

Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες (Δικτυακός Προγραμματισμός)

Ανδρουλακάκης Ιωάννης 1115201400012

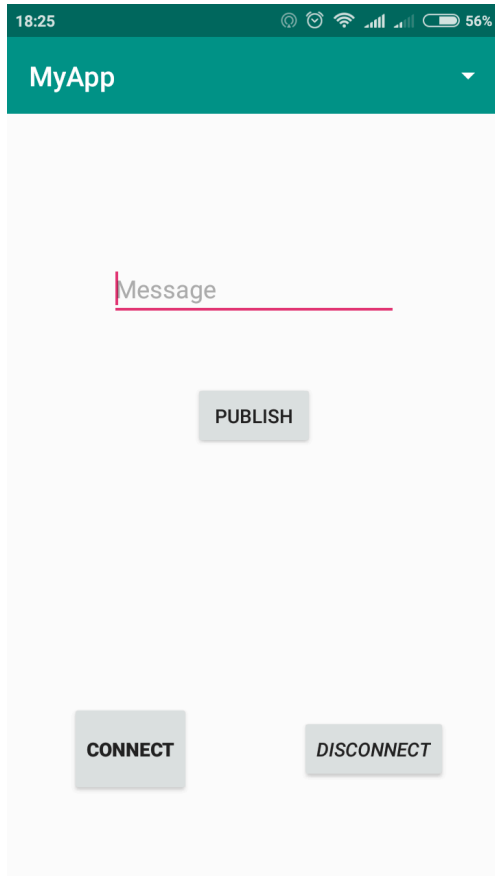
Ματάνης Παναγιώτης 1115201400297

Σακελλαρίδης Ιωάννης 1115201400307

Τραχανάς Γεώργιος 1115201300178

Android Studio

Η εφαρμογή android αναπτύχθηκε σε Android Studio.



Αριστερά φαίνεται το layout του Main Activity.

Αποτελείται από 3 buttons, 1 edit text για input και ένα spinner που φαίνεται από την επόμενη εικόνα.

Το κουμπί **Connect** συνδέει το τερματικό με τον Edge Server και κάνει subscribe σε ένα topic. Αντίστοιχα με το **Disconnect** αποσυνδέεται και κάνει unsubscribe.

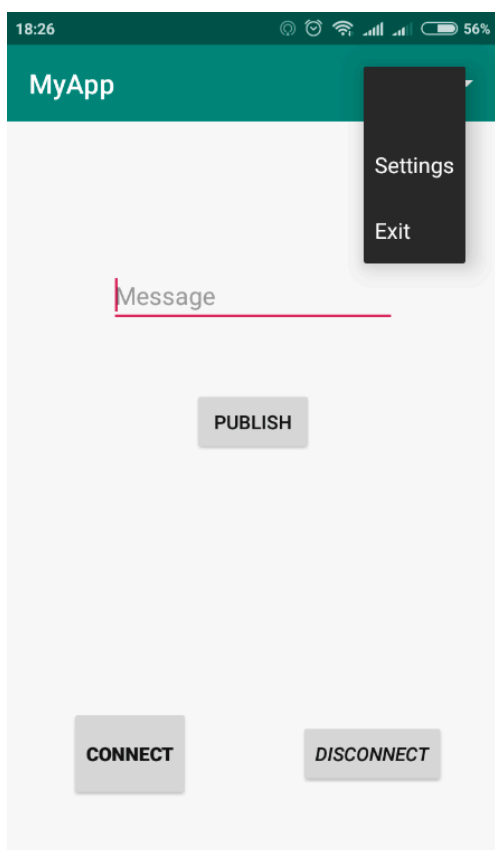
Το κουμπί **Publish** είναι συνδεδεμένο με το Text ώστε να στέλνει στον Edge το περιεχόμενο του.

Στο **background** τρέχει ένα thread το οποίο :

- Ελέγχει τη σύνδεση στο Wi-Fi.
- Κάνει Publish το αναγνωριστικό του εκάστοτε android, το στίγμα του GPS και τις συντεταγμένες από το accelerometer.
- Κάνει Publish τυχαία csv αρχεία που περιέχονται στον φάκελο Test_Set ο οποίος είναι αποθηκευμένος τοπικά στην sd card του τερματικού.

