Ελληνικό Μεσογειακό Πανεπιστήμιο
Τμήμα Ηλεκτρονικών Μηχανικών

Σύστημα εναλλακτικής επικοινωνίας και πληροφόρησης σε περιπτώσεις κρίσεων.

Μεταπτυχιακή Διπλωματική Εργασία
του
Ιωάννη Σαραντόπουλου

Επιβλέπων: Εμμανουήλ Αντωνιδάκης
Καθηγητής Ελληνικό Μεσογειακό Πανεπιστήμιο

Επιτροπή: Ιωάννης Βαρδιάμπασης
Αν. Καθηγητής Ελληνικό Μεσογειακό Πανεπιστήμιο
Ευάγγελος Κόκκινος
Αν. Καθηγητής Ελληνικό Μεσογειακό Πανεπιστήμιο

7 Φεβρουαρίου 2020

Περίληψη

Στην εργασία αυτή προτείνεται και αναπτύσσεται η αρχιτεκτονική και τα βασικά μέρη ενός συστήματος επικοινωνιών, που θα μπορούσε να χρησιμοποιηθεί σαν εναλλακτικός τρόπος επικοινωνίας και ελέγχου σε περιπτώσεις κρίσεων, όπου άλλες μορφές δικτύων (κυρίως κυψελωδών δικτύων) δεν θα είναι διαθέσιμες. Η αρχιτεκτονική και η λύση που προτείνεται μπορεί να είναι χρήσιμη γενικά σε LoRaWAN δίκτυα ελέγχου πολλών συσκευών μέσο multicast εκπομπών. Στην εργασία συγκρίνονται διάφορές τεχνολογίες ασυρμάτων δικτύων. Γίνεται ιδιαίτερη αναφορά στην τεχνολογία LoRa και LoRaWAN όπου και θα χρησιμοποιήσουμε. Κατά την ανάλυση και το σχεδιασμό του συστήματος παρατηρούμε ότι το σύστημα είναι αποδοτικότερο με χρήση πολυεκπομπών Multicast. Η γρήγορη εναλλαγή όμως των παραμέτρων του συστήματος και οι ιδιαίτερες συνθήκες που πρέπει να αντιμετωπιστούν μας αναγκάζουν να στέλνουμε πολλά ίδια μηνύματα σε διαφορετικούς χρήστες κάθε φορά, αυτό μας οδηγεί σε μια πιο ανατρεπτική θεώρηση του συστήματος , έτσι μετά από ερευνά, καταλήξαμε και υλοποιήσαμε μια μέθοδο Broadcast Encryption και συγκεκριμένα την μέθόδο Subset Difference πάνω από ένα δίκτυο LoRaWAN. Έτσι χρησιμοποιούμε LoRaWAN Multicast εκπομπές σε επίπεδο δικτύου αλλά σε επίπεδο εφαρμογής , υλοποιούμε την τεχνική Subset Diferrence που κάνει πιο ευέλικτη την επιλογή των χρηστών που θα λάβουν τις πληροφορίες. Τέλος αναπτύξαμε μια σειρά από εφαρμογές και κατασκευές που υλοποιούν το σχεδιασμό και τις ιδιαίτερες απαιτήσεις του συστήματος που περιγράψαμε.

Λέξεις κλειδιά: πολυεκπομπές, κρίσεις, Δίκτυα Ευρείας Περιοχής Χαμηλής Ισχύος, LoRa τεχνολογία, Κρυπτογράφηση πολυεκπομπής, Broadcast Encryption , LoRaWan τεχνολογία, Subset Difference

Abstract

In this thesis are proposed and developed the architecture and the basic components of an alternate comunication system in crisis situations.This system is useful when other comunication systems have crashed. Our proposals and architecture are useful generally in LoRaWan Networks that control devices with multicast transmitters. This work compares various wireless network technologies. Particular reference is made to the technology of LoRa and LoRaWAN, of which we will use. When analyzing and designing the system, we observed that the system is more efficient when using Multicast transmitters. The fast changes of system parameters and the necessity to send the same messages to different users each time, pushed us to research for a more efficient way.Thereafter we chose to use Broadcast Encryption method over LoRaWan network. The method we applied is the Subset Difference method. Thus we utilized LoRaWan Multicast transmitters on the network level and on the application level we used the Subset Diferrence method. Consequently,this makes it more flexible to select the number of users recieving messages. Conclusively, we developed a variety of applications that implemented our system.

Keywords: Multicast. crisis, LPWA, LoRa, LoRaWan,Broadcast Encryption, Subset Diferrence

Ευχαριστίες

Στο σημείο αυτό, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κύριο Αντωνιδάκη Εμμανουήλ για την υποστήριξη και καθοδήγηση που μου προσέφερε.
Ιδιαίτερες ευχαριστίες θα ήθελα να απευθύνω στο φίλο και εξαίρετο επαγγελματία Ρηγάκη Ηρακλή για την απλόχερη υποστήριξη του σε όλους τους τομείς. Στη παροχή υλικοτεχνικής υποδομής , σε χρήσιμες συμβουλές και τεχνικές γνώσεις που μου προσέφερε και κυρίως για την ενθάρρυνση και ψυχολογική υποστήριξη σε όλη την διάρκεια της εργασίας.
Τέλος, θα ήθελα εκφράσω την ευγνωμοσύνη μου στην οικογένεια μου ,την σύζυγο μου Ελένη και τις κόρες μου Ελευθερία και Αναστασία για όλη τη στήριξη, τη συμπαράσταση και την κατανόησή τους, καθ' όλη τη διάρκεια των σπουδών μου.

Σύστημα εναλλακτικής επικοινωνίας και πληροφόρησης σε περιπτώσεις κρίσεων.

Ιωάννης Σαραντόπουλος
ioansarant@gmail.com

7 Φεβρουαρίου 2020

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

Κατά τη διάρκεια μιας καταστροφής η ενημέρωση του πληθυσμού για το που μπορεί να προστατευτεί και τι πρέπει να κάνει είναι ζωτικής σημασίας. Σε τέτοιες περιπτώσεις θα ήταν καλό να μπορεί ο κόσμος να πληροφορηθεί είτε μέσα από τις πινακίδες που υπάρχουν στις εθνικές οδούς και δίνουν πληροφορίες κυκλοφορίας είτε από άλλου είδους πινακίδες που θα μπορούσαν να εγκαταστήσουν οι δήμοι σε διάφορα σημεία και να ενημερώνουν τον κόσμο για το τι πρέπει να κάνει σε έκτακτη ανάγκη. Αναγκαίο είναι επίσης τα σωστικά συνεργεία να επικοινωνούν με την κεντρική διοίκηση ώστε να δίνουν και να λαμβάνουν πληροφορίες. Σε περιπτώσεις σαν αυτές που αναφέραμε μπορεί να είναι χρήσιμο και ο εξ αποστάσεως μαζικός έλεγχος κάποιων υποδομών όπως σειρήνες εκτάκτου ανάγκης, αντλίες , γεννήτριες παραγωγής ρεύματος, φωτισμός ,κλπ. Σε όλες τις παραπάνω περιπτώσεις ο πιο συνηθισμένος τρόπος για ανταλλαγή πληροφοριών και μηνυμάτων είναι τα δίκτυα κινητής τηλεφωνίας είτε για τις πινακίδες που αναφέραμε προηγουμένως, είτε για την επικοινωνία με την κεντρική διοίκηση είτε για τον έλεγχο των υποδομών. Τι γίνεται όμως αν δεν υπάρχει η δυνατότητα χρήσης των δικτύων κινητής τηλεφωνίας ;
Έχει παρατηρηθεί σε καταστροφές είτε αυτές είναι φυσικές είτε προέρχονται από ανθρώπινο παράγοντα ότι τα ασύρματα δίκτυα κινητής τηλεφωνίας μπορεί να καταρρεύσουν. Αυτό γίνεται από τον αυξημένο φόρτο κλήσεων που λαμβάνουν εκείνη την χρονική περίοδο, είτε επειδή έχουν καταστραφεί οι υποδομές (πχ κεραίες κινητής τηλεφωνίας που καλύπτουν την περιοχή, δολιοφθορές και άλλα). Σε τέτοιες περιπτώσεις κρίνεται απαραίτητη από πριν , η δημιουργία ενός δευτερεύοντος δικτύου σε περίπτωση που δεν λειτουργούν τα δίκτυα κινητής τηλεφωνίας. Ένα εναλλακτικό δίκτυο επικοινωνίας θα πρέπει να μπορεί να παρέχει μία στοιχειώδη επικοινωνία μεταξύ των σωστικών συνεργείων και της κεντρικής διοίκησης , επίσης θα μπορούσε να μεταδίδει πληροφορίες προς τους πολίτες και να ελέγχει τις υποδομές. Τα δίκτυα LoRaWAN είναι πιθανόν μία καλή εναλλακτική λύση για τη δημιουργία ενός δευτερεύοντος βοηθητικού δικτύου ανταλλαγής πληροφοριών σε τέτοιες καταστάσεις. Το LoRa είναι ένα ασύρματο πρωτόκολλο φυσικού επιπέδου που χαρακτηρίζεται από μεγάλη εμβέλεια, χαμηλή ισχύς και με χαμηλό ρυθμό μετάδοσης δεδομένων (επιτρέπει τη μετάδοση πληροφορίας χαμηλού ρυθμού (250 bps μέχρι 50 kbps) σε μεγάλες αποστάσεις (πολλά km». Το LoRaWAN ένα πρωτόκολλο επιπέδου Mac το οποίο βασίζεται στο πρωτόκολλο φυσικού στρώματος Lora και αυτά τα δύο μαζί δημιουργούν την ασύρματη επικοινωνία των δικτύων LoRaWAN. Για τη λειτουργία ενός δικτύου LoRaWAN πρέπει να εγκατασταθούν διάφορες πύλες (gateways) σε διάφορα κομβικά σημεία των περιοχών που θέλουμε να καλύψου-

με, οι πύλες είναι το αντίστοιχο με τους σταθμούς βάσης στην κινητή τηλεφωνία . Οι τελικοί χρήστες (End Nodes) μπορεί να είναι άνθρωποι ή συσκευές , που επικοινωνούν ασύρματα μέσω του δικτύου LoRaWAN με τις πύλες. Για τη διαχείριση ενός τέτοιου δικτύου ,οι πύλες επικοινωνούν μέσω ευρυζωνικών συνδέσεων με έναν ή περισσότερους εξυπηρετητές δικτύου (Network Server) που ρυθμίζουν την επικοινωνία και την ροή των δεδομένων. Σε συνεργασία με τους εξυπηρετητές δικτύου και ως επέκταση αυτών δημιουργούνται οι εξυπηρετητές εφαρμογών (Application Server) για την διαχείριση , αποθήκευση των πληροφοριών και την αλληλεπίδραση με το χρήστη. Στην εργασία μας, θα αναπτύξουμε ένα δίκτυο LoRaWAN το οποίο θα καλύπτει ένα μέρος της πόλης πειραματικά. Θα δημιουργήσουμε ως τελικούς χρήστες , συσκευές που θα δίνουν την δυνατότητα σε άλλες συσκευές (πχ γιγαντοοθόνες)να συνδέονται με το δίκτυο LoRaWAN και και μέσο Multicast μεταδόσεων να προβάλλουν μηνύματα που θα ενημερώνουν τον πληθυσμό ή θα ελέγχουν υποδομές. Επίσης θα δημιουργήσουμε και ένα πολύ βασικό σύστημα ανταλλαγής μηνυμάτων με Android συσκευές.

1.1 Συνεισφορά και οργάνωση

Στα επόμενα κεφάλαια γίνετε αναφορά στα κυψελωτά δίκτυα και στα Low Power Wide Area δίκτυα. Έπειτα ακολουθεί μια ανάλυση του φυσικού επιπέδου του LoRa και του πρωτοκόλλου LoRaWAN. Μιας και η πρόταση μας στοχεύει στην μετάδοση πληροφοριών μέσω πολυεκπομπών γίνετε αναφορά στους ορούς Multicast , Broadcast και αναφορά στις οδηγίες της LoRaWAN Alliance για τις πολυεκπομπές. Στο τέταρτο κεφάλαιο κάνουμε αναφορά στο Broadcast Encryption και ιδιαίτερα την μέθοδο Subset Difference, επίσης γίνεται αναφορά σε διάφορές τεχνολογίες και εργαλεία που θα χρησιμοποιήσουμε. Στην Αρχιτεκτονική εξηγούμε γιατί διαλέξαμε τη τεχνολογία δικτύου LoRaWAN και γιατί δεν μας καλύπτει η Multicast εκπομπή και υλοποιήσαμε Broadast Encryption πάνω από ένα δίκτυο LoraWAN. Στο έκτο κεφάλαιο εξηγούμε την αρχιτεκτονική του συστήματος μας, πως υλοποιήσαμε τα επιμέρους τμήματα του συστήματος και αναλύουμε τις λειτουργίες στις εφαρμογές και τις συσκευές που δημιουργήσαμε. Στο τελευταίο κεφάλαιο εκφράζονται προβληματισμοί και προτείνονται σκέψεις για περαιτέρω εξέλιξη και βελτίωση του συστήματος.

Κεφάλαιο 2

Ασύρματα δίκτυα

2.1 Κυψελωτά δίκτυα

Οι τεχνολογίες Επικοινωνιών μέσω κινητής τηλεφωνίας (κυψελωτά δίκτυα) [7] είναι η πιο συνηθισμένη μορφή επικοινωνίας ανθρώπων και μηχανών στις μέρες μας. Έχουν μεγάλη χωρητικότητα, μεγάλη ταχύτητα ροής δεδομένων και καλύπτουν μεγάλες αποστάσεις. Τα δίκτυα των εταιρειών κινητής τηλεφωνίας είναι πλέον αρκετά πυκνά τουλάχιστον στις αστικές και ημιαστικές περιοχές, καλύπτοντας ικανοποιητικά τις ανάγκες μας. Το κόστος χρήσης δεν είναι μεγάλο αλλά όχι και αμελητέο. Εκατομμύρια είναι οι άνθρωποι και συσκευές επικοινωνούν μέσα από τέτοιου είδους δίκτυα. Τα δίκτυα κινητής τηλεφωνίας προσφέρουν γρήγορες συνδέσεις σε πραγματικό χρόνο, με μεγάλη διαθεσιμότητα, τη δυνατότητα να συνδεόμαστε από οπουδήποτε στον κόσμο και με σχετικά μικρή κατανάλωση ενέργειας. Μέχρι τώρα υπάρχουν τέσσερις γενιές στις κινητές επικοινωνίες και ήδη μπαίνει στο παιχνίδι και η πέμπτη [8] [9].

- 1η γενιά ήτανε αναλογική καλή για μεταφορά φωνής αλλά όχι για μεταφορά δεδομένων.
- 2η γενιά ήταν ψηφιακή προσέφερε υπηρεσίες ανταλλαγής δεδομένων και φωνής με ταχύτητες μικρότερες των 0.5 Mbps.
- 3η γενιά μας έδινε τη δυνατότητα ανταλλαγής δεδομένων μεγαλύτερη των 63 Mbps . Πράγμα που έκανε δυνατό την μεταφορά ήχου, εικόνας και βίντεο μέσα από τα δίκτυα κινητής τηλεφωνίας.
- 4η γενιά LTE (Λονγ Τερμ Εολυτιον) είναι γρηγορότερη και σαφώς καλύτερη, ξεπερνάει τα 300 Mbps μέχρι 1 Gbps στη γενια LTE-A μεταφέρει άνετα πολυμεσικό υλικό σε πραγματικό χρόνο.
- 5η γενιά έρχεται και θα μας δώσει ταχύτητες μεγαλύτερες του 1 Gbps,1ms χρόνο απόκρισης καλύτερη κάλυψη και διαθεσιμότητα.

2.2 Low Power Wide Area (LPWA)

Low Power Wide Area (LPWA) [10] Με αυτό τον όρο περιγράφουμε μία κατηγορία δικτύων τα οποία έχουν τα εξής χαρακτηριστικά :

- Καλύπτουν μεγάλες αποστάσεις μεγαλύτερες των 10 χιλιομέτρων από το σταθμό βάσης μέχρι τις συσκευές.
- Συνήθως έχουν χαμηλό Data Rate μερικές εκατοντάδες ή χιλιάδες bit/sec.
- Πολύ χαμηλή κατανάλωση σε ενέργεια , οι συσκευές που χρησιμοποιούν τέτοιου είδους δίκτυα συνήθως μπορούν να λειτουργήσουν με μπαταρία για κάποια χρόνια.

Λόγω της ραγδαίας αύξησης των internet of things τσ δίκτυα LPWA αναμένεται να έχουν τεράστια ανάπτυξη τα επόμενα χρόνια. Με LPWA δίκτυα μπορούμε να διαχειριστούμε χιλιάδες συσκευές οι οποίες βρίσκονται σε μεγάλη απόσταση συνήθως από το σταθμό βάσης ή μεταξύ τους. Συνήθως η ισχύς εκπομπής είναι γύρω στα 20dB και το εύρος συχοτήτων είναι γύρω στο 1GHz Sub-1GHz band. Οι τεχνικές διαμόρφωσης που χρησιμοποιούν είναι τέτοιες ώστε να μπορούν να δίνουν μεγάλο Link Budget γύρω στα 150 dB , για αυτό και οι συσκευές λήψης έχουνε μεγάλη ευαισθησία η οποία φτάνει και τα -130 dBm.

2.2.1 Χαρακτηριστικά LPWA

Τεχνικές διαμόρφωσης

Ένα σημαντικό χαρακτηριστικό των LPWA είναι οι τεχνική διαμόρφωσης που χρησιμοποιούν, υπάρχουν κυρίως δυο τεχνικές.

Narrowband modulation Κωδικοποιούν το σήμα με μικρό εύρος BandWidth συνήθως γύρω στα 25 KHz, έτσι μέσα από πολλαπλές συνδέσεις διαχειρίζονται με έξυπνο τρόπο το φάσμα συχνοτήτων. Ο θόρυβος σε μια τέτοια σύνδεση είναι μικρός. Η αποκωδικοποίηση είναι εύκολη και επιτυγχάνεται με φθηνές συσκευές λήψης. Στην ίδια κατηγορία ανήκει η Ultra narrow band διαμόρφωση ο οποία έχει BandWidth μόλις στα 100Hz. Αυτό εξαλείφει το θόρυβο αρά αυξάνει την αξιοπιστία της μετάδοσης , μειώνει το χρόνο λειτουργίας των συσκευών και αυξάνει πάρα πολύ τον αριθμό συσκευών μπορεί να υποστηρίξει το δίκτυο.

Spread spectrum techniques Αυτή η μορφοποίηση χρησιμοποιεί ένα πλατύ εύρος συχνοτήτων. Η διάδοση ενός σήματος στενής ζώνης σε μία διαμόρφωση ευρεία ζώνη οδηγεί σε λιγότερο αποτελεσματική χρήση του φάσματος. Αυτό το πρόβλημα ξεπερνιέται με τη χρήση πολλαπλών ορθογώνιων αλληλουχιών. Όσο οι τελικές συσκευές χρησιμοποιούν διαφορετικά κανάλια και / ή ορθογώνιες ακολουθίες, όλα μπορούν να κωδικοποιηθούν ταυτόχρονα, με αποτέλεσμα την αύξηση της συνολικής χωρητικότητας δικτύου.

Διαχείριση ενέργειας

Χαμηλή κατανάλωση ενέργειας είναι το κλειδί για τα LPWA δίκτυα. Ένας από τους τρόπους που μπορούμε να το πετύχουμε είναι η τοπολογία του δικτύου. Μία τοπολογία όπως είναι Mesh αναγκάζει τις συσκευές να είναι σε κατάσταση ακρόασις πάρα πολύ συχνά και είναι μία ενεργοβόρα τοπολογία, αντιθέτως τοπολογίες όπως του αστέρα που κάθε συσκευή ενεργοποιείται μόνο όταν θέλει να στείλει πληροφορίες είναι πολύ πιο αποδοτική ενεργειακά.

Duty Cycling Είναι επίσης μία καλή τεχνική η οποία απενεργοποιεί συσκευές όσο χρόνο δεν χρειάζεται να στείλουν ή να λάβουν κάποια πληροφορία και τις ενεργοποιεί μόνο την περίοδο που μπορούν να λάβουν ή να στείλουν πληροφορίες.

Lightweight Medium Access Control

Ένα πρωτόκολλο ελέγχου προσπέλασης μέσου το οποίο είναι ελαφρύ και κάνει μονό τις απαραίτητες διαδικασίες, μπορεί να βελτιώσει κατά πολύ την ενεργειακή απόδοση του δικτύου. Όταν οι συσκευές έχουν να διαχειριστούν απλές διαδικασίες δουλεύουν λιγότερο χρόνο και καταναλώνουν λιγότερη ενέργεια.

Μείωση της πολυπλοκότητας από τις τελικές συσκευές

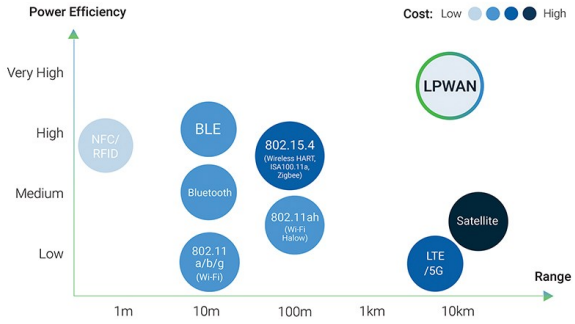
Όταν οι τελικές δεν έχουνε πολλές λειτουργίες να κάνουν σε σχέση με το κομμάτι του δικτύου αυτό τις κάνει λιγότερο ενεργοβόρες. Άρα όταν μεταφέρουμε την πολυπλοκότητα από την αποστολή και λήψη των πληροφοριών σε ένα δίκτυο στους σταθμούς βάσης και αφήσουμε πολύ απλές ενέργειες στις τελικές συσκευές αυτό τις κάνει να λειτουργούν λιγότερο και να καταναλώνουν λιγότερη ενέργεια.

Χαμηλό κόστος

Το χαμηλό κόστος των συσκευών και της επικοινωνίας γενικότερα είναι ένα από τα σημαντικά χαρακτηριστικά των LPWA δικτύων. Γενικά οι συσκευές των LPWA προσπαθούν να έχουν μικρή πολυπλοκότητα στο υλικό έτσι ώστε να είναι φτηνές. Επίσης προσπαθούν οι υποδομές που θα χρησιμοποιούν να έχουν μικρό κόστος. Άλλος ένας τρόπος για να μειωθεί το κόστος σε αυτά τα δίκτυα είναι να χρησιμοποιούν ελεύθερες συχνότητες έτσι ώστε να μη υπάρχει κόστος για τη χρήση αδειοδοτημένων συχνοτήτων.

Επεκτασιμότητα

Τα δίχτυα LPWA είναι δίχτυα που συνήθως χρησιμοποιούνται για Iot συσκευές. Αυτό σημαίνει ότι το πλήθος των συσκευών που πρέπει να υποστηρίξουν είναι μεγάλο ή μπορεί να μεγαλώσει πολύ στο μέλλον, άρα ένας σημαντικός παράγοντας είναι η επεκτασιμότητα τους.



Σχήμα 2.1: Τα LPWA σε σχέση με τα άλλα ασύρματα δίκτυα [1]

2.2.2 Χαρακτηριστικά παραδείγματα δικτύων LPWA

Παρακάτω θα αναφέρουμε μερικές χαρακτηριστικές τεχνολογίες δικτύων LPWA

NB-IoT (Narrowband IoT)

NB-IoT (Narrowband IoT) [11] είναι μια τεχνολογία δικτύων για Iot συσκευές η οποία αναπτύχθηκε από τους κατασκευαστές κινητών συσκευών και προτυπωποιήθηκε από την 3GPP σαν LTE Cat-NB1. Η 3gpp επίσης δημιούργησε άλλο ένα πρότυπο με παρόμοιες προδιαγραφές στο LTE το LTE Cat M για να μπορεί να καλύψει τα ιδιαίτερα χαρακτηριστικά των Iot συσκευών (Το πρότυπο αυτό μπορεί να το δείτε και σαν LTE-M) και τα δύο έχουν σχεδιαστεί με γνώμονα να εξυπηρετούν συσκευές:

- Που δεν διακινούν μεγάλο πλήθος δεδομένων.
- Η επικοινωνία πρέπει να έχει μικρό κόστος.
- Οι συσκευές πρέπει να καταναλώνουν πολύ λίγη ενέργεια για να επικοινωνήσουν.
- Πρέπει να υπάρχει κάλυψη σε μεγάλες αποστάσεις.

Το LTE-M φτιάχτηκε για να υποστηρίξει συσκευές που μέχρι τώρα δεν υποστηριζόταν από το κλασικό LTE. Δηλαδή συσκευές αισθητήρων που πρέπει να καταναλώνουν πάρα πολύ λίγη ενέργεια και να ανταλλάσσουν πληροφορίες μερικών δεκάδων Bytes Ο σχεδιασμός του NB-IoT έχει τρία μέρη:

- Να διαχειρίζεται μη αδειοδοτημένες μπάντες συχνοτήτων.
- Να χρησιμοποιεί την μπάντα συχνοτήτων στα 200 MHz που δεν χρησιμοποιείται πλέον από το GSM.
- Να μπορεί η διαχείριση να γίνει από τους υφιστάμενους σταθμούς βάσης του LTE.

Ο σχεδιασμός του LTE-M:

- Η συχνότητες που διαχειρίζεται είναι στο 1,4 GHz.

- Προσφέρει υψηλό Data Rate (>1Mbps).
- Υποστηρίζει κινητικότητα στη συσκευή.

Τα πλεονεκτήματα του NarrowBand IoT [12] είναι:

- Χρησιμοποιεί τεχνολογίες με απλές κυματομορφές οι οποίες καταναλώνουν πολύ λιγότερη ενέργεια.
- Είναι πολύ οικονομικότερο το υλικό γιατί το κύκλωμα είναι πιο απλό στα 200 MHz από το 1,4 GHz που δουλεύει το LTE-M1.
- Ένα από τα πλεονεκτήματα του NB-IoT είναι ότι έχει τη δυνατότητα καλύτερης διείσδυσης στα κτίρια, λόγω της χαμηλής συχνότητας λειτουργίας, έτσι μπορεί να καλύψει μεγαλύτερες αποστάσεις μέσα σε αστικές περιοχές.

Τα πλεονεκτήματα του LTE-M [12]

- Έχει μεγαλύτερο DataRate από το NB-IoT στα 250Kbps, το οποίο μερικές φορές είναι ένα πλεονέκτημα στις νέες αρχιτεκτονικές.
- Επειδή το LTE-M χρησιμοποιεί τις συχνότητες του LTE αυτό το κάνει περισσότερο απλό σε ότι αφορά τη σχεδίαση των κεραιών και τις ρύθμισης αυτών στους σταθμούς.

Γενικά το NB-IoT είναι η απάντηση των κυψελωτών δικτύων σε τεχνολογίες LoraWankαι Sigfox .

2.2.3 Sigfox

Το Sigfox [13] ανήκει στη εταιρεία SIGFOX Network Operators (SNOs). Το Sigfox είναι ένα Ultra NarrowBand δίκτυο το οποίο δουλεύει στα 200 KHz ή (868 μέχρι 869 MHz ή 902 μέχρι 928 MHz ανάλογα την περιοχή). Οι συχνότητες αυτές είναι δημόσια διαθέσιμες συχνότητες. Κάθε μήνυμα έχει BandWidth στα 100 Hz και μπορεί να μεταφέρει 100 ή 600 Bit/ second, το DataRate έχει να κάνει με την περιοχή. Οι εκπομπές είναι ασύγχρονες μεταξύ των συσκευών και του δικτύου κάθε συσκευή στέλνει κάθε μήνυμα τρεις φορές σε τρεις διαφορετικές συχνότητες, οι σταθμοί βάσης ακροαζονταί όλα τα σήματα. Το πρωτόκολλο είναι ελαφρύ το οποίο σημαίνει λιγότερα επιπλέον δεδομένα κατά τη λήψη και την αποστολή των μηνυμάτων, λιγότερη επεξεργασιά από τις τελικές συσκευές , λιγότερη κατανάλωση ενεργείας. Στην άνω ζεύξη τα μηνύματα μπορούν να έχουν ωφέλιμο μέγεθος μέχρι 12 Bytes και η μεταφορά τους παίρνει γύρω στα 2 δευτερόλεπτα, για κάθε 12 Bytes ωφέλιμο μήνυμα το Sigfox στέλνει συνολικά 26 Bytes. Το μέγιστο ωφέλιμο μήνυμα στη κάτω ζεύξη είναι 8 Bytes. Κάθε συσκευή δεν συνδέεται με ένα συγκεκριμένο βαθμό σταθμό βάσης, όπως συμβαίνει στα κυψελωτά δίκτυα κάθε αποστολή μηνύματος λαμβάνεται από πολλούς σταθμούς βάσης. Το πολύ χαμηλό Bit Rate και για απλή διαμόρφωση σήματος μπορούν να δώσουν ένα Link Budget στα 163,3 dB, αυτό σημαίνει μεγάλη εμβέλεια σήματος . Οι σταθμοί βάσης ανήκουν στην εταιρεία όπως επίσης και το δίκτυο διακίνησης των δεδομένων όταν κάποιος θέλει να βάλει μία συσκευή στο δίκτυο της Sigfox πρέπει να πληρώνει μία ετήσια συνδρομή για αυτή τη συσκευή.

2.2.4 INGENU RPMA

Είναι μία τεχνολογία της εταιρείας INGENU [14], Το RPMA, Random Phase Multiple Access [15]είναι μία τεχνολογία η οποία είναι ειδικά σχεδιασμένη για ασύρματη σύνδεση μηχανή προς μηχανή. Χρησιμοποιεί το ελεύθερο φάσμα συχνοτήτων στα 2,4 GHz. Εχει BandWidth στα 80 MHz με 1 MHz εύρος καναλιού ,η ισχύς εκπομπής στα 21 dB. Η ευαισθησία λήψης είναι στα -133 dB στη τελική συσκευή. Το μέγιστο Link Budget είναι στα 168 dBm. Η κάλυψη του δικτύου φτάνει τα 53 τετραγωνικά χιλιόμετρα. Το μέγεθος των πακέτων είναι από 6 μέχρι 10.000 Bytes.

2.2.5 DASH7 Alliance Protocol (D7A)

DASH7 [16] [17] είναι ένα Ανοιχτού κώδικα ασύρματο πρωτόκολλο δικτύου που έχει προκύψει από τη μη κερδοσκοπική κοινοπραξία DASH7 Alliance. Σε επίπεδο Mac είναι ένα πρωτόκολλο Aloha με σχισμές. Η συχνότητα του δικτύου είναι Sub-GHz , 433 MHz, 868 MHz και 915 MHz. Το Bit Rate είναι στα 9,6 Kbps ή 55.555 Kbps και 166.667 Kbps εξαρτάται από την ταχύτητα της ζώνης συχνοτήτων. Ισχύς εκπομπής είναι 27 dBm και παρέχει μέγιστο Link Budget στα 140 dB. Θεωρείται πάρα πολύ καλό πρωτόκολλο για αποστάσεις 100 έως 300 μέτρα. Σε ευθεία χωρίς εμπόδια μπορεί να φτάσει το 1 χιλιόμετρο και σε τεστ έχει φτάσει μέχρι τα 10 χιλιόμετρα.

2.2.6 Long Range (LoRa)

Το LoRa είναι ένα ασύρματο πρωτόκολλο φυσικού επιπέδου που χαρακτηρίζεται από μεγάλη εμβέλεια, χαμηλή ισχύς και με χαμηλό ρυθμό μετάδοσης δεδομένων (επιτρέπει τη μετάδοση πληροφορίας χαμηλού ρυθμού (250 bps μέχρι 50 kbps) σε μεγάλες αποστάσεις (πολλά km». Το LoRaWAN ένα πρωτόκολλο επιπέδου Mac το οποίο βασίζεται στο πρωτόκολλο φυσικού στρώματος Lora και αυτά τα δύο μαζί δημιουργούν την ασύρματη επικοινωνία των δικτύων LoRaWAN. LoRa [18] είναι μια τεχνολογία η οποία ανήκει στην εταίρια Semtech. Η συχνότητα του δικτύου είναι στα 433 MHz, 868 MHz και 915 MHz. Το Bit Rate είναι στα 0.3-37.5Kbps εξαρτάται από το Spread Factor και τη περιοχή. Ισχύς εκπομπής είναι 14-27dBm και παρέχει μέγιστο Link Budget στα 137dB. Η κάλυψη του δικτύου είναι • 2-5 km (urban),5-15 km (rural),> 15 km (LOS)

	Sigfox	LoRaWAN	Ingenu	Dash7
Modulation	UNB DBPSK(UL), GFSK(DL)	CSS	RPMA-DSSS(UL), CDMA(DL)	GFSK
Band	SUB-GHZ ISM:EU(868MHz), US(902MHz)	SUB-GHZ ISM:EU(433MHz 868MHz), US(915MHz), Asia(430MHz)	ISM 2.4GHz	SUB-GHZ 433MHz, 868MHz, 915MHz
Data Rate	100bps(UL), 600bps(DL)	0.3-37.5kbps(LoRa), 50kbps(FSK)	78kbps(UL), 19.5kbps(DL)	9.6,55.6,166.7 kbps
Range	10km(URBAN), 50km(RURAL)	5km(URBAN), 15km(RURAL)	15km(URBAN)	0-5km(URBAN)
Num.of channels/ orthogonal signals	360 channels	10inEU, 64+8(UL)and 8(DL)in US plus multiple SFs	40 1MHz-channels, up to 1200 signals per channel	3 different channel types (number depends on type & region)
Forward error correction	NO	YES	YES	YES
MAC	unslotted ALOHA	unslotted ALOHA	CDMA-like	CSMA/CA
Topology	STAR	STAR OF STARS	TREE,STAR	TREE,STAR
Payload length	12B(UL), 8B(DL)	up to 250B (depends on SF and region)	10KB	256B
Authentic. encryption	encryption not supported	AES 128b	16b hash, AES 256b	AES 128b

Table 2.1: Compare LWPA [5] [6]

Κεφάλαιο 3

LoRa - LoRaWAN

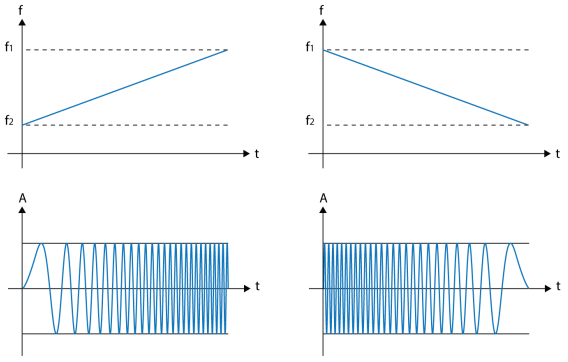
3.1 LoRa Long Radio

3.1.1 Chirp Spread Spectrum

Το Chirp Spread Spectrum [19] είναι μία διαμόρφωση η οποία είχε αναπτυχθεί για εφαρμογές σε ραντάρ το 1940. Είναι μια διαμόρφωση που έχει χρησιμοποιηθεί για στρατιωτικούς σκοπούς και τα τελευταία χρόνια χρησιμοποιείται σε Iot επικοινωνίες λόγο των σχετικά χαμηλών απαιτήσεών ισχύος μετάδοσης και της καλής συμπεριφοράς σε φαινόμενα όπως multipath, Fading, Doppler και παρεμβολείς θορύβου.

Chirp

Το chirp [20] είναι ένα σήμα του οποίου η συχνότητα αυξάνεται ή μειώνεται με την πάροδο του χρόνου. Στη διαμόρφωση του LoRa το Chirp σήμα είναι συνεχές σε διάφορες συχνότητες. Το εύρος συχνοτήτων που χρησιμοποιεί το Chirp είναι ανάλογο του BandWidth του καναλιού. Όταν η συχνότητα αυξάνεται λέγεται up-chirp και όταν συχνότητα μειώνεται λέγεται down-chirp.



Σχήμα 3.1: Τα up-chirp down-chirp

3.1.2 Chip

Το τελικό σήμα παράγεται από τον πολλαπλασιασμό του επιθυμητού σήματος δεδομένων με έναν κώδικα διασποράς, γνωστός ως ακολουθία Chip. Η ακολουθία Chip έχει πολύ ταχύτερο ρυθμό από το σήμα δεδομένων και έτσι μεγαλώνει το εύρος ζώνης σήματος πέρα από το αρχικό εύρος ζώνης του αρχικού σήματος. Ο όρος Chip χρησιμοποιείται για να διακρίνει τα μικρότερα κωδικοποιημένα δυαδικά ψηφία από τα μεγαλύτερα μη κωδικοποιημένα δυαδικά ψηφία του σήματος πληροφοριών.

3.1.3 Spreading Factors (SF)

Το Spreading Factors [21] δείχνει πόσο απλωμένο είναι ένα Chirp στο χρόνο. Η σχέση 2^{SF} μας δίνει των αριθμό των Chip που χρησιμοποιούνται για *SFbits* πληροφορίας και παίρνει τιμές από SF7 έως SF12. Κάθε φορά που αυξάνεται το SF μεγαλώνει και ο χρόνος μεταφοράς ενός Chirp, ενώ όλες οι άλλες παράμετροι μένουν ίδιες. Με το μεγάλωμα της διάρκειας ενός Chirp ο δέκτης έχει λαμβάνει μεγαλύτερη ισχύς του σήματος. Αυτό έχει ως αποτέλεσμα υψηλότερο λόγο σήματος προς θόρυβο (SNR) άρα και μεγαλύτερη πιθανότητα να ληφθεί σωστά κάθε Chirp.

3.1.4 Bandwidth(BW)

Το BandWidth επί της ουσίας είναι αυτό που ορίζει το εύρος συχνοτήτων μετάδοσης ενός σήματος και την διάρκεια ενός Chirp. Με δεδομένο ότι η διάρκεια ενός Chip είναι και το πλήθος των Chip ανα Chirp, καταλαβαίνουμε ότι αλλάζοντας το BandWidth ουσιαστικά αλλάζουμε και τη διάρκεια του Chip άρα και το SNR. Το εύρος του BandWidth που χρησιμοποιεί το LoRa είναι στα 125 KHz και στα 250 KHz.

3.1.5 Transmission Power (TX Pow)

Η ισχύς μετάδοσης έχει να κάνει με την ισχύς που χρειάζεται για να μεταφερθεί ένα Chirp. Όσο πιο αυξημένη είναι η TX Pow τόσο πιο έντονες είναι οι μεταβολές στη διαμόρφωση του σήματος και τόσο πιο δυνατό και καθαρό το σήμα που λαμβάνει ο δέκτης.

