



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**  
**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**  
**ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Υλοποίηση λειτουργίας roaming σε δίκτυα αρχιτεκτονικής LoRaWAN**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**ΤΟΥ**

**Βασιλείου Π. Λεμονιά**

**Επιβλέπων :** Ευστάθιος Συκάς  
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2022





Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής  
Εργαστήριο Δικτύων Υπολογιστών

## Υλοποίηση λειτουργίας roaming σε δίκτυα αρχιτεκτονικής LoRaWAN

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**Βασιλείου Π. Λεμονιά**

**Επιβλέπων :** Ευστάθιος Συκάς  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18η Μαρτίου 2022.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....

Ευστάθιος Συκάς

Καθηγητής Ε.Μ.Π.

.....

Νικόλαος Μήτρου

Καθηγητής Ε.Μ.Π.

.....

Ιωάννα Ρουσσάκη

Επίκουρη Καθηγήτρια Ε.Μ.Π.

Αθήνα, Μάρτιος 2022







Εθνικό Μετσόβιο Πολυτεχνείο  
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών  
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής  
Εργαστήριο Δικτύων Υπολογιστών

(Υπογραφή)

.....

Βασίλειος Π. Λεμονιάς

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Βασίλειος Π. Λεμονιάς, 2022.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.



## Περίληψη

Η παρούσα εργασία ασχολείται με τις ενέργειες που πρέπει να υλοποιηθούν ώστε να επιτευχθεί η λειτουργία roaming σε συστήματα LoRaWAN. Με τον όρο roaming εννοούμε την ικανότητα μιας δικτυακής συσκευής να συνδέεται με το δίκτυο στο οποίο είναι εγγεγραμμένη και ενεργοποιημένη κάνοντας χρήση ενός άλλου δικτύου. Η υπηρεσία αυτή είναι γνωστή σε παλαιότερες τεχνολογίες, όπως η κινητή τηλεφωνία, αλλά η ανάπτυξη του Διαδικτύου των Πραγμάτων (IoT) φέρνει στο προσκήνιο νέες τεχνολογίες διασύνδεσης των συσκευών και ειδικότερα τη διαμόρφωση LoRa και το πρωτόκολλο LoRaWAN. Συνεπώς τίθεται το ερώτημα πως το roaming μπορεί να υλοποιηθεί σε αρχιτεκτονικές LoRa/LoRaWAN.

Στο πρώτο μέρος γίνεται μια εκτενής παρουσίαση του Διαδικτύου των Πραγμάτων (IoT), των δικτύων LPWAN και ειδικότερα της διαμόρφωσης LoRa και του πρωτοκόλλου LoRaWAN. Παρουσιάζονται τα βασικά χαρακτηριστικά των παραπάνω τεχνολογιών και η αρχιτεκτονική ενός δικτύου LoRaWAN με τα αντίστοιχα στοιχεία που το αποτελούν,

Στο δεύτερο μέρος, κάνοντας χρήση κατάλληλου εξοπλισμού και της πλατφόρμας ανοικτού κώδικα Chirpstack παρουσιάζεται η υλοποίηση ενός δικτύου LoRaWAN στο οποίο γίνονται οι κατάλληλες ρυθμίσεις ώστε να αποκτήσει την λειτουργία του roaming. Ο βασικός σκοπός είναι η επίτευξη της σύνδεσης μιας συσκευής . μέσω ενός δικτύου στο οποίο δεν είναι εγγεγραμμένη, στο δίκτυο στο οποίο έχει εγγραφεί και ενεργοποιηθεί ακόμα και όταν βρίσκεται εκτός της εμβέλειας του.

## Λέξεις Κλειδιά

LoRa, LoRaWAN, IoT, LPWAN, end device/node, gateway, Network Server, Application Server, Join Server, Chirpstack, roaming



## **Abstract**

The present work deals with the steps that need to be done to achieve the roaming function in LoRaWAN systems. By roaming we mean the ability of a network device to connect to the network, in which it is registered and activated, using another network. This service is known in older technologies, such as cellular networks, but the development of the Internet of Things (IoT) brings to the fore new device interconnection technologies, such as the LoRa modulation and the LoRaWAN protocol. This raises the question of how we can achieve roaming in LoRa / LoRaWAN architectures.

In the first part there is an extensive presentation of the Internet of Things (IoT), LPWAN networks and in particular the LoRa modulation and the LoRaWAN protocol. The basic features of the above technologies and the architecture of a LoRaWAN network are presented with the corresponding elements that constitute it.

The second part, using appropriate equipment and the open source stack of Chirpstack, presents the implementation of a LoRaWAN network in which the appropriate settings are made to acquire the function of roaming. The main purpose is to connect a device. through a network to which it is not registered, to the network to which it is registered and activated even when it is out of range.

## **Keywords**

LoRa, LoRaWAN, IoT, LPWAN, end device/node, gateway, Network Server, Application Server, Join Server, Chirpstack, roaming



## **Ευχαριστίες**

Η εκπόνηση της παρούσας διπλωματικής εργασίας δεν θα ήταν δυνατή αν δεν είχα την βοήθεια και την υποστήριξη αρκετών ανθρώπων. Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή κ. Ευστάθιο Συκά και τον ερευνητή κ. Δημήτριο Καλογερά για την δυνατότητα που μου έδωσαν να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα, καθώς και για την βοήθεια και καθοδήγηση που μου παρείχαν κατά την διάρκεια της εκπόνηση αυτής της εργασίας. Ακόμα θα ήθελα να ευχαριστήσω τους γονείς μου, τα αδέρφια μου και τους φίλους μου για την στήριξη και το ενδιαφέρον τους.

Βασίλειος Λεμονιάς

Μάρτιος 2022





## **Περιεχόμενα**

<b>Περίληψη</b>	<b>7</b>
<b>Abstract</b>	<b>9</b>
<b>Ευχαριστίες</b>	<b>11</b>
<b>Μέρος Α: Θεωρητικό Υπόβαθρο</b>	<b>21</b>
<b>Κεφάλαιο 1: Εισαγωγή</b>	<b>21</b>
1.1 Αντικείμενο διπλωματικής εργασίας	21
1.2 Σκοπός διπλωματικής εργασίας	21
1.3 Συνεισφορά διπλωματικής εργασίας	22
1.4 Δομή διπλωματικής εργασίας	22
<b>Κεφάλαιο 2: Internet of Things</b>	<b>23</b>
2.1 Ορισμός	23
2.2 Αρχιτεκτονική Συστημάτων IoT	23
2.3 Συνηθισμένες τεχνολογίες δικτυακής διασύνδεσης συστημάτων IoT	25
2.4 Παραδείγματα Εφαρμογών IoT	26
<b>Κεφάλαιο 3: LPWAN</b>	<b>29</b>
3.1 Ορισμός	29
3.2 Πλεονεκτήματα δικτύων LPWAN	29
3.3 Τύποι LPWAN	29
<b>Κεφάλαιο 4: LoRa - LoRaWAN</b>	<b>32</b>
4.1 Εισαγωγή	32
4.2 LoRa Modulation/LoRa PHY	33
4.2.1 Εισαγωγή	33
4.2.2 Shannon – Hartley Theorem	33
4.2.3 Αρχές εξάπλωσης φάσματος (Spread-Spectrum Principles)	34
4.2.4 Direct Sequence Spread Spectrum (DSSS)	34
4.2.5 Chirp spread-spectrum – CSS	36

4.2.6 Παρουσίαση διαμόρφωσης LoRa	38
4.2.7 Ζώνες συχνοτήτων LoRa	40
4.2.8 Ιδιότητες Διαμόρφωσης LoRa	41
4.3 LoRaWAN	43
4.3.1 Εισαγωγή	43
4.3.2 Αρχιτεκτονική δικτύων LoRaWAN	44
4.3.2.1 Τερματικές συσκευές – End devices/nodes	44
4.3.2.2 Πύλες LoRaWAN – Gateways	47
4.3.2.3 Network Server	48
4.3.2.4 Application Server	49
4.3.2.5 Join Server	49
4.2.4 Ενεργοποίηση τερματικής συσκευής – End device activation	50
4.2.4.1 Over-The-Air-Activation στο LoRaWAN 1.0.x	50
4.2.4.2 Over-The-Air-Activation στο LoRaWAN 1.1	53
4.2.4.3 Activation By Personalisation in LoRaWAN 1.0.x	56
4.2.4.4 Activation By Personalisation in LoRaWAN 1.1	56
4.2.5 Δομές και τύποι μηνυμάτων LoRaWAN	57
4.2.6 Περιγραφή μηνύματος MAC δεδομένων	58
4.2.7 Πλεονεκτήματα και Περιορισμοί του LoRaWAN	59
<b>Μέρος Β: Υλοποίηση</b>	<b>61</b>
<b>Κεφάλαιο 5: Προετοιμασία της Υλοποίησης</b>	<b>61</b>
5.1 Εισαγωγή	61
5.2 ChipStack	61
5.3 Εξοπλισμός	62
5.3.1 Raspberry Pi 3 Model B	62
5.3.2 SX1308 Raspberry Pi LoRa Gateway Board	63
5.3.3 LoPy	64

<b>Κεφάλαιο 6: Υλοποίηση βασικής υποδομής LoRaWAN σε Chirpstack</b>	<b>65</b>
6.1 Εισαγωγή	65
6.2 Αρχικοποίηση πύλης (gateway)	65
6.3 Αρχικοποίηση Chirpstack Network Server	69
6.4 Αρχικοποίηση Chirpstack Application Server	71
6.5 Εγγραφή ενός gateway στον Network Server	73
6.6 Ρύθμιση τερματικής συσκευής (end device / node)	77
6.7 Εγγραφή ενός node στο δίκτυο LoRaWAN	77
<b>Κεφάλαιο 7: Υλοποίηση OTA activation μέσω DNS</b>	<b>85</b>
7.1 Εισαγωγή	85
7.2 Εγκατάσταση Home Network	86
7.3 Αρχικοποίηση DNS Server	87
7.4 Υλοποίηση OTAA over DNS	87
<b>Κεφάλαιο 8: Υλοποίηση Roaming</b>	<b>91</b>
8.1 Εισαγωγή	91
8.2 Αρχικοποίηση Home Network	93
8.3 Αρχικοποίηση Guest Network	94
8.4 Υλοποίηση Roaming	95
<b>Κεφάλαιο 9: Συμπεράσματα</b>	<b>97</b>
9.1 Συμπεράσματα	97
9.2 Μελλοντικές Επεκτάσεις	98
<b>Παράρτημα</b>	<b>99</b>
main.py	99
Configuration file Network Server Home Network	100
Configuration file Application Server Home Network Server	107
Configuration file Network Server Guest Network	109
Configuration file Application Server Guest Network	112

Configuration file Gateway Bridge	114
<b>Βιβλιογραφία</b>	<b>116</b>
<b>Κατάλογος Σχημάτων</b>	
Σχήμα 2.1: Αρχιτεκτονική 3 επιπέδων IoT	24
Σχήμα 2.2: Παράδειγμα λειτουργίας ενός συστήματος IoT	24
Σχήμα 2.3: Δέντρο επιλογής κατάλληλης τεχνολογίας δικτύωσης	28
Σχήμα 4.1: Αντιστοίχιση επιπέδων OSI – LoRa	32
Σχήμα 4.2: Αρχιτεκτονική δικτύου LoRaWAN	43
Σχήμα 4.3: Αρχιτεκτονική δικτύου LoRaWAN	44
Σχήμα 4.4: Τερματικές συσκευές (end devices/nodes)	45
Σχήμα 4.5: Συσκευές Class A	45
Σχήμα 4.6: Συσκευές Class B	46
Σχήμα 4.7: Συσκευές Class C	47
Σχήμα 4.8: Πύλες (gateways)	47
Σχήμα 4.9: Network Server	48
Σχήμα 4.10: Application Server	49
Σχήμα 4.11: Join Server	49
Σχήμα 4.12: Διαδικασία Over-The-Air-Activation στο LoRaWAN 1.0.x	51
Σχήμα 4.13: Διαδικασία Over-The-Air-Activation στο LoRaWAN 1.1	53
Σχήμα 4.14: Activation By Personalisation in LoRaWAN 1.0.x	56
Σχήμα 4.15: Activation By Personalisation in LoRaWAN 1.1	57
Σχήμα 4.16: Δομή μηνύματος MAC δεδομένων	58
Σχήμα 5.1: Αρχιτεκτονική Chirpstack	61
Σχήμα 6.1: Αρχιτεκτονική απλού δικτύου LoRaWAN	65

Σχήμα 6.2: Δομή μηνύματος Join Request	82
Σχήμα 6.3: Δομή μηνύματος Join Accept	83
Σχήμα 6.4: Διαδικασία Join	84
Σχήμα 7.1: Αρχιτεκτονική δικτύου LoRaWAN με χρήση OTA activation μέσω DNS	81
Σχήμα 8.1: Αρχιτεκτονική δικτύου LoRaWAN σε λειτουργία roaming	91
Σχήμα 8.2: Passive Roaming Activation	91

## **Κατάλογος Εικόνων**

Εικόνα 5.1: Raspberry Pi 3 Model B	62
Εικόνα 5.2: SX1308 Raspberry Pi LoRa Gateway Board	63
Εικόνα 5.3: LoPy	64
Εικόνα 6.1: Menu αρχικοποίησης Raspberry	66
Εικόνα 6.2: Ενεργοποίηση SPI	66
Εικόνα 6.3: Terminal Semtech UDP Packet Forwarder	69
Εικόνα 6.4: Log file Network Server	71
Εικόνα 6.5: Log file Application Server	73
Εικόνα 6.6: Log in page του web interface του Application Server	74
Εικόνα 6.7: Αρχική σελίδα του web interface του Application Server	74
Εικόνα 6.8: Δημιουργία νέου Network Server	75
Εικόνα 6.9: Δημιουργία νέου Service profile	75
Εικόνα 6.10: Δημιουργία νέου Gateway	76
Εικόνα 6.11: Εμφάνιση του gateway ως ενεργοποιημένο	76
Εικόνα 6.12: Προγραμματισμός του LoPy	77
Εικόνα 6.13: Δημιουργία νέου Device profile	78
Εικόνα 6.14: Ενεργοποίηση OTA activation	78

Εικόνα 6.15: Δημιουργία νέας εφαρμογής	79
Εικόνα 6.16: Εγγραφή νέας τερματικής συσκευής	79
Εικόνα 6.17: Παροχή AppKey	80
Εικόνα 6.18: Σύνδεση τερματικής συσκευής LoPy στο δίκτυο LoRaWAN	80
Εικόνα 6.19: Εμφάνιση τερματικής συσκευής ως ενεργοποιημένη	81
Εικόνα 6.20: Εμφάνιση μηνυμάτων LoRa που ανταλλάσσει η τερματική συσκευή	81
Εικόνα 6.21: Μήνυμα Join Request σε μορφή json	82
Εικόνα 6.22: Μήνυμα Join Accept σε μορφή json	83
Εικόνα 7.1: Αρχική σελίδα web interface του DNS server	87
Εικόνα 7.2: Εγγραφή νέας τερματικής συσκευής	88
Εικόνα 7.3: Μη παροχή του AppKey στον Join Server	88
Εικόνα 7.4: Παρουσίαση της επικοινωνίας του Network Server με τον DNS server	89
Εικόνα 7.5: Εμφάνιση της τερματικής συσκευής ως ενεργοποιημένη	89
Εικόνα 7.6: Ενεργοποίηση της συσκευής (Έχει πάρει διεύθυνση από το δίκτυο)	90
Εικόνα 7.7: Το AppKey δεν είναι γνωστό αλλά η συσκευή έχει ενεργοποιηθεί επιτυχώς	90
Εικόνα 8.1: Παροχή του κατάλληλου net_id στην τερματική συσκευή	95
Εικόνα 8.2: Εμφάνιση του ενεργοποιημένου Gateway και της έλλειψης ενεργοποιημένης συσκευής στο Guest Network	96
Εικόνα 8.3: Εμφάνιση ως ενεργής της τερματικής συσκευής στο Home Network παρά την έλλειψη ενεργού gateway	96

## **Κατάλογος Πινάκων**

Πίνακας 2.1: Προτεινόμενη χρήση διαφόρων τεχνολογιών δικτύωσης	27
Πίνακας 3.1: Συγκριτικός πίνακας τεχνολογιών LPWAN	31
Πίνακας 4.1: Χαρακτηριστικά δικτύων LoRa σε διάφορες γεωγραφικές περιοχές	41
Πίνακας 4.2: Εκδόσεις LoRaWAN	43

Πίνακας 4.3: Πεδία πληροφορίας μηνύματος Join Request Over-The-Air-Activation στο LoRaWAN 1.0.x	51
Πίνακας 4.4: Πεδία πληροφορίας μηνύματος Join Accept Over-The-Air-Activation στο LoRaWAN 1.0.x	52
Πίνακας 4.5: Πεδία πληροφορίας μηνύματος Join Request Over-The-Air-Activation στο LoRaWAN 1.1	54
Πίνακας 4.6: Πεδία πληροφορίας μηνύματος Join Accept Over-The-Air-Activation στο LoRaWAN 1.1	54
Πίνακας 4.7: Δομή μηνύματος uplink	57
Πίνακας 4.8: Δομή μηνύματος downlink	58
Πίνακας 4.9: Τύποι μηνυμάτων LoRaWAN	58
Πίνακας 4.10: Περιγραφή πεδίου MType	59
Πίνακας 9.1 Χρόνος ενεργοποίησης τερματικής συσκευής με ή χωρίς roaming	96

## **Κατάλογος Διαγραμμάτων**

Διάγραμμα 2.1: Συγκριτική ποιοτική αποτύπωση εμπλεκόμενων τεχνολογιών	26
Διάγραμμα 4.1: Περιγραφή ενός συστήματος διάδοσης φάσματος	35
Διάγραμμα 4.2: Διαμόρφωση – διαδικασία φασματικής διασποράς	35
Διάγραμμα 4.3: Αποδιαμόρφωση – διαδικασία φασματικής σύμπτυξης	36
Διάγραμμα 4.4: Παλμός up-chirp και down-chirp στο χρόνο	37
Διάγραμμα 4.5: Μεταβολή φέρουσας (up-chirp, down-chirp)	37
Διάγραμμα 4.6: Πομπός και δέκτης συστήματος CSS	38
Διάγραμμα 4.7: Μεταβολή SF	40





## **Μέρος Α: Θεωρητικό Υπόβαθρο**

### **Κεφάλαιο 1: Εισαγωγή**

#### **1.1 Αντικείμενο διπλωματικής εργασίας**

Το ευρύτερο αντικείμενο της παρούσας εργασίας είναι το Διαδίκτυο των Πραγμάτων (IoT) συνδυαζόμενο με δίκτυα της οικογένειας LPWAN (Low Power Wide Area Network) για τη διασύνδεση συσκευών με αυτό. Πιο συγκεκριμένα γίνεται χρήση της τεχνολογίας διαμόρφωσης LoRa και του MAC πρωτοκόλλου LoRaWAN, αξιοποιώντας την back-end υποστηρικτική υποδομή του Chirpstack. Η εργασία όμως εστιάζει στην χρήση της παραπάνω υποδομής για την επίτευξη λειτουργίας roaming μεταξύ δύο διαφορετικών δικτύων, δηλαδή τον τρόπο με τον οποίο ένα δίκτυο LoRaWAN μπορεί να προωθεί τα μηνύματα μιας συσκευής (που δεν είναι εγγεγραμμένη σε αυτό) στο δίκτυο στο οποίο έχει εγγραφεί αλλά βρίσκεται εκτός της εμβέλειας του και αντίστροφα.

#### **1.2 Σκοπός διπλωματικής εργασίας**

Η διπλωματική εργασία στοχεύει στην υλοποίηση της λειτουργίας του roaming στο περιβάλλον του πρωτοκόλλου LoRaWAN. Συνεπώς θα παρουσιαστεί η υλοποίηση δύο πανομοιότυπων δικτύων LoRaWAN, ενός Guest και ενός Home Network. Κάθε δίκτυο αποτελείται από μία τερματική συσκευή (end node / device), μία πύλη (gateway) και τέλος ένα network και έναν Application Server. Ακόμη εξετάζεται η λύση της ενεργοποίησης μιας τερματικής συσκευής με χρήση της τεχνολογίας DNS για την εύρεση της διεύθυνσης IP του κατάλληλου Join Server.

Η υλοποίηση των παραπάνω βημάτων προϋποθέτει τις εξής εργασίες:

- 1.** Διασύνδεση, προγραμματισμό και φυσική εγκατάσταση ηλεκτρονικών διατάξεων και συστημάτων, για την υλοποίηση μιας πύλης (gateway) για λήψη και αποστολή μηνυμάτων LoRa. Ως gateway χρησιμοποιείται ένας single-board υπολογιστής Raspberry Pi 3 Model B συνδεδεμένος με 1 expansion module SX1308 Raspberry Pi LoRa Gateway Board.
- 2.** Εγκατάσταση, ρύθμιση και διαχείριση λογαριασμών στην πλατφόρμα του Chirpstack για την υλοποίηση ενός Chirpstack Network Server και ενός Chirpstack Application Server. Για το Home Network θα χρησιμοποιηθεί ένα εικονικό μηχάνημα με IP static address και για το Guest Network θα χρησιμοποιηθεί το Raspberry που ήδη αναφέρθηκε.
- 3.** Διασύνδεση, προγραμματισμό και φυσική εγκατάσταση ηλεκτρικών διατάξεων και συστημάτων για την επιτυχή λειτουργία μιας τερματικής συσκευής (end node / device) LoRaWAN. Ως τερματική συσκευή θα αξιοποιηθεί ένας μικροεπεξεργαστής LoPy ο οποίος θα προγραμματιστεί κατάλληλα σε Python.
- 4.** Υλοποίηση των ρυθμίσεων που είναι απαραίτητες για την ενεργοποίηση της τερματικής συσκευής μέσω DNS. Θα γίνει χρήση κατάλληλου DNS server που παρέχεται από το ινστιτούτο AFNIC.
- 5.** Κατάλληλη Ρύθμιση των δικτύων Home και Guest Network ώστε η τερματική συσκευή να μπορεί να συνδεθεί στο Home Network (στο οποίο δεν είναι εγγεγραμμένη) μέσω του Guest Network (στο

οποίο είναι εγγεγραμμένη και ενεργοποιημένη αλλά βρίσκεται εκτός εμβέλειας) κάνοντας χρήση της υπηρεσίας roaming.

### 1.3 Συνεισφορά διπλωματικής εργασίας

Η χρήση της τεχνολογίας LoRa – LoRaWAN έχει επεκταθεί ραγδαία στις εφαρμογές του IoT καθώς έχει καταφέρει να προσφέρει μία αποδοτική εναλλακτική για την διασύνδεση μεταξύ των τερματικών συσκευών και των πυλών. Ως νέα τεχνολογία διασύνδεσης είναι λογικό να υπάρχει το ερώτημα αν μπορούν να υποστηριχθούν λειτουργίες παλαιότερων τεχνολογιών, όπως η κινητή τηλεφωνία (4G / 5G) και το WiFi, και ιδιαίτερα η λειτουργία roaming.

Ως roaming αναφερόμαστε στην ικανότητα χρήσης μιας συσκευής εκτός της εμβέλειας του εγγενούς δικτύου της, κάνοντας χρήση ενός άλλου δικτύου.

Η λειτουργία του roaming είναι ιδιαίτερα σημαντική στα συστήματα IoT ,άρα και στα δίκτυα αρχιτεκτονικής LoRaWAN, διότι επεκτείνει την εμβέλεια τους αφού επιτρέπει στις τερματικές συσκευές να συνδέονται με τις εφαρμογές που υποστηρίζουν ακόμα και αν βρίσκονται εκτός της εμβέλειας των πυλών του δικτύου τους. Είναι απαραίτητο όμως να βρίσκονται εντός της εμβέλειας των πυλών ενός άλλου δικτύου το οποίο θα έχει ρυθμιστεί κατάλληλα για να υποστηρίζει τη λειτουργία του roaming.

Η συνεισφορά της παρούσας διπλωματικής εργασίας συνίσταται στην εξέταση της χρήσης των δυνατοτήτων των τελευταίων εκδόσεων της πλατφόρμας LoRaWAN ανοικτού κώδικα Chirpstack για την υλοποίηση δικτύων LoRaWAN που μπορούν να υποστηρίξουν την υπηρεσία roaming και ο έλεγχος της επιτυχούς λειτουργίας της.

Η προσέγγιση αυτή οδηγεί σε αποδοτικότερη χρήση των διαθέσιμων πόρων και σε επέκταση της εμβέλειας των δικτύων, Απαιτείται όμως παρέμβαση στου διακομιστές των δικτύων από τους διαχειριστές τους. Η επιβάρυνση αυτή όμως σημαίνει ότι ο χρήστης μία τερματικής συσκευής δεν χρειάζεται να προβεί σε κάποια ρύθμιση γιατί η διαδικασία του roaming θα γίνει αυτόματα.

### 1.4 Δομή διπλωματικής εργασίας

Στο Μέρος Α (Θεωρητικό Υπόβαθρο) της εργασίας γίνεται παρουσίαση της θεωρίας αναφορικά με το IoT, τα δίκτυα LPWAN και πιο συγκεκριμένα την διαμόρφωση LoRa. Ακολουθεί μία παρουσίαση του MAC πρωτοκόλλου LoRaWAN με έμφαση στην αρχιτεκτονική του δικτύου και της διαδικασίας Join.

Στο Μέρος Β (Υλοποίηση) αρχικά παρατίθενται η περιγραφή του εξοπλισμού και του λογισμικού που χρησιμοποιήθηκαν στα πλαίσια της εργασίας. Στην συνέχεια γίνεται μία αναλυτική παράθεση του προγραμματισμού και των ρυθμίσεων που είναι απαραίτητα αφενός για την υλοποίηση ενός απλού δικτύου LoRaWAN και αφετέρου για την επέκταση του δικτύου αυτού ώστε να υποστηρίζει την λειτουργία του roaming. Στο Παράρτημα παρουσιάζονται τα configuration files των server και του gateway που χρησιμοποιήθηκαν.

## Κεφάλαιο 2: Internet of Things

### 2.1 Ορισμός

Ο όρος internet of thing (IoT) αναφέρεται σε ένα δίκτυο φυσικών αντικειμένων (things) στα οποία ενσωματώνονται αισθητήρες, λογισμικό ή άλλες τεχνολογίες με σκοπό την σύνδεση και ανταλλαγή πληροφορίας με άλλες συσκευές ή συστήματα μέσω του διαδικτύου[1].

Τα αντικείμενα (things) μπορεί να είναι οτιδήποτε, από οικιακές συσκευές έως αυτοκίνητα. Πρέπει όμως να διαθέτουν τα παρακάτω χαρακτηριστικά για να λειτουργούν ως μέρη ενός συστήματος IoT:

**Αισθητήρες:** Τα δεδομένα (data) που θα επεξεργαστούν από την τελική εφαρμογή ενός συστήματος IoT πρέπει να παραχθούν από το φυσικό περιβάλλον. Προφανώς τα αντικείμενα του συστήματος αυτού πρέπει να ενσωματώνουν τους κατάλληλους αισθητήρες για να συλλέξουν μετρήσιμα και αξιοποιήσιμα δεδομένα όπως είναι η θερμοκρασία και η ταχύτητα.

**Συνδεσιμότητα και Ταυτοποίηση:** Οι συσκευές χρειάζεται να διαθέτουν μία σύνδεση σε κάποιο δίκτυο όπως το internet και μία μοναδική ταυτότητα σε αυτό (π.χ. IP address) για να είναι εφικτή η επικοινωνία της με το υπόλοιπο σύστημα.

**Απομακρυσμένη Λειτουργία:** Μία συσκευή IoT πρέπει να λειτουργεί βασιζόμενη στα δεδομένα (data) που λαμβάνει και στις εντολές (commands-control) που λαμβάνει από το υπόλοιπο σύστημα. Συνεπώς πρέπει να μπορεί να λειτουργεί χωρίς την ανάγκη φυσικής ανθρώπινης παρουσίας[2].

### 2.2 Αρχιτεκτονική Συστημάτων IoT

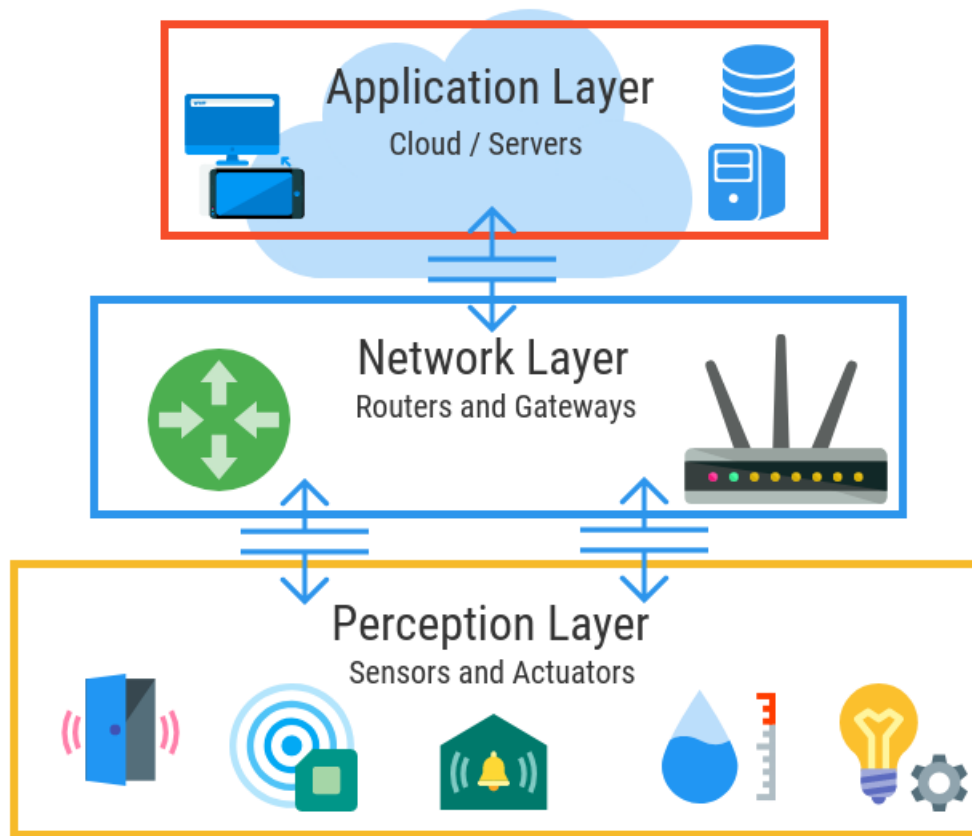
Συνήθως η αρχιτεκτονική των συστημάτων IoT περιλαμβάνει τρία επίπεδα.

**Perception Layer:** Στο επίπεδο αυτό βρίσκονται οι συσκευές και οι αισθητήρες που συλλέγουν τα δεδομένα.

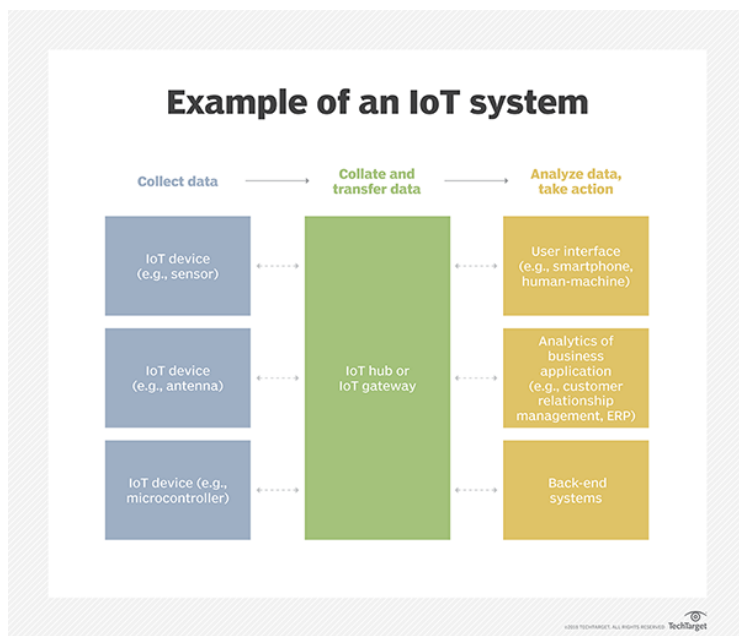
**Network Layer:** Αφορά στην μετάδοση των δεδομένων μεταξύ των συσκευών και των εφαρμογών που θα τα επεξεργαστούν οι οποίες συνήθως βρίσκονται στο cloud. Αυτό γίνεται μέσω συσκευών (router, gateways) που λαμβάνουν τα δεδομένα μέσω διαφόρων τύπων δικτύων όπως bluetooth ή wifi και τα προωθούν στο cloud μέσω του διαδικτύου (IP/TCP).

**Application Layer:** Αναφέρεται στην αποθήκευση και επεξεργασία των δεδομένων από τις κατάλληλες εφαρμογές [3].

Πιο συγκεκριμένα ένα σύστημα IoT περιλαμβάνει συσκευές που συλλέγουν δεδομένα τα οποία στην συνέχεια μέσω ενός δικτύου (4G, lora) αποστέλλουν σε μία συσκευή πύλη (gateway), η οποία θα τα στείλει στην κατάλληλη εφαρμογή για επεξεργασία τοπικά ή μέσω του cloud. Η συσκευή πύλη έχει την δυνατότητα να στείλει εντολές στις συσκευές, οπότε η σχέση τους είναι αμφίδρομη [4].



Σχήμα 2.1: Αρχιτεκτονική 3 επιπέδων IoT [3]



Σχήμα 2.2: Παράδειγμα λειτουργίας ενός συστήματος IoT [4]

## 2.3 Συνηθισμένες τεχνολογίες δικτυακής διασύνδεσης συστημάτων IoT

Για την υλοποίηση ενός συστήματος IoT και ειδικότερα για την επικοινωνία των συσκευών με το gateway μπορούν να χρησιμοποιηθούν πολλές τεχνολογίες. Οι πιο συνηθισμένες παρατίθενται παρακάτω:

**Low Power Wide Area Networks (LPWAN):** Πρόκειται για πρόσφατη τεχνολογία που προσφέρει επικοινωνία μεγάλης εμβέλειας με ελάχιστες απαιτήσεις ενέργειας. Τα δίκτυα LPWAN μπορούν να στείλουν μόνο μικρά μπλοκ δεδομένων με χαμηλό ρυθμό και προτιμούνται για περιπτώσεις χρήσης που δεν απαιτούν υψηλό εύρος ζώνης και δεν είναι ευαίσθητα στο χρόνο. Σήμερα, υπάρχουν τεχνολογίες που λειτουργούν τόσο στο αδειοδοτημένο (NB-IoT, LTE-M) όσο και στο μη αδειοδοτημένο (π.χ. MYTHINGS, LoRa, Sigfox κ.λπ.) φάσμα με διαφορετικούς βαθμούς απόδοσης σε βασικούς παράγοντες δικτύου.

**Cellular (3G/4G/5G):** Τα δίκτυα κινητής μπορεί να μην είναι ιδανικά για όλες τις περιπτώσεις συστημάτων IoT που χρησιμοποιούν μπαταρίες αλλά ενδείκνυνται για συνδεδεμένα αυτοκίνητα ή διαχείριση στόλου για μεταφορικές εταιρείες.

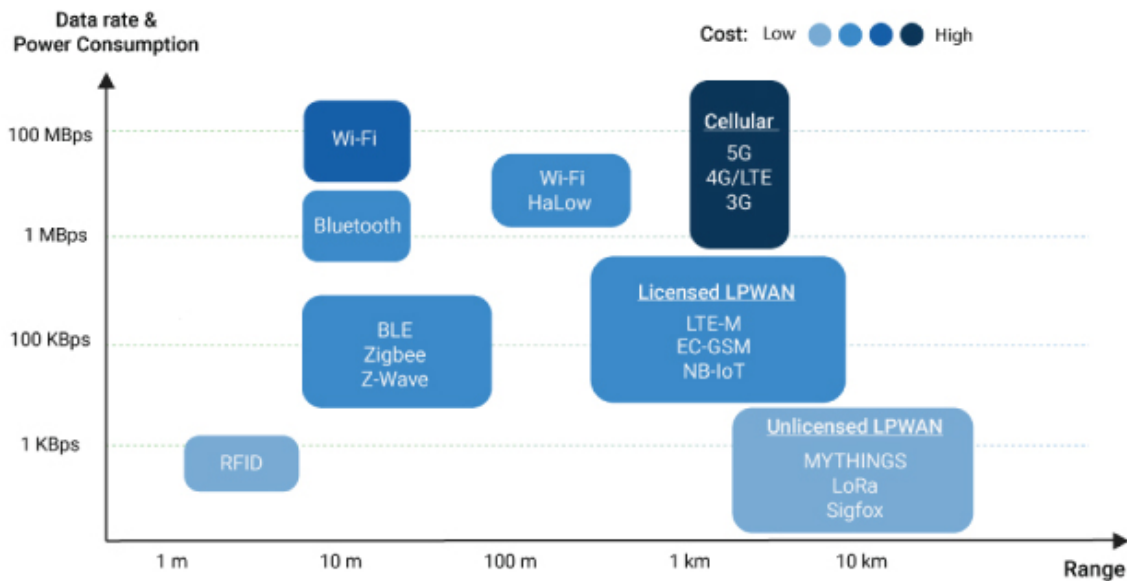
**Zigbee and Other Mesh Protocols:** Το Zigbee είναι ένα ασύρματο πρότυπο δικτύωσης μικρής εμβέλειας, χαμηλής κατανάλωσης, που αναπτύσσεται συνήθως σε τοπολογία πλέγματος για να επεκτείνει την κάλυψη αναμεταδίδοντας τα δεδομένα ενός αισθητήρα σε πολλούς κόμβους αισθητήρων. Σε σύγκριση με το LPWAN, το Zigbee παρέχει υψηλότερους ρυθμούς δεδομένων, αλλά ταυτόχρονα, πολύ μικρότερη απόδοση ισχύος λόγω της διαμόρφωσης πλέγματος ενώ διαθέτει πολύ μικρότερη εμβέλεια.

**Bluetooth and BLE:** Το Bluetooth, που εντάσσεται στην κατηγορία των Wireless Personal Area Networks, είναι μια τεχνολογία επικοινωνίας μικρής εμβέλειας με καλή θέση στην καταναλωτική αγορά. Το Bluetooth Classic προοριζόταν αρχικά για ανταλλαγή δεδομένων από σημείο σε σημείο ή από σημείο σε πολλαπλά σημεία (έως επτά υποτελείς κόμβους) μεταξύ συσκευών. Βελτιστοποιημένο για κατανάλωση ενέργειας, το Bluetooth Low-Energy εισήχθη αργότερα για την αντιμετώπιση εφαρμογών IoT για καταναλωτές μικρής κλίμακας.

**Wi-Fi:** Το Wi-Fi κατέχει ένα κρίσιμο ρόλο στην παροχή μεταφοράς δεδομένων υψηλής απόδοσης τόσο για εταιρικά όσο και για οικιακά περιβάλλοντα. Ωστόσο, στον χώρο του IoT, οι κύριοι περιορισμοί του είναι η περιορισμένη κάλυψη, η επεκτασιμότητα και η κατανάλωση ενέργειας καθιστούν την τεχνολογία αυτή πολύ λιγότερο διαδεδομένη.

**RFID:** Η αναγνώριση ραδιοσυχνοτήτων (RFID) χρησιμοποιεί ραδιοκύματα για τη μετάδοση μικρών ποσοτήτων δεδομένων από μια ετικέτα RFID σε έναν αναγνώστη σε πολύ μικρή απόσταση. Μέχρι τώρα, η τεχνολογία έχει διευκολύνει μια μεγάλη επανάσταση στο λιανικό εμπόριο και την εφοδιαστική και αποτελεί και μία ακόμη εναλλακτική για συστήματα IoT [5].

Κάθε τεχνολογία διασύνδεσης που αναφέρθηκε διαθέτει τα δυνατά και τα αδύνατα σημεία της. Το παρακάτω διάγραμμα αποτυπώνει ποιοτικά τη θέση των τεχνολογιών αυτών μεταξύ τους, με βάση το ρυθμό μετάδοσης/ κατανάλωση ενέργειας και την εμβέλεια καθεμιάς:



Διάγραμμα 2.1: Συγκριτική ποιοτική αποτύπωση εμπλεκόμενων τεχνολογιών [5]

## 2.4 Παραδείγματα Εφαρμογών IoT

Οι τεχνολογίες IoT βρίσκουν χρήση σε πολλές πλευρές της ζωής μας. Αξιοποιούνται στην βιομηχανική παραγωγή, στο αγροτικό τομέας ακόμα και σε καθημερινές μας ασχολίες. Μία κοινά αποδεκτή κατηγοριοποίηση είναι σε IoT industrial, IoT public, IoT appliance και IoT personal.

Στην κατηγορία **IoT industrial** περιλαμβάνονται:

**Βιομηχανία (Manufacturing):** Προληπτική συντήρηση εξοπλισμού, διαχείριση βιομηχανικών δεδομένων (big data), έξυπνα δίκτυα, συστήματα αυτόνομης οδήγησης, καταναμημένα ρομποτικά συστήματα, πλοήγηση, αισθητήρες παρακολούθησης, ενεργοποιητές, διακόπτες, έξυπνοι μετρητές, 3D εκτυπωτές, εξ αποστάσεως αυτοματοποίηση και βελτιστοποίηση της παραγωγικής διαδικασίας, διαχείριση αποθεμάτων, παρακολούθηση συνθηκών εργασίας (π.χ. ποιότητα αέρα).

**Γεωργία (Agriculture):** Συλλογή μετεωρολογικών και γεωλογικών δεδομένων (αισθητήρες θερμοκρασίας, υγρασίας, ταχύτητας ανέμου, βροχόπτωσης, σύστασης εδάφους), πλοήγηση και αυτόνομη οδήγηση γεωργικών μηχανημάτων.

**Κτηνοτροφία (Livestock farming):** Παρακολούθηση και εντοπισμός ζωικού κεφαλαίου, παρακολούθηση υγείας.

**Ιατρική και φροντίδα υγείας (Medical and healthcare):** Φορητές συσκευές παρακολούθησης ζωτικών σημείων, συστήματα ειδοποίησης έκτακτης ανάγκης, συσκευές επιτόπου διάγνωσης, συσκευές παρακολούθησης-ελέγχου-αντιμετώπισης χρόνιων παθήσεων.

**Μεταφορές (Transportation):** Έξυπνο σύστημα ελέγχου κυκλοφορίας, στάθμευσης, συλλογής διοδίων, διαχείρισης εφοδιασμού και στόλου οχημάτων, ελέγχου οχήματος, υποβοήθησης οδηγού, ασφαλείας, επικοινωνίας οχήματος με όχημα (V2V) / με υποδομή (V2I) / με πεζό (V2P), πλοήγηση.

**Ενεργειακή διαχείριση (Energy management):** Έξυπνα δίκτυα (smart grid), έξυπνοι μετρητές, διακόπτες, εξ αποστάσεως έλεγχος και βελτιστοποίηση κατανάλωσης ενέργειας.

Στην κατηγορία **IoT public** περιλαμβάνονται:

**Υποδομές (Infrastructure):** Παρακολούθηση και προληπτική συντήρηση αστικών και αγροτικών υποδομών (π.χ. γεφυρών, σιδηροδρομικών γραμμών), δομικός έλεγχος ασφάλειας κρίσιμων υποδομών, διαχείριση συμβάντων και συντονισμός ανταπόκρισης σε έκτακτη ανάγκη, συστήματα βελτιστοποίησης – προγραμματισμού κατασκευών, διαχείριση απορριμμάτων και αποβλήτων, έλεγχος επιπέδων ηχορύπανσης, αποδοτική διαχείριση δημοτικού φωτισμού.

**Περιβαλλοντικός έλεγχος (Environmental monitoring):** Παρακολούθηση ποιότητας αέρα, νερού, εδάφους, ατμοσφαιρικών συνθηκών, άγριας πανίδας, ανίχνευση μόλυνσης, παρακολούθηση στάθμης ποταμών, έγκαιρη προειδοποίηση για φυσικές καταστροφές.

Στην κατηγορία **IoT appliance** περιλαμβάνονται:

**Έξυπνο σπίτι (Smart home):** Έξυπνος φωτισμός/θέρμανση/κλιματισμός, πολυμέσα, ασφάλεια, έξυπνες συσκευές (π.χ. ψυγείο, τηλεόραση), ενεργειακή διαχείριση κτιρίου, οικιακοί αυτοματισμοί.