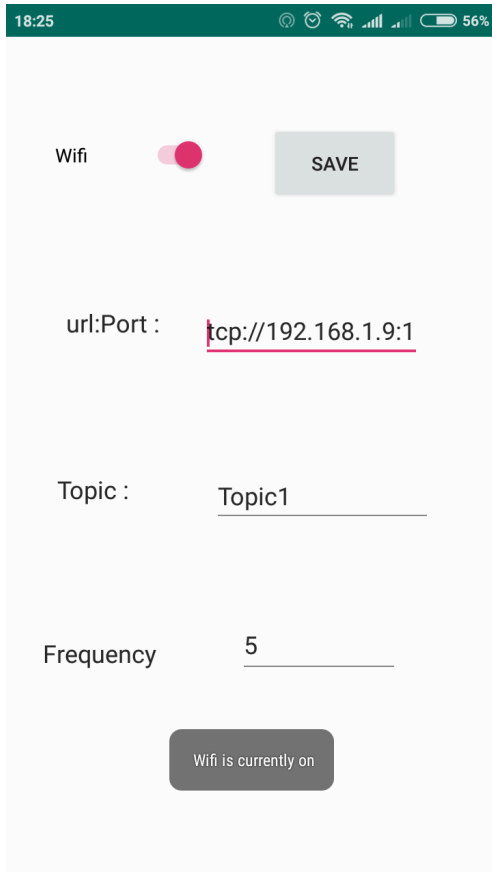
Η παραπάνω διαδικασίες επαναλαμβάνονται σε χρόνο frequency ο οποίος μπορεί να παραμετροποιηθεί στην συνέχεια από τις ρυθμίσεις.

Ακόμα έχουν υλοποιηθεί συναρτήσεις Ringtone και Flash για τα σύματα κινδύνου που θα έρχονται από τον Edge.



Δίπλα φαίνεται το **spinner** το οποίο περιέχει τις Ρυθμίσεις και το κουμπί Exit για έξοδο απ' την εφαρμογή.

Παρακάτω φαίνεται το layout του Activity Settings.



Αποτελείται από ένα **Switch** για ενεργοποίηση/απενεργοποίηση του WiFi,

3 text edits για παραμετροποίηση:

- των url & Port που βοηθάει στην εναλλαγή των wlans
- topic που κάνει subscribe και publish το τερματικό
- frequency που ορίζει την συχνότητα που τρέχει το thread και στέλνονται τα αρχεία csv

Τέλος ένα κουμπί **Save** το οποίο επιστρέφει τις μεταβλητές σωστά παραμετροποιημένες.

Edge Server

Ο **Edge Server** υλοποιεί τον **Mosquitto Broker**. Αποτελείται από 4 threads (ένα ζευγάρι **sender** και **receiver** για κάθε **topic**) τα οποία τρέχουν συνέχεια και είναι υπεύθυνα για την επικοινωνία με το android και την διαχείριση των topic.

Ο **receiver** επίσης δημιουργεί locally τα csv που του στέλνουν τα android τερματικά και τα αποθηκεύει.

Τέλος έχει ακόμα έναν receiver ο οποίος αναφέρεται στον backhaul server και λειτουργεί μέσω web socket.

Αυτός ανοίγει ένα socket και περιμένει να λάβει το training set από τον backhaul στο οποίο στην συνέχεια της εργασίας θα εφαρμόσει classification.

