPSKEY));
memcpy_P(nwkskey, NWKSKEY, sizeof(NWKSKEY));
LMIC_setSession (0x1, DEVADDR, nwkskey, appskey);
#else

// If not running an AVR with PROGMEM, just use the arrays directly
LMIC_setSession (0x1, DEVADDR, NWKSKEY, APPSKEY);
#endif

#if defined(CFG_eu868)

// Set up the channels used by the Things Network, which corresponds
// to the defaults of most gateways. Without this, only three base
// channels from the LoRaWAN specification are used, which certainly
// works, so it is good for debugging, but can overload those
// frequencies, so be sure to configure the full frequency range of
// your network here (unless your network autoconfigures them).
// Setting up channels should happen after LMIC_setSession, as that
// configures the minimal channel set.

// NA-US channels 0-71 are configured automatically

```

**Για να μην χρησιμοποιούμε μόνο τα βασικά κανάλια του LoRaWAN που είναι προκαθορισμένα για τη σύνδεση στο TTN , ορίζουμε όλες τις δυνατές επιλογές, δηλαδή 8 κανάλια επικοινωνίας.**

```

    LMIC_setupChannel(0, 868100000, DR_RANGE_MAP(DR_SF12, DR_SF7),
BAND_CENTI);    // g-band

    LMIC_setupChannel(1, 868300000, DR_RANGE_MAP(DR_SF12, DR_SF7B),
BAND_CENTI);    // g-band

    LMIC_setupChannel(2, 868500000, DR_RANGE_MAP(DR_SF12, DR_SF7),
BAND_CENTI);    // g-band

    LMIC_setupChannel(3, 867100000, DR_RANGE_MAP(DR_SF12, DR_SF7),
BAND_CENTI);    // g-band

    LMIC_setupChannel(4, 867300000, DR_RANGE_MAP(DR_SF12, DR_SF7),
BAND_CENTI);    // g-band

    LMIC_setupChannel(5, 867500000, DR_RANGE_MAP(DR_SF12, DR_SF7),
BAND_CENTI);    // g-band

    LMIC_setupChannel(6, 867700000, DR_RANGE_MAP(DR_SF12, DR_SF7),
BAND_CENTI);    // g-band

    LMIC_setupChannel(7, 867900000, DR_RANGE_MAP(DR_SF12, DR_SF7),
BAND_CENTI);    // g-band

    LMIC_setupChannel(8, 868800000, DR_RANGE_MAP(DR_FSK, DR_FSK),
BAND_MILLI);    // g2-band

    // TTN defines an additional channel at 869.525Mhz using SF9 for class B
    // devices' ping slots. LMIC does not have an easy way to define set this
    // frequency and support for class B is spotty and untested, so this
    // frequency is not configured here.
    #elif defined(CFG_us915)

    // NA-US channels 0-71 are configured automatically
    // but only one group of 8 should (a subband) should be active
    // TTN recommends the second sub band, 1 in a zero based count.
    // https://github.com/TheThingsNetwork/gateway-conf/blob/master/US-global_conf.json
    LMIC_selectSubBand(1);

    #endif

    // Disable link check validation
    LMIC_setLinkCheckMode(0);

    // TTN uses SF9 for its RX2 window.
    LMIC.dn2Dr = DR_SF9;

```

```

// Set data rate and transmit power for uplink (note: txpow seems to be ignored by the library)
LMIC_setDrTxpow(DR_SF7,14);

// Start job
do_send(&sendjob);
}

```

**Επειδή διαβάζουμε τις συντεταγμένες από το GPS σειριακά θα πρέπει οι επόμενες τιμές να λαμβάνονται μετά από κάποιο χρονικό διάστημα για να μην έχουμε συγκρούσεις στη σειριακή.**

```

static void smartdelay(unsigned long ms)
{
    unsigned long start = millis();
    do
    {
        while (ss.available())
        {
            //ss.print(Serial.read());
            gps.encode(ss.read());
        }
    } while (millis() - start < ms);
}