3.1.6 Data Rate

Το Data Rate είναι ο αριθμός συμβόλων που μπορούν να μεταδοθούν σε ένα δευτερόλεπτο . Η περίοδος κάθε συμβόλου είναι :

$$T_S = \frac{2^{SF}}{BW} secs \tag{3.1}$$

$$R_S = \frac{1}{T_S} = \frac{BW}{2^{SF}} symbols/sec \tag{3.2}$$

3.1.7 Code Rate (CR)

Το LoRa υποστηρίζει Forward Error Correction (FEC) σε κάθε εκπομπή. Το Coding Rate είναι αυτό που προσδιορίζει πόσα θα είναι τα επιπλέον bit στο χρήσιμο μήνυμα ούτως ώστε να μπορεί να γίνει η διόρθωση σφαλμάτων. Όσο περισσότερα είναι τα επιπλέον bit σε σχέση με το χρήσιμο μήνυμα τόσο καλύτερη είναι η διόρθωση του σφάλματος αλλά και περισσότερη η πληροφορία η οποία μεταδίδεται. Το CR παίρνει τιμές από 1 μέχρι 4.

$$RateCode = \frac{4}{4 + CR} \tag{3.3}$$

Coding Rate	CRC Rate
1	4/5
2	4/6
3	4/7
4	4/8

Table 3.1: Coding Rate

3.1.8 Bit Rate

Το Bit Rate της διαμόρφωσης ορίζεται ως :

$$R_b = SF * \frac{1}{\left[\frac{2^{SF}}{BW}\right]} bit/sec \tag{3.4}$$

όπου
SF=Spreading Factor (7..12)
BW=Modulation BandWidth (Hz)

$$R_C = R_S * 2^{SF} Chips/sec \tag{3.5}$$

$$R_b = SF * \frac{\left[\frac{4}{4+CR}\right]}{\left[\frac{2^{SF}}{BW}\right]} \tag{3.6}$$

όπου:
SF = Spreading Factor (7..12)
CR = Code Rate (1..4)
BW = Modulation BandWidth (Hz)

Spreading Factory (SF)	Bandwidth(Khz)	Bit Rate (bps)	Sensitivity (dbm)
7	125	5470	-123
8	125	3125	-126
9	125	1760	-129
10	125	980	-132
11	125	440	-134.5
12	125	250	-137
7	250	11000	-122

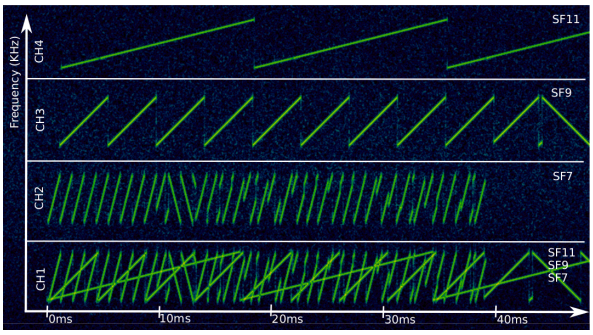
Πίνακας 3.2: Πως το SF και το BW επηρεάζουν το Bit Rate

Παρατηρούμε λοιπόν στο πίνακα ότι όσο αυξάνει το SF μειώνεται το Bit Rate αλλά αυξάνει η ευαισθησία του δέκτη.

3.1.9 Κύρια χαρακτηριστικά της διαμόρφωσης LoRa

Τα κύρια χαρακτηριστικά του LoRa όσο αφορά την κλιμάκωση εύρους Ζώνης είναι :

- Η διαμόρφωση LoRa είναι κλιμακωτή τόσο για το εύρος ζώνης όσο και για τη συχνότητα. Μπορεί να χρησιμοποιηθεί και για στενή ζώνη συχνοτήτων όσο και ευρεία ζώνη συχνοτήτων.
- Το LoRa όπως και το FSK περιλαμβάνει διαμόρφωση σταθερής περιβάλλουσας πράγμα που σημαίνει χαμηλό κατανάλωση ενέργειας.
- Η διαμόρφωση LoRa λόγω του BandWidth Time Product (>1) και της ασύγχρονης φύσης της μπορεί να έχει μεγαλύτερη ανοχή στο θόρυβο εντός και εκτός της ζώνης συχνοτήτων.
- Λόγω του ευρύ φάσματος συχνοτήτων που χρησιμοποιεί κάθε Chirp ο παλμός στο LoRa, δίνει μεγάλη ανοχή σε φαινόμενα Multipath και Fading.
- Το LoRa έχει τη δυνατότητα κάλυψης μεγάλων αποστάσεων,αυτό συμβαίνει γιατί η μεγάλη ανοχή που έχει στο θόρυβο και στο Fading δίνει μεγάλο Link Budget.
- Η μετατόπιση Doppler προκαλεί μικρή μετατόπιση συχνότητας στον παλμό LoRa που είναι σχεδόν αμελητέα.
- Η διαμόρφωση LoRa είναι μία ορθογωνική Spreading Factor διαμόρφωση η οποία επιτρέπει πολλαπλά Spread Signals να μεταδίδονται την ίδια στιγμή ,στην ίδια συχνότητα, αυτό αυξάνει κατά πολύ τη χωρητικότητα του δικτύου .



Σχήμα 3.2: Τα Chirp σε σχέση με τα SF

3.1.10 Μορφή των πακέτων

Preamble

Το Preamble [22] [23] LoRa χρησιμοποιείται για να συγχρονίσει το λήπτη για τα δεδομένα που έρχονται. Εξ ορισμού το Preamble είναι 12 σύμβολα Που εκπέμπονται πρώτα σε κάθε πακέτο LoRa αλλά μπορεί και να επεκταθεί. Οι τιμές που παίρνει είναι από 6 μέχρι 65535. Ο δέκτης για να λάβει ένα πακέτο ενεργοποιεί μια διαδικασία ανίχνευσης Preamble ανά τακτά χρονικά διαστήματα. Για το λόγο αυτό το μήκος του Preamble θα πρέπει να είναι κοινό και στο πομπό και το δέκτη. Όπου το μήκος του Preamble δεν είναι γνωστό το μέγιστο μήκος του Preamble πρέπει να προγραμματιστεί στην πλευρά του δέκτη.

Σχήμα 3.3: Στιγμιότυπο εκπομπής LoRa

Header (Physical Header PHDR)

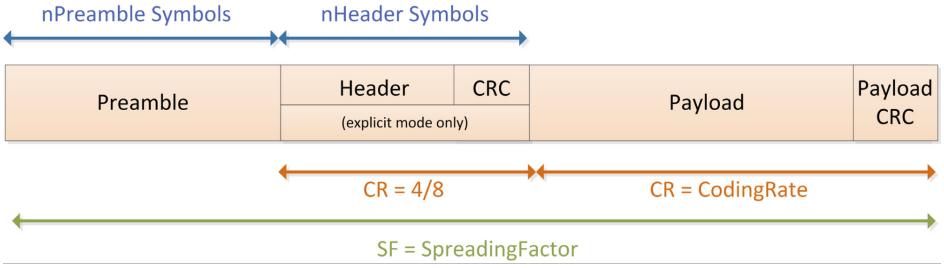
Κεφαλίδα είναι το δεύτερο κομμάτι του μηνύματος η οποία διαφοροποιείται ανάλογα με το είδος του μηνύματος. Υπάρχουν δύο είδη μηνυμάτων Explicit και Implicit.

Explicit Αυτό το είδος μηνύματος είναι που χρησιμοποιείται εξ ορισμού. Σε αυτό το είδος του μηνύματος η κεφαλίδα περιέχει τα εξής:

- Το μέγεθος του χρήσιμου μηνύματος το Payload.
- Το Code Rate που χρησιμοποιείται για την διόρθωση σφαλμάτων (PHDR CRC).
- Ένα 16bit CRC (Cyclic Redundancy Check) που αφορά τον έλεγχο και την διόρθωση σφαλμάτων στο Payload.

Implicit Το άλλο είδος μηνύματος είναι όταν το μέγεθος του μηνύματος Payload,το CR και το CRC (Cyclic Redundancy Check) είναι γνωστά και προκαθορισμένα εξαρχής, από τον κατασκευαστή και στο πομπό και στο δέκτη. Έτσι δεν χρειάζεται να μεταφερθούν μέσα στο μήνυμα, σε αυτή την περίπτωση το τμήμα του Header και του CRC αφαιρούνται από το μήνυμα. Με αυτό το τρόπο πετυχαίνουμε μικρότερα μηνύματα. Σημείωση μηνύματα σε Implicit κατάσταση μπορούν να μεταδοθούν μόνο σε SF6.

Payload Το τμήμα Payload του μηνύματος είναι το τμήμα που περιέχει την ωφέλιμη πληροφορία. Είναι μεταβλητό ως προς το μήκος του όταν πρόκειται για Explicit μήνυμα ή σταθερό όταν πρόκειται για Implicit μήνυμα.



Σχήμα 3.4: Μορφή πακέτου LoRa

3.1.11 Χρόνος μετάδοσης μηνύματος(Time on air)

Ο χρόνος μετάδοσης μηνύματος εξαρτά τε από το SF, CR και BW, ο συνολικός χρόνος μεταφοράς ενός LoRa πακέτου μπορεί να υπολογιστεί ως ακολούθως :

Ο χρόνος μετάδοσης ενός συμβόλου.

$$T_S = \frac{1}{R_S} \tag{3.7}$$

Ο χρόνος αποστολής του Preamble έχει να κάνει με το πλήθος των συμβολών του.

$$T_{preamble} = (n_{preamble} + 4.25) * T_{sym} \tag{3.8}$$

Ο χρόνος αποστολής του Payload έχει να κάνει με το μέγεθος του μηνύματος και το είδος του μηνύματος

$$n_{payload} = 8 + max\left(ceil\left[\frac{(8 * PL - 4 * SF + 28 + 16 * CRC - 20 * IH)}{4\,(SF - 2 * DE)}\right](CR + 4)\,, 0\right) \tag{3.9}$$

οπου
PL αριθμός Bytes του Payload.
SF Spread Factor.
IH 1 για Explicit , 0 για Implicit.
DE 1 για ρύθμιση χαμηλού Data Rate, 0 αν αυτή η επιλογή είναι απενεργοποιημένη.
CRC 1 αν υπάρχει CRC ,0 αν δεν υπάρχει CRC. CR παίρνει τιμές από 1 μέχρι 4.

Με βάση τα παραπάνω ο συνολικός χρόνος του Payload είναι:

$$T_{payload} = n_{payload} * T_S \tag{3.10}$$

Τέλος ο συνολικός χρόνος του πακέτου είναι

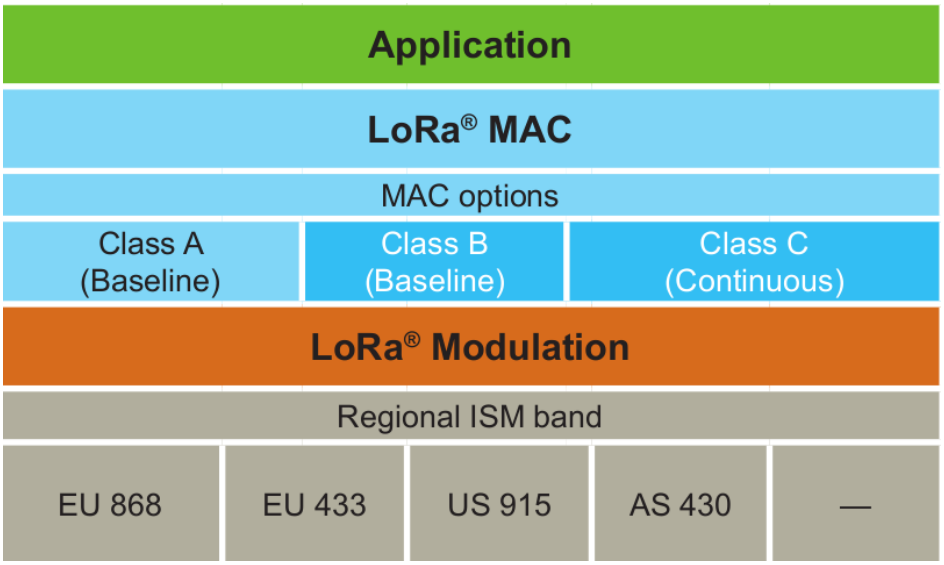
$$T_{packet} = T_{preamble} + T_{payload} \tag{3.11}$$

3.2 Το πρότυπο LoRaWAN

Το LoRa [18] είναι το πρωτόκολλο επικοινωνίας στο φυσικό επίπεδο. Όπως ήδη έχουμε δει τα κύρια χαρακτηριστικά του LoRa είναι η μικρή κατανάλωση ενέργειας των τελικών συσκευών End Node (EN), η μεγάλη απόσταση στην οποία μπορούν να ταξιδέψουν πληροφορίες και ο χαμηλός ρυθμός μετάδοσης πληροφοριών. Το LoRaWAN είναι το τηλεπικοινωνιακό πρωτόκολλο και η αρχιτεκτονική που κάνει χρήση της τεχνολογία LoRa στο φυσικό επίπεδο και επεκτείνει τις δυνατότητες σε επίπεδο δικτύου. Το πρωτόκολλο και αρχιτεκτονική του LoRaWAN έχει εστιάσει.

- Στη μικρή κατανάλωση των συσκευών.
- Τη χωρητικότητα του δικτύου.
- Την ποιότητα του δικτύου.
- Την ασφάλεια.
- Τις υπηρεσίες σε επίπεδο εφαρμογής .

Οι προδιαγραφές του LoRaWAN είναι ανοιχτές δημιουργούνται και καταγράφονται από μια κοινοπραξία εταιρειών και οργανισμών που ονομάζεται LoRaWAN Alliance.
Σε αυτή την εργασία αναφερόμαστε κυρίως στην έκδοση του πρωτοκόλλου LoRaWAN 1.0.3. Υπάρχει όμως και πιο πρόσφατη έκδοση η LoRaWAN v1.1.



Σχήμα 3.5: Τα επίπεδα διαστρωμάτωσης του LoRaWAN

3.2.1 Αρχιτεκτονική του δικτύου

Μέρη του δικτύου

End Node (EN) Είναι οι τελικές συσκευές δηλαδή συσκευές με αισθητήρες ή επενεργητές, που είναι οι παραγωγοί ή τελικοί δέκτες των πληροφοριών. Συνήθως έχουν χαμηλή κατανάλωση, δουλεύουν με μπαταρία και ένα από τα ζητούμενα είναι να μπορούν να δουλέψουν πολύ καιρό χωρίς να χρειαστεί να τους αλλάξουμε μπαταρία.

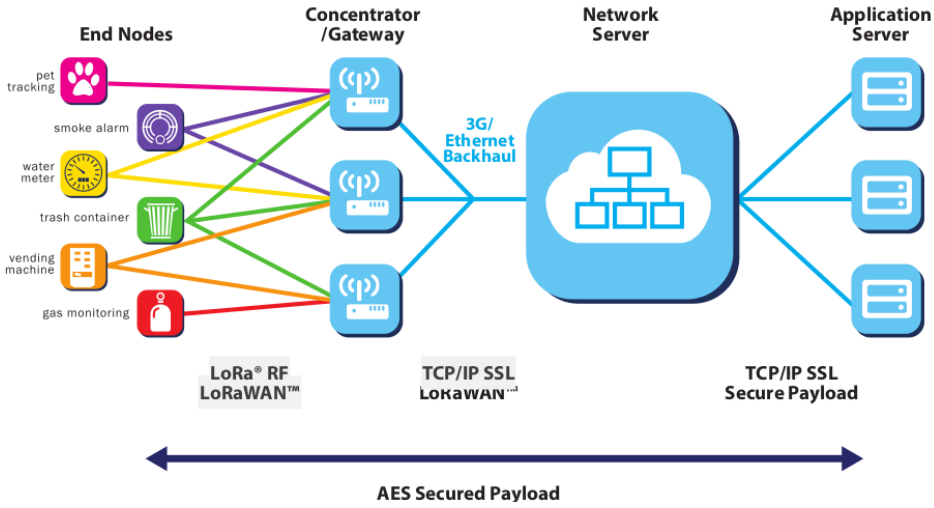
GateWay (GW) Είναι οι συσκευές που δέχονται τα μηνύματα μέσω του δικτύου LoRa και προωθούν μέσω ενός IP δικτύου τα μηνύματα σε έναν κεντρικό εξυπηρετητή. Αντίστροφα πάλι δέχονται τα μηνύματα από έναν εξυπηρετητή και στη συνέχεια αναλαμβάνουν να τα μεταδώσουν στις τελικές συσκευές μέσω του δικτύου LoRa. Τα GateWay μετατρέπουν τα LoRa πακέτα σε πακέτα που μεταδίδονται μέσω IP πρωτοκόλλου. Συνήθως χρησιμοποιείται το UDP για την επικοινωνία με το Network Server αλλά σε κάποιες υλοποιήσεις χρησιμοποιείται το πρωτόκολλο MQTT.

Network Server (NS) Εξυπηρετητής δικτύου είναι η υποδομή εκείνη η οποία αναλαμβάνει να δέχεται όλα τα μηνύματα από όλα τα GateWay, να προωθεί τα μηνύματα από τον Application Server στα GateWay, να στέλνει τις κατάλληλες ρυθμίσεις στα GateWay και στα End Node, να διαχειρίζεται τα Data Rate των End Node να μπορεί να ανταλλάσσει MACCommand με τα End Node και τέλος να μπορεί να επικοινωνεί με τον Application Server ώστε να δώσει και να πάρει τις απαραίτητες πληροφορίες που έρχονται από το επίπεδο της εφαρμογής.

Application Server (AS) Ο Application Server είναι η τελική εφαρμογή που αναλαμβάνει να διαχειριστεί τα End Nodes ανταλλάσσοντας μηνύματα με αυτά. Είναι η εφαρμογή που υλοποιεί μια εργασία κάνοντας χρήση του δικτύου LoRaWAN. Ο Application Server επικοινωνεί μόνο με τον Network Server και μέσο αυτού με τα End Nodes. Στη πραγματικότητα όλη αρχιτεκτονική του δικτύου είναι αδιαφανής για τον Application Server.Η Σύνδεση του Application Server με τον Network Server συνήθως γίνετε μέσω ενός IP δικτύου, ενώ ο τρόπος και τα πρωτοκολλά επικοινωνίας που θα χρησιμοποιήσει εξαρτώνται από το τι υποστηρίζει ο Network Server.

Αρχιτεκτονική LoRaWAN

Η τοπολογία ενός δικτύου LoRaWAN είναι Αστέρας, δηλαδή οι End Nodes επικοινωνούν με ένα κεντρικό σταθμό που το λέμε GateWay. Υπάρχει δυνατότητα πολλαπλών GateWay που επικοινωνούν με πολλαπλά End Nodes. Η επιλογή της αρχιτεκτονικής Αστέρα έχει γίνει με γνώμονα τη μεγαλύτερη διάρκεια της μπαταρίας των End Nodes όπως επίσης και με την μεγάλη κάλυψη που μπορεί να προσφέρει το δίκτυο.



Σχήμα 3.6: Η αρχιτεκτονική ενός δικτύου LoRaWAN

Κάθε End Node όταν στέλνει ένα μήνυμα ,αυτό μπορεί να λαμβάνεται από πολλά GateWay αυτά αναμεταδίδουν το μήνυμα στο Network Server (NS) μέσω της υποδομής κορμού. Η σύνδεση του GateWay με το Network Server μπορεί να είναι με οποιοδήποτε τρόπο πχ με Ethernet, 3G ,WiFi, Satellite. Το ξεκαθάρισμα των μηνυμάτων από τα διπλότυπα όπως επίσης και η διαχείριση του δικτύου γίνεται από το Network Server. Με αυτό τον τρόπο η πολυπλοκότητα και η ευφυΐα του δικτύου έχει προωθηθεί από τη μεριά των GateWay στο Network Servero οποίος λαμβάνει όλα τα μηνύματα και διαχειρίζεται όλα τα GateWay του δικτύου. Μια τέτοια υποδομή όπως ο Network Server μπορεί να έχει υλοποιηθεί και να βασίζεται στο Cloud. Το ότι κάθε End Node δεν εξαρτάται από κάποιο συγκεκριμένο GateWay κάνει εύκολα διαχειρίσιμα τα nd Nodes ακόμα και αν αυτά είναι κινούμενα. Η πιο συνηθισμένη υλοποίηση ενός δικτύου LoRaWAN είναι ασύγχρονη , κάθε End Node που έχει να στείλει ένα σήμα , ενεργοποιείται και μέσω ενός πρωτοκόλλου τύπου Aloha στέλνει το σήμα στο δίκτυο. Δεν είναι απαραίτητο οι End Nodes να συγχρονίζονται πάνω στο δίκτυο, αυτό σημαίνει ότι δεν χρειάζεται να ενεργοποιούνται ανά τακτά χρονικά διαστήματα και να ανταλλάσσουν πληροφορίες για να συγχρονιστούν με το δίκτυο, αυτό αυξάνει κατά πολύ το χρόνο ζωής μπαταρίας των End Nodes. Όπως έχουμε πει το πρωτόκολλο LoRa μπορεί μέσω διαφορετικών συχνοτήτων και διαφορετικών Spread Factor να υποστηρίζεις πολλαπλά κανάλια έτσι μπορούμε να έχουμε GateWay που μπορούν ταυτόχρονα να λάβουν σήματα σε πολλαπλά κανάλια και με αυτό τον τρόπο να έχουμε μεγάλη χωρητικότητα δικτύου.

3.3 Οι κλάσεις του δικτύου LoRaWAN

Κάθε End Node σε ένα δίκτυο LoRaWAN μπορεί να λειτουργεί με τρεις διαφορετικούς τρόπους τους οποίους τους λέμε κλάσεις.Υπάρχουν τρεις κλάσεις, Class A ,Class B και Class C , η καθεμία από αυτές τις κλάσεις προσδιορίζουν ένα διαφορετικό τρόπο λειτουργίας των End Nodes του δικτύου.

3.3.1 Class A (A Asynchronous)

Στην Class A οι End Nodes επικοινωνούν ασύγχρονα και με αμφίδρομο τρόπο με το GateWay. Δηλαδή κάθε End Node μπορεί να πάρει και να στείλει σήμα στο GateWay αλλά όχι ταυτόχρονα λήψη και αποστολή. Κάθε End Node αφού έχει εκπέμψει ένα σήμα ακολουθούν δύο μικρές χρονικές περίοδοι στις οποίες μπορεί να κάνει λήψη σήματος από το GateWay (δύο μικρά χρονικά παράθυρα). Η χρονική στιγμή που που θα εκπέμψουν ένα σήμα οι End Nodes δεν είναι προκαθορισμένος αλλά εξαρτάται από το πότε έχουν κάποιο σήμα που θέλουν να στείλουν. Χρησιμοποιούν ένα πρωτόκολλο τυχαίας χρονικής προσπέλασης στο μέσο τύπου ALOHA. Η Class A προτείνετε για χαμηλής κατανάλωσης End Nodes τα οποία εκπέμπουν όποτε θέλουν και μπορούν να κάνουν λήψη από το GateWay μόνο μετά την δική τους εκπομπή. Ένα End Node στέλνει ένα μήνυμα στο GateWay σε συγκεκριμένο κανάλι και με συγκεκριμένο DataRate αυτό διαρκεί ένα συγκεκριμένο χρονικό διάστημα που το λέμε Transmit Time On Air μετά από αυτό ακολουθεί ένα κενό χρονικό διάστημα το οποίο είναι το RECEIVE_DELAY1 συνήθως διαρκεί 1 δευτερόλεπτο και μετά από αυτό ξεκινάει το χρονικό παράθυρο λήψης RX1 σε αυτό το χρονικό παράθυρο το End Node περιμένει να λάβει ένα σήμα από το GateWay το σήμα θα είναι στο ίδιο κανάλι και με το ίδιο DataRate με το σήμα που έστειλε. Αν δεν γίνει δυνατή η λήψη ενός σήματος τότε μετά χρονικό διάστημα RECEIVE_DELAY2 =(RECEIVE_DELAY1 +1 συνήθως 2 δευτερόλεπτα από το τέλος της αποστολής του σήματος) ανοίγει το χρονικό παράθυρο λήψης RX2. Το παράθυρο RX2 είναι ένα χρονικό παράθυρο λήψης μηνύματος στο οποίο περιμένει λήψη ενός σήματος στο κανάλι με συχνότητα 869.525 MHz και DR0 (SF12, BW 125 KHz). Ο ελάχιστος χρόνος που μένει ανοιχτό ένα παράθυρο, είναι ο χρόνος που χρειάζεται για να εντοπιστεί το Preamble ενός σήματος. Ένα End Node δεν μπορεί να στείλει άλλο μήνυμα αν δεν περάσει το χρονικό διάστημα των 2 παραθύρων λήψης.

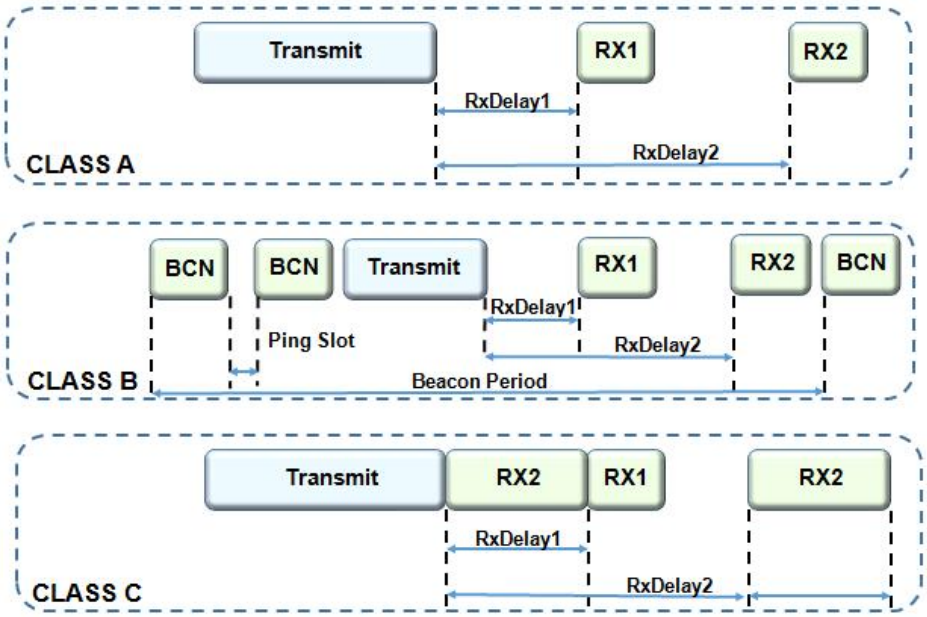
3.3.2 Class B(B Beacon)

Στην Class B οι End Nodes επικοινωνούν με αμφίδρομο τρόπο με το GateWay (όχι ταυτόχρονη λήψη ,αποστολή) και με χρονοπρογραμματισμο χρόνων λήψης. Στην Class B οι End Nodes συγχρονίζονται με βάση ένα σήμα που λαμβάνουν από το GateWay το Beacon και προγραμματίζονται χρόνοι λήψης από ένα αλγόριθμο προγραμματισμού. Με το τρόπο αυτό οι End Node έχουν περισσότερα χρονικά περιθώρια για λήψη σημάτων από ότι στην Class A, καταναλώνουν όμως περισσότερη ενέργεια γιατί πρέπει σε τακτά χρονικά διαστήματα να ενεργοποιούνται και να συγχρονίζονται με το GateWay ή να περιμένουν κάποιο σήμα από αυτόν. Σε ένα δίκτυο που υποστηρίζει Class B όλα τα GateWay ταυτόχρονα στέλνουν ένα Beacon σήμα σε προκαθορισμένο χρόνο στα End Nodes. Το χρονικό διάστημα μεταξύ δυο eacon ονομάζεται BEACON_PERIOD. Όταν ένα nd Node μπει σε κατάσταση Class B το πρώτο πράγμα που κάνει είναι να προσπαθεί να λάβει τα Beacon σήματα ώστε να συγχρονιστεί. Βασισμένο σε αυτές τις χρονικές αναφορές και το χρονικό διάστημα, το nd Node σε προκαθορισμένα περιοδικά χρονικά διαστήματα να ανοίγει χρονικά παράθυρα λήψης που λέγονται Ping Slot ή απλά Ping, τα οποία δίνουν την δυνατότητα στα GateWay να στείλουν σήματα στο nd Node. Ο GateWay και οι End Nodes ξέρουν πως χρονίζοντα τα Ping ανάμεσα σε δύο Beacon και έτσι ο GateWay ξέρει πότε να εκπέμψει ένα σήμα σε χρόνο που μπορεί να το λάβει ο nd Node. Επειδή τα Beacon στέλνονται από όλα GateWay για να ξέρει ο Network Server

σε ποιο GateWay να στείλει το σήμα σε ένα End Node. Πρέπει αν End Node καταλάβει ότι έχει αλλάξει GateWay από το οποίο λαμβάνει το Beacon να στείλει ένα σήμα στον Network Server για να τον αντιστοιχίσει σε άλλο GateWay.

3.3.3 Class C (C Continuously)

Στην Class C οι End Nodes επικοινωνούν με αμφίδρομο τρόπο με το GateWay (όχι ταυτόχρονη λήψη ,αποστολή) και με άμεση ανταπόκριση στη λήψης. Όταν ένα End Node είναι σε Class C είναι συνεχώς κατάσταση λήψης και κλείνει μόνο κατά την διάρκεια αποστολής ενός σήματος. Οι End Nodes χρειάζονται περισσότερη ενέργεια γιατί είναι συνέχεια ενεργοί προκειμένου να αφουγκράζονται το δίκτυο. Οι End Nodes έχουν το μικρότερο χρόνο απόκρισης στην λήψη σημάτων. Ένα End Node αρχικά ξεκινάει από Class A και κάνει Join , στη συνέχεια με κάποιο μήνυμα από τον Application Server ή μετά από δική του απόφαση μπορεί να γυρίσει σε Class C. Ο τρόπος λειτουργίας ενός End Node σε Class C μοιάζει πάρα πολύ με τον τρόπο της λειτουργίας σε Class A. Ο End Node αποστέλλει ένα σήμα σε χρόνο Transmit Time On Air στη συνέχεια περιμένει ένα συγκεκριμένο διάστημα RECEIVE_DELAY1 σε αυτό το διάστημα έχει ανοιχτό ένα παράθυρο RX2 αφού περάσει αυτό χρονικό διάστημα RECEIVE_DELAY1, ανοίγει ένα νέο χρονικό παράθυρο λήψης RX1 , μετά από χρονικό διάστημα RECEIVE_DELAY2=(RECEIVE_DELAY1 +1 2 δευτερόλεπτα από το τέλος της αποστολής του σήματος) κλείνει το παράθυρο RX1 και ανοίγει ένα καινούργιο χρονικό παράθυρο λήψης RX2 το οποίο το κρατάει ανοιχτό μέχρι την επόμενη αποστολή σήματος, όπου και επαναλαμβάνεται η ίδια διαδικασία. Όταν ένας End Node είναι σε Class C είναι σχεδόν πάντα διαθέσιμος να λάβει κάποιο σήμα.



Σχήμα 3.7: Τα χρονικά παράθυρα αναμονής και λήψης για τις τρεις κλάσεις του πρωτοκόλλου LoRaWAN

3.4 Η δομή του πακέτου

3.4.1 Φυσικό επίπεδο

Στο φυσικό επίπεδο υπάρχουν 2 τύποι μηνυμάτων, ο πρώτος τύπος μηνυμάτων είναι αυτά που πηγαίνουν από τα End Nodes στα GateWay (Uplink Messages), δεύτερος τύπος μηνύματών είναι τα μηνύματα που έρχονται από τα GateWay στα End Nodes (Downlink Messages)

Uplink

Τα (Uplink Messages) στο πρωτόκολλο LoRa αποτελούνται από μία επικεφαλίδα Header ή PHDR ακολουθούμενο από ένα κυκλικό κώδικά απόρριψης CRC για το περιεχόμενο της επικεφαλίδας PHDR_CRC. Στη συνέχεια είναι το χρήσιμο κομμάτι του μηνύματος PHYPayload και ένα CRC για ολόκληρο το μήνυμα.

Preamble	PHDR	PHDR_CRC	PHYPayload	CRC

Σχήμα 3.8: Δομή Uplink Messages

Downlink

Τα (Downlink Messages) στο πρωτόκολλο LoRa αποτελούνται από μία επικεφαλίδα PHDR ακολουθούμενο από ένα κυκλικό κώδικά απόρριψης CRC για το περιεχόμενο της επικεφαλίδας PHDR_CRC. Στη συνέχεια είναι το χρήσιμο κομμάτι του μηνύματος

Preamble	PHDR	PHDR_CRC	PHYPayload

Σχήμα 3.9: Δομή Downlink Messages

Η διαφορά των δύο μηνυμάτων είναι ο CRC κώδικας στο τέλος του μηνύματος Uplink ενώ δεν υπάρχει στα Ddownlink.

3.4.2 PHYPayload

Το PHYPayload είναι το Payload από το φυσικό επίπεδο , αυτό το μήνυμα έχει μια συγκεκριμένη μορφή που ορίζεται από το επίπεδο MAC του πρωτοκόλλου LoRaWAN. Ένα MAC μήνυμα αποτελείται από μία επικεφαλίδα Mac (MHDR) 1 Byte από το χρήσιμο μήνυμα MACPayload 7-M Bytes και τελειώνει με κώδικα MIC 4 Bytes για την διασφάλιση της ακεραιότητας της πληροφορίας (Message integrity Code MIC).

PHYPayload		
1 Byte	7..M Bytes	4 Bytes
MHDR	MACPayload	MIC

Σχήμα 3.10: Δομή PHYPayload

Join Request		
1 Byte	7..M Bytes	4 Bytes
MHDR	Join Request / Rejoin Request	MIC

Σχήμα 3.11: Δομή Join

MHDR

Η επικεφαλίδα προσδιορίζει τον τύπο του μηνύματος και την έκδοση του πρωτοκόλλου .

MHDR		
7-5 bit	4-2 bit	1-0 bit
Mtype	RFU	Major

Σχήμα 3.12: Δομή Join

MType Ο τύπος δεδομένων MType των μηνυμάτων χωρίζεται στα μηνύματα τα οποία χρειάζονται επιβεβαίωση Confirmed-Data Message, αυτά που δεν χρειάζονται επιβεβαίωση Unconfirmed-Data Message και στα ιδιωτικά μηνύματα Proprietary Messages τα οποία δεν ακολουθούν τη συγκεκριμένη μορφή των μηνυμάτων αλλά είναι μια δομή ορισμένη από το κατασκευαστή. Τέλος υπάρχουν και τα μηνύματα που χρησιμοποιούνται για το Join την σύνδεση του End Nodeστο δίκτυο.

MType	Description
000	Join-Request
001	Join-Accept
010	Unconfirmed Data Up
011	Unconfirmed Data Down
100	Confirmed Data Up
101	Confirmed Data Down
110	RFU V1.0.3 - Rejoin-Request v1.1
111	Proprietary

Πίνακας 3.3: Τιμές που παίρνει το πεδίο MType

Major Η έκδοση του πρωτοκόλλου που χρησιμοποιείται από το μήνυμα.

MACPayload

Το Χρήσιμο κομμάτι του μηνύματος MACPayload σε ένα μήνυμα μορφής MAC, έχει και αυτό δική του δομή. Αποτελείται από τη κεφαλίδα του πλαισίου FHDR από ένα προαιρετικό πεδίο FPort και από το προαιρετικό πεδίο με δεδομένα FRMPayload.

MACPayload		
7..22 Bytes	0..1 Byte	0..N Bytes
FHDR	FPort	FRMPayload (encrypted)

Σχήμα 3.13: Δομή MACPayload

FDHR

Η επικεφαλίδα FHDR αποτελείται από τη διεύθυνση της συσκευής DevAddr 4 Bytes από το πλαίσιο ελέγχου FCtrl 1 Byte ,το μετρητή πλαισίου FCnt 2 Bytes και από 0 μέχρι 15 Bytes το πλαίσιο επιλογής το οποίο χρησιμοποιείται για να μεταφέρει MAC εντολές .

FHDR			
4 Bytes	1 Byte	2 Bytes	0..15 Bytes
DevAddr	FCtrl	FCnt	FOpts

Σχήμα 3.14: Δομή FHDR

FCtrl Το FCtrl είναι διαφορετικό για τα πλαίσια χρησιμοποιούμε στην αποστολή και διαφορετικό για τα πλαίσια που χρησιμοποιούμε στη λήψη.

FCtrl Downlink				
7 bit	6 bit	5 bit	4 bit	3..0bit
ADR	RFU	ACK	FPending	FOptsLen

Σχήμα 3.15: Δομή FCtrl στο Downlink

FCtrl Uplink				
7 bit	6 bit	5 bit	4 bit	3..0bit
ADR	ADRACKReq	ACK	Class B	FOptsLen

Σχήμα 3.16: Δομή FCtrl στο Uplink

- ADR Adaptive Data Rate
  Όταν ADR είναι ενοποιημένο τότε το DR του σήματος αποστολής στα End Nodes καθορίζονται με εντολές MAC. Το δίκτυο προσπαθεί να περιορίσει το χρόνο που τα σήματα είναι στον αέρα και έτσι προσπαθεί να έχουν οι End Nodesto ταχύτερο DR που είναι δυνατόν. Αν το ADR δεν είναι ενεργοποιημένο τότε το δίκτυο δεν

θα επιχειρήσει να αλλάξει ούτε το DR ούτε το TX Power του End Node και το DR ορίζεται από το End Node.

- ADRACKReq
  Σε κάθε Uplink ο End Node αυξάνει ένα μετρητή ADR_ACK_CNT αν ο μετρητής ξεπεράσει ένα όριο ADR_ACK_LIMIT χωρίς να λάβει Downlink τότε στέλνει ένα Uplink με ενεργοποιημένο το ADRACKReq bit και το δίκτυο πρέπει να απαντήσει με ένα Downlink ACK σε χρόνο ADR_ACK_DELAY τότε μηδενίζει το ADR_ACK_CNT. Αν δεν λάβει απάντηση μειώνει το DR.

- ACK
  Όταν λαμβάνεται ένα μήνυμα που πρέπει να επιβεβαιωθεί, ο λήπτης πρέπει να ανταποκριθεί με ένα μήνυμα που θα έχει ενεργοποιημένο το ACK bit. Αν ο αποστολέας είναι End Node το δίκτυο θα πρέπει να στείλει μία επιβεβαίωση χρησιμοποιώντας τα παράθυρα λήψης του End Node. Αν ο αποστολέας είναι το GateWay τότε ο End Node θα στείλει μία επιβεβαίωση σε χρόνο της επιλογής του. Αν δεν ληφθεί η επιβεβαίωση το μήνυμα θα σταλεί ξανά. Η επιβεβαίωση στέλνετε μόνο για το αρχικό μηνύματα και όχι τις επανεκπέμπομπές .

- FPending (frame pending bit)
  Χρησιμοποιείται μόνος στην Download επικοινωνία. Το χρησιμοποιεί ο GateWay για να ενημερώσει το End Node ότι έχει κι άλλες πληροφορίες να στείλει, έτσι ο End Node πρέπει να στείλει όσο το δυνατόν γρηγορότερα ένα Uplink για να ανοίξει νέο παράθυρο λήψης.

- FOptsLen
  Ενημερώνει ποσά Bytes είναι η πληροφορία που περιέχεται στο πεδίο FOpts που βρίσκεται στο FHDR.

FCnt Κάθε End Node έχει δύο μετρητές τους οποίους παρακολουθεί. Έναν για τα μηνύματα που στέλνει (Uplink) FCntUp τον οποίο αυξάνει ο End Node κάθε φορά που στέλνει ένα μήνυμα και έναν για τα μηνύματα που λαμβάνει (Downlink) FCntDown που αυξάνει ο Network Server. Ο Network Server παρακολουθεί και κρατάει ένα ζεύγος μετρητών για κάθε End Node. Το πεδίο FCnt περιέχει στο Downlink τον μετρητή του Network Server και στο Uplink του End Node. Αν η διαφορά μεταξύ του μετρητή που έχει κάθε μέρος και που λαμβάνει από το μήνυμα είναι μεγαλύτερη από MAX_FCNT_GAP σημαίνει ότι πολλά μηνύματα έχουν χαθεί και η σύνδεση πρεπει να διακοπεί. Μετά από κάθε επιτυχημένο Join οι μετρητές μηδενίζονται.