Στην κατηγορία **IoT personal** περιλαμβάνονται:

**Προσωπική διασύνδεση:** Smartphones, wearables, tablets, trackers.

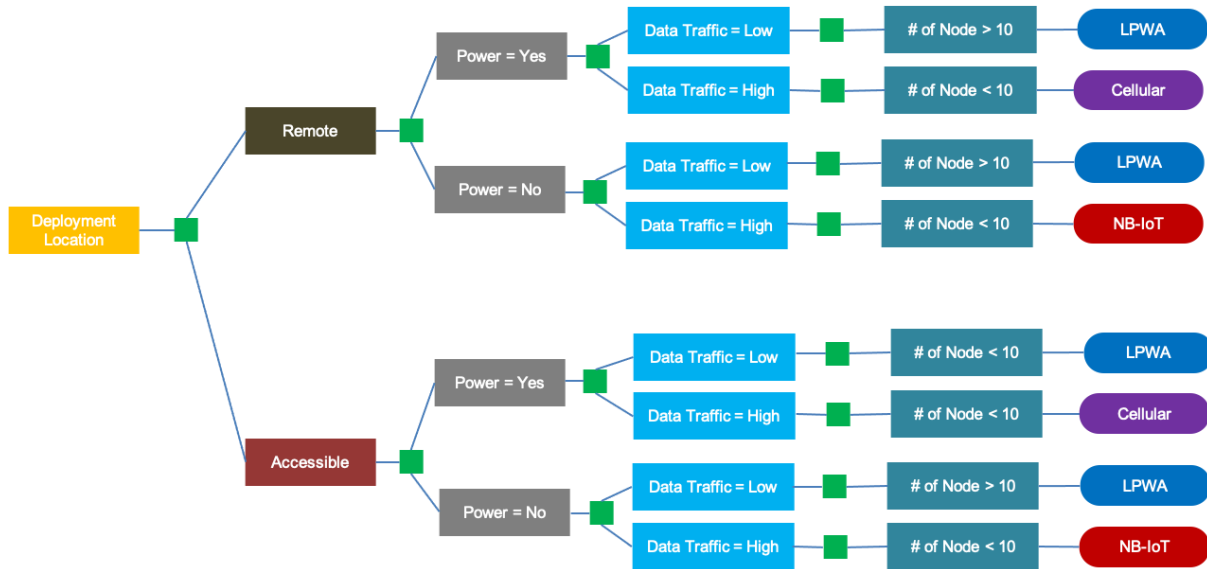
Στο παρακάτω διάγραμμα βλέπουμε σε ποιες εφαρμογές ταιριάζουν καλύτερα οι τεχνολογίες μετάδοσης που απαριθμήσαμε παραπάνω:

Key IoT Verticals	LPWAN (Star)	Cellular (Star)	Zigbee (Mostly Mesh)	BLE (Star & Mesh)	Wi-Fi (Star & Mesh)	RFID (Point-to-point)
Industrial IoT	●	○	○			
Smart Meter	●					
Smart City	●					
Smart Building	●		○	○		
Smart Home			●	●	●	
Wearables	○			●		
Connected Car					○	
Connected Health		●		●		
Smart Retail		○		●	○	●
Logistics & Asset Tracking	○	●				●
Smart Agriculture	●					

● Highly applicable      ○ Moderately applicable

Πίνακας 2.1: Προτεινόμενη χρήση διαφόρων τεχνολογιών δικτύωσης [5]

Ένας ακόμη τρόπος να αποφασίσουμε ποια τεχνολογία μετάδοσης θα χρησιμοποιήσουμε είναι να λάβουμε υπόψη μας τις ανάγκες σε ενέργεια, ταχύτητα μετάδοσης, όγκο δεδομένων και τον αριθμό συσκευών που θα χρησιμοποιήσουμε. Με αυτά τα δεδομένα δημιουργείται ένα διάγραμμα όπως το παρακάτω [6].



*Note: LoRa, Sigfox or NB-IoT depending on provider availability in that region.*

Σχήμα 2.3: Δέντρο επιλογής κατάλληλης τεχνολογίας δικτύωσης [6]

Στα πλαίσια της παρούσας εργασίας χρησιμοποιείται η τεχνολογία LoRa της οικογένειας LPWAN. Συνεπώς στα υπόλοιπα κεφάλαια θα εστιάσουν στις προδιαγραφές της.



## Κεφάλαιο 3: LPWAN

### 3.1 Ορισμός

Τα δίκτυα LPWAN (Low Power Wide Area Network) είναι μια τεχνολογία ασύρματου δικτύου ευρείας περιοχής που διασυνδέει συσκευές χαμηλού εύρους ζώνης τροφοδοτούμενες από μπαταρία με χαμηλούς ρυθμούς bit σε μεγάλες αποστάσεις. Τα LPWAN μπορούν να φιλοξενήσουν μεγέθη πακέτων από 10 έως 1.000 byte σε ταχύτητες ανερχόμενης ζεύξης έως 200 Kbps. Η μεγάλη αυτονομία του LPWAN ποικίλλει από 2 km έως 1.000 km, ανάλογα με την τεχνολογία. Τα περισσότερα LPWAN έχουν μια τοπολογία αστέρα όπου, παρόμοια με το Wi-Fi, κάθε τελικό σημείο συνδέεται απευθείας με κοινά κεντρικά σημεία πρόσβασης (gateways)[7].

### 3.2 Πλεονεκτήματα δικτύων LPWAN

Ο όρος LPWAN εμφανίστηκε το 2013 και δεν υποδηλώνει κάποια συγκεκριμένη τεχνολογική λύση, αλλά συμπεριλαμβάνει μια κατηγορία τεχνολογιών δικτύου που έχουν σχεδιαστεί για να επικοινωνούν ασύρματα σε σχετικά μεγάλες αποστάσεις χρησιμοποιώντας χαμηλότερη ισχύ σε σύγκριση με άλλα δίκτυα, όπως τηλεφωνία, δορυφορικές επικοινωνίες ή WiFi. Αυτά τα δύο βασικά χαρακτηριστικά, δηλαδή η ενεργειακή απόδοση και η ευρεία κάλυψη σήματος καθιστούν την τεχνολογία αυτή ιδανική για τις ανάγκες της συνεχώς εξελισσόμενης αγοράς εφαρμογών IoT [8].

Τα πλεονεκτήματα των δικτύων LPWAN μπορούν να κατηγοριοποιηθούν ως εξής:

**Μεγάλη εμβέλεια:** Η εμβέλεια λειτουργίας της τεχνολογίας LPWAN ποικίλλει από λίγα χιλιόμετρα σε αστικές περιοχές έως πάνω από 15 χιλιόμετρα σε αγροτικές περιοχές. Μπορεί επίσης να επιτρέψει την αποτελεσματική επικοινωνία δεδομένων σε εσωτερικές και υπόγειες τοποθεσίες που προηγουμένως δεν ήταν εφικτές.

**Χαμηλή ισχύς:** Έχοντας βελτιστοποιηθεί ενεργειακά, οι πομποδέκτες LPWAN μπορούν να λειτουργούν με μικρές, φθηνές μπαταρίες για 10-15 χρόνια. Αυτό μεταφράζεται σε μείωση του κόστους συντήρησης και ενέργειας.

**Χαμηλό κόστος:** Τα απλοποιημένα, ελαφριά πρωτόκολλα του LPWAN μειώνουν την πολυπλοκότητα στο σχεδιασμό του υλικού και μειώνουν το κόστος της συσκευής. Η μεγάλη τους εμβέλεια, σε συνδυασμό με μια τοπολογία αστέρα, μειώνει τις ακριβές απαιτήσεις υποδομής. Σε συνδυασμό με τη χρήση ελεύθερων ή ήδη δεσμευμένων για αυτή τη χρήση ζωνών συχνοτήτων μειώνει το κόστος του δικτύου [9].

### 3.3 Τύποι LPWAN

Το LPWAN δεν είναι μια ενιαία τεχνολογία, αλλά μια ομάδα διαφόρων τεχνολογιών που παίρνουν πολλά σχήματα και μορφές. Τα LPWAN μπορούν να χρησιμοποιούν αδειοδοτημένες ή μη συχνοτήτες καθώς και ελεύθερα ή όχι πρότυπα (open standards). Μερικά παραδείγματα είναι τα ακόλουθα:

**Sigfox:** Είναι τεχνολογία αποκλειστικής χρήσης (proprietary technology) που αναπτύσσεται από την εταιρία SigFox, αποτελώντας μία από τις πιο ευρέως διαδεδομένες της οικογένειας LPWAN. Πρόκειται

για τεχνολογία εκπομπής πολύ στενής ζώνης (Ultra Narrow Band – UNB), που βασίζεται στην εγκατάσταση σταθμών βάσης (base stations) για την ανάπτυξη και επέκταση του δικτύου της που χρησιμοποιεί τις ελεύθερες ζώνες των 868 MHz ή 902 MHz. Ενώ μπορεί να μεταφέρει μηνύματα σε αποστάσεις 30-50 km σε αγροτικές περιοχές και 3-10 km σε αστικές περιοχές, το μέγεθος του πακέτου του περιορίζεται στα 150 μηνύματα των 12 byte την ημέρα. Τα πακέτα downlink είναι μικρότερα, μόνο τέσσερα μηνύματα των 8 byte την ημέρα. Η αποστολή δεδομένων στις συσκευές μπορεί να είναι επιρρεπής σε παρεμβολές.

**RPMA(Random phase multiple access):** Είναι τεχνολογία αποκλειστικής χρήσης (proprietary technology) που αναπτύσσεται από την εταιρία Ingenu Inc. Αν και έχει μικρότερη εμβέλεια (έως 50 km οπτικής γραμμής και μεταξύ 5-10 km εκτός οπτικής επαφής), προσφέρει καλύτερη αμφίδρομη επικοινωνία από το Sigfox. Ωστόσο, επειδή λειτουργεί στο φάσμα των 2,4 GHz, είναι επιρρεπής σε παρεμβολές από Wi-Fi, Bluetooth και φυσικές δομές. Επίσης, έχει συνήθως υψηλότερη κατανάλωση ενέργειας από άλλες επιλογές LPWAN.

**LoRa:** Μία τεχνολογία αποκλειστικής χρήσης (proprietary technology) που υποστηρίζεται από τη LoRa Alliance. Εκπέμπει σε πολλές συχνότητες sub-gigahertz, και καθίσταται με αυτό το τρόπο λιγότερο επιρρεπής σε παρεμβολές. Χρησιμοποιεί διαμόρφωση εξάπλωσης φάσματος (CSS – Chirp Spread Spectrum) και επιτρέπει στους χρήστες να ορίζουν το μέγεθος του πακέτου. Το υποκείμενο τσιπ πομποδέκτη που χρησιμοποιείται για την υλοποίηση του LoRa είναι διαθέσιμο μόνο από την Semtech Corporation, την εταιρεία πίσω από την τεχνολογία. Το LoRaWAN είναι το πρωτόκολλο επιπέδου ελέγχου πρόσβασης πολυμέσων (MAC) που διαχειρίζεται την επικοινωνία μεταξύ συσκευών LPWAN και πυλών.

**Weightless-N, αμφίδρομο Weightless-P και Weightless-W:** Η Weightless SIG έχει αναπτύξει τρία πρότυπα LPWAN: Το μονής κατεύθυνσης Weightless-N, αμφίδρομο Weightless-P και Weightless-W, το οποίο είναι επίσης αμφίδρομο και δεν έχει αχρησιμοποίητο φάσμα. Το Weightless-N και το Weightless-P είναι πιο δημοφιλείς επιλογές λόγω της μικρότερης διάρκειας ζωής της μπαταρίας του Weightless-W. Το Weightless-N και το Weightless-P εκτελούνται στο μη αδειοδοτημένο φάσμα κάτω του 1 GHz, αλλά υποστηρίζουν επίσης αδειοδοτημένη λειτουργία φάσματος χρησιμοποιώντας τεχνολογία στενής ζώνης 12,5 kHz.

**Narrowband-IoT (NB-IoT) και LTE-M:** Είναι και τα δύο πρότυπα 3ης γενιάς Partnership Project (3GPP) που λειτουργούν στο αδειοδοτημένο φάσμα. Ενώ έχουν παρόμοιες επιδόσεις με άλλα πρότυπα, λειτουργούν στην υπάρχουσα υποδομή κινητής τηλεφωνίας, επιτρέποντας στους παρόχους υπηρεσιών να προσθέτουν γρήγορα συνδεσιμότητα κινητής τηλεφωνίας IoT στα χαρτοφυλάκια υπηρεσιών τους.

**NB-IoT (γνωστό και ως CAT-NB1):** Λειτουργεί στην υπάρχουσα υποδομή LTE και Global System for Mobile (GSM). Προσφέρει ταχύτητες uplink και downlink περίπου 200 Kbps, χρησιμοποιώντας μόνο 200 kHz διαθέσιμου εύρους ζώνης.

**LTE-M (γνωστό και ως CAT-M1):** Προσφέρει υψηλότερο εύρος ζώνης από το NB-IoT και το υψηλότερο εύρος ζώνης από οποιαδήποτε τεχνολογία LPWAN.

Υπάρχουν και άλλες τεχνολογίες LPWAN όπως οι GreenOFDM (GreenWaves Technologies), DASH7 (Haystack Technologies Inc.), Symphony Link (Link Labs Inc.), ThingPark Wireless (Actility), Ultra Narrow Band (διάφορες εταιρίες όπως Telensa, Nwave και Sigfox) και WAVIoT[7].

Στο επόμενο πίνακα παρουσιάζεται μία σύγκριση των παραπάνω τεχνολογιών LPWAN ως προς τα διάφορα χαρακτηριστικά τους όπως το εύρος ζώνης (bandwidth) και η εμβέλεια (range)[10].

Name of Standard	Weightless			SigFox	LoRaWAN	LTE-Cat M	IEEE P802.11ah (low power WiFi)	Dash7 Alliance Protocol 1.0	Ingenu RPMA	nWave
	-W	-N	-P							
Frequency Band	TV whitespace (400-800 MHz)	Sub-GHz ISM	Sub-GHz ISM	868 MHz/902 MHz ISM	433/868/780/915 MHz ISM	Cellular	License-exempt bands below 1 GHz, excluding the TV White Spaces	433, 868, 915 MHz ISM/SRD	2.4 GHz ISM	Sub-GHz ISM
Channel Width	5MHz	Ultra narrow band (200Hz)	12.5 kHz	Ultra narrow band	EU: 8x125kHz, US 64x125kHz/8x125kHz, Modulation: Chirp Spread Spectrum	1.4MHz	1/2/4/8/16 MHz	25 KHz or 200 KHz	1 MHz (40 channels available)	Ultra narrow band
Range	5km (urban)	3km (urban)	2km (urban)	30-50km (rural), 3-10km (urban), 1000km LoS	2-5k (urban), 15k (rural)	2.5- 5km	Up to 1Km (outdoor)	0 – 5 km	>500 km LoS	10km (urban), 20-30km (rural)
End Node Transmit Power	17 dBm	17 dBm	17 dBm	10μW to 100 mW	EU:<+14dBm, US:<+27dBm	100 mW	Dependent on Regional Regulations (from 1 mW to 1 W)	Depending on FCC/ETSI regulations	to 20 dBm	25-100 mW
Packet Size	10 byte min.	Up to 20 bytes	10 byte min.	12 bytes	Defined by User	~100-~1000 bytes typical	Up to 7,991 Bytes (w/o Aggregation), up to 65,535 Bytes (with Aggregation)	256 bytes max / packet	Flexible (6 bytes to 10 kbytes)	12 byte header, 2-20 byte payload
Uplink Data Rate	1 kbps to 10 Mbps	100bps	200 bps to 100 kbps	100 bps to 140 messages/day	EU: 300 bps to 50 kbps, US:900-100kbps	~200kbps	150 Kbps ~ 346.666 Mbps	9.6 kb/s, 55.55 kbps or 166.667 kb/s	AP aggregates to 624 kbps per Sector (Assumes 8 channel Access Point)	100 bps
Downlink Data Rate	1 kbps to 10 Mbps	No downlink	200 bps to 100 kbps	Max 4 messages of 8 bytes/day	EU: 300 bps to 50 kbps, US:900-100kbps	~200kbps	150 Kbps ~ 346.666 Mbps	9.6 kb/s, 55.55 kbps or 166.667 kb/s	AP aggregates to 156 kbps per Sector (Assumes 8 channel Access Point)	--
Devices per Access Point	Unlimited	Unlimited	Unlimited	1M	Uplink:>1M, Downlink:<100k	20k+	8191	NA (connectionless communication)	Up to 384,000 per sector	1M
Topology	Star	Star	Star	Star	Star on Star	Star	Star, Tree	Node-to-node, Star, Tree	Typically Star, Tree supported with an RPMA extender	Star
End node roaming allowed	Yes	Yes	Yes	Yes	Yes	Yes	Allowed by other IEEE 802.11 amendments (e.g., IEEE 802.11r)	Yes	Yes	Yes
Governing Body	<a href="#">Weightless SIG</a>			<a href="#">Sigfox</a>	<a href="#">LoRa Alliance</a>	<a href="#">3GPP</a>	IEEE 802.11 working group	<a href="#">Dash7 Alliance</a>	<a href="#">Ingenu (formerly OnRamp)</a>	<a href="#">Weightless SIG</a>
Status	Limited deployment awaiting spectrum availability	Deployment beginning	Standard in development. Scheduled release 4Q 2015	In deployment	Spec released June 2015, in deployment	Release 13 expected 2016	Targeting 2016 release	Released May 2015	In Deployment	In Deployment

Source: EDN.com - Copyright 2015 UBM Americas Rev. 9/15/15

Πίνακας 3.1: Συγκριτικός πίνακας τεχνολογιών LPWAN [10]

Παρατηρούμε ότι κάθε τεχνολογία υπερτερεί σε κάποια χαρακτηριστικά, ενώ υπολείπεται σε κάποια άλλα. Για παράδειγμα σε μία εφαρμογή που απαιτεί χαμηλό latency, θα προτιμηθεί η NB – IoT λύση, ενώ σε μία εφαρμογή μεταφοράς δεδομένων μερικών δεκάδων bytes υπό ελαστικά χρονικά όρια, θα προτιμηθεί μία λύση SigFox ή LoRa. Η επιλογή τεχνολογίας εξαρτάται από τον εκάστοτε χρήστη και την εφαρμογή που υλοποιεί ώστε να προσδιοριστούν και οι ανάγκες όπως το απαιτούμενο εύρος ζώνης και η απόσταση μεταξύ των συσκευών και της πύλης. Δεν πρέπει να ξεχνάμε ότι κάποιες τεχνολογίες δεν είναι διαθέσιμες δωρεάν (SigFox) ή έχουν περιορισμούς στον εξοπλισμό.

Στην παρούσα εργασία έχει χρησιμοποιηθεί το πρότυπο LoRa και στο επόμενο κεφάλαιο θα παρουσιαστεί με μεγαλύτερη λεπτομέρεια.

## Κεφάλαιο 4: LoRa - LoRaWAN

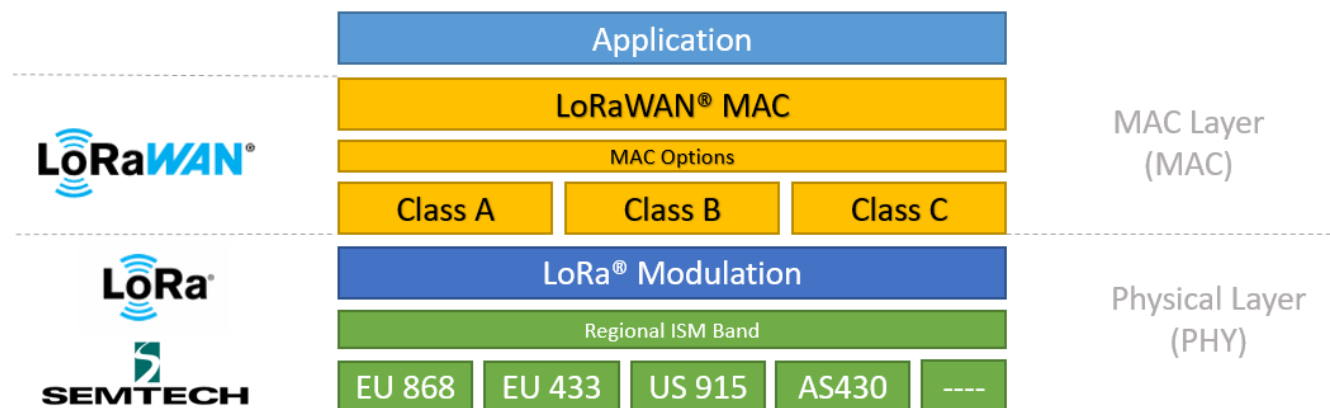
### 4.1 Εισαγωγή

Με τον όρο LoRa αναφερόμαστε σε μία ασύρματη τεχνολογία διαμόρφωσης που επιτρέπει την υλοποίηση δικτύων LPWAN και αποτελείται από τα παρακάτω δύο πρωτόκολλα:

**LoRa Modulation/LoRa PHY**: Υλοποιεί το φυσικό επίπεδο (Physical layer). Πρόκειται για ένα πρωτόκολλο διαμόρφωσης που επιτρέπει την επίτευξη ασύρματης ζεύξης μεγάλης εμβέλειας.

**LoRaWAN**: Υλοποιεί το επίπεδο ζεύξης δεδομένων/υποεπίπεδο MAC (Data link layer/MAC sublayer). Το LoRaWAN είναι ένα ανοιχτό πρωτόκολλο διασύνδεσης και δικτύωσης που παρέχει ασφαλείς υπηρεσίες αμφίδρομης επικοινωνίας, κινητικότητας και γεωγραφικού εντοπισμού[11].

Αν συνεχίσουμε την συσχέτιση με το OSI (Open Systems Interconnection model) υπάρχει και το επίπεδο εφαρμογών (Application layer) το οποίο υλοποιείται από εφαρμογές τεχνολογίας cloud και περιλαμβάνει την αποθήκευση και την επεξεργασία των δεδομένων που παρέχονται από τα προηγούμενα επίπεδα.



Σχήμα 4.1: Αντιστοίχιση επιπέδων OSI – LoRa [11]

## 4.2 LoRa Modulation/LoRa PHY

### 4.2.1 Εισαγωγή

Η τεχνολογία LoRa είναι μια ασύρματη τεχνολογία που παρέχει το πλεονέκτημα της μεγάλης εμβέλειας, χαμηλής ισχύος και ασφαλούς, έναντι παρεμβολών, μετάδοσης δεδομένων για εφαρμογές IoT. Βασίζεται στη διαμόρφωση εξάπλωσης φάσματος CSS (Chirp Spread Spectrum), η οποία έχει χαρακτηριστικά χαμηλής ισχύος όπως η διαμόρφωση FSK, αλλά μπορεί να χρησιμοποιηθεί για επικοινωνίες μεγάλης εμβέλειας. Λειτουργεί με μεταβλητό ρυθμό δεδομένων, χρησιμοποιώντας ορθογώνιους παράγοντες διασποράς (orthogonal spreading factors), θυσιάζοντας το ρυθμό δεδομένων για εμβέλεια ή ισχύ, έτσι ώστε να βελτιστοποιεί την απόδοση του δικτύου σε ένα περιορισμένο εύρος ζώνης[12]. Η τεχνολογία LoRa βρίσκει χρήση στην ασύρματη σύνδεση μηχανημάτων όπως αισθητήρες και πύλες(gateways).

Ανήκει στην κατηγορία των τεχνολογιών (LPWAN) και αναπτύχθηκε από μια γαλλική εταιρεία την Cycleo, η οποία στη συνέχεια εξαγοράστηκε το 2012 από τη αμερικάνικη Semtech. Η Semtech διαχειρίζεται την τεχνολογία LoRa ως ιδρυτικό μέλος της συμμαχίας LoRa Alliance η οποία έχει περισσότερα από 500 εταιρείες μέλη. Η τεχνολογία αποτελεί μία τεχνολογία φυσικού επιπέδου (physical layer) αποκλειστικής χρήσης (proprietary). Η τεχνολογία LoRa είναι ανεξάρτητη από τις υλοποιήσεις στα ανώτερα επίπεδα άρα και πιο εύχρηστη για χρήση σε υφιστάμενα συστήματα. Συνηθίζεται να συνδυάζεται με το πρωτόκολλο LoRaWAN.

### 4.2.2 Shannon – Hartley Theorem

Η αναφορά στο θεώρημα Shannon – Hartley είναι απαραίτητη για την κατανόηση των τεχνικών εξάπλωσης φάσματος. Το θεώρημα Shannon-Hartley καθορίζει τον μέγιστο ρυθμό με τον οποίο μπορούν να μεταδοθούν πληροφορίες μέσω ενός καναλιού επικοινωνίας ενός συγκεκριμένου εύρους ζώνης παρουσία θορύβου.

Ορίζει τη χωρητικότητα Shannon (C) ενός τηλεπικοινωνιακού διαύλου, δηλαδή το θεωρητικό άνω όριο του ρυθμού μετάδοσης δεδομένων χωρίς απώλεια πληροφορίας λόγω λάθους, με την παραδοχή ότι ο διάυλος υπόκειται σε προσθετικό λευκό θόρυβο Gauss[12].

$$C = B * \log_2\left(1 + \frac{N}{S}\right) \quad (1)$$

C: χωρητικότητα καναλιού (bit/s)

B: εύρος ζώνης καναλιού (Hz)

S: μέση ισχύς λαμβανόμενου σήματος (Watts)

N: μέση ισχύς θορύβου ή παρεμβολών (Watts)

S/N: λόγος σήματος προς θόρυβο (SNR) που εκφράζεται ως γραμμικός λόγος ισχύος

Για εφαρμογές ευρέως φάσματος ο λόγος σήματος προς θόρυβο είναι μικρός, καθώς η ισχύς του σήματος είναι συχνά χαμηλότερη από το επίπεδο του θορύβου. Υποθέτοντας ένα επίπεδο θορύβου τέτοιο ώστε  $S/N \ll 1$

$$\frac{C}{B} = 1.433 * \frac{S}{N} \quad (2)$$

Η εξίσωση 2 μπορεί να ξαναγραφεί προσεγγιστικά ως:

$$\frac{C}{B} \approx \frac{S}{N} \quad \text{ή} \quad \frac{N}{S} \approx \frac{B}{C} \quad (3)$$

Από την εξίσωση 3 φαίνεται ότι για τη μετάδοση πληροφοριών χωρίς σφάλματα σε ένα κανάλι σταθερού S/N (SNR) , μόνο το εύρος ζώνης (B) του μεταδιδόμενου σήματος χρειάζεται να αυξηθεί. Αυτό ισχύει αν διατηρείται σταθερή η χωρητικότητα (C) του διαύλου.

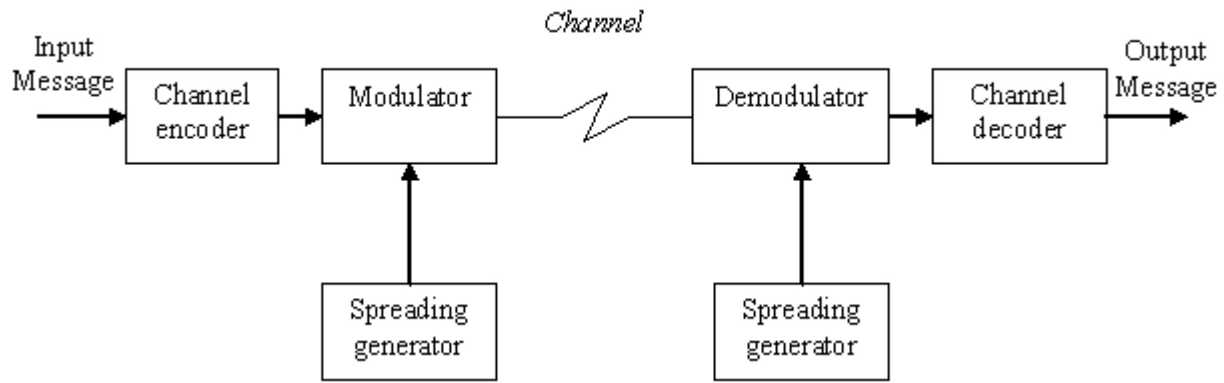
### 4.2.3 Αρχές εξάπλωσης φάσματος (Spread-Spectrum Principles)

Όπως σημειώθηκε παραπάνω, αυξάνοντας το εύρος ζώνης (bandwidth) του σήματος μπορούμε να αντισταθμίσουμε την υποβάθμιση της αναλογίας σήματος προς θόρυβο (SNR) ενός τηλεπικοινωνιακού διαύλου. Το θεώρημα Shannon – Hartley βρίσκει εφαρμογή στις τεχνικές εξάπλωσης φάσματος δηλαδή σε μεθόδους με τις οποίες ένα σήμα που παράγεται με ένα συγκεκριμένο εύρος ζώνης εξαπλώνεται σκόπιμα στον τομέα συχνότητας, με αποτέλεσμα ένα σήμα με μεγαλύτερο εύρος ζώνης. Η συνολική ισχύς του σήματος παραμένει σταθερή αλλά διασπείρεται φασματικά [14]. Ένα σήμα διαμόρφωσης εξάπλωσης φάσματος δεν έχει ξεκάθαρα διακριτή κορυφή στο φάσμα. Αυτό κάνει το σήμα πιο δύσκολο να ξεχωρίσει από το θόρυβο και επομένως πιο δύσκολο να μπλοκαριστεί ή να υποκλαπεί [15]. Οι κυριότερες τεχνικές διασποράς φάσματος είναι οι Direct-sequence spread-spectrum (DSSS), Frequency-hopping spread-spectrum (FHSS) και Chirp spread-spectrum (CSS).

### 4.2.4 Direct Sequence Spread Spectrum (DSSS)

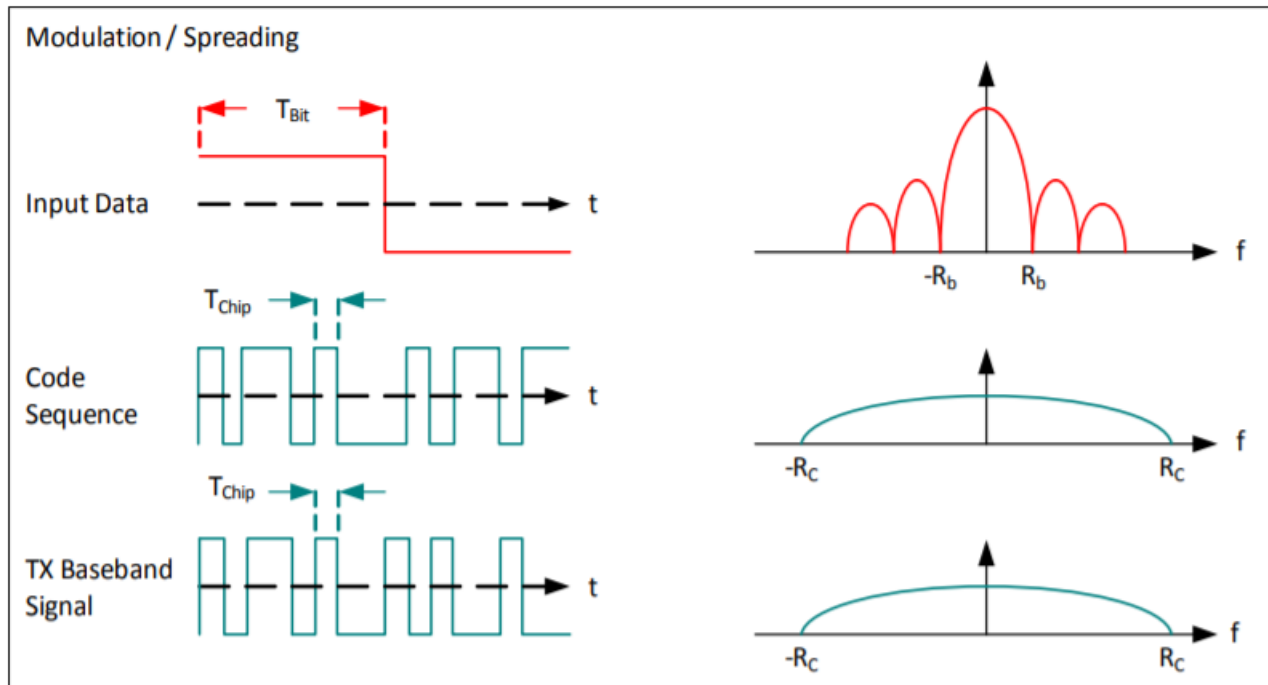
Στα παραδοσιακά συστήματα Direct Sequence Spread Spectrum (DSSS) το σήμα που θέλουμε να μεταδοθεί από το πομπό πολλαπλασιάζεται με έναν κώδικα εξάπλωσης, δηλαδή μια ακολουθία παλμών (chips) πολύ μικρότερης περιόδου από αυτούς του σήματος πληροφορίας (code sequence / chip sequence). Με αυτή την μέθοδο το εύρος ζώνης του σήματος που προκύπτει κατανέμεται πέρα από το εύρος ζώνης που καταλαμβάνει το αρχικό σήμα [12].

Στον δέκτη, το επιθυμητό σήμα δεδομένων ανακτάται πολλαπλασιάζοντας εκ νέου με ένα τοπικά παραγόμενο αντίγραφο του κώδικα εξάπλωσης. Αυτή η διαδικασία επαναφέρει το σήμα πίσω στο αρχικό εύρος ζώνης και μπορεί να περιγραφεί ως η αντίθετη διαδικασία από αυτή που συντελείται στον πομπό. Πρέπει να σημειωθεί ότι στον δέκτη είναι αναγκαίο να χρησιμοποιηθεί η ίδια ακολουθία παλμών όπως και στον πομπό για να γίνει αξιόπιστη ανάκτηση του αρχικού σήματος [12].

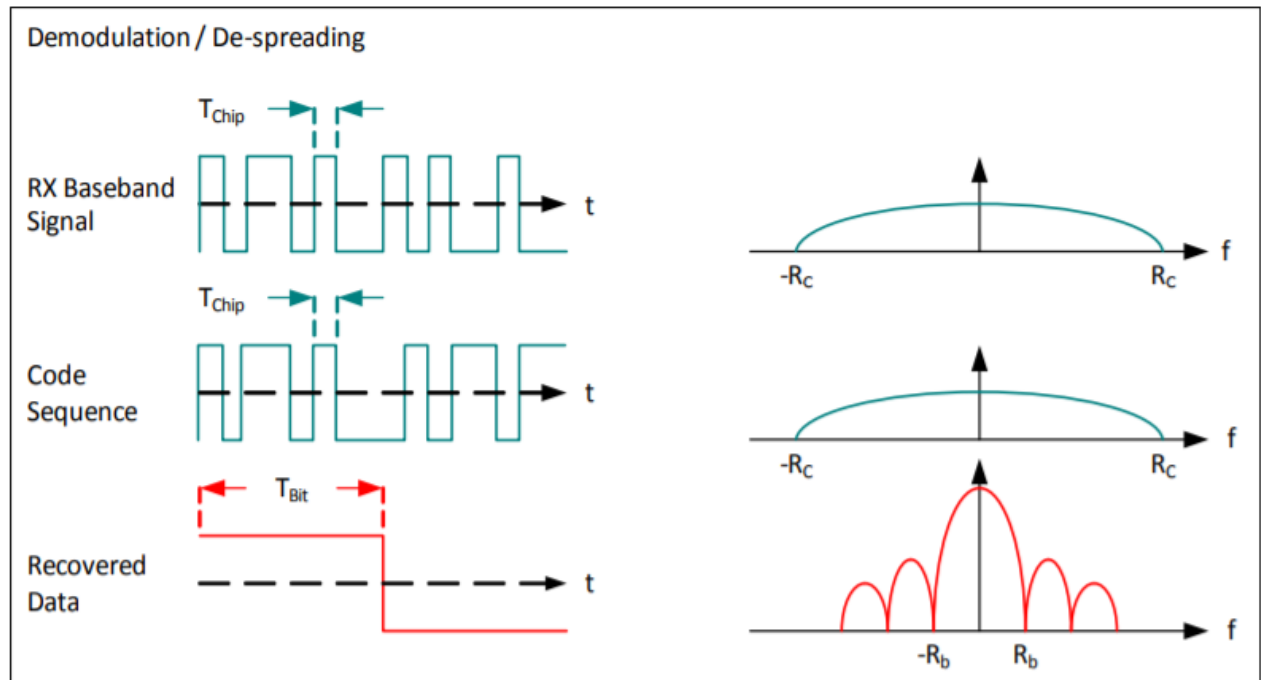


Διάγραμμα 4.1: Περιγραφή ενός συστήματος διάδοσης φάσματος [12]

Ακολουθούν διαγράμματα με την ποιοτική και φασματική διασπορά ενός σήματος πληροφορίας κατά τη διαμόρφωση και αποδιαμόρφωση ενός σήματος με χρήση DSSS.



Διάγραμμα 4.2: Διαμόρφωση – διαδικασία φασματικής διασποράς [12]



Διάγραμμα 4.3: Αποδιαμόρφωση – διαδικασία φασματικής σύμπτυξης [12]

Το μέγεθος της διασποράς εξαρτάται από την αναλογία "chip per bit" (ο λόγος της ακολουθίας παλμών προς τον επιθυμητό ρυθμό δεδομένων), αναφέρεται ως κέρδος επεξεργασίας (processing gain /  $G_p$ ) και συνήθως εκφράζεται σε dB [12].

$$G_p = 10 * \log_{10} \left( \frac{R_c}{R_b} \right) (dB) \quad (4)$$

$G_p$ : κέρδος επεξεργασίας (processing gain) (dB)

$R_c$ : chip rate (Chips/second)

$R_b$ : bit-rate (bits/second)

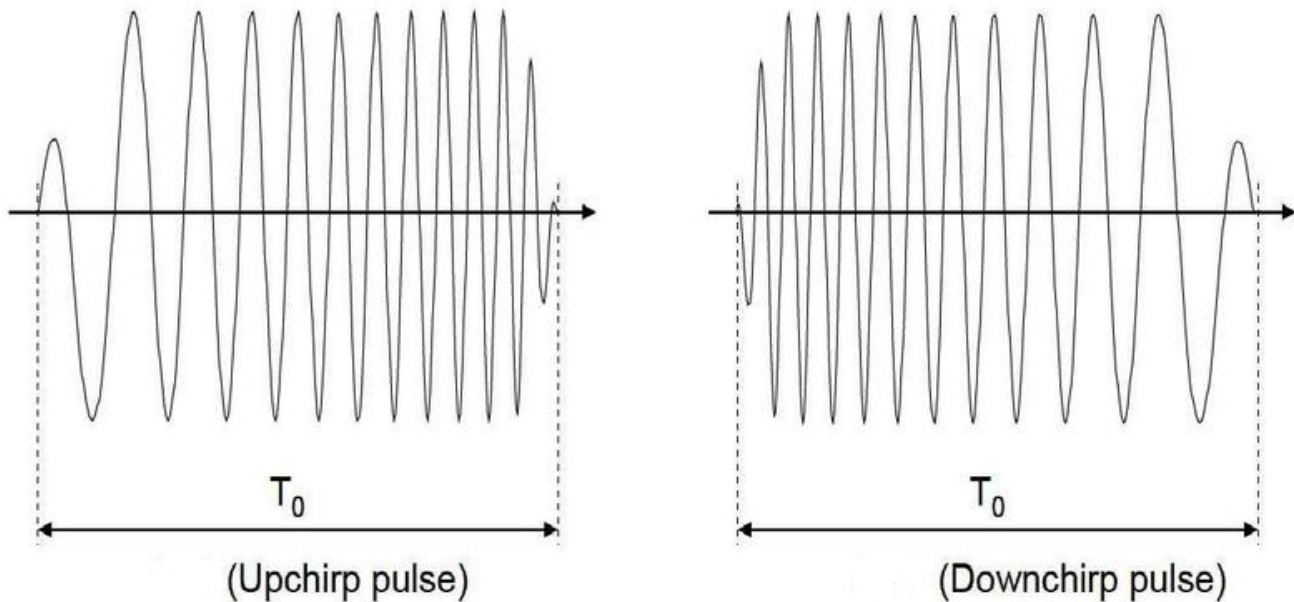
Το DSSS έχει ευρεία χρήση σε εφαρμογές επικοινωνίας δεδομένων. Για να λειτουργήσει όμως σωστά με απαιτητικές εφαρμογές απαιτεί μια εξαιρετικά ακριβή πηγή ρολογιού αναφοράς. Επιπλέον, όσο μεγαλύτερος είναι ο κώδικας ή η ακολουθία διασποράς, τόσο μεγαλύτερος είναι ο χρόνος που απαιτείται από τον δέκτη για να εκτελέσει μια συσχέτιση σε όλο το μήκος της αλληλουχίας κώδικα ή παράλληλα σε διάφορα κομμάτια του σήματος. Αυτό επιβαρύνει συσκευές που δεν έχουν πρόσβαση σε σταθερή πηγή ρεύματος, δεν μπορούν να είναι συνεχώς ενεργές και πρέπει κάθε φορά να συγχρονίζονται για να αποδίδουν σωστά [12].

#### 4.2.5 Chirp spread-spectrum – CSS

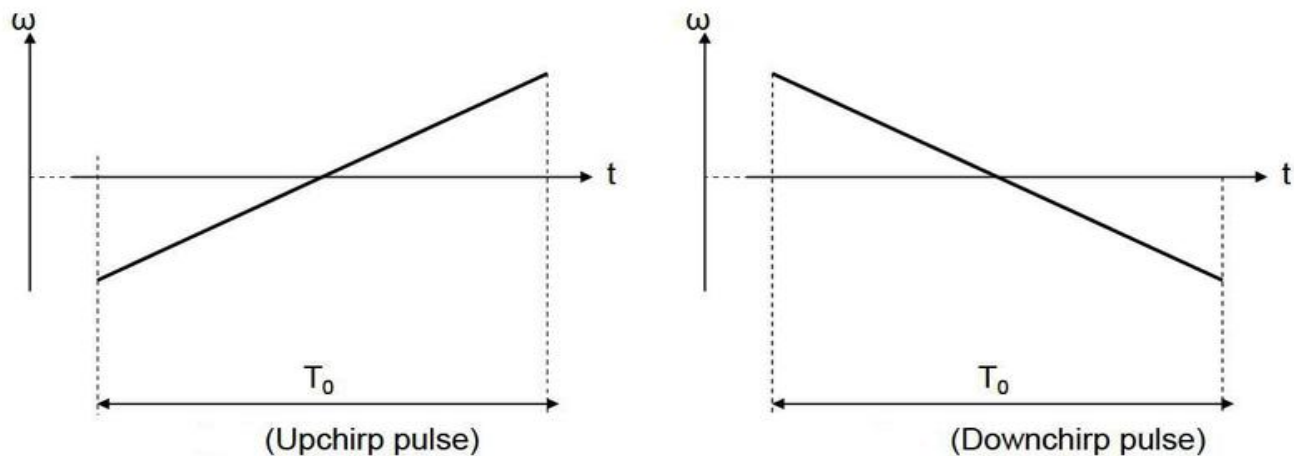
Τα παραπάνω προβλήματα αντιμετωπίζει μια άλλη τεχνική διάδοσης φάσματος η CSS. Η τεχνολογία CSS χρησιμοποιεί παλμούς τιτίβισματα (chirp) με συχνότητα που μεταβάλλεται γραμμικά είτε με αυξανόμενο είτε με φθίνοντα ρυθμό. Οι κατώτερες και οι ανώτερες συχνότητες του παλμού αντιστοιχούν στα όρια του εύρους συχνοτήτων που καθορίζονται από το πρότυπο. Οποιοδήποτε



πακέτο CSS μεταδίδεται με δύο τύπους παλμών: με αυξανόμενη (Upchirp) ή φθίνουσα (Downchirp) συχνότητα, όπως φαίνεται στα σχήματα [16]:



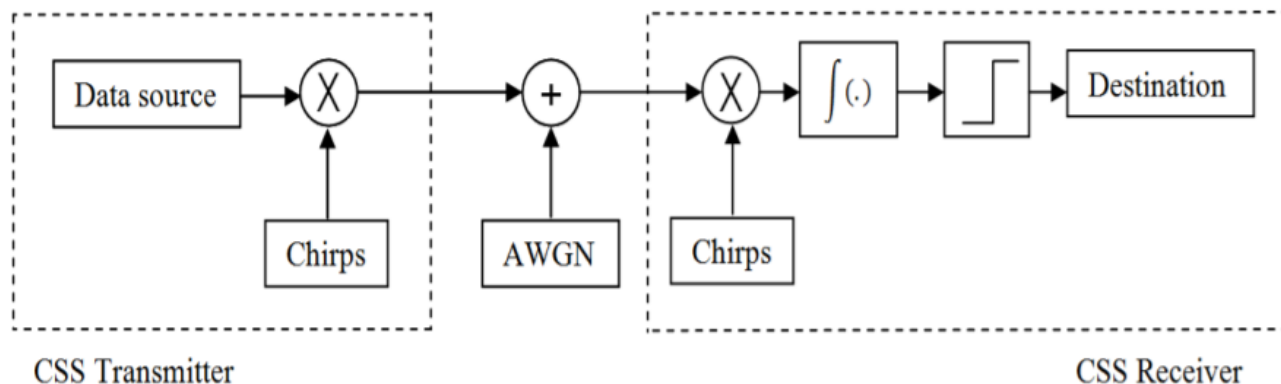
Διάγραμμα 4.4: Παλμός up-chirp και down-chirp στο χρόνο [16]



Διάγραμμα 4.5: Μεταβολή φέρουσας (up-chirp, down-chirp) [16]

Η τεχνολογία αυτή απλώνει το σήμα σε όλο το φάσμα όπως και όλες οι τεχνικές διάδοσης φάσματος που παρουσιάστηκαν στα προηγούμενα κεφάλαια. Αυτό οδηγεί σε αυξημένη αντοχή απέναντι στο θόρυβο του διαύλου και σε παρεμβολές ακόμη και υπό ιδιαίτερα χαμηλή ισχύ εκπομπής. Επιπρόσθετα, παρουσιάζει αντίσταση στο φαινόμενο Doppler. Συνδυαζόμενη με σχήματα ψηφιακής διαμόρφωσης (π.χ. BOK, QPSK, DQPSK), επιτυγχάνει καλύτερο ρυθμό BER [17]. Στον πομπό πολλαπλασιάζεται με ένα παλμό upchirp και στον δέκτη ακολουθείται η αντίστροφη διαδικασία με τον αντίστοιχο παλμό downchirp. Σε αντίθεση με άλλες τεχνολογίες όπως η DSSS δεν προσθέτει

ψευδοτυχαία στοιχεία στο σήμα για να το βοηθήσει να ξεχωρίζει από το θόρυβο του διαύλου, αντίθετα βασίζεται στη γραμμική φύση του παλμού τιτιβίσματος (chirp) [18].