void loop() {
    os_runloop_once();
}

```

# 5

## Οργάνωση πειραμάτων

### 5.1.1 Βασική παραμετροποίηση

Το LoRa χρησιμοποιεί μια μοναδική παραλλαγή της διαμόρφωσης του CSS με την ενσωμάτωση τεσσάρων βασικών παραμέτρων που είναι το κανάλι εκπομπής (channel), το εύρος συχνότητας (bandwidth), ο παράγοντας διασποράς (spreading factor), και η ισχύς μετάδοσης (transmission power).

Όπως είδαμε, το LoRa χρησιμοποιεί ένα συνδυασμό up-chirp και down-chirp για να διαμορφώσει ένα πλήρες πακέτο. Η παράμετρος angle of chirps ή γνωστή σαν παράγοντας διασποράς (SF), παραμένει συνεπής σε ολόκληρο το πακέτο. Για ένα σύστημα με σταθερή βασική συχνότητα μετάδοσης και εύρος ζώνης συχνότητας, η παράμετρος SF καθορίζει το ρυθμό μετάδοσης δεδομένων.

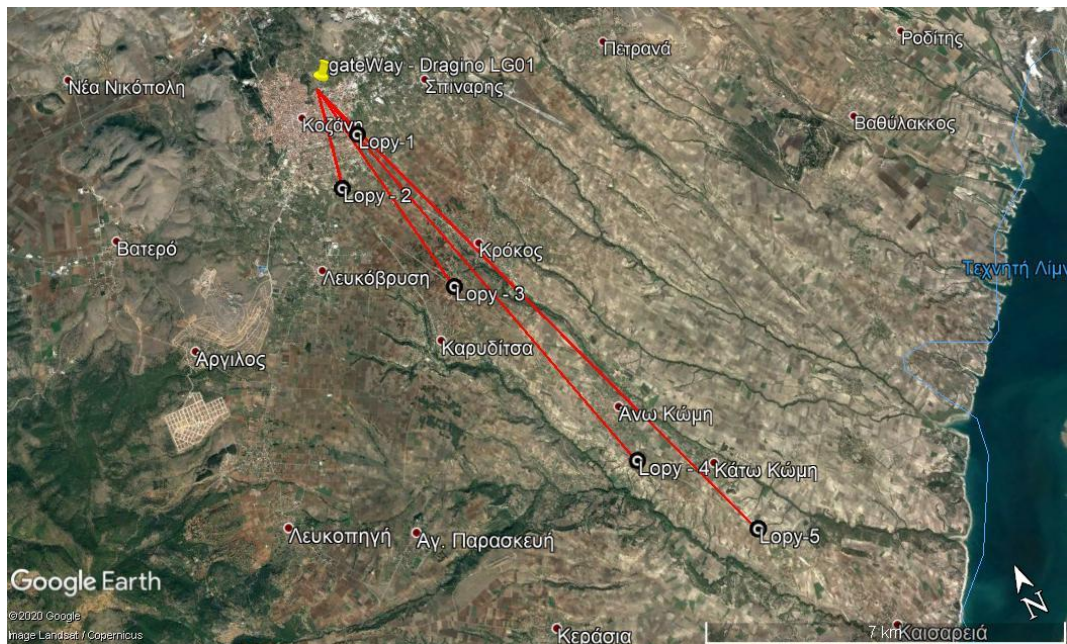
Η διαμόρφωση LoRaWAN υποστηρίζει επτά τιμές για την παράμετρο SF (SF6-SF12). Η παράμετρος SF6 που έχει το μεγαλύτερο ρυθμό μετάδοσης, απαιτεί μεγαλύτερο εύρος signal-to-noise ratio (SNR) για επιτυχή διαμόρφωση, σε σχέση με την παράμετρο SF12 που υποστηρίζει χαμηλότερο ρυθμό μετάδοσης αλλά, απαιτεί και ελάχιστο SNR για ίδια ισχύ μετάδοσης δεδομένων. Ένα Chirp αποτελείται από  $2^{SF}$  RF chips που μεταφέρουν SF στον αριθμό bits. Ο αριθμός των chips που διαμορφώνουν ένα ολοκληρωμένο chirp εξαρτάται από την παράμετρο BW (bandwidth) και υπολογίζεται σαν ένα chip ανά Hz του BW. Έτσι, ένα standard LoRa chirp που απλώνεται σε ένα BW 125KHZ αποτελείται από 125000 chips/s. Ο χρόνος μετάδοσης ενός τέτοιου chirp δίνεται από τη σχέση:

$$T_{sym} = \frac{2^{SF}}{BW}$$

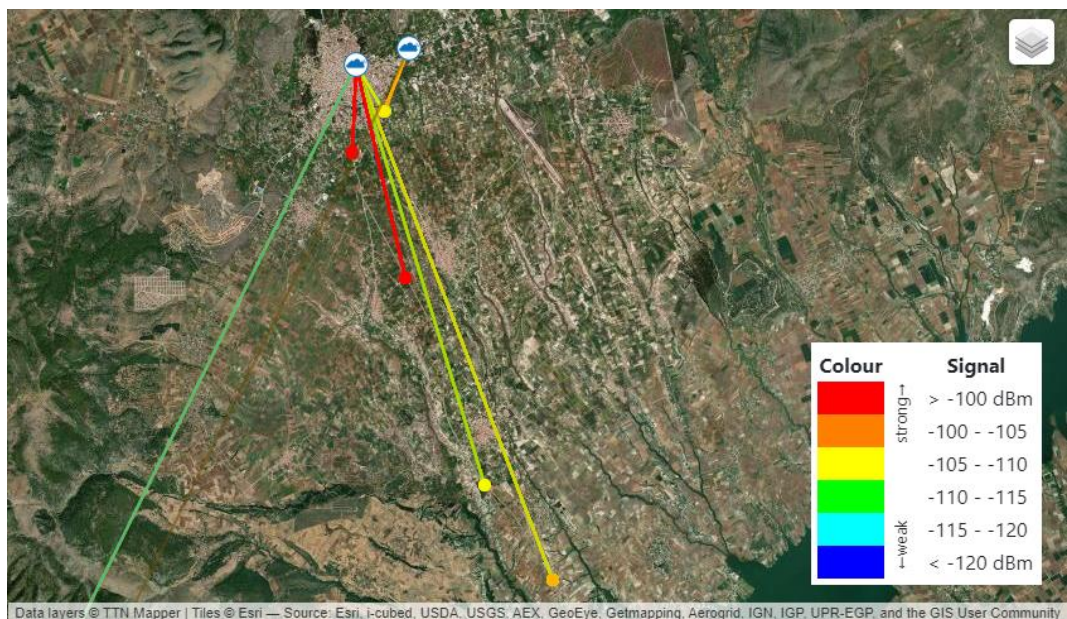
Εκτός από τον έλεγχο του ρυθμού μετάδοσης, η παράμετρος SF μας δίνει μοναδικά διαφορετικές κλίσεις chirp με αποτέλεσμα να επιτρέπουν την διαμόρφωση και από-διαμόρφωση πολλαπλών σειρών δεδομένων που προέρχονται από διαφορετικές συσκευές που έχουν πρόσβαση στο ίδιο κανάλι.

### 5.1.2 1<sup>ο</sup> πείραμα

Από σταθερό σημείο γινόταν αποστολές 10 πακέτων ανά ζεύγος παραμέτρων δηλαδή 60 πακέτα ανά θέση σε αποστάσεις 1.3 km, 2.3 km, 5.2km, 10.2km, 12.8 km.



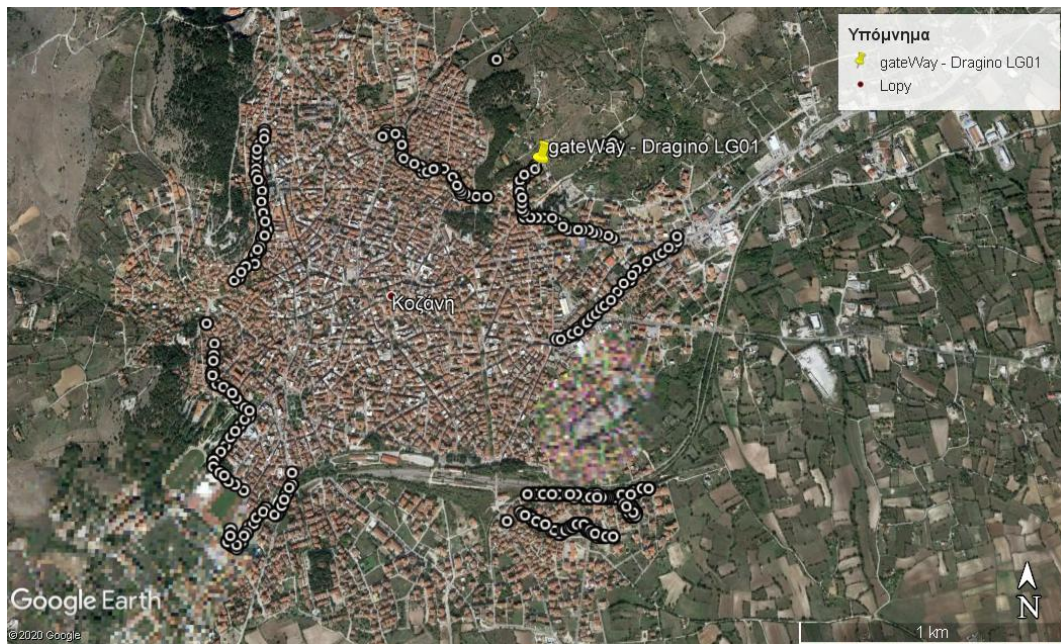
Εικόνα 62 – 1<sup>ο</sup> Πείραμα



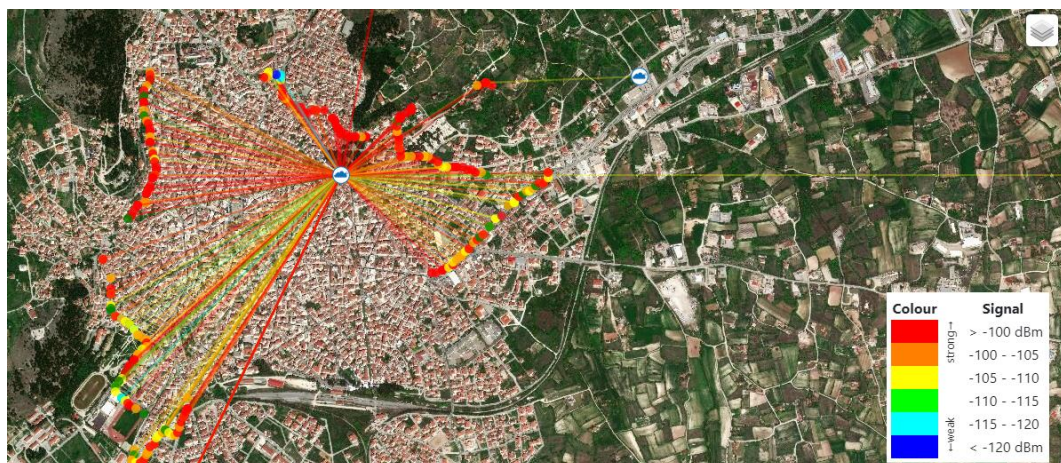
Εικόνα 63 -1<sup>ο</sup> Πείραμα ποιότητα σήματος

### 5.1.3 2<sup>ο</sup> Πείραμα

Η αποστολή πακέτων γινόταν συνεχόμενα με εναλλαγή των ζευγών (SF,Bitrate) παραμετροποίησης κατά τη διάρκεια βαδίσματος σε απομακρυσμένα αλλά εντός πόλης σημεία με ελάχιστη απόσταση τα 460 m και μέγιστη τα 1.8 km.



Εικόνα 64 – 2<sup>ο</sup> Πείραμα – βάδισμα εντός πόλης

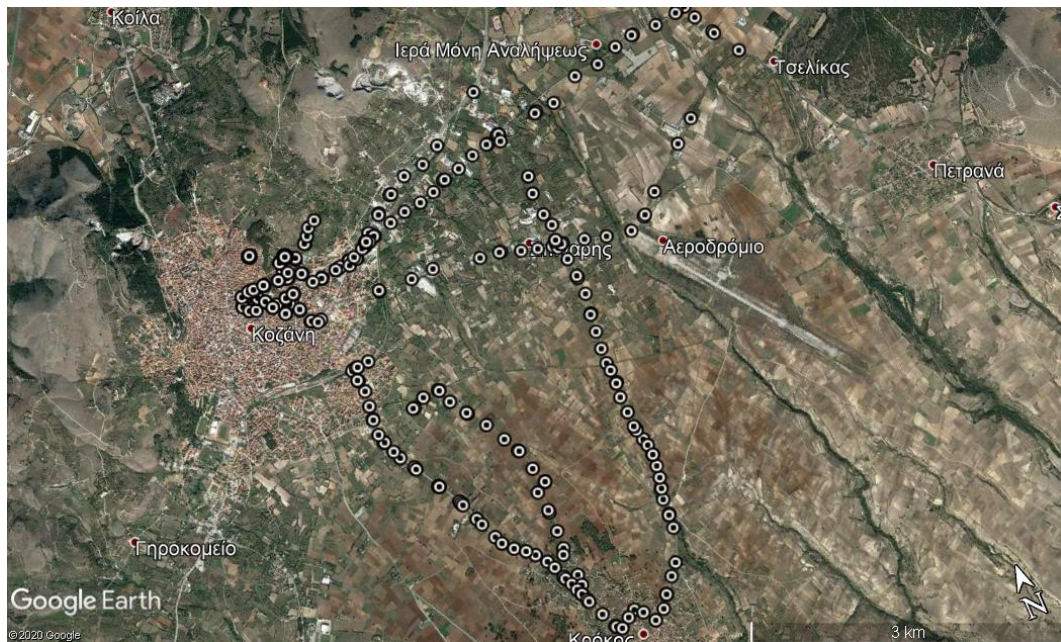


Εικόνα 65 – Ποιότητα σήματος 2<sup>ου</sup> πειράματος

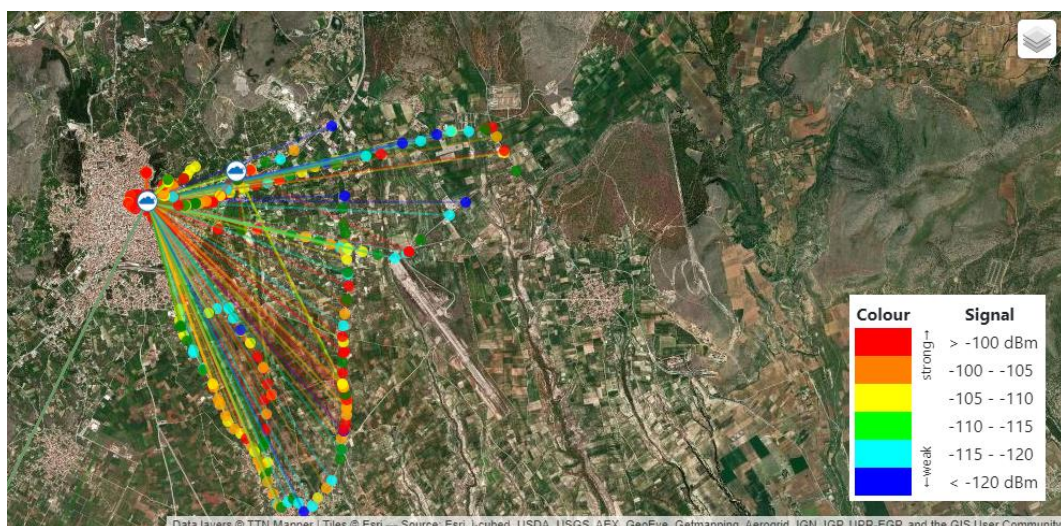
#### 5.1.4 3<sup>ο</sup> πείραμα

Πραγματοποιήθηκε με τον ίδιο τρόπο αποστολής πακέτων (συνεχής εναλλαγές σε ζεύγη παραμέτρων) με κίνηση αυτοκινήτου (<60km/h) και κυρίως σε μη αστικό περιβάλλον.

Ελάχιστη απόσταση 50m και μέγιστη απόσταση 4.8 km.



Εικόνα 66 - 3ο Πείραμα – Κίνηση με αυτοκίνητο



Εικόνα 67 – Ποιότητα σήματος 3<sup>ου</sup> Πειράματος

## 5.2 Αποτελέσματα

Βασικό ρόλο στο IoT (Internet of Things) παίζει η μετάδοση δεδομένων με ικανοποιητικό ρυθμό σε μεγάλη απόσταση. Παρακάτω περιγράφεται η επίδραση των βασικών παραμέτρων στη μετάδοση δεδομένων σε απόσταση.

### 5.2.1 Spreading Factor (SF) .

Η τιμή της παραμέτρου SF με απλά λόγια μας δίνει τον αριθμό των bits που μεταδίδονται σ' ένα απλό chirp. Παράδειγμα αν η παράμετρος οριστεί σε SF7 τότε αυτό σημαίνει ότι κάθε chirp αναπαριστά εφτά bits.

### 5.2.1.1 Air Time

Ο χρόνος μετάδοσης των πακέτων εξαρτάται κυρίως από τις παραμέτρους SF και BW.

Ο θεωρητικός υπολογισμός [23] του χρόνου μετάδοσης ενός πακέτου με payload size 25 bytes και overhead size 13 bytes φαίνεται στον πίνακα που ακολουθεί:

EU863-870 uplink and downlink

overhead size<sup>①</sup>

payload size<sup>②</sup>

share<sup>③</sup>

-13

-27

|                    | DR6 <sup>①</sup>                                       | DR5                                                    | DR4                                                   | DR3                                                   | DR2                                                    | DR1 <sup>①</sup>                                       | DR0                                                    |