FOpts Το πεδίο αυτό περιέχει MAC Commands εντολές του πρωτοκόλλου MAC. Οι MAC Commands δεν μπορούν να σταλούν ταυτόχρονα με δεδομένα και στέλνονται πάντα σε FPort 0 .

FOpts		
1..5 Bytes		1..5 Bytes
MACCommand 1	.....	MACCommand n

Σχήμα 3.17: Δομή FOpts

MACCommand Για να γίνει η διαχείριση του δικτύου χρησιμοποιούνται οι MACCommands αυτές ανταλλάσσονται αποκλειστικά ανάμεσα στον Network Server και στα End Nodes, ανήκουν στο επίπεδο διαχείρισης MAC και ποτέ δεν είναι εμφανής στο επίπεδο εφαρμογής και στους Application Sevrers. Αποτελείται από δύο μέρη από το CID 1 Byte που προσδιορίζει τον κωδικό της εντολής αλλά και την ίδια την εντολή και από τα ορίσματα της εντολής 0..4 Bytes.

MACCommand	
1 Byte	0..4 Bytes
CID	Arguments

Σχήμα 3.18: Δομή MACCommand

FPort Αν ένα FRMPayload δεν είναι κενό τότε το πεδίο FPort πρέπει να περιέχει μία τιμή. Αν έχει τιμή 0 τότε σημαίνει ότι περιέχει MACCommands. Οι τιμές που μπορεί να πάρει είναι από 1 έως 223 . Η τιμή 224 είναι για δοκιμές το επίπεδο του πρωτοκόλλου MAC. Οι τιμές από 225 έως 255 είναι για μελλοντική χρήση.

MAC Frame Payload Encryption (FRMPayload) Το FRMPayload είναι το χρήσιμο κομμάτι του μηνύματος είναι αυτό που έχει έρθει από τον Application Server. Τα περιεχόμενα του FRMPayload πρέπει να είναι κρυπτογραφημένα και κρυπτογραφούνται πριν υπολογιστεί το MIC.

Message Integrity Code MIC Το MIC είναι ένας αριθμός ο οποίος υπολογίζεται από όλα τα παιδιά του μηνύματος (MHDR | FHDR | FPort | FRMPayload) και χρησιμοποιείται για να πιστοποιήσει την ακεραιότητα του μηνύματος.

3.5 Ασφάλεια

Υπάρχουν δύο τρόποι για πιστοποίησης και ενεργοποίησής ενός End Node στο δίκτυο ο ένας είναι OTAA Over The Air Activation και ο άλλος είναι ο ABR Activation by Personalization. Μετά από κάθε ενεργοποίηση πρέπει σε κάθε End Node να υπάρχουν οι εξής πληροφορίες μια Device Address (DevAddr), ένα Application Identifier.

3.5.1 Device Address DevAddr

Είναι η διεύθυνση του δικτύου η οποία αποτελείται από 32 bit τα 25 λιγότερο σημαντικά είναι η διεύθυνση του δικτύου η οποία είναι αυθαίρετη. Τα 7 περισσότερο σημαντικά bit είναι το ID του δικτύου, το οποίο έχει να κάνει με το φορέα υλοποίησης του δικτύου.

3.5.2 Application intentifier AppEUI

Το AppEUI είναι ένα παγκόσμιο αναγνωριστικό της εφαρμογής στο χώρο διευθύνσεων IEEE EUI64 που αναγνωρίζει με μοναδικό τρόπο την εφαρμογή. Το AppEUI αποθηκεύεται στην τελική συσκευή πριν εκτελεστεί η διαδικασία ενεργοποίησης.

3.5.3 Network Session Key NwkSKey

Το NwkSKey είναι το κλειδί σύνδεσης του δικτύου που είναι μοναδικό για κάθε End Node. Χρησιμοποιείται από το Network Server και το End Node για τον υπολογισμό και την επαλήθευση του κώδικα ακεραιότητας μηνύματος (MIC) όλων των μηνυμάτων για την εξασφάλιση της ακεραιότητας των δεδομένων. Χρησιμοποιείται επίσης για την κρυπτογράφηση και αποκρυπτογράφηση του Payload στα MAC πακέτα.

3.5.4 Application Session Key AppSKey

Το AppSKey είναι ένα κλειδί συνόδου εφαρμογής μοναδικό για κάθε End Node. Χρησιμοποιείται από τον Application Server και τον End Node για την κρυπτογράφηση και αποκρυπτογράφηση των δεδομένων που προέρχονται σε επίπεδο εφαρμογής. Τα δεδομένα πριν σταλούν στο Network Server κρυπτογραφούνται με αυτό το κλειδί και αποκρυπτογραφουνται στο End Node με το ίδιο κλειδί, αυτό διασφαλίζει την ιδιωτικότητα των δεδομένων αλλά όχι την ακεραιότητα τους.

3.5.5 Over The Air Activation OTAA

Στη διαδικασία για την σύνδεση ενός End Node στο δίκτυο πρέπει εκ των προτέρων ο End Node και ο Network Serverva γνωρίζουν τις παρακάτω πληροφορίες.

AppEUI

Βλέπε παραπάνω

End-device identifier DevEUI

Το DevEUI είναι ένα παγκόσμιο αναγνωριστικό συσκευής στο χώρο διευθύνσεων IEEE EUI64 που αναγνωρίζει με μοναδικό τρόπο το End Node.

Application Key AppKey

Το AppKey είναι ένα βασικό κλειδί AES-128 ειδικά για τον End Node. Οποτεδήποτε End Node συνδέεται σε δίκτυο LoRaWAN μέσω πρέπει να έχει ένα AppKey το οποίο γνωρίζει και ο Network Server. Το AppKey χρησιμοποιείται για την εξαγωγή των κλειδιών NwkSKey και AppSKey που είναι μοναδικά για κάθε End Node και χρησιμοποιούνται για κρυπτογράφηση και επαλήθευση σε επίπεδο δικτύου και εφαρμογής.

Διαδικασία Join

Για να συνδεθεί ένας End Node μέσω OTAA σε ένα δίκτυο LoRaWAN πρέπει να κάνει ένα Join Request αυτό περιέχει τις πληροφορίες AppEUI , DevEUI ένα τυχαίο 16bit αριθμό (DevNonce).

Join Request		
8 Bytes	8 Bytes	8 Bytes
AppEUI	DevEUI	DevNonce

Σχήμα 3.19: Δομή Join Request

Το Join Request δεν είναι κρυπτογραφημένο και το MIC παράγεται από
cmac = aes128_cmac(AppKey, MHDR | AppEUI | DevEUI | DevNonce)
MIC = cmac[0..3]
Αν το αίτημα γίνει δεκτό μέσα σε προκαθορισμένα χρονικά διαστήματα JOIN_ACCEPT_DELAY1 και JOIN_ACCEPT_DELAY2 επιστρέφει ένα μήνυμα Join Accept.

Join Accept					
3 Bytes	3 Bytes	4 Bytes	1 Byte	1 Byte	15 Bytes
AppNonce	NetID	DevAddr	DLSettings	RxDelay	CFList(ή)

Σχήμα 3.20: Δομή Join Accept

- AppNonce
  Τυχαίος αριθμός 3 Bytes που παράγεται στο Network Server
- DLSettings
  Περιέχει ρυθμίσεις για την Downlink επικοινωνία.
- RxDelay
  Ορίζει τους χρόνους καθυστέρησης των παραθύρων RX.
- CFList
  Περιέχει πληροφορίες και ρυθμίσεις που έχουν να κάνουν με τις ιδιαίτερες γεωγραφικές ρυθμίσεις οτυ LoRaWAN για κάθε γεωγραφική περιοχή σε φυσικό επίπεδο.

Με βάση τις παραπάνω πληροφορίες που είναι πλέον γνωστές και στο End Node και στο Network Server δημιουργεί ο καθένας ξεχωριστά τα κλειδιά συνόδου.
NwkSKey = aes128_encrypt(AppKey, 0x01 | AppNonce | NetID | DevNonce | pad 16)
AppSKey = aes128_encrypt(AppKey, 0x02 | AppNonce | NetID | DevNonce | pad 16)
Το MIC για το Join-Accept υπολογίζεται ως εξής
cmac = aes128_cmac(AppKey, MHDR | AppNonce | NetID | DevAddr | DLSettings | RxDelay | CFList)
MIC = cmac[0..3]

Το Join-Accept μήνυμα κρυπτογραφείται με το παρακάτω τρόπο.
aes128_decrypt(AppKey, AppNonce | NetID | DevAddr | DLSettings | RxDelay | CFList | MIC)

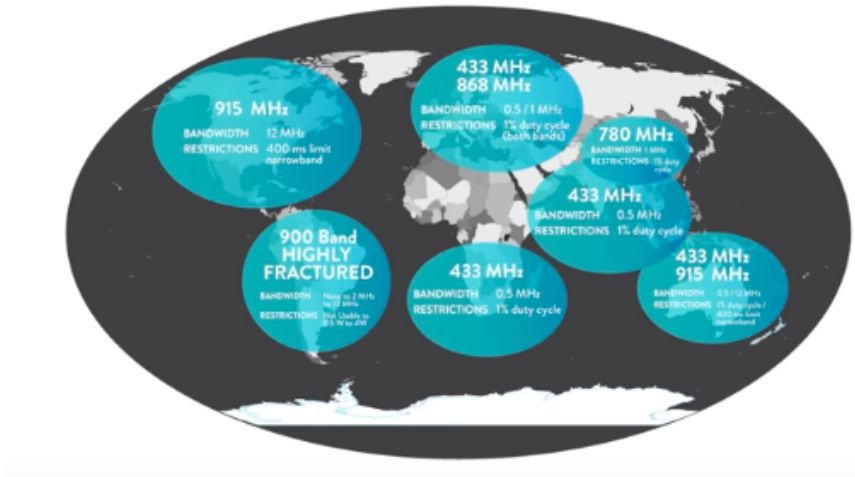
Πλέων και τα δύο μέρη έχουν τις πληροφορίες που χρειάζονται για την ανταλλαγή μηνυμάτων.

3.5.6 Activation By Personalization ABP

Σε ορισμένες περιπτώσεις, οι End Nodes μπορούν να είναι ρυθμισμένοι με τις κατάλληλες παραμέτρους από το κατασκευαστή για να μπορούν να συνδεθούν στο δίκτυο χωρίς καμία άλλη διαδικασία. Η ενεργοποίηση ABP συνδέει άμεσα ένα End Node σε ένα δίκτυο παρακάμπτοντας την αίτηση σύνδεσης Join. Η ενεργοποίηση ενός End Node με ABP σημαίνει ότι το DevAddr και τα κλειδιά NwkSKey και AppSKey αποθηκεύονται απευθείας στο End Node αντί για τα DevEUI, AppEUI και AppKey που είχαμε στο OTAA. Στην ABR ενεργοποίηση κάθε End Node διαθέτει ένα μοναδικό ζεύγος NwkSKey και AppSKey τα οποία είναι γνωστά α) το NwkSKey στο Network Server και β) το AppSKey στον Application server. Τα κλειδιά και οι κρίσιμες πληροφορίες πρέπει να διασφαλίζεται ότι δεν θα διαρρεύσουν θέτοντας σε κίνδυνο το δίκτυο.

3.6 Παράμετροι ανά περιοχή

Το LoRaWAN εναρμονίζεται πλήρως με τις ιδιαίτερες συνθήκες και νόμους που ισχύουν για κάθε γεωγραφική περιοχή [24] , παρακάτω θα εστιάσουμε στις ιδιαίτερες προδιαγραφές που ισχύουν για την περιοχή της Ευρώπης όπως ορίζονται στο ETSI [EN300.220] [25]. Στην Ελλάδα ισχύουν τα Chanel Plan EU863-870 με συχνότητες 868 - 870 MHz και EU433 με συχνότητες 433.05- 434.79 MHz. Εμείς θα εστιάσουμε στο Channel Plan που ισχύει γενικότερα στην Ευρώπη το EU863-870 ή πιο απλά EU868.



Σχήμα 3.21: Χάρτης περιοχών με τις ελεύθερες συχνότητες που χρησιμοποιεί το πρωτόκολλο LoRaWAN [2]

3.6.1 LoRa Preamble

Σύμφωνα με το EU868 το Preamble που πρέπει να χρησιμοποιεί το LoRaείναι 8 σύμβολα.

3.6.2 LoRa Default Channels

Κάθε End Node σύμφωνα με το EU868 πρέπει να υποστηρίζει υποχρεωτικά ως ελάχιστο σετ ρυθμίσεων τα τρία παρακάτω κανάλια:

BandWidth [Khz]	Channel Frequency [Mhz]	DR / Bit Rate	Duty Cycle
125	868.10	DR0-DR5 / 0.3 /5 Kbps	<1%
125	868.30	DR0-DR5 / 0.3 /5 Kbps	<1%
125	868.50	DR0-DR5 / 0.3 /5 Kbps	<1%

Πίνακας 3.4: Ρυθμίσεις που επιτρέπονται στα κανάλια του LoraWAN

Για την πρόσβαση στο φυσικό επίπεδο η ETSI έχει κάποιους περιορισμούς σε ότι αφορά το μέγιστο χρόνο εκπομπής ή το μέγιστο χρόνο εκπομπής ανά ώρα. Το LoRa υλοποιεί αυτό το περιορισμό χρησιμοποιώντας το Duty Cycle.

3.6.3 Duty Cycle

Το Duty Cycle ορίζει το ποσοστό χρόνου που μπορεί να χρησιμοποιείται ένα κανάλι από μια συσκευή. Αυτό είναι ένας περιορισμός για να μην μονοπωλεί μια συσκευή κάποιο κανάλι και να έχουν όλοι την ευκαιρία να εκπέμψουν σε αυτό. Δεν είναι προαιρετικό αλλά ένας νομικά υποχρεωτικός περιορισμός. Σύμφωνα με τις οδηγίες όπως ορίζονται στο ETSI [EN300.220] για συχνότητες 868,000 MHz - 868,600 MHz 25 mW erp (14dBm) είναι 1%

Ειδικά για την Ελλάδα [26]

Frequency	Duty Cycle	TX Power
863,0–868,0	<=0.1%	25mW - 14dBm
868,0–868,6	<=1%	25mW - 14dBm
868,7–869,2	<=1%	25mW - 14dBm
869,4–869,65	<=10%	500mW - 20dBm
869,7–870 MHz	<=0.1%	25mW - 14dBm

Πίνακας 3.5: Συχνότητες και επιτρεπόμενο Duty Cycle και Power

3.6.4 Επιπλέον LoRa Channels

Εκτός από τα τρία υποχρεωτικά κανάλια το LoRa μπορεί να υποστηρίξει σύνολο 16 κανάλια. Ο Network Server ενημερώνει τον End Node για τα κανάλια έκτος από τα βασικά που υποστηρίζονται από το δίκτυο. Η ενημέρωση γίνεται μετά από επιτυχημένο Join του End Node στο Join Accept μήνυμα που λαμβάνει μέσα στο πεδίο CFList μεταφέρεται η λίστα με 5 επιπλέον συχνότητες καναλιών που υποστηρίζονται. Για όλα τα κανάλια το DR είναι από DR0 μέχρι DR5 και το BandWidth 125KHz. Επίσης με εντολή MACCommand NewChannelReq απο τον Network Server ενημερώνεται ο End Node για τα υπόλοιπα κανάλια που υποστηρίζονται από το δίκτυο.

3.6.5 EU868 Μέγιστο Payload Size

Το μέγιστο MACPayload μέγεθος πακέτου μηνύματος LoRa δίνεται στον παρακάτω πίνακα ο οποίος προκύπτει από τους περιορισμούς που έχει στο φυσικό επίπεδο το LoRa. Το μέγιστο Payload σε επίπεδο εφαρμογής φαίνεται στη στήλη 3 από την οποία έχει αφαιρεθεί το περιεχόμενο του πεδίου FOpt.

Data Rate	MAX Payload	MAX Payload - FOpt
0	59	51
1	59	51
2	59	51
3	123	115
4	230	242
5	230	242
6	230	242
7	230	242

Πίνακας 3.6: Data Rate - Max Payload

3.6.6 EU868 Data Rate - Bit Rate- TX Power

Data Rate	Configuration	Indicative physical bit rate [bit/s]
0	SF 12 / 125 KHz	250
1	SF 11 / 125 KHz	440
2	SF 10 / 125 KHz	980
3	SF 9 / 125 KHz	1760
4	SF 8 / 125 KHz	3125
5	SF 7 / 125 KHz	5470
6	SF 7 / 250 KHz	11000

Πίνακας 3.7: Data Rate - Bit Rate

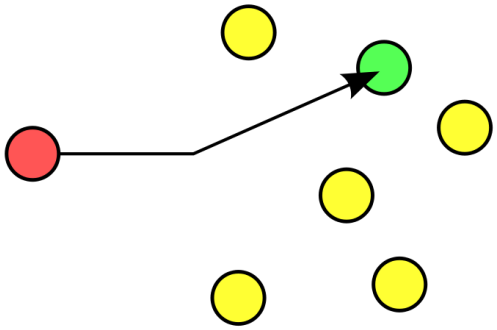
TX Pow	Configuration (EIRP)
0	Max EIRP
1	Max EIRP - 2 dB
2	Max EIRP - 4 dB
3	Max EIRP - 6 dB
4	Max EIRP - 8 dB
5	Max EIRP - 10 dB
6	Max EIRP - 12 dB
7	Max EIRP - 14 dB

Πίνακας 3.8: Data Rate - TX Power

3.7 Multicast

3.7.1 Unicast

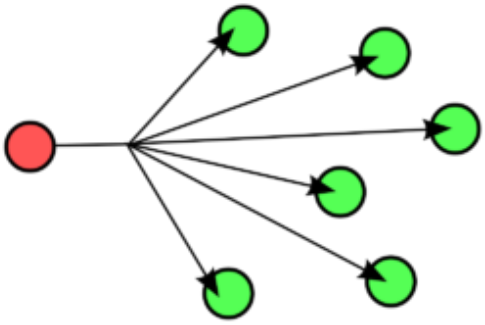
Με τον όρο Unicast [3] αναφερόμαστε σε μία σύνδεση σημείο προς σημείο μεταξύ ενός εξυπηρετητή και ενός τελικού χρήστη. Σε μία Unicast σύνδεση ο εξυπηρετητής συνδέεται με έναν τελικό χρήστη και στέλνει πληροφορίες μόνο σε αυτόν. Για να στείλει πληροφορίες σε κάποιο άλλο χρήστη πρέπει να διακόψει τη σύνδεση με τον προηγούμενο να δημιουργήσει μία καινούργια σύνδεση με το νέο χρήστη και να στείλει πληροφορίες.



Σχήμα 3.22: Παράδειγμα Unicast σύνδεσης [3]

3.7.2 BroadCast

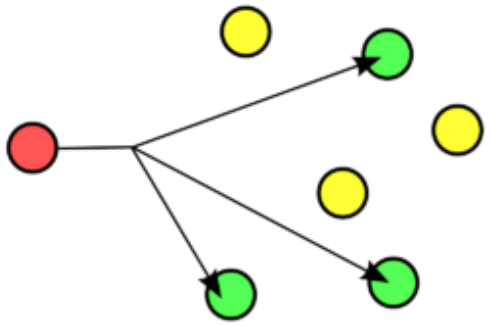
Broadcast είναι μία επικοινωνία ένας προς όλους , όταν ένας εξυπηρετητής στέλνει ένα μήνυμα Broadcast τότε αυτό λαμβάνεται από όλους τους τελικούς χρήστες ταυτόχρονα.



Σχήμα 3.23: Παράδειγμα Broadcast σύνδεσης [3]

3.7.3 Multicast

Multicast [27] σύνδεση είναι μία επικοινωνία ένας προς μερικούς, είναι πιο ευφυή σύνδεση από ότι η Broadcast η οποία στέλνει δεδομένα σε όλους ανεξαιρέτως. Η Multicast στέλνει ένα μήνυμα όχι προς όλους τους τελικούς χρήστες αλλά σε μία ομάδα αυτών. Η Multicast είναι η μετάδοση πακέτων σε μια ομάδα μηδέν ή περισσότερων χρηστών που ονομάζεται ομάδα Multicast, η οποία αναγνωρίζεται από μια ενιαία διεύθυνση προορισμού. Μια ομάδα Multicast είναι ένα σύνολο των χρηστών του δικτύου που ενδιαφέρονται να μοιραστούν ένα συγκεκριμένο σύνολο δεδομένων. Συνήθως, η συμμετοχή σε μια ομάδα Multicast είναι δυναμική: δηλαδή, οι χρήστες μπορούν να ανήκουν ή όχι ως μέλη στις ομάδες ανά πάσα στιγμή. Δεν υπάρχει περιορισμός στη θέση ή τον αριθμό των μελών σε μια ομάδα Multicast . Η Multicast εκπομπή είναι κατά κανόνα πολύ λιγότερο αξιόπιστη από την Unicast. Δεδομένου ότι η επικοινωνία Multicast πραγματοποιεί επικοινωνίες από σημείο σε πολλαπλά σημεία, θα χρειαστούν πολλαπλές επιβεβαιώσεις για να εξασφαλιστεί η λήψη σε όλους τους παραλήπτες. Έτσι ουσιαστικά δεν υπάρχουν επιβεβαιώσεις (Acknowledgement ACK) για πακέτα Multicast, άρα δεν είναι δυνατό ο αποστολέας να γνωρίζει εάν απαιτείται ή όχι αναμετάδοση. Συνεπώς, μπορεί να υπάρξει υψηλός ρυθμός σφάλματος λόγω έλλειψης αναμετάδοσης. Δεν είναι ασυνήθιστο να υπάρχει ποσοστό απώλειας πακέτων 5% ή περισσότερο, το οποίο είναι ιδιαίτερα ενοχλητικό για βίντεο και άλλα περιβάλλοντα όπου απαιτούνται υψηλά ποσοστά δεδομένων και υψηλή αξιοπιστία.



Σχήμα 3.24: Παράδειγμα Multicast σύνδεσης [3]

3.7.4 LoRaWAN Multicasting

Η LoRaWAN Alliance παρέχει ένα πρότυπο για το πως να υλοποιείται μια Multicast μετάδοση σε ένα LoRaWANδίκτυο. Στο LoRaWAN μία Multicast μετάδοση υλοποιείται στο επίπεδο εφαρμογής, όλα όσα περιγράφουν στη συνέχεια είναι σε επίπεδο εφαρμογής [28]. Για να μπορέσει να γίνει μία Multicast μετάδοση πρέπει οι End Nodes να είναι σε Class B ή Class C για να μπορούν να δεχτούν μηνύματα ανά πάσα στιγμή και ταυτόχρονα. Όλα τα UpLoad μηνύματα είναι Unicast μεταδόσεις, οι συσκευές στέλνουν σαν Class A ή Class B. Τα μηνύματα σε μια Multicast μετάδοση στο DownLink είναι κρυπτογραφημένα με το Multicast Application Session Key McAppSKey που είναι κοινό για όλη την Multicast ομάδα.

Ορισμός περιεχομένου Multicast Ομάδων

Όλοι οι End Nodes που ανήκουν σε μια Multicast ομάδα πρέπει να έχουν για κάθε ομάδα τις παρακάτω πληροφορίες, επίσης οι ίδιες πληροφορίες πρέπει να υπάρχουν και στην εφαρμογή που διαχειρίζεται τις ομάδες και τις Multicast μεταδόσεις.

1. Το McGroupID: ένας ακέραιος από 0 έως 3 , είναι ένας δείκτης με τον οποίο χωρίζουμε μια ομάδα Multicast σε 4 υποομάδες. Μια τελική συσκευή υποστηρίζει το πολύ 4 υποομάδες ταυτόχρονα και τουλάχιστον μια με McGroupID 0.
2. Multicast Address: Είναι η διεύθυνση δικτύου για την ομάδα Multicast, είναι 4 Bytes και είναι κοινή για όλες τις τελικές συσκευές της ομάδας.
3. Ένα κλειδί κρυπτογράφησης μοναδικό για την Multicast ομάδα Multicast Key (McKey) από το οποίο δημιουργείται ένα McAppSKey για την κρυπτογράφηση σε επίπεδο εφαρμογής της Multicast μετάδοσης και McNwkSKey για την κρυπτογράφηση σε επίπεδο δικτύου της Multicast μετάδοσης. Το McKey είναι μοναδικό για κάθε ομάδα αλλά κοινό για όλες τις τελικές συσκευές μιας ομάδας Multicast.
4. Ένας μετρητής πλαισίων Multicast μεταδόσεων.

Μηνύματα Ελέγχου Multicast

Για να μπορέσει να υλοποιηθεί μία Multicast μετάδοση υπάρχουν μία σειρά μηνυμάτων τα οποία στέλνει ο Application Server στα End Nodes για να μπορούν να υλοποιήσουν τη Multicast μετάδοση , αυτά τα μηνύματα γίνονται σε Unicast μετάδοση ξεχωριστά για κάθε End Node και στη Downlink ζεύξη χρησιμοποιείται πόρτα 200. Κάθε μήνυμα ελέγχου Multicast είναι ένα πάκέτο από μια η περισσότερες εντολές Multicast Commands. Οι εντολές όλες είναι σε επίπεδο εφαρμογής και δεν έχουν σχέση με τις MACCommands.

Command 1	Command 1 Payload	Command 2	Command 2 Payload	.....

Σχήμα 3.25: Μήνυμα με εντολές ελέγχου Multicast

Εντολές ελέγχου Multicast Commands

Παρακάτω παρατίθενται συνοπτικά οι εντολές ελέγχου για το Multicast για το πρότυπο του LoRaWAN

PackageVersion Ο Application Server στέλνει μια εντολή PackageVersionReq για να ρωτήσει ποια είναι η έκδοση των προδιαγραφών των πακέτων με τις εντολές που χρησιμοποιεί ο End Node. Ο End Node απαντάει με μια εντολή PackageVersionAns.

McGroupStatus Ο Application Server στέλνει μια εντολή ReqGroupMask με μια μάσκα 4bit ρωτώντας αν μπορεί να υποστηρίξει μια συγκεκριμένη υποομάδα. Όπως είπαμε προηγουμένως κάθε ομάδα ανάλογα με το McGroupID μπορεί να υποστηρίξει μέχρι 4 υποομάδες. Ο End Node απαντάει με μια εντολή AnsGroupMask. Αν υποστηρίζεται επιστρέφει το ίδιο αριθμό με την μάσκα αν όχι επιστρέφει το μέγιστο GroupID.

McGroupSetup Ο Application Server στέλνει μια εντολή McGroupSetupReq όταν θέλει να ενημερώσει ένα End Node για μια νέα ομάδα που ανήκει. Μαζί με την εντολή του στέλνει και το περιεχόμενο της ομάδας δηλαδή όλες τις πληροφορίες που πρέπει να γνωρίζει για να είναι μέλος της.

McGroupSetup				
1 Byte	4 Bytes	16 Bytes	4 Bytes	4 Bytes
McGroupIDHeader	McAddr	McKey encrypted	minMcFCount	maxMcFCount

Σχήμα 3.26: Εντολή ελέγχου McGroupSetup

- McGroupID
  Αριθμός υποομάδων που μπορεί να υποστηρίξει ταυτόχρονα.
- McAddr
  Η διεύθυνση της ομάδας χρησιμοποιείται σαν διεύθυνση DevAddr κατά την εκπομπή όχι για ένα End Node αλλά για όλη την ομάδα.
- McKey encrypted
  Είναι το κοινό κλειδί της ομάδας από αυτό και με το McAddr και το GenAppKey παράγονται τα McNwkSKey και McAppSKey που είναι μοναδικά για κάθε End Node.
- minMcFCount
  Είναι ο αύξων αριθμός του επόμενου Multicast μηνύματος που θα στείλει ο εξυπηρετητής , αυτός ο αριθμός παράγεται στον εξυπηρετητή.
- maxMcFCount Είναι ο μέγιστος αριθμός μηνυμάτων που μπορεί να δεχτεί σε κατάσταση Multicast ένας End Node από αυτή την ομάδα , έτσι θέτουμε ένα ανώτατο όριο χρήσης της ομάδας και της Multicast εκπομπής.

Αφού λάβει και αποθηκεύσει ο End Node τις παραπάνω πληροφορίες επιστρέφει ένα McGroupSetupAns το οποίο ενημερώνει αν μπόρεσε να δεχτεί ή όχι την ομάδα.

McGroupDelete Ο Application Server στέλνει μια εντολή McGroupDeleteReq όταν θέλει να αφαιρέσει μια ομάδα από ένα End Node. Ο End Node ανταποκρίνεται με μια εντολή McGroupDeleteAns.

McClassCSession Συνήθως ο End Node μετά το Join βρίσκεται σε κατάσταση Class A για να μπορέσει να υλοποιηθεί μία Multicast εκπομπή πρέπει αυτός να μπορεί να γυρίσει σε Class B ή Class C , για να έρθει από Class A που αρχικά βρίσκεται σε Class

C, ο Application Server στέλνει μία εντολή McClassCSessionReq. Το περιεχόμενο της εντολή είναι:

McClassCSessionReq				
1 Byte	4 Bytes	3 Bytes	1 Bytes	1 Bytes
McGroupIDHeader	Session Time	Session TimeOut	DLFrequ	DR

Σχήμα 3.27: Εντολή ελέγχου McGroupSetup

- McGroupIDHeader
  Το McGroupID
- Session Time
  Ο χρόνος που ξεκινάει να μπαίνει ο End Node σε Class C , είναι ορισμένο σε δευτερόλεπτα με αφετηρία 00:00:00, Sunday 6 th of January 1980.
- Session TimeOut
  Είναι ο μέγιστος χρόνος σε δευτερόλεπτα που η συσκευή θα μείνει σε Class C μετά από αυτό το χρονικό διάστημα επιστρέφει σε Class A.
- DlFrequ
  Είναι η συχνότητα που θα γίνει η Multicast εκπομπή. Τα 24bit είναι ο αριθμός επί 100 που ορίζουν την συχνότητα εκπομπής σε Hz.
- DR
  Ορίζει το Data Rate που θα χρησιμοποιηθεί στην Multicast εκπομπή.

Ο End Node απαντάει σε μια εντολή McClassCSessionReq με McClassCSessionAns και επιβεβαιώνει στέλνοντας το χρόνο που θα μπει σε Class C ή με κάποιο μήνυμα λάθους.

McClassBSessionReq Βάζει το End Node σε κατάσταση Class B και τον ενημερώνει για τους χρόνους, την συχνότητα και ότι χρειάζεται για να γίνει η μετάδοση. Ο End Node απαντάει με ένα μήνυμα McClassBSessionAns.

Class C Multicast Downlink

Για να υλοποιηθεί μια Multicast εκπομπή στο LoRaWAN [29], οι συσκευές πρέπει να βρίσκονται σε κατάσταση λειτουργίας Class B ή Class C, στην προκειμένη περίπτωση αναφερόμαστε σε Class C. Οι συσκευές σε Class C μπορούν να λαμβάνουν μηνύματα Multicast αφού είναι συνεχώς σε κατάσταση λήψης. Στη Multicast εκπομπή η διεύθυνση , το McKey, το McNwkSKey και το McAppSKey προέρχονται από το επίπεδο εφαρμογής. Οι παρακάτω περιορισμοί ισχύουν στις μεταδόσεις Multicast σε Class C:

- Οι μεταδόσεις Multicast δεν επιτρέπεται να μεταφέρουν MAC εντολές, ούτε στο πεδίο FOpt, ούτε στο Payload, ούτε να γίνονται εκπομπές στη θύρα 0. Αυτό συμβαίνει επειδή μια μετάδοση Multicast δεν είναι το ίδιο ασφαλής, δεν εγγυάται την ακεραιότητα των δεδομένων , δεν δέχεται επιβεβαίωση και δεν στέλνει απαντήσεις όπως μια μετάδοσης Unicast αφού είναι μονόδρομη.

- Τα bits ACK και ADRACKReq ΠΡΕΠΕΙ να είναι μηδέν.
- Το πεδίο MType ΠΡΕΠΕΙ να έχει τιμή για την επιλογή Unconfirmed Data Down.
- Το bit FPend είναι 0 έτσι δείχνει ότι DownLink ζεύξη ο End Node δεν περιμένει να λάβει επιπλέον μήνυμα. Αυτό συμβαίνει γιατί όταν ο End Node είναι σε Class C διατηρεί τον δέκτη ενεργό συνεχώς, έτσι το bit FPending δεν χρειάζεται να ενημερώνει το End Node να περιμένει και άλλο μήνυμα αφού ούτως η άλλος είναι σε κατάσταση λήψης.

Κεφάλαιο 4

Θεωρία

4.1 Broadcast Encryption

4.1.1 Εισαγωγή

Ας υποθέσουμε ότι έχουμε ένα σύνολο χρηστών στο οποίο θέλουμε να στείλουμε ένα μήνυμα σε όλους ταυτόχρονα (Broadcast) στη συνέχεια ας υποθέσουμε ότι από αυτό το σύνολο των χρηστών δεν θέλουμε να λάβουν όλοι το μήνυμα αλλά κάποιοι από αυτούς, μια συγκεκριμένη ομάδα (Multicast). Έπειτα μετά από λίγο φτιάχνουμε μία άλλη ομάδα, ένα άλλο υποσύνολο από το σύνολο των χρηστών στους οποίους θα στείλουμε ένα μήνυμα και θα πρέπει να λάβουν ταυτόχρονα το μήνυμα που στέλνουμε . Γενικά επιθυμούμε να δημιουργούμε δυναμικά υποσύνολα από ένα σύνολο χρηστών στα οποία θα στέλνουμε ένα μήνυμα , το οποίο θα πρέπει να λαμβάνουν ταυτόχρονα. Αν έχουμε ένα ασύρματο δίκτυο και όλοι δέκτες είναι συντονισμένοι στο ίδιο κανάλι μπορούμε να κάνουμε μια εκπομπή και όλοι να λάβουν το μήνυμα ταυτόχρονα (Broadcast). Αν όμως θέλουμε να λάβουν το μήνυμα κάποιοι συγκεκριμένοι δέκτες ,μια ομάδα αυτών από το σύνολο με τους δέκτες που είναι συντονισμένοι ,αυτό είναι ένα πρόβλημα σε ένα ασύρματο δίκτυο. Γιατί όλοι ακούν σε ένα συγκεκριμένο κανάλι αλλά εμείς θέλουμε μόνο κάποιοι από αυτούς να μπορέσουν να το αναγνωρίσουν το μήνυμα κάθε φορά.
Μία εύκολη λύση είναι να δώσουμε σε κάθε έναν από τους δέκτες ένα κλειδί κρυπτογράφησης στη συνέχεια κρυπτογραφούμε το μήνυμα με αυτό το ξεχωριστό κλειδί για κάθε δέκτη και να το μεταδίδουμε κάθε φορά για έναν από αυτούς έτσι κάθε φορά ένας από τους δεκτές θα μπορεί να λάβει και να αποκρυπτογραφήσει με επιτυχία το μήνυμα (Unicast). Το πρόβλημα είναι ότι θα πρέπει να μεταδώσουμε το μήνυμα μας τόσες φορές όσοι είναι και οι δέκτες που θα το λάβουν. Μία δεύτερη πιθανή λύση είναι να χωρίσουμε από πριν τους δέκτες σε ομάδες για κάθε ομάδα δεκτών να δημιουργήσουμε ένα κλειδί κρυπτογράφησης το οποίο θα το μοιράσουμε στους δέκτες κάθε ομάδας ,έτσι όταν εμείς στέλνουμε ένα μήνυμα κρυπτογραφημένο με αυτό το κλειδί θα μπορεί να το λάβει μία συγκεκριμένη ομάδα δεκτών. Αυτή η λύση φαίνεται να καλυτερεύει τα πράγματα μιας και δεν χρειάζεται να στείλουμε ένα μήνυμα τόσες φορές όσοι είναι και οι δέκτες που θα το λάβουν (Multicast). Αν οι ομάδες είναι σταθερές στέλνουμε μία φορά το μήνυμα κρυπτογραφημένο με το κλειδί της ομάδος και όλοι οι δέκτες της ομάδας θα πάρουν το μήνυμα. Τι γίνεται όμως στην περίπτωση που οι ομάδες είναι δυναμικές. Δηλαδή κάθε φορά ο συνδυασμός των δεκτών που θα αποτελούν μία ομάδα αλλάζει. Σε

αυτή την περίπτωση θα πρέπει να δημιουργήσουμε τόσες υποομάδες όσα είναι και τα πιθανά υποσύνολα που μπορούν να προκύψουν από το σύνολο των δεκτών , για κάθε μία από αυτές τις ομάδες θα μπορούμε να δημιουργήσουμε ένα κλειδί , να μοιράσουμε τα κλειδιά των ομάδων στους δέκτες που ανήκουν στην κάθε ομάδα. Στέλνοντας λοιπόν ένα μήνυμα κρυπτογραφημένο με το κλειδί της ομάδας θα το πάρουν πάλι οι συγκεκριμένοι δέκτες που ανήκουν σε αυτή την ομάδα , καταλαβαίνουμε βέβαια ότι αυτό είναι αδύνατον να συμβεί γιατί το σύνολο των ομάδων που θα πρέπει να δημιουργηθούν θα είναι τεράστιο.

4.1.2 BroadCast Encryption

Αυτό που χρειαζόμαστε είναι έναν τρόπο να μπορούμε δίδοντας διάφορα κλειδιά στους δέκτες να μπορούμε να στέλνουμε ένα μήνυμα και να το λαμβάνει ένα υποσύνολο των συνολικών δεκτών που είναι συντονισμένοι στο ασύρματο κανάλι μας . Αυτό ονομάζεται Broadcast Encryption(κρυπτογράφηση πολυεκπομπής) [30] και στη βιβλιογραφία υπάρχουν πάρα πολλοί αλγόριθμοι που υλοποιούνται τέτοιου είδους διαδικασίες. Η κρυπτογράφηση πολυεκπομπής (Broadcast Encryption) [31] είναι το κρυπτογραφικό πρόβλημα της παράδοσης κρυπτογραφημένου περιεχομένου μέσω ενός καναλιού εκπομπής με τέτοιο τρόπο ώστε μόνο συγκεκριμένοι δέκτες να μπορούν να αποκρυπτογραφούν το περιεχόμενο. Η πρόκληση προκύπτει από την απαίτηση ότι το σύνολο των δεκτών μπορεί να αλλάξει σε κάθε εκπομπή και επομένως πρέπει να ανακληθούν μεμονωμένοι δέκτες ή ομάδες δεκτών , πρέπει να είναι δυνατή η χρήση εκπομπών μετάδοσης χωρίς να επηρεάζονται οι υπόλοιποι δέκτες.

Οι αλγόριθμοι Broadcast Encryption χωρίζονται σε δύο μεγάλες κατηγορίες

- Σε αυτούς που αναφέρονται σε σταθερό αριθμό δεκτών ή σε αυτούς που αναφέρονται σε μεταβλητό αριθμό δεκτών. Αν λοιπόν ο αριθμός των δεκτών μας είναι σταθερός δηλαδή δεν μπαίνουν νέοι δέκτες στο σύστημα και δεν φεύγουν δέκτες από το σύστημα αλλά θεωρούμε ότι ο αριθμός τους είναι σταθερός.
- Εάν οι δέκτες έχουν αμφίδρομη επικοινωνία με τον κεντρικό σταθμό ή μεταξύ τους ανταλλάσσοντας πληροφορίες ή καταγράφοντας πληροφορίες από προηγούμενες καταστάσεις.