Διάγραμμα 4.6: Πομπός και δέκτης συστήματος CSS [17]

Η διαμόρφωση CSS αναπτύχθηκε για εφαρμογές ραντάρ τη δεκαετία του 1940 και χρησιμοποιήθηκε αρχικά για επικοινωνίες που απαιτούσαν μεγάλη ασφάλεια από υποκλοπές όπως οι στρατιωτικές. Τα τελευταία είκοσι χρόνια αυτή η τεχνική έχει δει αυξημένη χρήση από πολλές εφαρμογές επικοινωνίας δεδομένων λόγω των σχετικά χαμηλών ενεργειακών αναγκών, της αντοχής σε παρεμβολές και στο θόρυβο [12].

#### 4.2.6 Παρουσίαση διαμόρφωσης LoRa

Η διαμόρφωση LoRa χρησιμοποιεί την τεχνική CSS για να αντιμετωπίσει τις αδυναμίες της DSSS και να παρέχει μια τεχνική διαμόρφωσης σήματος χαμηλού κόστους και χαμηλών ενεργειακών απαιτήσεων ώστε να μπορεί να χρησιμοποιηθεί σε εφαρμογές IoT [12].

Η εξάπλωση του φάσματος επιτυγχάνεται με την παραγωγή ενός σήματος παλμού (chirp) που ποικίλλει συνεχώς σε συχνότητα. Ένα πλεονέκτημα αυτής της μεθόδου είναι ότι οι διαφορές (offsets) χρονισμού και συχνότητας μεταξύ πομπού και δέκτη είναι ισοδύναμες, μειώνοντας σημαντικά την πολυπλοκότητα του σχεδιασμού του δέκτη. Το εύρος ζώνης αυτού του παλμού είναι ισοδύναμο με το φασματικό εύρος ζώνης του σήματος [12].

Η σχέση μεταξύ του επιθυμητού ρυθμού δεδομένων bit (wanted data bit rate), του ρυθμού συμβόλων (symbol rate) και του ρυθμού παλμού (chip rate) για τη διαμόρφωση LoRa μπορεί εκφραστεί ως εξής:

Ορίζουμε τον ρυθμό δεδομένων bit διαμόρφωσης (modulation bit rate),  $R_b$ , ως:

$$R_b = SF * \left[ \frac{1}{2^{SF}} \right] \frac{bits}{sec} \quad (5)$$

SF: παράγοντας εξάπλωσης (spreading factor) [7-12]

BW: εύρος ζώνης διαμόρφωσης (modulation bandwidth) (Hz)

Ορίζουμε την περίοδο συμβόλων (symbol period) ,  $T_s$ , ως:

$$T_s = \frac{2^{SF}}{BW} \text{secs} \quad (6)$$

Έτσι, ο ρυθμός συμβόλων (symbol rate)  $R_s$ , είναι το αντίστροφο του  $T_s$ :

$$R_s = \frac{1}{T_s} = \frac{BW}{2^{SF}} \frac{\text{symbols}}{\text{sec}} \quad (7)$$

Τέλος μπορούμε να ορίσουμε τον ρυθμό chip (chip rate),  $R_c$ , ως:

$$R_c = R_s * 2^{SF} \frac{\text{chips}}{\text{sec}} \quad (8)$$

Στην επόμενη σχέση παρατηρούμε ότι στην διαμόρφωση LoRa ένα chip αποστέλλεται ανά sec ανά Hz του εύρος ζώνης.

$$R_c = R_s * 2^{SF} = \frac{BW}{2^{SF}} * 2^{SF} = BW \quad (9)$$

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

Μπορούμε να ορίσουμε τον ονομαστικό ρυθμό μετάδοσης bit του σήματος δεδομένων (nominal bit rate of the data signal) ως:

$$R_b = SF * \frac{[1+CR]}{\left[\frac{2^{SF}}{BW}\right]} \quad (10)$$

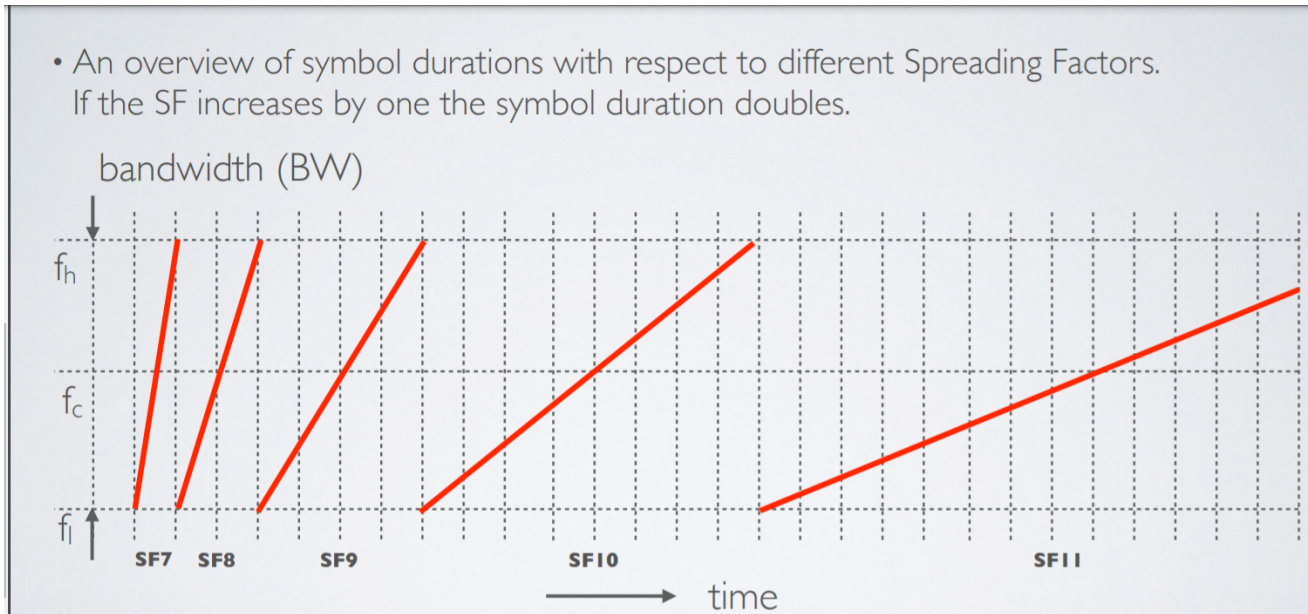
SF: παράγοντας εξάπλωσης (spreading factor) [7-12]

CR = ρυθμός κώδικα (code rate) [1-4]

BW: εύρος ζώνης διαμόρφωσης (modulation bandwidth) (Hz)

Η τελευταία σχέση μας δείχνει ότι αν αυξήσουμε το εύρος ζώνης(BW) αυξάνεται και ο ρυθμός μετάδοσης bit ( $R_b$ ). Αν αυξήσουμε τον παράγοντα εξάπλωσης (SF) τότε μειώνεται. Σε κάθε περίπτωση τα υπόλοιπα μεγέθη παραμένουν σταθερά [19].

Η σχέση (6) μας δείχνει ότι όταν αυξάνεται το εύρος ζώνης τότε μειώνεται η περίοδος συμβόλου και όταν αυξάνεται ο παράγοντας εξάπλωσης τότε αυξάνεται. Πιο συγκεκριμένα αν αυξηθεί κατά 1 ο παράγοντας εξάπλωσης τότε διπλασιάζεται η διάρκεια συμβόλου, υποδιπλασιάζεται ο ρυθμός μετάδοσης bit και αυξάνεται ο χρόνος που είναι ενεργή η κεραία (Time on Air -ToA). Αυτό μεταφράζεται σε μεγαλύτερη κατανάλωση ενέργειας αλλά και μεγαλύτερη εμβέλεια [19].



Διάγραμμα 4.7: Μεταβολή SF [19]

Μπορούμε να πούμε ότι οι συσκευές LoRa χρησιμοποιούν υψηλότερο συντελεστή διασποράς όταν το σήμα είναι αδύναμο ή υπάρχουν παρεμβολές και οδηγούνται σε μεγαλύτερο Time on Air (ToA). Εάν μια τερματική συσκευή βρίσκεται αρκετά μακριά από μια πύλη, το σήμα εξασθενεί και επομένως χρειάζεται υψηλότερο συντελεστή διασποράς[19].

#### 4.2.7 Ζώνες συχνότητων LoRa

Το LoRaWAN λειτουργεί σε ζώνες συχνότητων sub-gigahertz οι οποίες διαφέρουν από περιοχή σε περιοχή λόγω κανονιστικών απαιτήσεων.

Το LoRa ορίζει δέκα κανάλια για την Ευρώπη. Τα 8 από αυτά είναι πολλαπλού ρυθμού δεδομένων από 250 bps έως 5,5 kbps. Ένα κανάλι μπορεί να λειτουργεί με υψηλότερο ρυθμό μετάδοσης δεδομένων με ταχύτητα 11 kbps. Ένα κανάλι είναι τύπου FSK στα 50 kbps. Η μέγιστη επιτρεπόμενη ισχύς είναι +14 dBm.

Το LoRa στη Βόρεια Αμερική ορίζει 64 κανάλια 125 kHz από 902,3 έως 914,9 MHz. Υπάρχουν επιπλέον οκτώ κανάλια ανερχόμενης ζεύξης 500 KHz σε βήματα των 1,6 MHz από 903 MHz έως 914 MHz. Τα οκτώ κανάλια κατερχόμενης ζεύξης έχουν πλάτος 500 kHz ξεκινώντας από τα 923,3 MHz έως τα 927,5 MHz. Η μέγιστη ισχύς εξόδου για τη Βόρεια Αμερική είναι +30 dBm [20].

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

Πίνακας 4.1: Χαρακτηριστικά δικτύων LoRa σε διάφορες γεωγραφικές περιοχές [20]

Ο μέγιστος κύκλος λειτουργίας, (duty cycle) που ορίζεται ως το μέγιστο ποσοστό του χρόνου κατά τον οποίο μια τερματική συσκευή μπορεί να καταλάβει ένα κανάλι, αποτελεί βασικό περιορισμό για δίκτυα που λειτουργούν σε μη αδειοδοτημένες συχνότητες. Επομένως, η επιλογή του καναλιού πρέπει να υλοποιηθεί με ψευδοτυχαίο τρόπο για κάθε μετάδοση και να συμμορφώνεται με το μέγιστο duty cycle. Για παράδειγμα, το duty cycle είναι 1% για τερματικές συσκευές στην Ευρώπη (EU 868) [26].

#### 4.2.8 Ιδιότητες Διαμόρφωσης LoRa

**Εύρος ζώνης με δυνατότητα κλιμάκωσης:** Η διαμόρφωση LoRa είναι επεκτάσιμη τόσο σε εύρος ζώνης όσο και σε συχνότητα. Μπορεί να χρησιμοποιηθεί για εφαρμογές στενής ή ευρείας ζώνης με λίγες μόνο προσαρμογές.

**Σταθερή περιβάλλουσα / Χαμηλή ισχύς:** Παρόμοια με το FSK, το LoRa είναι ένα σχήμα διαμόρφωσης σταθερής περιβάλλουσας που σημαίνει ότι το ίδιο χαμηλού κόστους και υψηλής απόδοσης χαμηλής ισχύος μπορούν να επαναχρησιμοποιηθούν οι ίδιες βαθμίδες χαμηλού κόστους / χαμηλής ισχύος και υψηλής απόδοσης. Χάρης στο κέρδος επεξεργασίας που σχετίζεται με το LoRa, η ισχύς εξόδου του πομπού μπορεί να μειωθεί σε σύγκριση με μια συμβατική σύνδεση FSK διατηρώντας τον ίδιο ή χαμηλότερο κόστος.

**Υψηλή στιβαρότητα:** Η διαμόρφωση LoRa είναι ιδιαίτερα ανθεκτική σε παρεμβολές.

**Ανθεκτικό σε πολλαπλές διαδρομές / ξεθώριασμα:** Ο παλμός (chirp) είναι σχετικά ευρυζωνικός και έτσι το LoRa προσφέρει ανοσία σε πολλαπλές διαδρομές και εξασθένηση, καθιστώντας το ιδανικό για χρήση σε αστικά και προαστιακά περιβάλλοντα, όπου κυριαρχούν και οι δύο μηχανισμοί.

**Ανθεκτικό σε Doppler:** Η μετατόπιση Doppler προκαλεί μια μικρή μετατόπιση συχνότητας στον παλμό LoRa που εισάγει μια σχετικά αμελητέα μετατόπιση στον άξονα χρόνου του σήματος της ζώνης βάσης. Αυτή η ανοχή μετατόπισης συχνότητας μετριάζει την απαίτηση για πηγές ρολογιού αναφοράς στενής ανοχής.

**Ικανότητα μεγάλης εμβέλειας:** Για σταθερή ισχύ εξόδου η απόδοση της σύνδεσης του LoRa υπερβαίνει αυτή του συμβατικού FSK. Όταν λαμβάνεται σε συνδυασμό με την αποδεδειγμένη αντοχή σε παρεμβολές και μηχανισμούς εξασθένησης μπορεί να μεταφραστεί σε τουλάχιστον τετραπλασιασμό της εμβέλειας.

**Ενισχυμένη χωρητικότητα δικτύου:** Η διαμόρφωση LoRa χρησιμοποιεί ορθογώνιους παράγοντες διασποράς που επιτρέπουν σε πολλαπλά σήματα με διαφορετικούς παράγοντες διασποράς να μεταδίδονται ταυτόχρονα στο ίδιο κανάλι. Τα διαμορφωμένα σήματα σε διαφορετικούς παράγοντες διασποράς εμφανίζονται ως θόρυβος στον δέκτη και αντιμετωπίζονται κατάλληλα.

**Εύρος / Εντοπισμός:** Μια εγγενής ιδιότητα του LoRa είναι η ικανότητα γραμμικής διάκρισης μεταξύ σφαλμάτων συχνότητας και χρόνου. Το LoRa είναι η ιδανική διαμόρφωση για εφαρμογές ραντάρ και εντοπισμού ακόμα και σε πραγματικό χρόνο [12].

## 4.3 LoRaWAN

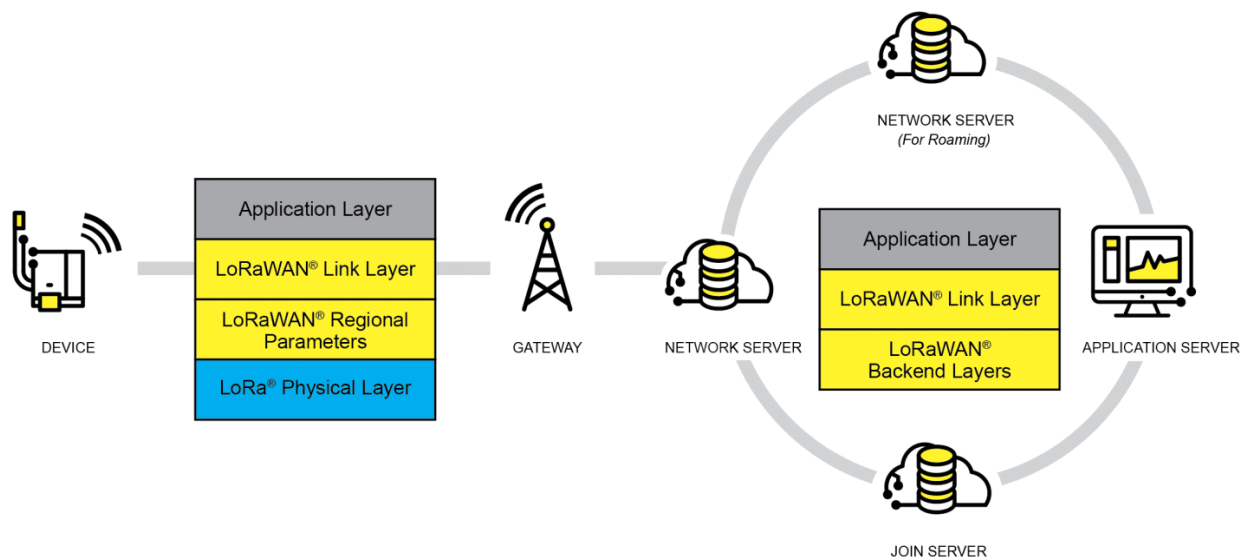
### 4.3.1 Εισαγωγή

Το LoRaWAN είναι ένα πρωτόκολλο επιπέδου MAC που εφαρμόζεται συνήθως συνδυαστικά με τη διαμόρφωση LoRa στο φυσικό επίπεδο. Παρέχει το λογισμικό που ορίζει τον τρόπο με τον οποίο οι συσκευές χρησιμοποιούν το hardware του LoRa, για παράδειγμα πότε μεταδίδουν και ποια είναι η μορφή των μηνυμάτων. Το πρωτόκολλο LoRaWAN είναι ανοιχτό (open) και αναπτύσσεται από τη LoRa Alliance. Η πρώτη προδιαγραφή LoRaWAN κυκλοφόρησε τον Ιανουάριο του 2015 [21]. Στον παρακάτω πίνακα παρουσιάζονται οι εκδόσεις που έχουν κυκλοφορήσει μέχρι σήμερα:

Version	Release date
1.0	January 2015
1.0.1	February 2016
1.0.2	July 2016
1.1	October 2017
1.0.3	July 2018
1.0.4	October 2020

Πίνακας 4.2: Εκδόσεις LoRaWAN [21]

Χρησιμοποιείται για υλοποίηση δικτύων LPWAN και έχει σχεδιαστεί για να συνδέει ασύρματα συσκευές που λειτουργούν με μπαταρία στο διαδίκτυο σε περιφερειακά, εθνικά ή παγκόσμια δίκτυα και καλύπτει βασικές απαιτήσεις του Διαδικτύου των Πραγμάτων (IoT), όπως αμφίδρομη επικοινωνία, ασφάλειας από άκρο σε άκρο, κινητικότητας και γεωγραφικό εντοπισμό [22].

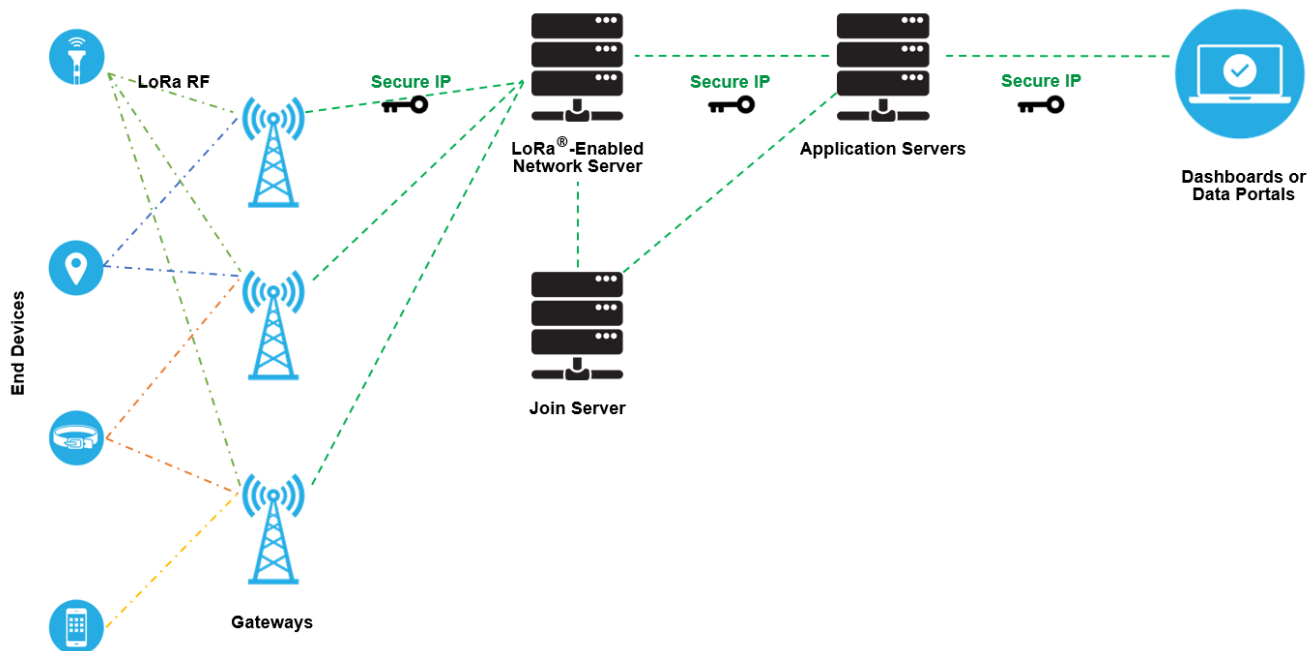


Σχήμα 4.2: Αρχιτεκτονική δικτύου LoRaWAN [22]



### 4.3.2 Αρχιτεκτονική δικτύων LoRaWAN

Η αρχιτεκτονική ενός δικτύου LoRaWAN αναπτύσσεται με τοπολογία τύπου star-of-stars στην οποία οι πύλες (gateways) αναμεταδίδουν μηνύματα μεταξύ τερματικών συσκευών (end nodes) και κεντρικού διακομιστή δικτύου (Network Server). Οι πύλες συνδέονται με τον διακομιστή δικτύου μέσω τυπικών συνδέσεων IP και απλώς μετατρέπουν πακέτα RF σε πακέτα IP και αντίστροφα. Κάθε πύλη μπορεί να εξυπηρετεί πολλές τερματικές συσκευές ταυτόχρονα. Η ασύρματη επικοινωνία εκμεταλλεύεται τα χαρακτηριστικά μεγάλης εμβέλειας του φυσικού στρώματος LoRa, επιτρέποντας μια σύνδεση single-hop μεταξύ της τερματικής συσκευής και μιας ή πολλών πυλών. Όλες οι λειτουργίες είναι ικανές για αμφίδρομη επικοινωνία και υπάρχει υποστήριξη για ομάδες διευθυνσιοδότησης πολλαπλής εκπομπής για την αποτελεσματική χρήση του φάσματος κατά τη διάρκεια εργασιών όπως οι αναβαθμίσεις Over-The-Air (FOTA) ή άλλα μηνύματα μαζικής διανομής



[22].

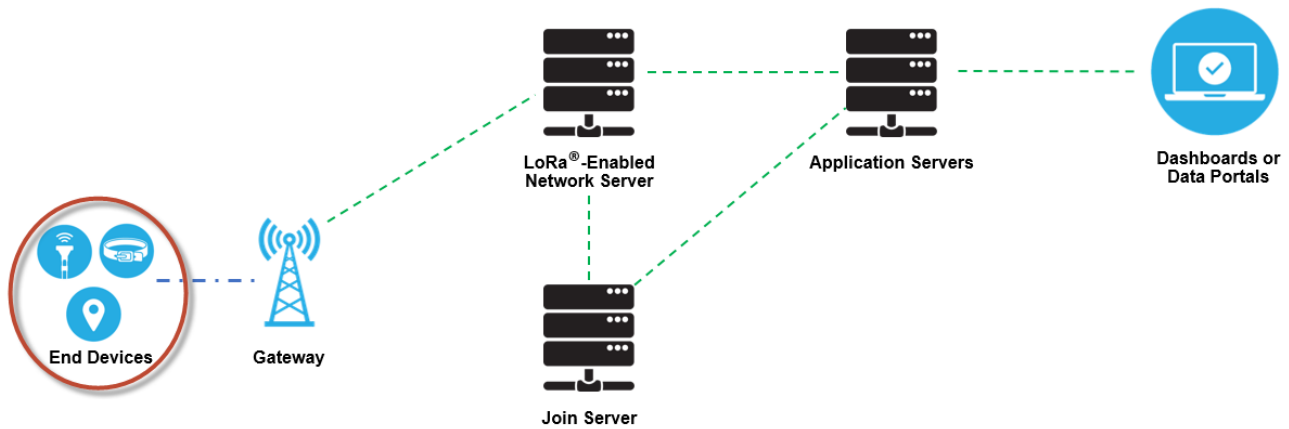
Σχήμα 4.3: Αρχιτεκτονική δικτύου LoRaWAN [11]

#### 4.3.2.1 Τερματικές συσκευές – End devices/nodes

Μια τερματική συσκευή με δυνατότητα LoRaWAN είναι συνήθως ένας αισθητήρας που συνδέεται ασύρματα σε ένα δίκτυο LoRaWAN μέσω μιας πύλης (gateway) χρησιμοποιώντας τη διαμόρφωση LoRa. Σε κάθε τέτοια κατασκευή της έχουν εκχωρηθεί πολλά μοναδικά αναγνωριστικά. Αυτά τα αναγνωριστικά χρησιμοποιούνται για την ασφαλή ενεργοποίηση και διαχείριση της συσκευής, για τη διασφάλιση της ασφαλούς μεταφοράς πακέτων μέσω ενός ιδιωτικού ή δημόσιου δικτύου και για την παράδοση κρυπτογραφημένων δεδομένων στο Cloud [11].



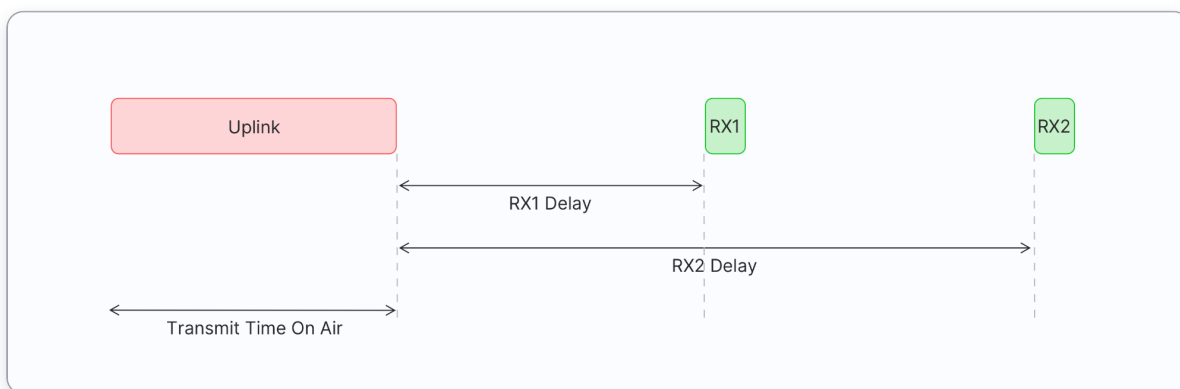
Η προδιαγραφή LoRaWAN ορίζει τρεις τύπους συσκευών: Class A, Class B και Class C. Όλες οι συσκευές LoRaWAN πρέπει να υλοποιούν την Class A αφού οι υπόλοιπες είναι επεκτάσεις της. Όλες οι κατηγορίες συσκευών υποστηρίζουν αμφίδρομη επικοινωνία (uplink και downlink) [11].



Σχήμα 4.4: Τερματικές συσκευές (end devices/nodes) [11]

## Class A

Η επικοινωνία κλάσης A ξεκινά πάντα από την τερματική συσκευή, η οποία μπορεί να στείλει ένα μήνυμα uplink ανά πάσα στιγμή. Μόλις ολοκληρωθεί η ανοδική μετάδοση (uplink), η συσκευή ανοίγει δύο σύντομα παράθυρα κατερχόμενης λήψης (downlink). Υπάρχει μια καθυστέρηση μεταξύ του τέλους της μετάδοσης uplink και της έναρξης των παραθύρων downlink (RX1 και RX2 αντίστοιχα). Εάν ο διακομιστής δικτύου δεν ανταποκρίνεται κατά τη διάρκεια αυτών των δύο παραθύρων λήψης, η επόμενη μετάδοση downlink θα είναι μετά την επόμενη μετάδοση uplink. Ο διακομιστής δικτύου μπορεί να ανταποκριθεί κατά το πρώτο παράθυρο λήψης (RX1) ή κατά το δεύτερο παράθυρο λήψης (RX2), αλλά δεν χρησιμοποιεί και τα δύο παράθυρα [21].

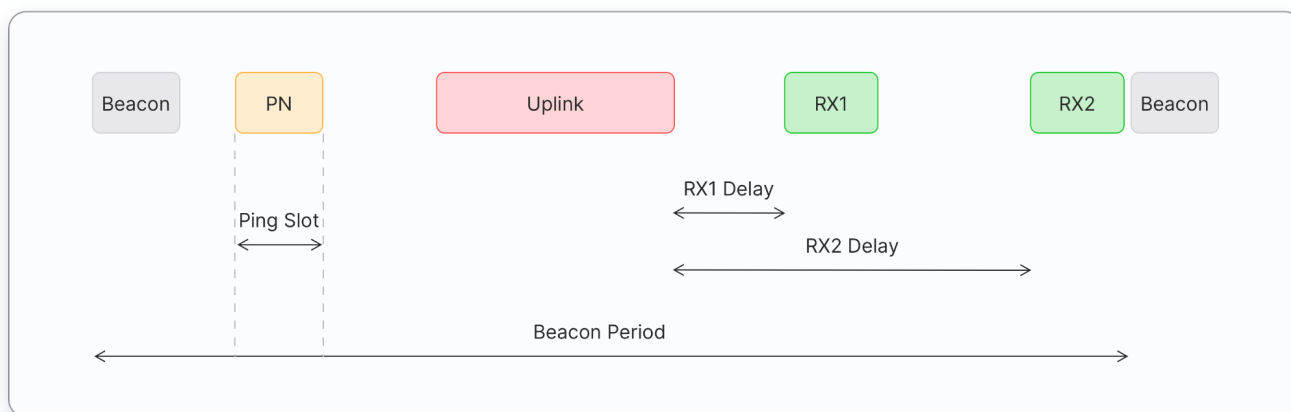


Σχήμα 4.5: Συσκευές Class A [21]

Οι συσκευές κλάσης A συχνά τροφοδοτούνται με μπαταρία αφού έχουν τη χαμηλότερη κατανάλωση ενέργειας και περνούν τον περισσότερο χρόνο σε κατάσταση ύπνου (sleep mode). Συνήθως διατηρούν μεγάλα διαστήματα μεταξύ των uplinks και έχουν υψηλή καθυστέρηση μεταφοράς (latency) [21].

## Class B

Εκτός από τα παράθυρα λήψης που εκκινούν από την συσκευή όπως αναλύθηκε για τις συσκευές κλάσης A, οι συσκευές κλάσης B ανοίγουν παράθυρα προγραμματισμένης λήψης για τη λήψη μηνυμάτων downlink από τον Network Server. Χρησιμοποιώντας χρονικά συγχρονισμένα μηνύματα “φάρους” (beacons) που μεταδίδονται από την πύλη, οι συσκευές ανοίγουν περιοδικά παράθυρα λήψης. Ο χρόνος μεταξύ δύο beacons είναι γνωστός ως περίοδος beacons. Η συσκευή ανοίγει «υποδοχές ring» κατερχόμενης ζεύξης σε προγραμματισμένες ώρες για λήψη μηνυμάτων downlink από τον Network Server. Οι συσκευές κλάσης B ανοίγουν επίσης παράθυρα λήψης μετά την αποστολή μιας σύνδεσης uplink, όπως μπορείτε να δείτε παρακάτω:

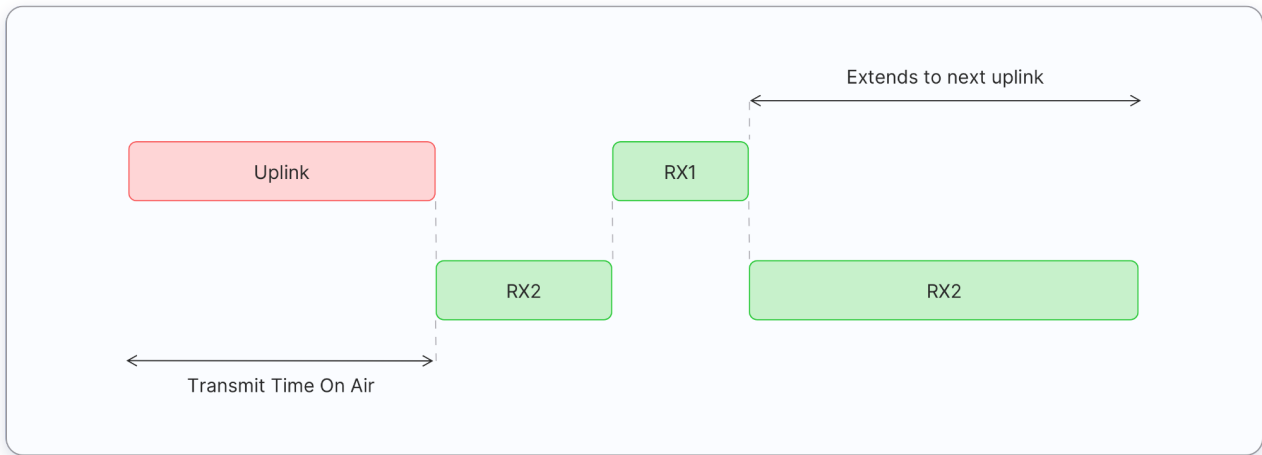


Σχήμα 4.6: Συσκευές Class B [21]

Οι τερματικές συσκευές κλάσης B έχουν χαμηλότερο latency από τις τερματικές συσκευές κλάσης A επειδή είναι προσβάσιμες σε προσυμφωνημένους χρόνους και δεν χρειάζεται να στείλουν μια ανοδική σύνδεση για να λάβουν μια κατερχόμενη σύνδεση. Η διάρκεια ζωής της μπαταρίας είναι μικρότερη από τις συσκευές κλάσης A, επειδή η συσκευή ξοδεύει περισσότερο χρόνο σε ενεργή λειτουργία κατά τη διάρκεια των beacon και των υποδοχών ring [21].

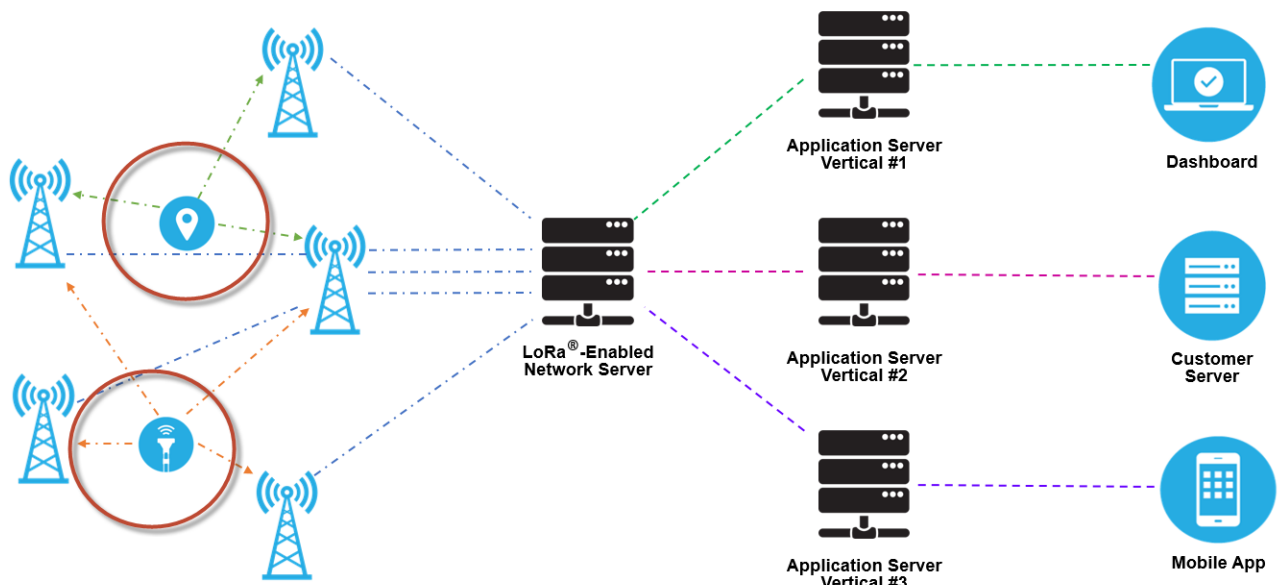
## Class C

Οι συσκευές κλάσης C επεκτείνουν την κλάση A διατηρώντας τα παράθυρα λήψης ανοιχτά εκτός από τις χρονικές στιγμές στις οποίες δεν εκπέμπουν. Αυτό επιτρέπει επικοινωνία χαμηλής καθυστέρησης μεταφοράς (latency) αλλά οδηγεί σε αρκετά μεγαλύτερη κατανάλωση ενέργειας σε σύγκριση με αυτή της κλάσης A [21].



Σχήμα 4.7: Συσκευές Class C [21]

### 4.3.2.2 Πύλες LoRaWAN – Gateways

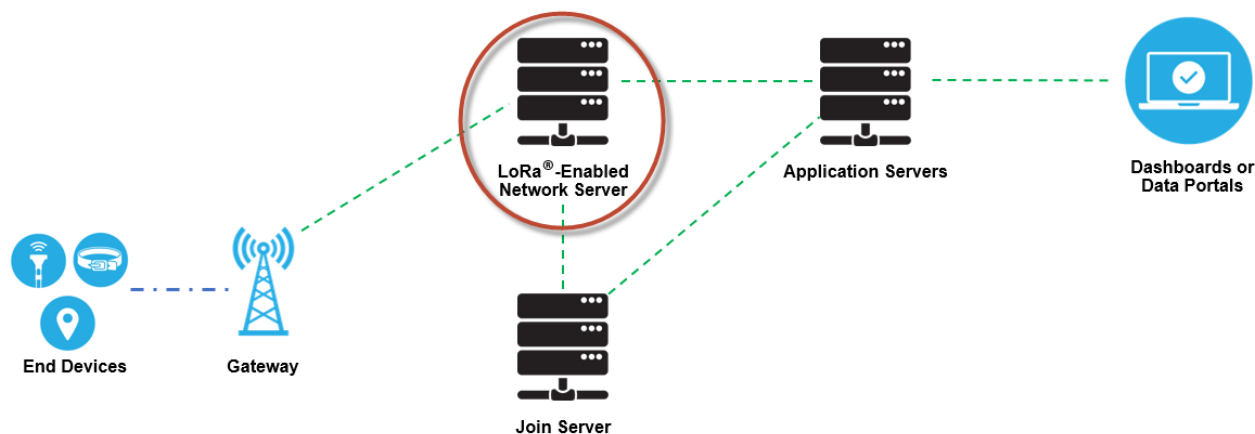


Σχήμα 4.8: Πύλες (gateways) [11]

Η αποστολή μιας πύλης LoRaWAN είναι να λαμβάνει μηνύματα με διαμόρφωση LoRa από οποιαδήποτε τερματική συσκευή (end node) που βρίσκεται στην εμβέλεια της και να τα προωθεί στον Network Server με τον οποίο είναι συνδεδεμένη μέσω ενός δικτύου IP. Δεν υπάρχει σταθερή συσχέτιση μεταξύ μιας τερματικής συσκευής και μιας συγκεκριμένης πύλης. Κάθε πακέτο uplink που αποστέλλεται από την τερματική συσκευή θα λαμβάνεται από όλες τις πύλες που είναι προσβάσιμες. Αυτή η διάταξη μειώνει σημαντικά το ποσοστό σφαλμάτων (καθώς οι πιθανότητες τουλάχιστον μία πύλη να λάβει το μήνυμα είναι πολύ υψηλές), μειώνει σημαντικά την επιβάρυνση της μπαταρίας για κινητούς αισθητήρες και προσφέρει την δυνατότητα γεωγραφικού εντοπισμού [11].

Η κίνηση IP από μια πύλη προς τον Network Server μπορεί να υλοποιηθεί μέσω Wi-Fi, ενσύρματου Ethernet ή μέσω σύνδεσης κινητής τηλεφωνίας. Οι πύλες LoRaWAN λειτουργούν εξ ολοκλήρου στο φυσικό επίπεδο και ουσιαστικά είναι προωθητές μηνυμάτων LoRa. Ελέγχουν μόνο την ακεραιότητα των δεδομένων κάθε εισερχόμενου μηνύματος. Εάν η ακεραιότητα δεν είναι άθικτη, δηλαδή εάν το CRC είναι λανθασμένο, το μήνυμα θα απορριφθεί. Εάν είναι σωστό, η πύλη θα την προωθήσει στο Network Server μαζί με κάποια μεταδεδομένα που περιλαμβάνουν το επίπεδο λήψης RSSI του μηνύματος καθώς και μια προαιρετική χρονική σήμανση. Για τις κατερχόμενες ζεύξεις LoRaWAN, μια πύλη εκτελεί αιτήματα μετάδοσης που προέρχονται από τον Network Server χωρίς να εξετάζει το περιεχόμενο των μηνυμάτων. Δεδομένου ότι πολλές πύλες μπορούν να λάβουν το ίδιο μήνυμα LoRa από την ίδια τερματική συσκευή, ο Network Server διαγράφει όλα τα αντίγραφα και επιλέγει αυτή με το καλύτερο RSSI για να αποστείλει το μήνυμα downlink που συνήθως είναι η πλησιέστερη στην συσκευή [11].

#### 4.3.2.3 Network Server



Σχήμα 4.9: Network Server [11]

Ο διακομιστής δικτύου LoRaWAN (Network Server) διαχειρίζεται ολόκληρο το δίκτυο. Ελέγχει δυναμικά τις παραμέτρους δικτύου, δημιουργεί ασφαλείς συνδέσεις AES 128-bit για τη μεταφορά δεδομένων από τις τερματικές συσκευές στις εφαρμογές στο cloud και ελέγχει την κυκλοφορία που ρέει από την τερματική συσκευή στον Network Server. Ο διακομιστής δικτύου διασφαλίζει την αυθεντικότητα κάθε αισθητήρα στο δίκτυο και την ακεραιότητα κάθε μηνύματος. Πρέπει να τονιστεί ότι δεν μπορεί να αποκτήσει πρόσβαση στα δεδομένα της εφαρμογής [11].

Γενικά, όλοι οι διακομιστές δικτύου μοιράζονται τις ακόλουθες δυνατότητες:

Έλεγχος διεύθυνσης συσκευής.

Έλεγχος ταυτότητας πλαισίων και διαχείριση μετρητή πλαισίων.

Επιβεβαιώσεις ληφθέντων μηνυμάτων.

Προσαρμογή ρυθμών δεδομένων χρησιμοποιώντας το πρωτόκολλο ADR.

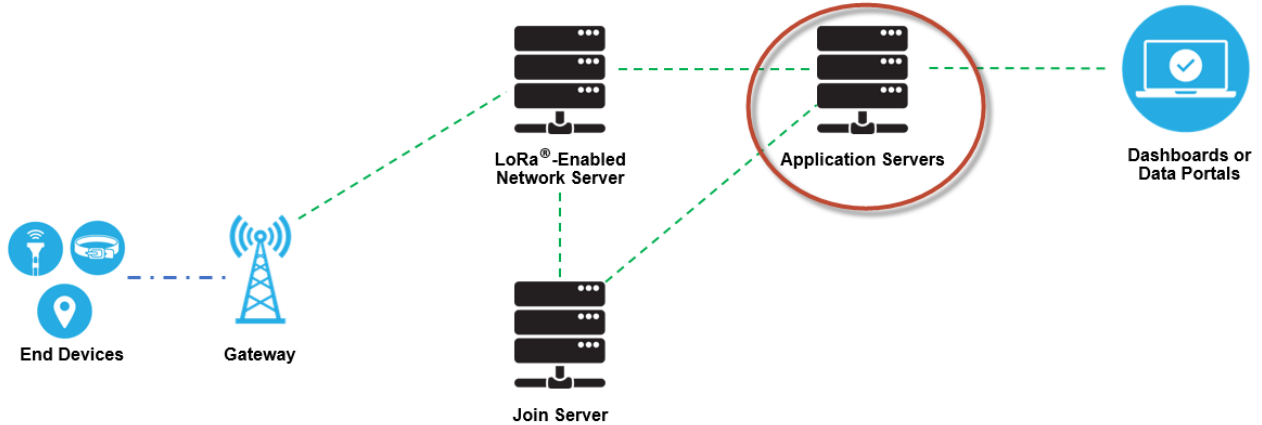
Απάντηση σε όλα τα αιτήματα επιπέδου MAC.

Προώθηση ωφέλιμων μηνυμάτων uplink στους κατάλληλους διακομιστές εφαρμογών.

Τοποθέτηση σε ουρά για μετάδοση των μηνυμάτων downlink που προέρχονται από οποιονδήποτε

διακομιστή εφαρμογών σε οποιαδήποτε συσκευή συνδεδεμένη στο δίκτυο. Προώθηση μηνυμάτων Join Request και Join Accept μεταξύ των συσκευών και του Join Server [11].