|--------------------|--------------------------------------------------------|--------------------------------------------------------|-------------------------------------------------------|-------------------------------------------------------|--------------------------------------------------------|--------------------------------------------------------|--------------------------------------------------------|
| data rate          | <b>SF7</b> <sup>BW 250</sup>                           | <b>SF7</b> <sup>BW 125</sup>                           | <b>SF8</b> <sup>BW 125</sup>                          | <b>SF9</b> <sup>BW 125</sup>                          | <b>SF10</b> <sup>BW 125</sup>                          | <b>SF11</b> <sup>BW 125</sup>                          | <b>SF12</b> <sup>BW 125</sup>                          |
| airtime            | 41.1 ms                                                | 82.2 ms                                                | 154.1 ms                                              | 287.7 ms                                              | 534.5 ms                                               | 1,069.1 ms                                             | 1,974.3 ms                                             |
| 1% max duty cycle  | 4.1 sec<br>876 msg/hour                                | 8.2 sec<br>438 msg/hour                                | 15.4 sec<br>233 msg/hour                              | 28.8 sec<br>125 msg/hour                              | 53.5 sec<br>67 msg/hour                                | 106.9 sec<br>33 msg/hour                               | 197.4 sec<br>18 msg/hour                               |
| fair access policy | 118.3 sec (avg)<br><b>30.4</b> avg/hour<br>730 msg/24h | 236.7 sec (avg)<br><b>15.2</b> avg/hour<br>365 msg/24h | 443.8 sec (avg)<br><b>8.1</b> avg/hour<br>194 msg/24h | 828.7 sec (avg)<br><b>4.3</b> avg/hour<br>104 msg/24h | 1,539.4 sec (avg)<br><b>2.3</b> avg/hour<br>56 msg/24h | 3,078.9 sec (avg)<br><b>1.2</b> avg/hour<br>28 msg/24h | 5,685.9 sec (avg)<br><b>0.6</b> avg/hour<br>15 msg/24h |

Εικόνα 68 – AirTime

Στα πειράματά μας είχαμε ακριβώς τους ίδιους χρόνους μετάδοσης για ένα πακέτο με payload size 25 bytes και overhead size 13 bytes.

| SF | DR | airtime  |
|----|----|----------|
| 7  | 5  | 82,176   |
| 8  | 4  | 154,112  |
| 9  | 3  | 287,744  |
| 10 | 2  | 534,528  |
| 11 | 1  | 1069,056 |
| 12 | 0  | 1974,272 |

Εικόνα 69 - Πραγματικές μετρήσεις

Διαπιστώσαμε λοιπόν στην πράξη ότι, η παράμετρος SF με αντίστοιχο DR (data rate) μπορεί να προσδιορίσει τον χρόνο μετάδοσης ενός πακέτου. Συνεπώς θα μπορούσε να μην καταγράφεται στα πειράματά μας αφού μπορεί να υπολογιστεί.

### 5.2.2 Received Signal Strength Indication (RSSI)

Η παράμετρος RSSI είναι η προσλαμβανόμενη ισχύς σήματος σε milliwatts και μετριέται σε dBm. Αυτή η παράμετρος μπορεί να χρησιμοποιηθεί για τη μέτρηση του κατά πόσο ο δέκτης «ακούει» ένα σήμα από έναν αποστολέα. Μετριέται σε dBm και είναι αρνητική τιμή. Όσο πιο κοντά στο 0 τόσο καλύτερο σήμα έχουμε. Τυπικά οι τιμές RSSI για το LORA είναι :

RSSI minimum = -120 dBm.

If RSSI=-30dBm ισχυρό σήμα

If RSSI=-120dBm: ασθενές σήμα.

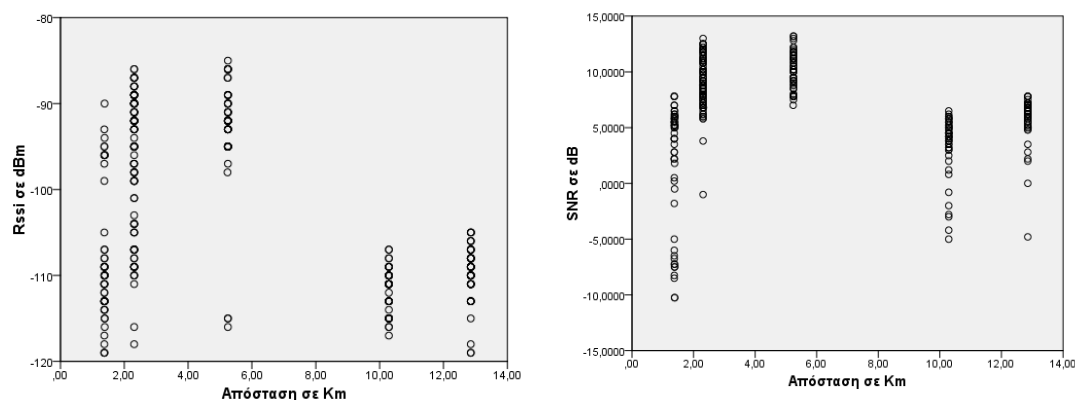
### 5.2.3 Signal-to-Noise Ratio (SNR)

Το SNR είναι ο λόγος μεταξύ της ισχύος του λαμβανόμενου σήματος και του επιπέδου ισχύος του θορύβου. Το όριο θορύβου είναι μια περιοχή όλων των ανεπιθύμητων παρεμβαλλόμενων πηγών σήματος που μπορεί να καταστρέψουν το μεταδιδόμενο σήμα και συνεπώς στη συνέχεια θα πραγματοποιηθούν εκ νέου μεταδόσεις. Όταν το SNR είναι θετικό, τότε το προσλαμβανόμενο σήμα λειτουργεί πάνω από το επίπεδο θορύβου. Όταν το SNR είναι αρνητικό, τότε το προσλαμβανόμενο σήμα λειτουργεί κάτω από το επίπεδο θορύβου. Τυπικά οι τιμές για την παράμετρο SNR στο LORA είναι: μεταξύ -20dB και +10dB. Μια τιμή SNR +10dB σημαίνει ότι το προσλαμβανόμενο σήμα είναι κατεστραμμένο. Το LORA τυπικά αποκωδικοποιεί σήματα που μπορεί να είναι μεταξύ -7.5dB και -20 dB.

### 5.2.4 Παράγοντας απόσταση.

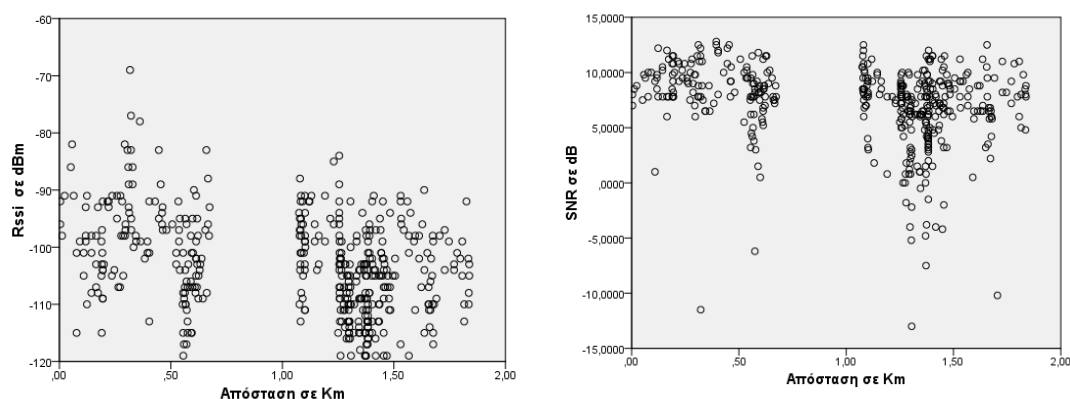