Μια περίπτωση είναι όταν οι δέκτες είναι Stateless. Σε ένα τέτοιο σενάριο, ένας δέκτης δεν καταγράφει το ιστορικό μεταδόσεων έτσι ώστε να αλλάζει ανάλογα την κατάστασή του. Αντ άυτού, η λειτουργία του βασίζεται στην τρέχουσα μετάδοση και στην αρχική διαμόρφωσή της. Οι Stateless δέκτες είναι σημαντικοί σε περίπτωση όπου ο δέκτης είναι μια συσκευή που δεν είναι συνεχώς On-Line, όπως μια συσκευή αναπαραγωγής πολυμέσων (π.χ. DVD player όπου η ¨μετάδοση¨ είναι ο τρέχων δίσκος), ένας δορυφορικός δέκτης (GPS) και σε περιπτώσεις Multicast εφαρμογών. Τα κρίσιμα θέματα συνήθως σε τέτοιους αλγόριθμους είναι

- Το πλήθος και το μέγεθος των πληροφοριών που πρέπει να σταλούν μαζί με το μήνυμα σαν επικεφαλίδα για να μπορέσει να γίνει η αποκρυπτογράφηση. Αυτό σημαίνει περισσότερα δεδομένα κατά την εκπομπή.
- Τα δεδομένα που είναι αποθηκευμένα σε κάθε χρήστη, το πλήθος των κλειδιών, έτσι ώστε αυτός να μπορεί να αποκρυπτογραφήσει το μήνυμα με τη βοήθεια των

στοιχείων της επικεφαλίδας.

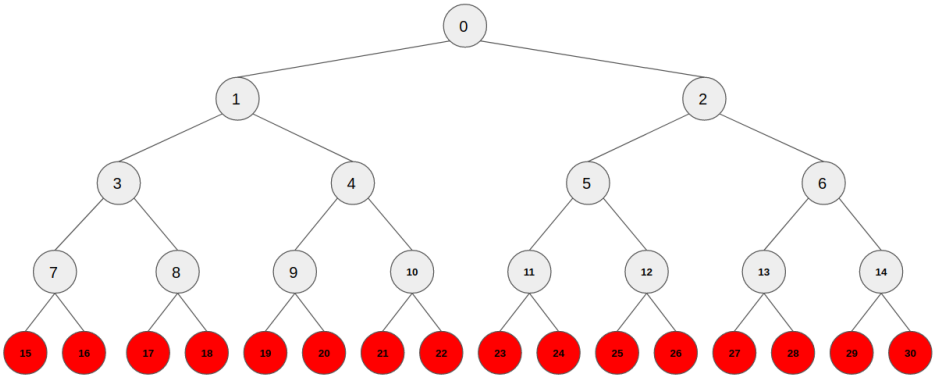
- Η πολυπλοκότητα και ο χρόνος του για τις απαιτούμενες αποκρυπτογραφήσεις.

4.1.3 Subset Difference Method (SD)

Η τεχνική που θα περιγράψουμε στη συνέχεια είναι μία από τις βασικές τεχνικές πού χρησιμοποιήθηκαν στο Broadcast Encryption και που πάνω σε αυτή βασίστηκε ένα πλήθος άλλων τεχνικών . Η τεχνική ονομάζεται Subset Difference Method (SD) [4] και είναι μία παραλλαγή της Complete Subtree Method (CS).

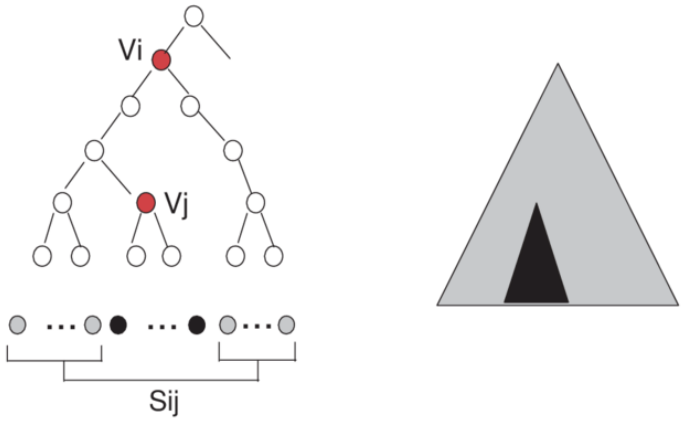
Αρχικοποίηση Αλγορίθμου

Αρχικά δημιουργούμε ένα τέλειο δυαδικό δέντρο τα φύλλα του δέντρου είναι οι δέκτες του συστήματός μας. Αφού το δέντρο είναι τέλειο δυαδικό αυτό σημαίνει ότι ο αριθμός των χρηστών (τα φύλλα του δέντρου) πρέπει να είναι ένας αριθμός που προκύπτει ως δύναμη του 2. Αν ο αριθμός των χρηστών μας δεν είναι ένας αριθμός που να είναι δύναμη του 2, τότε θα πρέπει να δημιουργήσουμε ψεύτικους δέκτες έτσι ώστε το σύνολό τους να είναι δύναμη του 2 και να δημιουργεί ένα τέλειο δυαδικό δέντρο. Αν N ο αριθμός των φύλλων (δεκτών). Όπως προκύπτει από τη θεωρία των δυαδικών δέντρων το πλήθος των κόμβων με τα φύλλα που θα έχει αυτό το δέντρο είναι $(2N - 1)$.

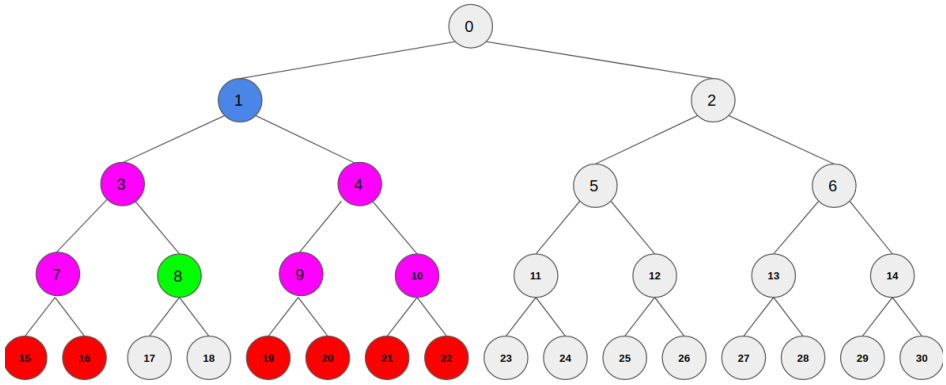


Σχήμα 4.1: Δυαδικό δέντρο με End Node τα φύλλα του δέντρου

Στη μέθοδο SD ορίζουμε υποσύνολα S των κόμβων u με τα οποία υποσύνολα θα ορίσουμε την ομάδα των δεκτών που θα λάβουν το μήνυμα. Ένα υποσύνολο στη μέθοδο SD ορίζεται ως $S_{i,j}$ χρησιμοποιώντας δύο κόμβους του δέντρου (u_i, u_j). Ο κόμβος u_i ορίζει ότι όλοι οι κόμβοι πού είναι παιδιά του θα μπορούν να λάβουν το μήνυμά. Ο κόμβος u_j ο οποίος είναι παιδί του κόμβου u_i είναι ο κόμβος που ο ίδιος αλλά και τα παιδιά του θα αποκλειστούν από τη λήψη του μηνύματος. Με αυτό τον τρόπο δημιουργούμε δύο υπό δέντρα το ένα περιλαμβάνει τους κόμβους που θα λάβουν το μήνυμα αλλά και ένα δεύτερο που είναι μέρος του πρώτου το οποίο περιλαμβάνει τους κόμβους που δεν θα λάβουν το μήνυμα και θα πρέπει να αποκλειστούν. Στην πορεία λοιπόν αυτό που πρέπει να κάνουμε είναι να δημιουργήσουμε τα υποσύνολα $S_{i1,j1}S_{i2,J2}$ έως $S_{im,jm}$ τα οποία θα περιέχουν όλους τους κόμβους που πρέπει να πάρουν το μήνυμα και θα αποκλείουν όλους τους κόμβους τους οποίους θα πρέπει να μην λάβουν το μήνυμα.



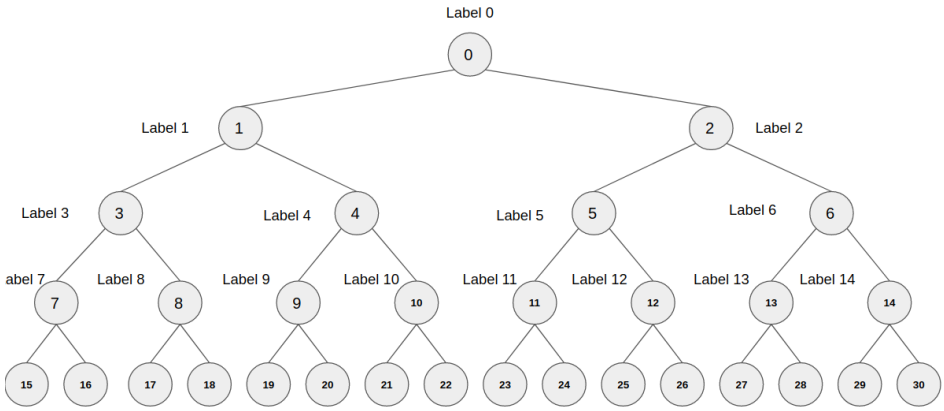
Σχήμα 4.2: Ορισμός ενός υποσυνόλου χρηστών που θα λάβουν το μήνυμα με την μέθοδο Subset Difference Method (SD) [4]



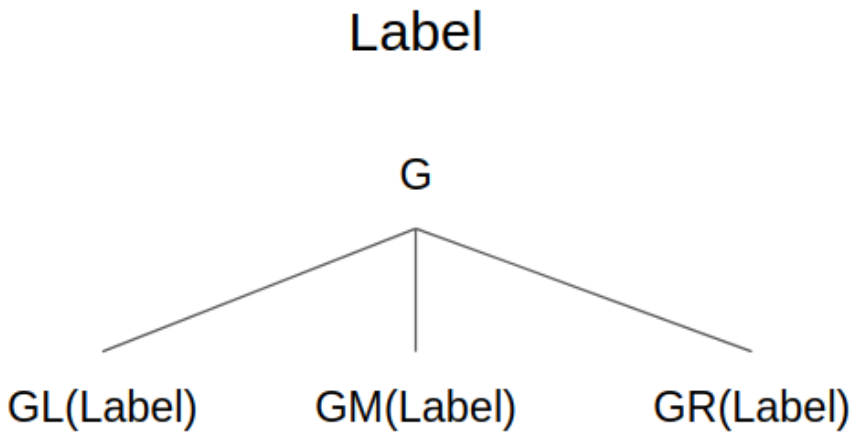
Σχήμα 4.3: Το υποδέντρο $S_{1,8}$ προκύπτει για το υποσύνολο $(u_{15}, u_{16}, u_{19} - u_{22})$

Τα υποσύνολα που μπορούν να προκύψουν από μία τέτοια διαδικασία μπορούν να είναι στη χειρότερη περίπτωση $2r - 1$ όπου r ο αριθμός των χρηστών που θα αποκλειστούν από τη μετάδοση. Ο τύπος αυτός δίνει την χειρότερη δυνατή περίπτωση και ο αριθμός που προκύπτει είναι αρκετά μεγάλος αλλά αν δούμε μία μέση η τιμή αυτή συνήθως υπολογίζεται στο $1,25r$. Αυτό είναι πολύ σημαντικό γιατί από αυτό θα προκύψουν και τα επιπλέον στοιχεία που θα σταλούν μαζί με το μήνυμα. Ένα ακόμη κρίσιμο σημείο του αλγόριθμου είναι το πλήθος των κλειδιών που πρέπει να αποθηκευτούν σε κάθε δεκτή. Αν κρατήσουμε ένα κλειδί για κάθε πιθανό υποσύνολο που ανήκει ο χρήστης το πλήθος των κλειδιών είναι πολύ μεγάλο, προτείνεται λοιπόν μια μέθοδος κατά την οποία ο συνολικός αριθμός που πρέπει να κρατήσει κάθε δέκτης είναι $O(logN^2)$.
Η ιδέα προέρχεται από την μέθοδο των Goldreich, Goldwasser and Micali [32]. Το πρώτο βήμα είναι για κάθε ενδιάμεσο κόμβο του συνολικού δέντρου να επιλέγουμε μια ετικέτα LABEL με τυχαίο τρόπο , από αυτές θα παραχθούν τα κλειδιά για όλα τα υποσύνολα της μορφής $S_{i,j}$. Ας υποθέσουμε ότι έχουμε μια G ψευδοτυχαία γεννήτρια ακολουθιών , αυτή θα είναι μια μονόδρομη συνάρτηση οποία τριπλασιάζει την έξοδο της σε σχέση με την είσοδο. Αυτό γίνεται χρησιμοποιώντας τρεις γεννήτριες ψευδοτυχαίων ακολουθιών $G_L(S)$ που μας δίνει το πρώτο αριστερό μέρος της εξόδου της G την $G_R(S)$ που μας δίνει το τρίτο δεξί μέρος της G και την $G_M(S)$ που μας δίνει του δεύτερο και

μεσαίο μέρος της G. Το G λοιπόν απαρτίζεται από τρεις συναρτήσεις και δημιουργεί τυχαία σειρά χαρακτήρων από το Label S που δεν επιτρέπει την αντιστροφή και την εύρεσή του αρχικού.



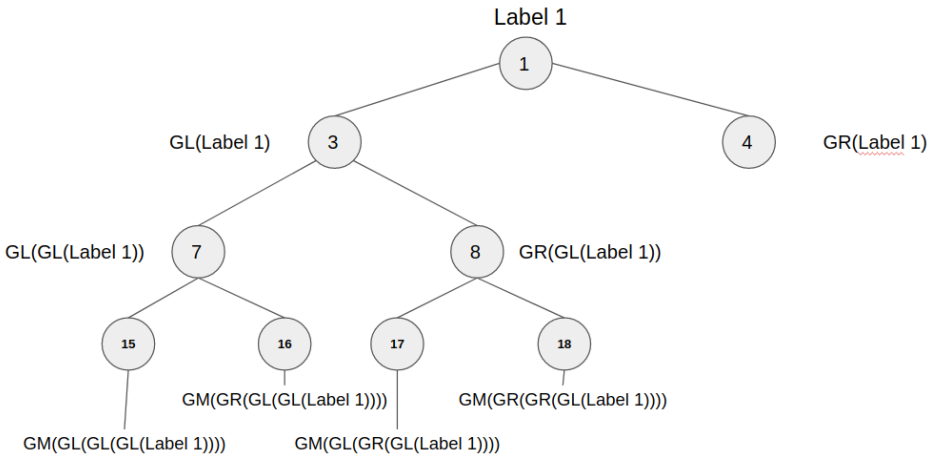
Σχήμα 4.4: Τυχαίες ετικέτες σε όλους τους κόμβουςLabel



Σχήμα 4.5: Συναρτήσεις G

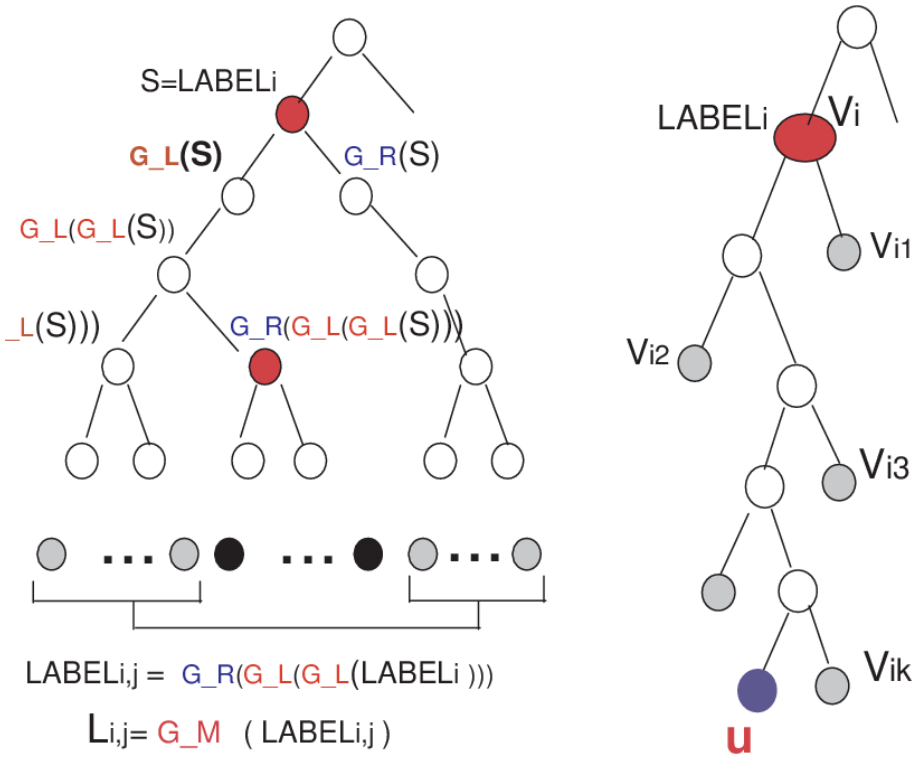
Αρχικά δίνουμε τυχαίες ετικέτες σε όλους τους κόμβους του δέντρου. Η διαδικασία του Labeling είναι μια top down διαδικασία για κάθε υποδέντρο T_i με ρίζα u_i : Η ρίζα διαθέτει μια ετικέτα S, δεδομένου ότι ένας γονέας έχει την ετικέτα από την αρχικοποίηση S, τα δύο παιδιά του θα αποκτήσουν την ψευδοετικέτα $G_L(S)$ το αριστερό παιδί και $G_R(S)$ το δεξί αντίστοιχα. Χρησιμοποιώντας τις συναρτήσεις G_L και G_R δημιουργούμε τις ψευδοτυχαίες ετικέτες μέχρι το κόμβοu_j. Ως $LABEL_{i,j}$ είναι η ψευδοετικέτα του κόμβου u_j που προέρχεται από το υποδεντρο $T_i(LABEL_i)$. Μετά από μια τέτοια διαδικασία Labeling , το κλειδί $L_{i,j}$ που ορίζεται για το υποσύνολο $S_{i,j}$ είναι το G_M του $LABEL_{i,j}$. Άρα για κάθε ετικέτα το G παράγει τρία μέρη: G_L - η ετικέτα για το αριστερό παιδί, G_R - η ετικέτα για το δεξί παιδί και το G_M το κλειδί στον κόμβο. Η διαδικασία δημιουργίας ετικετών και κλειδιών για κάθε υποδέντρο είναι :

- Στη διαδικασία ετικετοποίησης Labeling λοιπόν κάθε ενδιάμεσος κόμβος δημιουργεί ένα δικό του υποδέντρο , ξέροντας την ετικέτα του αρχικού κόμβου, είναι δυνατόν να υπολογιστούν οι ψευδοετικέτες (και τα κλειδιά) όλων των απογόνων του.
- Αν ξέρουμε την ψευδοετικέτα ενός κόμβου μπορούμε να βρούμε τη κάθε ψευδοετικέτα που προκύπτει από την διαδίκασια το Labeling για κάθε απόγονο κόμβο του υποδέντρου , άλλα δεν μπορούμε να βρούμε την ψευδοέτικετα των προγόνων ούτε την αρχική ετικέτα της ρίζας του υποδέντρου.
- Αν ξέρουμε όλες οι ψευδοετικέτες όλων των απογόνων από το κόμβο u_i (μη συμπεριλαμβανομένου του κόμβου j) , μπορούμε να υπολογίσουμε κλειδί $L_{i,j}$ θα είναι ψευδοτυχαίο από το $LABEL_{i,j}$.



Σχήμα 4.6: παράδειγμα ψευδοετικέτες για το κόμβο 1

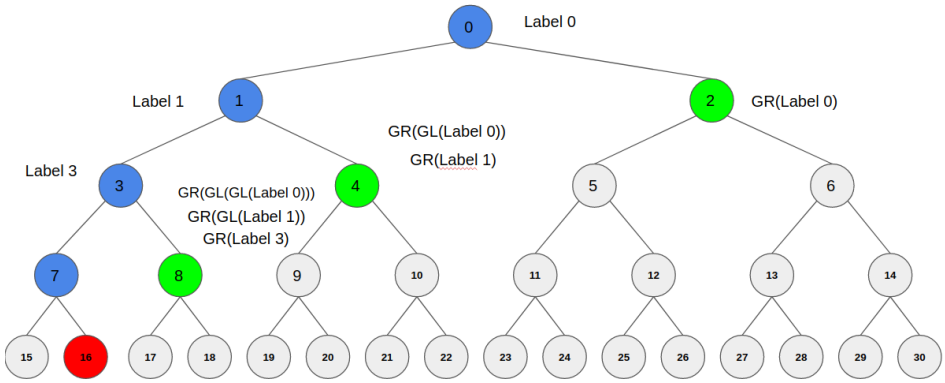
Είναι σημαντικό να σημειώσουμε ότι για την δεδομένη ετικέτα $LABEL_i$ ο υπολογισμός $L_{i,j}$ απαιτεί το πολύ $logN$ φορές χρήσης της συνάρτησης G. Η πληροφορία I_u είναι αυτή που κάθε δέκτης u αποθηκεύει για να παράγει το κλειδί. Για να βρεθούν το I_u πρέπει να βρεθεί κάθε υποδέντρο T_i έτσι ώστε το u να είναι ένα φύλλο του T_i, ο δέκτης u θα πρέπει να είναι σε θέση να υπολογίσει το L_{ij} όπου το i είναι προγονος του αλλά το j δεν πρέπει είναι πρόγονος του u. Αν υποθέσουμε την διαδρομή από u_i σε u τότε οι κόμβοι $u_{i1}, u_{i2}, .., u_{ik}$ που θα αποθηκεύσει ο δέκτης για το υποδέντρο T_i θα είναι οι κόμβοι δίπλα στη διαδρομή από το u_i στο u αλλά όχι πρόγονοι του u. Κάθε j στο T_i δεν είναι ένας πρόγονος του u , κάθε j είναι απόγονος ενός από τους κόμβους $u_{i1}, u_{i2}, , u_{ik}$. Επομένως, αν το u γνωρίζει τις ψευδοετικέτες των $u_{i1}, u_{i2}, , u_{ik}$ ως μέρος του I_u, στη συνέχεια χρησιμοποιώντας το G το πολύ N φορές μπορεί για να υπολογίσει το $L_{i,j}$ για κάθε j που δεν είναι πρόγονος του u και ανήκει στο T_i.



Σχήμα 4.7: Διαδικασία ετικετοποίησης [4]

Ο συνολικός αριθμός των ψευδοετικέτων που αποθηκεύονται στο δέκτη είναι για κάθε δέντρο T_i με βαθμό k το οποίο περιέχει το u δίνει $k-1$ ψευδοετικέτές. Συν ένα κλειδί στο τέλος σε περίπτωσή που δεν υπάρχουν δεκτές που θα αποκλειστούν . Ο μέγιστος αριθμός κλειδιών που θα αποθηκεύσει ο κάθε δέκτης είναι:

$$\frac{1}{2}log^2N + \frac{1}{2}logN + 1 \tag{4.1}$$

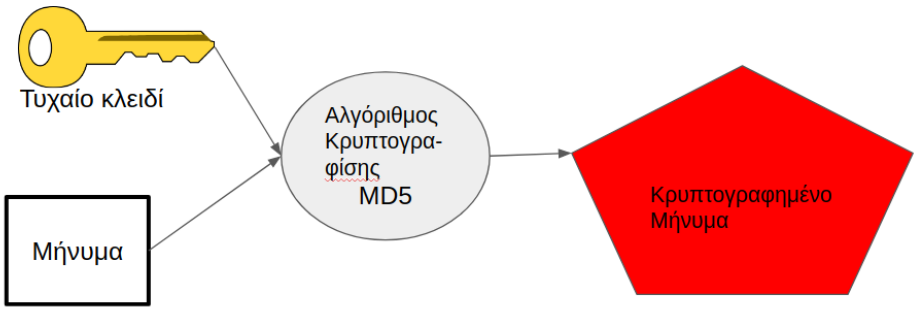


Σχήμα 4.8: παράδειγμα ψευδοετικέτες που θα αποθηκευτούν για το End Node 15

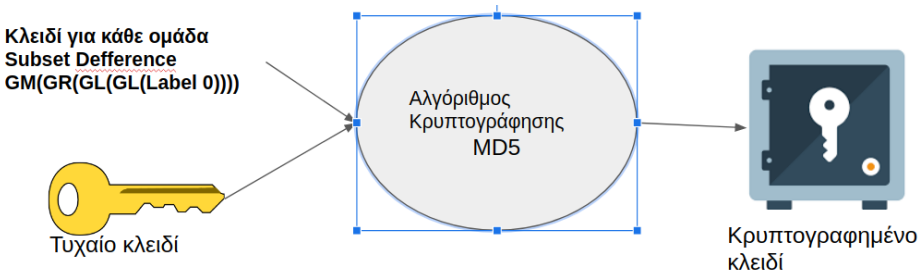
4.1.4 Κρυπτογράφηση

Η διαδικασία της κρυπτογράφησης έχει ως εξής :

- Διαλέγουμε ένα τυχαίο κλειδί Κ με το οποίο κρυπτογραφούμε το μήνυμά μας
- Στη συνέχεια για κάθε υποσύνολο $S_{i,j}$ υπολογίζουμε τις ψευδοετικέτες και το κλειδί του υποσυνόλου.
- Κρυπτογραφούμε το κλειδί Κ με το κλειδί του κάθε υποσύνολο .
- Φτιάχνουμε την κεφαλίδα του μηνύματος βάζοντας τους δείκτες ι , θ των υποσυνόλων S_{ij} με το κρυπτογραφημένο κλειδί Κ Μετά την κεφαλίδα ακολουθεί το κρυπτογραφημένο μήνυμα .



Σχήμα 4.9: Κρυπτογράφηση μηνύματος



Σχήμα 4.10: Κρυπτογράφηση μηνύματος

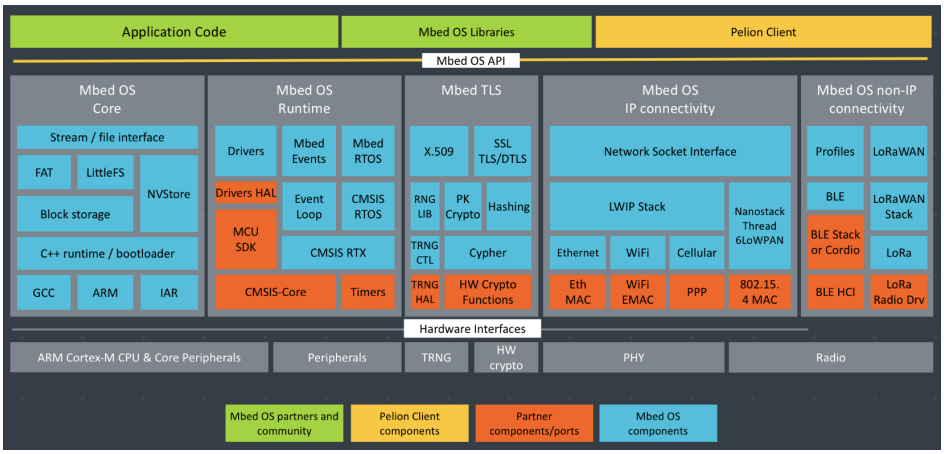
4.1.5 Αποκρυπτογράφηση

Κατά τη διαδικασία της αποκρυπτογράφησης ο δέκτης λαμβάνει την κεφαλίδα και το κύριο μήνυμα κρυπτογραφημένο.

- Από τους δείκτες της κεφαλίδας βλέπει αν ανήκει σε κάποιο από τα υποσύνολα.
- Αν ανήκει σε κάποιο από τα υποσύνολα υπολογίζει το κλειδί του υποσυνόλου.
- Αποκρυπτογραφεί το κρυπτογραφημένο κλειδί Κ που υπάρχει στη κεφαλίδα με το κλειδί του υποσυνολου.
- Στη συνέχεια αφού γνωρίζει το κεντρικό κλειδί Κ της κρυπτογράφησης αποκρυπτογραφεί με αυτό το κύριο μήνυμα.

4.2 Mbed OS

Το Mbed OS [33] είναι λειτουργικό σύστημα για μικροέλεγκτες και μικροεπεξεργαστές, το έχει φτιάξει και το υποστηρίζει η εταιρία Arm. Το Mbed OS είναι ένα ανοιχτό λογισμικό βασισμένο στο Real Time Operating System RTOS και προορισμένο για Iot συσκευές. Υποστηρίζει βιβλιοθήκες και δομές οι οποίες επιταχύνουν την δημιουργία και την ανάπτυξη Iot συσκευών, οι οποίες βασίζονται σε ARM επεξεργαστές. Με το Mbed OS μπορούμε να προγραμματίσουμε Iot συσκευές χρησιμοποιώντας την γλώσσα προγραμματισμού C++. Με τη βοήθεια διαφορών IDE τα οποία είναι είτε Online , είτε Offline μπορούμε να δημιουργήσουμε κώδικα για τον Arm C/C++ Compiler τα οποία τα παραμετροποιούμε και τρέχουν για ένα μεγάλο πλήθος Arm επεξεργαστών. Οι εφαρμογές που είναι γραμμένες σε Mbed OS μπορούν να ξαναχρησιμοποιηθούν από οποιαδήποτε συσκευή είναι συμβατή με το Mbed OS.



Σχήμα 4.11: Η βασική Αρχιτεκτονική του Mbed OS

Το Mbed OS [34] χρησιμοποιεί ένα στρώμα αφαίρεσης υλικού (HAL) για την υποστήριξη των πιο κοινών τμημάτων ενός μικρο ελεγκτή, όπως χρονοδιακόπτες , είσοδοι έξοδοι κλπ. Αυτό το στρώμα διευκολύνει την δημιουργία εφαρμογών με ένα κοινό σύνολο APIs. Οι συσκευές περιλαμβάνουν αυτόματα τις απαραίτητες βιβλιοθήκες και υποστήριξη οδηγών για τα τυπικά περιφερειακά MCU, όπως το I2C, το Serial και το SPI. Το Mbed OS βασίζετε σε έναν πυρήνα RTOS, οπότε υποστηρίζει πολυνηματική εκτέλεση σε πραγματικό χρόνο. Το RTOS παρέχει διάφορες δυνατότητες διαχείρισης νημάτων όπως Semaphores, Mutexes και άλλα. Το Mbed OS διαθέτει ένα έξυπνο μηχανισμό διαχείρισης νημάτων και ενέργειας αυτό το κάνει ιδανικό για κατασκευές που χαρακτηρίζονται από χαμηλή κατανάλωση ενέργειας. Ένα από τα κύρια χαρακτηριστικά των Iot είναι οι επικοινωνίες, το Mbed OS υποστηρίζει όλα τα πρότυπα επικοινωνιών όπως Bluetooth Low Energy, NFC, RFID, LoRa, 6LoWPAN-ND, Thread, Wi-SUN, Ethernet, Wi-Fi και κυψελωτές επικοινωνίες. Το Mbed OS προσφέρει έναν μεγάλο πυρήνα των υφιστάμενων τεχνολογιών συνδεσιμότητας. Παράλληλα, με τριμηνιαίες εκδόσεις που υποστηρίζουν νεα χαρακτηριστικά, κρατάει την κοινότητα του Mbed OS ενήμερη για τις τάσεις της βιομηχανίας ώστε να μπορούν να δουλέψουν σε νέες καινοτόμες λύσεις που παράγουν επιχειρηματική αξία. Το Mbed OS προσφέρει δύο ενσωματωμένα δομικά στοιχεία για την ασφάλεια

- Arm Mbed TLS

- Secure Partition Manager (SPM)

Το Mbed TLS εξασφαλίζει τα κανάλια επικοινωνίας μεταξύ συσκευής και πύλης ή διακομιστή τα οποία χρησιμοποιούν το SPM αλλά και απομονωμένους τομείς ασφαλείας για υπηρεσίες αξιόπιστων συστημάτων που δεν χρησιμοποιούν το SPM. Γενικά το Mbed OS προσφέρει όλα τα κορυφαία πρωτόκολλα βιομηχανικών προτύπων, στις ψηφιακές κρυπτογράφησεις και τις σουίτες κρυπτογράφησης σύμφωνα με τις συστάσεις της NIST και άλλων σχετικών οργανισμών.

4.3 React

Το React [35] είναι μια βιβλιοθήκη Javascript ανοικτού κώδικα για την δημιουργία διεπαφής χρήστη (User Interface). Έχει φτιαχτεί και υποστηρίζεται από την εταιρεία FaceBook και άλλες συνεργαζόμενες εταιρίες. Το React διευκολύνει την δημιουργία διαδραστικών περιβαλλόντων διεπαφής χρήστη και αναλαμβάνει να ενημερώνει τα στοιχεία του περιβάλλοντος όταν κάτι αλλάξει. Η λογική του React είναι ότι ένα περιβάλλον χρήσης αποτελείται από πολλά μικρότερα στοιχεία τα Component. Μας δίνει λοιπόν την δυνατότητα δημιουργίας Components τα οποία είναι τα δομικά στοιχεία του περιβάλλον χρήσης και τα οποία στην συνέχεια συνδυάζονται για την τελική διεπαφή χρήστη ή μπορούν να χρησιμοποιηθούν για την δημιουργία άλλων περιβαλλόντων χρήσης.

4.4 React Native

Το React Native [36] είναι μια πλατφόρμα η οποία χρησιμοποιεί τη βιβλιοθήκη του React για να δημιουργεί εφαρμογές για κινητές συσκευές. Το React Native όπως και το React είναι ένα εργο ανοιχτου λογισμικού της εταιρείας Facebook. Με το React Native μπορούμε να δημιουργήσουμε εφαρμογές για κινητές συσκευές με λειτουργικό σύστημα Android ή IOS. Τα δομικά στοιχεία των εφαρμογών όπως και οι ίδιες οι εφαρμογές μπορούν να χρησιμοποιηθούν για συσκευές και των δυο λειτουργικών συστημάτων χωρίς ιδιαίτερες αλλαγές. Οι εφαρμογές που μπορούμε να δημιουργήσουμε είναι Native δηλαδή εκτελούνται κατ ευθεία από το λειτουργικό χωρίς ενδιάμεσες καταστάσεις πράγμα που σημαίνει καλή ταχύτητα και απόκριση των εφαρμογών.

4.5 Nodejs

Το Nodejs [37] είναι ένα Javascript περιβάλλον δημιουργίας εφαρμογών και εκτέλεσής βασισμένο στη Chrome V8 Javascript Engine, είναι ανοιχτού κώδικα και τρέχει από την πλευρά του εξυπηρετητή. Το Nodejs υποστηρίζεται από το ίδρυμα Node.js το οποίο έχει συγχωνευτεί με το JS και τώρα είναι το OpenJS. Το Nodejs είναι ένα ασύγχρονο περιβάλλον εκτέλεσης εφαρμογών, είναι σχεδιασμένο να μπορεί να δημιουργεί κλιμακωτές δικτυακές εφαρμογές. Σε αντίθεση με το μοντέλο των νημάτων που είναι δυσνόητα το Nodejs χρησιμοποιεί απλές διεργασίες, έτσι δεν έχουμε να ανησυχούμε για ατέρμονες διεργασίες που κλειδώνουν ξεκλειδώνουν τους πόρους. Σχεδόν καμία λειτουργία δεν κάνει κατευθείαν είσοδο και έξοδο άρα δεν μπλοκάρουν το σύστημα

σε περίπτωσή σφάλματος. Το Nodejs δουλεύει ασύγχρονα χρησιμοποιώντας την λογική των συμβάντων. Η κοινότητα έχει δημιουργήσει ένα ολόκληρο οικοσύστημα από βιβλιοθήκες που προορίζονται ή είναι συμβατές με το Nodejs.

4.6 MQTT

Το MQTT [38] είναι ένα εξαιρετικά απλό και ελαφρύ πρωτόκολλο ανταλλαγής μηνυμάτων, σχεδιασμένο για συσκευές με περιορισμένες δυνατότητες και δίκτυα χαμηλού εύρους ζώνης, υψηλής καθυστέρησης ή αναξιόπιστων δικτύων. Οι αρχές σχεδιασμού είναι να ελαχιστοποιηθούν οι απαιτήσεις για το εύρος ζώνης δικτύου, ενώ ταυτόχρονα επιχειρείται η διασφάλιση της αξιοπιστίας και ο βαθμός διασφάλισης της παράδοσης των μηνυμάτων. Αυτές οι δυνατότητες κάνουν το πρωτόκολλο ιδανικό για M2M επικοινωνίες ή Iot ή για κινητές εφαρμογές όπου το εύρος ζώνης και η ισχύς της μπαταρίας είναι εξαιρετικά κρίσιμα . Το MQTT επινοήθηκε από τον Δρ Andy Stanford-Clark της IBM και από την Arlen Nipper της Arcom (τώρα Eurotech), το 1999. Το πρωτόκολλο εκτελείται μέσω TCP / IP ή μέσω άλλων πρωτοκόλλων δικτύου που παρέχουν χωρίς απώλειες αμφίδρομες συνδέσεις. Τα χαρακτηριστικά είναι:

- Χρήση του του συστήματος μηνυμάτων Publish/Subscribe.Αυτό παρέχει δυνατότητα αποστολής ένα μήνυμα σε πολλούς αποδέκτες και είναι ιδανικό για εφαρμογές που οι αποδέκτες δεν είναι συνέχεια συνδεδεμένοι.
- Το μήνυμα είναι ανεξάρτητο του περιεχομένου.
- Υποστηρίζει 3 τρόπους εξασφάλισης παράδωσεις των μηνυμάτων:
  - “At most once” Τα μηνύματα λαμβάνονται το πολύ μία φορά, αυτό μειώνει την κυκλοφορία του δικτύου. Μπορεί να παρουσιαστεί απώλεια μηνυμάτων. Αυτό το επίπεδο θα μπορούσε να χρησιμοποιηθεί, για παράδειγμα, με δεδομένα αισθητήρων περιβάλλοντος όπου δεν έχει σημασία αν χάνονται μεμονωμένες μετρήσεις καθώς το σύντομα θα υπάρξει μια νέα.
  - “At least once” Τα μηνύματα λαμβάνονται τουλάχιστον μία φορά, όπου τα μηνύματα είναι εξασφαλισμένα να φτάσουν, αλλά μπορούν να εμφανιστούν αντίγραφα.
  - “Exactly once” Τα μηνύματα λαμβάνονται ακριβώς μία φορά, τα μηνύματα είναι εξασφαλισμένα να φτάνουν ακριβώς μία φορά. Αυτό το επίπεδο θα μπορούσε να χρησιμοποιηθεί, για παράδειγμα, με συστήματα χρέωσης όπου διπλά ή χαμένα μηνύματα θα μπορούσαν να οδηγήσουν σε εσφαλμένες χρεώσεις
- Μικρού μεγέθους επικεφαλίδες για την ελαχιστοποίηση του μεγέθους των μηνυμάτων.
- Μηχανισμός για την ανίχνευση των συσκευών σε περίπτωση μη φυσιολογικής αποσύνδεσης.