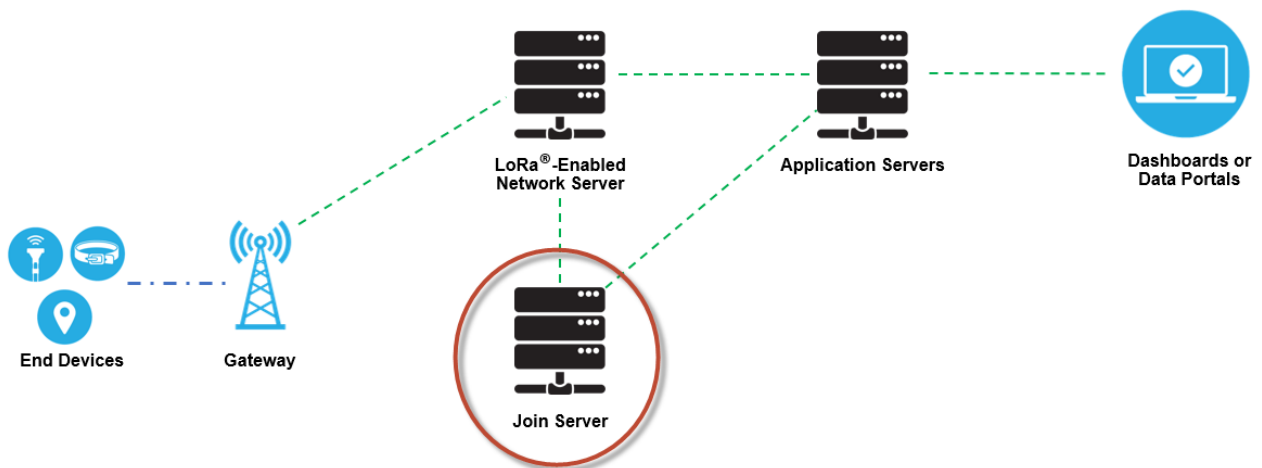
#### 4.2.3.4 Application Server



Σχήμα 4.10: Application Server [11]

Ο διακομιστής εφαρμογών (Application Server) επεξεργάζεται μηνύματα δεδομένων που λαμβάνονται από τις τερματικές συσκευές. Επίσης, δημιουργεί όλα τα downlink μηνύματα του επιπέδου εφαρμογής και τα στέλνει στις συνδεδεμένες τερματικές συσκευές μέσω του Network Server. Ένα δίκτυο LoRaWAN μπορεί να έχει περισσότερους από έναν διακομιστές εφαρμογών. Τα δεδομένα που συλλέγονται μπορούν να υποστούν επεξεργασία με τρόπους όπως η μηχανική μάθηση και η τεχνητή νοημοσύνη για την επίλυση επιχειρηματικών προβλημάτων [21]. Η επικοινωνία του application και του Network Server γίνεται μέσω δικτύου IP. Μπορεί να βρίσκονται στο ίδιο μηχάνημα ή και όχι,

#### 4.2.3.5 Join Server



Σχήμα 4.11: Join Server [11]

Ο διακομιστής σύνδεσης (Join Server) διαχειρίζεται τη διαδικασία ενεργοποίησης over-the-air για τις τερματικές συσκευές που θα προστεθούν στο δίκτυο. Ο Join Server περιέχει τις πληροφορίες που απαιτούνται για την επεξεργασία πλαισίων uplink Join Request και τη δημιουργία των πλαισίων downlink Join Accept. Σηματοδοτεί στον Network Server ποιος Application Server πρέπει να συνδεθεί στην τερματική συσκευή και παράγει τα απαραίτητα κλειδιά κρυπτογράφησης. Επικοινωνεί το Network Session Key της συσκευής στον Network Server και το Application Session Key στον αντίστοιχο Application Server [11]. Ο Join Server εισήχθη με την έκδοση LoRaWAN v1.1. και υπάρχει και στην έκδοση LoRaWAN v1.0.4 [21].

#### 4.2.4 Ενεργοποίηση τερματικής συσκευής – End device activation

Κάθε τερματική συσκευή για να στείλει και να λάβει μηνύματα από ένα δίκτυο πρέπει πρώτα να έχει εγγραφεί σε αυτό. Αυτή η διαδικασία είναι γνωστή ως ενεργοποίηση (activation). Υπάρχουν δύο διαθέσιμες μέθοδοι ενεργοποίησης:

**Over-The-Air-Activation (OTAA):** Είναι η πιο ασφαλής και ευρέως χρησιμοποιούμενη μέθοδος ενεργοποίησης. Οι συσκευές εκτελούν μια διαδικασία σύνδεσης με το δίκτυο, κατά την οποία εκχωρείται μια δυναμική διεύθυνση συσκευής και διαπραγματεύονται τα κλειδιά ασφαλείας με τη συσκευή.

**Activation By Personalization (ABP):** Συνδέει απευθείας μια τερματική συσκευή σε ένα προεπιλεγμένο δίκτυο, παρακάμπτοντας τη διαδικασία over-the-air και είναι η λιγότερο ασφαλής μέθοδος ενεργοποίησης. Μια τερματική συσκευή που ενεργοποιείται χρησιμοποιώντας τη μέθοδο ABP μπορεί να λειτουργήσει μόνο με ένα μόνο δίκτυο και διατηρεί την ίδια περίοδο λειτουργίας ασφαλείας για όλη τη διάρκεια ζωής της. Η αλλαγή δικτύου πρέπει να γίνει χειροκίνητα ενώ δεν χρησιμοποιείται Join Server.

Η διαδικασία σύνδεσης για τα LoRaWAN 1.0.x και 1.1 είναι ελαφρώς διαφορετική [23]. Στην συνέχεια θα περιγραφεί η διαδικασία σύνδεσης για LoRaWAN 1.0.x και 1.1 χωριστά.

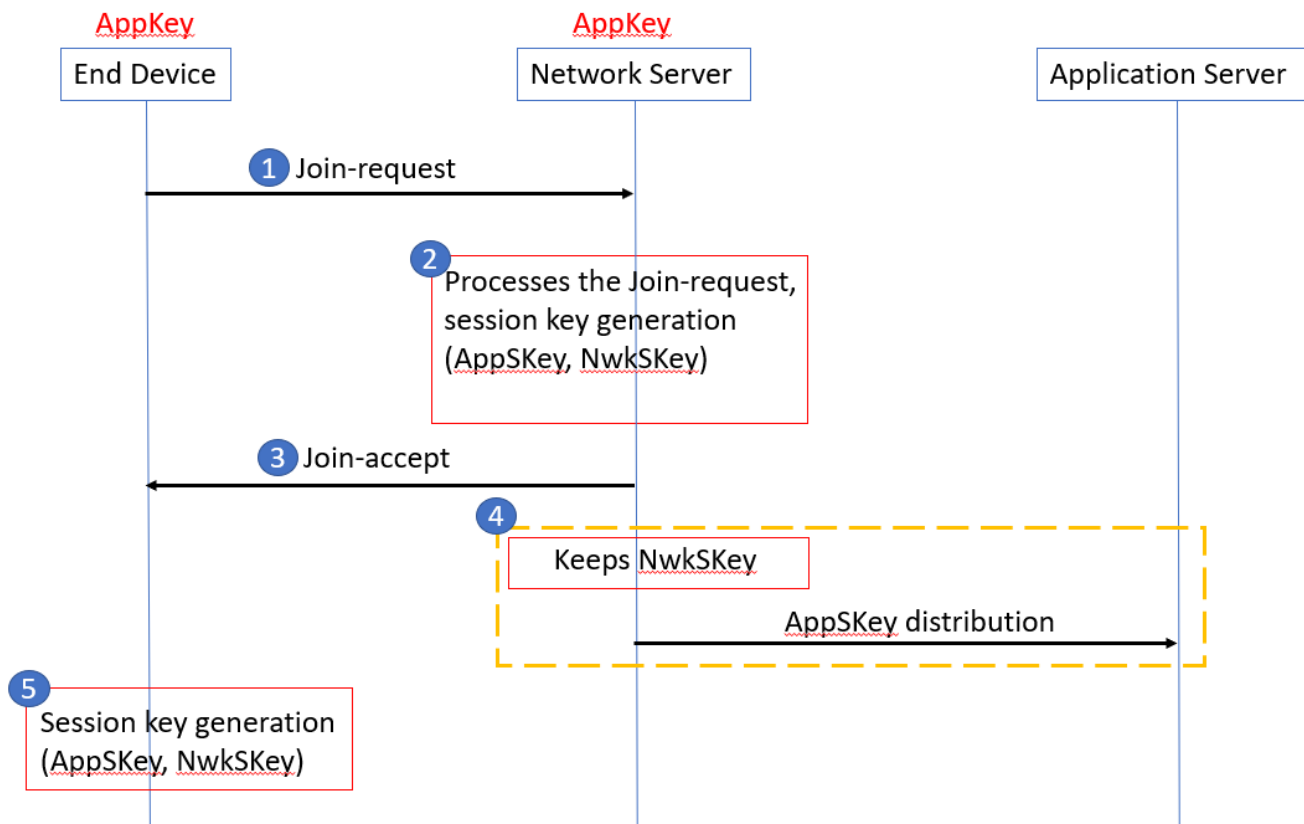
##### 4.2.4.1 Over-The-Air-Activation στο LoRaWAN 1.0.x

Στο LoRaWAN 1.0.x, η διαδικασία σύνδεσης απαιτεί την ανταλλαγή δύο μηνυμάτων MAC μεταξύ της τερματικής συσκευής και του Network Server:

Join Request - από την τερματική συσκευή στον Network Server (downlink)

Join Accept - από τον Network Server στην τερματική συσκευή (uplink)

Πριν από την ενεργοποίηση, το AppEUI, το DevEUI και το AppKey θα πρέπει να έχουν αποθηκευτεί στην τερματική συσκευή. Το AppKey είναι ένα μυστικό κλειδί AES-128 bit γνωστό ως AES root key. Το AppKey θα πρέπει να είναι γνωστό στον Network Server αφού δεν αποστέλλεται ποτέ μέσω του δικτύου. Τα AppEUI και DevEUI, αντίθετα, δεν είναι μυστικά αλλά μπορεί να είναι ορατά σε όλους.



Σχήμα 4.12: Διαδικασία Over-The-Air-Activation στο LoRaWAN 1.0.x [23]

Τα ακόλουθα βήματα περιγράφουν τη διαδικασία Ενεργοποίησης Over-The-Air (OTAA).

### Βήμα 1:

Η διαδικασία εκκινεί πάντα από την τερματική συσκευή (end device) που αποστέλλει ένα μήνυμα Join Request με τα παρακάτω πεδία:

8 bytes	8 bytes	2 bytes
AppEUI	DevEUI	DevNonce

Πίνακας 4.3: Πεδία πληροφορίας μηνύματος Join Request Over-The-Air-Activation στο LoRaWAN 1.0.x [23]

**AppEUI:** Ένα παγκόσμια μοναδικό αναγνωριστικό του χώρου διευθύνσεων IEEE EUI64 που αντιστοιχεί μόνο στην εφαρμογή που θα επεξεργαστεί το πλαίσιο Join Request.

**DevEUI:** Ένα παγκόσμια μοναδικό αναγνωριστικό του χώρου διευθύνσεων IEEE EUI64 που αντιστοιχεί μόνο στην τερματική συσκευή (end device/node).

**DevNonce:** Μια μοναδική και τυχαία τιμή 2 byte που υπολογίζεται από την τερματική συσκευή. Ο Network Server χρησιμοποιεί το DevNonce κάθε τερματικής συσκευής για να παρακολουθεί τα αιτήματά σύνδεσης. Εάν μια τερματική συσκευή στείλει ένα Join Request με ένα DevNonce που χρησιμοποιήθηκε στο παρελθόν (αυτή η κατάσταση είναι γνωστή ως replay attack), ο Network Server το απορρίπτει και δεν επιτρέπει σε αυτήν την τερματική συσκευή να εγγραφεί στο δίκτυο.

Το Message Integrity Code (MIC) υπολογίζεται σε όλα τα πεδία του μηνύματος αιτήματος σύνδεσης χρησιμοποιώντας το AppKey. Στη συνέχεια, το υπολογιζόμενο MIC προστίθεται στο μήνυμα. Το AppKey δεν στέλνεται με το Join Request, το οποίο μεταδίδεται δίχως να έχει κρυπτογραφηθεί.

## Βήμα 2:

Ο Network Server επεξεργάζεται το μήνυμα Join Request, παράγει τα session keys (NwkSKey και AppSKey) και αν η τερματική συσκευή επιτρέπεται να συνδεθεί στο δίκτυο, τότε παράγει το μήνυμα Join Accept με τα παρακάτω πεδία:

3 bytes	3 bytes	4 bytes	1 bytes	1 bytes	16 bytes (optional)
AppNonce	NetID	DevAddr	DLSettings	RXDelay	CFList

Πίνακας 4.4: Πεδία πληροφορίας μηνύματος Join Accept Over-The-Air-Activation στο LoRaWAN 1.0.x [23]

**AppNonce:** Μια τυχαία τιμή ή μοναδικό αναγνωριστικό που παρέχεται από τον Network Server. Χρησιμοποιείται από την τερματική συσκευή για τον υπολογισμό των AppSKey και NwkSKey.

**NetID:** Αποτελείται από το αναγνωριστικό δικτύου (NwkID), τα πιο σημαντικά 7 bit.

**DevAddr:** Μια διεύθυνση 32-bit που εκχωρείται από τον Network Server για την αναγνώριση της τερματικής συσκευής εντός του τρέχοντος δικτύου.

**DLSettings:** Ένα πεδίο 1 byte που αποτελείται από τις ρυθμίσεις που πρέπει να χρησιμοποιεί η τερματική συσκευή για την λειτουργία downlink.

**RxDelay:** Περιέχει την καθυστέρηση μεταξύ TX και RX.

**CFList:** Μια προαιρετική λίστα συχνοτήτων καναλιού για το δίκτυο στο οποίο συνδέεται η τερματική συσκευή. Αυτές οι συχότητες είναι συγκεκριμένες για κάθε περιοχή.

Το Message Integrity Code (MIC) υπολογίζεται σε όλα τα πεδία του μηνύματος Join Accept χρησιμοποιώντας το AppKey. Στη συνέχεια, το υπολογιζόμενο MIC προστίθεται στο μήνυμα. Το AppKey χρησιμοποιείται από τον Network Server για την κρυπτογράφηση του Join Accept με χρήση κρυπτογράφησης AES.

## Βήμα 3:

Ο Network Server στέλνει το κρυπτογραφημένο μήνυμα Join Accept πίσω στην τερματική συσκευή ως κανονική σύνδεση downlink. Αν δεν αποδέχεται την σύνδεση τότε δεν στέλνει κανένα μήνυμα.

## Βήμα 4:

Ο Network Server διατηρεί το NwkSKey και διανέμει το AppSKey στον Application Server.



## Βήμα 5:

Η τερματική συσκευή αποκρυπτογραφεί το μήνυμα Join Accept χρησιμοποιώντας τη λειτουργία κρυπτογράφησης AES. Χρησιμοποιεί το AppKey και το AppNonce για να εξάγει τα NwkSKey και AppSKey. Μετά από αυτό είναι πλέον ενεργοποιημένη στο Δίκτυο.

Μετά την ενεργοποίηση, οι ακόλουθες πρόσθετες πληροφορίες αποθηκεύονται στην τερματική συσκευή.

**DevAddr:** Μια διεύθυνση 32 bit που εκχωρείται από τον Network Server για την αναγνώριση της τερματικής συσκευής εντός του τρέχοντος δικτύου.

**NwkSKey:** Το network session key χρησιμοποιείται από την τερματική συσκευή και τον Network Server για τον υπολογισμό και την επαλήθευση του MIC όλων των μηνυμάτων δεδομένων για τη διασφάλιση της ακεραιότητας του μηνύματος. Το NwkSKey χρησιμοποιείται επίσης για την κρυπτογράφηση και αποκρυπτογράφηση payloads με εντολές MAC.

**AppSKey:** Το application session key χρησιμοποιείται για την κρυπτογράφηση και την αποκρυπτογράφηση των payloads εφαρμογών σε μηνύματα δεδομένων για τη διασφάλιση της εμπιστευτικότητας των μηνυμάτων.

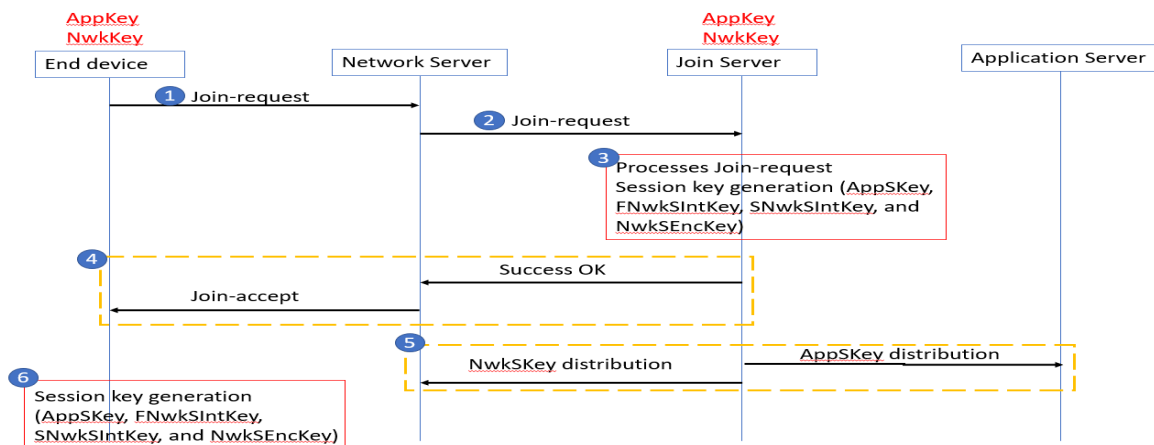
### 4.2.4.2 Over-The-Air-Activation στο LoRaWAN 1.1

Στο LoRaWAN 1.1, η διαδικασία σύνδεσης απαιτεί την ανταλλαγή δύο μηνυμάτων MAC μεταξύ της τερματικής συσκευής και του Network Server:

**Join Request** - από την τερματική συσκευή στον Network Server (downlink)

**Join Accept** - από τον Network Server στην τερματική συσκευή (uplink)

Πριν από την ενεργοποίηση, το AppEUI, το DevEUI, AppKey και NwkKey θα πρέπει να έχουν αποθηκευτεί στην τερματική συσκευή. Τα AppKey και NwkKey είναι μυστικά κλειδιά AES-128 bit γνωστά ως root keys. Τα AppKey, NwkKey και DevEUI θα πρέπει να παρέχονται και στον Join Server του δικτύου στον οποίο πρόκειται να εγγραφεί η τερματική συσκευή. Τα AppKey και NwkKey δεν αποστέλλονται ποτέ μέσω του δικτύου. Τα JoinEUI και DevEUI, αντίθετα, δεν είναι μυστικά αλλά μπορεί να είναι ορατά σε όλους.



Σχήμα 4.13: Διαδικασία Over-The-Air-Activation στο LoRaWAN 1.1 [23]

Τα ακόλουθα βήματα περιγράφουν τη διαδικασία Ενεργοποίησης Over-The-Air (OTAA).

### Βήμα 1:

Η διαδικασία εκκινεί πάντα από την τερματική συσκευή (end device) που αποστέλλει ένα μήνυμα Join Request με τα παρακάτω πεδία:

<b>8 bytes</b>	<b>8 bytes</b>	<b>2 bytes</b>
JoinEUI	DevEUI	DevNonce

Πίνακας 4.5: Πεδία πληροφορίας μηνύματος Join Request Over-The-Air-Activation στο LoRaWAN 1.1 [23]

**JoinEUI:** Ένα παγκόσμια μοναδικό αναγνωριστικό του χώρου διευθύνσεων IEEE EUI64 που αντιστοιχεί στον Join Server που χειρίζεται τα αιτήματά Join Request του δικτύου.

**DevEUI:** Ένα παγκόσμια μοναδικό αναγνωριστικό του χώρου διευθύνσεων IEEE EUI64 που αντιστοιχεί μόνο στην τερματική συσκευή (end device/node).

**DevNonce:** Μια μοναδική και τυχαία τιμή 2 byte που υπολογίζεται από την τερματική συσκευή. Ο Network Server χρησιμοποιεί το DevNonce κάθε τερματικής συσκευής για να παρακολουθεί τα αιτήματά σύνδεσης. Εάν μια τερματική συσκευή στείλει ένα Join Request με ένα DevNonce που χρησιμοποιήθηκε στο παρελθόν (αυτή η κατάσταση είναι γνωστή ως replay attack), ο Network Server το απορρίπτει και δεν επιτρέπει σε αυτήν την τερματική συσκευή να εγγραφεί στο δίκτυο.

Το Message Integrity Code (MIC) υπολογίζεται σε όλα τα πεδία του μηνύματος αιτήματος σύνδεσης χρησιμοποιώντας το NwkKey. Στη συνέχεια, το υπολογιζόμενο MIC προστίθεται στο μήνυμα. Το NwkKey δεν στέλνεται με το Join Request, το οποίο μεταδίδεται δίχως να έχει κρυπτογραφηθεί.

### Βήμα 2:

Ο Network Server προωθεί το μήνυμα στον αντίστοιχο Join Server.

### Βήμα 3:

Ο Join Server επεξεργάζεται το μήνυμα Join Request και αν η συσκευή επιτρέπεται να συνδεθεί στο δίκτυο τότε παράγει τα απαραίτητα session keys (AppSKey, FNwkSIntKey, SNwkSIntKey και NwkSEncKey).

### Βήμα 4:

Ο Network Server παράγει το μήνυμα Join Accept με τα παρακάτω πεδία:

<b>3 bytes</b>	<b>3 bytes</b>	<b>4 bytes</b>	<b>1 bytes</b>	<b>1 bytes</b>	<b>16 bytes (optional)</b>
JoinNonce	NetID	DevAddr	DLSettings	RXDelay	CFList

Πίνακας 4.6: Πεδία πληροφορίας μηνύματος Join Accept Over-The-Air-Activation στο LoRaWAN 1.1 [23]

**JoinNonce:** Μια τιμή μετρητή που παρέχεται από τον Join Server και χρησιμοποιείται από την τερματική συσκευή για την εξαγωγή των FNwkSIntKey, SNwkSIntKey, NwkSEncKey και AppSKey.

**NetID:** Αποτελείται από το αναγνωριστικό δικτύου (NwkID), τα πιο σημαντικά 7 bit.

**DevAddr:** Μια διεύθυνση 32-bit που εκχωρείται από τον Network Server για την αναγνώριση της τερματικής συσκευής εντός του τρέχοντος δικτύου.

**DLSettings:** Ένα πεδίο 1 byte που αποτελείται από τις ρυθμίσεις που πρέπει να χρησιμοποιεί η τερματική συσκευή για την λειτουργία downlink.

**RxDelay:** Περιέχει την καθυστέρηση μεταξύ TX και RX.

**CFList:** Μια προαιρετική λίστα συχνοτήτων καναλιού για το δίκτυο στο οποίο συνδέεται η τερματική συσκευή. Αυτές οι συχότητες είναι συγκεκριμένες για κάθε περιοχή.

Το Message Integrity Code (MIC) υπολογίζεται σε όλα τα πεδία του μηνύματος Join Accept χρησιμοποιώντας το NwkKey (LoRaWAN 1.0 ) ή JSIntKey (LoRaWAN 1.1). Στη συνέχεια, το υπολογιζόμενο MIC προστίθεται στο μήνυμα. Το NwkKey (για Join Request) ή το JSEncKey (για Rejoin request) χρησιμοποιείται από τον Network Server για την κρυπτογράφηση του Join Accept με χρήση κρυπτογράφησης AES.

Ο Network Server στέλνει το κρυπτογραφημένο μήνυμα Join Accept πίσω στην τερματική συσκευή ως κανονική σύνδεση downlink. Αν δεν αποδέχεται την σύνδεση τότε δεν στέλνει κανένα μήνυμα.

#### **Βήμα 5:**

Ο Join Server στέλνει το AppSKey στον Application Server και τα FNwkSIntKey, SNwkSIntKey και NwkSEncKey στον Network Server.

#### **Βήμα 6:**

Η τερματική συσκευή αποκρυπτογραφεί το μήνυμα Join Accept χρησιμοποιώντας τη λειτουργία κρυπτογράφησης AES. Χρησιμοποιεί τα AppKey, NwkKey και JoinNonce για να εξάγει τα AppSKey, FNwkSIntKey, SNwkSIntKey, και NwkSEncKey. Μετά από αυτό είναι πλέον ενεργοποιημένη στο Δίκτυο.

Μετά την ενεργοποίηση, οι ακόλουθες πρόσθετες πληροφορίες αποθηκεύονται στην τερματική συσκευή.

**DevAddr:** Μια διεύθυνση 32 bit που εκχωρείται από τον Network Server για την αναγνώριση της τερματικής συσκευής εντός του τρέχοντος δικτύου.

**NFNwkSIntKey:** Ένα network session key που χρησιμοποιείται από την τερματική συσκευή για τον υπολογισμό του MIC (μερικώς) όλων των μηνυμάτων δεδομένων uplink για τη διασφάλιση της ακεραιότητας του μηνύματος.

**SNwkSIntKey:** Ένα network session key που χρησιμοποιείται από την τερματική συσκευή για τον υπολογισμό του MIC (μερικώς) όλων των μηνυμάτων δεδομένων uplink και τον υπολογισμό του MIC όλων των μηνυμάτων δεδομένων downlink για τη διασφάλιση της ακεραιότητας του μηνύματος.

**NwkSEncKey:** Ένα network session key που χρησιμοποιείται για την κρυπτογράφηση και την αποκρυπτογράφηση των payloads με εντολές MAC των μηνυμάτων δεδομένων uplink και downlink για τη διασφάλιση της εμπιστευτικότητας των μηνυμάτων.

**AppSKey:** Το application session key χρησιμοποιείται για την κρυπτογράφηση και την αποκρυπτογράφηση των payloads εφαρμογών σε μηνύματα δεδομένων για τη διασφάλιση της εμπιστευτικότητας των μηνυμάτων.

#### 4.2.4.3 Activation By Personalisation in LoRaWAN 1.0.x

Το DevAddr και τα NwkSKey και AppSKey αποθηκεύονται απευθείας στην τερματική συσκευή αντί για το DevEUI, το AppEUI και το AppKey. Κάθε τερματική συσκευή θα πρέπει να έχει ένα μοναδικό σύνολο NwkSKey και AppSKey. Τα DevAddr και NwkSKey θα πρέπει να αποθηκευτούν στον Network Server και το AppSKey θα πρέπει να αποθηκευτεί στον application server.



Σχήμα 4.14: Activation By Personalisation in LoRaWAN 1.0.x [23]

#### 4.2.4.4 Activation By Personalisation in LoRaWAN 1.1

Το DevAddr και τα FNwkSIntKey, SNwkSIntKey, NwkSEncKey και AppSKey αποθηκεύονται απευθείας στην τερματική συσκευή αντί για τα DevEUI, JoinEUI, AppKey και NwkKey. Τα DevAddr, FNwkSIntKey, SNwkSIntKey και NwkSEncKey θα πρέπει να αποθηκευτούν στον Network Server και το AppSKey θα πρέπει να αποθηκευτεί στον application server.



Σχήμα 4.15: Activation By Personalisation in LoRaWAN 1.1 [23]

#### 4.2.5 Δομές και τύποι μηνυμάτων LoRaWAN

Οι δύο τύποι μηνυμάτων που χρησιμοποιούνται σε όλες τις εκδόσεις LoRaWAN είναι τα uplink και downlink μηνύματα. Χρησιμοποιούνται για την μεταφορά εντολών MAC και δεδομένων.

**Uplink:** Στέλνονται από τις τερματικές συσκευές στον Network Server χρησιμοποιώντας μία ή περισσότερες πύλες (gateways). Ο Network Server έπειτα τα στέλνει στους σωστούς Application ή Join Servers.

Preamble	PHDR	PHDR_CRC	PHYPayload	CRC
12 symbols	4 bytes	4 bytes	13-235 bytes	2 bytes

Πίνακας 4.7: Δομή μηνύματος uplink [24]

**Preamble;** 12 bytes στην αρχή του μηνύματος

**PHDR:** Επικεφαλίδα φυσικού επιπέδου LoRa

**PHDR\_CRC:** Χρησιμοποιείται για τον έλεγχο ακεραιότητας της επικεφαλίδας

**PHYPayload:** Το πεδίο με τα ωφέλιμα δεδομένα (payload)

**CRC:** Χρησιμοποιείται για τον έλεγχο ακεραιότητας των δεδομένων.

Τα πεδία Preamble, PHDR, PHDR\_CRC και CRC προστίθενται από τον αναμεταδότη LoRa [24].

**Downlink:** Στέλνονται από τον Network Server στην τερματική συσκευή που απευθύνεται χρησιμοποιώντας μόνο μία πύλη. Τα μηνύματα που παράγονται από τους Application και Join Servers προωθούνται στην τερματική συσκευή με αυτό τον τρόπο.

Preamble	PHDR	PHDR_CRC	PHYPayload
12 symbols	4 bytes	4 bytes	13-235 bytes

Πίνακας 4.8: Δομή μηνύματος downlink [24]

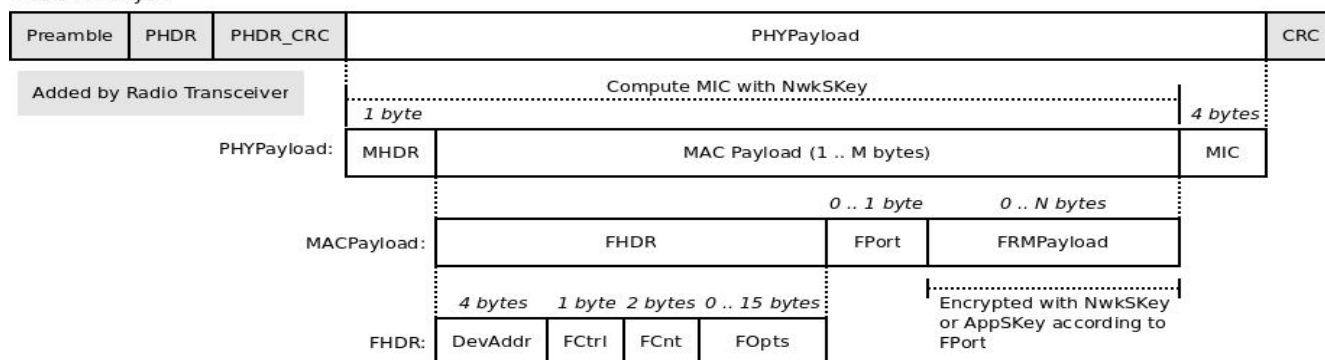
Στις εκδόσεις LoRaWAN 1.0.x και 1.1. υπάρχουν οι παρακάτω τύποι μηνυμάτων:

LoRaWAN 1.0.x	LoRaWAN 1.0.x	Περιγραφή
Join-request	Join-request	Μήνυμα uplink με χρήση στο OTAA activation
Join-accept	Join-accept	Μήνυμα downlink με χρήση στο OTAA activation
Unconfirmed Data Up	Unconfirmed Data Up	Μήνυμα uplink δίχως επιβεβαίωση
Unconfirmed Data Down	Unconfirmed Data Down	Μήνυμα downlink δίχως επιβεβαίωση
Confirmed Data Up	Confirmed Data Up	Μήνυμα uplink με απαίτηση επιβεβαίωσης
Confirmed Data Down	Confirmed Data Down	Μήνυμα downlink με απαίτηση επιβεβαίωσης
RFU	Rejoin-request	1.0.x: Δεσμευμένο για μελλοντική χρήση 1.1: Μήνυμα uplink με χρήση στο OTAA activation (rejoin)
Proprietary	Proprietary	Υλοποίηση μη ορισμένων τύπων μηνυμάτων

Πίνακας 4.9: Τύποι μηνυμάτων LoRaWAN [24]

#### 4.2.6 Περιγραφή μηνύματος MAC δεδομένων

Radio PHY layer:



Σχήμα 4.16: Δομή μηνύματος MAC δεδομένων [25]

**MHDR:** Επικεφαλίδα MAC (MAC header)

**MACPayload:** Ωφέλιμο φορτίο MAC (MAC payload)

**FHDR:** Επικεφαλίδα πλαισίου (Frame header)

**DevAddr:** Device address

**FCtrl:** Έλεγχος πλαισίου (frame control octet)

**FCnt:** Μετρητής πλαισίων (frame counter)

**FOpts:** Εντολές MAC

**FPort:** Καθορισμός τρόπου διαχείρισης μηνυμάτων

**FRMPayload:** Ωφέλιμο φορτίο πλαισίου (Frame payload)

**MIC:** Κώδικας ακεραιότητας μηνύματος (Message integrity code)

Το είδος του μηνύματος καθορίζεται από το πεδίο MType που περιέχεται στην επικεφαλίδα MHDR ως εξής:

MType	Περιγραφή
000	Join Request
001	Join Accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	RFU
111	Proprietary

Πίνακας 4.10: Περιγραφή πεδίου MType [25]

#### 4.2.7 Πλεονεκτήματα και Περιορισμοί του LoRaWAN

Το LoRaWAN δεν είναι κατάλληλο για κάθε περίπτωση χρήσης. Τα πλεονεκτήματά του είναι η μεγάλη εμβέλεια, η χαμηλή ισχύς, το χαμηλό κόστος, το χαμηλό εύρος ζώνης, η μεγάλη κάλυψη και η ασφάλεια που προσφέρει.

Υπάρχουν όμως οι παρακάτω περιορισμοί:

Το payload πρέπει να είναι όσο μικρότερο γίνεται.

Η συχνότητα αποστολής των μηνυμάτων πρέπει να είναι της τάξεως των λεπτών (π.χ. ανά 5 ή 10 λεπτά)

Ο ρυθμός μετάδοσης να είναι ο μεγαλύτερος δυνατός.

Η αποστολή downlink μηνυμάτων να αποφεύγεται ή να περιορίζεται τόσο σε συχνότητα, όσο και σε μέγεθος [27].



## Μέρος Β: Υλοποίηση

### Κεφάλαιο 5: Προετοιμασία της Υλοποίησης

#### 5.1 Εισαγωγή

Σκοπός αυτού του μέρους της εργασίας είναι η υλοποίηση δικτύων LoRaWAN που να υποστηρίζουν την λειτουργία roaming και ο έλεγχος της επιτυχούς επίτευξης αυτού του στόχου.

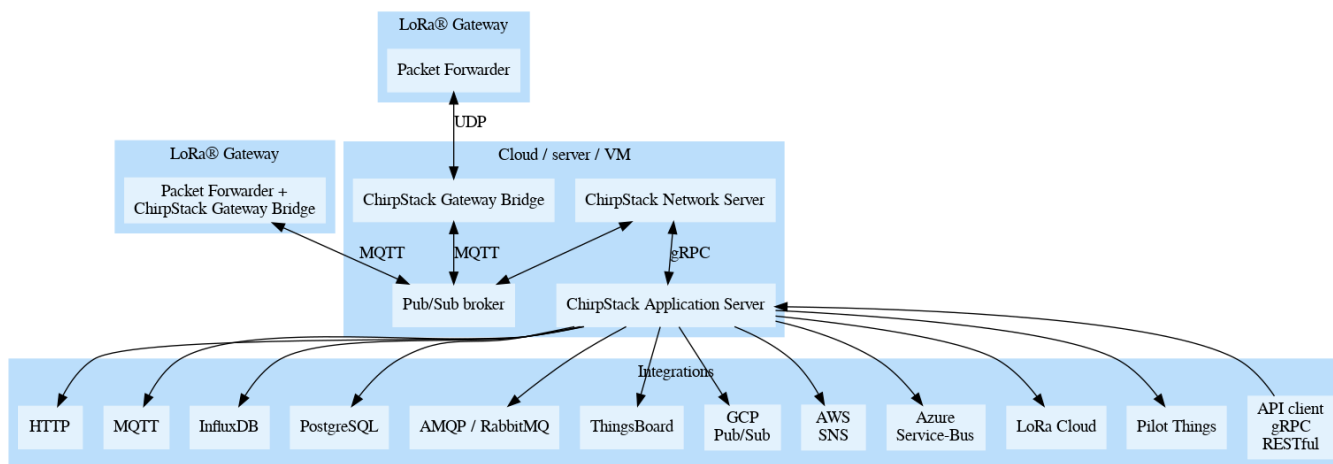
Για να γίνει πιο εύκολα κατανοητή η διαδικασία που απαιτείται, η παρουσίαση της θα χωριστεί σε τρία κεφάλαια. Αρχικά θα παρουσιαστεί η εγκατάσταση ενός απλού δικτύου LoRaWAN με μία πύλη, μία τερματική συσκευή και από έναν network και Application Server. Στην συνέχεια θα παρουσιαστεί η επέκταση της προηγούμενης εγκατάστασης για την επίτευξη ενεργοποίησης της τερματικής συσκευής μέσω OTAA activation over DNS, Στο τέλος να παρουσιαστεί η επέκταση των προηγούμενων εγκαταστάσεων για την επίτευξη του στόχου της υποστήριξης roaming.

Η υλοποίηση των παραπάνω θα βασιστεί στην πλατφόρμα ανοικτού κώδικα Chirpstack και στην χρήση ενός μικροϋπολογιστή Raspberry Pi 3 Model B, ενός expansion module SX1308 Raspberry Pi LoRa Gateway Board και ενός μικροεπεξεργαστή LoPy.

Ακολουθεί μια πιο αναλυτική περιγραφή του εξοπλισμού και του λογισμικού.

#### 5.2 ChipStack

Το ChirpStack είναι μία πλατφόρμα ανοικτού κώδικα για την υλοποίηση των στοιχείων (components) ενός δικτύου LoRaWAN. Αποτελεί μια έτοιμη προς χρήση λύση που περιλαμβάνει μια φιλική προς το χρήστη διεπαφή ιστού για την διαχείριση των συσκευών και των API για την ενοποίηση των εφαρμογών που επεξεργάζονται τα δεδομένα που συλλέγουν οι συσκευές. Η αρθρωτή αρχιτεκτονική καθιστά δυνατή την ενσωμάτωση του σε ήδη υπάρχουσες υποδομές. Όλα τα στοιχεία έχουν άδεια χρήσης βάσει της άδειας MIT και μπορούν να χρησιμοποιηθούν για εμπορικούς σκοπούς.



Σχήμα 5.1: Αρχιτεκτονική Chirpstack [29]

Παρέχονται τα ακόλουθα components:

**ChirpStack Gateway Bridge:** Χειρίζεται την επικοινωνία με τις πύλες (gateways) LoRaWAN.

**ChirpStack Network Server:** Υλοποιεί τον LoRaWAN Network Server.

**ChirpStack Application Server:** Υλοποιεί τον LoRaWAN Application Server.

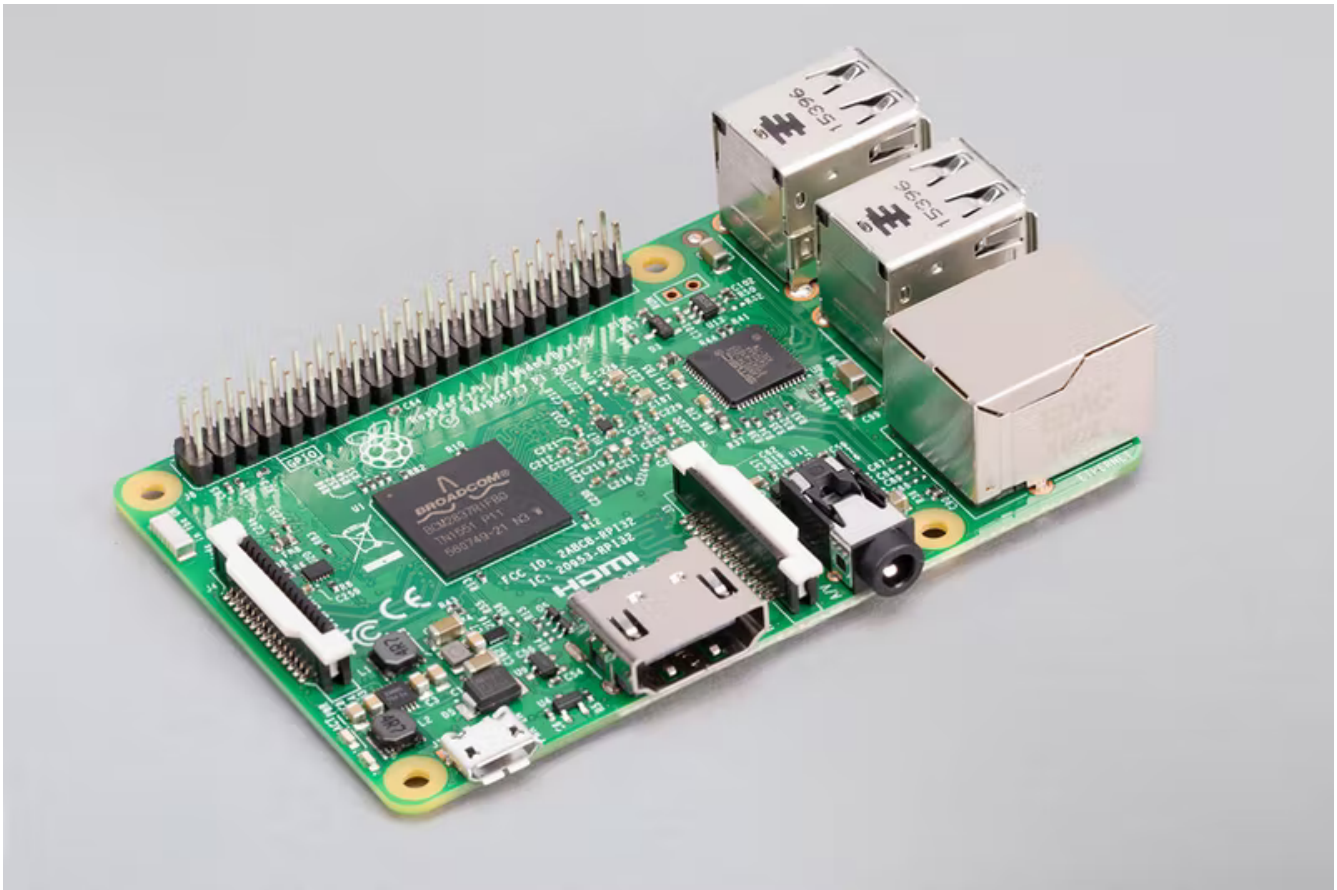
**ChirpStack Gateway OS:** Λειτουργικό σύστημα που βασίζεται σε Linux για την εκτέλεση της (πλήρους) στοίβας ChirpStack σε μια πύλη LoRa που βασίζεται στο Raspberry Pi [28].

Αξίζει να σημειωθεί ότι οι πύλες χρησιμοποιούν ένα λογισμικό που ονομάζεται Packet Forwarder για να δέχονται και να στέλνουν μηνύματα. Συνήθως χρησιμοποιείται το Semtech UDP Packet Forwarder. Μετά από τον Packet Forwarder υπάρχει το ChirpStack Gateway Bridge που μετατρέπει τα μηνύματα αυτά σε κατάλληλη μορφή για τα components του Chirpstack [29].

## 5.3 Εξοπλισμός

### 5.3.1 Raspberry Pi 3 Model B

Το Raspberry Pi 3 Model B είναι ένας single-board υπολογιστής σε μέγεθος πιστωτικής κάρτας που κυκλοφόρησε τον Φεβρουάριο του 2016 από την βρετανική εταιρεία Raspberry Pi. Υποστηρίζει συνδεσιμότητα Bluetooth και LAN. Για να λειτουργήσει χρειάζεται τροφοδοσία 5V 2.5A. Το λειτουργικό σύστημα εγκαθίσταται σε κάρτα μνήμης microSD και το πιο ευρέως χρησιμοποιούμενο είναι το Raspbian, μία διανομή Linux βασισμένη στο Debian (32-bit) [30].



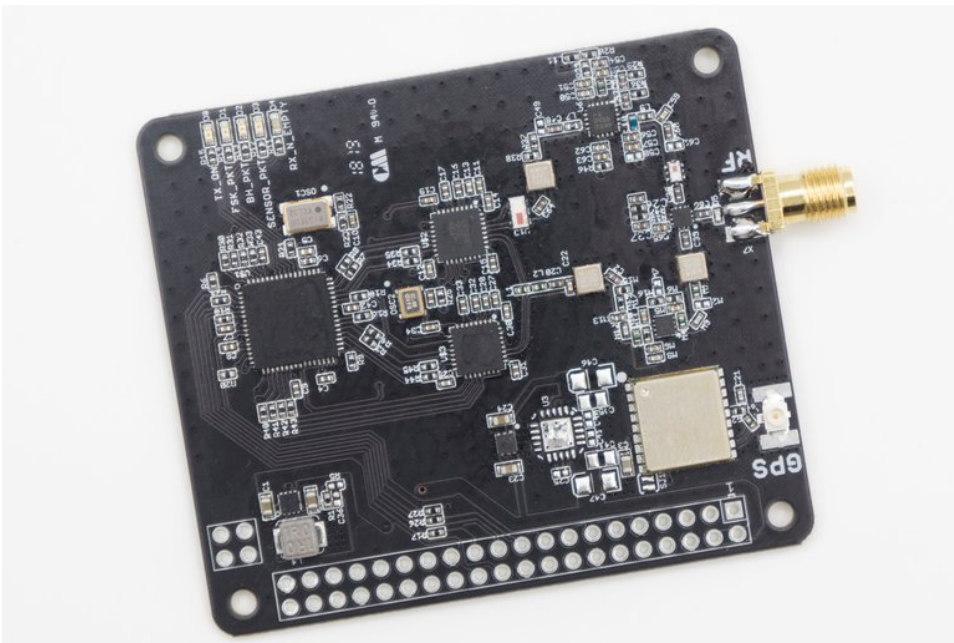
Εικόνα 5.1: Raspberry Pi 3 Model B [30]

Χαρακτηριστικά:

Quad Core 1.2GHz Broadcom BCM2837 64bit CPU  
1GB RAM  
BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board  
100 Base Ethernet  
40-pin extended GPIO  
4 USB 2 ports  
4 Pole stereo output and composite video port  
Full size HDMI  
CSI camera port for connecting a Raspberry Pi camera  
DSI display port for connecting a Raspberry Pi touchscreen display  
Micro SD port for loading your operating system and storing data  
Upgraded switched Micro USB power source up to 2.5A

### 5.3.2 SX1308 Raspberry Pi LoRa Gateway Board

Είναι βαθμίδα επέκτασης (expansion module) του Raspberry Pi για υλοποίηση λειτουργιών δικτύων LoRAWAN και κατασκευάζεται από την εταιρία Will Whang's Electronics. Χρησιμοποιεί ένα Semtech chip SX1308 με δύο transceivers SX1257 και on-board LNA and PA with RF switch που έχει σχεδιαστεί ειδικά για το Raspberry Pi. Αυτή η πλακέτα διαθέτει επίσης μονάδα GPS MAX-7Q, η έξοδος pulse-per-second του οποίου είναι συνδεδεμένη τόσο με το SX1308 όσο και με το Raspberry Pi, οπότε όχι μόνο το SX1308 μπορεί να έχει ακριβή χρονισμό, αλλά και το Raspberry Pi μπορεί να χρησιμοποιηθεί ως NTP server.