Λαμβάνοντας υπόψη μόνο τη χιλιομετρική απόσταση σε στάση και την απόσταση με κίνηση πήραμε τα παρακάτω αποτελέσματα.

Πίνακας 2 - Παράγοντας Απόσταση – Μη αστικό περιβάλλον



Το ότι δεν έχουμε ενδιάμεσες τιμές οφείλεται στο ότι δεν έγιναν μετρήσεις σε θέσεις με ενδιάμεσες αποστάσεις. Παρατηρούμε ότι οι τιμές για το RSSI έχουν ένα εύρος από -120 μέχρι και -80 ενώ σε μεγάλες αποστάσεις η παράμετρος RSSI βρίσκεται μεταξύ του -120 και -115 dBm. Αντίστοιχα συμπεριφέρεται και η παράμετρος SNR. Σε μικρές αποστάσεις έχουμε ένα μεγάλο εύρος -10 μέχρι και 15, ενώ σε μεγάλες αποστάσεις το εύρος περιορίζεται από -5 μέχρι 8 dB.

Πίνακας 3 - Παράγοντας Απόσταση - Αστικό Περιβάλλον



Παρατηρούμε ότι σε αστικό περιβάλλον παρόλο που έχουμε μικρότερες αποστάσεις η ποιότητα του σήματος έχει μεγαλύτερη διασπορά. Η παράμετρος RSSI βρίσκεται στα ίδια επίπεδα με το 1<sup>ο</sup> πείραμα -120 με -80 dBm ενώ η παράμετρος SNR βρίσκεται μεταξύ -5 και 15 dB.

Με βάση τις παραπάνω παρατηρήσεις συμπεραίνουμε, ότι σε αστικό περιβάλλον έχουμε παραπάνω παρεμβολές από παράγοντες, που μπορεί να είναι ή ύπαρξη ηλεκτρομαγνητικής ενέργειας που οφείλεται σε άλλα δίκτυα ή το ότι η συσκευή μετάδοσης βρίσκεται σε κίνηση.

Σίγουρα όμως η απώλεια πακέτων τελικά είναι μικρή, σε σημείο που μπορεί να είναι μηδενική ανάλογα με την περίπτωση χρήσης που θα χρησιμοποιηθεί ένα τέτοιο δίκτυο.

### 5.2.5 Παράγοντας SF

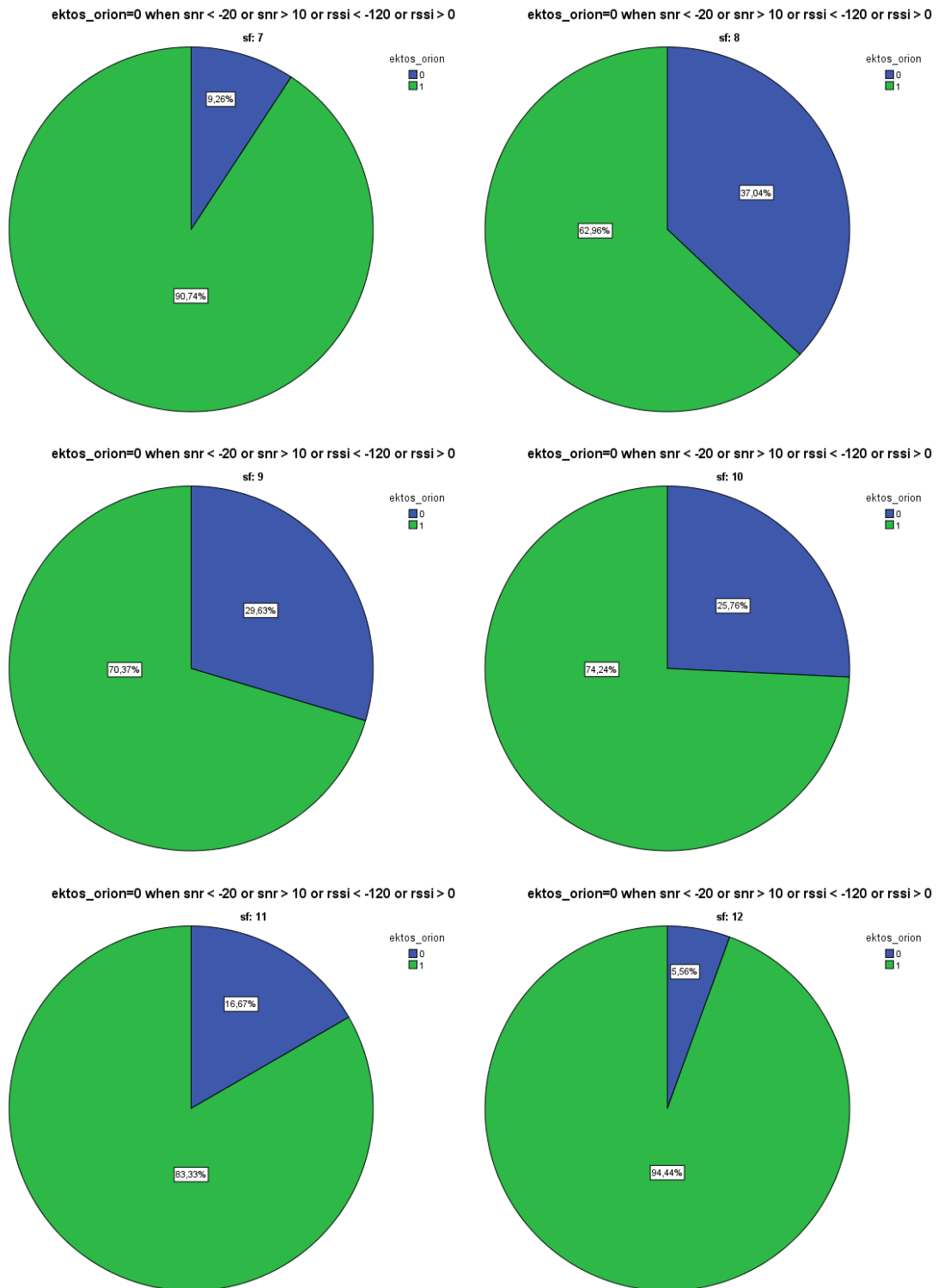
Αν θεωρήσουμε σαν ασθενές σήμα ένα σήμα που είναι:

είτε εκτός ορίων η παράμετρος RSSI < -120 ή RSSI > 0

είτε εκτός ορίων η παράμετρος SNR < -20 ή SNR > 10

τότε, λαμβάνοντας υπόψη τη βασική παράμετρο SF, έχουμε την παρακάτω κατανομή όσον αφορά τα πιθανά σφάλματα με δεδομένη παραμετροποίηση της gateway.

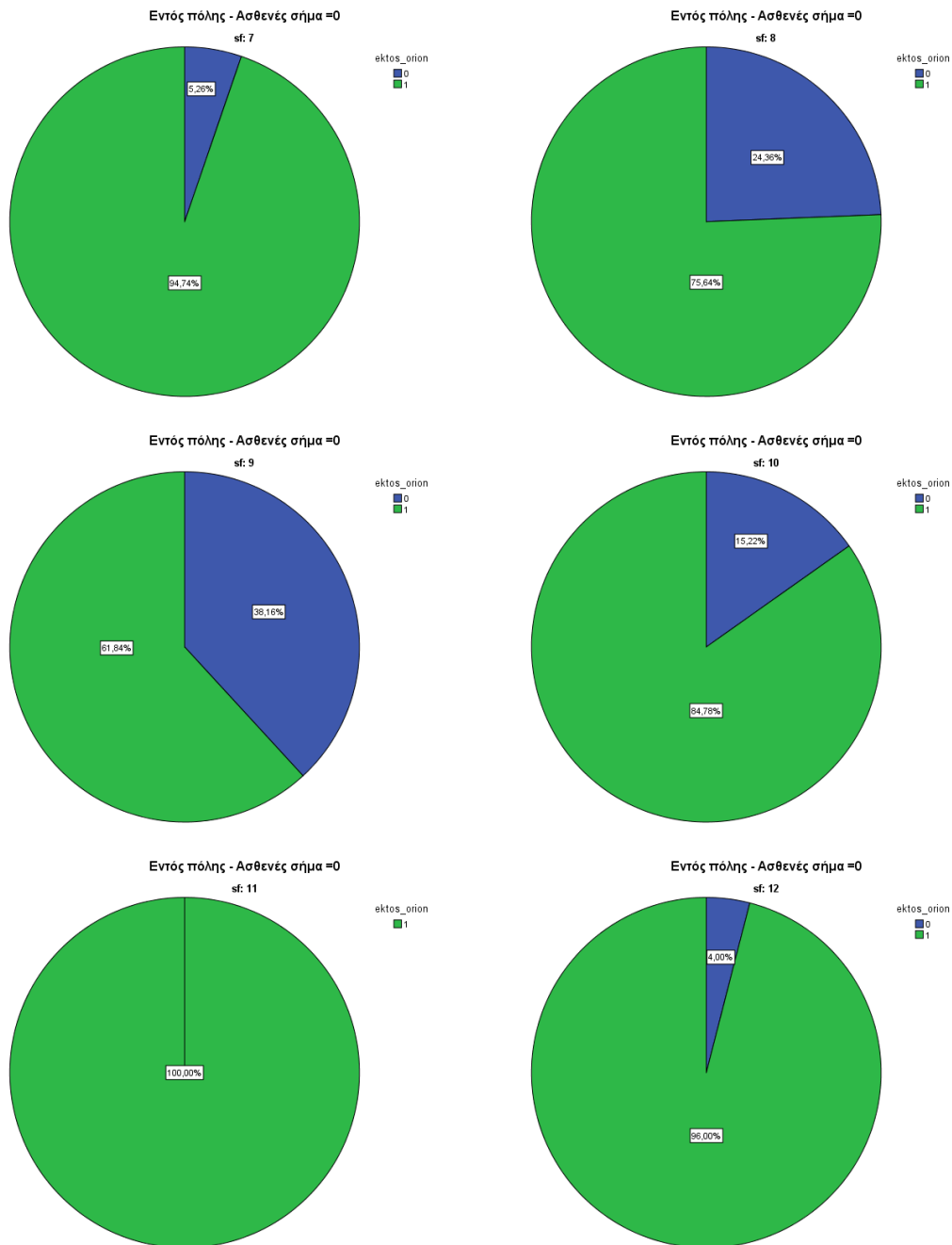
Πίνακας 4 - Σφάλματα σε μη αστικό Περιβάλλον



Παρατηρούμε ότι στο 1<sup>ο</sup> πείραμα υπάρχει ζεύγος παραμέτρων SF=7 , SF=12 στο οποίο οι απώλειες πακέτων με αυστηρό τρόπο περιορίζονται περίπου στο 5%.

Στα επόμενα διαγράμματα βλέπουμε την απώλεια πακέτων με τις ίδιες προϋποθέσεις του 1<sup>ου</sup> πειράματος για αστικό περιβάλλον.

Πίνακας 5 - Σφάλματα σε αστικό περιβάλλον



Στο 2<sup>ο</sup> πείραμα βλέπουμε ότι υπάρχει παραμετροποίηση SF=11 που έχουμε μηδενικές απώλειες πακέτων.

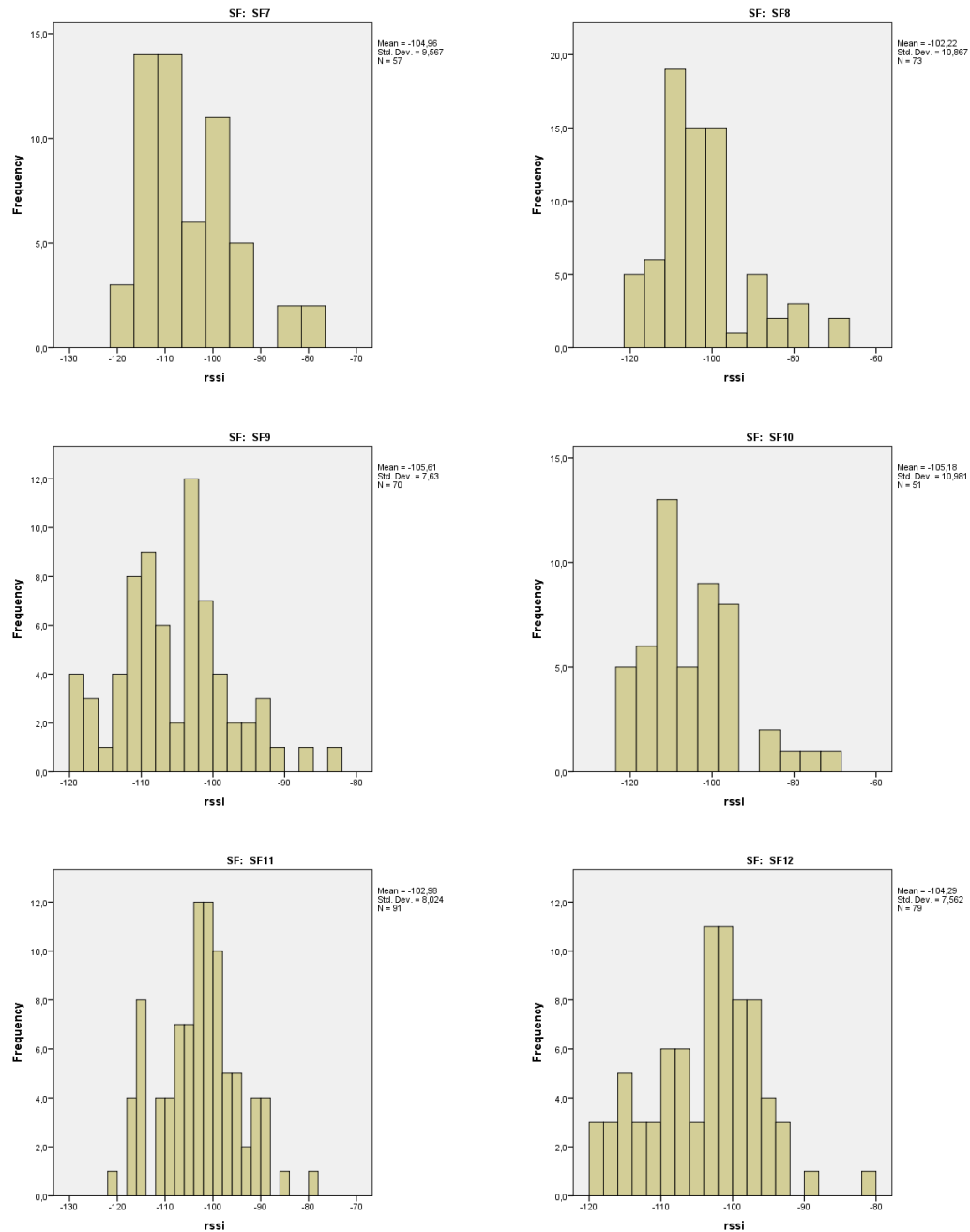
## 5.2.6 Παράγοντας κίνηση

### 5.2.6.1 Κατανομή μετρήσεων RSSI και SNR σε κίνηση

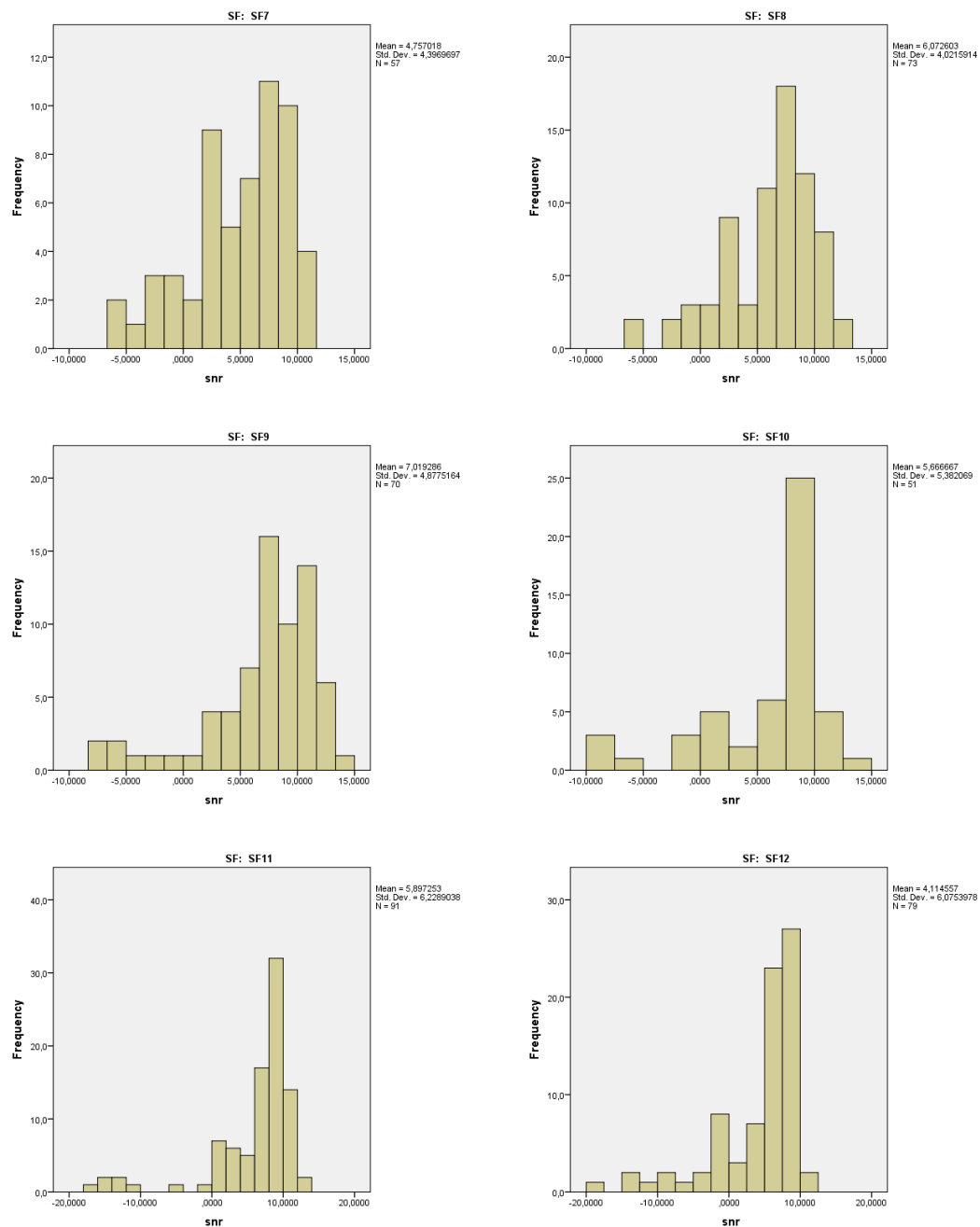
Στο τρίτο πείραμα που είχαμε αποστολή πακέτων σε συνεχή ροή με μεταβαλλόμενη κυκλικά παραμετροποίηση, όπως παρατηρούμε στα ιστογράμματα που ακολουθούν, βλέπουμε ότι

υπάρχει μια μικρή μετατόπιση προς πιο ασθενές σήμα, χωρίς όμως να έχουμε σοβαρή απώλεια πακέτων.

Πίνακας 6 – Κατανομή RSSI και SNR σε κίνηση



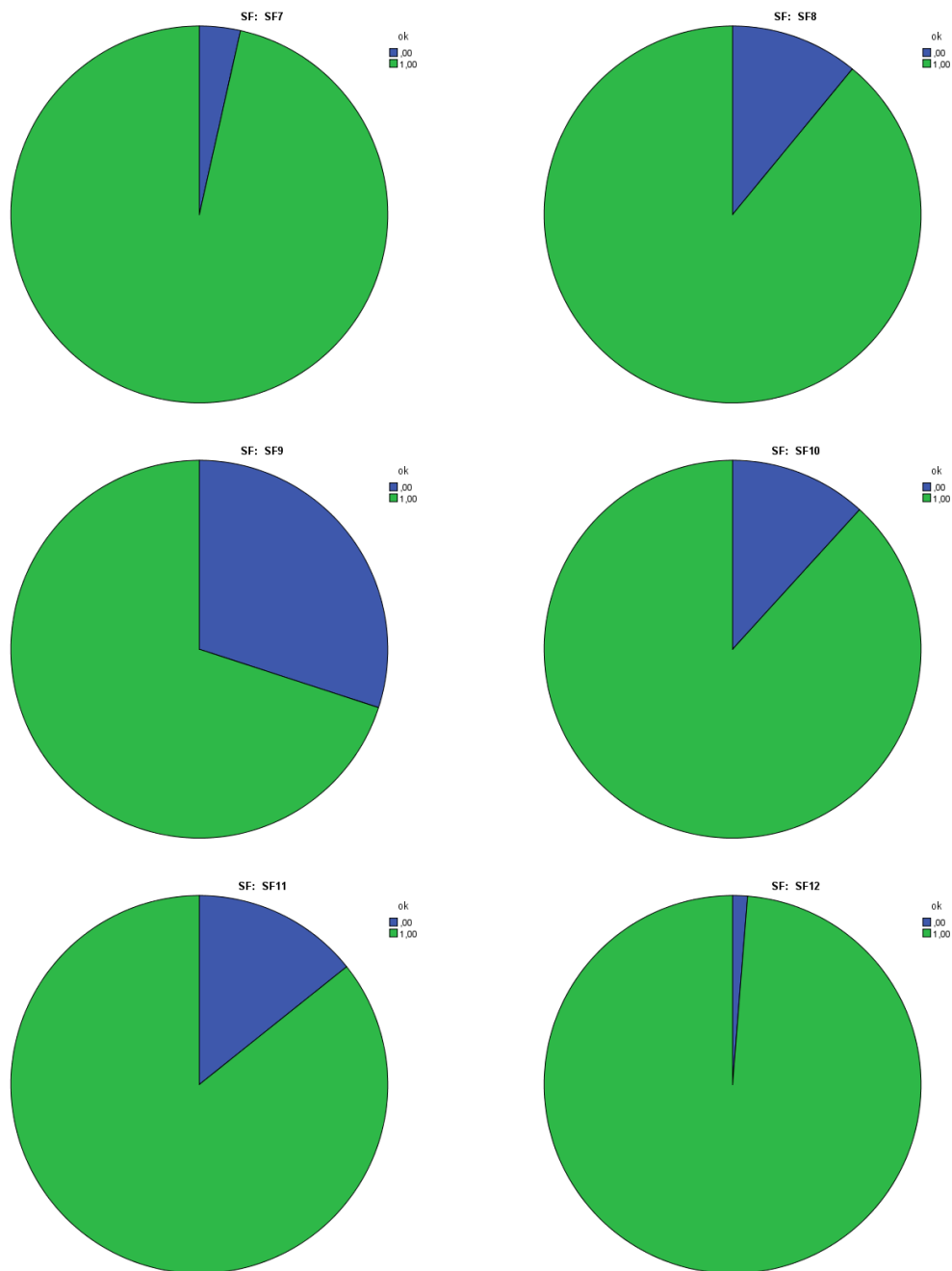
Πίνακας 7 – Κατανομή RSSI και SNR σε κίνηση μικτό περιβάλλον



Βλέπουμε ότι και σε αυτή την περίπτωση υπάρχει ζεύγος παραμέτρων SF11 για το οποίο έχουμε την καλύτερη ποιότητα σήματος.

Όπως παρατηρούμε στα διαγράμματα που ακολουθούν για SF7 και SF12 η απώλεια πακέτων μπορεί να είναι κάτω από 2%.

Πίνακας 8 – Απώλεια πακέτων σε κίνηση



## 5.3 Παράγοντας αριθμός end-devices

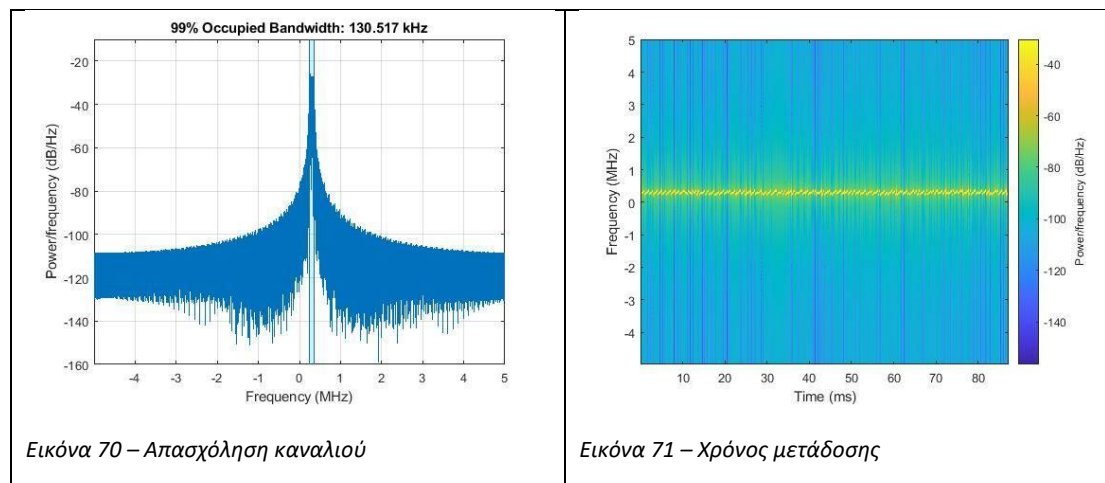
### 5.3.1 Προσομοίωση εξυπηρέτησης πολλών end-devices

Με δεδομένο ότι δεν μπορούσαμε να έχουμε ταυτόχρονα πολλές φυσικές συσκευές επιλέξαμε την προσομοίωση τους με χρήση λογισμικού.

Με βάση τις προηγούμενες μετρήσεις που αφορούσαν την βέλτιστη παραμετροποίηση με συμμετοχή μιας συσκευής end-device, έγινε μια προσομοίωση ανταπόκρισης του δικτύου σε μεταβλητό αριθμό συσκευών που υλοποιήθηκε με αλγόριθμο σε MatLab.

Οι δοκιμές έγιναν για μια ισχύ εξόδου POWER=14 dB, βασική συχνότητα F=868.3, BW=125KHZ. Σε πρώτη φάση συγκρίναμε τα πειραματικά αποτελέσματα με τα πραγματικά ώστε να έχουμε μια εκτίμηση αν η προσομοίωση πλησιάζει την πραγματικότητα.

Έτσι για SF7 είχαμε τα παρακάτω αποτελέσματα:



Αφού πετύχαμε μια κοντινή κατάσταση όσον αφορά το χρόνο μετάδοσης, το RSSI και το SNR, έγινε επιβάρυνση του συστήματος με εικονικές συσκευές. Ο σκοπός μας ήταν να δούμε την ανταπόκρισή του συστήματος με βάση τις μετρήσεις για το χρόνο απασχόλησης του καναλιού από τα πειράματα που κάναμε.