4.7 InfluxDB

Το InfluxDB [39] είναι μια βάση δεδομένων υψηλής απόδοσης που είναι γραμμένη ειδικά για δεδομένα χρονοσειρών. Επιτρέπει την είσοδο υψηλού ρυθμού δεδομένων, τη συμπίεση και την αναζήτηση σε πραγματικό χρόνο. Το InfluxDB είναι γραμμένο εξ ολοκλήρου στο Go και μεταγλωττίζεται σε ένα ενιαίο εκτελέσιμο αρχείο χωρίς εξωτερικές εξαρτήσεις. Παρέχει δυνατότητες εγγραφής και αναζήτησης με μια διασύνδεση γραμμής εντολών, ενσωματωμένο API HTTP, ένα σύνολο βιβλιοθηκών πελάτη (π.χ. Go, Java JavaScript) και plugins για γνωστές εφαρμογές διαχείρισης δεδομένων όπως Telegraf, Graphite, Collectd και OpenTSDB. Το InfluxDB συνεργάζεται με την InfluxQL, μια γλώσσα αναζήτησης τύπου SQL για αλληλεπίδραση με τα δεδομένα. Έχει δημιουργηθεί με σκοπό να είναι εύκολο στη χρήση από αυτούς που είναι εξοικειωμένοι με άλλα περιβάλλοντα SQL, ενώ παράλληλα παρέχει χαρακτηριστικά ειδικά για την αποθήκευση και την ανάλυση δεδομένων μεγάλων χρονοσειρών. Το InfluxQL υποστηρίζει κανονικές εκφράσεις, αριθμητικές εκφράσεις και συναρτήσεις χρονικής σειράς για την επιτάχυνση της επεξεργασίας δεδομένων. Το InfluxDB μπορεί να χειριστεί εκατομμύρια σημεία δεδομένων ανά δευτερόλεπτο. Η εργασία με αυτά τα δεδομένα σε μεγάλο χρονικό διάστημα μπορεί να οδηγήσει σε προβλήματα αποθήκευσης. Το InfluxDB συμπιέζει αυτόματα τα δεδομένα για να ελαχιστοποιήσει τον αποθηκευτικό σας χώρο.

Κεφάλαιο 5

Αρχιτεκτονική

5.1 Αρχιτεκτονική

5.1.1 Γιατί επιλέξαμε το LoRaWAN

Όπως έχουμε αναφέρει σκοπός αυτής της εργασίας είναι να αναπτύξει ένα σύστημα εναλλακτικών επικοινωνιών σε περιπτώσεις κρίσεων . Με τον όρο κρίσης εννοούμε διαφορά φυσικά φαινόμενα όπως φωτιές πλημμύρες, τσουνάμι σεισμοί ή και κρίσης από ανθρώπινα αιτία πχ. τρομοκρατικές ενέργειες , ατυχήματα και λοιπά . Ήδη Υπάρχουν διάφορα συστήματα διαχείρισης κρίσεων και επικοινωνιών τα περισσότερα όμως από αυτά λειτουργούν χρησιμοποιώντας τα δίκτυα κινητής τηλεφωνίας. Τι γίνεται όμως στις περιπτώσεις που τα δίκτυα κινητής τηλεφωνίας έχουν καταστραφεί ή που ο φόρτος κλήσεων είναι τόσο πολύς που τα καθιστά εκτός λειτουργίας ή έστω δύσκολα προσπελάσιμα. Σε αυτές τις περιπτώσεις χρειαζόμαστε ένα εναλλακτικό δίκτυο κατά προτίμηση ασύρματο το οποίο θα μπορεί να μεταφέρει στοιχειώδης κάποιες πληροφορίες από τα κέντρα επιχειρήσεων προς τις περιφερειακές μονάδες ή και τους πολίτες. Προτιμάμε αυτά τα δίκτυα να είναι ασύρματα γιατί:

- Δεν χρειάζονται μεγάλες υποδομές μπορούν εύκολα και με μικρό κόστος να υλοποιηθούν.
- Οι σταθμοί μπορούν να μετακινηθούν, έτσι ώστε να βρίσκονται οπουδήποτε ανά πάσα στιγμή σε αντίθεση με τα ενσύρματα δίκτυα.
- Μπορούν να είναι κινούμενα.
- Μπορούν να είναι σε δυσπρόσιτα σημεία.

Αυτά κάνουν τα ασύρματα δίκτυα μια καλή επιλογή για να υπάρχουν σαν εφεδρείες στα ήδη υπάρχοντα συστήματα επικοινωνιών, ελπίζοντας ότι δεν θα χρειαστεί να χρησιμοποιηθούν ποτέ. Αυτό που προτείνεται λοιπόν είναι ένα ασύρματο δίκτυο εύκολο για να δημιουργηθεί και φτηνό που θα αναλάβει να μεταφέρει κάποιες βασικές πληροφορίες όταν όλα τα άλλα θα έχουν καταρρεύσει. Προσανατολιζόμαστε προς δίκτυα LPWA , γιατί αυτά μπορούν να δουλέψουν σε μεγάλες αποστάσεις και παρόλο που το σύστημά μας δεν θεωρεί ως κρίσιμο στοιχείο την κατανάλωση ενέργειας εντούτοις δεν θα θέλαμε να είναι και πολύ δαπανηρό από άποψη ενέργειας. Δεν μας ενδιαφέρει ένας σταθμός

να μπορεί να δουλέψει 5 χρόνια με την ίδια μπαταρία αλλά σίγουρα μας ενδιαφέρει ένας σταθμός να μπορεί να λάβει για μία ή δύο μέρες ή έστω για κάποιες ώρες μηνύματα χωρίς να χρειαστεί να τροφοδοτηθεί με ρεύμα. Από την παρουσίαση των διαφόρων LPWA δικτύων στο κεφάλαιο 1 καταλήξαμε ότι το πιο κατάλληλο για αυτή τη δουλειά είναι το δίκτυο LoRaWAN γιατί :

- Μπορεί καλύπτει μεγάλες αποστάσεις .
- Έχει μικρή κατανάλωσης ισχύς.
- Υποστηρίζει αμφίδρομη επικοινωνία σε πραγματικό χρόνο.
- Έχει καλή συμπεριφορά στο θόρυβο.
- Έχει καλή συμπεριφορά στο φαινόμενο Doppler αρά κάνει για κινούμενους σταθμούς.
- Έχει σχετικά φθηνό κόστος .
- Είναι ανοιχτό σαν τεχνολογία ως προς το μεγαλύτερο του μέρος και σε σχέση με άλλες τεχνολογίες.
- Υποστηρίζει BroadCast εκπομπες.
- Είναι ασφαλές.
- Είναι εύκολα προσαρμόσιμο.

Τα αρνητικά είναι :

- Ο ρυθμός μετάδοσης πληροφορίας και τα μηνύματα που μπορούμε να στείλουμε είναι μικρά , αλλά μην ξεχνάμε ότι αναφερόμαστε σε ένα εναλλακτικό σύστημα διαχείριση κρίσεων δηλαδή χρειαζόμαστε να μεταφέρουμε μόνο την ελάχιστη αναγκαία πληροφορία.
- Το Gateway επικοινωνεί με το Network Server και τον Application Server μέσω IP δικτύων. Αυτό λύνεται όμως:
  - Εχοντας το Gateway, το Network Server, τον Application Server στο ίδιο τοπικό δίκτυο.
  - Χρησιμοποιώντας δορυφορικές συνδέσεις μεταξύ των Gateway και του Network Server.
  - Επειδη τα Gateway θα είναι λίγα σε επιλεγμένα σημεία μπορούμε να τα τοποθετήσουμε όπου μας βολεύει και να τα διαχειριστούμε με ότι υποδομή και δίκτυο κορμού θέλουμε έτσι πχ μπορούμε να φροντίσουμε να έχουν οπτική επαφή και να υλοποιήσουμε μεταξύ τους μια άλλου είδους ζεύξη. Αυτή την πολυτέλεια δεν την έχουμε με τους σταθμούς και αυτό μας κάνει αναγκαία μια τεχνολογία όπως το LoRaWAN.

Το σύστημα που προτείνουμε χωρίζεται σε δύο μέρη το πρώτο μέρος θα παρέχει τη δυνατότητα ανταλλαγής μηνυμάτων μικρού μήκους μεγέθους από κέντρο επιχειρήσεων στους επιμέρους σταθμούς και αντίστροφα. Το δεύτερο μέρος θα περιλαμβάνει ένα πλήθος σταθμών οι οποίοι θα είναι απλοί δεκτές μηνυμάτων από το Κέντρο Επιχειρήσεων και θα ενημερώνουν τους σταθμούς για τις ενέργειες που πρέπει να κάνουν ή μέσω

γιγαντοοθόνων θα ενημερώνουν τους πολίτες ή θα μπορούν να ενεργοποιούν ή να απ ενεργοποιούν διάφορα συστήματα αυτοματισμών πχ γεννήτριες, φωτοσημάνσεις, αντλίες, κλπ. Καταρχήν για να μπορέσουν να υλοποιηθούν και τα δύο μέρη του συστήματος που προτείνουμε πρέπει να δημιουργηθεί ένα δίκτυο LoRaWAN αυτό σημαίνει ότι σε κομβικά σημεία της πόλης θα πρέπει να τοποθετηθούν τα Gateway τα οποία θα καλύπτουν την γύρο περιοχή . Τα Gateway πρέπει να είναι έτσι τοποθετημένα ώστε να μπορούν να έχουν όσο το δυνατόν καλύτερη οπτική επαφή με τους δέκτες και να είναι όσο πιο ψηλά γίνεται ούτως ώστε να μπορούν να καλύψουν μία μεγάλη περιοχή. Αν η λύση που θα υλοποιήσουμε έχει μόνο ένα Gateway τότε μπορεί ο Network Server και ο Application Server να συνδέονται με τον κεντρικό μέσω ενός τοπικού δικτύου. Αν τα Gateway που θα χρησιμοποιήσουμε είναι περισσότερα από ένα τότε αυτά πρέπει με κάποιο τρόπο να μπορούν είτε να συνδεθούν στο διαδίκτυο είτε να συνδέονται με ένα κλειστό Μητροπολιτικό δίκτυο με το Network Server. Το κλειστό Μητροπολιτικό δίκτυο θα μπορούσε να είναι ένα δίκτυο που θα χρησιμοποιεί υπάρχουσες υποδομές οπτικών ινών ή δορυφορικές ζεύξεις ή θα μπορούσε να είναι ένα σύστημα μικρο κυματικών ζεύξεων ή οτιδήποτε άλλο. Έπειτα πρέπει να εγκατασταθεί ένας Network Server για την συνολική διαχείριση των μηνυμάτων και τον έλεγχο του δικτύου LoRaWAN. Τέλος πρέπει να δημιουργηθεί και ο Application Server για να μπορούν να μπορούν να υλοποιηθούν οι δυο υπηρεσίες του συστήματος που προτείνουμε.

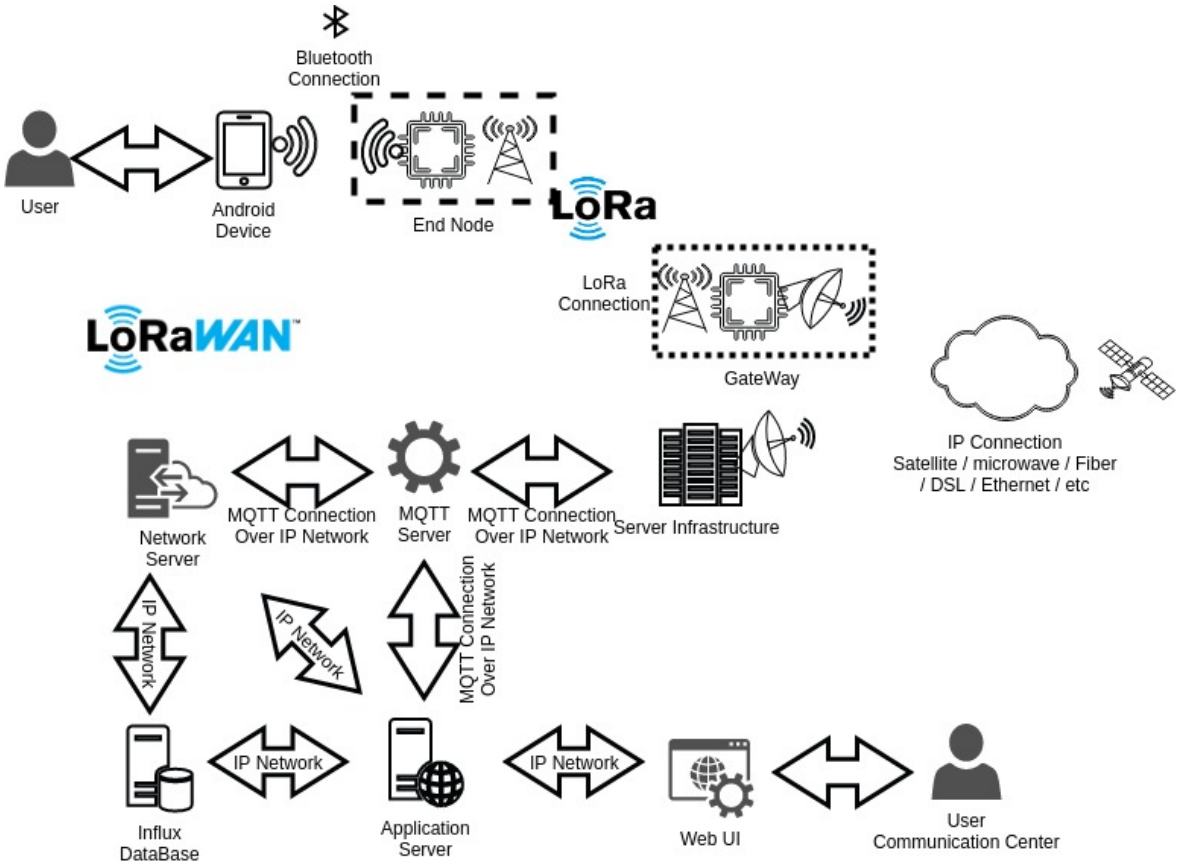
5.1.2 Συστημα ανταλλαγης Μηνυμάτων

Θεωρούμε ότι έχουμε δημιουργήσει LoRaWAN δίκτυο για να μπορούμε να κάνουμε την μετάδοση μηνυμάτων. Το σύστημα ανταλλαγής μηνυμάτων θα λειτουργεί ως εξής θα υπάρχει μία εφαρμογή η οποία θα δουλεύει σε Android συσκευές, αυτή η εφαρμογή θα μπορεί μέσω σύνδεσης Bluetooth να συνδέεται με ειδικές συσκευές που θα κατασκευάσουμε (EndNode). Οι End Node συσκευές παράλληλα με την Bluetooth σύνδεση, θα υλοποιούν και την σύνδεση με το δίκτυο LoRaWAN. Δηλαδή η συσκευή θα δουλεύει σαν γέφυρα ανάμεσα στην εφαρμογή του χρήστη και στο δίκτυο LoRaWAN. Όταν κάποιος θέλει να στείλει ένα μήνυμα θα πρέπει :

- Να ενεργοποιήσει την συσκευή End Node.
- Να χρησιμοποιήσει η συσκευή Android που έχει και να ενεργοποιήσει την ειδική εφαρμογή. Αφού αποκτήσει σύνδεση με την End Node συσκευή να ζητήσει από την End Node συσκευή να συνδεθεί στο δίκτυο LoRaWAN.
- Όταν ολοκληρωθεί η σύνδεση από το Chat περιβάλλον που του παρέχει η εφαρμογή να πληκτρολογεί το μήνυμα που θέλει να στείλει.
- Το μήνυμα θα περάσει από την Android συσκευή μέσω του Bluetooth στην End Node συσκευή και στη συνέχεια από αυτή θα το προωθήσει το δίκτυο LoRaWAN.
- Μέσω του δικτύου LoRaWAN το μήνυμα θα φτάσει στο Gateway.
- Ο Gateway μέσω του δικτύου που έχει υλοποιηθεί σαν κορμός θα στείλει το μήνυμα στο Network Server.
- Ο Network Server θα προωθήσει το μήνυμα στον Application Server.

Ο Application Server θα είναι μία εφαρμογή που θα μπορεί να λάβει αυτά τα μηνύματα

και να τα εμφανίσει στο χρήστη στο κέντρο επιχειρήσεων. Η ίδια διαδικασία θα μπορεί να γίνει και αντίθετα δηλαδή εφόσον ο χρήστης σταθμός έχει ήδη ενεργοποιημένη την End Node συσκευή και συνδεδεμένη Android εφαρμογή θα μπορεί ανά πάσα στιγμή να λάβει κάποιο μήνυμα από το Κέντρο Επιχειρήσεων που θα διαχειρίζεται τον Application Server. Μέσω του Application Server θα μπορεί να γράφει ένα μήνυμα αυτό θα προωθηθεί στο Network Server από κει στο Gateway μέσω του δικτύου LoRaWAN το Gateway θα μεταδώσει το μήνυμα σαν ένα Unicast μήνυμα το οποίο θα μπορεί να λάβει μόνο ο σταθμός για τον οποίο αναφέρεται , στη συνέχεια μέσω Bluetooth σύνδεσης που έχει ήδη υλοποιηθεί μεταξύ του End Node και της Android εφαρμογής το μήνυμα θα εμφανιστεί στην οθόνη του χρήστη. Εδώ να επισημάνουμε τα εξής ότι για να μπορεί η Android συσκευή να λάβει ένα μήνυμα οποιαδήποτε χρονική στιγμή θα πρέπει καταρχήν να έχει συνδεθεί μέσω Bluetooth με την End Node συσκευή. Η End Node συσκευή να έχει κάνει Join πάνω στο LoRaWAN δίκτυο. Δεύτερον η End Node συσκευή αφού κάνει Join να περάσει από κατάσταση Class A σε κατάσταση Class C. Έτσι Αφού θα είναι σε κατάσταση συνεχούς λήψης θα μπορεί να λάβει το μήνυμα που θα στείλει ο Application Server ανά πάσα στιγμή. Το να έχουμε τη συσκευή μας συνεχώς σε Class C σημαίνει ότι αυξάνουμε την κατανάλωση ενέργειας, όπως όμως ήδη έχουμε αναφέρει το θέμα της χαμηλής κατανάλωσης ενέργειας είναι επιθυμητό αλλά όχι κρίσιμο, μιας και η μπαταρία που μπορούμε να χρησιμοποιήσουμε για τη συσκευή θα μπορεί να είναι μεγάλης χωρητικότητας αλλά και ο χρόνος για τον οποίο θα χρειάζεται να δουλέψει η συσκευή δεν είναι μεγάλος. Το μέγεθος των μηνυμάτων που θα μπορεί να ανταλλάξει το σύστημα έχει να κάνει με το DataRate που θα χρησιμοποιηθεί. Σε προηγούμενο κεφάλαιο έχουμε αναφέρει ότι το LoRaWAN έχει Duty Cycle 1% για τις περισσότερες συχνότητες αλλά επειδή τα κανάλια που χρησιμοποιεί είναι 8 και σε κάθε εκπομπή αλλάζει κανάλι στο Upload θεωρούμε ότι δεν θα υπάρχει πρόβλημα. Στα μηνύματα Download που έρχονται από το Application Server αυτά επειδή οι συσκευές είναι σε Class C έρχονται στη συχνότητα 869.525MHz η όποια έχει Duty Cycle 10% έτσι θεωρούμε ότι δε θα υπάρχει πρόβλημα δεδομένου πάντα το σύστημα είναι για αποστολή και λήψη επείγον μηνυμάτων όχι ένα σύστημα διασκέδαση Chat.



Σχήμα 5.1: Διαδικασία ανταλλαγής μηνυμάτων

5.1.3 Σύστημα πολυεκπομπών από το Κέντρο Επιχειρήσεων

Το δεύτερο μέρος της πρότασής μας έχει να κάνει με ένα πλήθος συσκευών οι οποίες θα λειτουργούν σαν δέκτες μηνυμάτων από το κέντρο Επιχειρήσεων και θα μπορούν να ενημερώνουν τα μέλη των σταθμών και ομάδων επιχειρήσεων σε κάθε περιοχή ή θα μπορούν να ενημερώνουν μέσω γιγαντοοθόνης ή με οποιοδήποτε άλλο τρόπο τους πολίτες για το τι πρέπει να κάνουν. Επίσης το σύστημα πολυεκπομπών μπορεί ελαφρώς τροποποιημένο να διαχειριστεί υποδομές και συσκευές αυτοματοποιημένου έλεγχου. Δηλαδή παραδείγματος χάριν μπορεί να ενεργοποιήσεις ή να απενεργοποιήσεις αντλίες , συστήματα φωτισμού, παροχές ηλεκτρικού ρεύματος , ηλεκτρογεννήτριες σειρήνες,κλπ. Το πλήθος αυτών των συσκευών θα είναι αρκετά μεγαλύτερο από το πλήθος των συσκευών που έχουμε στην πρώτη κατηγορία, για να μπορούμε να ενημερώνουμε όσο το δυνατόν περισσότερες περιοχές ή ομάδες για το πώς πρέπει να αντιδράσουν σε κάθε δεδομένη στιγμή. Στο σύστημα που έχουμε υλοποιήσει εμείς μπορούμε να έχουμε μέχρι 128 τέτοιες συσκευές αλλά κάλλιστα μπορεί να τροποποιηθεί ώστε να υποστηρίξει πολύ περισσότερες.

Unicast

Αν υποθέσουμε ότι έχουμε 200 συσκευές οι οποίες λειτουργούν ως δέκτες μηνυμάτων και το Κέντρο Επιχειρήσεων θέλει να στείλει μηνύματα σε αυτές ή να στείλει μηνύματα

σε κάποιες από αυτές θα πρέπει να στείλει τόσα Unicast μηνύματα όσο είναι και το πλήθος των συσκευών που θέλει να ενημερώσει. Καταλαβαίνουμε ότι αυτό είναι ένα προβληματικό σημείο γιατί αν και οι συσκευές θα είναι σε Class C και τα μηνύματα θα σταλούν στη συχνότητα 869.525MHz η όποια έχει Duty Cycle 10% και πάλι ο χρόνος χρήσης του καναλιού είναι μεγάλος λόγο του πλήθους μηνυμάτων. Το Time On Air για ένα μήνυμα 110 Bytes σε SF9 σύμφωνα με τύπο (3.11) είναι $0,7792 secs$. Ο χρόνος αναμονής για κάθε αποστολή με DutyCycle 10% είναι :

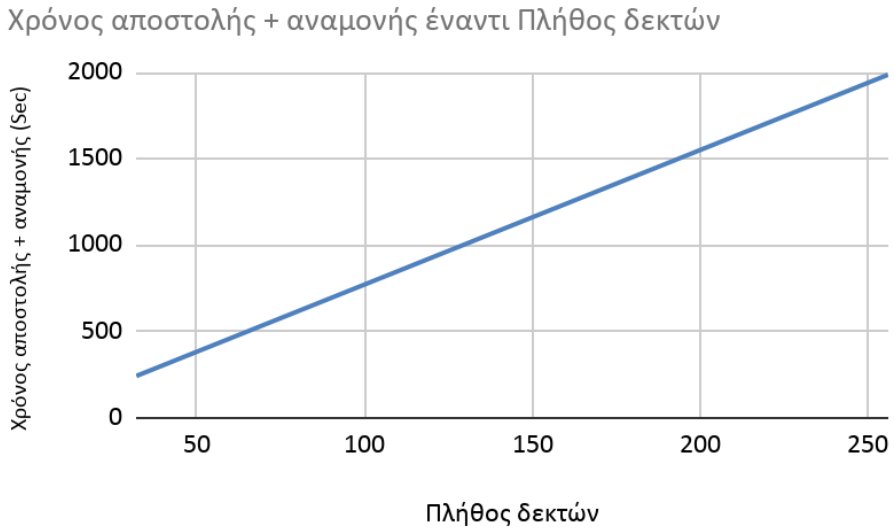
$$(ToA/DutyCylce) - ToA \tag{5.1}$$

Άρα 7.0133 *secs*
Ο χρόνος αποστολής ενός μηνύματος 110 Βψτες με ΣΦ9 σε Ν συσκευές είναι:

$$0.7792 * N + (N - 1) * 7.0133 \tag{5.2}$$

Πλήθος δεκτών	Χρόνος αποστολής + αναμονής (Sec)
32	242,351
64	491,715
128	990,444
256	1987,902

Πίνακας 5.1: Συνολικοί χρόνοι αποστολής μηνυμάτων μαζί με την καθυστέρηση του Duty Cycle



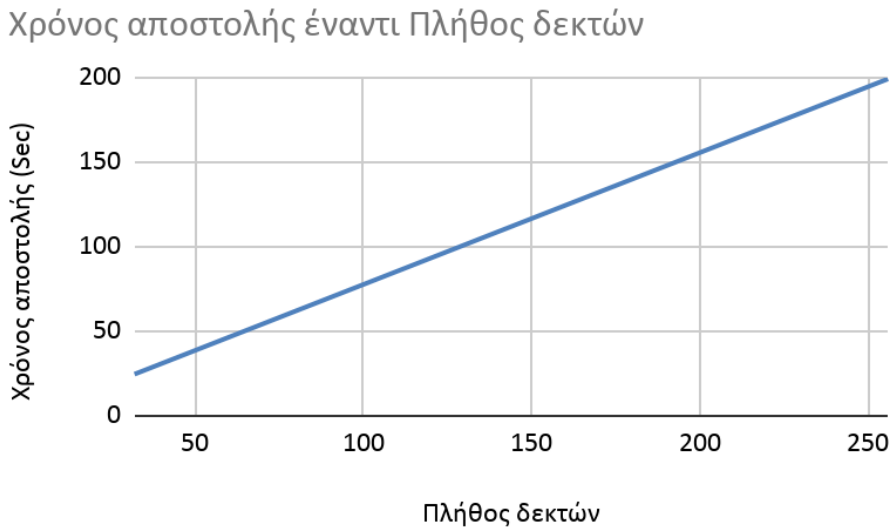
Σχήμα 5.2: Συνολικοί χρόνοι αποστολής μηνυμάτων μαζί με την καθυστέρηση του Duty Cycle

Ακόμα και αν τα στείλουμε όλα μηνύματα μαζί αφού καμία άλλη συσκευή δεν στέλνει σε αυτή την συχνότητα ο χρόνος είναι μεγάλος.

Πλήθος δεκτών	Χρόνος αποστολής (Sec)
32	24,936
64	49,872
128	99,745
256	199,491

Πίνακας 5.2: Συνολικοί χρόνοι αποστολής μηνυμάτων χωρίς την καθυστέρηση του Duty Cycle

Σε γραφική παράσταση θα είναι



Σχήμα 5.3: Συνολικοί χρόνοι αποστολής μηνυμάτων χωρίς την καθυστέρηση του Duty Cycle

Σε περίπτωση που το μήνυμα είναι μεγαλύτερο από 110 Bytes θα πρέπει να σταλούν 2 ή και περισσότερα μηνύματα σε κάθε συσκευή. Αν πάλι αποστάσεις μεταξύ των Gateway και των συσκευών είναι μεγάλες τότε πρέπει οι εκπομπές να γίνουν με SF12 όποτε οι χρόνοι γίνονται πολύ μεγαλύτεροι.

LoRaWAN Multicast

Σκεφτήκαμε λοιπόν ότι μία καλή λύση σε αυτό το πρόβλημα είναι να μη στείλουμε τα μηνύματα σαν Unicast μηνύματα αλλά σαν Multicast μηνύματα δηλαδή να στείλουμε κατά ομάδες. Σε πρώτη φάση αυτό που ήταν μία καλή σκέψη μπορούμε να χωρίσουμε τους σταθμούς μας σε ομάδες και να στέλνουμε το μήνυμα που θέλουμε σε μία ή περισσότερες ομάδες αυτό είναι εύκολο να το σκεφτούμε και να καταλάβουμε πως το πλήθος των μηνυμάτων θα μειωθεί δραματικά το ίδιο και ο χρόνος αποστολής και αναμονής. Υλοποιώντας τις οδηγίες της LoRaWAN Alliance για το Multicast μπορούμε στον Application Server να ρυθμίσουμε ομάδες έτσι ώστε η κάθε συσκευή να ανήκει σε μία ομάδα, κάθε συσκευή αφού κάνει Join να ενημερώνεται με Unicast εκπομπή με Multicast Commands σε ποια ομάδα ανήκει και ποια τα χαρακτηριστικά αυτής της

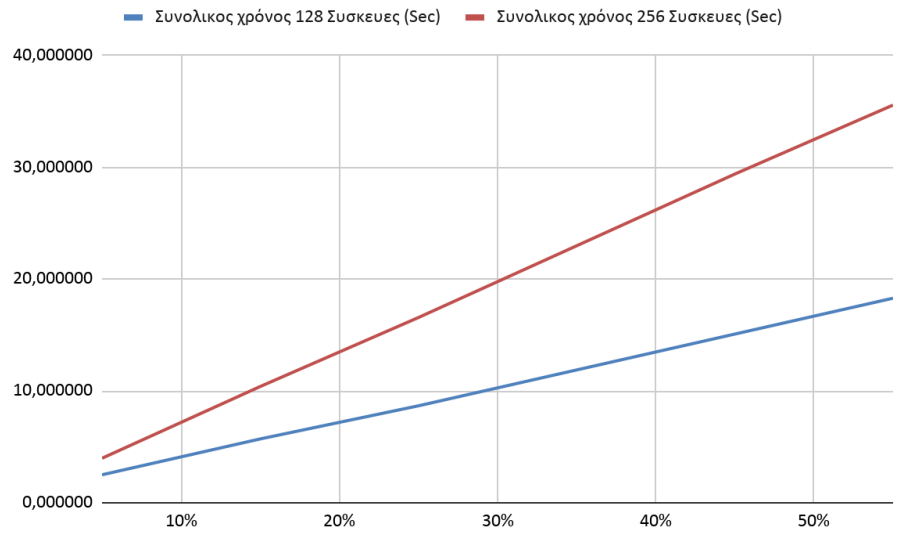
ομάδας, στη συνέχεια αφού κάθε συσκευή γνωρίζει την ομάδα στην οποία ανήκει μπορούμε από τον Application Server να στείλουμε μηνύματα σε κάθε ομάδα ξεχωριστά. Το πρόβλημα όμως στη δική μας περίπτωση είναι ότι οι ομάδες δεν είναι σταθερές, δεν έχουμε δηλαδή ένα πλήθος σταθμών χωρισμένο σε κάποιες ομάδες που μένουν σταθερές και δεν αλλάζουν, αλλά οι ομάδες μας αλλάζουν δυναμικά. Μην ξεχνάμε ότι εμείς αναφερόμαστε σε φυσικές καταστροφές και σε φαινόμενα που αλλάζουν συνεχώς άρα και οι ομάδες που θα πρέπει να πάρουν τα μηνύματα αλλάζουν κάθε φορά, άλλες ομάδες πρέπει να πάρουν το μήνυμα να κινηθούν Βόρεια, άλλες ομάδες πρέπει να πάρουν το μήνυμα να κινηθούν Ανατολικά ,άλλες ομάδες πρέπει να πάρουν το μήνυμα να μείνουν εκεί που βρίσκονται ή να προετοιμαστούν για μία κατάσταση και αυτό αλλάζει συνεχώς όπως εξελίσσονται τα διάφορα φαινόμενα που προκαλούν την κρίση. Αυτό σημαίνει ότι θα κάνουμε αρχικά έναν σχεδιασμό των ομάδων αλλά στη συνέχεια θα πρέπει πριν στείλουμε κάθε Multicast μήνυμα να στείλουμε μία σειρά Unicast μηνύματα που θα βγάζουν κάποιες συσκευές από την ομάδα που θέλουμε να ενημερώσουμε και παραλλήλως θα βάζουν κάποιες άλλες συσκευές σε αυτή την ομάδα. Με αυτή τη λογική ομως και πάλι αυξάνουμε αρκετά τα μηνύματα που θα πρέπει να στείλει ο Application Server προς τις συσκευές δεκτές. Ο χρόνος αποστολής ενός μηνύματος είναι το άθροισμα όλων των Unicast μηνυμάτων με τις Mmulticast Commands που θα στείλουμε συν ο χρόνος που κάνει για να σταλεί το Multicast μήνυμα. Ας δούμε μερικά παραδείγματα ας υποθέσουμε τέσσερα σενάρια με διαφορετικό πλήθος συσκευών που λαμβάνουν τα μηνύματα σε κάθε μία τέτοια υπόθεση βλέπουμε πιθανά ποσοστά σταθμών που πρέπει να τροποποιηθούν σε σχέση με το συνολικό πλήθος των σταθμών, πόσο χρόνο θα χρειαστεί έχοντας υποθέσει ότι το μήνυμα που θέλουμε να στείλουμε είναι στα 110 Bytes ενώ το μέσο Unicast μήνυμα με Multicast Commands θα είναι γύρω στα 20 Bytes με SF9.

Ποσοστό συσκευών	Πλήθος μηνυμάτων Unicast	Χρόνος αποστολής Unicast (Sec)	Χρόνος αποστολής Multicast (Sec)	Συνολικός χρόνος (Sec)	Χρόνος Αναμονής (Sec)
5%	7	1,727	0,779	2,506	22,568
15%	20	4,935	0,779	5,714	51,436
25%	32	7,897	0,779	8,676	78,087
35%	45	11,105	0,779	11,884	106,960
45%	58	14,313	0,779	15,092	135,834
55%	71	17,521	0,779	18,300	164,708

Πίνακας 5.3: Χρόνοι αποστολής μηνυμάτων για 1 Multicast μήνυμα 110Bytes σε δίκτυο 128 δεκτών με SF9 με την μέθοδο Multicast

Ποσοστό συσκευών	Πλήθος μηνυμάτων Unicast	Χρόνος αποστολής Unicast (Sec)	Χρόνος αποστολής Multicast (Sec)	Συνολικός χρόνος (Sec)	Χρόνος Αναμονής (Sec)
5%	13	3,208	0,779	3,987	35,887
15%	39	9,624	0,779	10,403	93,634
25%	64	15,794	0,779	16,573	149,160
35%	90	22,210	0,779	22,989	206,908
45%	116	28,626	0,779	29,406	264,655
55%	141	34,796	0,779	35,575	320,182

Πίνακας 5.4: Χρόνοι αποστολής μηνυμάτων για 1 Multicast μήνυμα 110Bytes σε δίκτυο 256 δεκτών με SF9 με την μέθοδο Multicast



Σχήμα 5.4: Χρόνοι αποστολής μηνυμάτων των πιθανών περιπτώσεων

Παρατηρούμε ότι οι χρόνοι είναι σαφώς καλύτεροι από ότι αν κάναμε Unicast εκπομπή αλλά πάλι σε μερικές περιπτώσεις όταν το ποσοστό μεγαλώνει και ειδικότερα όταν είναι μεγάλο και το πλήθος των σταθμών οι χρόνοι γίνονται αρκετά σημαντικοί. Έτσι λοιπόν σκεφτήκαμε μήπως μπορούσαμε να λύσουμε αυτό το πρόβλημα δανειζόμενοι στοιχεία από άλλες τεχνολογίες με Multicast εκπομπές και να τα προσαρμόσουμε πάνω στο δίκτυο LoRaWAN.

Multicast με Subset Difference

Μία εναλλακτική λύση είναι να χρησιμοποιήσουμε κάποια μέθοδος εκπομπής βασισμένη στο Broadcast Encryption. Τα προβλήματα που επιλύουν η μεθοδολογίες Broadcast Encryption έχουν πάρα πολλά κοινά σημεία με το πρόβλημα που εμείς αντιμετωπίζουμε δηλαδή έχουμε ένα πλήθος σταθμών οι οποίοι καλό είναι να μην ενημερώνονται συνέχεια από το Application Server για τις αλλαγές στις ομάδες τους γιατί σπαταλάμε πολύτιμο χρόνο εκπομπής αλλά ο Application Server να μπορεί ανά πάσα

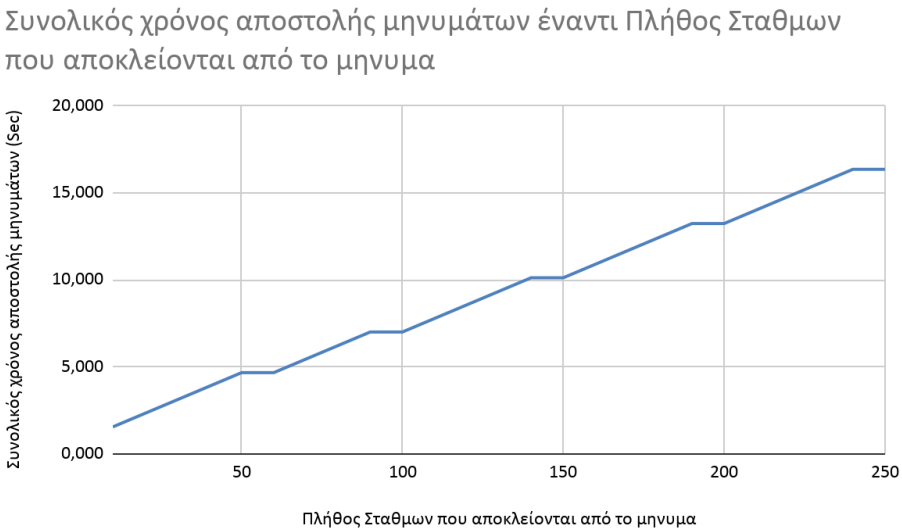
στιγμή να αναπροσαρμόζει τις ομάδες που στέλνει κάθε φορά το μήνυμα . Υπάρχουν διάφορες τεχνικές Broadcast Encryption , εμείς αποφασίσαμε να υλοποιήσουμε τη μέθοδο Subset Difference που είναι από τις πιο βασικές , πάνω σε αυτή στηρίζεται ένα πλήθος άλλων τεχνικών.

- Ειναι Stateless τεχνική.
  - Ο αριθμός των δεκτών είναι σταθερός δηλαδή δεν μπαίνουν νέοι δέκτες στο σύστημα και δεν φεύγουν δέκτες από το σύστημα.
  - Οι δέκτες δεν έχουν αμφίδρομη επικοινωνία με τον κεντρικό σταθμό ή μεταξύ τους ανταλλάσσοντας πληροφορίες ή καταγράφοντας πληροφορίες από προηγούμενες καταστάσεις.
- Έχει μικρή υπολογιστική κατανάλωση στην αποκρυπτογράφηση .
- Μέτριο μέγεθος πληροφοριών που πρέπει να αποθηκευτεί σε κάθε σταθμό.
- Σχετικά μικρό μέγεθος επικεφαλίδων σε σχέση με τις άλλες συμμετρικές μεθόδους, κρυπτογράφησης που χρησιμοποιούνται στο Broadcast Encryption.
- Το πλήθος των μηνυμάτων που θα εκπέμψει ο Application Server έχει να κάνει με το μήκος και της κεφαλίδας όπως αυτή θα διαμορφωθεί από την μέθοδο Subset Difference.