BackhaulServer

Ο φάκελος **BackhaulServer** περιλαμβάνει 5 java αρχεία που υλοποιούν τα ζητούμενα του 1ου παραδοτέου. Πιο συγκεκριμένα:

listFile.java: Για το σύνολο των αρχείων που περιέχει ένας φάκελος(το όνομα του οποίου δίνεται ως όρισμα), επιστρέφεται μία List<String> που περιλαμβάνει τις διευθύνσεις των αρχείων που είναι της μορφής csv.

```
public class listFile {  
  
    //input:    a folder with Files such as CSV  
    //output:   list of strings with the paths of each CSV file  
  
    //default constructor  
    public listFile(){}  
  
    public List<String> listFilesForFolder(final File folder) {  
        List<String> filenames = new ArrayList<>();  
        for (final File fileEntry : folder.listFiles()) {  
            //  
                if (fileEntry.getName().contains(".csv"))  
                    filenames.add(fileEntry.getAbsolutePath());  
            }  
        return filenames;  
    }  
}
```

readingCSV.java: Περιλαμβάνει την συνάρτηση createCSV(String file).Για κάθε αρχείο csv από το δοσμένο φάκελο, γίνεται επεξεργασία των στοιχείων και επιστρέφεται ένα String με το όνομα του πειράματος και τις 14 τιμές για τις εντροπίες/feature vector. Για τον υπολογισμό της εντροπίας για κάθε ένα από 14 διαφορετικά κανάλια, αξιοποιείται η συνάρτηση που περιλαμβάνεται στο αρχείο Entropy.java.

readCSVFolder.java: Είναι το αρχείο που περιλαμβάνει τη main για τον Backhaul Server. Υλοποιούνται 2 λειτουργίες σε αυτό το κομμάτι:

- 1)Για κάθε αρχείο του φακέλου Training Set, υπολογίζεται η εντροπία του και αποθηκεύεται σε ένα νέο αρχείο csv. Έτσι,δημιουργείται το ζητούμενο training set.
- 2)Το ζητούμενο training set αποστέλλεται μέσω socket στον Edge Server.

```

public static void main(String args[]){
    try {
        readCSVFolder fold = new readCSVFolder();
        readingCSV read = new readingCSV();
        FileWriter writer = new FileWriter(FILE_NAME);
        File folder = new File( pathname: "/Users/georgetrachanas/Desktop/Training_Set");

        listFile listOfFiles = new listFile();

        List<String> filenames = listOfFiles.listFilesForFolder(folder);

        for (int i = 0; i < filenames.size(); i++) {
            writer.append(read.createCSV(filenames.get(i)));
            writer.append("\n");
        }
        System.out.println("Training set is ready!");
        System.out.println("Backhaul server is ready to send the training set to the Edge Server!");
        writer.close();

        fold.sendFile();
    } catch (Exception ex){
        System.out.println("Exception occured!");
        ex.printStackTrace();
    }
}

```

```

public void sendFile(){
    try {
        receiver = new ServerSocket( port: 9099);
        socket = receiver.accept();
        System.out.println("Accepted connection from : " + socket);
        FileInputStream fis = new FileInputStream(FILE_NAME);

        BufferedInputStream in = new BufferedInputStream(fis);
        in.read(buffer, off: 0, buffer.length);
        out = socket.getOutputStream();

        System.out.println("Sending files");
        out.write(buffer, off: 0, buffer.length);
        out.flush();

        /*while ((count = in.read(buffer)) > 0){
            out.write(buffer,0,count);
            out.flush();
        }*/

        out.close();
        in.close();
        socket.close();
        System.out.println("Finished sending");
    } catch (IOException ioe){
        System.out.println("Something went wrong with the connection!");
        ioe.printStackTrace();
    }
}

```

MySQLDatabase

Ο φάκελος περιλαμβάνει το αρχείο **mysql.java** που υλοποιεί τα ζητούμενα. Πιο συγκεκριμένα, δημιουργείται η βάση δεδομένων με τα ζητούμενα πεδία και υπάρχει σύνδεση μέσω JDBC Driver με τον Backhaul Server.

```

DROP TABLE IF EXISTS `Data`;
/*!40101 SET @saved_cs_client      = @@character_set_client */;
SET character_set_client = utf8mb4 ;
CREATE TABLE `Data` (
  `androidIdentifier` int(11) NOT NULL,
  `timeStamp` timestamp NOT NULL,
  `gpsSignal` double NOT NULL,
  `criticalityLevel` double NOT NULL,
  PRIMARY KEY (`androidIdentifier`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
/*!40101 SET character_set_client = @saved_cs_client */;

```