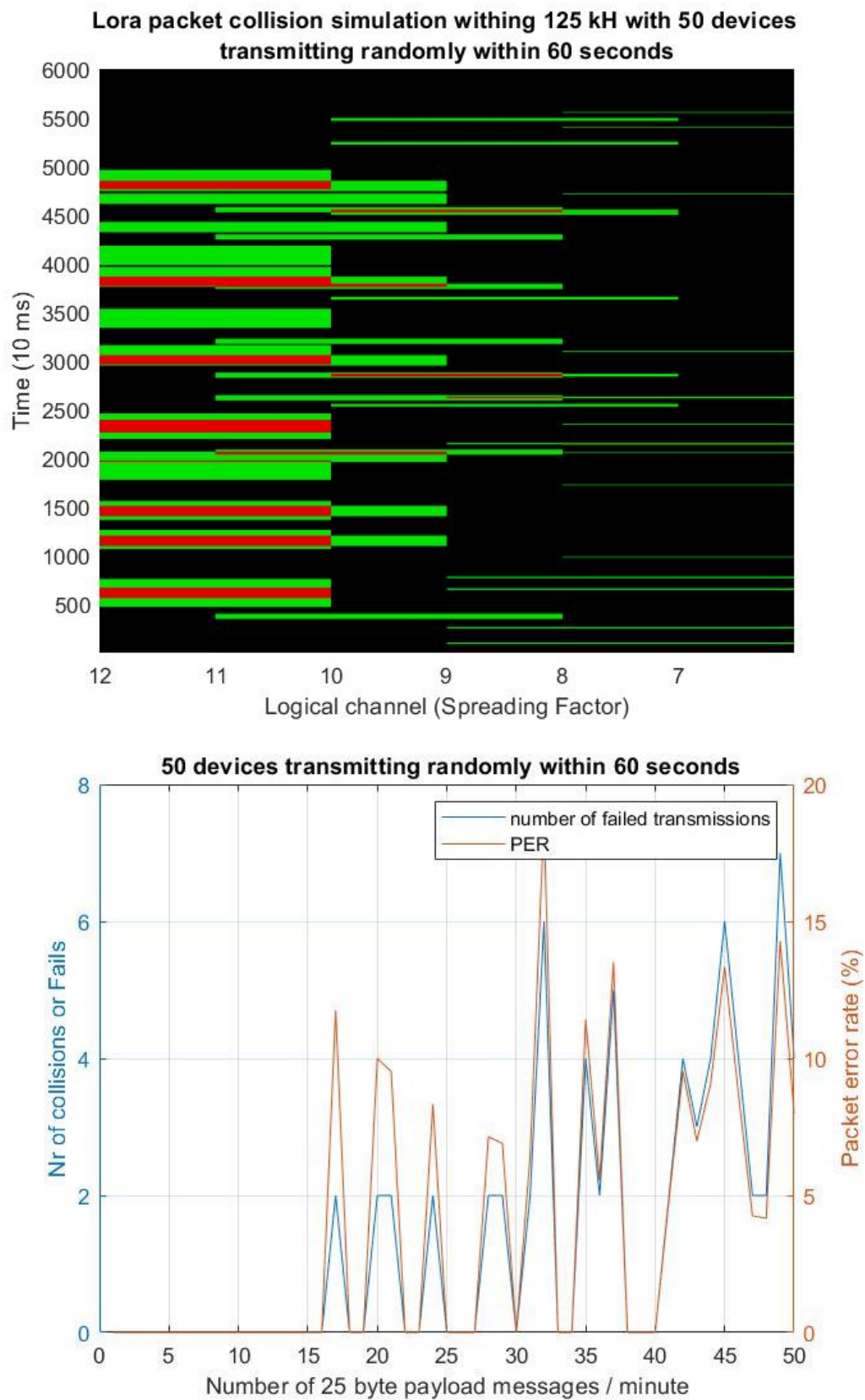
Συγκεκριμένα χρησιμοποιήθηκαν οι παρακάτω παράμετροι:

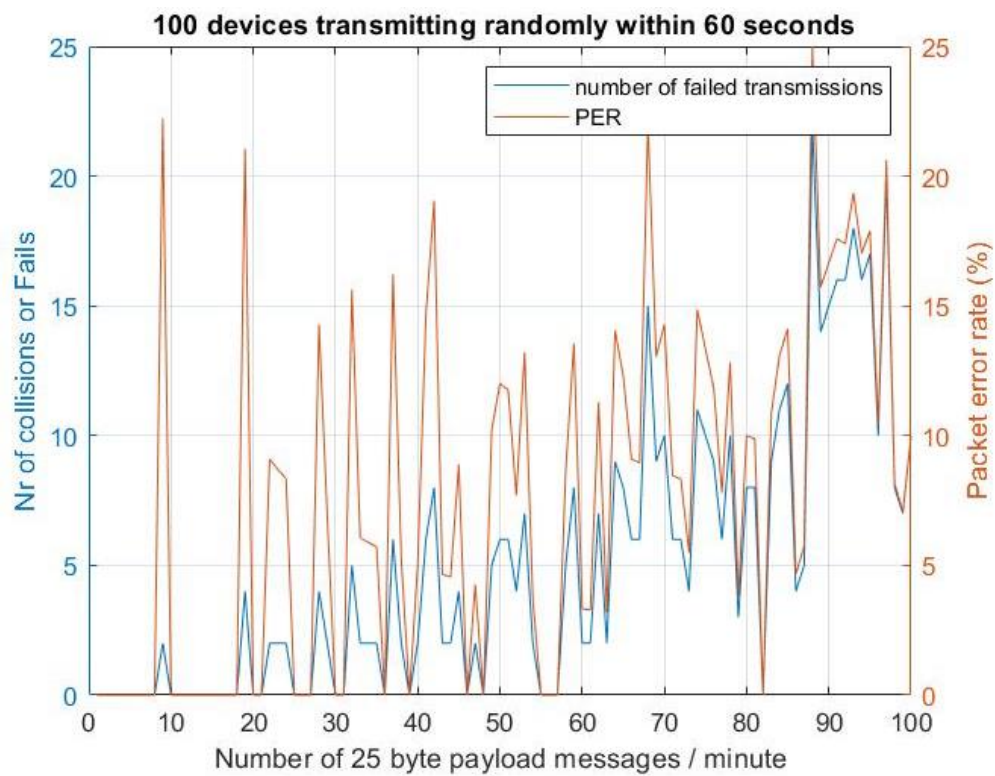
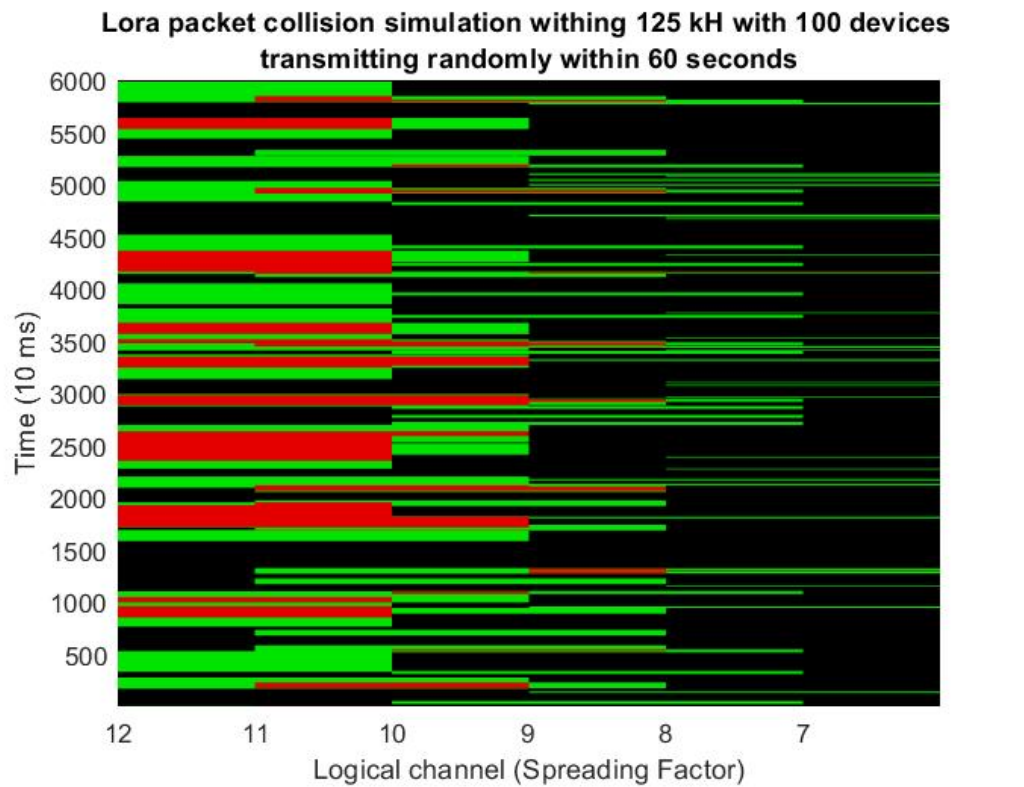
Πίνακας 9 – Ρυθμός μετάδοσης με βάση την παράμετρο SF

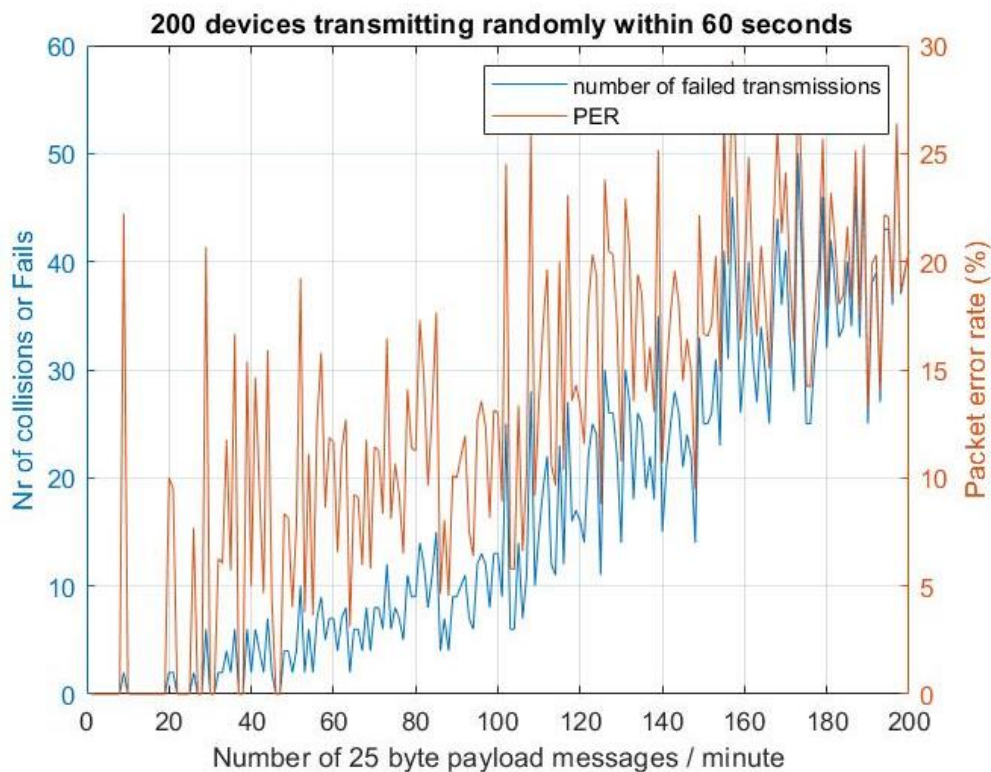
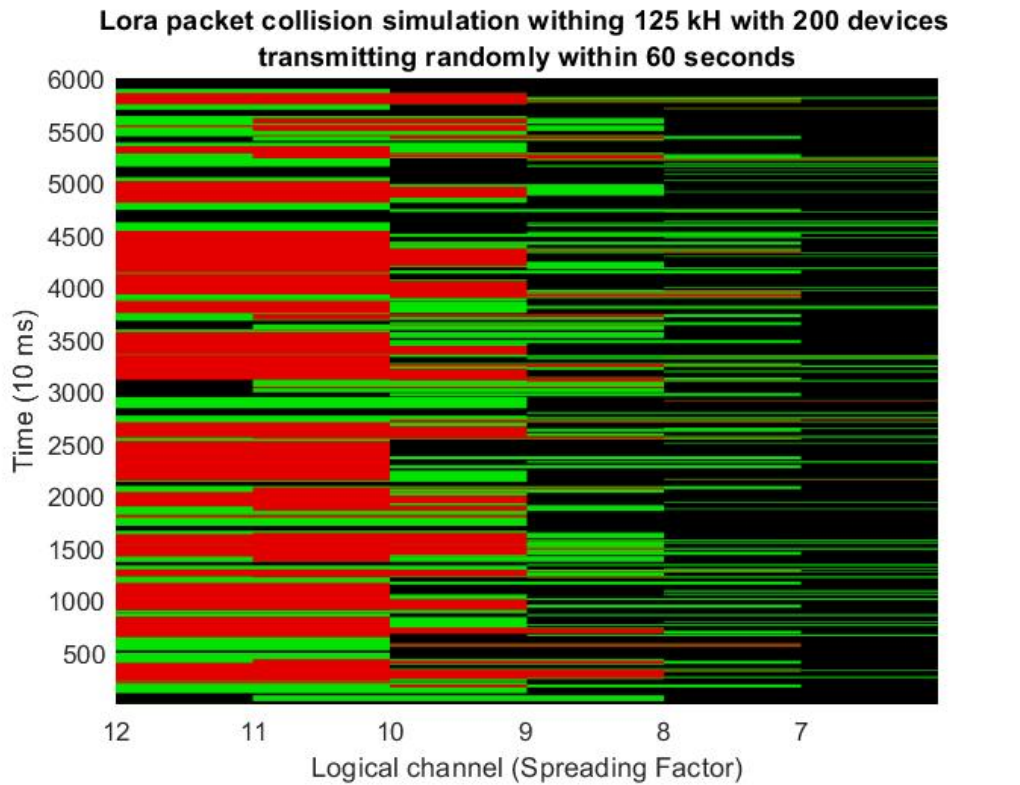
| SF | Bit rate (bit/sec) | Duration time (ms) |
|----|--------------------|--------------------|
| 12 | 293                | 1974.3             |
| 11 | 976                | 1069.1             |
| 10 | 1757               | 493.6              |
| 09 | 3125               | 267.3              |
| 08 | 5478               | 143.9              |
| 07 | 9375               | 82.2               |

Η πρώτη δοκιμή έγινε για 50 συσκευές που στέλνουν πακέτα μεγέθους όσο είναι το μέγεθος του πακέτου που στέλνει το pytrack-lorpy κάθε φορά, δηλαδή 25 bytes payload + 13 bytes

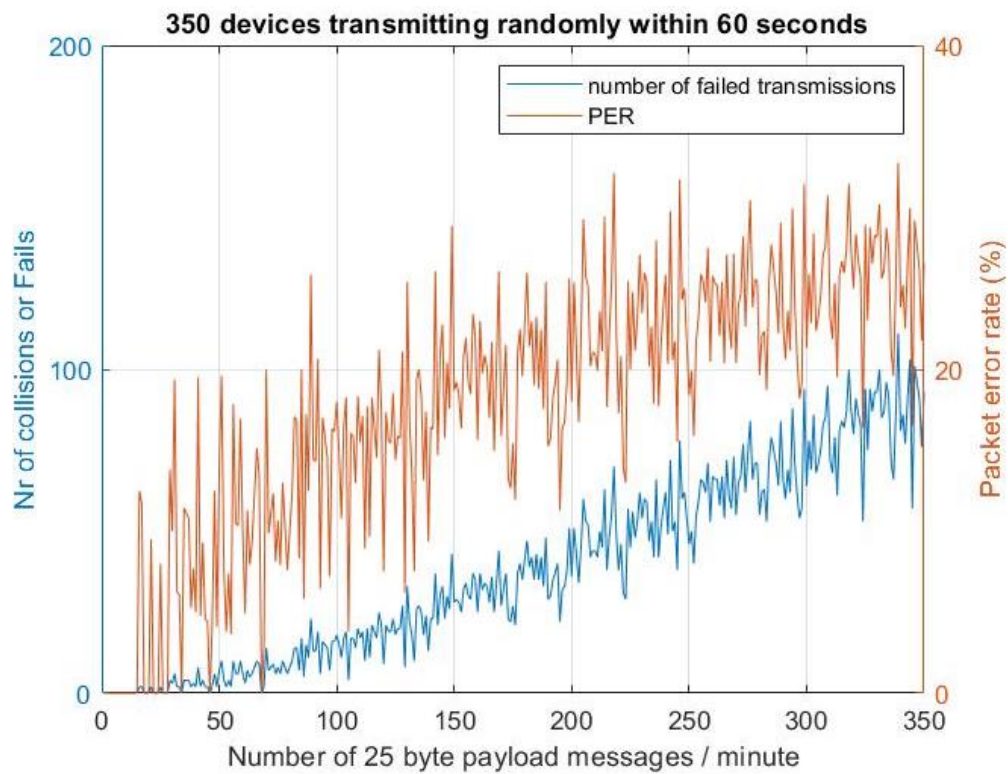
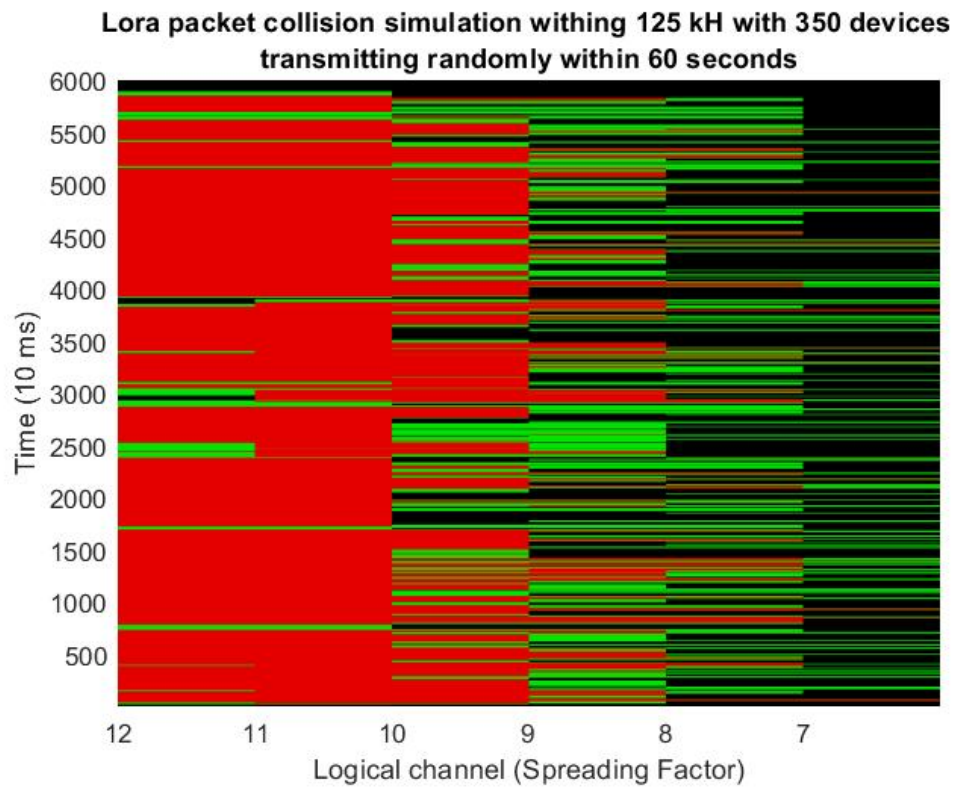
overhead . Χρησιμοποιούνται τυχαία 5 κανάλια με BW που έχει οριστεί στα 125 Khz.

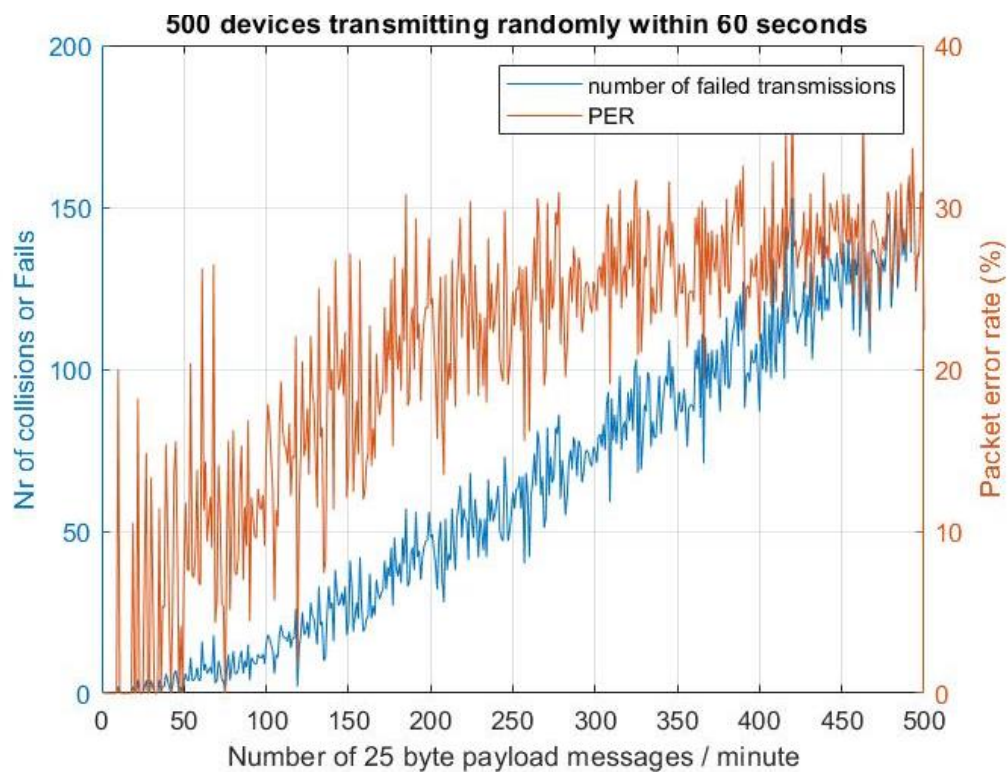
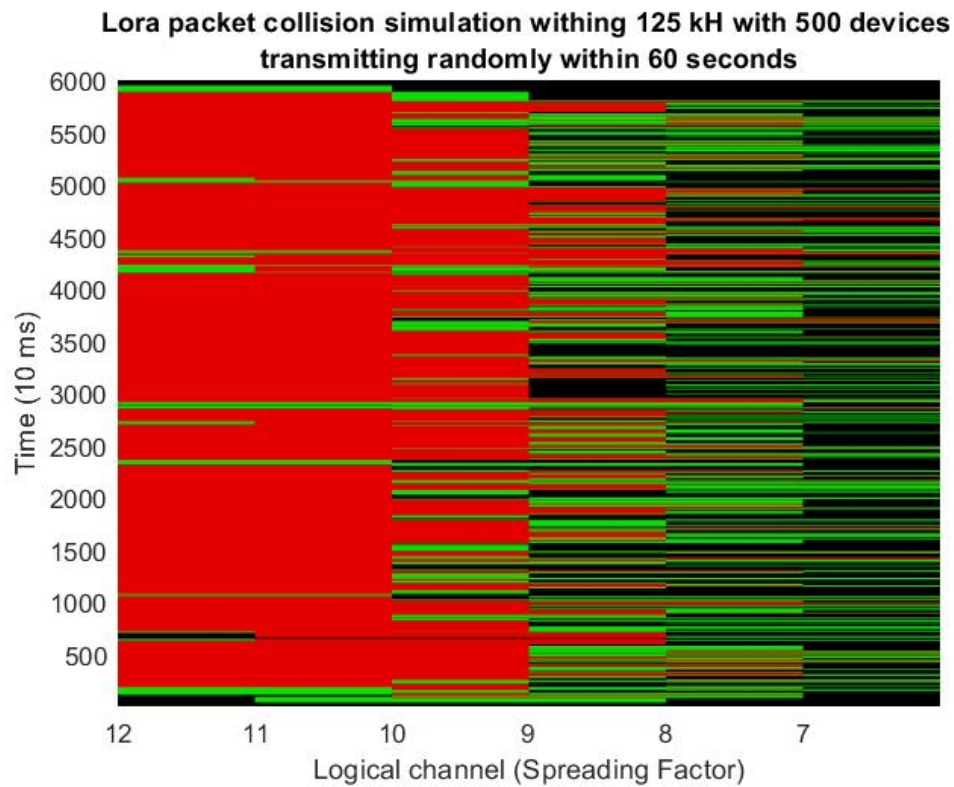


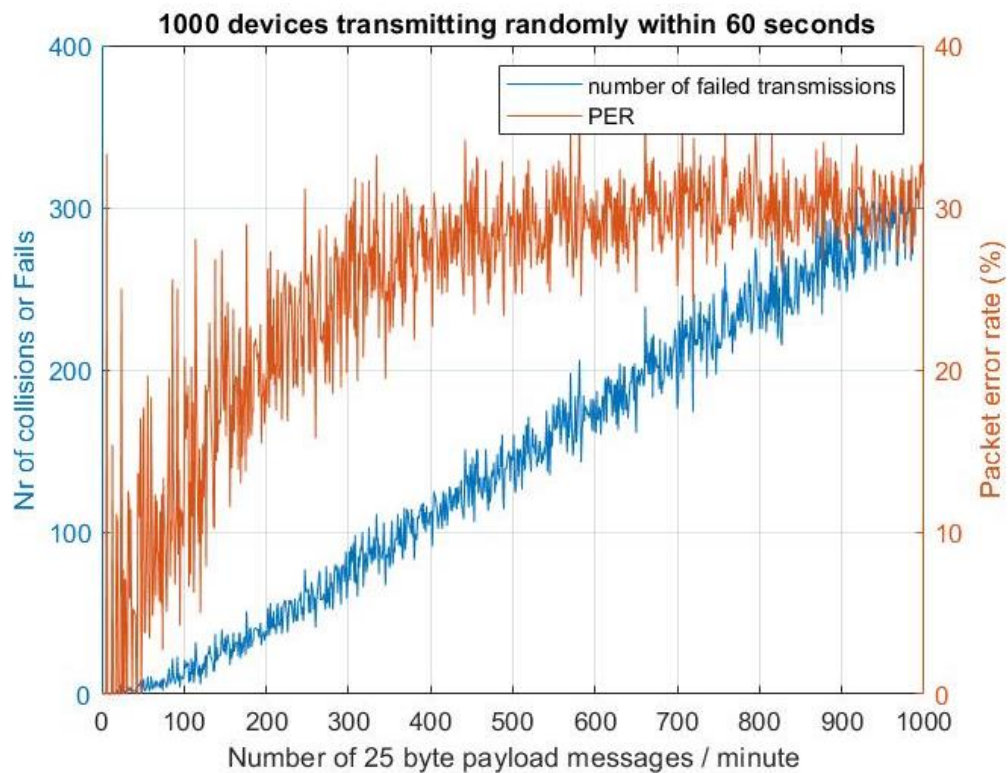
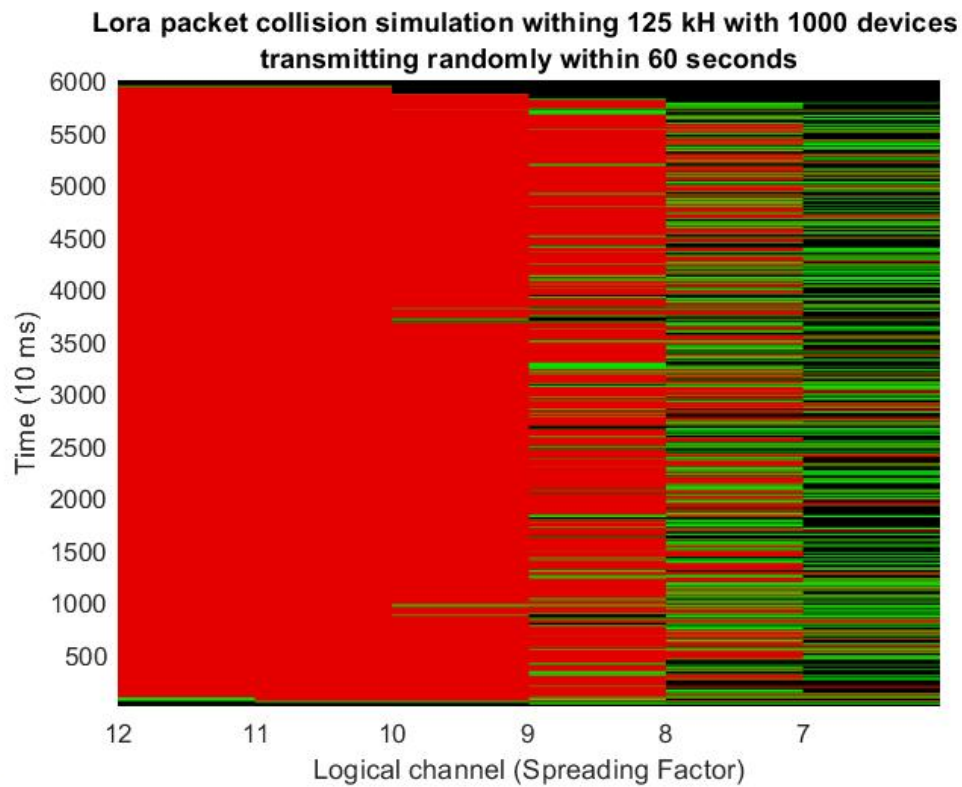




Παρατηρούμε ότι το σύστημά μας ανταποκρίνεται χωρίς συγκρούσεις για 50 συσκευές ακόμη και με SF12. Πρέπει να λάβουμε υπόψη μας ότι όσο μεγαλύτερη είναι η τιμή για την παράμετρο SF τόσο μικρότερο είναι το BR και κατά συνέπεια περιμένουμε περισσότερες συγκρούσεις πακέτων. Αυξάνοντας τις συσκευές που συμμετέχουν ταυτόχρονα, βλέπουμε ότι αρχίζουν να αυξάνονται οι συγκρούσεις και για SF7.







Στις προηγούμενες προσομοιώσεις υποθέτουμε ότι οι συσκευές που συμμετέχουν επιλέγουν τυχαία κάποια παράμετρο SF7 – SF12

## **5.4 Σύνοψη συμπερασμάτων αξιολόγησης**

### **5.5 Επιλογή Gateway**

Για τη σύνδεση της συσκευής έγινε χρήση Activation by Personalization (ABP). Δηλαδή η σύνδεση του end-device και της Gateway γινόταν με βάση την ταυτότητα(Device Address) της συσκευής στο TTN.

Το αποτέλεσμα ήταν να γίνεται επιλογή της gateway με βάση την ποιότητα σήματος. Από τις πρώτες μετρήσεις παρατηρήσαμε ότι κατά 93% των μεταδόσεων γινόταν επιλογή της 2<sup>ης</sup> gateway .

Όπως ήταν αναμενόμενο η 2<sup>η</sup> Gateway είχε πολύ πιο σταθερή απόδοση με τις λιγότερες απώλειες στην λήψη πακέτων. Με δεδομένη την απαίτηση αξιόπιστης μετάδοσης δεδομένων και τον αριθμό ταυτόχρονης εξυπηρέτησης πολλών end-devices θα λέγαμε ότι αξίζει να επενδυθούν χρήματα στην αγορά μιας αξιόλογης και ποιοτικής κατασκευαστικά gateway.

#### **5.5.1 Περίπτωση χρήσης που προκύπτει από το 1<sup>ο</sup> πείραμα.**

Επιλέγοντας σαν gateway τη δεύτερη (Kerlink) έχουμε τις εξής δυνατότητες:

Αν δεν μας ενδιαφέρει η ταχύτητα μετάδοσης παράδειγμα, μετρήσεις σε μια αγροτική καλλιέργεια με στόχο την εποπτεία και ενδεχόμενα τον έλεγχο αυτοματισμών, θα μπορούσαμε να προτείνουμε μια παραμετροποίηση με : χρήση SF12 , bitrate 293 bit/sec. Σ' αυτή την περίπτωση μπορούμε να έχουμε αξιόπιστη εξυπηρέτηση μέχρι και 50 end-devices ,ίσως και παραπάνω, αν λάβουμε υπόψη μας ότι δεν έχουμε απαίτηση μεγάλης ροής δεδομένων.

Τα end-devices μπορούν να είναι αισθητήρες θερμοκρασίας, υγρασίας, ποιότητας εδάφους, (ph, νιτρικά άλατα, κ.λ.π) και ενδεχόμενα αυτοματισμοί ελέγχου.

#### **5.5.2 Περίπτωση χρήσης που προκύπτει από το 2<sup>ο</sup> πείραμα.**

Παρατηρούμε ότι θα μπορούσαμε να κάνουμε χρήση παραμετροποίησης SF11 με bitrate 547 bit/sec ή SF7 με bitrate 5478 bit/sec. Σ' αυτή την περίπτωση θα έχουμε μια ταυτόχρονη εξυπηρέτηση 100-350 end-devices. Για την ταχύτερη μετάδοση δεδομένων θα προτιμούσαμε μια παραμετροποίηση με SF7. Μια περίπτωση χρήσης για ένα γρήγορο ρυθμό μετάδοσης που μας προσφέρει μια παραμετροποίηση με SF7 θα μπορούσε να είναι η εποπτεία ατόμων με ειδικά προβλήματα με τον εξής τρόπο. Οι μετρήσεις που αφορούν ζωτικές λειτουργίες ενός ανθρώπου , παράδειγμα ή πίεση, οι καρδιακοί παλμοί, η οξυγόνωση του αίματος, το ζάχαρο καθώς και η θέση του ατόμου που μας ενδιαφέρουν μπορούν μετά από αποστολή να

καταγράφονται από ένα κεντρικό σύστημα. Το σύστημα θα μπορεί με κατάλληλο λογισμικό να προειδοποιεί, να εντοπίζει, και να προλαμβάνει επικίνδυνες καταστάσεις.