Το μήνυμα που θα εκπέμπουμε θα είναι η κεφαλίδα που θα ορίζει τα υποσύνολα των ομάδων που θα λάβουν το μήνυμα και αυτών που θα αποκλειστούν και βέβαια το μήνυμα που θέλουμε να στείλουμε. Το μέγεθος της κεφαλίδας στη χειρότερη περίπτωση μπορεί να είναι $2r-1$ όπου r το πλήθος των δεκτών που θα αποκλειστεί. Αυτή είναι η χειρότερη περίπτωση η πιο συνηθισμένη περίπτωση είναι το πλήθος των επικεφαλίδων να είναι $1,25*r$. Η επικεφαλίδα που εμείς υλοποιούμε χρησιμοποιεί ένα Byte για να προσδιορίσει τον αρχικό κόμβο u_i του συνόλου S_{ij} και ένα Byte για να προσδιορίσει τον τελικό κόμβο u_j, περιέχει επίσης και το κρυπτογραφημένο κλειδί που έχει μήκος 5 Bytes. Με βάση τα παραπάνω για κάθε στοιχείο της επικεφαλίδας χρειαζόμαστε 7 Bytes ,αρα το μέσο μέγεθος της επικεφαλίδας είναι $1,25r*7Bytes$. Είναι πολύ πιθανό η επικεφαλίδα σε συνδυασμό με το μήνυμα που θέλουμε να στείλουμε να είναι μεγαλύτερα από το μέγιστο αριθμό Bytes που υποστηρίζει LoRaWAN σαν μέγιστο αριθμό χαρακτήρων στο SF που εκπέμπουμε, για το λόγο αυτό δημιουργήσαμε μία διαδικασία που σπάει το μήνυμα μας σε μικρότερα ανάλογα με το SF που χρησιμοποιούμε και στη συνέχεια δίνει την δυνατότητα στο δέκτη να επανασυνδεθεί το μήνυμα. Για αυτή τη διαδικασία θα μιλήσουμε αργότερα. Το μέγεθος του μηνύματος εξαρτάται από το ίδιο μήνυμα και από το πλήθος των σταθμών που θα αποκλειστούν από αυτό το μήνυμα. Ας δούμε μερικά παραδείγματα σε ένα πίνακα όπου υποθέτουμε ότι το μέγεθος του μηνύματος είναι 110 Bytes η εκπομπή γίνεται σε SF9 και ότι το πλήθος των δεκτών είναι 200. Πιο κάτω βλέπουμε πώς θα διαμορφωθεί το μήνυμα μας ανάλογα με το πλήθος των δεκτών που θα αποκλειστούν από το μήνυμα.

Πλήθος Σταθμών που απο-κλείονται	Αριθμός Κεφαλίδων	Bytes Κεφαλίδων	Μέγεθος μηνύματος	Μηνύματα που θα διασπαστεί το αρχικό	Συνολικός χρόνος αποστολής μηνυμάτων (Sec)
10	13	91	110	2	1,558
20	25	175	110	3	2,337
30	38	266	110	4	3,117
40	50	350	110	5	3,896
50	63	441	110	6	4,675
60	75	525	110	6	4,675
70	88	616	110	7	5,454
80	100	700	110	8	6,234
90	113	791	110	9	7,013
100	125	875	110	9	7,013
110	138	966	110	10	7,792
120	150	1050	110	11	8,571
130	163	1141	110	12	9,351
140	175	1225	110	13	10,130
150	188	1316	110	13	10,130
160	200	1400	110	14	10,909
170	213	1491	110	15	11,688
180	225	1575	110	16	12,468
190	238	1666	110	17	13,247
200	250	1750	110	17	13,247
210	263	1841	110	18	14,027
220	275	1925	110	19	14,806
230	288	2016	110	20	15,585
240	300	2100	110	21	16,365
250	313	2191	110	21	16,365

Πίνακας 5.5: Χρόνοι αποστολής μηνυμάτων για 1 Multicast μήνυμα 110Bytes σε δίκτυο 256 δεκτών με SF9 με την μέθοδο Broadcast Encryption Subset Difference



Σχήμα 5.5: Χρόνοι αποστολής μηνυμάτων Broadcast Encryption Subset Defference

Αυτό που παρατηρούμε είναι ότι αν το πλήθος των ομάδων και οι δεκτές της κάθε ομάδας είναι λίγοι και ισομερώς κατανεμημένοι για την Multicast τεχνική το ποσοστό των δεκτών που θα πρέπει να τροποποιήσουν την ομάδας τους θα είναι μικρό, αντίθετα με την Broadcast Encryption τεχνική αφού οι ομάδες είναι μικρές το πλήθος των δεκτών που πρέπει να αποκλείσουμε θα είναι μεγάλο. Άρα για μικρές ισοκατανεμημένες ομάδες η τεχνική του Multicast δείχνει πιο αποδοτική. Το αντίθετο συμβαίνει αν το πλήθος των ομάδων είναι μικρό, ο αριθμός των δεκτών μεγάλος και ισοκατανεμημενος, σε αυτή την περίπτωση η μέθοδος του Broadcast encryption είναι αποδοτικότερη. Βέβαια και στις δύο περιπτώσεις έχει μεγάλη σημασία σε τι βαθμό τροποποιούνται οι ομάδες, πως είναι κατανεμημένοι οι δέκτες σε αυτές τις ομάδες και πως το πλήθος των ομάδων αλλάζει στη εξέλιξη των φαινομένων.

Παράδειγμα 1 Αν εκπέμπουμε σε SF9 ο χρόνος αποστολής μηνύματος Multicast 110 Bytes Payload είναι 0,7792 secs. Ο χρόνος αποστολής Unicast μηνύματος 20 Bytes είναι 0,2467 secs. Αν έχουμε 128 Δέκτες χωρισμένους σε 8 ομάδες περίπου κάθε στιγμή και κάθε ομάδα έχει περίπου τον ίδιο αριθμό δεκτών δηλαδή 16 κάθε στιγμή. Τότε με την μέθοδο του Multicast.

Ποσοστό αλλαγής	Πλήθος Δεκτών που αλλάζουν στην ομάδα σε σχέση με το Πλήθος δεκτών	Χρόνος αποστολής Unicast (Sec)	Συνολικός χρόνος Unicats + Multicast(Sec)
5%	7	1,727	2,507
10%	13	3,208	3,987
15%	20	4,936	5,715
20%	26	6,416	7,196
25%	32	7,897	8,676
30%	39	9,625	10,404
35%	45	11,105	11,885
40%	52	12,833	13,612

Αφού κάθε ομάδα έχει 16 δέκτες για να ορίσουμε μια ομάδα πρέπει να αποκλείσουμε 112 δέκτες με την μέθοδο του Broadcast Encryption Αρα.

Πλήθος Σταθμών που αποκλείονται από το μήνυμα	Αριθμός Κεφαλίδων	Bytes Κεφαλίδων	Πλήθος μηνυμάτων που θα διασπαστεί το αρχικό	Συνολικός χρόνος αποστολής μηνυμάτων (Sec)
112	140	980	10	7,793

Παρατηρούμε ότι όσο το ποσοστό αλλαγής είναι ανάμεσα στο 20% με 25% τοτε η Multicast μέθοδος είναι αποδοτικότερη. Ακόμα όμως και στις μικρές ομάδες αν το ποσοστό αλλαγής μεγαλώσει τότε η μέθοδος του Broadcast Encryption γίνεται αρκετά αποδοτικότερη.

Παράδειγμα 2 Αν έχουμε 128 δέκτες χωρισμένους σε 2 ομάδες περίπου κάθε στιγμή και κάθε ομάδα έχει περίπου τον ίδιο αριθμό δεκτών δηλαδή 64 κάθε στιγμή. Τότε με την μέθοδο του Multicast.

Ποσοστό αλλαγής	Πλήθος Δεκτών που αλλάζουν στην ομάδα σε σχέση με το Πλήθος δεκτών	Χρόνος αποστολής Unicast (Sec)	Συνολικός χρόνος Unicats + Multicast(Sec)
5%	7	1,727	2,507
10%	13	3,208	3,987
15%	20	4,936	5,715
20%	26	6,416	7,196
25%	32	7,897	8,676
30%	39	9,625	10,404
35%	45	11,105	11,885
40%	52	12,833	13,612

Αφού κάθε ομάδα έχει 64 δέκτες για να ορίσουμε μια ομάδα πρεπει να αποκλείσουμε 64 δέκτες με την μέθοδο του Broadcast Encryption Αρα.

Πλήθος Σταθμών που αποκλείονται από το μήνυμα	Αριθμός Κεφαλίδων	Bytes Κεφαλίδων	Πλήθος μηνυμάτων που θα διασπαστεί το αρχικό	Συνολικός χρόνος αποστολής μηνυμάτων (Sec)
64	80	560	7	5,455

Παρατηρούμε ότι όσο το ποσοστό αλλαγής είναι ανάμεσα 10% με 15% περίπου τότε η Multicast μέθοδος είναι ελαφρός αποδοτικότερη, όταν όμως τα ποσοστά μεγαλώνουν τότε οι διαφορές είναι ύπερ της μεθόδου του Broadcast Encryption.

Παράδειγμα 3 Αν υποθέσουμε ότι στέλνουμε μήνυμα στους 100 από τους 128 δέκτες τότε θα αποκλείσουμε 28 μόνο δέκτες Αρα.

Πλήθος Σταθμών που αποκλείονται από το μήνυμα	Αριθμός Κεφαλίδων	Bytes Κεφαλίδων	Πλήθος μηνυμάτων που θα διασπαστεί το αρχικό	Συνολικός χρόνος αποστολής μηνυμάτων (Sec)
28	35	245	4	3,117

Τώρα αν αναλογιστούμε ότι εμείς σε πραγματικές συνθήκες δεν μπορούμε να χωρίζουμε τους δέκτες σε ομάδες συνεχώς για να στείλουμε ένα μήνυμα αλλά θέλουμε να επιλέγουμε χ δεκτές κάθε φορά αναλόγως τις ανάγκες μας , οι οποίοι θα λάβουν το μήνυμα μας τότε ο μέθοδος του Broadcast Encryption Θεωρούμε ότι είναι η πλέον κατάλληλη.

5.1.4 Διάσπαση μηνύματος (Framing)

Όπως είπαμε παραπάνω επειδή κάθε μήνυμα που θέλουμε να στείλουμε όταν προσαρτώνται κεφαλίδες μεγαλώνει αρκετά και μπορεί να ξεπερνά το όριο του μηνύματος που υποστηρίζει το δίκτυο LoRaWAN συνάρτηση του SF που χρησιμοποιούμε, για αυτό το λόγο είμαστε αναγκασμένοι ένα μήνυμα που θέλουμε να στείλουμε, να το σπάσουμε σε περισσότερα από ένα LoRaWAN μηνύματα Frames). Αυτά τα μηνύματα πρέπει να μπορούν όταν γίνει η λήψη τους από το δέκτη να ξανά συναρμολογηθούν στο αρχικό μήνυμα ούτως ώστε να γίνει η αποκωδικοποίηση του στην συνέχεια. Η LoRaWAN Alliance μέσα από το έργο FOTA έχει δημιουργήσει μια συγκεκριμένη οδηγία για το κατακερματισμό [40] πληροφοριών σε ένα LoRaWAN δίκτυο. Η οδηγία όμως είναι πολύ συνθέτη, χρονοβόρα και μεγαλώνει τα μηνύματα έτσι δεν αξιολογείται ως καλή επιλογή για να την χρησιμοποιήσουμε στη δική μας περίπτωση. Έτσι δημιουργήσαμε μια δική μας διαδικασία για τον κατακερματισμό των μηνυμάτων. Κάθε μήνυμα που στέλνει ο Application Server ξεκινάει με τις ετικέτες της Subset Difference μεθόδου και στην συνέχεια ακολουθεί το μήνυμα που θέλουμε να στείλουμε .

1 Byte	1 Byte	5 Bytes	1 Byte	1 Byte	5 Bytes		N Bytes
Αρχικός κόμβος πρώτου Subset	Τελικός κόμβος πρώτου Subset	Κρυπτογραφημένο κλειδί πρώτου Subset	Αρχικός κόμβος δεύτερου Subset	Τελικός κόμβος δεύτερου Subset	Κρυπτογραφημένο κλειδι δεύτερου Subset	.....	Κύριο μήνυμα

Σχήμα 5.6: Μήνυμα που πρέπει να σταλεί

Σε κάθε μήνυμα που στέλνουμε βάζουμε μπροστά 2 Byte. Στο πρώτο Byte χρησιμοποιούμε τα 2 πρώτα Bit από αριστερά για να ορίσουμε αν το μήνυμα είναι:

- 00 Μοναδικό και αυτοτελές.
- 01 Το Πρώτο από μια σειρά μηνυμάτων
- 11 Αν είναι ένα από τα επόμενα μηνύματα.

Τα υπόλοιπα 6 Bit τα χρησιμοποιούμε για αναφέρουμε τον αύξων αριθμό του τμήματος που στείλαμε σε σχέσει με όλο το μήνυμα. Το Δεύτερο Byte χρησιμοποιείται μόνο στο πρώτο μήνυμα και ενημερώνει το δέκτη πόσα Subset θα περιέχει ολόκληρο το μήνυμα στο τέλος.

2 Bit	6 Bit	1 Byte
00 Μοναδικό και αυτοτελές. 01 Το Πρώτο από μια σειρά μηνυμάτων. 11 Ένα από τα επόμενα μηνύματα.	Αύξων αριθμός Frame	(Μόνο στο πρώτο Frame) Πλήθος Subsets του πλήρους μηνύματος

Σχήμα 5.7: Επικεφαλίδα για το Framing

Για κάθε μήνυμα που στέλνει ο Application Server ελέγχει αν το συνολικό μήκος του μηνύματος συν την κεφαλίδα κατακερματισμού που θα προσθέσει είναι μικρότερο ή ίσο από το μεγαλύτερο μήκος μηνύματος που επιτρέπει το SF της εκπομπής. Τότε βάζει μπροστά από το μήνυμα 2 Bytes το πρώτο έχει την τιμή 000000000 που δηλώνει ότι είναι ένα αυτοτελές μήνυμα και το δεύτερο Byte ορίζει το πλήθος των Subset που ακολουθούν.

1 Byte	1 Byte	M Byte
00 Μοναδικό και αυτοτελές. 000000 για να γεμίσει το Byte.	Πλήθος Subsets που ακολουθούν στο μήνυμα.	Μήνυμα που πρέπει να σταλεί.

Σχήμα 5.8: Επικεφαλίδα αυτοτελή μηνύματος Frame

Αν το μήνυμα που πρέπει να σταλεί συν την κεφαλίδα κατακερματισμού που θα προσθέσει είναι είναι μεγαλύτερο από το μεγαλύτερο μήκος μηνύματος που επιτρέπει το SF της εκπομπής. Τότε το μήνυμα σπάει σε περισσότερα τμήματα (Frames) το πρώτο τμήμα θα είναι

1 Byte	1 Byte	(Μέγιστο πλήθος Bytes που υποστηρίζει το SF που εκπέμπουμε -2) Bytes
01 Το Πρώτο από μια σειρά μηνυμάτων 0000001-111111 είναι το πλήθος των Frame που θα ακολουθήσουν.	Πλήθος Subsets που ακολουθούν στο μήνυμα.	μέρος μηνύματος που πρέπει να σταλεί.

Σχήμα 5.9: Επικεφαλίδα πρώτου μη αυτοτελή μηνύματος

Τα επόμενα Frames θα είναι

1 Byte	(Μέγιστο πλήθος Bytes που υποστηρίζει το SF που εκπέμπουμε-1) Bytes Στο τελευταίο θα είναι μικρότερο από το (Μέγιστο πλήθος Bytes που υποστηρίζει το SF που εκπέμπουμε-1)
11 Ένα από τα επόμενα μηνύματα. 000001 - 111111 Αύξων αριθμός του Frame.	μέρος μηνύματος που πρέπει να σταλεί.

Σχήμα 5.10: Επικεφαλίδα επόμενων μη αυτοτελή μηνύματος

Ο δέκτης τώρα με τη σειρά του όταν το μήνυμα είναι αυτοτελές παίρνει το κύριο μήνυμα και εφαρμόζει την τεχνική του Ssubset Difference . Αν το μήνυμα δεν είναι αυτοτελές τότε στο πρώτο Frame μαθαίνει των αριθμό των Frame που περιμένει και το πλήθος των Subsets , αποθηκεύει μέρος του κύριου μηνύματος. Κάθε φορά που λαμβάνει ένα επόμενο Frame ελέγχει ότι ο αύξων αριθμός είναι ένα παραπάνω από το προηγούμενο έτσι ξέρει ότι δεν έχει χάσει κανένα μήνυμα και αποθηκεύει το κύριο μήνυμα ώστε να το συνθέσει ολόκληρο στο τέλος και συνεχίζει τη διαδικασία μέχρι να λάβει και το τελευταίο Frame. Τέλος αν όλα πήγαν καλά έχει ολόκληρο το κύριο μήνυμα. Ξέρει πόσα Subsets υπάρχουν άρα μπορεί να ξεχωρίσει τα Subsets από το μήνυμα του Application Server και να ολοκληρώσει την διαδικασία.

Κεφάλαιο 6

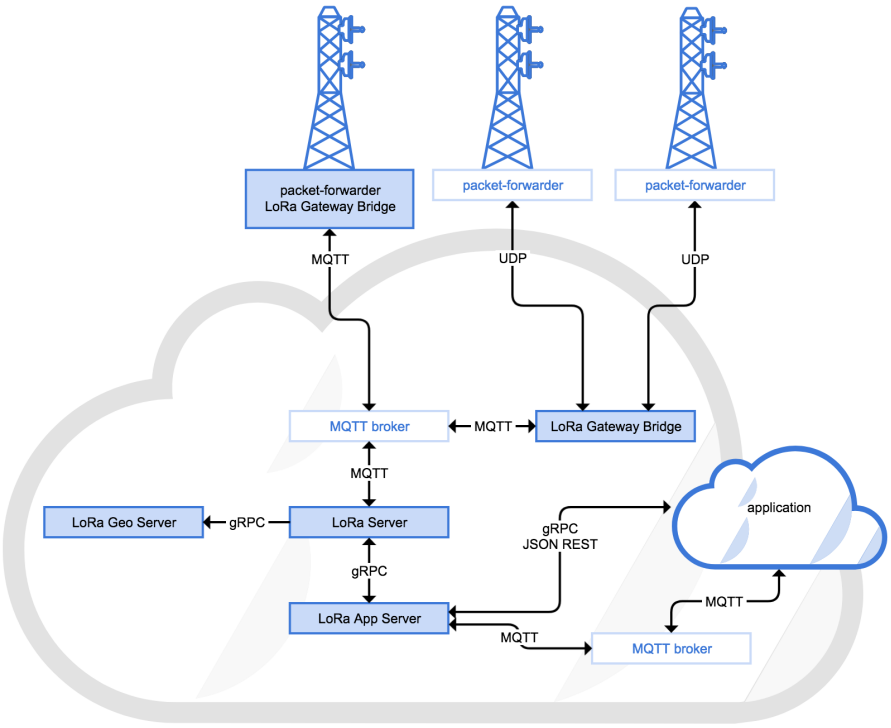
Υλοποίηση

6.1 Υλοποίηση

6.1.1 Αρχιτεκτονική του συστήματος LoRaWAN

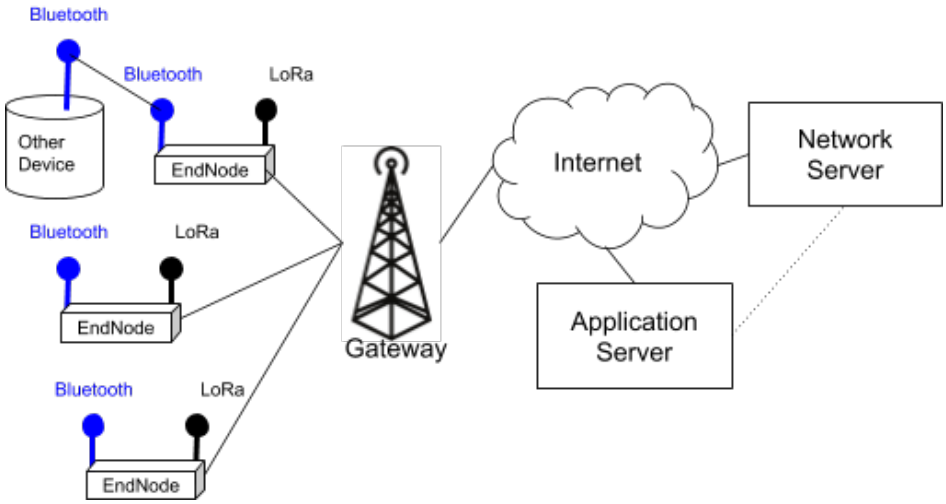
Όπως ήδη έχουμε αναφέρει σε προηγούμενο κεφάλαιο η αρχιτεκτονική ενός δικτύου LoRaWAN περιλαμβάνει τον Application Server on Network Server τα Gateway και φυσικά End Nodes.

Γενική λοιπόν αρχιτεκτονική του συστήματος που θα υλοποιήσουμε περιλαμβάνει το Network Server ο οποίος θα είναι εγκατεστημένος σε ένα Virtual Server στο διαδίκτυο, ένα MQTT Server που θα αναλάβει την επικοινωνία μεταξύ του Network Server , του Gateway και του Application Server. Τη δημιουργία του Αππλιςατιον Σερερ που θα περιλαμβάνει την δημιουργία ενός ωεβ περιβάλλοντος διεπαφής με το χρήστη και και όλη την απαραίτητη επεξεργασιά που χρειάζεται για την υλοποίηση του Broadcast Encryption. Τη Υλοποίηση του Gateway .



Σχήμα 6.1: Αρχιτεκτονική του δικτύου LoRaWAN για το LoRaServer Network Server

Την υλοποίηση Android εφαρμογής για την ανταλλαγή μηνυμάτων και τέλος την υλοποίηση δύο κατηγοριών End Nodes μια με Bluetooth διεπαφή για την ανταλλαγή μηνυμάτων και μια με TFT οθόνη που θα προσομοιάζει τις οθόνες πληροφόρησης και θα υλοποιεί και το Multicast κομμάτι του δικτύου υλοποιώντας την Broadcast Encryption μέθοδο που έχουμε



Σχήμα 6.2: Αρχιτεκτονική του δικτύου LoRaWAN με τις Bluetooth διεπαφές

Virtual Server

Για να μπορεί να λειτουργεί συνεχώς και χωρίς προβλήματα επικοινωνίας ο Network Server και ο Application Server ενοικιάστηκε ένας Virtual Server στην εταιρία Scaleway [41]. Οι πόροι που μας προσφέρει είναι διπύρηνος επεξεργαστής στα 2 GHz με 2 GBytes Ram και 50 GBytes HDD και μια Public IP, σε αυτό εγκαταστήσαμε ένα Image Ubuntu Bionic Beaver. Η πρόσβαση στο Server γίνεται μέσω SSH.

Network Server

Αφού δημιουργήσαμε και παραμετροποιήσαμε το Virtual Server ήρθε η ώρα να εγκαταστήσουμε το Network Server. Ο Network Server που χρησιμοποιούμε είναι ο LoRaServer [42] , είναι ένα έργο ανοικτού λογισμικού φτιαγμένο από τον Orne Brocaar, που μας επιτρέπει να χτίζουμε LoRaWAN δίκτυα. Ο LoRaServer περιλαμβάνει ένα LoRaWAN Network Server που μπορεί να εξυπηρετήσει πολλαπλά Gateway. Υποστηρίζει όλες τις κλάσεις A,B,C του πρωτοκόλλου LoRaWan, υποστηρίζει όλες τις εκδόσεις του LoRaWan μέχρι την 1.1 . Έχει Web Περιβάλλον διεπαφής για να μπορεί ο χρήστης να κάνει τις απαραίτητες ρυθμίσεις στο δίκτυο. Δίνει την δυνατότητα :

- Διαχείρισης από πολλούς χρήστες με διαφορετικό επίπεδο πρόσβασης για πολλαπλά έργα.
- Εισαγωγής πολλών Gateway και την ρυθμίσει αυτών.
- Εισαγωγή πολλών End Nodes την ρύθμιση τους και την παραγωγή των κατάλληλων κλειδιών.
- Κατηγοριοποίηση των End Nodes σε Organization για την λειτουργία πολλαπλών έργων.
- Ο LoRaServer για την επικοινωνία με τον Application Server υποστηρίζει MQTT , gRPC and REST APIs.

Για να μπορέσει να λειτουργήσει ο LoRaServe πρέπει να έχει γίνει μια προεργασία και να έχουν εγκατασταθεί κάποια αλλά βοηθητικά λογισμικά. Πρώτα από όλα πρέπει να εγκατασταθεί ένας MQTT Server, για την γρήγορη απόκριση , την ασφάλεια και την επεκτασιμότητα του συστήματος ο LoRaServe λειτουργεί κάνοντας χρήση του πρωτοκόλλου MQTT, αυτό προϋποθέτει έναν MQTT Server. Επίσης για την αποθήκευση των δεδομένων ο LoRaServe χρησιμοποιεί μια βάση δεδομένων βασισμένη στο PostgresSQL Database. Άρα πρέπει να γίνει και η εγκατάσταση του ανάλογου λογισμικού. Επίσης για την διαχείριση των προσωρινών δεδομένων χρησιμοποιεί το Redis Database. Αναλυτικές οδηγίες για την εγκατάσταση , την ρύθμιση και το τρόπο λειτουργίας όλων των παραπάνω προσφέρονται στη δικτυακή σελίδα του έργου. Αφού εγκαταστήσουμε το LoRaServe , συνδεόμαστε μέσω του γραφικού περιβάλλοντος και κάνουμε τις απαραίτητες ρυθμίσεις. Αρχικά ρυθμίζουμε γενικά το Network Server την IP Address και τα πιστοποιητικά ασφάλειας. Ορίζουμε τα GateWay που έχουμε και τα κανάλια που θα υποστηρίζει κάθε GateWay. Δημιουργούμε Organization για να ομαδοποιήσουμε γενικά τους πόρους μας. Δημιουργούμε Device-profiles στα οποία ρυθμίζουμε τις κλάσεις θα υποστηρίζουν οι συσκευές που ανήκουν σε αυτά τα προφίλ, πως θα συνδέονται με τον Network Server κλπ. Ορίζουμε Application δηλαδή τα έργα στα οποία ανήκουν οι

συσκευές. Εισάγουμε συσκευές στα Application προσδιορίζοντας το Device-profile της συσκευής και ορίζουμε τα DevAddr, AppEUI, AppSKey ή AppEUI, DevEUI, AppKey της συσκευής ανάλογα αν έχουμε επιλέξει η συσκευή να συνδέεται με ABR ή OTAA. Τέλος ο LoRaServe υλοποιήσει μια πολύ απλή ρύθμιση για Multicast, στην οποία μας δίνει την δυνατότητα να φτιάξουμε ομάδες Multicast ,να εισάγουμε συσκευές, να βάλουμε ένα προκαθορισμένο κλειδί MulticastNetwork και ένα προκαθορισμένο κλειδί MulicastApplication και να ορίσουμε τα χαρακτηριστικά του καναλιού. Ότι μήνυμα στείλουμε στην Multicast ομάδα που δημιουργήσαμε αυτό θα κρυπτογραφήσει με τα παραπάνω κλειδιά και θα σταλεί με τις ρυθμίσεις καναλιού της ομάδας, επίσης το πακέτο έχει την μορφή ενός Multicast πακέτου όπως αναφέραμε σε προηγούμενο κεφάλαιο. Σε καμία περίπτωσή δεν υλοποιεί τις οδηγίες LoRaWAN Alliance για Multicast εκπομπές. Εμάς όμως μας βολεύει αυτό που προσφέρει από πλευράς Multicast έτσι φτιάχνουμε μια ομάδα και βάζουμε σε αυτό όλες τις συσκευές.

MQTT Server

Όπως προαναφέραμε ο LoRa Server ανταλλάσσει δεδομένα μέσα από το πρωτόκολλο MQTT. Για το λόγο αυτό πρέπει να εγκαταστήσουμε ένα MQTT Server στον ίδιο υπολογιστή που έχουμε το Network Server ή σε κάποιον άλλο υπολογιστή ή να χρησιμοποιήσουμε κάποιον έτοιμο σαν υπηρεσία. Στη δική μας υλοποίηση εγκαταστήσουμε στον ίδιο Virtual Server που έχουμε το LoRa Server, έναν MQTT Server τον Eclipse Mosquito [43]. Ο Eclipse Mosquitto είναι ένα έργο ανοιχτού λογισμικού από τον οργανισμό της Eclipse Foundation. Αναλαμβάνει να διαχειριστεί και να δια μοιράσει μηνύματα βασισμένα στο πρωτόκολλο MQTT, είναι ελαφρύς, αξιόπιστος και υποστηρίζει μέχρι την έκδοση 5.0 του πρωτοκόλλου . Προσφέρει ισχυρή ασφάλεια μέσω του πρωτοκόλλου SSL/TLS και ενός μηχανισμού διαχείρισης χρηστών που υποστηρίζει . Επίσης υπάρχουν πολλαπλές εκδόσεις το Mosquitto για διάφορα λειτουργικά συστήματα.

Gateway

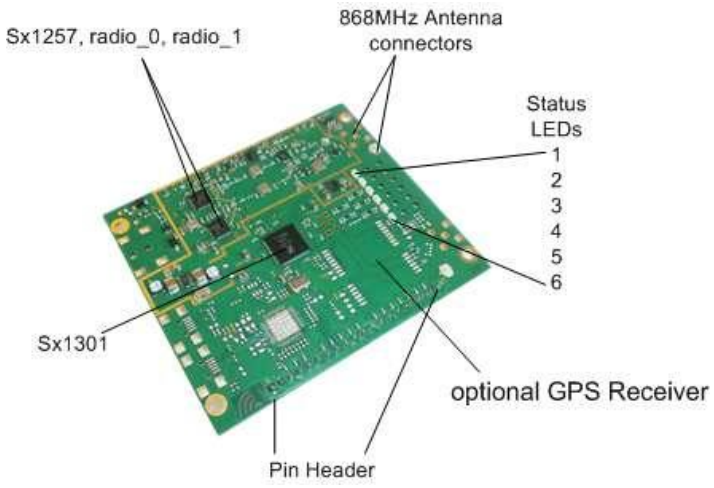
Ένα από τα πιο σημαντικά κομμάτια ενός LoRaWAN δικτύου είναι τα Gateway. Στα περισσότερα LoRaWAN έργα που υλοποιούνται τα Gateway είναι αγορασμένα από κάποια εταιρεία που δραστηριοποιείται στο χώρο των δικτύων. Εμείς λόγο του ότι το έργο είναι ακαδημαϊκού χαρακτήρα αλλά και επειδή το κόστος των εταιρικών Gateway είναι πολύ υψηλό, επιλέξαμε μια λύση ανοιχτού λογισμικού και ανοικτού υλικού. Χρησιμοποιήσαμε ένα πολύ διαδεδομένο και αξιόπιστο Gateway το οποίο αποτελείται από δύο μέρη. Το πρώτο μέρος είναι ένας μικροϋπολογιστής Raspberry pi 3 [44]. Το Raspberry pi 3 με τεχνικά 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

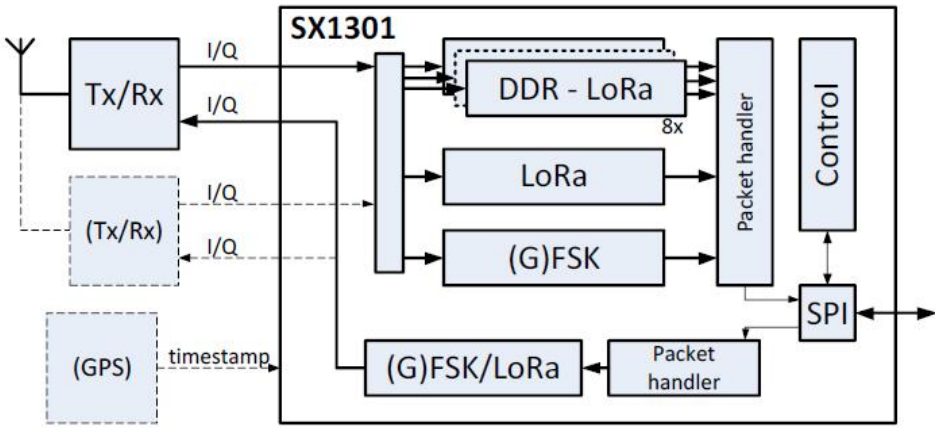
Σχήμα 6.3: Raspberry Pi 3 Model B

Πάνω σε αυτό το μικροϋπολογιστή και συγκεκριμένα στη θύρα SPI προσαρτάται η ηλεκτρονική πλακέτα IC880a [45] της εταιρίας IMST αυτή η πλακέτα αναλαμβάνει όλο το κομμάτι της ασύρματης επικοινωνίας με το LoRa και τα End Nodes ενώ ο μικροϋπολογιστής αναλαμβάνει την επικοινωνία και την μεταφορά των μηνυμάτων από και προς τον Network Server.



Σχήμα 6.4: IC880a

Το IC880a [46] έχει ένα ολοκληρωμένο SX1301 που αναλαμβάνει την παράλληλη διαχείριση των καναλιών και των συντονισμό των δύο SX1257 ολοκληρωμένων που διαχειρίζονται το κομμάτι της ραδιοζεύξης του δικτύουLora.



Σχήμα 6.5: Αρχιτεκτονική IC880a

Τα χαρακτηριστικά του IC880a:

- LoRa ® modulation technology
- Frequency band 868 MHz
- Orthogonal spreading factors
- Sensitivity down to -137 dBm
- SPI interface
- SX1301 base band processor
- Emulates up to 49 x LoRa demodulators
- 10 parallel demodulation paths
- 1 (G)FSK demodulator
- 2 x SX1257 Tx/Rx front-ends
- Supply voltage 5 V
- RF interface optimized to 50
- Output power level up to 20 dBm
- GPS receiver (optional)
- Range up to 15 km (Line of Sight)

Συνήθως στο Raspberry pi υπάρχει εγκατεστημένο στην κάρτα SD μία έκδοση του λειτουργικού raspbian που είναι ένα λειτουργικό τύπου Linux για τη λειτουργία του μικροϋπολογιστή . Για την επικοινωνία του Raspberry pi με το IC880a εγκαθιστούμε το λογισμικό ανοιχτού κώδικα Lora Gateway [47] μια συνεργασία της Lora Alliance και της εταιρίας Semtech. Επίσης εγκαθιστούμε το packet_forwarder που προέρχεται από

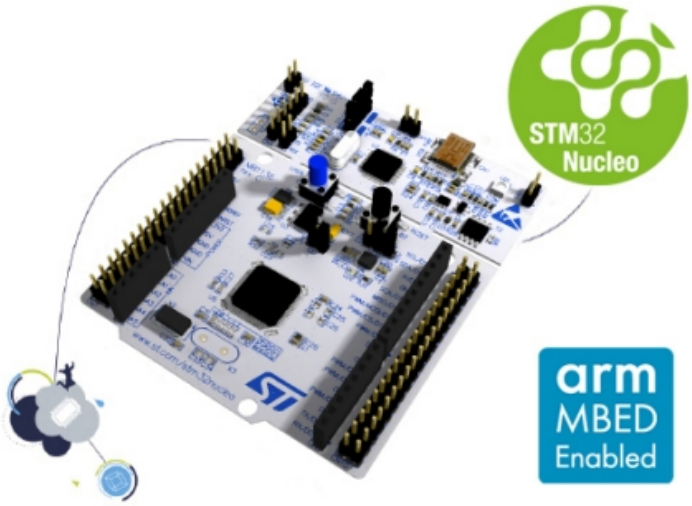
την ίδια συνεργασία για την επικοινωνία του Raspberry pi με το Network Server μέσο πρωτοκόλλου UDP. Εμείς χρησιμοποιούμε μια έτοιμη διανομή για το Raspberry pi το Lora Gateway OS που είναι κομμάτι του έργου LoRaServer. Αυτό περιέχει εκτός από τα παραπάνω επιπλέον ένα MQTT Client και ένα γραφικό περιβάλλον ρυθμίσεων.

End Nodes

Οι πλακέτες που έχουμε χρησιμοποιήσει στις τελικές συσκευές έχουν όλες επεξεργαστές της ARM και υποστηρίζουν το λειτουργικό σύστημα Mbed OS. Οι τύποι των τελικών συσκευών που χρησιμοποιήσαμε είναι δύο .

Τύπος Συσκευών Α Ο πρώτος τύπος συσκευών είναι αυτές που χρησιμοποιούνται για την ανταλλαγή μηνυμάτων με το κέντρο διαχείρισης. Έχουν γίνει δοκιμές με δυο διαφορετικές πλακέτες:
Πρώτη Συσκευή
Η πρώτη συσκευή είναι το NUCLEO-L073RZ [48] της εταιρείας ST με μικρο ελεγκτή STM32L073RZT6 in LQFP64 package και χαρακτηριστικά:

- ARM®32-bit Cortex®-M0+ CPU
- 32 MHz max CPU frequency
- VDD from 1.65 V to 3.6 V
- 192 KB Flash
- 20 KB SRAM



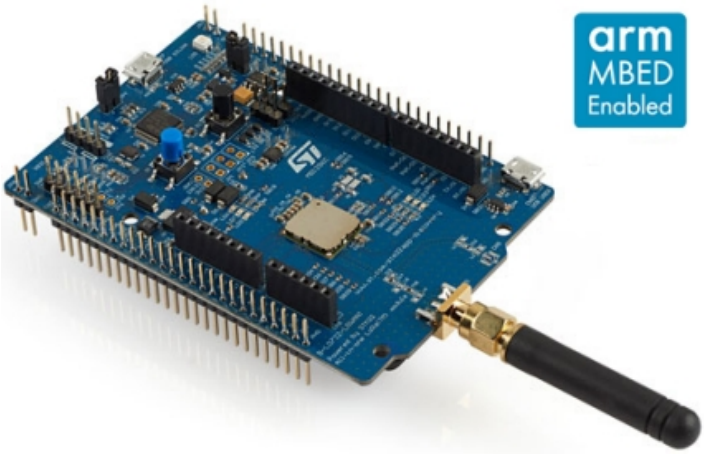
Σχήμα 6.6: NUCLEO-L073RZ

Πάνω σε αυτό το Board τοποθετήσαμε για την σύνδεση με το δικτυο LoRa το SX1272MB2xAS [49].

Σχήμα 6.7: SX1272MB2xAS

Το SX1272MB2xAS έχει πάνω του ένα ολοκληρωμένο SX1272 για την επικοινωνία με το δίκτυο LoRa ,το SX1272 επικοινωνεί με τον μικρο ελεγκτή STM32L073RZT6 μέσω θύρας SPI.
Δεύτερη συσκευή
Η δεύτερη συσκευή είναι το DISCO-L072CZ-LRWAN1 [50] της εταιρείας ST με μικρο-ελεγκτή CMWX1ZZABZ-091 και χαρακτηριστικά :

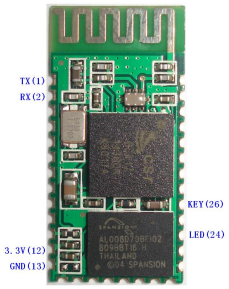
- LoRa® module with STM32L072CZ
- ARM® 32-bit Cortex®-M0+ CPU
- 32 MHz max CPU frequency
- VDD from 1.65 V to 3.6 V
- 192 KB Flash
- 20 KB SRAM
- SX1276 transceiver.