Εικόνα 5.2: SX1308 Raspberry Pi LoRa Gateway Board [31]

### 5.3.3 LoPy

Το LoPy είναι ένας triple bearer MicroPython enabled μικροεπεξεργαστής που υποστηρίζει συνδεσιμότητα LoRa, Wifi και Bluetooth και κυκλοφορεί από την εταιρία ADAFRUIT INDUSTRIES. Χρησιμοποιεί Semtech LoRa transceiver SX1272 και μπορεί να λειτουργήσει σαν Class A ή C end device [32].

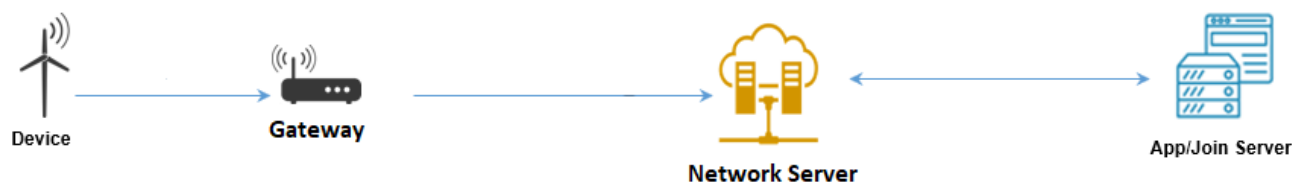


Εικόνα 5.3: LoPy [32]

## Κεφάλαιο 6: Υλοποίηση βασικής υποδομής LoRaWAN σε Chirpstack

### 6.1 Εισαγωγή

Η πιο απλή μορφή δικτύου LoRaWAN αποτελείται από μία τερματική συσκευή (end node), μία πύλη (gateway), ένα Network Server και έναν Application Server.



Σχήμα 6.1: Αρχιτεκτονική απλού δικτύου LoRaWAN [33]

Στην συνέχεια θα αναλυθεί πως υλοποιείται αυτή η αρχιτεκτονική σε περιβάλλον Chirpstack.

Πρέπει να επισημάνουμε ότι στα configuration files που θα ακολουθήσουν έχει ρυθμιστεί η λειτουργία του δικτύου LoRaWAN στις ευρωπαϊκές συχνότητες EU868.

### 6.2 Αρχικοποίηση πύλης (gateway)

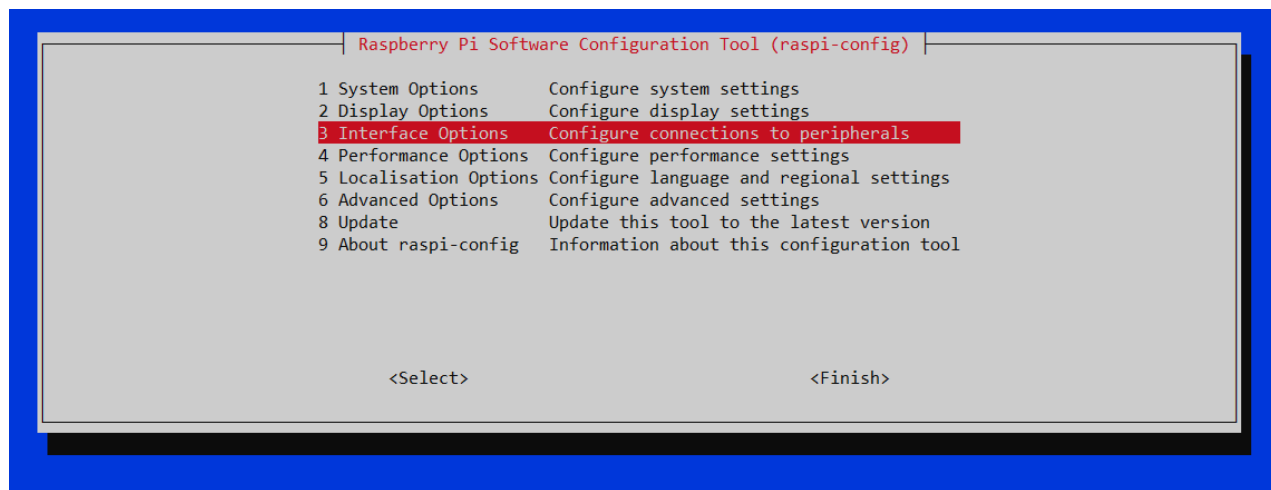
Το ρόλο του gateway θα παίξει ένας υπολογιστής Raspberry Pi 3 Model B στον οποίο έχει συνδεθεί ένα SX1308 Raspberry Pi LoRa Gateway Board, το οποίο θα επεκτείνει το Raspberry με λειτουργικότητα LoRa.

Στο Raspberry εγκαθιστούμε το λειτουργικό σύστημα Raspberry Pi OS Lite, το οποίο βρίσκουμε σε μορφή ISO στην ιστοσελίδα <https://www.raspberrypi.com/software/operating-systems/>. Η εγκατάσταση γίνεται στην κάρτα microSD με χρήση εργαλείων όπως το balenaEtcher (<https://www.balena.io/etcher/>).

Έπειτα ενεργοποιούμε την συσκευή και με την βοήθεια μιας οθόνης και ενός πληκτρολογίου προβαίνουμε στην αρχικοποίηση της (αλλαγή password, ρύθμιση WiFi) η οποία δεν αποτελεί μέρος της εργασίας αφού πρόκειται για τυπική διαδικασία. Μόλις αρχικοποιηθεί το Raspberry και συνδεθεί στο internet μπορούμε να συνδεόμαστε και μέσω ssh.

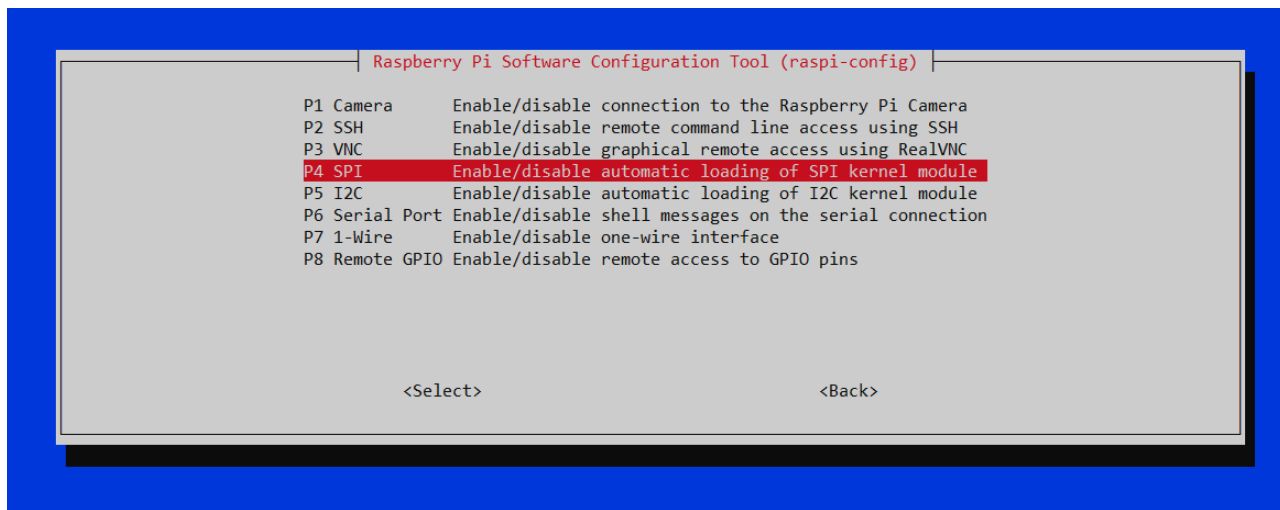
Ακολουθούν τα βήματα που αφορούν στη εγκατάσταση του λογισμικού για την λειτουργία LoRaWAN gateway.

Αρχικά ανοίγουμε ένα τερματικό και δίνουμε την εντολή: `sudo raspi-config` για να βρεθούμε στο menu στο οποίο αρχικοποιούμε την συσκευή.



Εικόνα 6.1: Menu αρχικοποίησης Raspberry

Επιλέγουμε Interface Options και στην συνέχεια SPI. Ακολουθούμε την ίδια διαδικασία και με την επιλογή Remote GPIO. Αυτές οι διεπαφές πρέπει να ενεργοποιηθούν για να υπάρχει επικοινωνία μεταξύ του Raspberry και του gateway board.



Εικόνα 6.2: Ενεργοποίηση SPI

Έπειτα επιστρέφουμε στο τερματικό και τρέχουμε τις παρακάτω εντολές για να υπάρξει επικαιροποίηση όλων των πακέτων του συστήματος.

```
sudo apt-get update

sudo apt-get install
```

Στην συνέχεια εγκαθιστούμε την κατάλληλη βιβλιοθήκη για τον έλεγχο των θυρών γενικής χρήσης (GPIO):

```
sudo apt-get install wiringpi
```

Το λογισμικό που εκτελεί την λειτουργία LoRa gateway εγκαθίσταται με τις παρακάτω εντολές:

```
sudo git clone https://github.com/will127534/lora\_gateway.git

cd lora_gateway

sudo make
```

Επειδή το gateway θα υλοποιηθεί σε περιβάλλον Chirpstack πρέπει να εγκατασταθεί ο Semtech UDP Packet Forwarder σύμφωνα με τις παρακάτω εντολές:

```
cd

sudo git clone https://github.com/Lora-net/packet\_forwarder.git

cd packet_forwarder

sudo make
```

Στην συνέχεια στο subdirectory `/packet_forwarder/lora_pkt_fwd` υπάρχει το αρχείο `global_conf.json` στο οποίο πρέπει να ορίσουμε τις παρακάτω μεταβλητές, έτσι ώστε ο packet forwarder να μπορεί να επικοινωνεί με το Chirpstack Gateway Bridge:

```
"gateway_ID": "b827ebFFFEae8205",
  "server_address": "localhost",
  "serv_port_up": 1700,
  "serv_port_down": 1700,
```



Παρατηρούμε ότι έχουμε ορίσει το DevEUI της συσκευής στο πεδίο gateway\_ID και στα πεδία server\_address και serv\_port\_up και serv\_port\_down έχουμε ορίσει localhost και 1700 αντίστοιχα. Αυτό γίνεται για να υπάρχει επικοινωνία με το Chirpstack Gateway Bridge που θα εγκαταστήσουμε στο ίδιο μηχάνημα (επιλογή localhost) και θα “ακούει” στο port 1700.

Τέλος εγκαθιστούμε το Chirpstack Gateway Bridge σύμφωνα με τις οδηγίες:

```
cd

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys
1CE2AFD36DBCCA00

sudo echo "deb https://artifacts.chirpstack.io/packages/3.x/deb stable
main" | sudo tee /etc/apt/sources.list.d/chirpstack.list

sudo apt update

sudo apt install chirpstack-gateway-bridge
```

Στο configuration file (/etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml) ορίζουμε:

```
server="tcp://localhost:1883"
```

Με αυτό τον τρόπο επιτρέπουμε στο Chirpstack Gateway Bridge να επικοινωνεί με το Chirpstack Network Server στη προκαθορισμένη διεύθυνση και port. Στην περίπτωση μας Ο Network Server θα εγκατασταθεί στο ίδιο μηχάνημα με το gateway.

Τώρα αν δώσουμε τις παρακάτω εντολές θα ενεργοποιήσουμε το packet forwarder και το Chirpstack Gateway Bridge και το gateway θα είναι έτοιμο για λειτουργία:

```
sudo systemctl start chirpstack-gateway-bridge

cd packet_forwarder/lora_pkt_fwd/

./lora_pkt_fwd
```



```
pi@raspberrypi3:~/packet_forwarder/lor_pkt_fwd
pi@raspberrypi3:~/packet_forwarder/lor_pkt_fwd $ ./lor_pkt_fwd
*** Beacon Packet Forwarder for Lora Gateway ***
Version: 4.0.1
*** Lora concentrator HAL library version info ***
Version: 5.0.1;
***
INFO: little endian host
INFO: found global configuration file global_conf.json, parsing it
INFO: global_conf.json does contain a JSON object named SX1301_conf, parsing SX1301 parameters
INFO: lorawan_public 1, clksrc 1
INFO: no configuration for LBT
INFO: antenna_gain 0 dbi
INFO: no configuration for tx gain lut 12
INFO: no configuration for tx gain lut 13
INFO: no configuration for tx gain lut 14
INFO: no configuration for tx gain lut 15
INFO: Configuring TX LUT with 12 indexes
INFO: radio 0 enabled (type SX1257), center frequency 867500000, RSSI offset -166.000000, tx enabled 1, tx_notch_freq 0
INFO: radio 1 enabled (type SX1257), center frequency 868500000, RSSI offset -166.000000, tx enabled 0, tx_notch_freq 0
INFO: Lora multi-SF channel 0> radio 1, IF -400000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 1> radio 1, IF -200000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 2> radio 1, IF 0 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 3> radio 0, IF -400000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 4> radio 0, IF -200000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 5> radio 0, IF 0 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 6> radio 0, IF 200000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora multi-SF channel 7> radio 0, IF 400000 Hz, 125 kHz bw, SF 7 to 12
INFO: Lora std channel> radio 1, IF -200000 Hz, 250000 Hz bw, SF 7
INFO: FSK channel> radio 1, IF 300000 Hz, 125000 Hz bw, 50000 bps datarate
INFO: global_conf.json does contain a JSON object named gateway_conf, parsing gateway parameters
INFO: gateway MAC address is configured to B827EBFFFAE8205
INFO: server hostname or IP address is configured to "localhost"
INFO: upstream port is configured to "1700"
INFO: downstream port is configured to "1700"
INFO: downstream keep-alive interval is configured to 10 seconds
INFO: statistics display interval is configured to 30 seconds
INFO: upstream PUSH_DATA time-out is configured to 100 ms
INFO: packets received with a valid CRC will be forwarded
INFO: packets received with a CRC error will NOT be forwarded
INFO: packets received with no CRC will NOT be forwarded
INFO: [main] concentrator started, packet can now be received

INFO: Disabling GPS mode for concentrator's counter...
INFO: [down] PULL_ACK received in 0 ms
INFO: host/sx1301 time offset=(1645464414s:120453µs) - drift=-1981486587µs
INFO: Enabling GPS mode for concentrator's counter.

INFO: [down] PULL_ACK received in 0 ms
INFO: [down] PULL_ACK received in 0 ms
```

Εικόνα 6.3: Terminal Semtech UDP Packet Forwarder

### 6.3 Αρχικοποίηση Chirpstack Network Server

Ο Chirpstack Network Server θα εγκατασταθεί στο Raspberry στο οποίο έχουμε υλοποιήσει το gateway.

Αρχικά ανοίγουμε ένα τερματικό στο μηχάνημα και με τις παρακάτω εντολές εγκαθιστούμε κάποια προαπαιτούμενα προγράμματα. Πρόκειται για τα Mosquitto, PostgreSQL database και Redis.

```
sudo apt install mosquitto

sudo apt-get install postgresql

sudo -u postgres psql
```

Μέσα στο PostgreSQL prompt εκτελούμε τα παρακάτω

```
create role chirpstack_ns with login password 'dbnspassword';

create database chirpstack_ns with owner chirpstack_ns;
```

```
\q
```

```
sudo apt-get install redis-server
```

Έπειτα με τις παρακάτω εντολές εγκαθιστούμε το Chirpstack Network Server:

```
sudo apt install apt-transport-https
```

```
sudo apt-get install dirmngr -install-recommends
```

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys  
1CE2AFD36DBCCA00
```

```
sudo echo "deb https://artifacts.chirpstack.io/packages/3.x/deb stable  
main" | sudo tee /etc/apt/sources.list.d/chirpstack.list
```

```
sudo apt-get update
```

```
sudo apt-get install chirpstack-network-server
```

Στο configuration file (/etc/chirpstack-network-server/chirpstack-network-server.toml) ορίζουμε:

```
dsn="postgres://chirpstack_ns:dbnpassword@localhost/chirpstack_ns?sslmode=  
disable"
```

Με αυτό τον τρόπο υπάρχει πρόσβαση του Chirpsatc Network Server στην βάση δεδομένων PostgreSQL ώστε να αποθηκεύει τα δεδομένα που χρησιμοποιεί.

```
[join_server]
```

```
[join_server.default]
```

```
server="http://localhost:8003"
```

Ορίζουμε την διεύθυνση του Join Server που θα χρησιμοποιηθεί για την ενεργοποίηση της τερματικής συσκευής. Στην περίπτωση αυτή ο Join Server βρίσκεται στο ίδιο μηχάνημα με τον Network Server. Πιο συγκεκριμένα είναι ο Application Server και “ακούει” στην θύρα 8003.

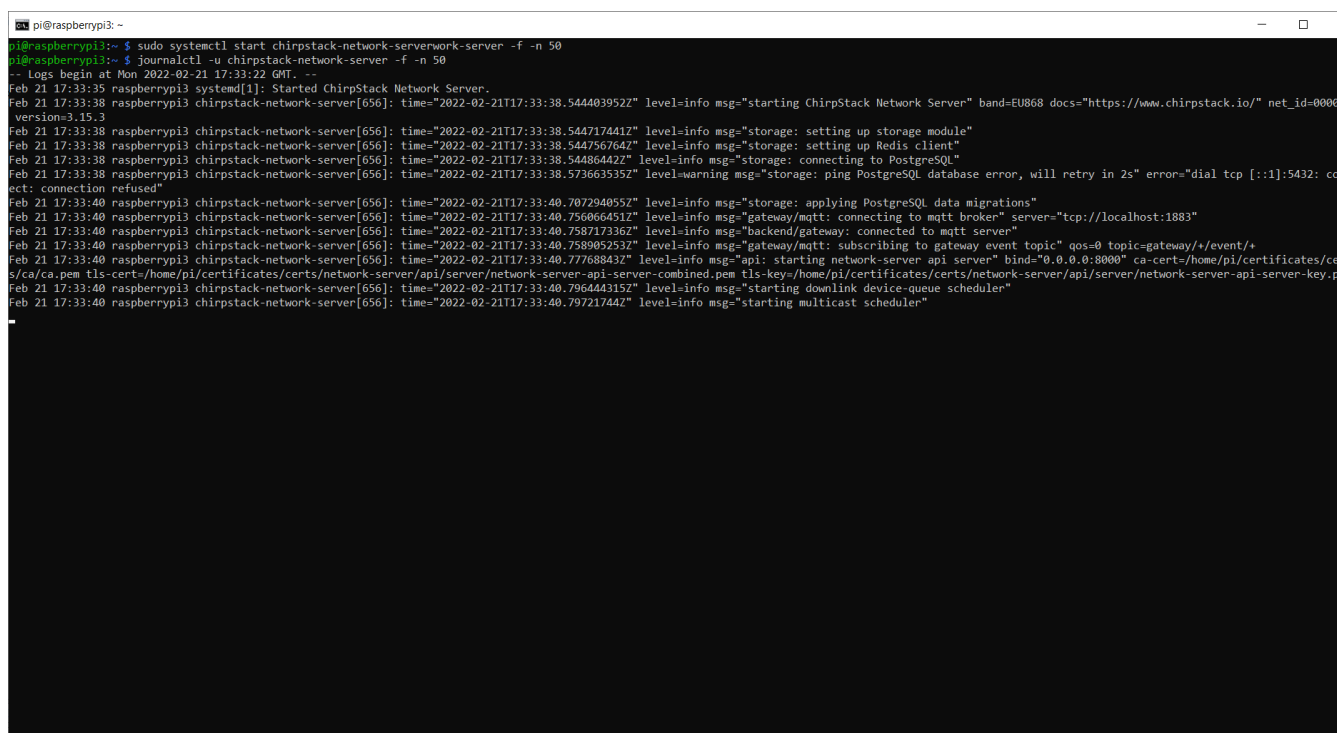
Στο τέλος με την εντολή:

```
sudo systemctl start chirpstack-network-server
```

εκκινούμε τον Chirpsack Network Server και με την εντολή:

```
journalctl -u chirpstack-network-server -f -n 50
```

μπορούμε να βλέπουμε τα περιεχόμενα του log file.



```
pi@raspberrypi3: ~
pi@raspberrypi3:~$ sudo systemctl start chirpstack-network-server -f -n 50
pi@raspberrypi3:~$ journalctl -u chirpstack-network-server -f -n 50
-- Logs begin at Mon 2022-02-21 17:33:22 GMT. --
Feb 21 17:33:35 raspberrypi3 systemd[1]: Started ChirpStack Network Server.
Feb 21 17:33:38 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:38.544403952Z" level=info msg="starting ChirpStack Network Server" band=EU868 docs="https://www.chirpstack.io/" net_id=0000
version=3.15.3
Feb 21 17:33:38 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:38.544717441Z" level=info msg="storage: setting up storage module"
Feb 21 17:33:38 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:38.544756764Z" level=info msg="storage: setting up Redis client"
Feb 21 17:33:38 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:38.54486442Z" level=info msg="storage: connecting to PostgreSQL"
Feb 21 17:33:38 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:38.573663535Z" level=warning msg="storage: ping PostgreSQL database error, will retry in 2s" error="dial tcp [::1]:5432: connect: connection refused"
Feb 21 17:33:40 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:40.707294055Z" level=info msg="storage: applying PostgreSQL data migrations"
Feb 21 17:33:40 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:40.756066451Z" level=info msg="gateway/mqtt: connecting to mqtt broker" server="tcp://localhost:1883"
Feb 21 17:33:40 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:40.758717336Z" level=info msg="backend/gateway: connected to mqtt server"
Feb 21 17:33:40 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:40.758905253Z" level=info msg="gateway/mqtt: subscribing to gateway event topic" qos=0 topic=gateway+/event/+
Feb 21 17:33:40 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:40.77768843Z" level=info msg="api: starting network-server api server" bind="0.0.0.0:8000" ca-cert=/home/pi/certificates/certs/ca/ca.pem tls-certs=/home/pi/certificates/certs/network-server/api/server/network-server-api-server-combined.pem tls-key=/home/pi/certificates/certs/network-server/api/server/network-server-api-server-key.pem
Feb 21 17:33:40 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:40.796444315Z" level=info msg="starting downlink device-queue scheduler"
Feb 21 17:33:40 raspberrypi3 chirpstack-network-server[656]: time="2022-02-21T17:33:40.79721744Z" level=info msg="starting multicast scheduler"
```

Εικόνα 6.4: Log file Network Server

## 6.4 Αρχικοποίηση Chirpstack Application Server

Ο Chirpstack Application Server θα εγκατασταθεί στο Raspberry στο οποίο έχουμε υλοποιήσει το gateway.

Αρχικά ανοίγουμε ένα τερματικό στο μηχάνημα και με τις παρακάτω εντολές εγκαθιστούμε κάποια προαπαιτούμενα προγράμματα. Πρόκειται για τα Mosquitto, PostgreSQL database και Redis.

```
sudo apt install mosquitto
```

```
sudo apt-get install postgresql
```

```
sudo -u postgres psql
```

Μέσα στο PostgreSQL prompt εκτελούμε τα παρακάτω

```
create role chirpstack_as with login password 'dbaspassword';  
  
create database chirpstack_as with owner chirpstack_as;  
  
\c chirpstack_as  
  
create extension pg_trgm;  
  
create extension hstore  
  
\q
```

```
sudo apt-get install redis-server
```

Έπειτα με τις παρακάτω εντολές εγκαθιστούμε το Chirpstack Application Server:

```
sudo apt install apt-transport-https  
  
sudo apt-get install dirmngr --install-recommends  
  
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys  
1CE2AFD36DBCCA00  
  
sudo echo "deb https://artifacts.chirpstack.io/packages/3.x/deb stable  
main" | sudo tee /etc/apt/sources.list.d/chirpstack.list  
  
sudo apt-get update  
  
sudo apt-get install chirpstack-application-server
```

Στο configuration file (/etc/chirpstack-application-server/chirpstack-application-server.toml) ορίζουμε:

```
dsn="postgres://chirpstack_as:dbaspassword@localhost/chirpstack_as?sslmode=  
disable"
```

Με αυτό τον τρόπο υπάρχει πρόσβαση του Chirpstack Application Server στην βάση δεδομένων PostgreSQL ώστε να αποθηκεύει τα δεδομένα που χρησιμοποιεί.

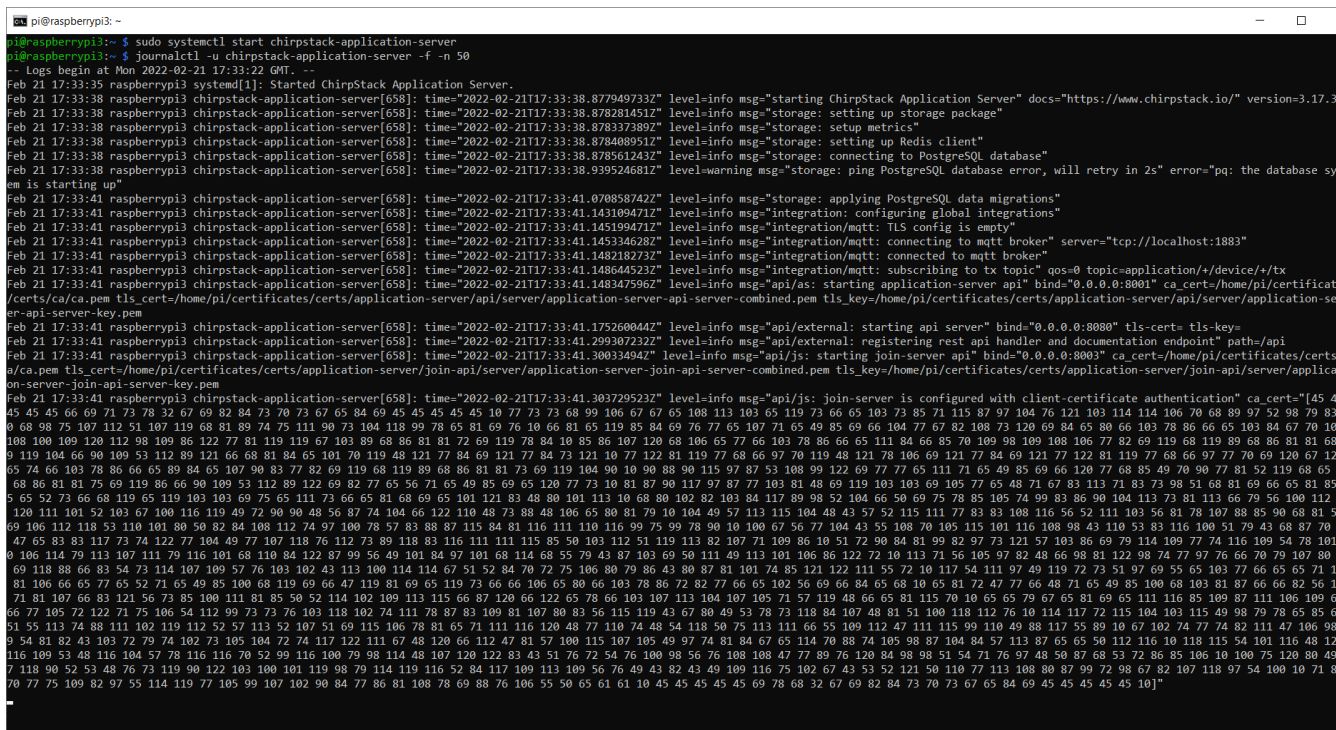
Στο τέλος με την εντολή:

```
sudo systemctl start chirpstack-application-server
```

εκκινούμε τον Chirpstack Application Server και με την εντολή:

```
journalctl -u chirpstack-application-server -f -n 50
```

μπορούμε να βλέπουμε τα περιεχόμενα του log file.



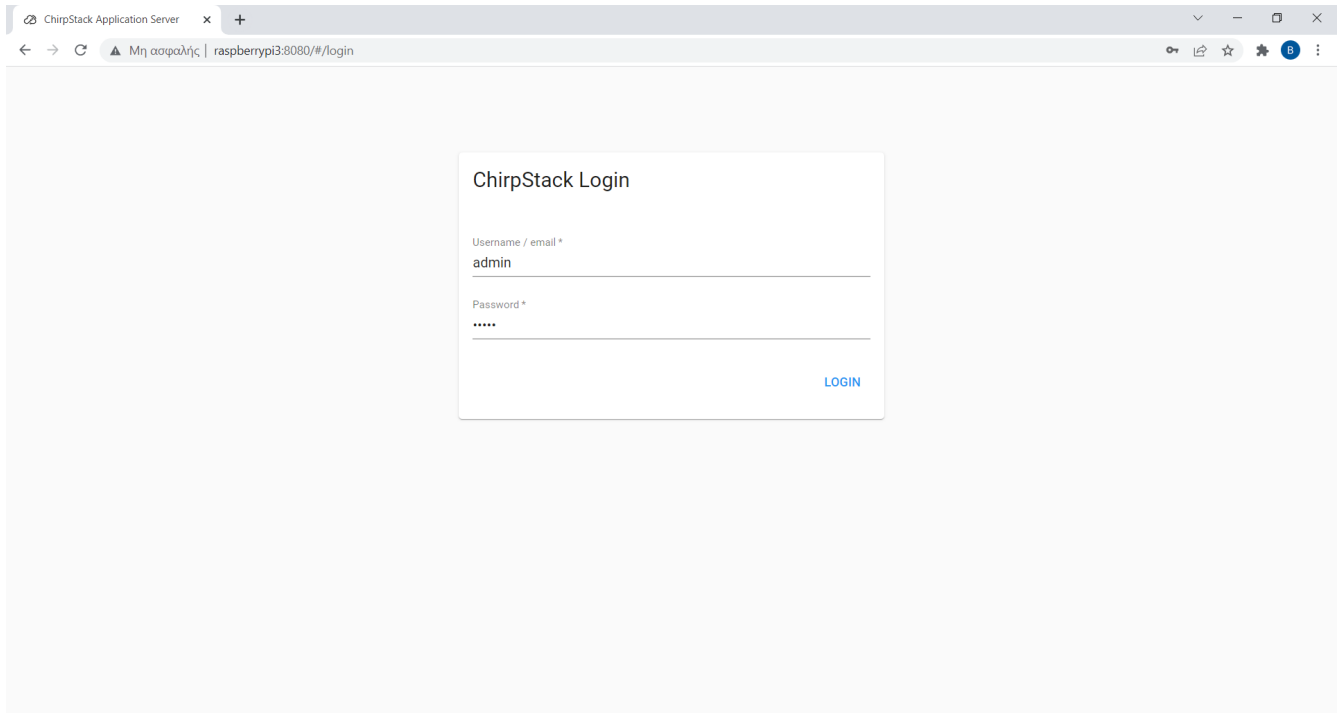
```
pi@raspberrypi3:~$ sudo systemctl start chirpstack-application-server
pi@raspberrypi3:~$ journalctl -u chirpstack-application-server -f -n 50
-- Logs begin at Mon 2022-02-21 17:33:22 GMT.
Feb 21 17:33:35 raspberrypi3 systemd[1]: Started ChirpStack Application Server.
Feb 21 17:33:38 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:38.877949733Z" level=info msg="starting ChirpStack Application Server" docs="https://www.chirpstack.io/" version=3.17.3
Feb 21 17:33:38 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:38.8782814517Z" level=info msg="storage: setting up storage package"
Feb 21 17:33:38 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:38.8783373897Z" level=info msg="storage: setup metrics"
Feb 21 17:33:38 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:38.8784089517Z" level=info msg="storage: setting up Redis client"
Feb 21 17:33:38 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:38.878561243Z" level=info msg="storage: connecting to PostgreSQL database"
Feb 21 17:33:38 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:38.939524681Z" level=warning msg="storage: ping PostgreSQL database error, will retry in 25s" error="pq: the database sy
em is starting up"
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.070858742Z" level=info msg="storage: applying PostgreSQL data migrations"
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.143109471Z" level=info msg="integration: configuring global integrations"
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.145199471Z" level=info msg="integration/matt: TLS config is empty"
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.145334628Z" level=info msg="integration/matt: connecting to matt broker" server="tcp://localhost:1883"
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.148218273Z" level=info msg="integration/matt: connected to matt broker" qos=0
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.148644523Z" level=info msg="integration/matt: subscribing to tx topic" qos=0 topic=application/+device/+tx
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.148347596Z" level=info msg="api/as: starting application-server api" bind="0.0.0.0:8001" ca_cert=/home/pi/certificat
er/certs/ca.ca.pem tls_cert=/home/pi/certificates/certs/application-server/api/server/application-server-combined.pem tls_key=/home/pi/certificates/certs/application-server/api/server/application-se
er-api-server-key.pem
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.175260044Z" level=info msg="api/external: starting api server" bind="0.0.0.0:8000" tls_cert=tls-key-
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.299307232Z" level=info msg="api/external: registering rest api handler and documentation endpoint" paths/api
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.30033494Z" level=info msg="api/js: starting join-server api" bind="0.0.0.0:8003" ca_cert=/home/pi/certificates/certs
a/ca.pem tls_cert=/home/pi/certificates/certs/application-server/join-api/server/application-server-combined.pem tls_key=/home/pi/certificates/certs/application-server/join-api/server/applica
on-server-join-api-server-key.pem
Feb 21 17:33:41 raspberrypi3 chirpstack-application-server[658]: time="2022-02-21T17:33:41.303729523Z" level=info msg="api/js: join-server is configured with client-certificate authentication" ca_cert="[45 4
4 45 45 66 69 71 73 78 32 67 69 82 84 73 70 73 67 65 84 69 45 45 45 45 10 77 73 73 68 99 106 67 67 65 108 113 103 65 119 73 66 65 103 73 85 71 115 87 97 104 76 121 103 114 114 106 70 68 89 97 52 98 79 83
0 68 98 75 107 112 51 107 119 68 81 89 74 75 111 90 73 104 118 99 78 65 81 69 76 10 66 81 65 119 85 84 69 76 77 65 107 71 65 49 85 69 66 104 77 67 82 108 73 120 69 84 65 80 66 103 78 86 66 65 103 84 67 70 10
108 100 109 120 112 98 109 86 122 77 81 119 119 67 103 89 68 86 81 81 72 69 119 78 84 10 85 86 107 120 68 106 65 77 66 103 78 86 66 65 111 84 66 85 70 109 98 109 108 106 77 82 69 119 68 119 89 68 86 81 81 68
9 119 104 66 90 109 53 112 89 121 66 68 81 84 65 101 70 119 48 121 77 84 69 121 77 84 73 121 10 77 122 81 119 77 68 66 97 77 70 69 120 67 12
65 74 66 103 78 86 66 65 89 84 65 107 90 83 77 82 69 119 68 119 89 68 86 81 81 73 69 119 104 90 10 90 88 115 97 87 53 108 99 122 69 77 77 65 111 71 65 49 85 69 66 120 77 68 85 49 90 70 77 81 52 119 68 65
68 86 81 81 75 69 119 86 66 90 109 53 112 89 122 69 82 77 65 56 71 65 49 85 69 65 120 77 73 10 81 87 90 117 97 87 77 103 81 48 69 119 103 103 69 105 77 65 48 71 67 83 113 71 83 73 98 51 68 81 69 66 65 81 85
5 65 52 73 66 68 119 65 119 103 103 69 75 65 111 73 66 65 81 68 69 65 101 121 83 48 80 101 113 10 68 80 102 82 103 84 117 89 98 52 104 66 50 69 75 78 85 105 74 99 83 86 90 104 113 73 81 113 66 79 56 100 112
120 111 101 52 103 67 100 116 119 49 72 90 90 48 56 87 74 104 66 122 110 48 73 88 48 106 65 80 81 79 10 104 49 57 113 115 104 48 43 57 52 115 111 77 83 83 108 116 56 52 111 103 56 81 78 107 88 85 90 68 81 5
69 106 112 118 53 110 101 80 50 82 84 108 112 74 97 100 78 57 83 88 77 115 84 81 116 111 110 116 99 75 99 78 90 100 67 56 77 104 43 55 108 70 105 110 101 116 108 98 43 110 53 83 116 100 51 79 43 68 87 70
47 65 83 83 117 73 74 122 77 104 49 77 107 118 76 112 73 89 118 83 116 111 111 115 85 50 103 112 51 119 113 82 107 71 109 86 10 51 72 90 84 81 99 82 97 73 121 57 103 86 69 79 114 109 77 74 116 109 54 78 101
0 106 114 79 113 107 111 79 116 101 68 110 84 122 87 99 56 49 101 84 97 101 68 114 68 55 79 43 87 103 69 50 111 49 113 101 106 86 122 72 10 113 71 56 105 97 82 48 66 98 81 122 98 74 77 97 76 66 70 79 107 80
69 118 88 66 83 54 73 114 107 109 57 76 103 43 113 100 114 114 67 51 52 84 70 72 75 106 80 79 86 43 80 87 81 101 74 85 121 122 111 55 72 10 117 54 111 97 49 119 72 73 51 97 69 55 65 103 77 66 65 65 71 1
81 106 66 65 77 65 52 71 65 49 85 100 66 119 69 66 47 119 81 69 65 110 73 66 66 106 65 80 66 103 78 86 72 82 77 66 65 102 36 69 66 84 65 68 10 65 81 72 47 77 66 48 71 65 49 85 100 68 103 81 87 66 66 82 56 1
71 81 107 66 83 121 56 73 85 100 111 81 85 50 52 114 102 109 113 115 66 87 120 66 122 65 78 66 103 107 113 104 107 105 71 57 119 48 66 65 81 115 70 10 65 65 79 67 65 81 69 65 111 116 85 109 87 111 106 109 6
66 77 105 72 122 71 75 106 54 112 99 73 73 76 103 118 102 74 111 78 87 83 109 81 107 80 83 56 115 119 43 67 80 49 53 78 73 118 84 107 48 81 51 100 118 112 76 10 114 117 72 115 104 103 115 49 98 79 78 65 85 6
51 55 113 74 88 111 109 112 52 57 113 52 107 51 69 115 106 78 81 65 71 111 116 120 48 77 110 74 48 54 118 50 75 113 111 66 65 109 112 47 111 115 99 110 49 88 117 55 89 10 67 102 74 77 74 82 111 47 106 98
9 54 81 82 43 103 72 79 74 102 73 105 104 72 74 117 122 111 67 48 120 66 112 47 81 57 100 115 107 105 49 97 74 81 84 67 65 114 70 88 74 105 98 87 104 84 57 113 87 65 65 50 112 116 10 118 115 54 101 116 48 12
116 109 53 48 116 104 57 78 116 116 70 52 99 116 100 79 98 114 48 107 120 123 83 43 51 76 72 54 76 100 98 56 76 108 108 47 77 89 76 120 84 98 98 51 54 71 76 97 48 50 87 68 53 72 86 85 106 10 100 75 120 80 49
7 118 90 52 53 48 76 73 119 90 102 103 100 101 119 98 79 114 119 116 52 84 117 109 113 109 56 76 49 43 82 43 49 109 116 75 102 67 43 53 52 121 50 110 77 113 108 80 87 99 72 98 67 82 107 118 97 54 100 10 71 8
70 77 75 109 82 97 55 114 119 77 105 99 107 102 90 84 77 86 81 108 78 69 88 76 106 55 50 65 61 61 10 45 45 45 45 45 69 78 68 32 67 69 82 84 73 70 73 67 65 84 69 45 45 45 45 10"]"
```

Εικόνα 6.5: Log file Application Server

## 6.5 Εγγραφή ενός gateway στον Network Server

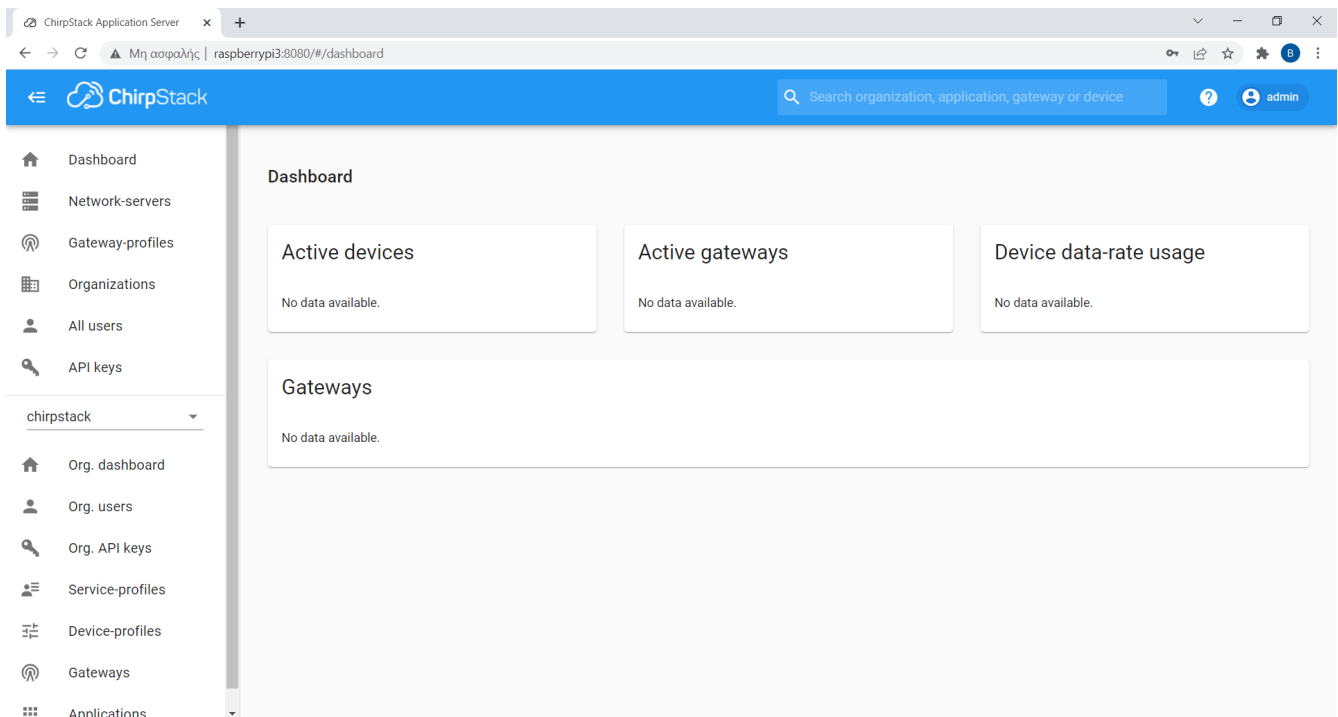
Ο Chirpstack Application Server προσφέρει ένα Web Interface ώστε να μπορεί ο χρήστης με ευκολία να ρυθμίζει και να επιτηρεί το δίκτυο LoraWAN που έχει στήσει. Αυτό το Web Interface βρίσκεται στην διεύθυνση <http://machine-IP:8080> όπου το machine-IP αναφέρεται στην διεύθυνση IP που διαθέτει η συσκευή στην οποία έχει εγκατασταθεί ο Chirpstack Application Server. Το port 8080 ορίζεται στο configuration file και μπορούμε να επιλέξουμε κάποιο άλλο port αν είναι απαραίτητο.

Συνεπώς ανοίγουμε έναν browser και αποκτούμε πρόσβαση στο Web Interface



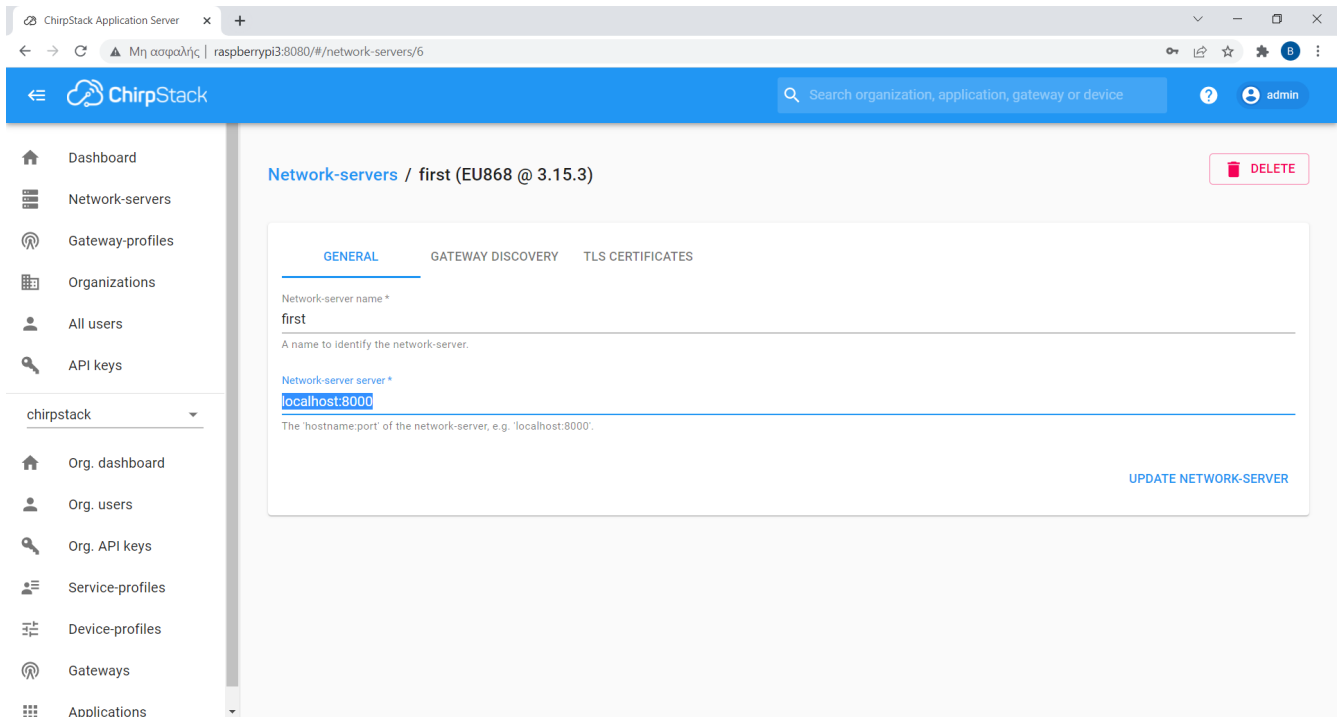
Εικόνα 6.6: Log in page του web interface του Application Server

Στην αρχική σελίδα κάνουμε login χρησιμοποιώντας την πρώτη φορά το προεπιλεγμένο username και password admin. Στην συνέχεια μπορούμε να επιλέξουμε κάποια άλλα συνθηματικά εισόδου.



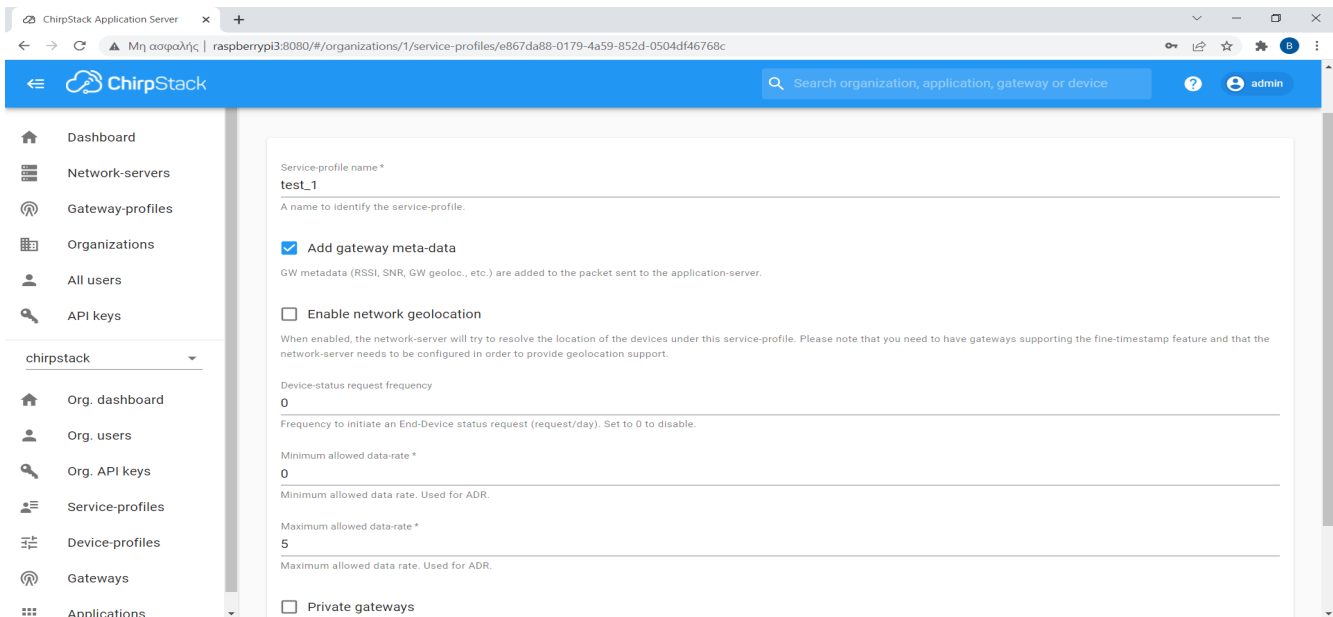
Εικόνα 6.7: Αρχική σελίδα του web interface του Application Server

Επιλέγουμε Network servers και δημιουργούμε ένα νέο Network Server. Τον ονομάζουμε όπως θέλουμε και ορίζουμε ως διεύθυνση του localhost:8000. Η διεύθυνση και το port αυτό έχουν οριστεί στο configuration file του Chirpstack Network Server και προκύπτει από το γεγονός ότι Network και Application Server βρίσκονται στο ίδιο μηχάνημα.



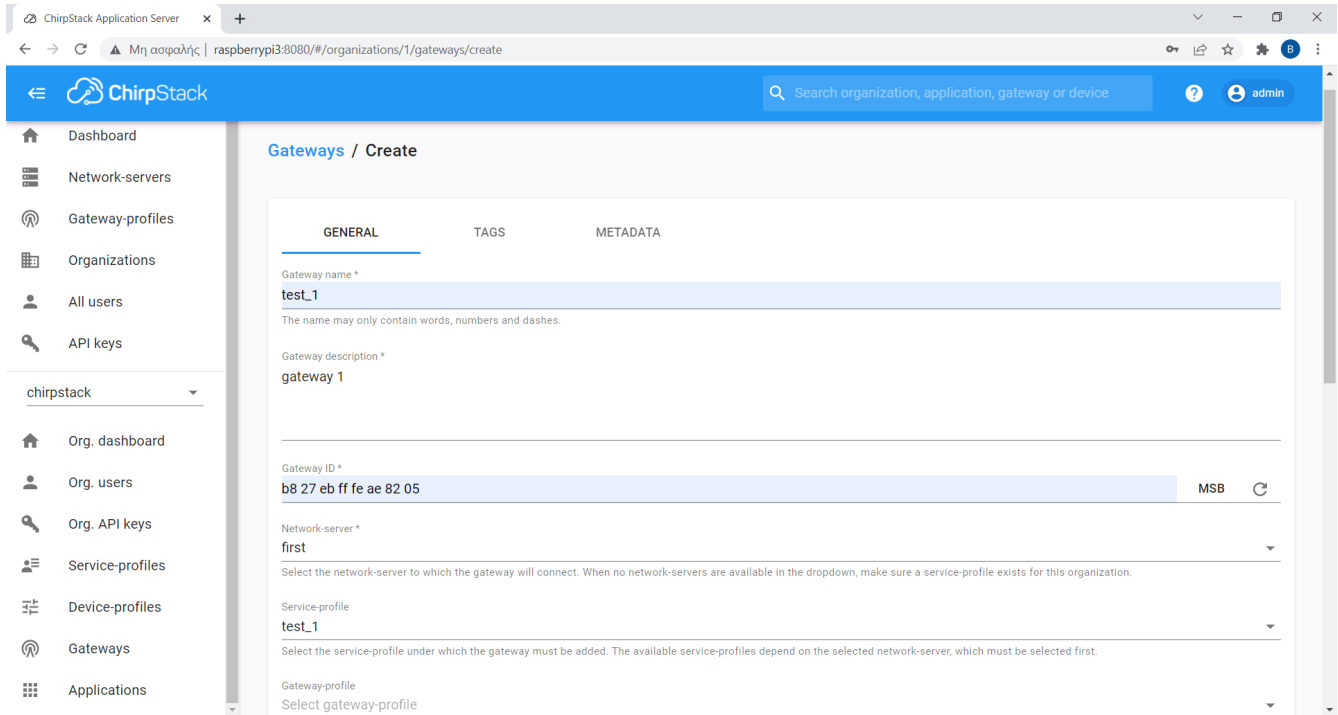
Εικόνα 6.8: Δημιουργία νέου Network Server

Στην συνέχεια επιλέγουμε Service profiles και δημιουργούμε ένα νέο δίνοντας τις παρακάτω τιμές στις ζητούμενες μεταβλητές



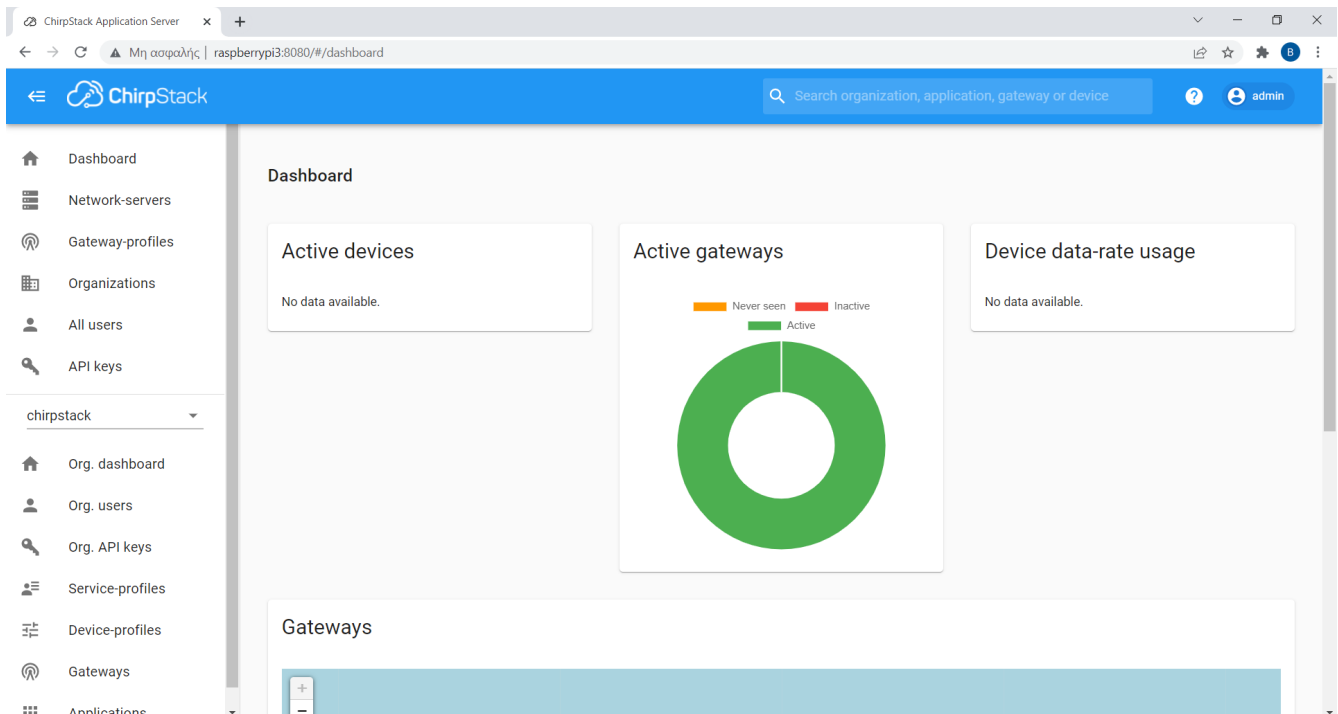
Εικόνα 6.9: Δημιουργία νέου Service profile

Τώρα είμαστε έτοιμοι να εγγράψουμε το gateway στον Network Server. Επιλέγουμε Gateways και δημιουργούμε ένα νέο gateway συμπληρώνοντας τα απαιτούμενα πεδία και κυρίως δίνοντας το σωστό DevEUI του gateway.



Εικόνα 6.10: Δημιουργία νέου Gateway

Αν έχουμε εισάγει τα σωστά στοιχεία τότε εμφανίζεται το gateway ως ενεργό.



Εικόνα 6.11: Εμφάνιση του gateway ως ενεργοποιημένο

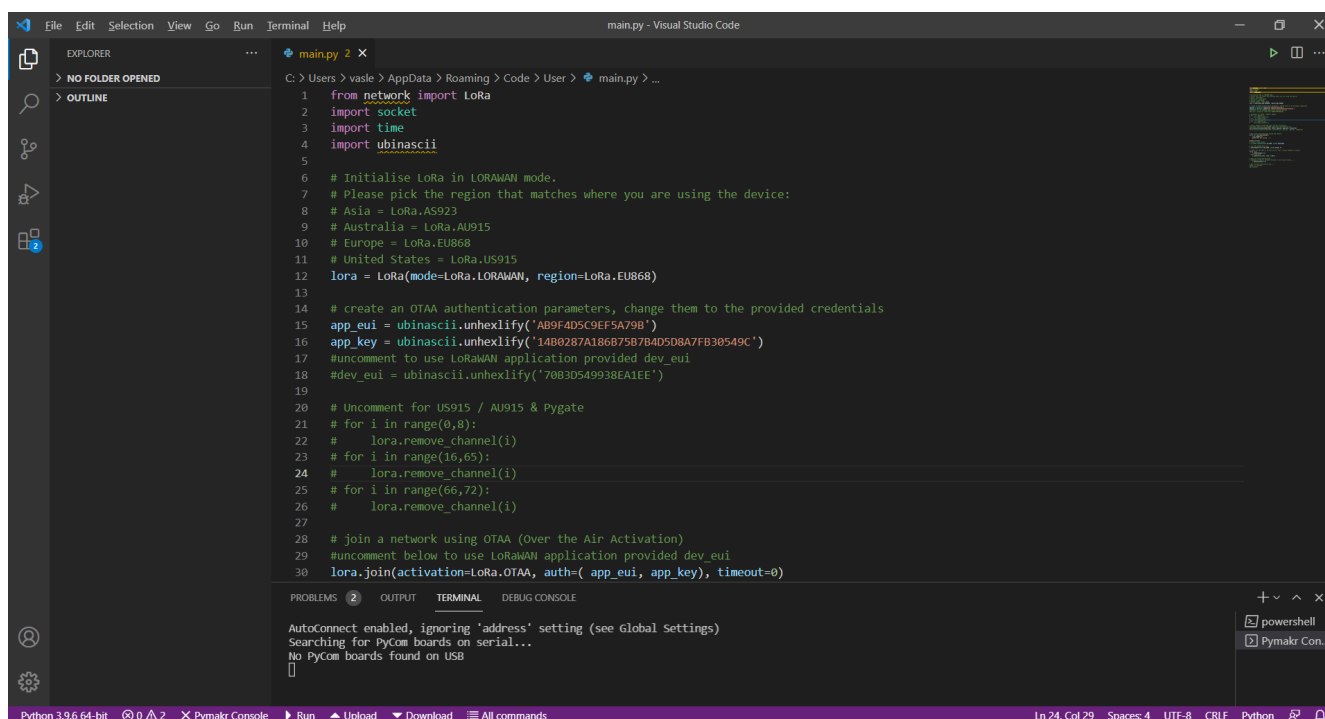


## 6.6 Ρύθμιση τερματικής συσκευής (end device / node)

Ως τερματική συσκευή έχει χρησιμοποιηθεί ένας μικροεπεξεργαστής LoPy ο οποίος τρέχει κατάλληλο πρόγραμμα ώστε να συμπεριφέρεται ως τερματική συσκευή. Για τον προγραμματισμό του LoPy χρησιμοποιήθηκε το πρόγραμμα Visual Studio Code (<https://code.visualstudio.com/>) στο οποίο είχε προστεθεί το plug in Pymakr (<https://pycom.io/products/supported-networks/pymakr/>) έτσι ώστε να μπορεί να τρέξει ο κώδικας από το LoPy.

Το LoPy συνδέθηκε μέσω των pins του με μία βάση Pytrack και μέσω usb με έναν υπολογιστή. Στο Visual Studio Code σε ένα νέο Project φορτώθηκε ο κώδικας που αποτελεί τροποποίηση του κώδικα συσκευής LoRa για OTAA activation που παρέχεται στην διεύθυνση <https://docs.pycom.io/tutorials/networks/lora/lorawan-otaa/>.

Στο πρόγραμμα αυτό ορίζουμε το AppKey της εφαρμογής και ένα AppEUI.



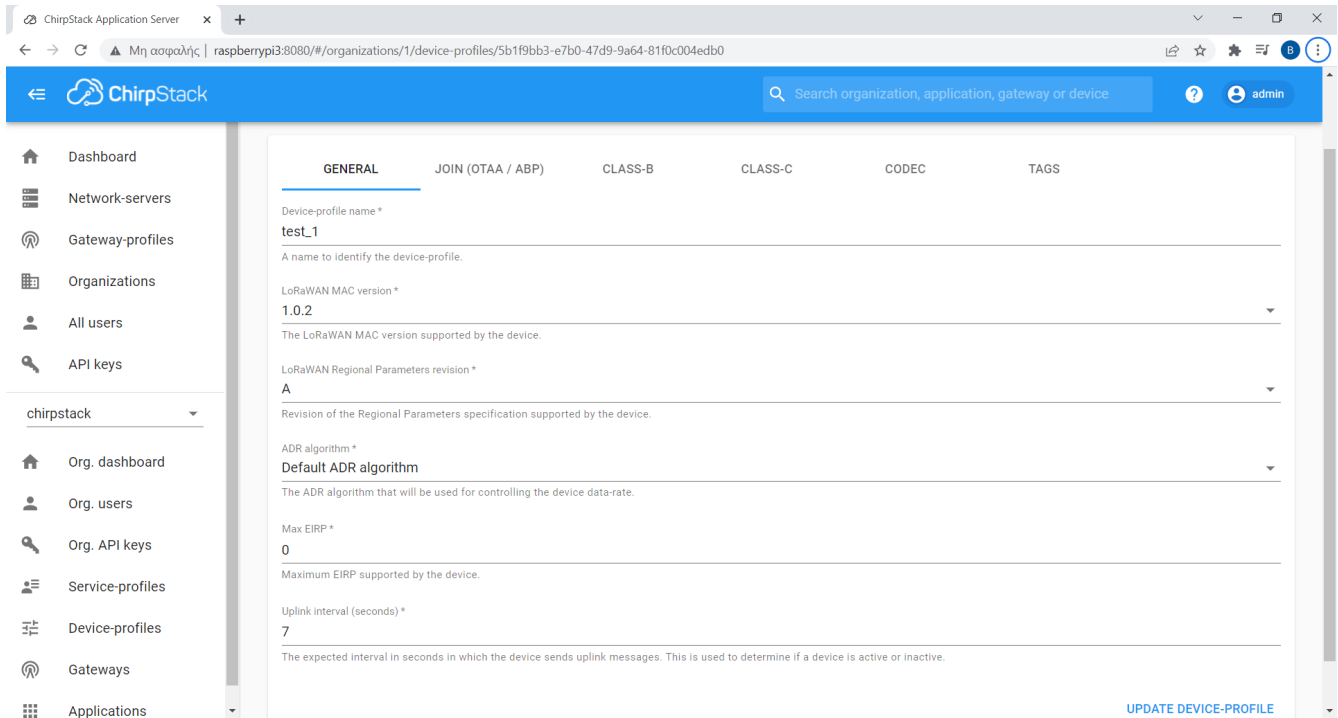
```
1 from network import LoRa
2 import socket
3 import time
4 import ubinascii
5
6 # Initialise LoRa in LORAWAN mode.
7 # Please pick the region that matches where you are using the device:
8 # Asia = LoRa.AS923
9 # Australia = LoRa.AU915
10 # Europe = LoRa.EU868
11 # United States = LoRa.US915
12 lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868)
13
14 # create an OTAA authentication parameters, change them to the provided credentials
15 app_eui = ubinascii.unhexlify('A89F4D5C9EF5A79B')
16 app_key = ubinascii.unhexlify('14B0287A186875B7B4D5D8A7FB30549C')
17 #uncomment to use LoRaWAN application provided dev_eui
18 #dev_eui = ubinascii.unhexlify('70B3D549938EA1EE')
19
20 # Uncomment for US915 / AU915 & Pygate
21 # for i in range(0,8):
22 #     lora.remove_channel(i)
23 # for i in range(16,65):
24 #     lora.remove_channel(i)
25 # for i in range(66,72):
26 #     lora.remove_channel(i)
27
28 # join a network using OTAA (Over the Air Activation)
29 #uncomment below to use LoRaWAN application provided dev_eui
30 lora.join(activation=LoRa.OTAA, auth=(app_eui, app_key), timeout=0)
```

Εικόνα 6.12: Προγραμματισμός του LoPy

Το LoPy όταν εκτελείται το παραπάνω πρόγραμμα θα στέλνει μηνύματα Join Request μέχρι να συνδεθεί σε ένα δίκτυο LoRaWAN και στην συνέχεια θα στέλνει περιοδικά κάποια μηνύματα με δεδομένα.

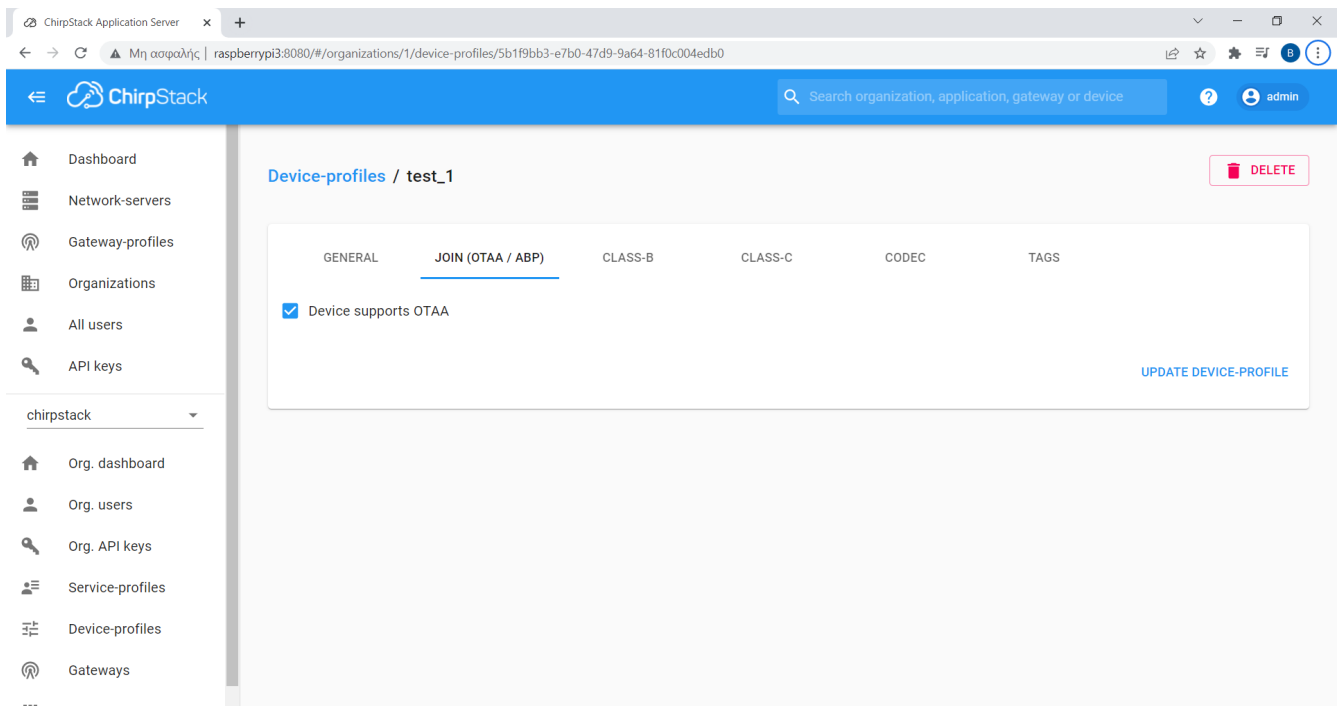
## 6.7 Εγγραφή ενός node στο δίκτυο LoRaWAN

Στο Web Interface επιλέγουμε Device-profiles και δημιουργούμε ένα νέο συμπληρώνοντας κατάλληλα τα απαιτούμενα πεδία.



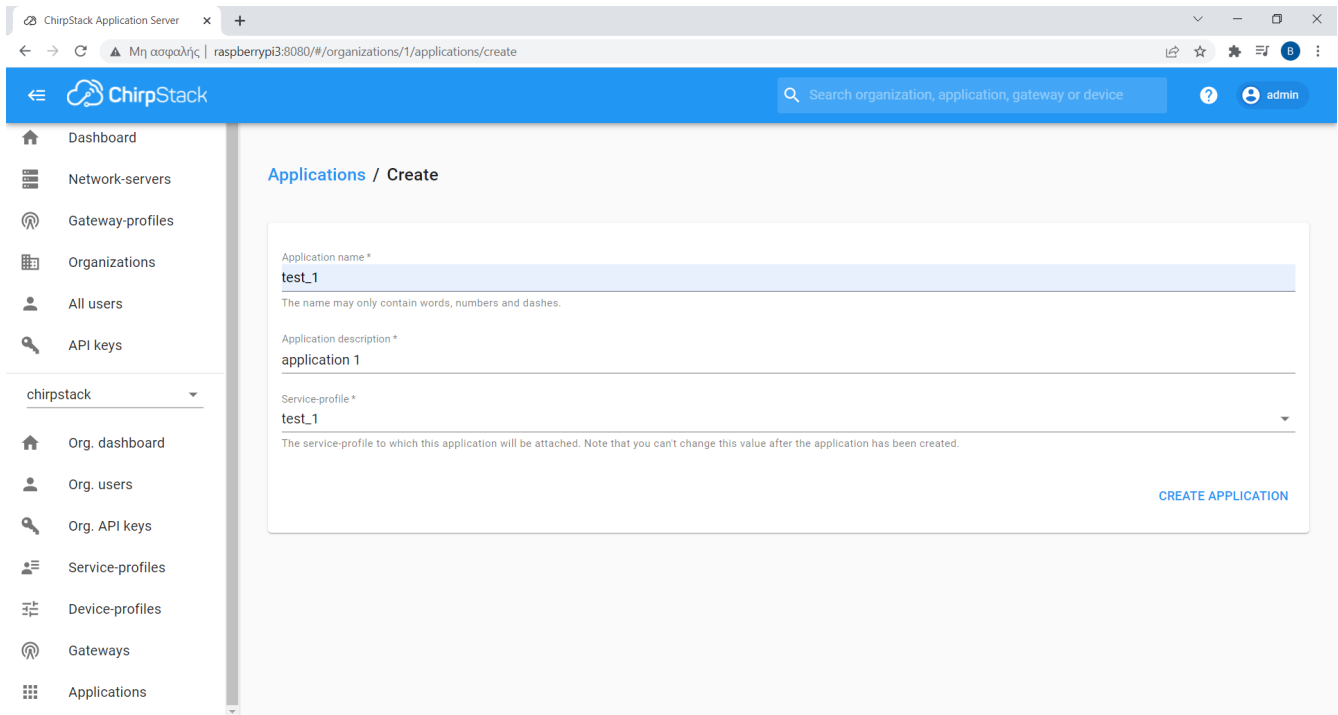
Εικόνα 6.13: Δημιουργία νέου Device profile

Παρατηρούμε ότι επιλέγουμε την έκδοση 1.0.2 του LoRaWAN διότι αυτή υποστηρίζεται από το LoPy. Επιλέγουμε ως κλάση συσκευής Class A για τον ίδιο λόγο. Ακόμη στην καρτέλα JOIN (OTAA / ABP) ενεργοποιούμε την λειτουργία OTAA.



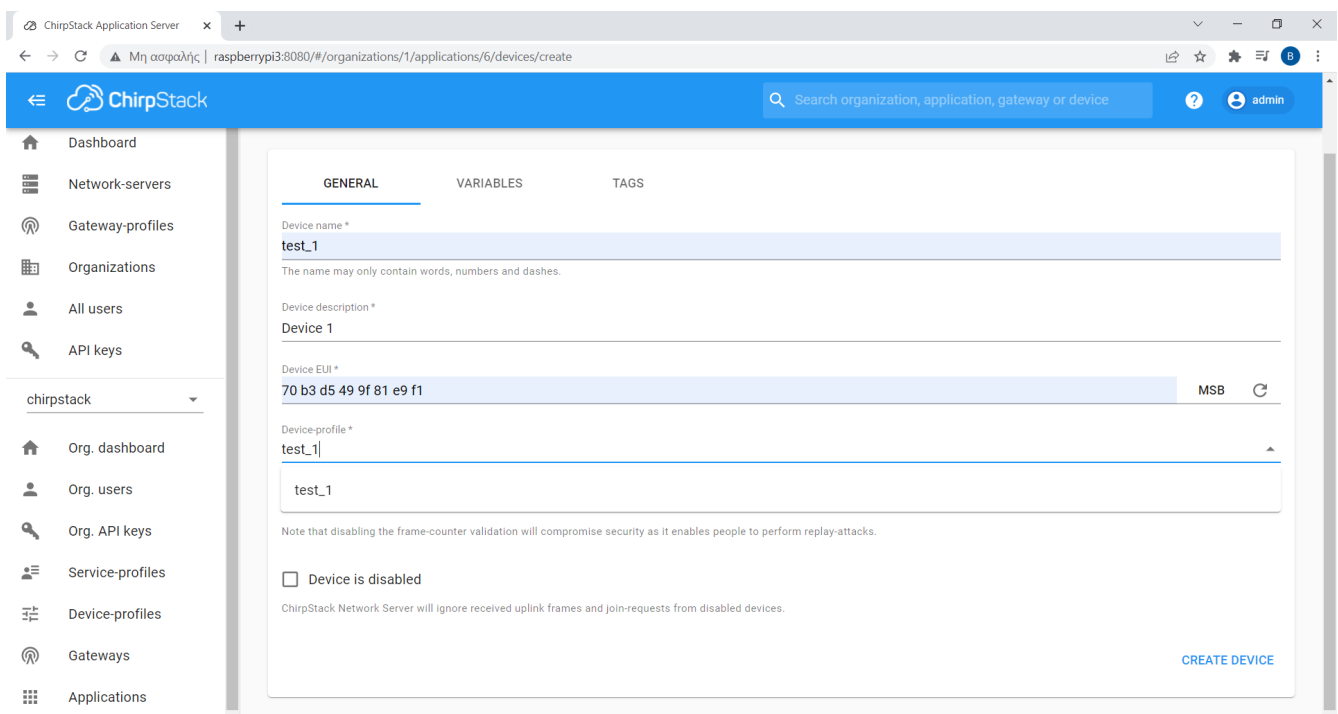
Εικόνα 6.14: Ενεργοποίηση OTA activation

Επιλέγουμε την καρτέλα Applications και δημιουργούμε μία νέα εφαρμογή συμπληρώνοντας κατάλληλα τα πεδία.



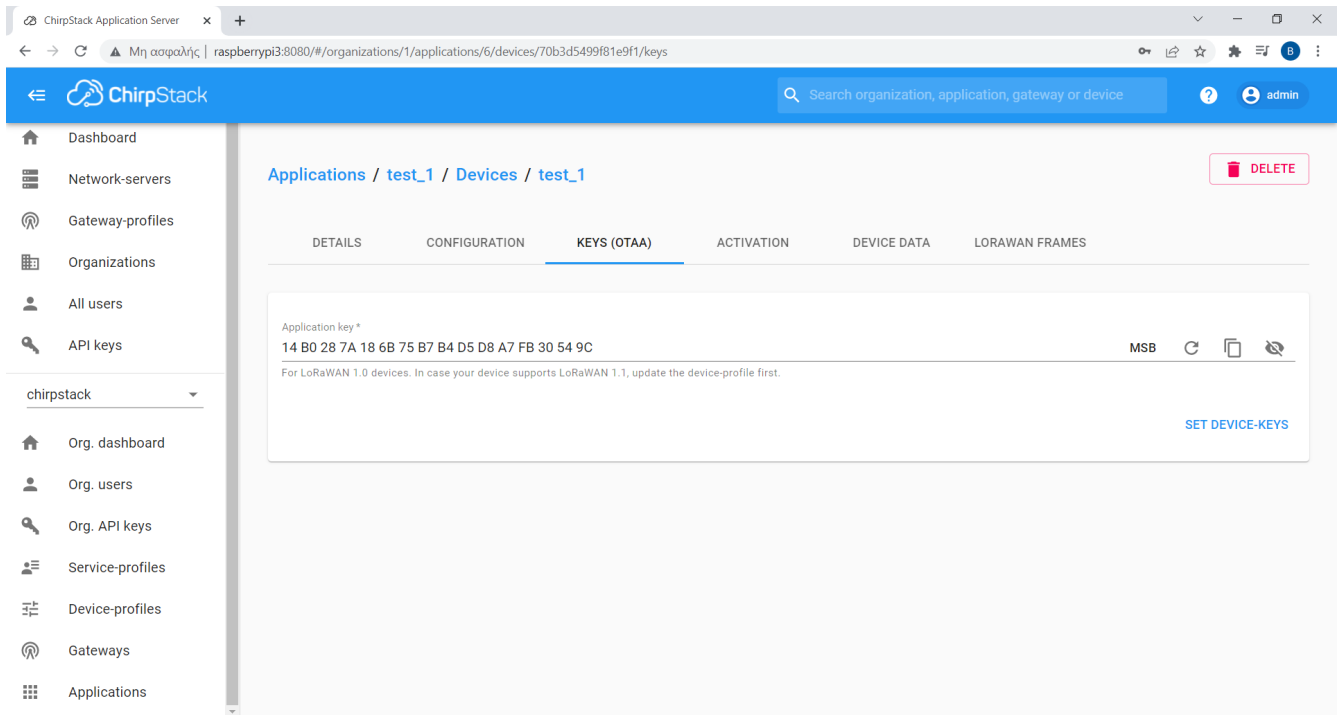
Εικόνα 6.15: Δημιουργία νέας εφαρμογής

Τώρα είμαστε έτοιμοι να ενεργοποιήσουμε την τερματική συσκευή. Στην εφαρμογή που δημιουργήσαμε προσθέτουμε μία νέα συσκευή δίνοντας το DevEUI της συσκευής.



Εικόνα 6.16: Εγγραφή νέας τερματικής συσκευής

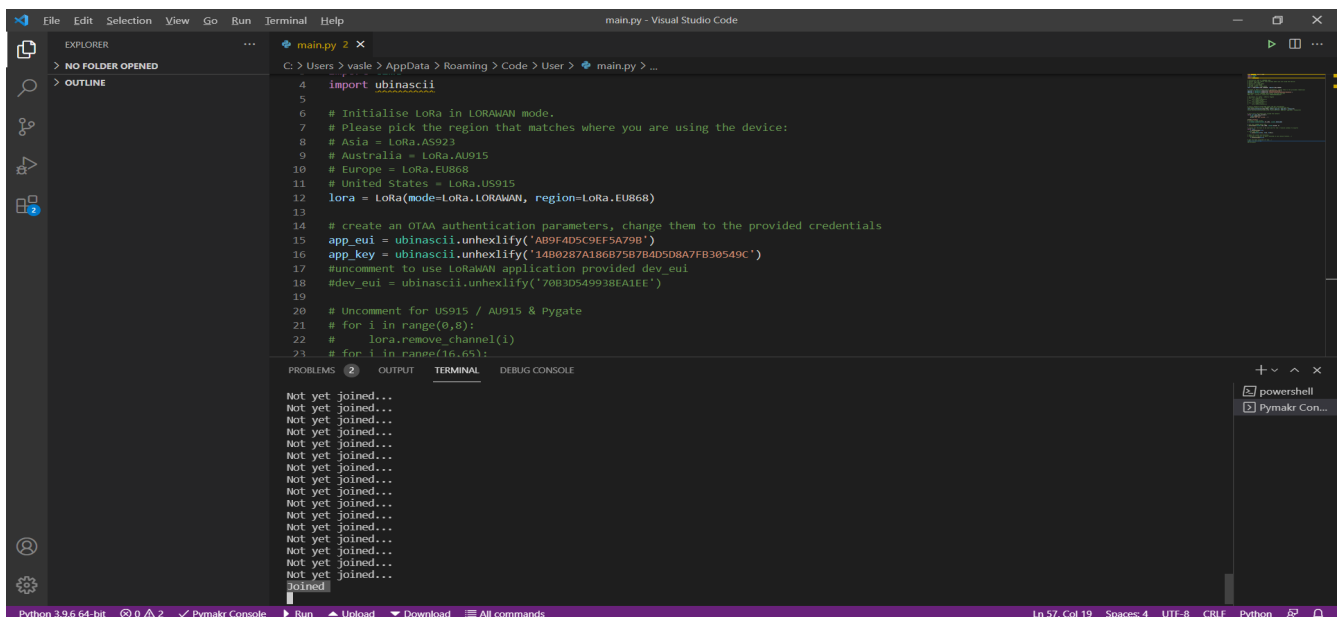
Στην συνέχεια στο πεδίο KEYS(OTAA) δίνουμε το ίδιο κλειδί AppKey που έχουμε ορίσει και στο LoPy.



Εικόνα 6.17: Παροχή AppKey

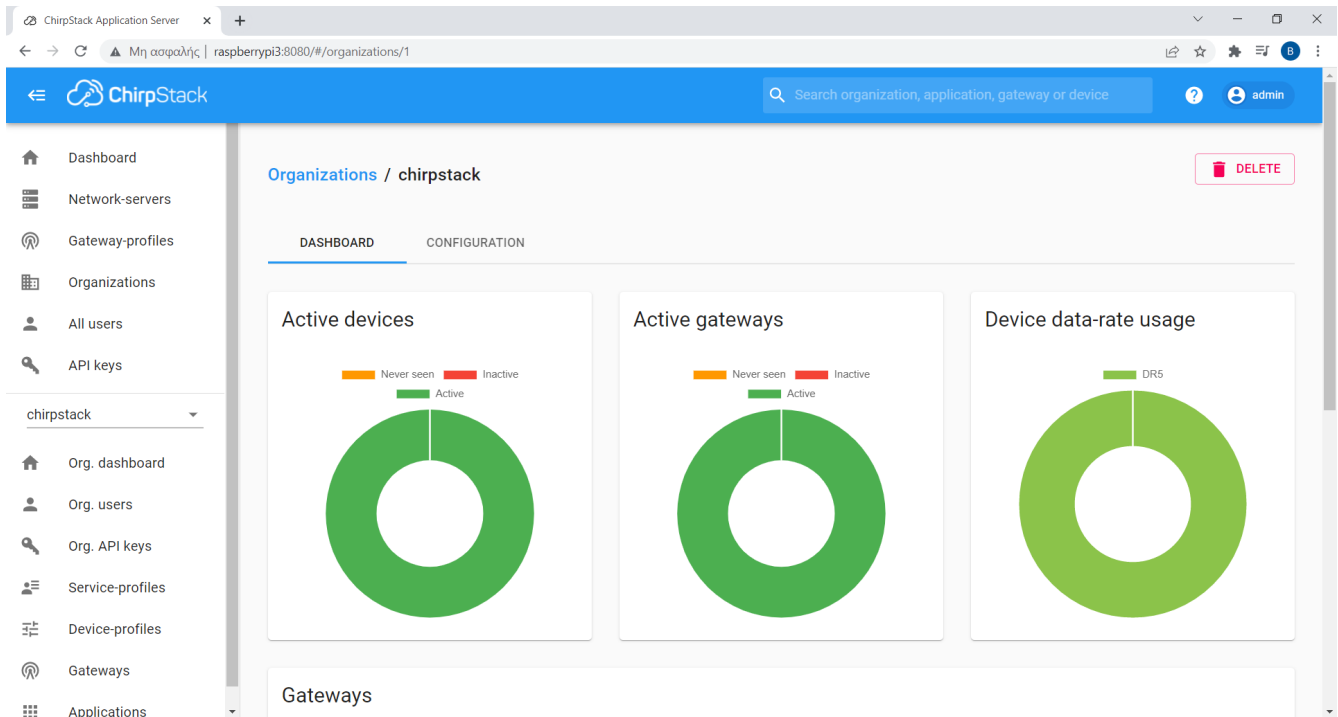
Τώρα η συσκευή έχει εγγραφεί και απομένει να ενεργοποιηθεί (activation).

Ξεκινάμε την εκτέλεση του κατάλληλου προγράμματος στο LoPy και περιμένουμε να στείλει το Join Request.



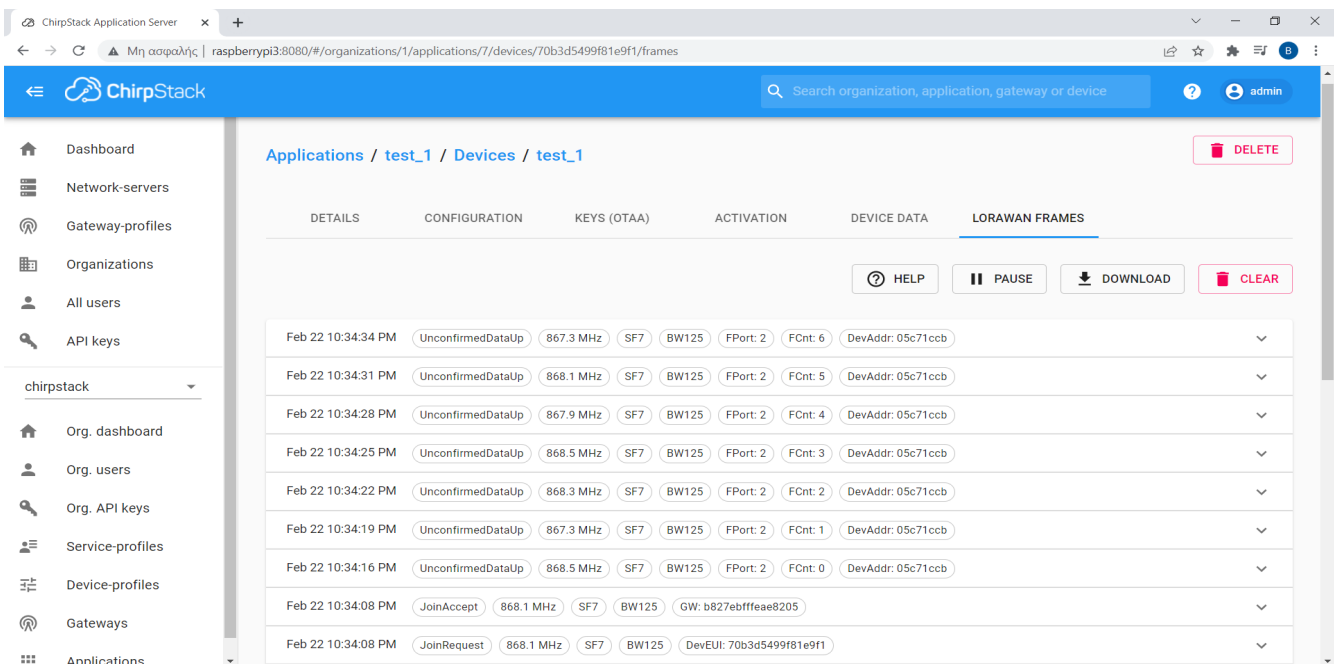
Εικόνα 6.18: Σύνδεση τερματικής συσκευής LoPy στο δίκτυο LoRaWAN

Βλέπουμε ότι αν υπάρξει θετική απάντηση Join Accept από τον Network Server η συσκευή είναι πλέον ορατή.



Εικόνα 6.19: Εμφάνιση τερματικής συσκευής ως ενεργοποιημένη

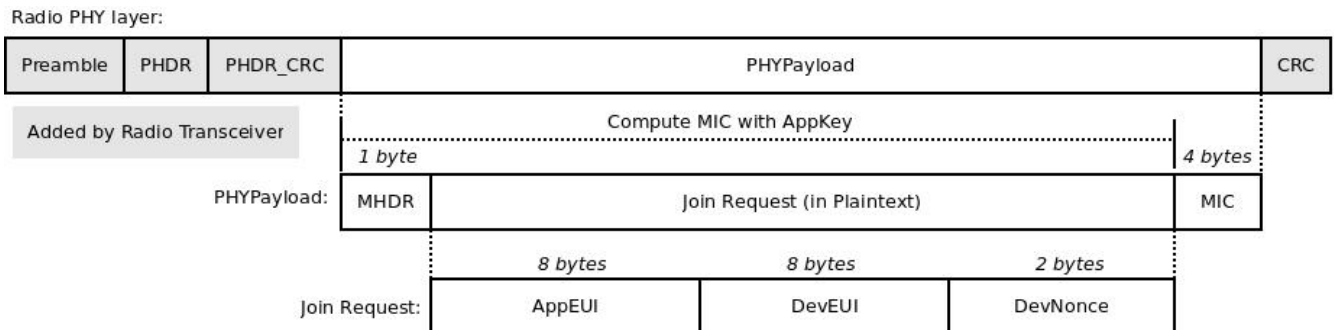
Στην καρτέλα LORAWAN FRAMES παρουσιάζονται τα μηνύματα LoRaWAN που προωθεί το gateway στον Network Server και αντίστροφα.



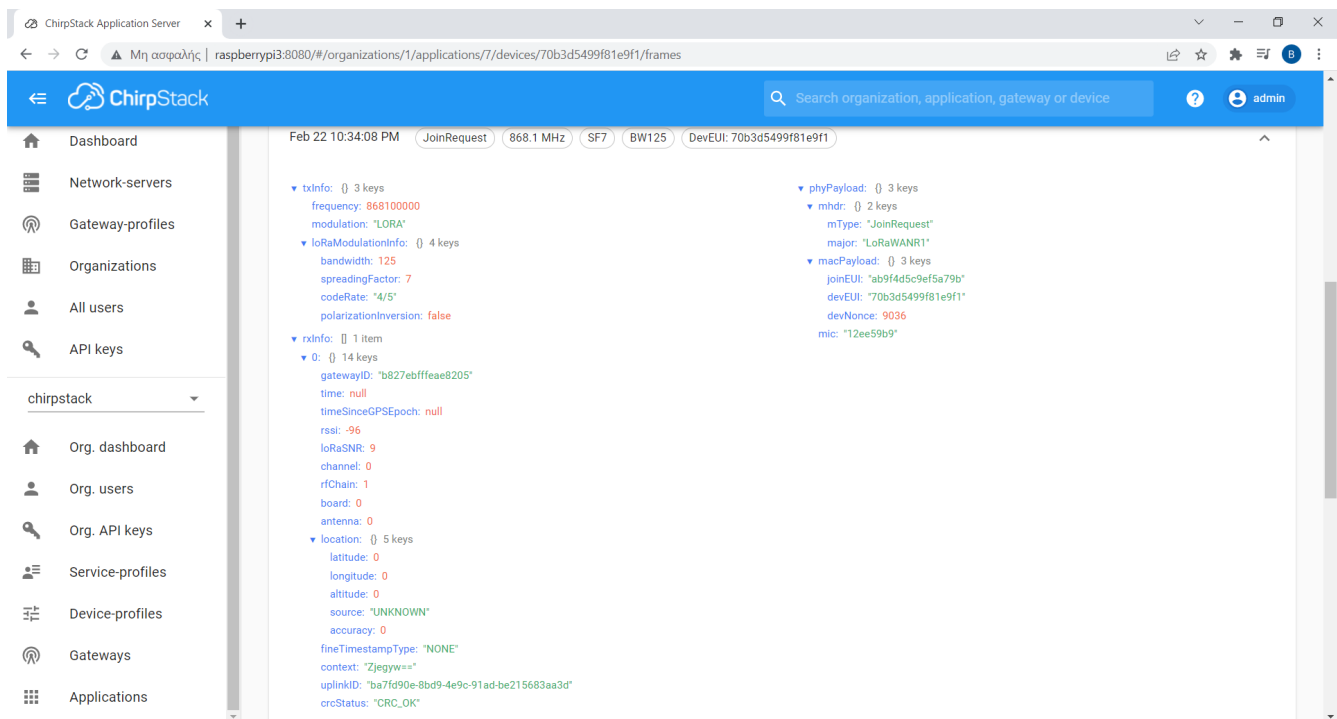
Εικόνα 6.20: Εμφάνιση μηνυμάτων LoRa που ανταλλάσσει η τερματική συσκευή

Βλέπουμε το μήνυμα Join Request, το οποίο ακολουθείται από ένα Join Accept και στην συνέχεια ακολουθούν μηνύματα που περιέχουν τα δεδομένα που στέλνει το node.

Αν επιλέξουμε το Join Request βλέπουμε τα περιεχόμενα του σε μορφή json.



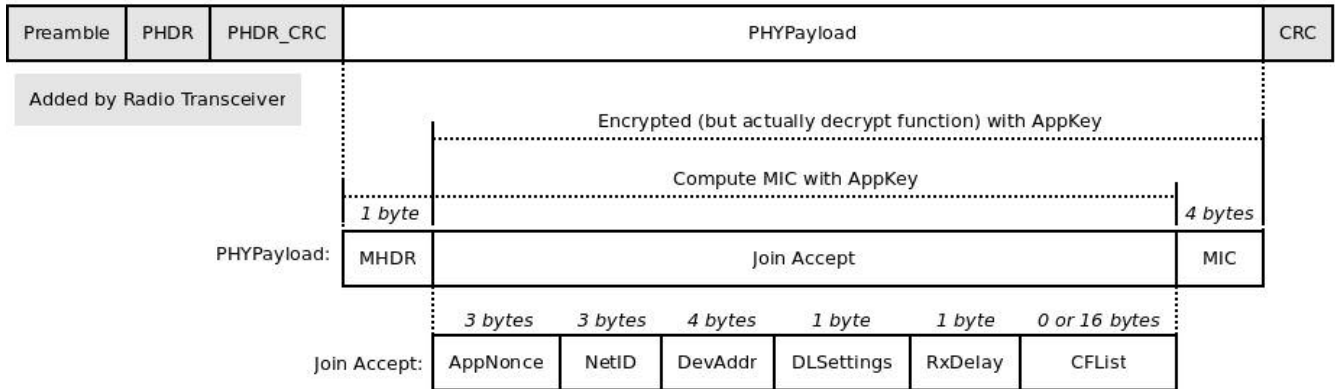
Σχήμα 6.2: Δομή μηνύματος Join Request [25]



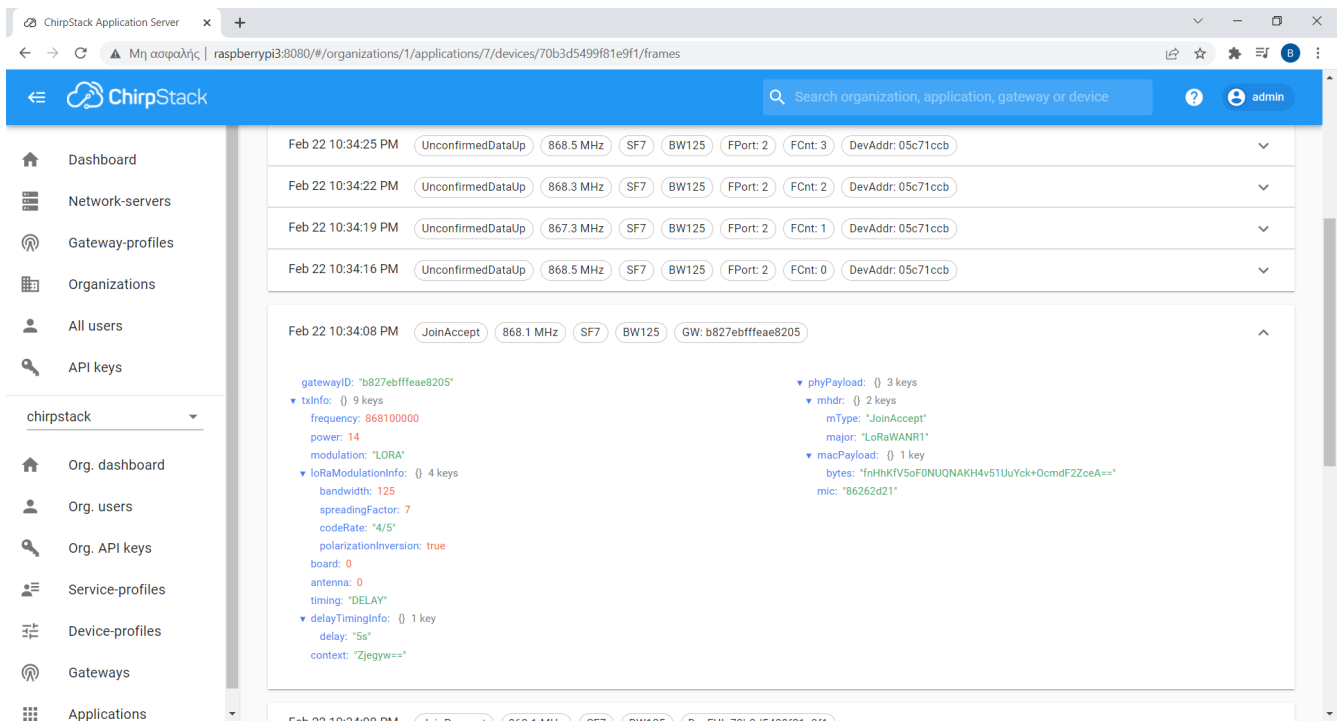
Εικόνα 6.21: Μήνυμα Join Request σε μορφή json

## Αντίστοιχα για το Join Accept:

Radio PHY layer:

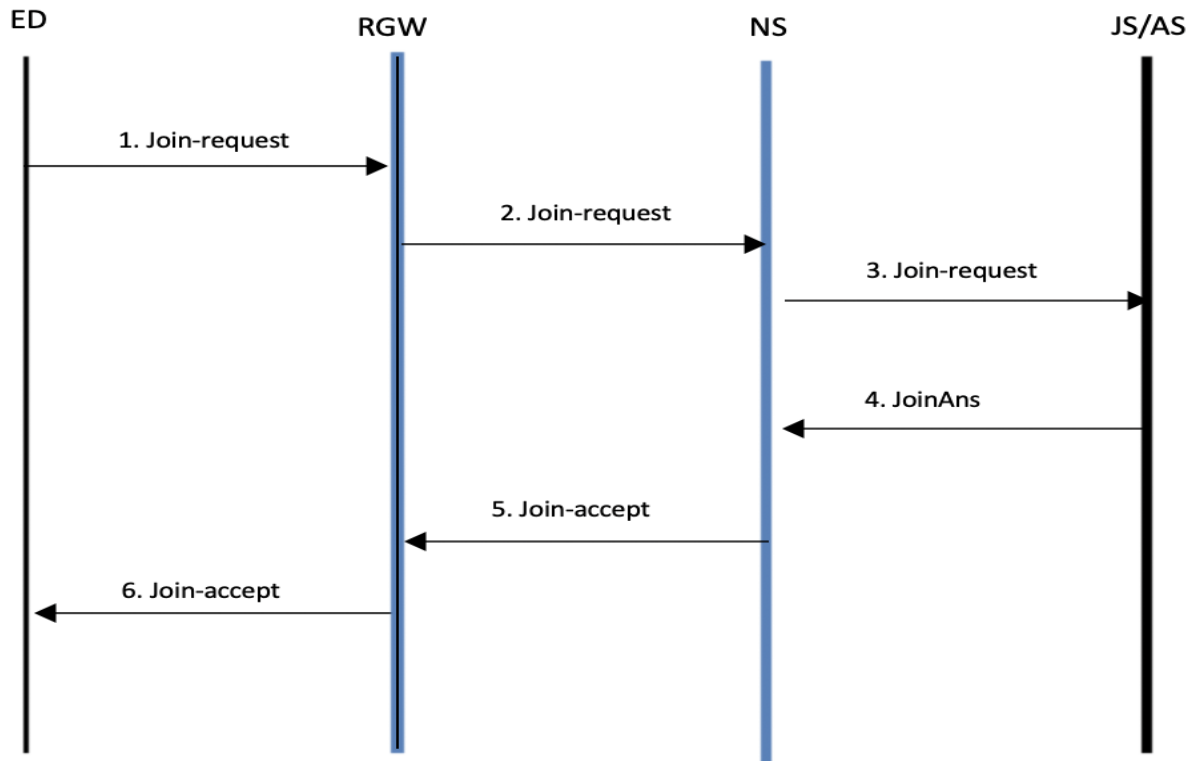


Σχήμα 6.3: Δομή μηνύματος Join Accept [25]



Εικόνα 6.22: Μήνυμα Join Accept σε μορφή json

Η διαδικασία που ακολουθήθηκε για το Join μπορεί να σχηματιστεί ως εξής:



Σχήμα 6.4: Διαδικασία Join

Το μήνυμα Join Request ταξιδεύει από την τερματική συσκευή στο gateway και από εκεί στον Network Server. Το gateway απλώς προωθεί το μήνυμα. Έπειτα ο Network Server το προωθεί στον Join Server ο οποίος είναι ο Application Server. Αν έχουμε κάνει με σωστό τρόπο την εγγραφή της συσκευής τότε ο Application/Join Server απαντά θετικά και παράγεται ένα μήνυμα Join Accept που ακολουθεί την αντίθετη πορεία από το Join request [33].



## Κεφάλαιο 7: Υλοποίηση OTA activation μέσω DNS

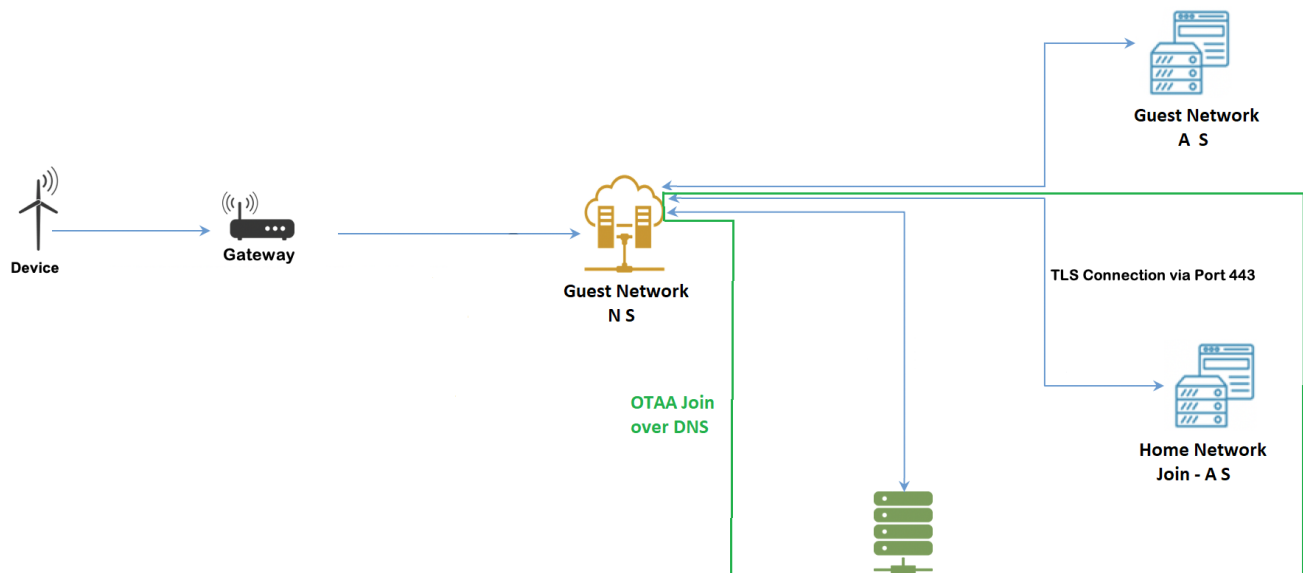
### 7.1 Εισαγωγή

Στην απλή περίπτωση αρχιτεκτονικής δικτύου LoRaWAN για να ενεργοποιηθεί μία τερματική συσκευή μέσω OTAA είναι απαραίτητο να δοθούν στον Application Server ,που λειτουργεί ως Join Server, τα DevEUI, AppEUI (JoinEUI) και AppKey. Το τελευταίο δεν μεταδίδεται ποτέ και χρησιμεύει για την δημιουργία των κλειδιών ασφαλείας της σύνδεσης.

Υπάρχουν περιπτώσεις που θέλουμε να χρησιμοποιήσουμε μια τερματική συσκευή σε δίκτυα που δεν το έχουμε ενεργοποιήσει. Μία προσέγγιση είναι να χρησιμοποιήσουμε έναν άλλο Join Server, στον οποίο έχει εγγραφεί η τερματική συσκευή, ώστε να πραγματοποιηθεί η διαδικασία του Join.

Ένας τρόπος είναι να χρησιμοποιηθεί ένας server DNS ο οποίος θα δίνει στον Network Server την διεύθυνση του Join server στον οποίο θα απευθύνει το Join Request.

Το γαλλικό ινστιτούτο AFNIC προσφέρει ένα DNS server για αυτή την διαδικασία και παρέχει οδηγίες στην διεύθυνση <https://github.com/AFNIC/IoTRoam-Tutorial>. Προτείνει την παρακάτω αρχιτεκτονική [34].



Σχήμα 7.1: Αρχιτεκτονική δικτύου LoRaWAN με χρήση OTA activation μέσω DNS

Πιο συγκεκριμένα γίνεται χρήση του πεδίο AppEUI (JoinEUI) το οποίο μετατρέπεται σε dns εγγραφή με την αντιστροφή του, την προσθήκη τελιών ανάμεσα στα ψηφία του και την προσθήκη τελικά του suffix `joineuis.iotreg.net` [35].

Θα ορίσουμε ως Guest Network το Raspberry που έχουμε χρησιμοποιήσει και ως Home Network ένα εικονικό μηχάνημα. Η τερματική συσκευή είναι εγγεγραμμένη στο Home Network και θέλουμε να τη συνδέσουμε με το Guest μέσω OTA activation με DNS.

Η τερματική συσκευή θα προσπαθήσει να ενεργοποιηθεί στο Guest Network αλλά αφού δεν θα έχουμε δώσει το AppKey δεν θα μπορεί να γίνει το activation μέσω του Join Server του Guest Network. Έτσι θα αποστέλλει ένα DNS query με τον AppEUI (JoinEUI) της συσκευής. Αν υπάρχει η κατάλληλη εγγραφή στο DNS server τότε θα στέλνει στον Network Server του Guest Network την διεύθυνση IP του Join Server του Home Network ο οποίος και θα απαντήσει με Join Accept. Στο τέλος η συσκευή ενεργοποιείται στο Guest Network και αρχίζει να αποστέλλει δεδομένα. Πρέπει να σημειωθεί ότι το Home Network χρησιμοποιήθηκε μόνο για την εξυπηρέτηση του Join Request.

## 7.2 Εγκατάσταση Home Network

Σε ένα εικονικό μηχάνημα του εργαστηρίου με στατική διεύθυνση IP (lora.cn.ntua.gr / 147.102.40.49) εγκαθιστούμε με τον τρόπο που έχει ήδη παρουσιαστεί ένα Chirpstack Network Server και ένα Chirpstack Application Server.

Στη συνέχεια εγγράφουμε την συσκευή στο Home Network με τον ίδιο τρόπο.

Στον configuration file του Chirpstack Network Server ορίζω ότι:

```
[join_server.default]

server="https://147.102.40.49:443"
```

Στο configuration file του Chirpstack Application Server ορίζω ότι:

```
[join_server]

bind="0.0.0.0:443"
```

Με αυτό τον τρόπο ορίζω ότι ο Join Server του Home Network θα “ακούει” στο port 443 (https).

## Εγκατάσταση Guest Network

Η εγκατάσταση έχει ήδη παρουσιαστεί. Στο configuration file του Chirpstack Network Server προσθέτω:

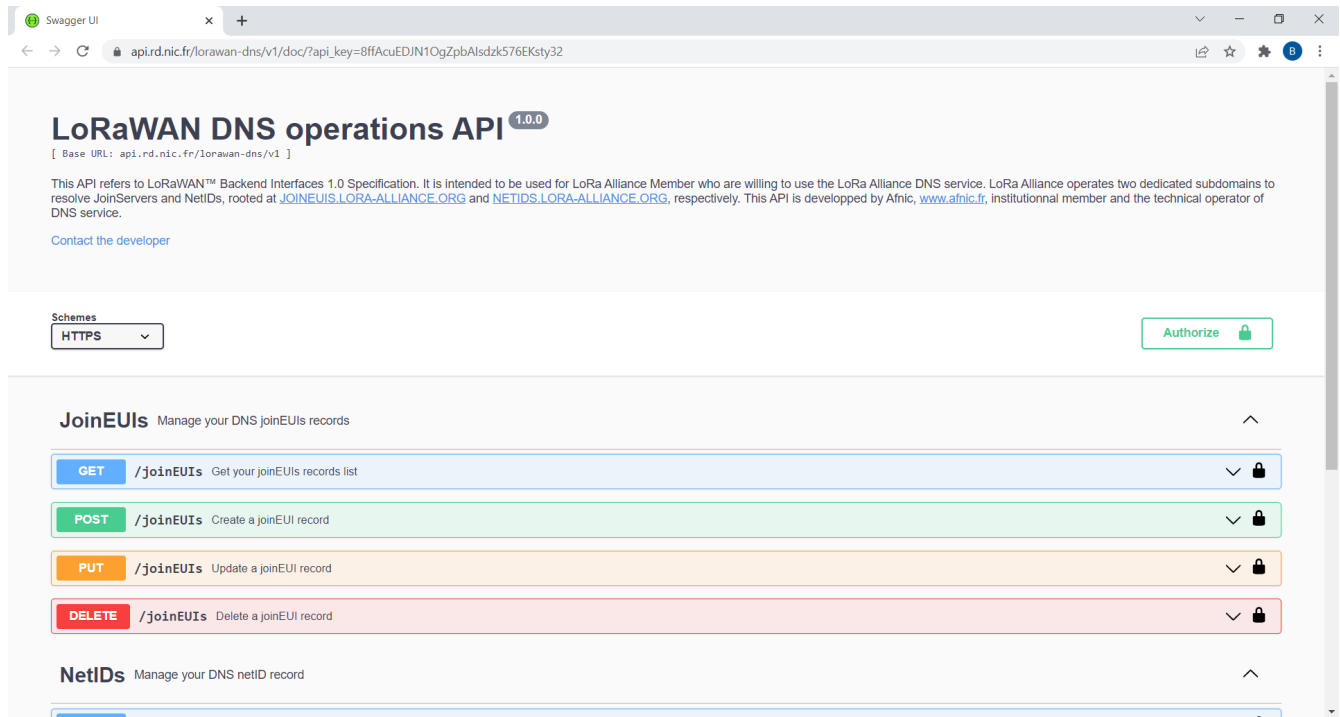
```
[join_server]

resolve_join_eui=true
resolve_domain_suffix=".joineui.iotreg.net"
```

Με αυτό τον τρόπο ενεργοποιώ την υπηρεσία OTAA over DNS και δίνω το κατάλληλο domain suffix που θα προστεθεί στο AppEUI (JoinEUI) ώστε να απαντήσει ο DNS server.

## 7.3 Αρχικοποίηση DNS Server

Ο DNS server παρέχεται από το AFNIC στην διεύθυνση <https://api.rd.nic.fr/lorawan-dns/v1/doc/> και προσφέρει ένα user interface στο οποίο μπορούμε να κάνουμε DNS εγγραφές.



Εικόνα 7.1: Αρχική σελίδα web interface του DNS server

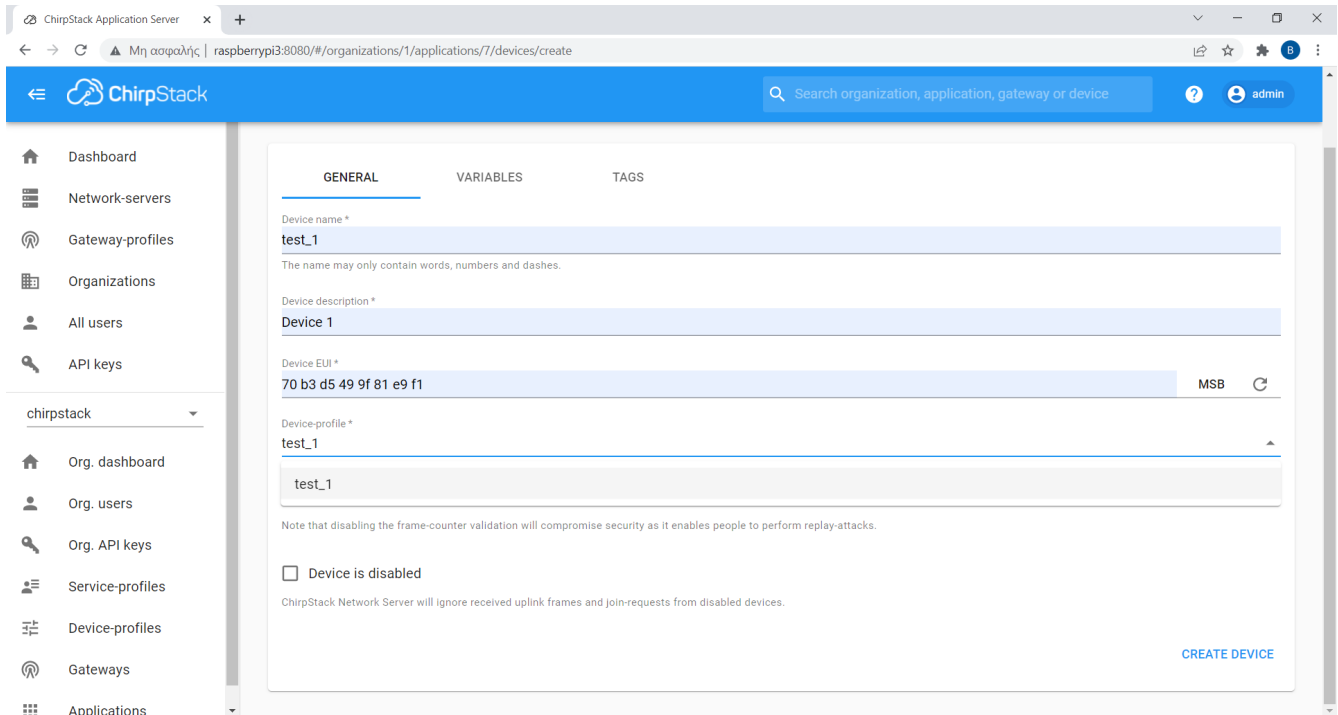
Επιλέγουμε POST και κάνουμε την παρακάτω εγγραφή:

```
{
  "name": "AB9F4D5C9EF5A79B",
  "type": "AAAA",
  "value": "147.102.40.49"
}
```

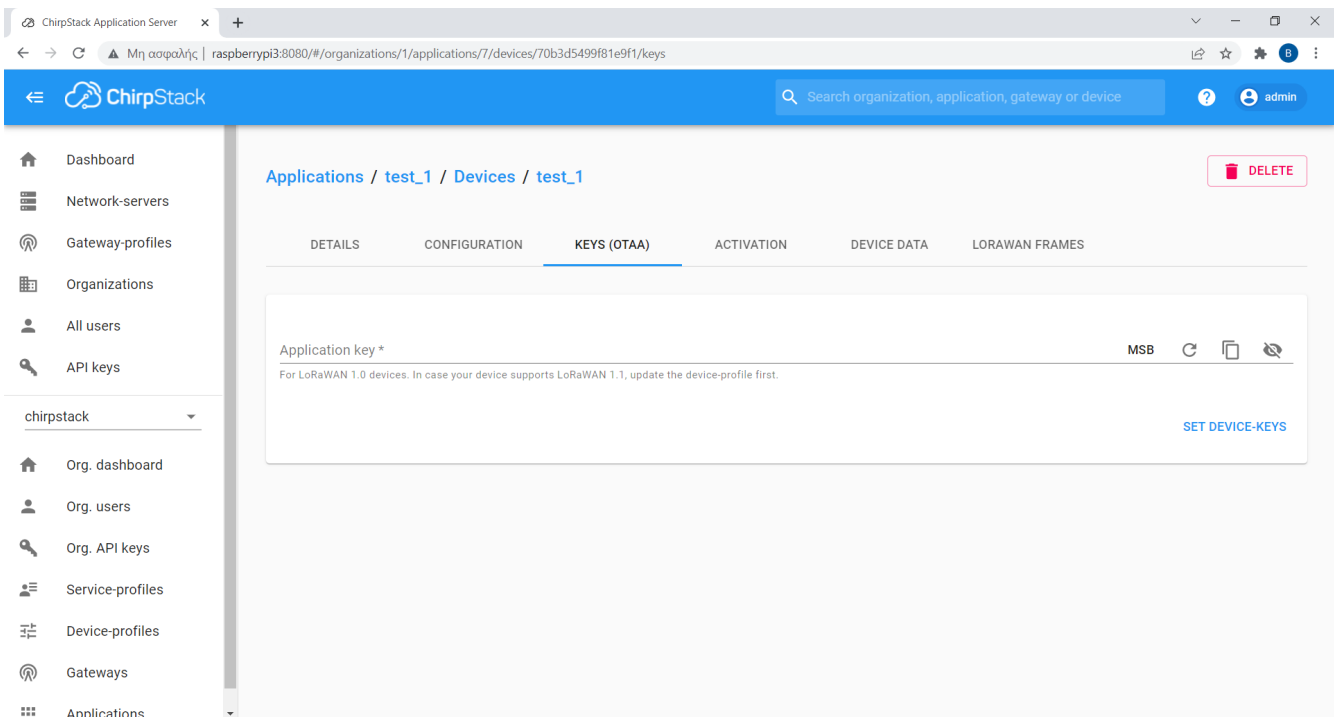
Στο πεδίο name ορίζουμε το AppEUI (JoinEUI) το οποίο έχουμε ορίσει στην τερματική συσκευή και στο πεδίο value την IP διεύθυνση του Join Server που θα χρησιμοποιηθεί.

## 7.4 Υλοποίηση OTAA over DNS

Στο Guest Network προσθέτουμε μία νέα συσκευή, με τον τρόπο που έχουμε παρουσιάσει, χωρίς όμως να δίνουμε το AppKey.



Εικόνα 7.2: Εγγραφή νέας τερματικής συσκευής



Εικόνα 7.3: Μη παροχή του AppKey στον Join Server

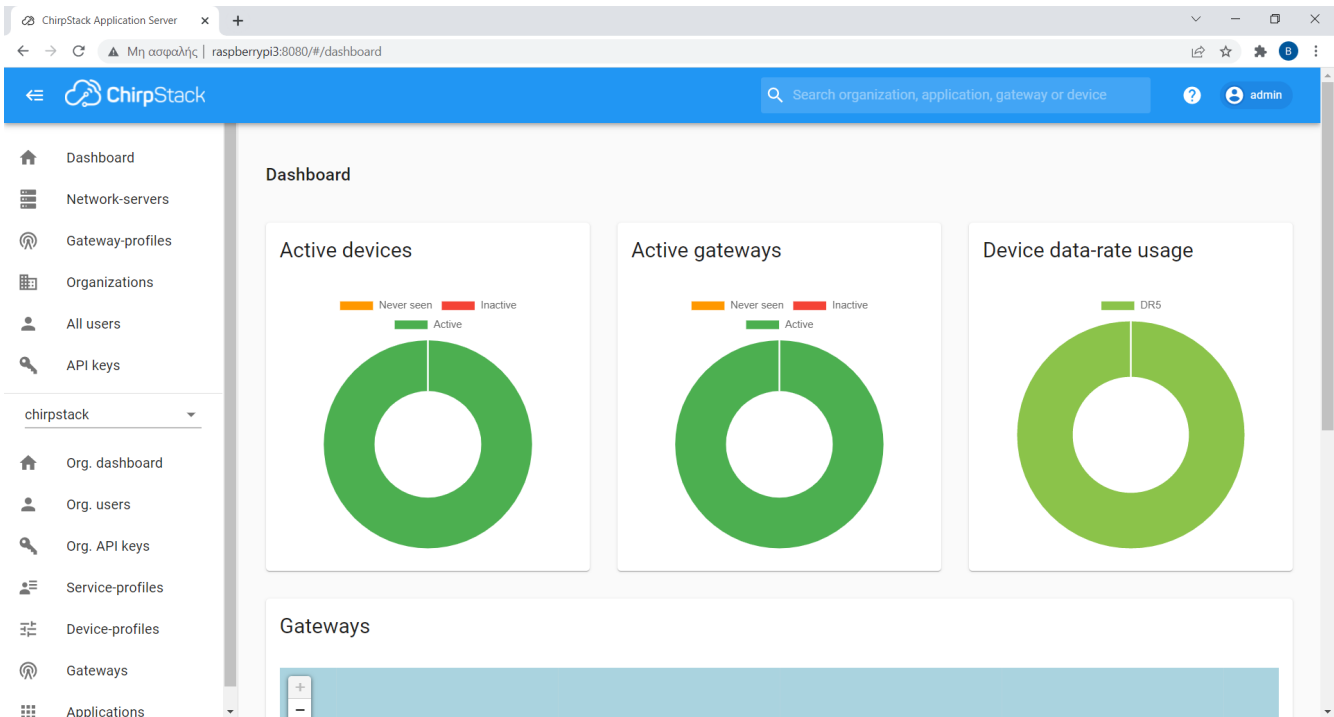
Όταν προσπαθήσουμε να κάνουμε activate την συσκευή ο Join Server δεν έχει το AppKey οπότε πραγματοποιεί ένα DNS query για να μάθει την διεύθυνση του κατάλληλου Join Server. Με την παρακάτω εντολή μπορούμε να δούμε την διαδικασία αυτή.

sudo tcpdump port 53

```
lora@lora:~$ sudo tcpdump port 53
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on ens160, link-type EN10MB (Ethernet), capture size 262144 bytes
12:24:01.512742 IP lora.cn.ntua.gr.44315 > psyche.cn.ece.ntua.gr.domain: 62186+ AAAA? b.9.7.a.5.f.e.9.c.5.d.4.f.9.b.a.joineuis.iotreg.net. (69)
12:24:01.512834 IP lora.cn.ntua.gr.34375 > psyche.cn.ece.ntua.gr.domain: 37119+ A? b.9.7.a.5.f.e.9.c.5.d.4.f.9.b.a.joineuis.iotreg.net. (69)
12:24:01.513201 IP psyche.cn.ece.ntua.gr.domain > lora.cn.ntua.gr.34375: 37119 1/3/4 A 147.102.40.49 (236)
12:24:01.513681 IP lora.cn.ntua.gr.55973 > psyche.cn.ece.ntua.gr.domain: 900+ PTR? 1.40.102.147.in-addr.arpa. (43)
12:24:01.514025 IP psyche.cn.ece.ntua.gr.domain > lora.cn.ntua.gr.55973: 900* 1/3/6 PTR psyche.cn.ece.ntua.gr. (273)
12:24:01.569447 IP psyche.cn.ece.ntua.gr.domain > lora.cn.ntua.gr.44315: 62186 0/1/0 (126)
12:24:01.571449 IP lora.cn.ntua.gr.32935 > psyche.cn.ece.ntua.gr.domain: 6814+ AAAA? b.9.7.a.5.f.e.9.c.5.d.4.f.9.b.a.joineuis.iotreg.net. (69)
12:24:01.571731 IP psyche.cn.ece.ntua.gr.domain > lora.cn.ntua.gr.32935: 6814 0/1/0 (126)
12:24:01.571763 IP lora.cn.ntua.gr.47750 > psyche.cn.ece.ntua.gr.domain: 60059+ A? b.9.7.a.5.f.e.9.c.5.d.4.f.9.b.a.joineuis.iotreg.net. (69)
12:24:01.572009 IP psyche.cn.ece.ntua.gr.domain > lora.cn.ntua.gr.47750: 60059 1/3/4 A 147.102.40.49 (236)
```

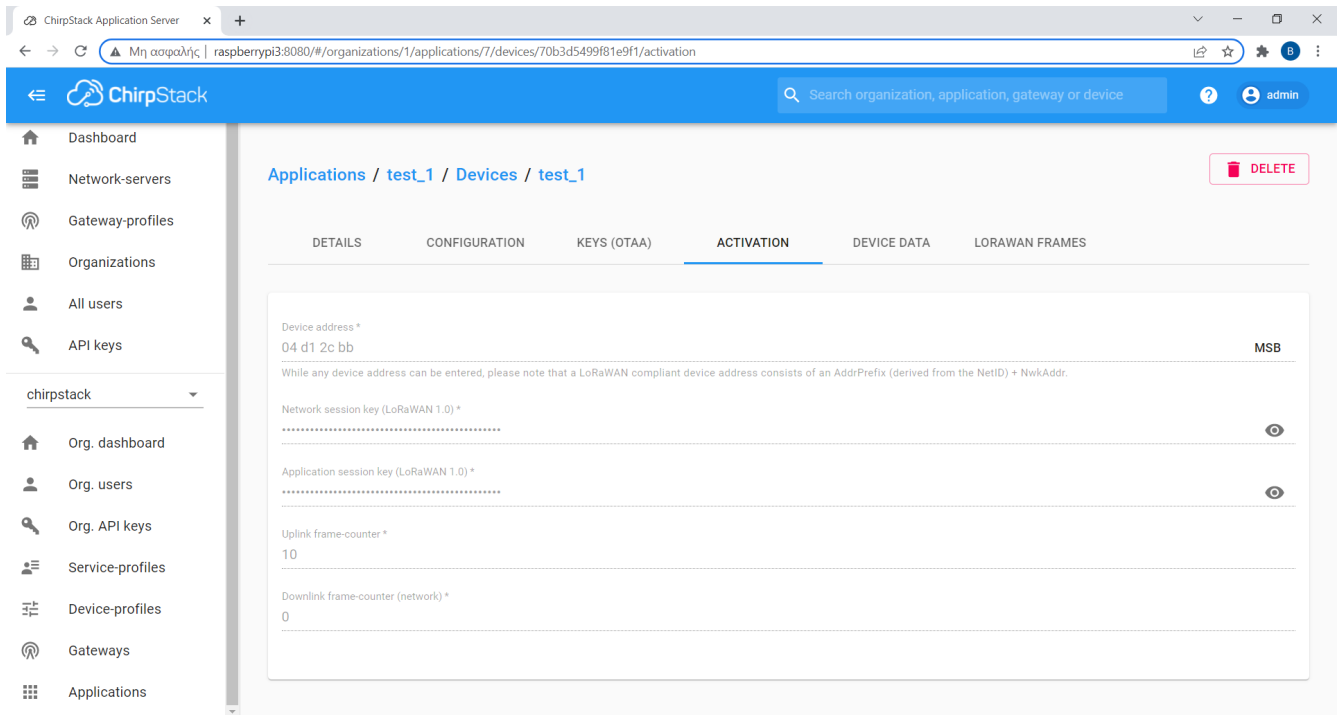
Εικόνα 7.4: Παρουσίαση της επικοινωνίας του Network Server με τον DNS server

Τελικά η συσκευή ενεργοποιείται στο Guest Network.

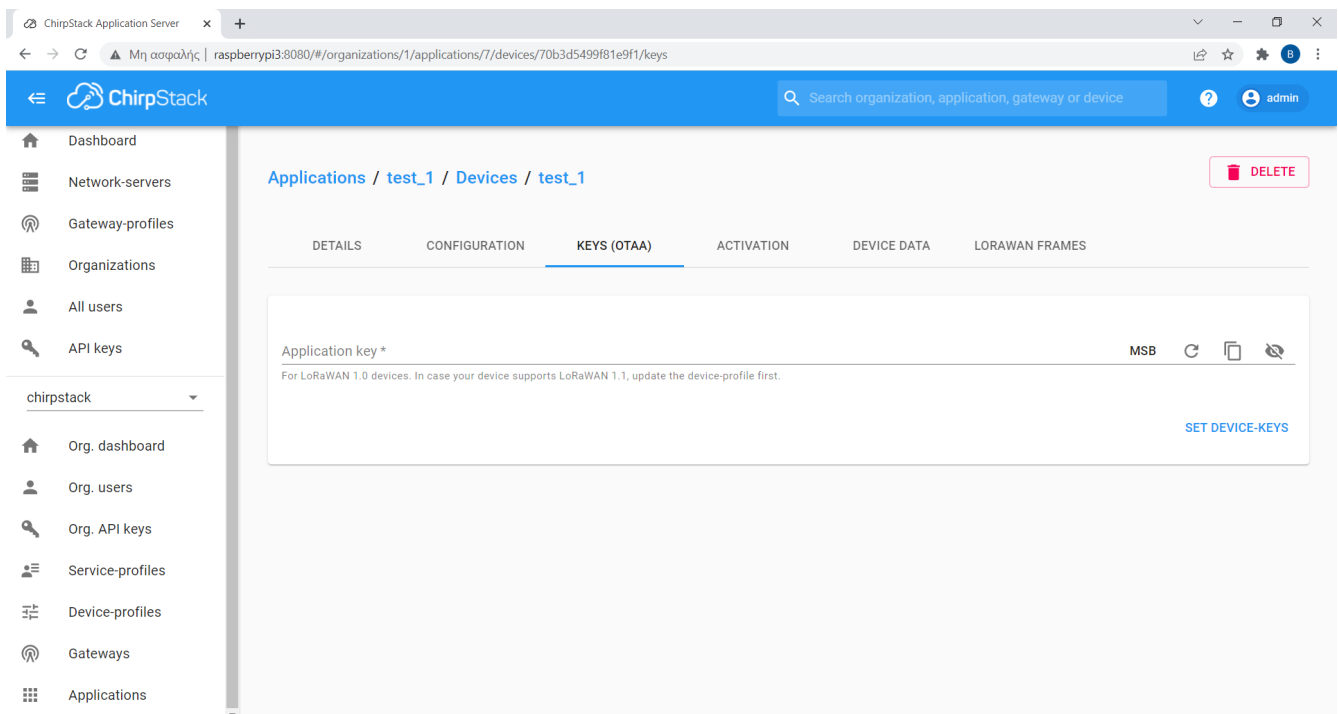


Εικόνα 7.5: Εμφάνιση της τερματικής συσκευής ως ενεργοποιημένη

Ενώ το AppKey δεν έχει δοθεί βλέπουμε ότι η ενεργοποίηση είναι επιτυχής.



Εικόνα 7.6: Ενεργοποίηση της συσκευής (Έχει πάρει διεύθυνση από το δίκτυο)



Εικόνα 7.7: Το AppKey δεν είναι γνωστό αλλά η συσκευή έχει ενεργοποιηθεί επιτυχώς

## Κεφάλαιο 8: Υλοποίηση Roaming

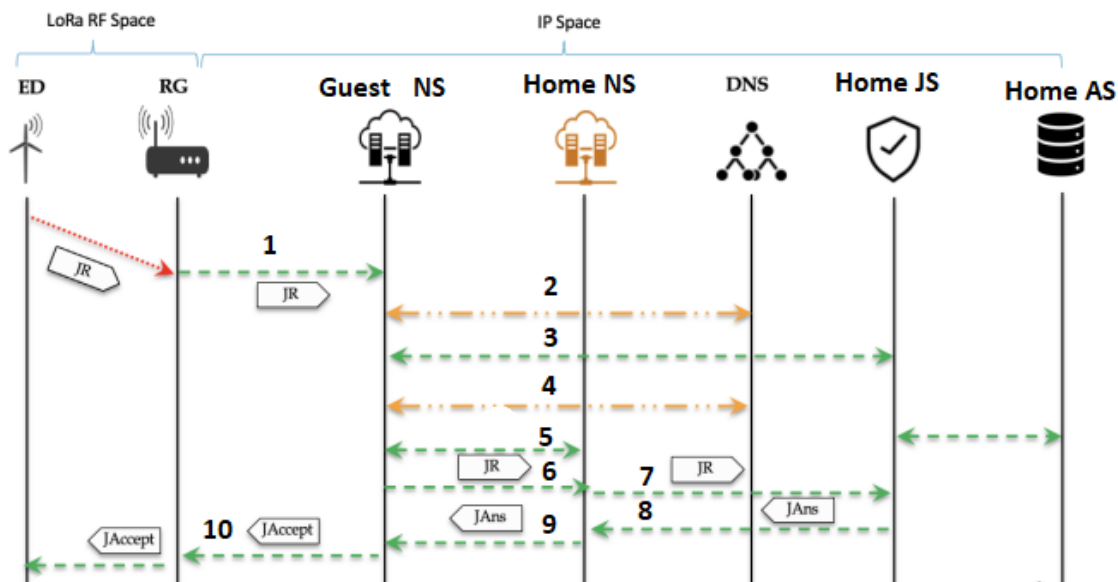
### 8.1 Εισαγωγή

Η παραπάνω αρχιτεκτονική μπορεί να επεκταθεί περαιτέρω για να υλοποιεί λειτουργίες roaming. Με λίγα λόγια η τερματική συσκευή είναι εγγεγραμμένη και ενεργοποιημένη στο Home Network αλλά βρίσκεται εκτός της εμβέλειας του. Βρίσκεται όμως εντός της εμβέλειας του δικτύου Guest Network στο οποίο είναι τελείως άγνωστη. Η λειτουργία roaming επιτρέπει την χρήση των υποδομών του δικτύου Guest Network για την επικοινωνία της τερματικής συσκευής με το δίκτυο Guest Network σαν να βρίσκεται εντός της εμβέλειας του.



Σχήμα 8.1: Αρχιτεκτονική δικτύου LoRaWAN σε λειτουργία roaming

Για να επιτευχθεί η λειτουργία του roaming η LoRa Alliance έχει εισάγει μία παραλλαγμένη διαδικασία Over-The-Air-Activation στο LoRaWAN 1.0.x, όπως παρουσιάστηκε στο Κεφάλαιο 4, για την επίτευξη Passive Roaming Activation [36]. Θεωρείται σίγουρο πως η τερματική συσκευή είναι ήδη ενεργοποιημένη στο Guest Network.



Σχήμα 8.2: Passive Roaming Activation

#### Βήμα 1:

Η τελική συσκευή αποστέλλει ένα μήνυμα Join Request το οποίο μέσω της πύλης του Guest Network

καταλήγει στον Network Server του Guest Network. Περιέχει τα πεδία DevEUI, AppEUI (JoinEUI) και DevNonce.

### **Βήμα 2:**

Ο Network Server του Guest Network καταλαβαίνει, λόγω του άγνωστου σε αυτόν DevEUI, ότι η τερματική συσκευή ανήκει σε άλλο δίκτυο. Επομένως πρέπει να του γίνει γνωστή η IP διεύθυνση του Join Server του Home Network.

Η αντιστοίχιση του JoinEUI με την IP Διεύθυνση του Join Server του Home Network μέσω ενός DNS query είναι μία λύση που εξασφαλίζει αυτή την ανάγκη. Η διαδικασία αυτή είναι το OTAA over DNS που έχει περιγραφεί στο προηγούμενο κεφάλαιο. Επομένως ο Network Server αποκτά την IP διεύθυνση του Join Server του Home Network.

### **Βήμα 3:**

Ο Network Server του Guest Network αποστέλλει στον Join Server του Home Network μήνυμα HomeNSReq με το DevEUI της τερματικής συσκευής.

Στην συνέχεια ο Join Server του Home Network αποστέλλει στον Network Server του Guest Network μήνυμα HomeNSAns που περιέχει το NetID του Home Network.

### **Βήμα 4:**

Ο Network Server του Guest Network πρέπει να αντιστοιχίσει το NetID με την διεύθυνση IP του Network Server του Home Network.

Αυτό μπορεί να γίνει είτε με αντίστοιχο τρόπο του Βήματος 2 δηλαδή μέσω DNS είτε με την απευθείας παροχής αυτής της διεύθυνσης στο configuration file του Network Server.

### **Βήμα 5:**

Ο Network Server του Guest Network αποστέλλει στον Network Server του Home Network μήνυμα ProfileReq με το DevEUI της τερματικής συσκευής.

Στην συνέχεια ο Network Server του Home Network αποστέλλει στον Network Server του Guest Network μήνυμα ProfileAns που σηματοδοτεί την έναρξη του roaming.

### **Βήμα 6:**

Ο Network Server του Guest Network αποστέλλει στον Network Server του Home Network μήνυμα PRStartReq που περιέχει το μήνυμα **Join Request**.

### **Βήμα 7:**

Ο Network Server του Home Network αποστέλλει στον Join Server του Home Network μήνυμα JoinReq που αποτελείται από:



**Μήνυμα Join Request**  
**DevEUI**  
**DevAddr**

όπως αυτά έχουν υπολογιστεί από τον Network Server του Home Network.

#### **Βήμα 8:**

Ο Join Server του Home Network αποστέλλει στον Network Server του Home Network μήνυμα JoinAns που περιέχει:

**Μήνυμα Join Accept**  
**NwkSKey**  
**AppSKey**

Τα κλειδιά παρήχθησαν μέσω του γνωστού στον Join Server του Home Network AppKey.

#### **Βήμα 9:**

Ο Network Server του Home Network αποστέλλει στον Network Server του Guest Network μήνυμα PRStartAns που περιέχει:

**Μήνυμα Join Accept**  
**DevEUI**

#### **Βήμα 10:**

Ο Network Server του Guest Network προωθεί στην τερματική συσκευή (μέσω της πύλης) το μήνυμα Join Accept όπως παράχθηκε από το Join Server του Home Network.

Η τερματική συσκευή παράγει τα κλειδιά NwkSKey και AppSKey.

Πλέον όλη η κίνηση που αφορά την τερματική συσκευή προωθείται από τον Network Server του Guest Network στον Network Server του Home Network, δίχως καμία επεξεργασία, μέσω της αντιστοίχισης NetID και IP διεύθυνσης του Network Server του Home Network η οποία έχει ληφθεί σύμφωνα με το Βήμα 4.

Ακολουθεί η παρουσίαση των επεμβάσεων που πρέπει να γίνουν στους Network και Application Servers των Home και Guest Networks για την επίτευξη του roaming όπως παρουσιάστηκε.

## **8.2 Αρχικοποίηση Home Network**

Στο configuration file του Network Server του Home Network προσθέτουμε τα παρακάτω:

```
net_id="000001"

[roaming]

resolve_netid_domain_suffix=".netids.iotreg.net"

[roaming.api]

bind="0.0.0.0:7443"

[roaming.default]

enabled=true
```

Όπως αναφέρθηκε ρόλο roaming server εκτελεί ο Network Server και επομένως ορίζεται η θύρα στην οποία απαντά στα αντίστοιχα αιτήματα.

Στο configuration file του Application Server του Home Network ορίζουμε το id του server:

```
id="2bbd0562-ae84-2dcb-c20f-6b9506f6812a"
```

### 8.3 Αρχικοποίηση Guest Network

Στο configuration file του Network Server του Guest Network προσθέτουμε τα παρακάτω:

```
net_id="000002"

[roaming.default]

enabled=true

    server="https://147.102.40.49:7443"

passive_roaming=true

passive_roaming_lifetime="24h"
```

Ορίζουμε την διεύθυνση του roaming server του Home Network.

Στο configuration file του Application Server του Guest Network προσθέτουμε τα παρακάτω:

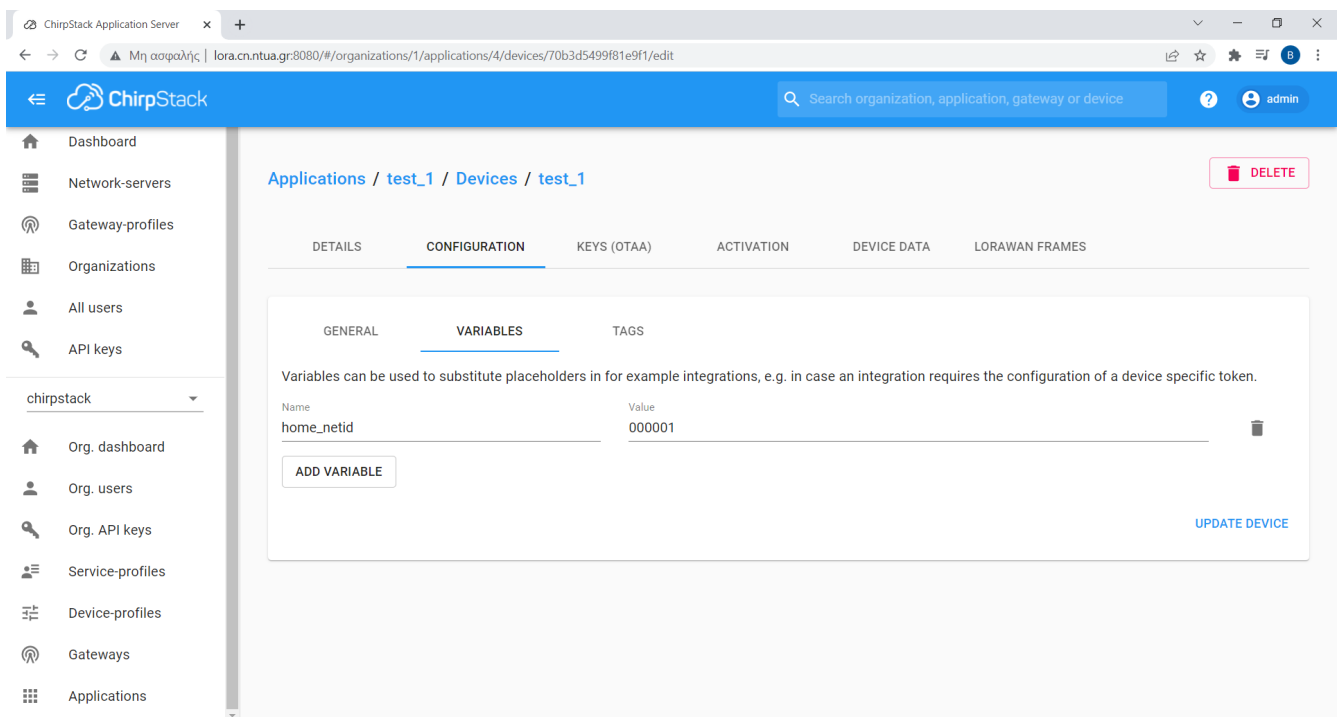
id="2bbd0562-ae84-2dcb-c20f-6b9506f6812b"

Παρατηρούμε ότι με τα διαφορετικά net\_id διαχωρίζουμε τα δύο δίκτυα έτσι ώστε να μπορεί υλοποιηθεί η υπηρεσία του roaming, γνωρίζοντας έτσι πιο δίκτυο είναι το Guest και το Home. Αντίστοιχα ορίζουμε και διαφορετικά id για τους Application Servers.

Για να υπάρχει ασφαλής επικοινωνία μεταξύ των Home και Guest Networks πρέπει να υπάρχουν τα κατάλληλα ssl x509 certificates. Το ινστιτούτο AFNIC προσφέρει ένα intermediate authority για την επιβεβαίωση τους καθώς και οδηγίες για την παραγωγή τους. Δεν γίνεται μεγαλύτερη ανάλυση γιατί εκφεύγει από την υλοποίηση της υποδομής LoRaWAN αλλά περιγράφεται στην σελίδα <https://github.com/AFNIC/IoTRoam-Tutorial/blob/master/Certificates-Tutorial.md>.

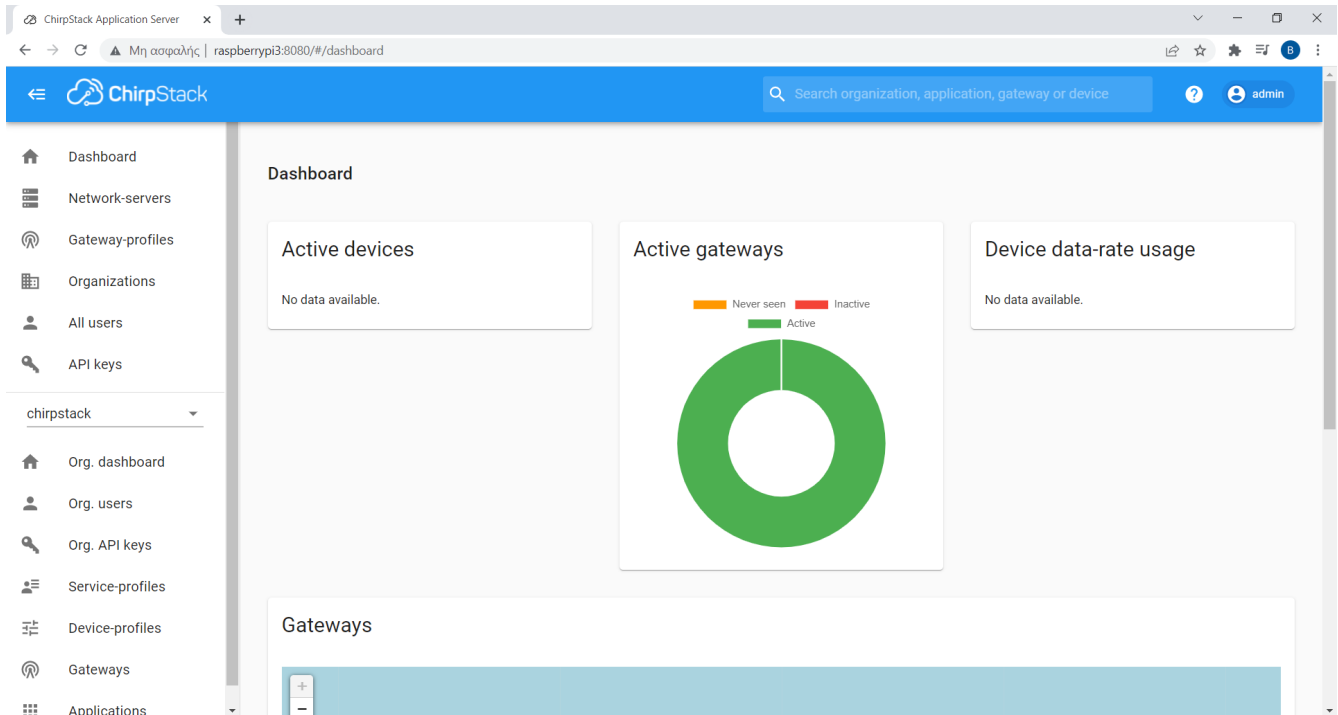
## 8.4 Υλοποίηση Roaming

Στο web interface του Application Server του Home Network επιλέγουμε την συσκευή και στην καρτέλα Variables προσθέτουμε την μεταβλητή home\_netid = 000001.

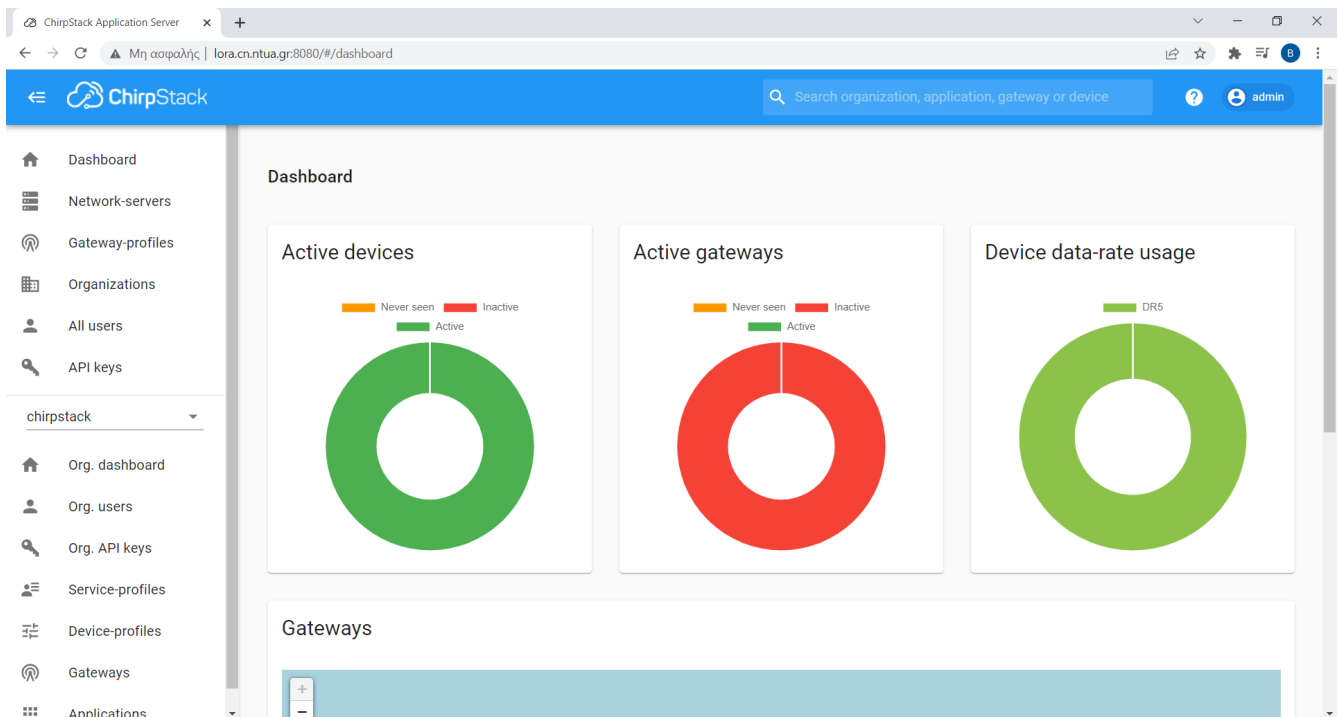


Εικόνα 8.1: Παροχή του κατάλληλου net\_id στην τερματική συσκευή

Ενεργοποιούμε την τερματική συσκευή και παρατηρούμε ότι παρόλο που δεν είναι εγγεγραμμένη στο Guest Network και στο Home Network δεν έχουμε συνδέσει κάποιο gateway αυτή ενεργοποιείται κανονικά σαν να βρίσκεται εντός της εμβέλειας του. Ουσιαστικά γίνεται χρήση των υποδομών (πύλη) του Guest Network για να συνδεθεί η τερματική συσκευή με το Home Network.



Εικόνα 8.2: Εμφάνιση του ενεργοποιημένου Gateway και της έλλειψης ενεργοποιημένης συσκευής στο Guest Network



Εικόνα 8.3: Εμφάνιση ως ενεργής της τερματικής συσκευής στο Home Network παρά την έλλειψη ενεργού gateway

Η υλοποίηση του roaming πραγματοποιείται μέσω της προώθησης όλων των μηνυμάτων Lora που λαμβάνει το gateway του Guest Network στο Home Network μέσω του Network Server του Guest Network που αναλαμβάνει αυτό τον ρόλο.

## Κεφάλαιο 9: Συμπεράσματα

### 9.1 Συμπεράσματα

Στα προηγούμενα κεφάλαια παρουσιάστηκαν τα βήματα υλοποίησης δύο δικτύων αρχιτεκτονικής LoRaWAN με χρήση του κατάλληλου εξοπλισμού και της πλατφόρμας ανοικτού κώδικα Chirpstack. Ειδικότερα στο κεφάλαιο 8 επιτεύχθηκε η λειτουργία roaming μεταξύ των δύο δικτύων αφού η τερματική συσκευή συνδέθηκε με το Home Network, μέσω του Guest Network, αν και βρίσκονταν ουσιαστικά εκτός της εμβέλειας του.

Συνεπώς στην παρούσα εργασία τίθεται το ερώτημα του μεγέθους της καθυστέρησης στην ενεργοποίηση μιας συσκευής με την διαδικασία που προτείνεται στο Κεφάλαιο 8 σε σχέση με την παραδοσιακή αρχιτεκτονική ενός δικτύου LoRaWAN.

Στον παρακάτω πίνακα βλέπουμε μετρήσεις για τον χρόνο ενεργοποίησης μιας τερματικής συσκευής με ή χωρίς roaming. Συγκεκριμένα μετρήθηκε ο χρόνος από την αποστολή ενός μηνύματος Join Request έως την λήψη του μηνύματος Join Accept. Αναφερόμαστε πάντα σε περιπτώσεις που η τερματική συσκευή είναι κοντά στην πύλη άρα δεν υπάρχουν καθυστερήσεις λόγω μη λήψης του μηνύματος από τις κεραιές της τερματικής συσκευής και της πύλης. Έγιναν διάφορες μετρήσεις και είχαμε τους παρακάτω μέσους όρους:

Ενεργοποίηση δίχως roaming	Ενεργοποίηση με roaming
5.12s	5.13s

Πίνακας 9.1 Χρόνος ενεργοποίησης τερματικής συσκευής με ή χωρίς roaming

Παρατηρούμε ότι ουσιαστικά δεν υπάρχει διαφορά όταν η συσκευή βρίσκεται σε λειτουργία roaming. Αυτό μπορεί να εξηγηθεί με το γεγονός ότι η επικοινωνία μεταξύ της πύλης και των υπόλοιπων στοιχείων του δικτύου LoRaWAN γίνεται μέσω διαδικτύου στο χώρο του IP ακόμα και αν έχουν εγκατασταθεί στο ίδιο μηχάνημα. Συνεπώς η καθυστέρηση που προσθέτει το DNS είναι ελάχιστη.

Πρέπει να τονιστεί ότι ο χρόνος των 5 s αντιστοιχεί στον χρόνο καθυστέρησης που αναφέραμε στο Κεφάλαιο 4 για συσκευές Class A που αντιστοιχεί στο χρόνο καθυστέρησης μέχρι να ανοίξει το παράθυρο λήψης και να ληφθεί το μήνυμα Join Request από την τερματική συσκευή. Συνεπώς ακόμα και αν σταλεί νωρίτερα το μήνυμα Join Accept στην πύλη θα πρέπει να περιμένει μέχρι να περάσει αυτός ο χρόνος.

Συνεπώς καταλήγουμε στο συμπέρασμα ότι η λειτουργία roaming δεν επιβαρύνει τον χρόνο ενεργοποίησης παρά τα βήματα που προσθέτει στην διαδικασία της ενεργοποίησης της τερματικής συσκευής. Παρέχει όμως την δυνατότητα της επέκτασης της εμβέλειας ενός δικτύου LoRaWAN κάνοντας χρήση των πυλών άλλων δικτύων αξιοποιώντας αποδοτικότερα τους διατιθέμενους πόρους.

Τελικά, αξίζει να τονιστεί ότι προϋπόθεση για την λειτουργία του roaming είναι η κατάλληλη ρύθμιση από τους διαχειριστές των δικτύων στους αντίστοιχους διακομιστές. Ο χρήστης της τερματικής συσκευής απλά εγγράφει και ενεργοποιεί την συσκευή του στο δίκτυο LoRaWAN που

χρησιμοποιεί και απολαμβάνει τα οφέλη του roaming αφού αυτό επιτυγχάνεται αυτόματα. Πρόκειται για αναλογία με το roaming που γνωρίζουμε στην κινητή τηλεφωνία.

## **9.2 Μελλοντικές Επεκτάσεις**

Η εφαρμογή θα μπορούσε να επεκταθεί στο μέλλον με την προσθήκη λειτουργία αυθεντικοποίησης (authentication) για την είσοδο στο web interface των Application Servers των δικτύων LoRaWAN. Ακόμα θα μπορούσε να χρησιμοποιηθούν άλλες πλατφόρμες για την υλοποίηση των στοιχείων του δικτύου LoRaWAN, αντί του Chirpstack, όπως το The Things Network. Τέλος μπορεί να διερευνηθεί η περίπτωση του roaming μεταξύ δικτύων υλοποιημένων σε Chirpstack και σε The Things Network ή ακόμα και μεταξύ δικτύων LoRaWAN και δικτύων άλλων τεχνολογιών όπως 4G/5G και WiFi.

## Παράρτημα

### main.py

Πρόκειται για τον κώδικα σε γλώσσα Python που υλοποιεί την ενεργοποίηση OTAA και την αποστολή 3 bytes κάθε 5 s από το LoPy

```
from network import LoRa
import socket
import time
import ubinascii

# Initialise LoRa in LORAWAN mode.
# Please pick the region that matches where you are using the device:
# Asia = LoRa.AS923
# Australia = LoRa.AU915
# Europe = LoRa.EU868
# United States = LoRa.US915
lora = LoRa(mode=LoRa.LORAWAN, region=LoRa.EU868)

# create an OTAA authentication parameters, change them to the provided
credentials
app_eui = ubinascii.unhexlify('AB9F4D5C9EF5A79B')
app_key = ubinascii.unhexlify('14B0287A186B75B7B4D5D8A7FB30549C')
#uncomment to use LoRaWAN application provided dev_eui
#dev_eui = ubinascii.unhexlify('70B3D549938EA1EE')

# Uncomment for US915 / AU915 & Pygate
# for i in range(0,8):
#     lora.remove_channel(i)
# for i in range(16,65):
#     lora.remove_channel(i)
# for i in range(66,72):
#     lora.remove_channel(i)

# join a network using OTAA (Over the Air Activation)
#uncomment below to use LoRaWAN application provided dev_eui
lora.join(activation=LoRa.OTAA, auth=( app_eui, app_key), timeout=0)
#lora.join(activation=LoRa.OTAA, auth=(dev_eui, app_eui, app_key),
timeout=0)
```

```

# wait until the module has joined the network
while not lora.has_joined():
    time.sleep(2.5)
    print('Not yet joined...')

print('Joined')
# create a LoRa socket
s = socket.socket(socket.AF_LORA, socket.SOCK_RAW)

# set the LoRaWAN data rate
s.setsockopt(socket.SOL_LORA, socket.SO_DR, 5)

# (waits for the data to be sent and for the 2 receive windows to expire)
while True:
    s.setblocking(True)
# send some data
    s.send(bytes([0x01, 0x02, 0x03]))

# make the socket non-blocking
# (because if there's no data received it will block forever...)
    s.setblocking(False)

# get any data received (if any...)
#data = s.recv(64)
#print(data)

```

## Configuration file Network Server Home Network

(/etc/chirpstack-network-server/chirpstack-network-server.toml)

```

# This configuration configures ChirpStack Network Server for the EU868
band using a MQTT
# broker to communicate with the gateways. Many options and defaults have
been
# omitted for simplicity.
#
# See https://www.chirpstack.io/network-server/install/config/ for a full
# configuration example and documentation.

# PostgreSQL settings.

```



```

#
# Please note that PostgreSQL 9.5+ is required.
[postgresql]
# PostgreSQL dsn (e.g.:
postgres://user:password@hostname/database?sslmode=disable).
#
# Besides using an URL (e.g.
'postgres://user:password@hostname/database?sslmode=disable')
# it is also possible to use the following format:
# 'user=chirpstack_ns dbname=chirpstack_ns sslmode=disable'.
#
# The following connection parameters are supported:
#
# * dbname - The name of the database to connect to
# * user - The user to sign in as
# * password - The user's password
# * host - The host to connect to. Values that start with / are for unix
domain sockets. (default is localhost)
# * port - The port to bind to. (default is 5432)
# * sslmode - Whether or not to use SSL (default is require, this is not
the default for libpq)
# * fallback_application_name - An application_name to fall back to if
one isn't provided.
# * connect_timeout - Maximum wait for connection, in seconds. Zero or
not specified means wait indefinitely.
# * sslcert - Cert file location. The file must contain PEM encoded data.
# * sslkey - Key file location. The file must contain PEM encoded data.
# * sslrootcert - The location of the root certificate file. The file
must contain PEM encoded data.
#
# Valid values for sslmode are:
#
# * disable - No SSL
# * require - Always SSL (skip verification)
# * verify-ca - Always SSL (verify that the certificate presented by the
server was signed by a trusted CA)
# * verify-full - Always SSL (verify that the certification presented by
the server was signed by a trusted CA and the server host name matches
the one in the certificate)
dsn="postgres://chirpstack_ns:dbnpassword@localhost/chirpstack_ns?sslmod

```

```

e=disable"
automigrate=true
# Redis settings
#
# Please note that Redis 2.6.0+ is required.
[redis]
# Redis url (e.g. redis://user:password@hostname/0)
#
# For more information about the Redis URL format, see:
# https://www.iana.org/assignments/uri-schemes/prov/redis
url="redis://localhost:6379"

# Network-server settings.
[network_server]
# Network identifier (NetID, 3 bytes) encoded as HEX (e.g. 010203)
net_id="000001"

# LoRaWAN regional band configuration.
#
# Note that you might want to consult the LoRaWAN Regional Parameters
# specification for valid values that apply to your region.
# See: https://www.lora-alliance.org/lorawan-for-developers
[network_server.band]
name="EU868"

# LoRaWAN network related settings.
[network_server.network_settings]

# Extra channel configuration.
#
# Use this for LoRaWAN regions where it is possible to extend the by
default
# available channels with additional channels (e.g. the EU band).
# The first 5 channels will be configured as part of the OTAA
join-response
# (using the CFList field).
# The other channels (or channel / data-rate changes) will be
(re)configured

```

```
# using the NewChannelReq mac-command.
#
[[network_server.network_settings.extra_channels]]
frequency=867100000
min_dr=0
max_dr=5

[[network_server.network_settings.extra_channels]]
frequency=867300000
min_dr=0
max_dr=5

[[network_server.network_settings.extra_channels]]
frequency=867500000
min_dr=0
max_dr=5

[[network_server.network_settings.extra_channels]]
frequency=867700000
min_dr=0
max_dr=5

[[network_server.network_settings.extra_channels]]
frequency=867900000
min_dr=0
max_dr=5

# Class B settings
[network_server.network_settings.class_b]
# Ping-slot data-rate.
ping_slot_dr=0

# Ping-slot frequency (Hz)
#
# Set this to 0 to use the default frequency plan for the configured
region
# (which could be frequency hopping).
ping_slot_frequency=0
```

```

# Network-server API
#
# This is the network-server API that is used by ChirpStack Application
Server or other
# custom components interacting with ChirpStack Network Server.
[network_server.api]
# ip:port to bind the api server
bind="0.0.0.0:8000"
ca_cert="/home/lora/certificates/certs/ca/ca.pem"

tls_cert="/home/lora/certificates/certs/network-server/api/server/network
-server-api-server-combined.pem"

tls_key="/home/lora/certificates/certs/network-server/api/server/network-
server-api-server-key.pem"

# Backend defines the gateway backend settings.
#
# The gateway backend handles the communication with the gateway(s)
part of
# the LoRaWAN network.
[network_server.gateway.backend]
# Backend
type="mqtt"

# MQTT gateway backend settings.
#
# This is the backend communicating with the LoRa gateways over a
MQTT broker.
[network_server.gateway.backend.mqtt]
# MQTT topic templates for the different MQTT topics.
#
# The meaning of these topics are documented at:
# https://www.chirpstack.io/gateway-bridge/
#
# The meaning of these topics are documented at:

```

```

# https://www.chirpstack.io/gateway-bridge/
#
# The default values match the default expected configuration of the
# ChirpStack Gateway Bridge MQTT backend. Therefore only change these
values when
# absolutely needed.

# Event topic template.
event_topic="gateway/+/event/+"

# Command topic template.
#
# Use:
# * "{{ .GatewayID }}" as an substitution for the LoRa gateway ID
# * "{{ .CommandType }}" as an substitution for the command type
command_topic_template="gateway/{{ .GatewayID }}/command/{{
.CommandType }}"

# MQTT server (e.g. scheme://host:port where scheme is tcp, ssl or
ws)
server="tcp://localhost:1883"

# Connect with the given username (optional)
username=""

# Connect with the given password (optional)
password=""

# Metrics collection settings.
[metrics]
# Timezone
#
# The timezone is used for correctly aggregating the metrics (e.g. per
hour,
# day or month).
# Example: "Europe/Amsterdam" or "Local" for the the system's local time
zone.
timezone="Local"

```

```

# Join-server settings.
[join_server]
  resolve_join_eui=true
  #resolve_domain_suffix=".joineuis.iotreg.net"
  # Default join-server settings.
  #
  # This join-server will be used when resolving the JoinEUI is set to
false
  # or as a fallback when resolving the JoinEUI fails.
  [join_server.default]
  # hostname:port of the default join-server
  #
  # This API is provided by ChirpStack Application Server.
  server="https://147.102.40.49:8443"
  ca_cert="/home/lora/certificates/certs/ca/ca.pem"

tls_cert="/home/lora/certificates/certs/application-server/join-api/client/application-server-join-api-client-combined.pem"

tls_key="/home/lora/certificates/certs/application-server/join-api/client/application-server-join-api-client-key.pem"

  [roaming]
  resolve_netid_domain_suffix=".netids.iotreg.net"

  [roaming.api]
  bind="0.0.0.0:7443"
  ca_cert="/home/lora/certificates/certs/ca/ca.pem"
  tls_cert="/home/lora/certificates/certs/network-server/roaming/000001/server/server-combined.pem"
  tls_key="/home/lora/certificates/certs/network-server/roaming/000001/server/server-key.pem"

  [roaming.default]
  enabled=true
  passive_roaming=true
  passive_roaming_lifetime="24h"
  ca_cert="/home/lora/certificates/certs/ca/ca.pem"

```

```
tls_cert="/home/lora/certificates/certs/network-server/roaming/000001/client/client-combined.pem"
tls_key="/home/lora/certificates/certs/network-server/roaming/000001/client/client-key.pem"
```

## Configuration file Application Server Home Network Server

(/etc/chirpstack-application-server/chirpstack-application-server.toml)

```
[general]
log_level=4

[postgresql]
dsn="postgres://chirpstack_as:dbaspassword@localhost/chirpstack_as?sslmode=disable"
automigrate=true

[redis]
url="redis://localhost:6379"

# Application-server settings.
[application_server]
id="2bbd0562-ae84-2dcb-c20f-6b9506f6812a"

[application_server.integration]
enabled=["mqtt"]

[application_server.integration.mqtt]
uplink_topic_template="application/{{ .ApplicationID }}/device/{{ .DevEUI }}/rx"
downlink_topic_template="application/{{ .ApplicationID }}/device/{{ .DevEUI }}/tx"
join_topic_template="application/{{ .ApplicationID }}/device/{{ .DevEUI }}/join"
ack_topic_template="application/{{ .ApplicationID }}/device/{{ .DevEUI }}/ack"
error_topic_template="application/{{ .ApplicationID }}/device/{{ .DevEUI }}/error"
status_topic_template="application/{{ .ApplicationID }}/device/{{ .DevEUI }}/status"
```

```

location_topic_template="application/{{ .ApplicationID }}/device/{{
.DevEUI }}/location"

# MQTT server (e.g. scheme://host:port where scheme is tcp, ssl or
ws)
server="tcp://localhost:1883"
# Connect with the given username (optional)
username=""
# Connect with the given password (optional)
password=""

[application_server.api]
# ip:port to bind the api server
bind="0.0.0.0:8001"

public_host="https://147.102.40.49:8001"
ca_cert="/home/lora/certificates/certs/ca/ca.pem"

tls_cert="/home/lora/certificates/certs/application-server/api/server/app
lication-server-api-server-combined.pem"

tls_key="/home/lora/certificates/certs/application-server/api/server/appl
ication-server-api-server-key.pem"

[application_server.external_api]
bind="0.0.0.0:8080"
# http server TLS certificate (optional)
tls_cert=""

# http server TLS key (optional)
tls_key=""

# JWT secret used for api authentication / authorization
# You could generate this by executing 'openssl rand -base64 32' for
example
jwt_secret="xpAbnfT7S4BuFr976JU60VoxsIc2BL8GyfrVyp59ofc="

[join_server]
## ip:port to bind the join-server api interface to
bind="0.0.0.0:8443"

```



```
ca_cert="/home/lora/certificates/certs/ca/ca.pem"  
tls_cert="/home/lora/certificates/certs/application-server/join-api/server/application-server-join-api-server-combined.pem"  
tls_key="/home/lora/certificates/certs/application-server/join-api/server/application-server-join-api-server-key.pem"
```

## Configuration file Network Server Guest Network

(/etc/chirpstack-network-server/chirpstack-network-server.toml)

```
# See https://www.chirpstack.io/network-server/install/config/ for a full  
# configuration example and documentation.  
[general]  
log_level=4  
  
# PostgreSQL settings.  
[postgresql]  
dsn="postgres://chirpstack_ns:dbnspassword@localhost/chirpstack_ns?sslmode=disable"  
  
# Redis settings  
[redis]  
url="redis://localhost:6379"  
  
# Network-server settings.  
[network_server]  
net_id="000002"  
  
[network_server.band]  
name="EU868"  
  
[network_server.network_settings]  
[[network_server.network_settings.extra_channels]]  
frequency=867100000  
min_dr=0  
max_dr=5  
  
[[network_server.network_settings.extra_channels]]  
frequency=867300000  
min_dr=0
```

```

max_dr=5

[[network_server.network_settings.extra_channels]]
frequency=867500000
min_dr=0
max_dr=5

[[network_server.network_settings.extra_channels]]
frequency=867700000
min_dr=0
max_dr=5

[[network_server.network_settings.extra_channels]]
frequency=867900000
min_dr=0
max_dr=5

[network_server.network_settings.class_b]
ping_slot_dr=0
ping_slot_frequency=0

[network_server.api]
# ip:port to bind the api server
bind="0.0.0.0:8000"
ca_cert="/home/pi/certificates/certs/ca/ca.pem"

tls_cert="/home/pi/certificates/certs/network-server/api/server/network-server-api-server-combined.pem"

tls_key="/home/pi/certificates/certs/network-server/api/server/network-server-api-server-key.pem"

[network_server.gateway.backend]
type="mqtt"

[network_server.gateway.backend.mqtt]
event_topic="gateway/+/event/+"
command_topic_template="gateway/{{ .GatewayID }}/command/{{
.CommandType }}"
server="tcp://localhost:1883"

```

```

username=""
password=""

[metrics]
timezone="Local"

# Join-server settings.
[join_server]
# Settings for DNS resolution
    #resolve_join_eui=true
    resolve_join_eui=false

#resolve_domain_suffix=".joineuis.iotreg.net"

[[join_server.servers]]
    join_eui="AB9F4D5C9EF5A79B"
    server="https://147.102.40.49:8443/join/endpoint"
    ca_cert="/home/pi/certificates/certs/ca/ca.pem"
tls_cert="/home/pi/certificates/certs/application-server/join-api/client/
application-server-join-api-client-combined.pem"

tls_key="/home/pi/certificates/certs/application-server/join-api/client/a
pplication-server-join-api-client-key.pem"

# Default join-server settings.
[join_server.default]
server="https://localhost:8003"
ca_cert="/home/pi/certificates/certs/ca/ca.pem"

tls_cert="/home/pi/certificates/certs/application-server/join-api/client/
application-server-join-api-client-combined.pem"

tls_key="/home/pi/certificates/certs/application-server/join-api/client/a
pplication-server-join-api-client-key.pem"

[roaming]
#resolve_netid_domain_suffix=".netids.iotreg.net"

[roaming.api]
#bind="0.0.0.0:443"

```

```

ca_cert="/home/pi/certificates/certs/ca/ca.pem"
tls_cert="/home/pi/certificates/certs/network-server/roaming/000002/server/server-combined.pem"
tls_key="/home/pi/certificates/certs/network-server/roaming/000002/server/server-key.pem"

[roaming.default]
enabled=true
  server="https://147.102.40.49:7443"
passive_roaming=true
passive_roaming_lifetime="24h"

```

## Configuration file Application Server Guest Network

(/etc/chirpstack-application-server/chirpstack-application-server.toml)

```

[general]
log_level=4

[postgresql]
dsn="postgres://chirpstack_as:dbaspassword@localhost/chirpstack_as?sslmode=disable"

[redis]
url="redis://localhost:6379"
# Application-server settings.
[application_server]
id="2bbd0562-ae84-2dcb-c20f-6b9506f6812b"

[application_server.integration]
enabled=["mqtt"]

[application_server.integration.mqtt]
uplink_topic_template="application/{{ .ApplicationID }}/device/{{ .DevEUI }}/rx"
downlink_topic_template="application/{{ .ApplicationID }}/device/{{ .DevEUI }}/tx"
join_topic_template="application/{{ .ApplicationID }}/device/{{ .DevEUI }}/join"
ack_topic_template="application/{{ .ApplicationID }}/device/{{

```

```

.DevEUI }}/ack"
    error_topic_template="application/{{ .ApplicationID }}/device/{{
.DevEUI }}/error"
    status_topic_template="application/{{ .ApplicationID }}/device/{{
.DevEUI }}/status"
    location_topic_template="application/{{ .ApplicationID }}/device/{{
.DevEUI }}/location"

# MQTT server (e.g. scheme://host:port where scheme is tcp, ssl or
ws)
server="tcp://localhost:1883"
# Connect with the given username (optional)
username=""
# Connect with the given password (optional)
password=""

[application_server.api]
# ip:port to bind the api server
bind="0.0.0.0:8001"

ca_cert="/home/pi/certificates/certs/ca/ca.pem"

tls_cert="/home/pi/certificates/certs/application-server/api/server/applic
ation-server-api-server-combined.pem"

tls_key="/home/pi/certificates/certs/application-server/api/server/applic
ation-server-api-server-key.pem"

[application_server.external_api]
bind="0.0.0.0:8080"

# http server TLS certificate (optional)
tls_cert=""

# http server TLS key (optional)
tls_key=""

# JWT secret used for api authentication / authorization
# You could generate this by executing 'openssl rand -base64 32' for
example

```

```
jwt_secret="UYr6rx8DQu8FGst8WtnyuEdj3F/6vNoJA1/eIjKqLqc="
```

```
[join_server]
```

```
# ip:port to bind the join-server api interface to
```

```
bind="0.0.0.0:8003"
```

## Configuration file Gateway Bridge

(`/etc/chirpstack-gateway-bridge/chirpstack-gateway-bridge.toml`)

```
# This configuration provides a Semtech UDP packet-forwarder backend and  
# integrates with a MQTT broker. Many options and defaults have been  
omitted
```

```
# for simplicity.
```

```
#
```

```
# See https://www.chirpstack.io/gateway-bridge/install/config/ for a full  
# configuration example and documentation.
```

```
# Gateway backend configuration.
```

```
[backend]
```

```
# Backend type.
```

```
type="semtech_udp"
```

```
# Semtech UDP packet-forwarder backend.
```

```
[backend.semtech_udp]
```

```
# ip:port to bind the UDP listener to
```

```
#
```

```
# Example: 0.0.0.0:1700 to listen on port 1700 for all network  
interfaces.
```

```
# This is the listener to which the packet-forwarder forwards its data
```

```
# so make sure the 'serv_port_up' and 'serv_port_down' from your
```

```
# packet-forwarder matches this port.
```

```
udp_bind = "0.0.0.0:1700"
```

```
# Integration configuration.
```

```
[integration]
```

```
# Payload marshaler.
```

```
#
```

```

# This defines how the MQTT payloads are encoded. Valid options are:
# * protobuf: Protobuf encoding
# * json:      JSON encoding (easier for debugging, but less compact than
'protobuf')
marshaller="protobuf"

# MQTT integration configuration.
[integration.mqtt]
# Event topic template.
event_topic_template="gateway/{{ .GatewayID }}/event/{{ .EventType }}"

# Command topic template.
command_topic_template="gateway/{{ .GatewayID }}/command/#"

# MQTT authentication.
[integration.mqtt.auth]
# Type defines the MQTT authentication type to use.
#
# Set this to the name of one of the sections below.
type="generic"

# Generic MQTT authentication.
[integration.mqtt.auth.generic]
# MQTT server (e.g. scheme://host:port where scheme is tcp, ssl or
ws)

#server="tcp://147.102.40.49:1883"

server="tcp://localhost:1883"

# Connect with the given username (optional)
username=""

# Connect with the given password (optional)

```

## Βιβλιογραφία

- [1] *What is the Internet of Things (IoT)?*. [online] Available at: <<https://www.oracle.com/internet-of-things/what-is-iot/>>
- [2] *Internet of Things (IoT)*. [online] Available at: <<https://www.trendmicro.com/vinfo/us/security/definition/internet-of-things>>
- [3] *Internet of Things — mPython board 2.2.2 documentation*. [online] Available at: <<https://mpython.readthedocs.io/en/master/tutorials/advance/iot/>>
- [4] Gillis, A. S., 2022. *What is IoT (Internet of Things) and How Does it Work?* [online] Available at: <<https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>>
- [5] <https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/>
- [6] *Types of IoT Networks*. [online] Available at: <<https://www.fogwing.io/types-of-iot-networks/>>
- [7] Shea, S., 2017. *What is LPWAN (low-power wide area network)?*. [online] Available at: <<https://internetofthingsagenda.techtarget.com/definition/LPWAN-low-power-wide-area-network>>
- [8] 2020. *What is LPWAN?*. [online] Available at: <<https://www.avsystem.com/blog/LPWAN/>>
- [9] *LPWAN Technology for IoT*. [online] Available at: <<https://behrtech.com/lpwan-technology/>>
- [10] Aufranc, J., 2015. *Comparison Table of Low Power WAN Standards for Industrial Applications*. [online] Available at: <<https://www.cnx-software.com/2015/09/21/comparison-table-of-low-power-wan-standards-for-industrial-applications/>>
- [11] *What are LoRa® and LoRaWAN®?*. [online] Available at: <<https://lora-developers.semtech.com/documentation/tech-papers-and-guides/lora-and-lorawan/>>
- [12] *AN1200.22 LoRa™ Modulation Basics*, 2nd ed. Semtech Corporation, 2015 [online]. Available at: <<https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf>>
- [13] 2018. *What is LoRa?*. [online] Available at: <<https://www.everythingrf.com/community/what-is-lora>>
- [14] *Spread spectrum - Wikipedia*. [online] <[https://en.wikipedia.org/wiki/Spread\\_spectrum](https://en.wikipedia.org/wiki/Spread_spectrum)>
- [15] 2022. *Understanding Spread Spectrum for Communications*. [online] Available at: <<https://www.ni.com/en-us/innovations/white-papers/06/understanding-spread-spectrum-for-communications.html>>
- [16] 2012. *CSS (ISO 24730-5) Measurement of distances without tape measure and wires*. [online] Available at: <https://sudonull.com/post/136810-CSS-ISO-24730-5-Measurement-of-distances-without-tape-measure-and-wires>>



- [17] A. Thomas and N. V. Eldhose, “Scalability concerns of chirp spread spectrum for LPWAN applications,” *International Journal of Ad hoc, Sensor & Ubiquitous Computing*, vol. 10, no. 01, pp. 1–11, 2019. [Online]. Available at: <<https://airccj.org/cseconf/library/pdffiles/10119ijasuc01.pdf>>
- [18] *Chirp\_spread\_spectrum - Wikipedia*. [online] Available at: <[https://en.wikipedia.org/wiki/Chirp\\_spread\\_spectrum](https://en.wikipedia.org/wiki/Chirp_spread_spectrum)>
- [19] *LORA / LORAWAN TUTORIAL Data Rate, Chip Rate, Symbol Rate Chip Duration & Symbol Duration*. [online] Available at: [https://www.mobilefish.com/download/lora/lora\\_part15.pdf](https://www.mobilefish.com/download/lora/lora_part15.pdf)
- [20] *LoRaWAN Frequency Bands*. [online] Available at: <<https://www.3glteinfo.com/lora/lorawan-frequency-bands/>>
- [21] *What are LoRa and LoRaWAN?*. [online] Available at: <<https://www.thethingsnetwork.org/docs/lorawan/what-is-lorawan/>>
- [22] *What is LoRaWAN® Specification - LoRa Alliance®*. [online] Available at: <<https://loralliance.org/about-lorawan/>>
- [23] *End Device Activation*. [online] Available at: <<https://www.thethingsnetwork.org/docs/lorawan/end-device-activation/>>
- [24] Sornin, N., Luis, M., Eirich, T., Kramp, T. and Hersent, O., 2015. *LoRaWAN™ Specification*. 1st ed. LoRa Alliance. [online] Available at: <<https://www.frugalprototype.com/wp-content/uploads/2016/08/LoRaWAN-Specification-1R0.pdf>>
- [25] *LoRaWAN 1.0.2 - HackMD*. [online] Available at: <<https://hackmd.io/@starnight/S1kg6Ymo-#LoRaWAN-102>>
- [26] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, “Understanding the limits of Lorawan,” *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017. [online] Available at: <<https://arxiv.org/pdf/1607.08011.pdf>>
- [27] *Limitations*. [online] Available at: <<https://www.thethingsnetwork.org/docs/lorawan/limitations/>>
- [28] *The ChirpStack project - ChirpStack open-source LoRaWAN® Network Server*. [online] Available at: <<https://www.chirpstack.io/project/>>
- [29] *ChirpStack architecture - ChirpStack open-source LoRaWAN® Network Server*. [online] Available at: <<https://www.chirpstack.io/project/architecture/>>
- [30] *Raspberry Pi 3 Model B*. [online] Available at: <<https://www.raspberrypi.com/products/raspberry-pi-3-model-b/>>

- [31] *SX1308 Raspberry Pi LoRa Gateway Board*. [online] Available at: <<https://www.tindie.com/products/will123321/sx1308-raspberry-pi-lora-gateway-board/#product-name>>
- [32] *LoPy*. [online] Available at: <<https://www.botnroll.com/en/others/2552-lopy.html>>
- [33] Afnic, 2020, *IoTRoam-tutorial/applicationserver-setup.md at master · AFNIC/IOTROAM-tutorial, GitHub*. [online]. Available at: <<https://github.com/afnic/IoTRoam-Tutorial/blob/master/ApplicationServer-Setup.md>>
- [34] Afnic, 2020, *IoTRoam-tutorial/OTAA-using-dns.md at master · AFNIC/IOTROAM-tutorial, GitHub*. [online]. Available at: <<https://github.com/afnic/IoTRoam-Tutorial/blob/master/OTAA-Using-DNS.md>>
- [35] Afnic, 2020, *IoTRoam-tutorial/DNS-Setup.md at master · AFNIC/IOTROAM-tutorial, GitHub*. [online]. Available at: <<https://github.com/AFNIC/IoTRoam-Tutorial/blob/master/DNS-Setup.md>>
- [36] *LoRaWAN® Backend Interfaces Technical Specification (TS002-1.1.0)*, LoRa Alliance , 2020 [online]. Available at: <[https://lora-alliance.org/wp-content/uploads/2020/11/TS002-1.1.0\\_LoRaWAN\\_Backend\\_Interfaces.pdf](https://lora-alliance.org/wp-content/uploads/2020/11/TS002-1.1.0_LoRaWAN_Backend_Interfaces.pdf)>