### ***5.5.3 Περίπτωση χρήσης που προκύπτει από το 3<sup>ο</sup> πείραμα.***

Παρατηρούμε ότι θα μπορούσαμε να κάνουμε χρήση παραμετροποίησης SF12 με bitrate 293 bit/sec για την περίπτωση στην οποία δεν στέλνουμε μεγάλα πακέτα δεδομένων. Σ' αυτή την περίπτωση θα έχουμε μια ταυτόχρονη εξυπηρέτηση 50-100 end-devices.

Περίπτωση χρήσης που θα μπορούσαμε να προτείνουμε με βάση το 3<sup>ο</sup> πείραμα είναι για παράδειγμα εποπτεία συσκευών που είναι κατάστασης on – off, παράδειγμα κάδοι συλλογή ανακυκλώσιμων υλικών. Το σενάριο θα μπορούσε να είναι η καταγραφή της κατάστασης των κάδων και ο υπολογισμός της βέλτιστης διαδρομής για ένα συλλογέα του οποίου οι κινήσεις επίσης καταγράφονται.

Σε όλες τις παραπάνω περιπτώσεις η εκτίμηση της εξυπηρέτησης ταυτόχρονα πολλών συσκευών έγινε με βάση ένα χρονικό διάστημα δευτερολέπτων. Αν λάβουμε υπόψη ότι στην πραγματικότητα δεν απαιτούν όλες οι συσκευές ταυτόχρονα σύνδεση θα λέγαμε ότι η εξυπηρέτηση είναι πολύ μεγαλύτερου αριθμού συσκευών.

# 6

## Επίλογος

Μετά από μακρόχρονη ενασχόληση και δοκιμές με διαφορετικά συστήματα Hardware και Software προέκυψε η συνεργασία με το τμήμα ορυχείων της ΔΕΗ στην Κοζάνη. Έγινε μια πρόταση για υλοποίηση ενός δικτύου LoRa που αυτή τη στιγμή βρίσκεται στη φάση των δοκιμών.

### 6.1 Πρόταση για την περίπτωση «Ορυχεία ΔΕΗ»

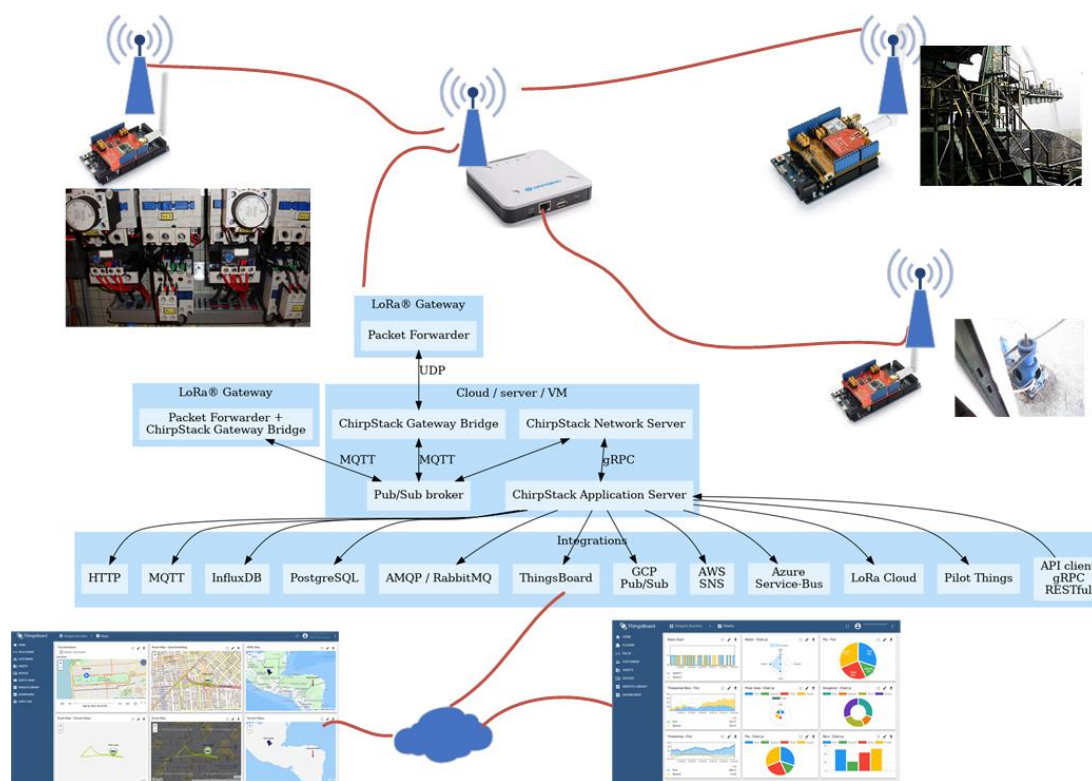
Στα ορυχεία της ΔΕΗ στην Πτολεμαΐδα υπήρξε ανάγκη της καταγραφής της κίνησης των καδοφόρων μηχανημάτων για την εξόρυξη του Λιγνίτη αλλά και της καταγραφής μετρήσεων που προέρχονται από διάφορα μηχανήματα τα οποία είναι του τύπου , Θερμοκρασία, στροφές, υγρασία, καταστάσεις on/off . Προτάθηκε λοιπόν ένα δίκτυο LoRaWAN το οποίο μπορεί να καλύπτει την έκταση των ορυχείων με το πλεονέκτημα ότι οι συσκευές καταγραφής δεν χρειάζεται να βρίσκονται σε ένα κλασικό δίκτυο το οποίο είναι απαιτητικό σε ενέργεια αλλά και πόρους δικτύου.



Εικόνα 72 – Κάλυψη

## 6.2 Πρόταση

Προτάθηκε και υλοποιήθηκε σε φάση δοκιμών το παρακάτω μοντέλο



Εικόνα 73 - Δομή δικτύου που στήθηκε στο Ορυχείο

### 6.2.1 Server

Λόγω ιδιωτικότητας των δεδομένων θα έπρεπε να στηθεί ένα ιδιωτικό δίκτυο LoRaWAN και μάλιστα σε πειραματικό στάδιο.

Από τη μεριά του Server καταλήξαμε να χρησιμοποιήσουμε την στοίβα (stack) Network, Application, Gateway, Bridge Server με το Chirpstack και Integration για οπτικοποίηση δεδομένων με το ThingsBoard.

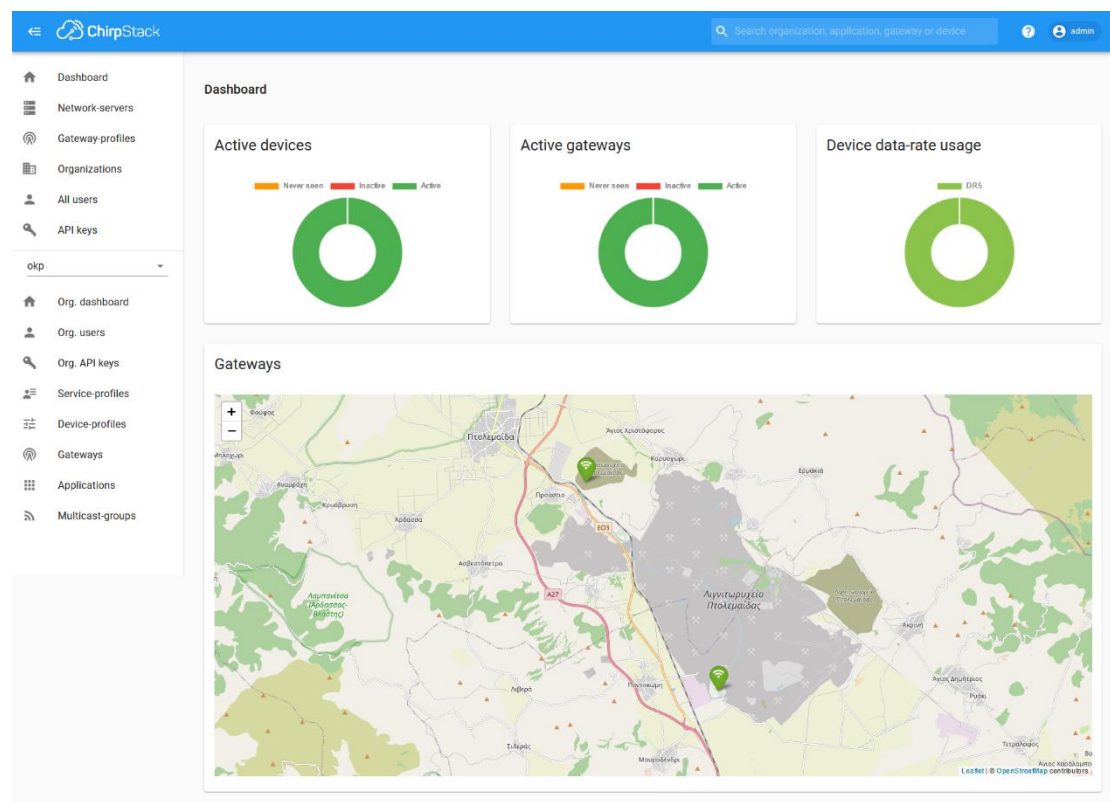
Έγινε εγκατάσταση του Chirpstack και του ThingsBoard σε περιβάλλον Ubuntu 18.0 και σε περιβάλλον Docker για Windows 10. Για λόγους φορητότητας των παραμετροποιήσεων προτιμήσαμε την περίπτωση του Docker για Windows 10.

Έτσι σε τοπικό δίκτυο στο ορυχείο στήθηκε και παραμετροποιήθηκε η σουίτα :

- ChirpStack Gateway Bridge: handles the communication with the LoRaWAN gateways
- ChirpStack Network Server: a LoRaWAN Network Server implementation
- ChirpStack Application Server: a LoRaWAN Application Server implementation
- ThingsBoard Community Edition

## 6.3 Gateway

Στήθηκαν και παραμετροποιήθηκαν δύο gateway Dragino LP08 indoor multichannel gateway με παράλληλες δοκιμές για δημιουργία μιας gateway που στηρίζεται σε Raspberry Pi 4 , το IoT (Internet of Things) Gateway HAT for Raspberry Pi (868MHz/915MHz) with LoRa και λειτουργικό ChirpStack Gateway OS.



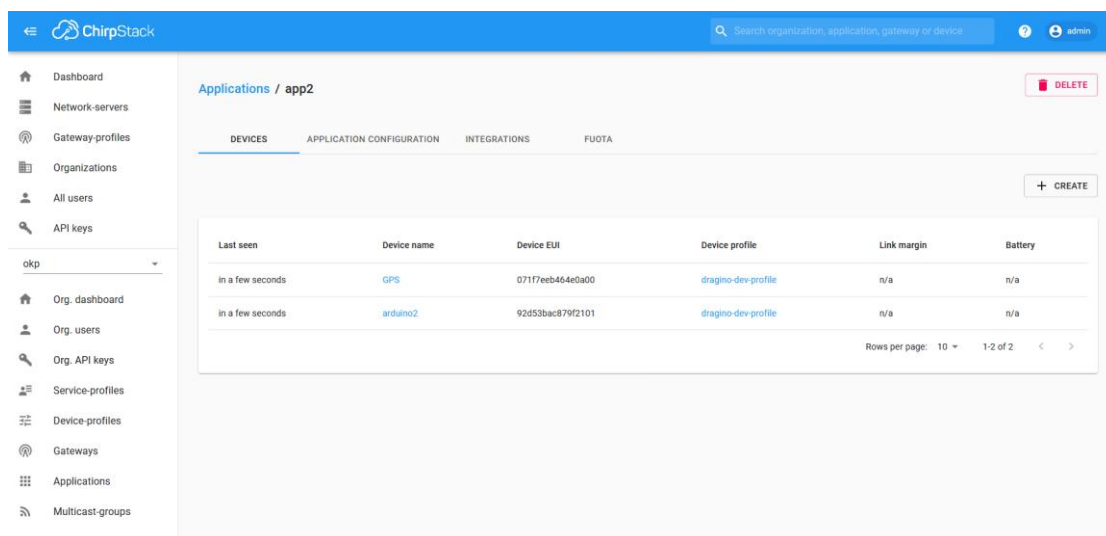
Εικόνα 74- Θέση των Gateways στο ορυχείο

## 6.4 End-nodes

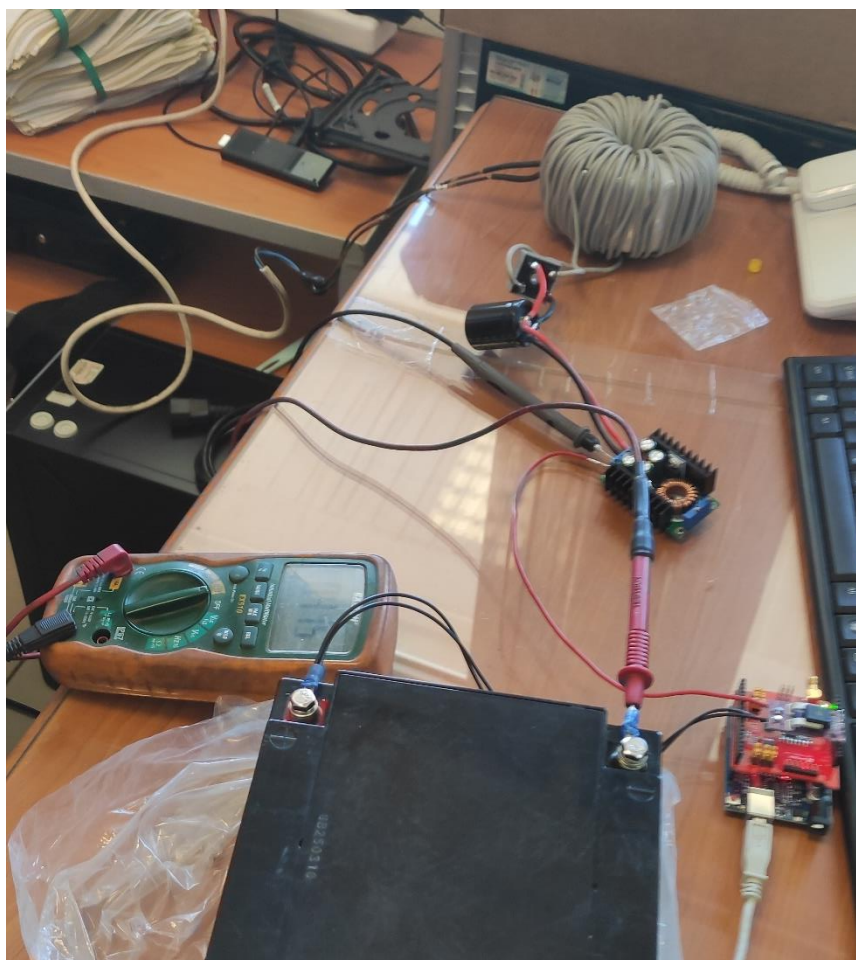
Σαν end-nodes χρησιμοποιήθηκε συνδυασμός Arduino uno με Dragino Uno shield και κατάλληλους για κάθε περίπτωση αισθητήρες.

Ενδεικτικά αναφέρουμε :

- Ambiente Light Sensor – VEML7700
- Current sensor IC – ACS5758LCB
- High Sensitive vibration Sensor
- Hall effect sensor – AH3503
- Infrared Thermometer – MLX90614
- Pressure Transducer - DCSV 0 – 1.2 Mpa



Εικόνα 75- Συσκευές σε Application του ChirpStack



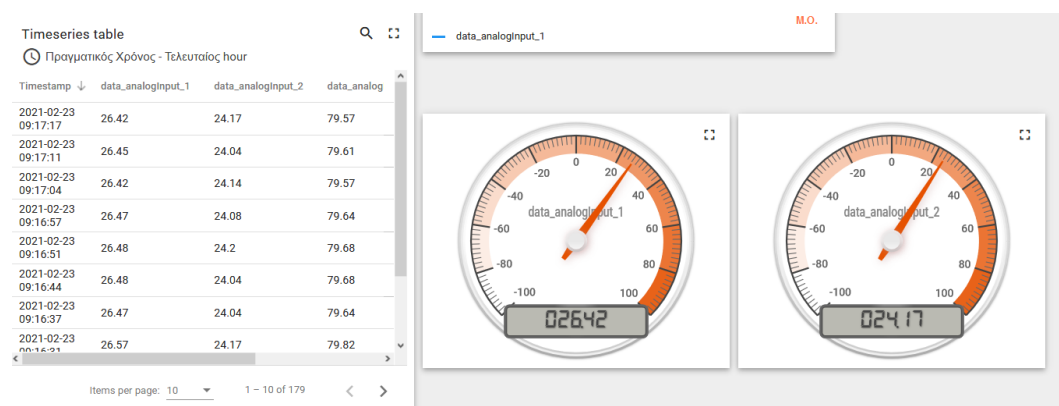
Εικόνα 76- Αποστολή μετρήσεων έντασης και τάσης ρεύματος

## 6.5 Ολοκλήρωση για Οπτικοποίηση δεδομένων

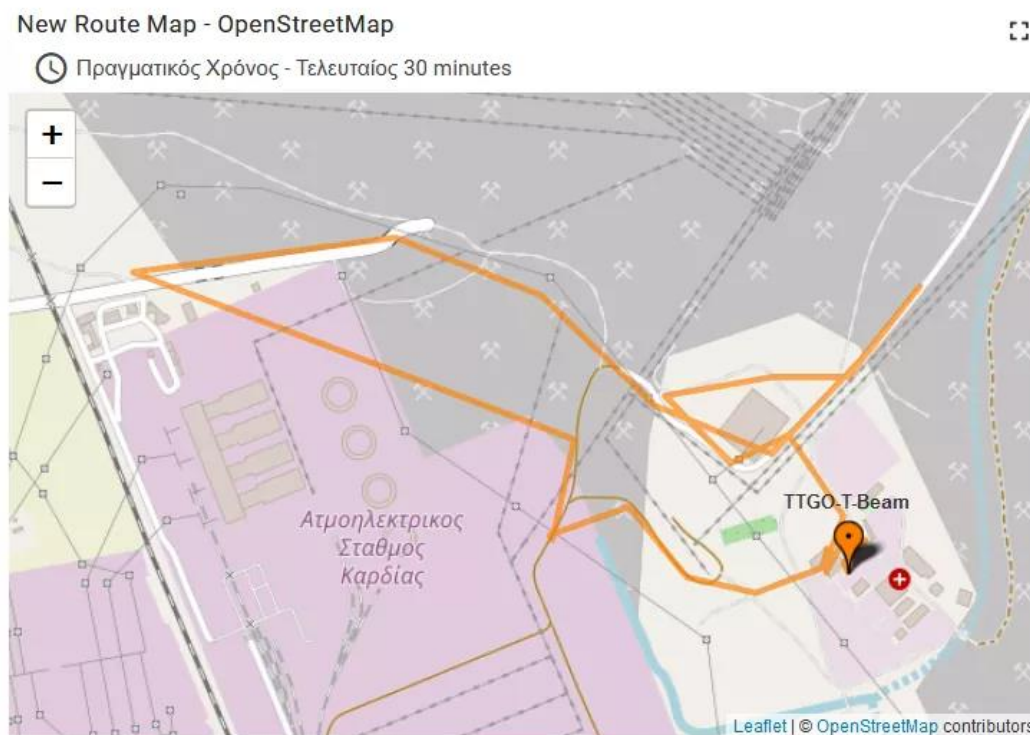
Έγινε διασύνδεση του ChirpStack με τον ThingsBoard Server και στη συνέχεια η οπτικοποίηση των δεδομένων έγινε με Widgets που διαθέτει το ThingsBoard.