Σχήμα 6.8: DISCO-L072CZ-LRWAN1

Αυτό έχει το ολοκληρωμένο SX1276 για την επικοινωνία με το δίκτυο LoRa. Όπως

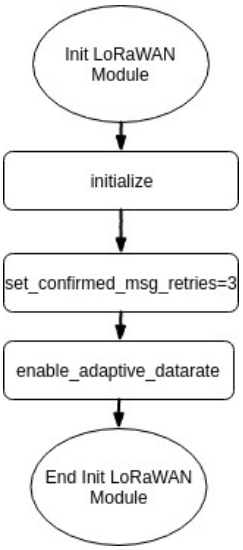
με την προηγούμενη συσκευή η επικοινωνία μικρο ελεγκτή , ολοκληρωμένου LoRa γίνεται μέσω θύρας SPI. Για να μπορούν οι συσκευές μας να επικοινωνούν με άλλες συσκευές , όπως για παράδειγμα Tablet ή Smart Phone συνδέσαμε πάνω στα Board ένα Bluetooth Module. Έτσι μπορούμε να συνδεόμαστε με Android συσκευές μέσω ενός λογισμικού που φτιάξαμε κάνοντας χρήση του Bluetooth.Στη συνέχεια στέλνουμε μηνύματα στοBboard μέσω Bluetooth , τα μηνύματα μέσω του SX1276/SX1272 περνάνε στο δίκτυο LoRaWAN. Βέβαια συμβαίνει και το αντίστροφο τα μηνύματα εκπέμπονται στο δίκτυο LoraWAN μέσω του Gateway λαμβάνονται από το ολοκληρωμένο SX1276/SX1272 και στη συνέχεια μέσω του Bluetooth περνάνε στην Android συσκευή και εμφανίζονται στην οθόνη. Serial Bluetooth.
Για την Bluetooth επικοινωνία χρησιμοποιούμε τη συσκευή HC-06. Είναι μία συσκευή χαμηλού κόστους και εύκολη στη λειτουργία. Συνδέεται μέσω σειριακής θύρας με το μικροεπεξεργαστή και επικοινωνεί με αυτόν χρησιμοποιώντας ΑΤ εντολές.



Σχήμα 6.9: HC-06

Και στους δυο τύπους συσκευών χάρης το λειτουργικό MBed OS ο κώδικας προγραμματισμού που χρησιμοποιούμε είναι ο ίδιος εκτός από μερικές αρχικοποιήσεις στην αρχή. Το πρόγραμμα που γράψαμε περιλαμβάνεις δυο κλάσεις και την αρχική συνάρτηση main που καλεί και συνδέει τις κλάσεις μεταξύ τους. Η πρώτη κλάση είναι η SerialBluetooth που χρησιμοποιεί την βιβλιοθήκη Raw Serial του MBed OS ελέγχει την επικοινωνία του Bluetooth, λαμβάνει τα μηνύματα που έρχονται από το Bluetooth και τα προωθεί στη κλάση που ελέγχει το δίκτυο LoRaWAN επίσης λαμβάνει τα μηνύματα που έρχονται από το δίκτυο LoRaWAN και τα προωθεί στο Bluetooth. Η δεύτερη κλάση είναι η MultiLora που ελέγχει το δίκτυο LoRaWAN χρησιμοποιώντας την βιβλιοθήκη LoraWAN network interface του MBed OS αυτή κάνει τα εξής.

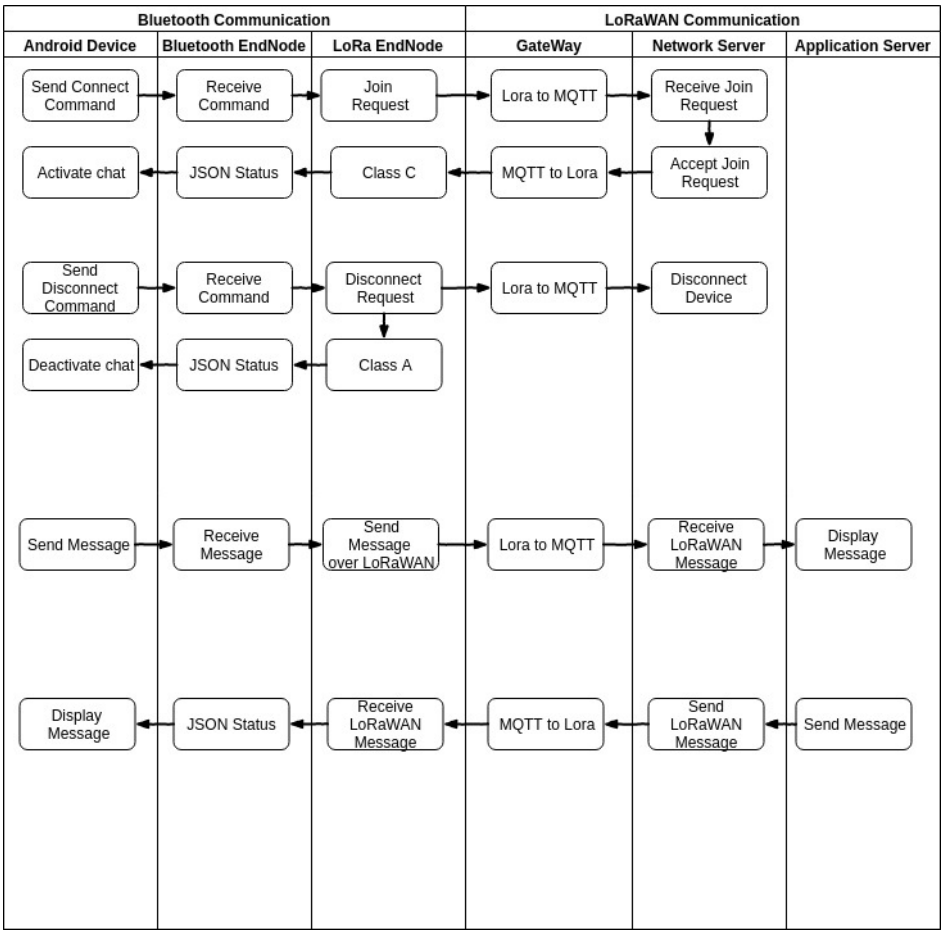
- Αρχικοποιεί τα ολοκληρωμένα SX1276 / SX1272
- Κάνει τις απαραίτητες ενέργειες για να κάνει Join στο δίκτυο LoRaWAN.
- Αφού κάνει Join Βάζει την συσκευή σε κατάσταση Class C στο δίκτυο LoRaWAN.
- Προωθεί τα μηνύματα που έρχονται από το δίκτυο LoRaWAN στη κλάση του Bluetooth.
- Λαμβάνει τα μηνύματα από την κλάση του Bluetooth και τα προωθεί στο δίκτυο LoRaWAN.



Σχήμα 6.10: Διαδικασία Αρχικοποίησης από την κλάση Multilora

Στο επόμενο σχεδιάγραμμα παρουσιάζονται συνοπτικά οι 4 διαδικασίες που υπάρχουν για την ανταλλαγή μηνυμάτων, τα μέρει που περνούν μέρος στην διαδικασίες όπως επίσης και τις βασικές λειτουργίες που γίνονται. Οι βασικές διαδικασίες είναι :

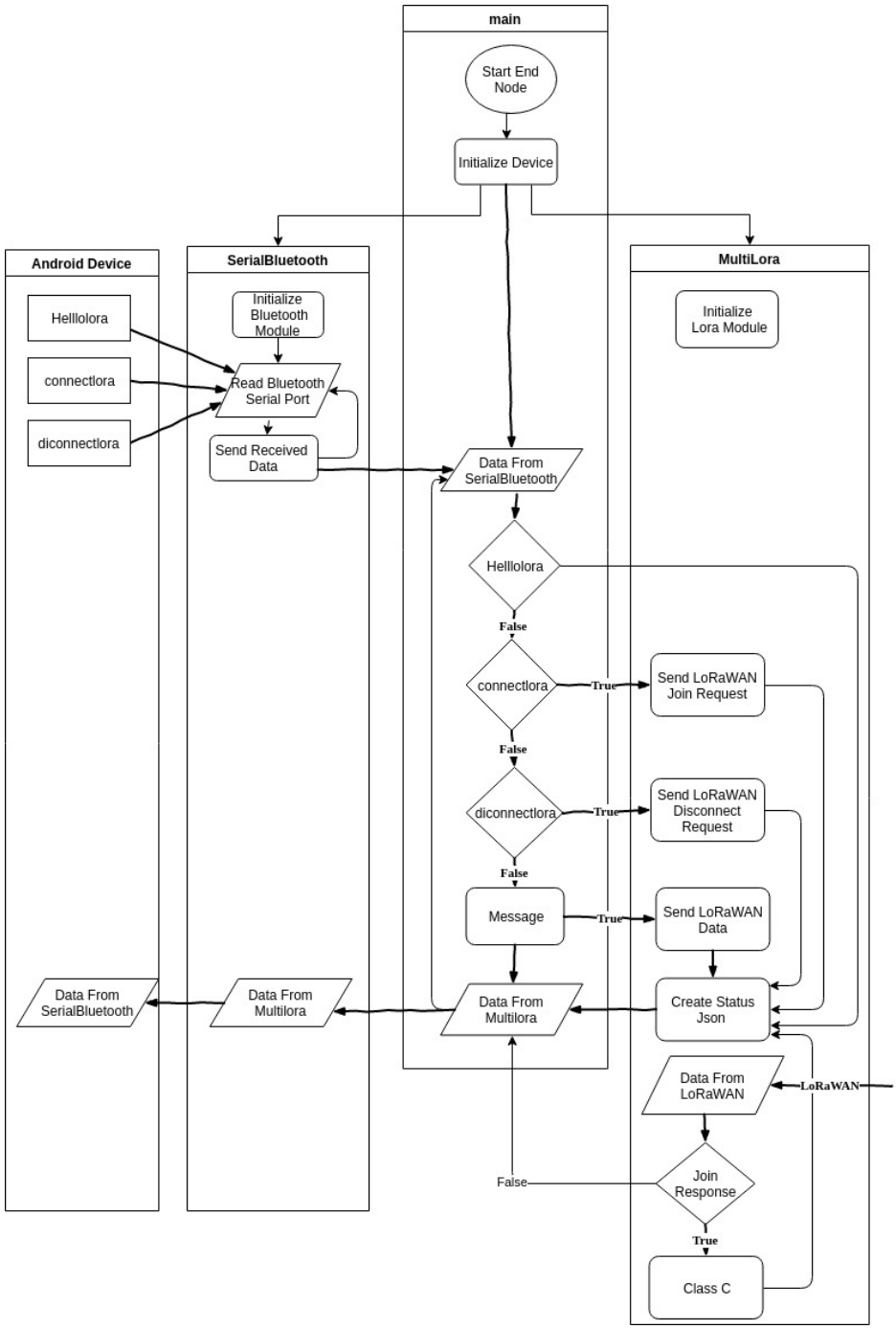
- Όταν ο χρήστης ζητά να συνδεθεί στο LoRaWAN δίκτυο από την Android συσκευή του μέχρι τον NetWork Server.
- Όταν ο χρήστης ζητά να αποσυνδεθεί από το LoRaWAN δίκτυο.
- Όταν ο χρήστης στέλνει ένα μήνυμα από την Android συσκευή του στον Application Server.
- Όταν από τον Application Server στέλνεται ένα μήνυμα στην Android συσκευή του χρήστη .



Σχήμα 6.11: Διαδικασίες αποστολής μηνυμάτων

Ο χρήστης για να επικοινωνήσει με το κέντρο επιχειρήσεων πρέπει να ενεργοποίηση την End Node συσκευή και να εκτελέσει στην Android συσκευή του την ειδική εφαρμογή. Πρέπει επίσης να κάνει Bluetooth pair τις δυο συσκευές, έπειτα από το γραφικό περιβάλλον της εφαρμογής επιλέγει την Bluetooth σύνδεση με το End Node. Τότε η εφαρμογή στέλνει ένα μήνυμα hellolora και ο End Node απαντάει με ένα μήνυμα σε μορφή JSON που περιέχει την κατάσταση που βρίσκεται. Στη συνέχεια ο χρήστης από την Android εφαρμογή μπορεί να ζητήσει να συνδεθεί στο δίκτυο LoRaWAN. Η εφαρμογή στέλνει ένα μήνυμα connectlora, μόλις ο End Node λάβει το μήνυμα κάνει Join Request στον Network Server, όταν πάρει απάντηση Join Response από τον Network Server ή αν περάσει ο προκαθορισμένος χρόνος , στέλνει πίσω στην εφαρμογή ένα μήνυμα σε μορφή JSON που περιέχει την κατάσταση του, αν είναι ή όχι συνδεδεμένος στο LoRaWAN δίκτυο. Αν το Join Response από τον Network Server είναι θετικό τότε ο End Node γυρίζει από Class A που ήταν αρχικά σε Class C. Αν ο End Node είναι σε κατάσταση Connect τότε η εφαρμογή επιτρέπει την πληκτρολόγηση και την αποστολή μηνύματος. Όταν ο χρήστης θέλει να στείλει ένα μήνυμα το πληκτρολογεί και πατάει την αποστολή. Το μήνυμα στέλνεται από την εφαρμογή μέσω Bluetooth στον End Node και από εκεί προωθείται στο LoRaWAN δίκτυο, μέχρι να καταλήξει στον Application Server και στην εφαρμογή που διαχειρίζεται τα εισερχόμενα μηνύματα, αυτή η εφαρμογή αναλαμβάνει να προβάλει τα μηνύματα στην οθόνη του χρήστη στο κέντρο επιχειρήσεων. Στην περίπτωση που ο χρήστης από το κέντρο επιχειρήσεων θέλει να στείλει ένα μήνυμα τότε από την εφαρμογή στον Application Server επιλέγει

τον End Node που θέλει να στείλει το μήνυμα, γράφει το μήνυμα και πατάει αποστολή. Το μήνυμα προωθείται στο Network Server από εκεί στο κατάλληλο Gateway και μέσω του δικτύου LoRa στο End Node εκεί γίνονται οι απαραίτητες ενέργειες (έλεγχοι , αποκρυπτογράφηση, κλπ), αφού γίνουν όλα αυτά φτάνει στην κλάση Multilora, αυτή μόλις λάβει το μήνυμα το κωδικοποιεί σε μορφή JSON το εμπλουτίζει με πληροφορίες όπως το RSSI,SNR, κάποιες άλλες πληροφορίες και το στέλνει μέσω της κλάσης SerialBluettoth και της Bluetooth σύνδεσης στην εφαρμογή που εκτελείται στην Android συσκευή. Η Εφαρμογή αναλαμβάνει να πάρει το μήνυμα να το επεξεργαστεί και να το εμφανίσει στο χρήστη , παράλληλα εμφανίζει και κάποιες άλλες πληροφορίες όπως τα RSSI, SNR, Class. Τέλος για να αποσυνδεθεί ο χρήστης από το δίκτυο μέσω της εφαρμογής στέλνει ένα μήνυμα disconnectlora στον End Node, αυτός στέλνει ένα μήνυμα αποσύνδεσης στο Network Server και περνάει σε Class A και στέλνει πίσω στον End Node ένα μήνυμα σε μορφή JSON που περιέχει την κατάσταση του.

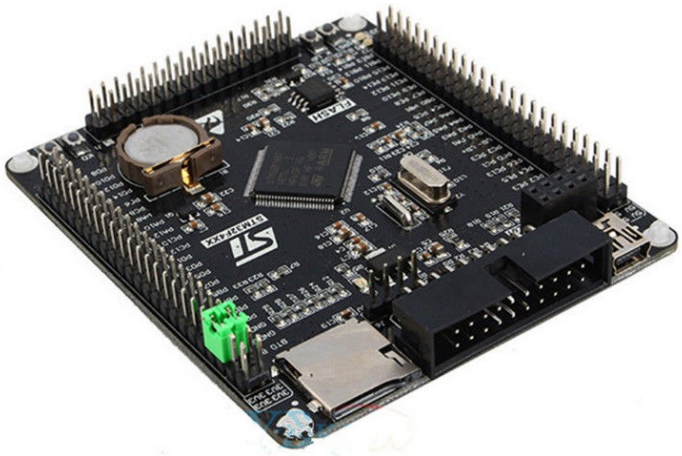


Σχήμα 6.12: Βασικές εργασίες που εκτελούνται στο End Node

Τύπος Συσκευών Β Η δεύτερη κατηγορία συσκευών βασίζεται στο Board STM32F407VET6 black board [51] έχει τον μικροελεγκτή STM32F407VET6 in LQFP100 package [52] με τα παρακάτω χαρακτηριστικά:

- ARM®32-bit Cortex®-M4 CPU + FPU
- 168 MHz max CPU frequency
- VDD from 1.8 V to 3.6 V
- 512 KB Flash

- 192+4 Kbytes of SRAM including 64-Kbyte of CCM (core coupled memory) data RAM



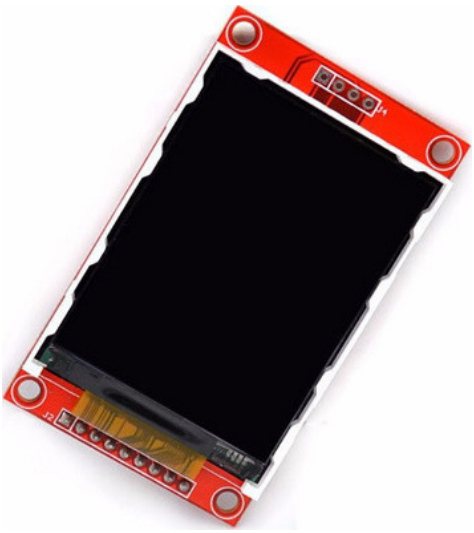
Σχήμα 6.13: STM32F407VET6

Για το δεύτερο τύπο συσκευών επειδή υλοποιούν τη διαδικασία του Broadcast Encrypt χρειαζόμασταν ένα πιο ισχυρό επεξεργαστή με περισσότερη μνήμη. Στο STM32F407VET6 συνδέσαμε μέσω της θύρας SPI ένα RFM95 [53] για να μπορούμε να επικοινωνήσουμε με το δίκτυο LoRaWAN, επίσης πάνω στη δεύτερη SPI θύρα συνδέσαμε μια μια οθόνη ILI9341 [54] για να μπορούμε να βλέπουμε τα μηνύματα που στέλνει ο Application Server. Όπως και οι προηγούμενοι μικροελεγκτές έτσι και αυτός υποστηρίζει το λειτουργικό σύστημά MBed OS .

Σχήμα 6.14: INAIR9

Σχήμα 6.15: RFM95

Για τον προγραμματισμό και την διαχείριση της οθόνης χρησιμοποιήσαμε την βιβλιοθήκη SPI driven QVGA TFT που υπάρχει στο αποθετήριο του MBed OS. Για την λειτουργία και την διαχείριση του δικτύου LoRaWAN χρησιμοποιούμε την κλάση Mutli-Lora από τον προηγούμενο τύπο συσκευών. Μόνο που εδώ έχουμε επέμβει και έχουμε πειράξει ελαφρός την βιβλιοθήκη LoraWAN network interface του MBed OS για να μπορούμε να ρυθμίσουμε το Multicast.



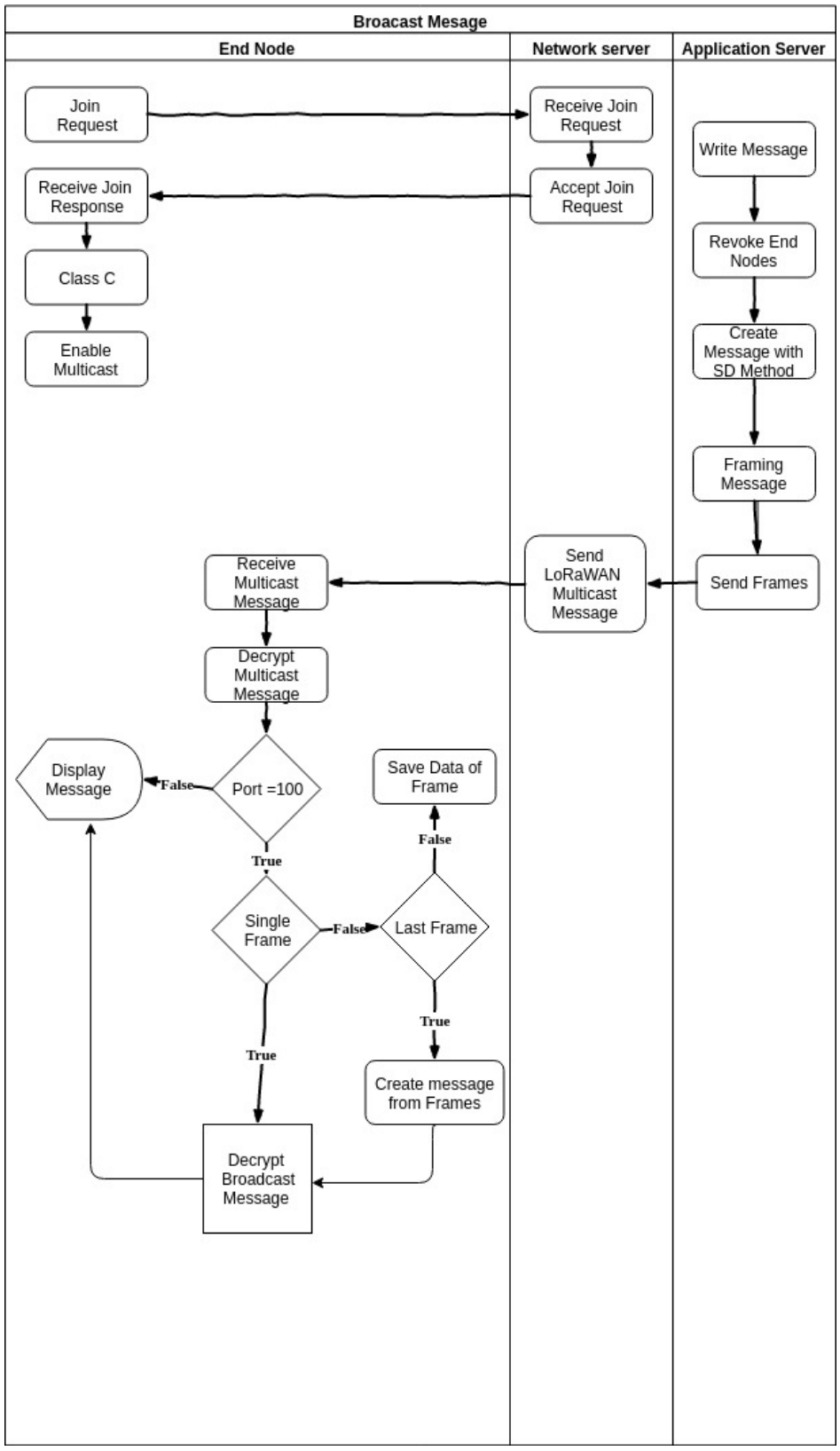
Σχήμα 6.16: ILI9341

Επίσης εδώ υλοποιούμε και τους αλγόριθμους της τεχνικής Subset Difference έτσι έχουμε τις κλάσεις btree που φτιάχνει και διαχειρίζεται το δυαδικό δέντρο για να αναπαραχθούν τα κλειδιά , και την κλάση NNLDecrypt που αναλαμβάνει σε συνεργασία με τη κλάση btree και την βιβλιοθήκη mbedtls να υλοποιήσει την εύρεση των κλειδιών και τις αποκρυπτογραφήσεις.

Μια End Node συσκευή στη δεύτερη κατηγορία μόλις ξεκινήσει αρχικοποιεί τα περιφερειακά της αρχικοποιεί τον εαυτό της σαν μια συσκευή τύπου Class A για το δίκτυο

LoRaWAN και κάνει μια αίτηση Join Request στον Network Server. Αν ο Network Server δεχτεί την αίτηση στέλνει ένα Accept Join Response, μόλις ο End Node λάβει την απάντηση, βάζει το εαυτό του σε Class C, απενεργοποιεί το ADR και ρυθμίζεται να μπορεί να δεχτεί LoRaWAN Multicast μηνύματα, με προκαθορισμένη διεύθυνση και κλειδιά. Μέτα από αυτή την διαδικασία είναι σε κατάσταση αναμονής για τη λήψη Multicast μηνυμάτων.

Στον Application Server και συγκεκριμένα στη εφαρμογή που υπάρχει εκεί για αυτό το σκοπό επιλέγονται ποιοι End Nodes θα πάρουν το μήνυμα, αυτό κρυπτογραφείται, προστίθενται οι κεφαλίδες (Headers), τμηματοποιήται (Framing) αν χρειάζεται και στέλνετε στο Network Server από εκεί στέλνονται ένα ή περισσότερα LoRaWAN μηνύματα. Τα μηνύματα είναι Multicast μηνύματα σε μια προκαθορισμένη ομάδα με προκαθορισμένα χαρακτηριστικά (διεύθυνση , κλειδιά ,συχνότητα, DR, κλπ). Ο End Node αφού είναι σε Class C και γνωρίζει τις ρυθμίσεις της ομάδας μπορεί να πάρει και να αποκρυπτογραφήσει το μήνυμα. Αρχικά ελέγχει την πόρτα στην οποία στάλθηκε το μήνυμα, αν δεν είναι στην πόρτα 100 το θεωρεί απλό Multicast μήνυμα και απλός το εμφανίζει. Αν είναι στη πόρτα 100 τότε το θεωρεί Broadcast Encrypted μήνυμα και ακολουθεί την παρακάτω διαδικασία.



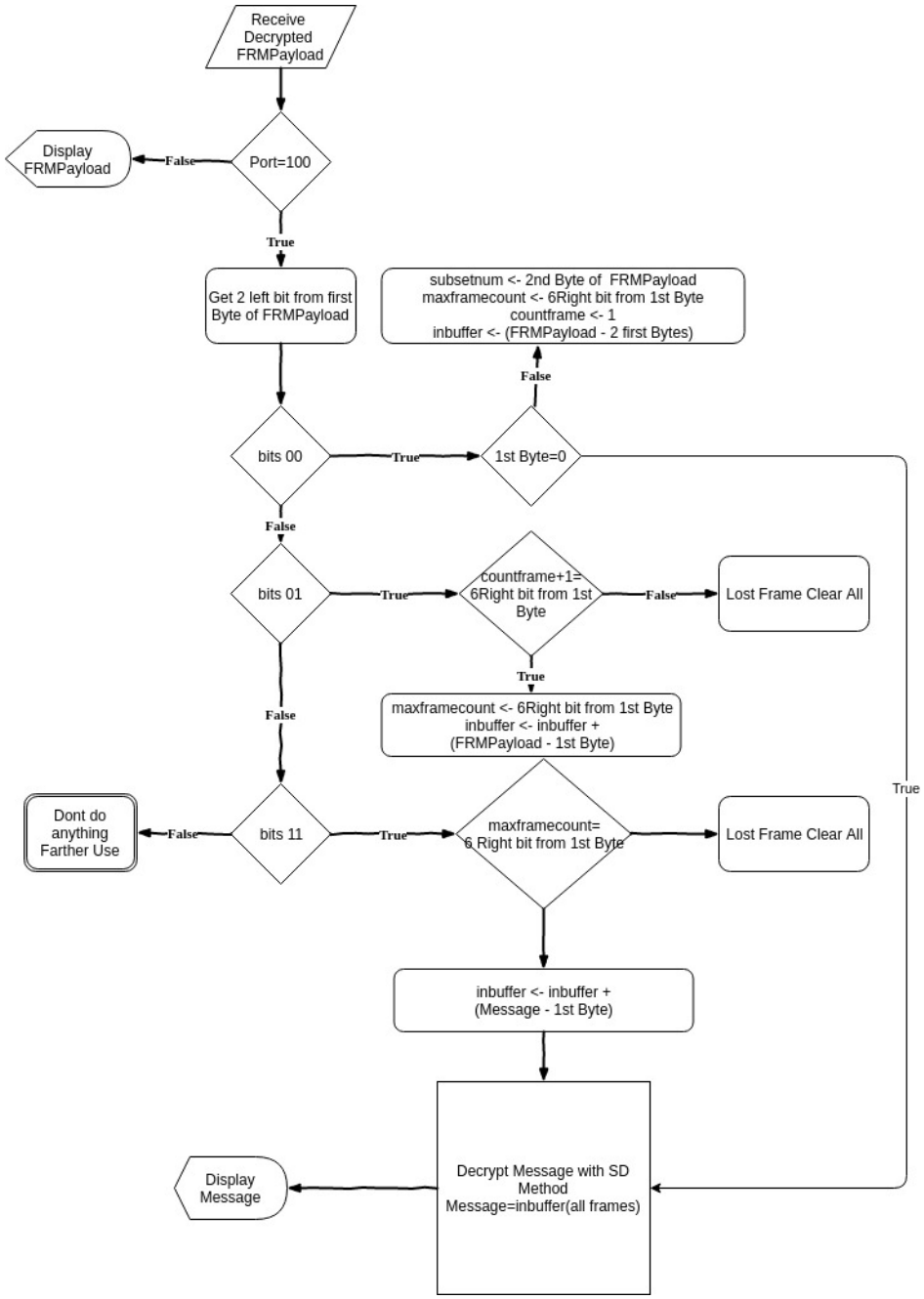
Σχήμα 6.17: Αρχικοποίηση του End Node και διαδικασία λήψης Multicast Μηνύματος

Στο πρώτο μήνυμα ελέγχει τα πρώτα 2 bit του πρώτου Byte αν είναι 00 και όλο το Byte μηδέν τότε είναι ένα αυτοδύναμο μήνυμα, διαβάζει το δεύτερο Byte από εκεί ξέρει πόσο μεγάλο είναι το Header του μηνύματος για να το ξεχωρίσει από το κρυπτογραφημένο

μήνυμα. Στην συνέχεια το στέλνει για αποκρυπτογράφηση.
Αν τα πρώτα 2 bit του πρώτου Byte αν είναι 00 αλλά το υπόλοιπό Byte έχει μια τιμή τότε σημαίνει ότι είναι το πρώτο τμήμα μηνύματος από ένα μήνυμα που έχει τόσα τμήματα όσα η τιμή των 6 τελευταίων Bit του Byte. Πάλι διαβάζει το δεύτερο Byte για να ξέρει πόσο μεγάλο είναι το Header του μηνύματος για να το ξεχωρίσει από το συνολικό κρυπτογραφημένο μήνυμα, έπειτα αποθηκεύει το τμήμα μηνύματος χωρίς τα πρώτα 2 Byte σε ένα Buffer (inbuffer). Επίσης στη μεταβλητή countframe που δείχνει τον αύξων αριθμό του τμήματος που έχει έρθει μέχρι τώρα βάζει την τιμή 0.
Αν από το πρώτο μήνυμα γνωρίζει ότι θα ακολουθήσουν και άλλα τμήματα του μηνύματος για κάθε μήνυμα ελέγχει το πρώτο Byte.

- Αν τα πρώτα 2 bit του πρώτου Byte αν είναι 01 τότε σημαίνει ότι είναι ένα ακόμα τμήμα του συνολικού μηνύματος. Ελέγχει τα 6 τελευταία Bit του πρώτου Byte αυτά τώρα δείχνουν το αύξων αριθμό του τμήματος. Αν μεταβλητή countframe +1 είναι ίδια με την τιμή από τα 6 τελευταία Bit τότε βάζει στο Buffer (inbuffer) το τμήμα μηνύματος χωρίς το πρώτο Byte ,αλλιώς θεωρεί ότι έχασε κάποιο τμήμα του μηνύματος και μηδενίζει όλα τα προηγούμενα.

- Αν τα πρώτα 2 bit του πρώτου Byte αν είναι 11 τότε σημαίνει ότι είναι το τελευταίο τμήμα του συνολικού μηνύματος. Ελέγχει τα 6 τελευταία Bit του πρώτου Byte αυτά τώρα δείχνουν το αύξων αριθμό του τμήματος. Αν μεταβλητή countframe +1 είναι ίδια με την τιμή από τα 6 τελευταία Bit τότε βάζει στο Buffer (inbuffer) το τμήμα μηνύματος χωρίς το πρώτο Byte και το στέλνει για αποκρυπτογράφηση, αλλιώς θεωρεί ότι έχασε κάποιο τμήμα του μηνύματος και μηδενίζει όλα τα προηγούμενα.



Σχήμα 6.18: Διαδικασία ανασύνθεσης των τμημάτων ενός Multicast Μηνύματος

Αφού έχουμε επανασυνθέσει το μήνυμα ξεχωρίζουμε την κεφαλίδα Header που περιέχει τα Subset[1] (Start Node[2], Revoke Node[3]) και τα κρυπτογραφημένα κλειδιά EncGeneralKey από το κρυπτογραφημένο μήνυμα. Ελέγχουμε τα Subset από την κεφαλίδα αν ταιριάζουν με τα αποθηκευμένα Subset.

- Το Start Node κάποιου Subset από την κεφαλίδα να ταιριάζει με το Start Node κάποιου τα αποθηκευμένα Subset και επίσης να ταιριάζουν και τα Revoke Node.

[1]Υποσύνολο που ορίζει τους κόμβους που θα πάρουν το μήνυμα, τους κόμβους που θα αποκλειστούν και για κάθε υποσυνολο υπάρχει ένα κλειδί αποκρυπτογράφησης.

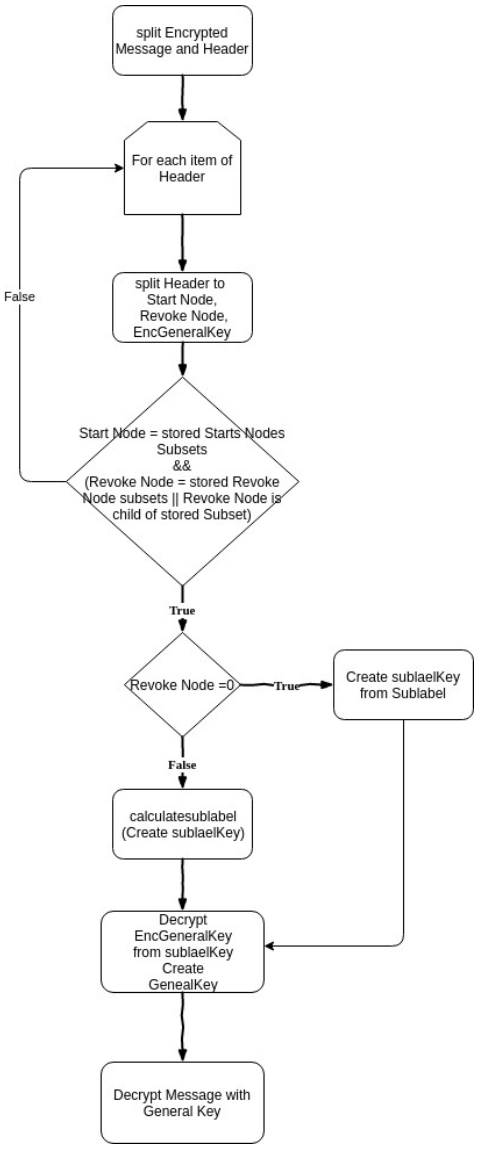
[2]Ο κόμβος κάθε Subset που ορίζει ποιοι κόμβοι κάτω απο αυτόν θα πάρουν το μήνυμα.

[3]Οι κόμβοι κάθε Subset που είναι απόγονοι του Start Node και ορίζουν ότι οι απόγονοι τους δεν θα λάβουν το μήνυμα.

Σε αυτή την περίπτωση παίρνουμε το sublabel[4] από το αποθηκευμένο Subset το βάζουμε στην συνάρτηση G_M[5] και παίρνουμε το κλειδί της αποκρυπτογράφησης του κρυπτογραφημένου κλειδιού του μηνύματος.

- Το Start Node κάποιου Subset από την κεφαλίδα να ταιριάζει με το Start Node κάποιου τα αποθηκευμένα Subset και επίσης το Revoke Node του Subset από την κεφαλίδα να είναι απόγονος του Start Node από τα αποθηκευμένα Subset και πρόγονος του Revoke Node από τα αποθηκευμένα. Σε αυτή τη περίπτωση πρέπει να βρούμε με την διαδικασία της ετικετοποίησης Labeling το κλειδί της αποκρυπτογράφησης του κρυπτογραφημένου κλειδιού του μηνύματος.

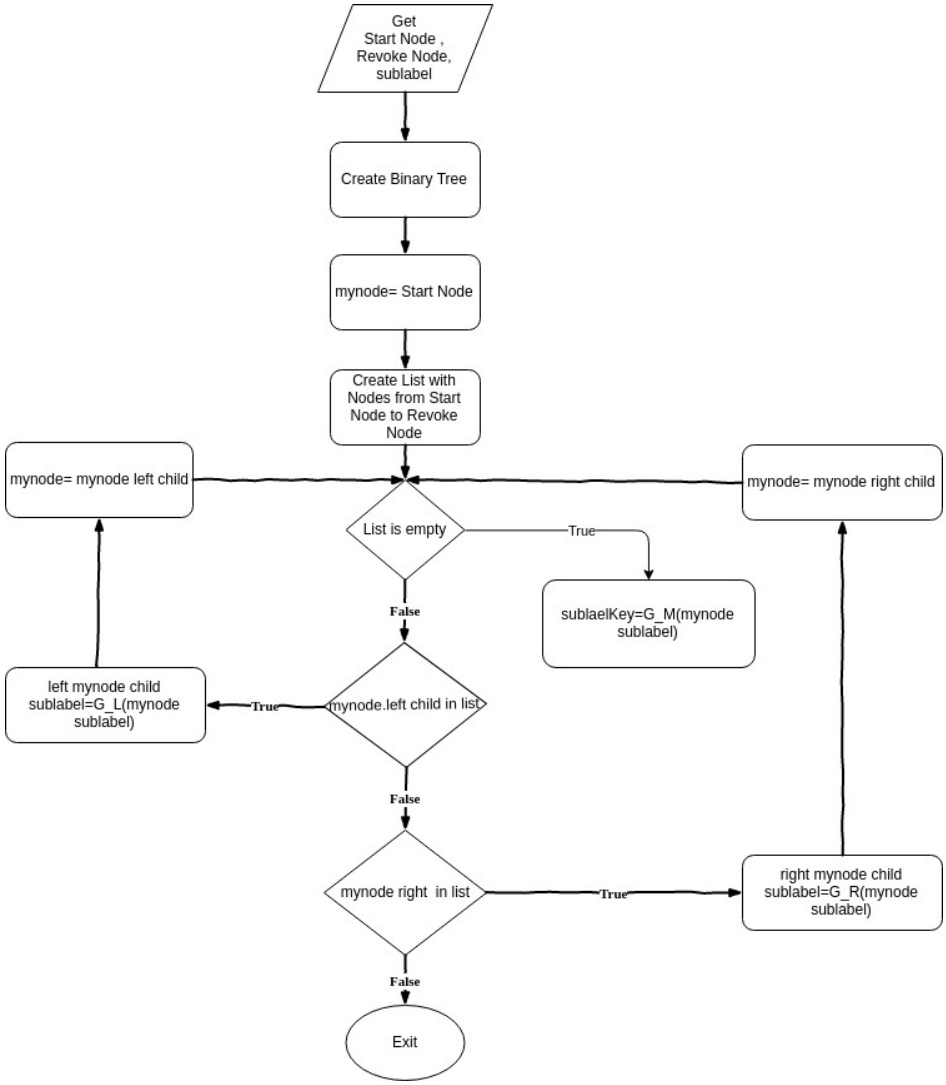
Αφού έχουμε το κλειδί αποκρυπτογραφούμε το κρυπτογραφημένο κλειδί του Subset στην κεφαλίδα και έχουμε το κλειδί με το οποίο θα αποκρυπτογραφήσουμε το κρυπτογραφημένο μήνυμα.



Σχήμα 6.19: Διαδικασία αποκρυπτογράφησης

[4]Το Sublabel που έχει δημιουργηθεί από την διαδικάσια ετικετοποιήσης στον Application Server.
[5]Μονόδρομη συνάρτηση που από το Sublabel δημιουργεί το κλειδί.