Να σημειώσουμε ότι σαν Open Source εφαρμογές οι δυνατότητες που μας δίνονται για επεκτάσεις στην οπτικοποίηση και διαχείριση των δεδομένων είναι τεράστιες.

Η συγκεκριμένη υλοποίηση βρίσκεται αυτή τη στιγμή σε ανάπτυξη και πειραματισμό, αλλά τα αποτελέσματα της υλοποίησης μας δίνουν ενέργεια για την παραπάνω ενασχόληση και εφαρμογή της τεχνογνωσίας που αποκτήθηκε σε πραγματικές συνθήκες.



Εικόνα 77-Λήψη και οπτικοποίηση αναλογικών μετρήσεων



Εικόνα 78 - Καταγραφή θέσης με GPS

## 6.6 Συμπεράσματα - Μελλοντικές επεκτάσεις

Στα προηγούμενα κεφάλαια παρουσιάστηκε το δίκτυο LoRaWAN ως λύση για IoT εφαρμογές και μελετήθηκε το φυσικό του επίπεδο LoRa αλλά και το πρότυπο του δικτύου. Ακόμη, μελετήθηκαν τμήματα του δικτύου όπως end nodes, gateWays και ο Server. Παρουσιάστηκαν οι υλοποιήσεις αυτών αλλά και οι εφαρμογές που αναπτύχθηκαν με σκοπό την αξιοποίηση των παραπάνω. Η διεξοδική ανάλυση κάθε μίας από τις τρεις βασικές δομές του δικτύου LoRa (End-Nodes, Base Station, Server), θα μπορούσε να είναι αντικείμενο ξεχωριστής μελέτης

Η ανάπτυξη τερματικών συσκευών με ιδιαίτερα χαμηλή κατανάλωση ενέργειας αλλά και αξιοπιστίας για παντός τύπου δίκτυα αποτελεί μείζον θέμα σε εφαρμογές IOT και όχι μόνο. Η επιλογή των λύσεων στο κομμάτι των end nodes είχε σκοπό να αναδείξει τέτοιου είδους διαφοροποιήσεις τόσο στο κομμάτι της κατανάλωσης ενέργειας όσο και της αξιοπιστίας. Καταλήξαμε ότι βασική παράμετρος για την επιλογή μιας end node συσκευής είναι η αξιοπιστία τόσο στο θέμα των μετρήσεων(GPS) όσο και στη μετάδοση. Υπάρχουν επίσης διαφορετικά επίπεδα υλοποίησης άρα και πολυπλοκότητας για τον τελικό χρήστη. Παράδειγμα η υλοποίηση με το dragino είναι πιο απαιτητική σε προγραμματιστικές γνώσεις απ' αυτή του TTN-go. Το GPS του Pycom – Pytrack όταν συνοδεύεται από κεραία είναι σίγουρα πιο αξιόπιστο από το αντίστοιχο του dragino shield.

Η ανάπτυξη της gateway εμπίπτει σε ένα τεράστιο κλάδο που ονομάζεται linux embedded systems. Ουσιαστικά, αποτελεί ένα ενσωματωμένο σύστημα με διαθέσιμο linux λειτουργικό, το οποίο είναι ιδιαίτερα βοηθητικό. Υποστηρίζει μεγάλο αριθμό γλωσσών ανάπτυξης λογισμικού, κάτι το οποίο είναι δυσεύρετο σε ενσωματωμένα συστήματα. Δεν ασχοληθήκαμε στην εργασία μας με την ανάπτυξη λογισμικού για gateway αλλά, μετά από μελέτη διαφορετικών έτοιμων συστημάτων αποκτήθηκε η τεχνογνωσία για την υλοποίηση μιας custom λύσης που είναι ίσως και η πρόταση για μια επόμενη μελέτη.

Η πρόταση για μελλοντική μελέτη σ' αυτό το αντικείμενο είναι να χρησιμοποιήσουμε ένα linux embedded σύστημα όπως αυτό του Raspberry 4 για την κατασκευή μιας gateway μικτού μοντέλου. Μια gateway η οποία θα μπορεί να δρομολογήσει πακέτα που προέρχονται από διαφορετικά πρωτόκολλα επικοινωνίας φυσικού μέσου. Ανάλογα λοιπόν με τις ανάγκες που αφορούν την ταχύτητα μετάδοσης, τον όγκο δεδομένων και την ασφάλεια δεδομένων, μια συσκευή IoT θα μπορεί να επιλέξει πως θέλει να κάνει τη μετάδοση δεδομένων.

Η μελέτη που κάναμε και η πρακτική εφαρμογή σε πραγματικές συνθήκες (ΔΕΗ) ανέδειξε ότι η gateway είναι ίσως το πιο σημαντικό στοιχείο σε ένα LoRaWan δίκτυο. Το σίγουρο είναι ότι για να υλοποιήσουμε εφαρμογές ευρείας κλίμακας δεν αρκεί μία gateway άλλα ένα δίκτυο όπως αυτό των ραδιοφωνικών κεραιών. Για ιδιωτική χρήση και σε μικρή κλίμακα θεωρούμε ότι ήδη υπάρχουν λύσεις με μόνο μειονέκτημα αυτό της ασφάλειας της μετάδοσης των

δεδομένων. Συνεπώς ένα ακόμη αντικείμενο μελέτης είναι η ασφάλεια στο επίπεδο του Application Server .

# 7

## Βιβλιογραφία

- [1] O. Standard, «MQTT Version 5.0,» OASIS Standard, 2019.
- [2] «IOT Analytics,» 6 2021. [Ηλεκτρονικό]. Available: <https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/>. [Πρόσβαση 6 2021].
- [3] J. L. T. D. Yonghua Song, «An Internet of Energy Things Based on Wireless LPWAN,» *Engineering*, p. 460–466, 1 8 2017.
- [4] S.-K. & H. K.-i. & K. H.-S. & S. B. Park, «Challenges and Experiment with LoRaWAN,» *Conference: International Conference on Multimedia and Ubiquitous Engineering International Conference on Future Information Technology*, pp. 269-276, 5 2017.
- [5] A. b. t. L. A. T. Committee, «LoRaWAN 1.0.3 Specification,» 7 2018. [Ηλεκτρονικό]. Available: <https://loro-alliance.org/wp-content/uploads/2020/11/lorawan1.0.3.pdf>. [Πρόσβαση 12 2020].
- [6] Semtech, «LoRa® and LoRaWAN®:A Technical Overview,» 12 2019. [Ηλεκτρονικό]. Available: [https://loro-developers.semtech.com/uploads/documents/files/LoRa\\_and\\_LoRaWAN-A\\_Tech\\_Overview-Downloadable.pdf](https://loro-developers.semtech.com/uploads/documents/files/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf). [Πρόσβαση 12 2020].
- [7] A. b. t. L. A. T. C. R. P. Workgroup, «RP2-1.0.1 LoRaWAN® Regional Parameters,» 2 2020. [Ηλεκτρονικό]. Available: [https://loro-alliance.org/wp-content/uploads/2020/11/rp\\_2-1.0.1.pdf](https://loro-alliance.org/wp-content/uploads/2020/11/rp_2-1.0.1.pdf). [Πρόσβαση 12 2020].
- [8] M. & S. B. Knight, «Decoding LoRa: Realizing a Modern LPWAN with SDR,» 6 9 2016. [Ηλεκτρονικό]. Available: <https://pubs.gnuradio.org/index.php/grcon/article/view/8>. [Πρόσβαση 12 2020].
- [9] P. Marcelis, «Frame Loss Characterisation and Data Erasure Coding for LoRaWAN,» Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands, 2016.

- [10] A. S. Matricola, «Enabling relay-based communication in LoRa networks for the Internet of Things: design, implementation and experimental evaluation,» Universit`a di Pisa, Pisa, 2016.
- [11] S. Corporation, «LoRa Modulation Basics,» Semtech , Camarillo, 2015.
- [12] Chirpstack, «ChirpStack open-source LoRaWAN® Network Server,» Chirpstack, [Ηλεκτρονικό]. Available: <https://www.chirpstack.io/project/architecture/>. [Πρόσβαση 12 2020].
- [13] D. W. Andrew Tanenbaum, Computer Networks (5th Edition), 2009.
- [14] J. Y. ,. C. W. M. T. Aloÿs Augustin, «A Study of LoRa: Long Range & Low Power Networks for the Internet of Things,» MDPI, 2016.
- [15] I. Eclipse Foundation, «Eclipse Mosquitto™ An open source MQTT broker,» Eclipse Foundation, Inc., Canada, 2020.
- [16] O. source, «<https://grpc.io/docs/>,» Open source, [Ηλεκτρονικό]. Available: <https://grpc.io/docs/>. [Πρόσβαση 12 2020].
- [17] M. T. Inc, «RN2483 LoRa® Technology Module Command Reference User’s Guide,» Microchip, 2015.
- [18] ChirpStack, «ChirpStack Gateway Bridge,» 12 2020. [Ηλεκτρονικό]. Available: <https://www.chirpstack.io/gateway-bridge/>.
- [19] ChirpStack, «ChirpStack Network Server,» 12 2020. [Ηλεκτρονικό]. Available: <https://www.chirpstack.io/network-server/>.
- [20] D. &. Rescorla, «The Transport Layer Security (TLS) Protocol,» 8 2008. [Ηλεκτρονικό]. Available: <https://tools.ietf.org/html/rfc5246>.
- [21] ChirpStack, «ChirpStack Application Server,» 12 2020. [Ηλεκτρονικό]. Available: <https://www.chirpstack.io/application-server/>.
- [22] T. t. N. TTN, «ABP vs OTAA,» 2020. [Ηλεκτρονικό]. Available: <https://www.thethingsindustries.com/docs/devices/abp-vs-otaa/>.
- [23] <https://avbentem.github.io/airtime-calculator/ttn/eu868/27>, «<https://avbentem.github.io/airtime-calculator/ttn/eu868/27>,» 12 2020. [Ηλεκτρονικό]. Available: <https://avbentem.github.io/airtime-calculator/ttn/eu868/27>.
- [24] J. S. A. Joakim Eriksson, «Investigating the practical performance of the LoRaWAN technology,» Linköping University | Department of Computer Science, 2017.

- [25] F. A. V. Fialho, «Wireless Communication Based on Chirp Signals for LoRa IoT Devices,» 2018.
- [26] A. L. Robin Franksson, «Measuring a LoRa Network Performance, Possibilities and Limitations,» Faculty of Computing Blekinge Institute of Technology, SE-371 79 Karlskrona, Sweden, 2018.
- [27] Semtech, «Semtech and MXC Foundation Announce Open LoRaWAN® Networks,» 2020.
- [28] Dragino, «LG308 LoRaWAN Gateway User Manual,» Dragino, 2020.
- [29] Dragino, «LPS8 LoRaWAN Gateway User Manual,» Dragino, 2021.
- [30] pycom, «lopy-specsheet v1.0,» pycom, 12 2020. [Ηλεκτρονικό]. Available: <https://pycom.io/wp-content/uploads/2018/08/lopy-specsheet.pdf>. [Πρόσβαση 12 2020].
- [31] ThingsBoard, «ThingsBoard Documentation,» ThingsBoard, [Ηλεκτρονικό]. Available: <https://thingsboard.io/docs/>. [Πρόσβαση 1 2021].
- [32] T. Alam, «A Reliable Communication Framework and Its Use in Internet of Things (IoT),» Department of Computer Science, Faculty of Computer and Information Systems, Islamic University of Madinah, Saudi Arabia.