Στις κρυπτογραφήσεις και αποκρυπτογραφήσεις χρησιμοποιούμε τον αλγόριθμο RC4 [55] με 40 Bit κλειδί. Για την διαδικασία της ετικετοποίησης Labeling που έχουμε περιγράψει σε προηγούμενο κεφάλαιο σαν μονόδρομες συναρτήσεις έχουμε χρησιμοποιήσεις τον αλγόριθμο HMAC SHA1 [56] για την συνάρτηση G_L χρησιμοποιούμε τον αλγόριθμο με κλειδί left, για την συνάρτηση G_R χρησιμοποιούμε τον αλγόριθμο με κλειδί right και για την συνάρτηση G_M χρησιμοποιούμε τον αλγόριθμο με κλειδί key.

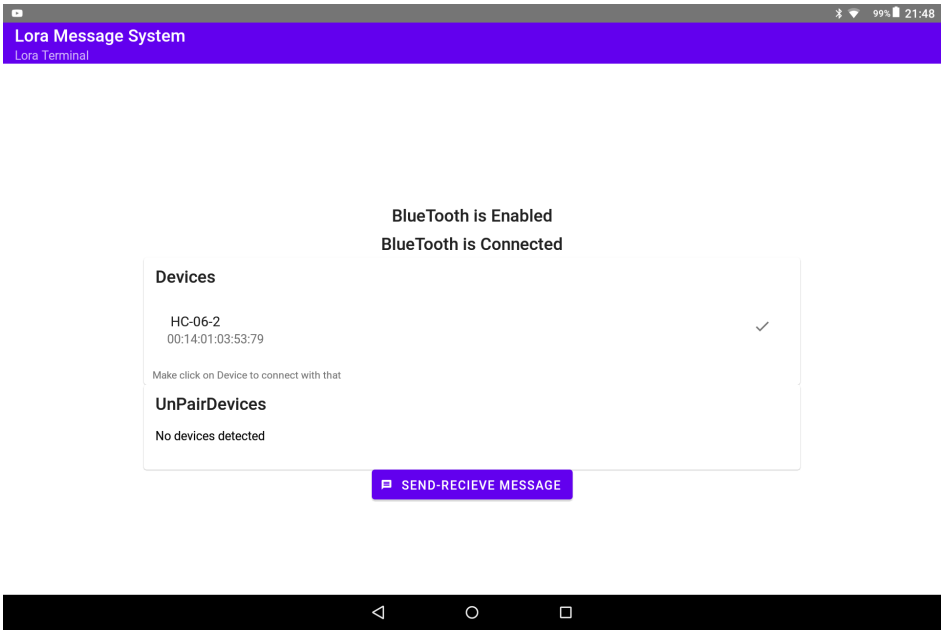


Σχήμα 6.20: Διαδικασία εύρεσης κλειδιού με ετικετοποίηση

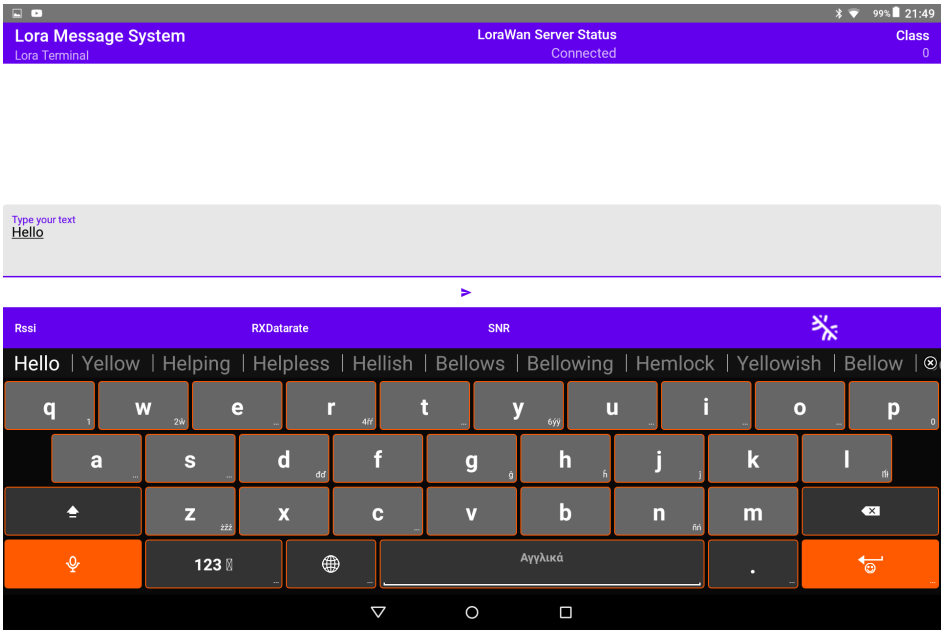
Android Εφαρμογή Ανταλλαγής Μηνυμάτων

Η εφαρμογή που αναπτύχθηκε για συσκευές Android , δίνει την δυνατότητα αφού έχει ενεργοποιήσει το Bluetooth της Android συσκευής να δείχνει τις συσκευές για τις οποίες γίνει αποδεχτή η σύνδεση (pair). Αν έχει ενεργοποιηθεί το End node, στη λίστα αυτών των συσκευών θα εμφανίζεται το Bluetooth που έχει τοποθετηθεί στο μικροελεγκτή στις συσκευές Α κατηγορίας και να μας διδέται η δυνατότητα να συνδεθούμε με αυτό 6.21. Αφού συνδεθούμε εμφανίζεται μια δεύτερη οθόνη στη συσκευή Android που μας ενημερώνει για την κατάσταση του δικτύου LoRaWAN στο μικροελεγκτή και

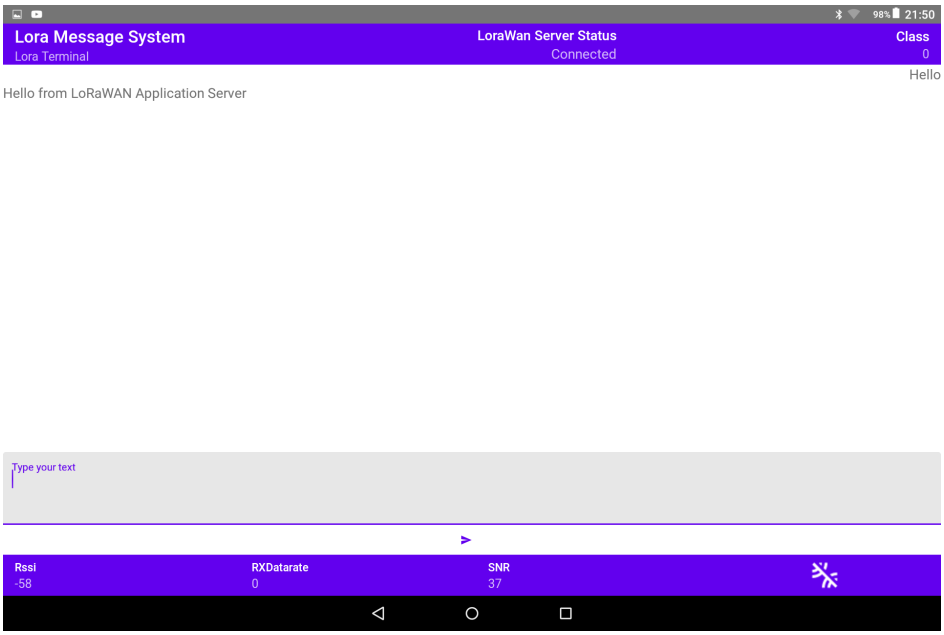
μας δίνει την δυνατότητα να συνδέσουμε , να αποσυνδέουμε το μικροελεγκτή από το δίκτυο LoRaWAN 6.25. Επίσης μέσω του γραφικού περιβάλλοντος που μας δίνει η εφαρμογή μπορούμε να γράψουμε και να στείλουμε ένα μήνυμα 6.22 μέσω της Bluetooth σύνδεσης στο μικροελεγκτή και από εκεί να προωθηθεί στο δίκτυο LoRaWAN και τελικά στο Application Server. Το ίδιο γίνεται και αντίστροφα δηλαδή ένα μήνυμα που έχει φτάσει στο μικροελεγκτή από το δίκτυο LoRaWAN προωθείται μέσω του Bluetooth στη εφαρμογή και αυτή παρουσιάζει το μήνυμα στο χρήστη. Η εφαρμογή έχει φτιαχτεί με την χρήση του Framework React Native, είναι συμβατή με τις περισσότερες εκδόσεις Android, τουλάχιστον όσες υποστηρίζει το React Native και είναι εύκολα προσαρμόσιμο και για συσκευές με IOS λειτουργικό.



Σχήμα 6.21: Αρχική Οθόνη Εφαρμογής σύνδεσης με την συσκευή



Σχήμα 6.22: Οθόνη αποστολής μηνύματος



Σχήμα 6.23: Οθόνη μηνυμάτων

Εφαρμογή υλοποίησης Broadcast Encryption - Subset Difference

Για να υλοποιηθεί η τεχνική του Subset Difference έχει δημιουργηθεί από την μεριά του Application Server μια εφαρμογή που κάνει δύο πολύ βασικά πράγματα. Το πρώτο είναι να προετοιμάζει το δυαδικό δέντρο με τις ετικέτες Labels , τις ψευδοετικέτες SubLabels και τα κλειδιά Keys. Το δεύτερο είναι να δέχεται το μήνυμα που πρέπει να σταλεί , όπως επίσης να δέχεται και τους End Nodes που δεν πρέπει να πάρουν το μήνυμα. Στη συνέχεια να δημιουργεί τα κατάλληλα υποσύνολα Subset με τις κατάλληλες ετικέτες και τα κλειδιά που θα χρησιμοποιηθούν στη κεφαλίδα του μηνύματος. Επίσης κάνει τις απαραίτητες κρυπτογραφήσεις και την τμηματοποίηση και τα προωθεί στο Network Server.
Αναλυτικότερα στο πρώτο μέρος μόνο την πρώτη φορά που εγκατασταθεί το σύστημα κάνει τις εξής λειτουργίες.

- Δημιουργεί ένα τέλειο δυαδικό δέντρο ανάλογα με το πλήθος των End Nodes.
- Βάζει σε όλους τους κόμβους τυχαίες ετικέτες των 5 Bytes.
- Για κάθε υποδεντρο που έχει ως ρίζα ένα εσωτερικό κόμβο του δέντρου (όχι τα φύλλα) δημιουργεί τις ψευδοετικέτες Sublabels για όλους τους απογόνους, χρησιμοποιώντας τις συναρτήσεις G_L και G_R.
- Αποθηκεύει το δέντρο με τις ετικέτες του.
- Αποθηκεύει όλα τα υποδέντρα που προκύψαν από το προηγούμενη διαδικασία μαζί με τις ψευδοετικέτες όλων των κόμβων.
- Αποθηκεύει και δημιουργεί ξεχωριστό αρχείο για κάθε φύλο του δυαδικού δέντρου που αντιπροσωπεύει και ένα End Node ή ένα ψεύτικο End Node(προκειμένου το δέντρο να είναι τέλειο). Το αρχείο περιέχει τα υποσύνολα και τις ψευδοετικέτες τις ρίζας που πρέπει να γνωρίζει κάθε End Node προκείμενου να μπορεί να βρει όλα

τα κλείδα για όλα τα υποσύνολα που τον αφορούν. Για κάθε υποσύνολο περιέχει τον αρχικό κόμβο του υποδέντρου - υποσυνόλου που ορίζει τους απογόνους που θα λάβουν το μήνυμα και τον κόμβο απόγονο του υποδέντρου - υποσυνόλου που ορίζει τους απογόνους που θα αποκλειστούν από το μήνυμα. Για κάθε υποσύνολο δίδεται και η ψευδοετικέτα της ρίζας 20 Bytes.

Αφού την πρώτη φορά που θα λειτουργήσει το σύστημα μας εκτελέσουμε το πρώτο μέρος της εφαρμογής και δημιουργήσουμε τα αρχεία που περιγράψαμε, τότε δεν επαναλαμβάνουμε την διαδικασία αυτή ποτέ ξανά.

Παράδειγμα του αρχείου που αποθηκεύει το δυαδικό δέντρο (10 πρώτες εγγραφές)

{"data":"0","isnode":"true","fake":"true","flag":"5","label":"871e77f258","key":"70ab40fec7a34f4fb8357d5dd10d5e6cbd623ff4"}
{"data":"1","isnode":"true","fake":"true","flag":"5","label":"4a819a79c6","key":"b16372517b6f9ec4f7e50b6f11a28332b8ee604b"}
{"data":"2","isnode":"true","fake":"true","flag":"5","label":"c7dbb541c2","key":"ea144922875bdff2b2f8ba890ba0de3969babd3f"}
{"data":"3","isnode":"true","fake":"true","flag":"5","label":"3dd18c1dc5","key":"75cf9f2beb30bdd45a684de382aa25958c5c122e"}
{"data":"4","isnode":"true","fake":"true","flag":"5","label":"40b5e2eb03","key":"837b41eb4e22f32f2b9618ed8c97ec1d3503862e"}
{"data":"5","isnode":"true","fake":"true","flag":"5","label":"b5b0ac6025","key":"b06f3064f5311c88aab02c9d9eadcd63461d3d86"}
{"data":"6","isnode":"true","fake":"true","flag":"5","label":"3db4beb5d5","key":"8f968279a99d71059b27fe9772c9113efc9b89c0"}
{"data":"7","isnode":"true","fake":"true","flag":"5","label":"d29f584556","key":"aebecd33242bfd405034103d5c2d7ab4de5f39a5"}
{"data":"8","isnode":"true","fake":"true","flag":"5","label":"cbf0416d47","key":"e714b67409b761e5e2548b3245f006a5f638f31b"}
{"data":"9","isnode":"true","fake":"true","flag":"5","label":"5f115f388b","key":"523f7ecaae36d85f0d215c03324facb5f72f459b"}

Παράδειγμα αρχείου με τα υποσύνολα που πρέπει να γνωρίζει ένας End Node

{ {0x00,0x00,0x70,0xab,0x40,0xfe,0xc7,0xa3,0x4f,0x4f,0xb8,0x35,0x7d,0x5d,0xd1,0x0d,0x5e,0x6c,0xbd,0x62,0x3f,0xf4},
{0x07,0x0f,0x1f,0xd0,0x8e,0x88,0xa6,0xc5,0x72,0x60,0x9b,0xf5,0x9c,0xce,0xc5,0x3b,0x61,0x3e,0x8d,0xb8,0x90,0x70},
{0x03,0x0f,0xe6,0x77,0x8b,0xf2,0xd5,0x13,0x28,0x34,0x10,0xa9,0xa6,0x50,0x0b,0x07,0xf0,0xfd,0x18,0x02,0x85,0x2c},
{0x03,0x08,0x09,0x00,0x8f,0x08,0x19,0x72,0x00,0x3f,0x1e,0x32,0x50,0xb3,0x3b,0x4a,0xab,0xae,0x26,0xf0,0x27,0xf9},
{0x01,0x0f,0xa8,0x97,0xb3,0xfe,0x0b,0x5b,0xfd,0x9e,0x4a,0x85,0xfa,0x5c,0x88,0x0a,0xf3,0xa2,0xb2,0x39,0x35,0xb3},
{0x01,0x08,0x30,0xc7,0xf1,0x2c,0x7c,0xf4,0xf6,0x6f,0xe9,0x2a,0x63,0x27,0xa4,0x3e,0x50,0x33,0x7a,0x0c,0xae,0xf2},
{0x01,0x04,0xca,0x45,0xb8,0xcc,0xaa,0x90,0x6f,0x80,0xab,0x60,0xff,0x67,0xc4,0x59,0x0c,0x92,0xcd,0xfb,0xe5,0x25},
{0x00,0x0f,0xb8,0x41,0x3d,0x52,0x43,0xb3,0xd6,0x52,0xd5,0xce,0x6f,0x2f,0x3f,0x17,0xdd,0x84,0xd3,0xfa,0x38,0x84},
{0x00,0x08,0x75,0x46,0xc4,0x9b,0x09,0x53,0xb2,0xdf,0x3f,0x43,0xb0,0x69,0x7f,0xbb,0x0a,0xfe,0x7c,0xb5,0xf4,0xbc},
{0x00,0x04,0xf8,0x21,0x8d,0x79,0x4d,0x12,0xe4,0xe8,0x5d,0xfa,0xd7,0x02,0x1f,0xc0,0x22,0xa1,0xf2,0x0e,0x68,0x49},
{0x00,0x02,0x0b,0xd3,0xa0,0x32,0x17,0x10,0x27,0xb9,0xf2,0xb2,0xf0,0xbf,0x7a,0x74,0xb9,0x01,0x53,0x3b,0xf2,0xff} }//

Το δεύτερο μέρος της εφαρμογής στο ξεκίνημα της διαβάζει τα αρχεία που έχουν δημιουργηθεί από το πρώτο μέρος. Για να μπορεί να επικοινωνεί με το περιβάλλον που διαχειρήζετε ο χρήστης τους κόμβους , ενεργοποιεί ένα Web Socket Server, ώστε να μπορεί να δέχεται μέσω http Request το μήνυμα και τους χρήστες που θα πρέπει να λάβουν το μήνυμα. Αφού στείλουμε μια http Request με τις παραπάνω πληροφορίες γίνονται οι παρακάτω διαδικασίες. Κάθε φορά που πρέπει να στείλουμε ένα μήνυμα κάνει τις εξής διαδικασίες.

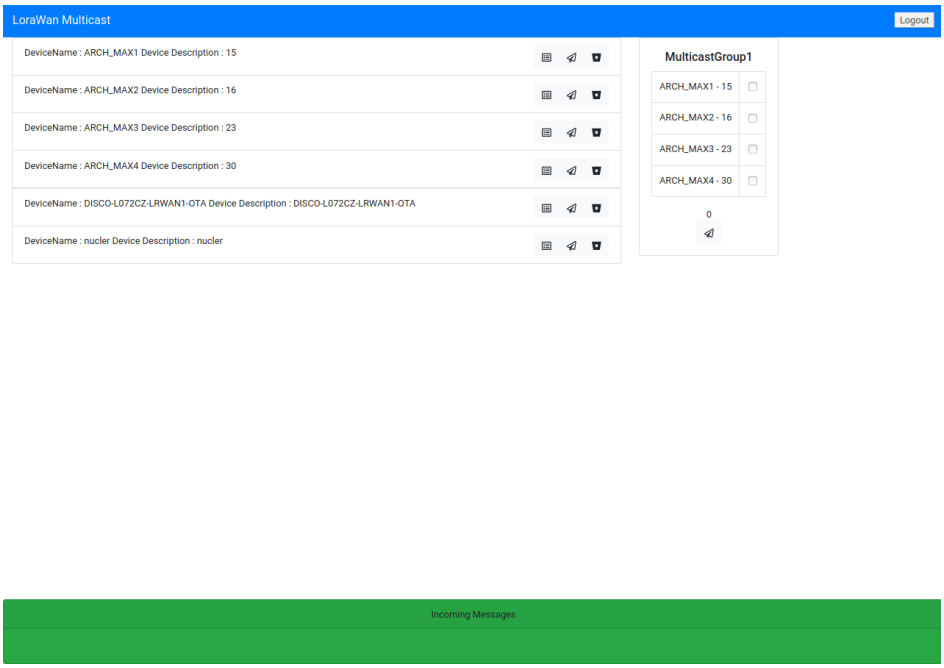
- Δέχεται σε ένα πίνακα τους κόμβους που πρέπει να στείλει το μήνυμα και φτιάχνει ένα πίνακα με τους κόμβους που πρέπει να αποκλείσει.
- Δέχεται το μήνυμα που πρέπει να σταλεί.
- Δημιουργεί τα υποσύνολα και τις ψευδοετικέτες που θα χρησιμοποιηθούν στην κεφαλίδα και στις κρυπτογραφήσεις.
- Δημιουργεί ένα τυχαίο κλειδί 5 Bytes και κρυπτογραφει το μήνυμα με τον αλ-

γόριθμο RC4.

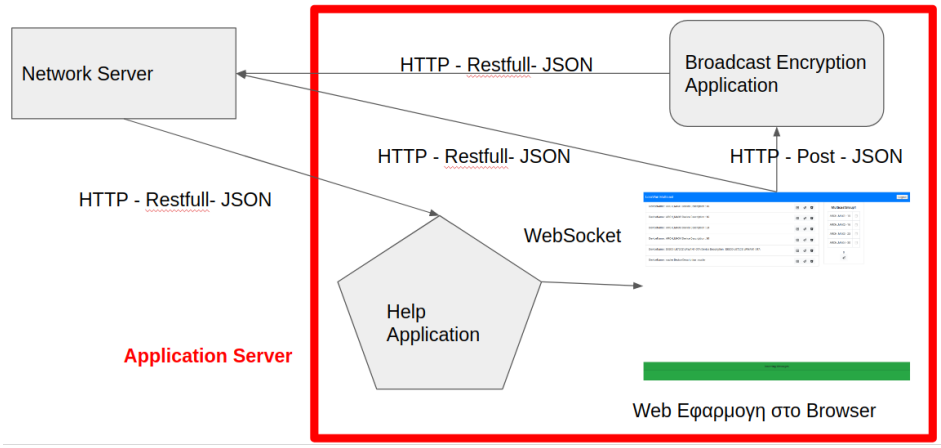
- Για κάθε υποσύνολο που έχει δημιουργήσει παίρνει την ψευδοετικέτα και με την συνάρτηση G_M η οποία είναι μια μονοδρομη συνάρτηση (HMAC SHA1) δημιουργεί το κλειδί με το οποίο κρυπτογραφει το κλειδί που κρυπτογραφησε το μήνυμα.
- Βάζει σε ένα Buffer τους κόμβους των υποσυνόλων , τα κρυπτογραφημένα κλειδιά και το κρυπτογραφημένο μήνυμα.
- Ελέγχει και αν το μήνυμα με την κεφαλίδα είναι μεγαλύτερο από το επιτρεπτό μέγεθος μηνύματος LoRaWAN σύμφωνα με το Spread Factor που χρησιμοποιούμε και αν χρειαστεί τμηματοποιεί το μήνυμα σε μικρότερα τμήματα σύμφωνα με την διαδικασία Framing που έχουμε περιγράψει.
- Σε κάθε τμήμα βάζει τα κατάλληλα Bytes μπροστά και προωθεί κάθε τμήμα στον Network Server

Εφαρμογή Ανταλλαγής Μηνυμάτων στον Application Server

Για την από την μεριά του Application Server αναπτύχθηκε μια Web Εφαρμογή η οποία εμφανίζει μια λίστα με όλους τους End Nodes που έχουμε στο LoRaWAN δίκτυο μας . Μας δίνει την δυνατότητα να δούμε κάποιες πληροφορίες για αυτούς και να επιλέξουμε σε ποιον θέλουμε να στείλουμε ένα Unicast μήνυμα. Ανοίγει ένα παράθυρο για να γράψουμε το μήνυμα και στη συνέχεια πατώντας το κουμπί της αποστολής να το στείλουμε. Στην ίδια οθόνη μπορούμε να δούμε και όλα τα μηνύματα που έρχονται από κάποιο end Node, αυτά εμφανίζονται αυτόματα την στιγμή που λαμβάνονται από το Network ServerΣ. Αυτό γίνεται μέσο μια εφαρμογής φτιάξαμε στο Virtual ServerΣ που δημιουργεί μια WebSocket σύνδεση με την εφαρμογή μηνυμάτων , δέχεται μέσω http post request από το Network Server τα εισερχόμενα μηνύματα. Στη συναίχεια προδοθεί σε πραγματικό χρόνο μέσω της Websocket σύνδεσης, τα μηνύματα στέλνονται από τα End Nodes στην εφαρμογή ανταλλαγής μηνυμάτων. Η εφαρμογή μας επικοινωνεί με τον Network ServerΣ μέσω RestFul Κλήσεων , ο LoRaServer υποστηρίζει ένα Rest API με το οποίο δίνεται η δυνατότητα μέσω RestFul Κλήσεων να πάρουμε πληροφορίες για τους End Node ,τα Gateway , τα Organization που έχουμε δημιουργήσει. Επίσης μέσω RestFul Κλήσεων μπορούμε να στείλουμε στο Network Server τα μηνύματα που θέλουμε να προωθήσει στα Gateway. Η Εφαρμογή έχει υλοποιηθεί σε React.



Σχήμα 6.24: Οθόνη μηνυμάτων Application Server



Σχήμα 6.25: Αρχιτεκτονική Application Server

Κεφάλαιο 7

Συμπεράσματα και μελλοντικές επεκτάσεις

Το σύστημα που προτείνουμε δεν είναι μια ολοκληρωμένη εφαρμογή επικοινωνιών για περιπτώσεις κρίσεων αλλά ο αρχικός σχεδιασμός ενός τέτοιου συστήματος που θα μπορούσε να αναπτυχθεί και να αποτελέσει ένα ολοκληρωμένο σύστημα επικοινωνιών. Αρχικά προσπαθήσαμε να φανταστούμε το πρόβλημα και τα ερωτήματα που πρέπει να απαντήσουμε και να βρούμε λύσεις. Μελετήσαμε διάφορες τεχνολογίες δικτύων για να βρούμε την καταλληλότερη για το πρόβλημά μας. Αφού καταλήξαμε στην τεχνολογία LoRa και το δίκτυο LoRaWan κάναμε επισταμένη μελέτη για την δομή και το τρόπο λειτουργίας του. Κατά την διάρκεια του σχεδιασμού του συστήματος μας, προβληματιστήκαμε λόγο του πιθανού μεγάλου πλήθους End node που θα υπάρχουν και της ανάγκης για άμεση αποστολή μηνυμάτων σε πολλούς ταυτόχρονα, έτσι δημιουργήθηκε η υποχρέωση για αποστολή μηνυμάτων μέσω multicast μεταδόσεων. Αυτό μας ανάγκασε να μελετήσουμε το σύστημα των multicast μεταδόσεων που χρησιμοποιεί το δίκτυο LoRaWan. Προσπαθώντας να προβλέψουμε τις απαιτήσεις του συστήματος σε πραγματικές συνθήκες αντιληφθήκαμε ότι πιθανών τα μηνύματα που πρέπει να σταλούν θα είναι σε διαφορετικούς End node κάθε φορά και όχι σε ομάδες End node όπως αρχικά είχαμε υποθέσει. Γιαυτό το λόγο και αναγκαστήκαμε να ψάξουμε και να βρούμε ένα πιο έξυπνο τρόπο για να κάνουμε τις multicast μεταδόσεις. Η τεχνικές του Broadcast Encryption φαίνεται να μπορούν να λύσουν το πρόβλημα που αντιμετωπίσαμε. Σε αυτή λοιπόν την εργασία παντρέψαμε δύο πράγματα την τεχνολογία και το δίκτυο LoRaWan και την τεχνική Subset Difference από τις τεχνικές του Broadcast Encryption. Έπειτα στήσαμε ένα δίκτυο LoRaWan με όλα τα συστατικά του μέρη Gateway, Network Server,Application Server και End nodes. Εγκαταστήσαμε το απαραίτητο Software , επιλέξαμε , δοκιμάσαμε και συνθέσαμε το κατάλληλο Hardware. Αναπτύξαμε εφαρμογή που θα δείχνει το τρόπο επικοινωνίας των συσκευών Android με το δίκτυο LoRaWan. Υλοποιήσαμε τα κατάλληλα προγράμματα που θα εκτελούνται στους μικροελεγκτές ,θα διαχειρίζονται τις περιφερειακές συσκευές και θα υλοποιούν τις συνδέσεις για την αποστολή και λήψη μηνυμάτων μέσω του LoRaWan. Πιο απαιτητικό ήταν η υλοποιήση του Broadcast Encryption στο μικροελεγκτή. Τέλος υλοποιήσαμε μια σειρά εφαρμογών,(τρεις για την ακρίβεια) οι οποίες παίζουν το ρόλο ενός πολύ βασικού Application Server. Μέσο αυτών των εφαρμογών μπορούμε να κάνουμε κάποιες βασικές ενέργειες όπως αποστολή και λήψη μηνυμάτων αλλά και όλες τις απαραίτητες ενέργειες για το Broadcast Encryption.

Γενικά μέχρι αυτή τη στιγμή η βιβλιογραφία είναι περιορισμένη σε ότι αφορά τις multicast μεταδόσεις και τα δίκτυα LoRaWan. Από την άλλη φαίνεται να υπάρχουν μια μεγάλη σειρά προβλημάτων που απαιτούν multicast μεταδόσεις και που τα δίκτυα LoRaWan είναι ιδανικά δίκτυα μεταφοράς πληροφοριών. Τα δίκτυα LoRaWan είναι ιδανικά για IoT συσκευές και υπάρχουν ένα μεγάλο πλήθος εφαρμογών με IoT συσκευές που απαιτούν ταυτόχρονες ενημερώσεις μεγάλου πλήθους συσκευών. Οι συσκευές όμως μπορεί να είναι διαφορετικές για κάθε ενημέρωση οι λύση που προτείνουμε έρχεται και ταιριάζει πολύ καλά με αυτά τα προβλήματα. Εμείς χρησιμοποιήσαμε την τεχνική Subset Difference την πιο βασική τεχνική με συμμετρική κρυπτογράφηση Broadcast Encryption που υπάρχει. Βεβαίως υπάρχουν και άλλες πιο αποδοτικές μέθοδοι συμμετρικής κρυπτογράφησης που μπορούν να χρησιμοποιηθούν κατά περίπτωση. Επίσης θα ήταν πολύ πιο αποδοτικό να χρησιμοποιηθούν μέθοδοι ασύμμετρης κρυπτογράφησης με ένα δημόσιο κλειδί και πολλαπλά ιδιωτικά κλειδιά Public Key Broadcast Encryption.

Παράρτημα Α΄

Ακρωνύμια και συντομογραφίες

LAN Local Area Network

RTOS Real Time Operating System

Iot Internet of Things

IDE Interface Development Environment

HAL Hardware Abstraction Layer

API Application Programming Interface

MQTT MQ Telemetry Transport

M2M Machine to Machine

LPWA Low Power Wide Area

NB-IoT Narrowband IoT

LTE Long Term Evolution

ETSI European Telecommunications Standards Institute

LoRa Long Range

LoRaWAN Long Range Wide Area Network

FSK Frequency Shift Keying modulation technique

SF Spread Factor

DR Data Rate

BW BandWidth

TX Pow Transmission Power

IP Internet Protocol

CR Coding Rate

CRC Cyclic Redundancy Check

PHDR Physical Header

PHDR_CRC Physical Header Cyclic Redundancy Check

MHDR Mac Header

MAC Medium Access Contro

MIC Message Integrity Code

RFU Reserved for Future Usage

FHDR Frame Header

EN End Node

GW GateWay

NS Network Server

OTAA Over The Air Activation

ABR Activation By Personalization

DevAddr Device Address

AppEUI Application identifier

NwkSKey Network Session Key

AppSKey Application Session Key

DevEUI End-device identifier

AppKey Application Key

McKey Multicast Key

McNwkSKey Multicast Network Session Key

McAppSKey Multicast Application Session Key

SD The Subset Difference Method

VS Virtual Server

gRPC google Remote Procedures Calls

Bibliography

[1] "5 compelling cases of lpwan for process industries." [Online]. Available: https://www.processingmagazine.com/lpwan-for-process-industries/

[2] "Lpwan, lora, lorawan and the internet of things." [Online]. Available: https://medium.com/coinmonks/lpwan-lora-lorawan-and-the-internet-of-things-aed7d5975d5d

[3] "Unicast vs multicast vs broadcast: What are the differences?" [Online]. Available: http://www.fiber-optical-networking.com/unicast-vs-multicast-vs-broadcast-differences.html

[4] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Annual International Cryptology Conference*. Springer, 2001, pp. 41–62.

[5] H. Lee, S.-H. Chung, Y.-S. Lee, and Y. Ha, "Performance comparison of dash7 and iso/iec 18000-7 for fast tag collection with an enhanced csma/ca protocol," in *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing*. IEEE, 2013, pp. 769–776.

[6] P. Tuset-Peiró, A. Anglès-Vazquez, J. López-Vicario, and X. Vilajosana-Guillén, "On the suitability of the 433 mhz band for m2m low-power wireless communications: propagation aspects," *Transactions on Emerging Telecommunications Technologies*, vol. 25, no. 12, pp. 1154–1168, 2014.

[7] "Qualcomm, inc.: The evolution of mobile technologies: 1g 2g 3g 4g lte." [Online]. Available: https://www.qualcomm.com/media/documents/files/the-evolution-of-mobile-technologies-1g-to-2g-to-3g-to-4g-lte.pdf

[8] "International journal of advanced science and technology vol.108 (2017), pp.1-10." [Online]. Available: http://dx.doi.org/10.14257/ijast.2017.108.01

[9] "Comparison: 3g wireless networks with 4g wireless networks technology wise." [Online]. Available: http://article.nadiapub.com/IJAST/vol108/1.pdf

[10] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[11] "Know the difference between nb-iot vs. cat-m1 for your massive iot deployment." [Online]. Available: https://www.ericsson.com/en/blog/2019/2/difference-between-NB-IoT-CaT-M1

[12] “Link labs, inc.: An overview of narrowband iot (nb-iot).” [Online]. Available: https://www.link-labs.com/blog/overview-of-narrowband-iot,Jul.27,2016

[13] “Sigfox s.a.: Sigfox – the global communications service provider for the internet of things (iot).” [Online]. Available: https://www.sigfox.com

[14] “Igenus.” [Online]. Available: https://www.ingenu.com

[15] “How rpma works.”

[16] “Dash7 alliance.” [Online]. Available: https://dash7-alliance.org

[17] M. Weyn, G. Ergeerts, R. Berkvens, B. Wojciechowski, and Y. Tabakov, “Dash7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication,” 10 2015.

[18] “Lora alliance: Lora alliance tm – technology.” [Online]. Available: https://www.lora-alliance.org/what-is-lora

[19] “An1200.22 lora tm modulation basics semtech corporation.” [Online]. Available: https://www.semtech.com/uploads/documents/an1200.22.pdf

[20] A. Springer, W. Gugler, M. Huemer, L. Reindl, C. Ruppel, and R. Weigel, “Spread spectrum communications using chirp signals,” in *IEEE/AFCEA EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security (Cat. No. 00EX405)*. IEEE, 2000, pp. 166–170.

[21] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, “Known and unknown facts of lora: Experiences from a large-scale measurement study,” *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 2, p. 16, 2019.

[22] “Sx1272/73 datasheet semtech corporation.” [Online]. Available: https://www.semtech.com/uploads/documents/SX1272_DS_V4.pdf

[23] E. Ruano Lin, “Lora protocol. evaluations, limitations and practical test,” 2016.

[24] I. LoRa Alliance, “Lorawan 1.1ra regional parameters,” 2017. [Online]. Available: https://net868.ru/assets/pdf/LoRaWAN-Regional-Parameters-v1.1rA.PDF

[25] E. T. S. Institute., “Etsi en 300 220-2 v3.1.1,” 2016. [Online]. Available: https://www.etsi.org/deliver/etsi_en/300200_300299/30022002/03.01.01_30/en_30022002v030101v.pdf

[26] “Eett.” [Online]. Available: https://www.eett.gr/opencms/export/sites/default/EETT/Electronic_Communications/Radio_Communications/TelecommunicationsEquipment/105v2.pdf

[27] C. Perkins, D. Stanley, W. Kumari, and J. Zuniga, “Multicast considerations over ieee 802 wireless media,” *Internet Draft*, 2017.

[28] I. LoRa Alliance, “Lorawan remote multicast setup specification v1.0.0,” 2018. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-09/remote_multicast_setup_v1.0.0.pdf

[29] ——, “Lorawan® specification v1.0.3,” 2018. [Online]. Available: https://lora-alliance.org/sites/default/files/2018-07/lorawan1.0.3.pdf

[30] A. Fiat and M. Naor, "Broadcast encryption," in *Annual International Cryptology Conference*. Springer, 1993, pp. 480–491.

[31] R. V. Mahammad Salmasultana, "A secure symmetric key broadcast encryption (skbe) for sharingdata over dynamic group members," *International Journal of Advanced Research in Computer Science and Software Engineering*, 2016. [Online]. Available: http://ijarcsse.com/Before_August_2017/docs/papers/Volume_6/10_October2016/V6I10-0120.pdf

[32] O. Goldreich, S. Goldwasser, and S. Micali, "How to construct random functions," *Journal of the ACM (JACM)*, vol. 33, no. 4, pp. 792–807, 1986.

[33] "Mbed os 5." [Online]. Available: https://os.mbed.com/

[34] "An introduction to arm mbed os 5." [Online]. Available: https://os.mbed.com/docs/mbed-os/v5.13/introduction/index.html

[35] "React - facebook inc." [Online]. Available: https://reactjs.org

[36] "React native - facebook inc." [Online]. Available: https://facebook.github.io/react-native/

[37] "Nodejs nodejs foundation." [Online]. Available: https://nodejs.org/

[38] "Mqtt." [Online]. Available: https:/http://mqtt.org/

[39] "Influxdb." [Online]. Available: https://www.influxdata.com/products/influxdb-overview/

[40] I. LoRa Alliance, "Lorawan® fragmented data block transport specification v1.0.0," 2018. [Online]. Available: https://lora-alliance.org/resource-hub/lorawanr-fragmented-data-block-transport-specification-v100

[41] "Scaleway." [Online]. Available: https://www.scaleway.com/

[42] "Loraserver." [Online]. Available: https://www.loraserver.io/

[43] "Eclipse mosquitto." [Online]. Available: https://mosquitto.org

[44] "Raspberry pi 3 model b." [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[45] "ic880a - lorawan® concentrator 868mhz." [Online]. Available: https://wireless-solutions.de/products/radiomodules/ic880a.html

[46] I. GmbH, "Wimod ic880a datasheet." [Online]. Available: https://webshop.ideetron.nl/Files/3/1000/1211/Attachments/Product/IB4c6A1J5Uh6Ej5D3i6cQ88q1P2D1404.pdf

[47] "Lora net." [Online]. Available: https://github.com/Lora-net

[48] STMicroelectronics, "Nucleo-l073rz data brief." [Online]. Available: https://www.st.com/resource/en/data_brief/nucleo-l073rz.pdf

[49] S. Corporation, "Sx1272mb2das hardware description." [Online]. Available: https://media.digikey.com/pdf/Data%20Sheets/Semtech%20PDFs/SX1272MB2DAS_HM.PDF

[50] STMicroelectronics, “B-l072z-lrwan1 data brief.” [Online]. Available: https://www.st.com/resource/en/data_brief/b-l072z-lrwan1.pdf

[51] “Stm32f407vet6.” [Online]. Available: https://www.st.com/en/microcontrollers-microprocessors/stm32f407ve.html#quality-reliability

[52] STMicroelectronics, “Stm32f405xx.” [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f407ve.pdf

[53] L. HOPE MICROELECTRONICS CO., “Rfm95w/96w/98w.” [Online]. Available: https://www.hoperf.com/data/upload/portal/20190801/RFM95W-V2.0.pdf

[54] ILITEK, “Ili9341 a-si tft lcd single chip driver 240rgbx320 resolution and 262k color version: V1.02 document no.: Ili9341_ds_v1.02.pdf.” [Online]. Available: https://cdn-shop.adafruit.com/datasheets/ILI9341.pdf

[55] I. Sumartono, A. P. U. Siahaan, and N. Mayasari, “An overview of the rc4 algorithm,” *IOSR Journal of Dental and Medical Sciences*, vol. 18, pp. 2278–661, 12 2016.

[56] R. C. I. H. Krawczyk IBM, M. Bellare UCSD, “Hmac: Keyed-hashing for message authentication.” [Online]. Available: https://tools.ietf.org/pdf/rfc2104.pdf