Κωνσταντίνος Τσιάρας ΑΜ: 1115201500165

Αθηνά Χατζοπούλου ΑΜ: 1115201500175

**Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες**

**Project – Φάση 1η**

**Edge Server (ES):**

Το κομμάτι του Java project επιτελεί 3 διαφορετικές λειτουργίες, οι οποίες ενεργοποιούνται μέσω κατάλληλων flags από τη γραμμή εντολών:

* XML to CSV conversion ( *-c* )

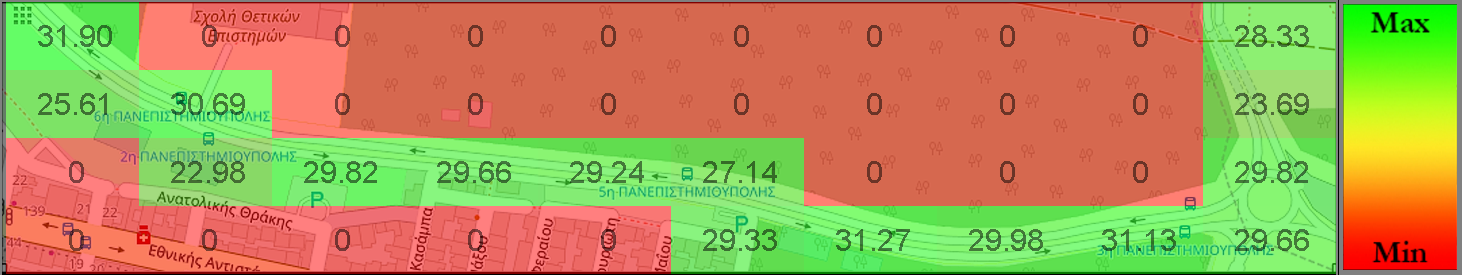
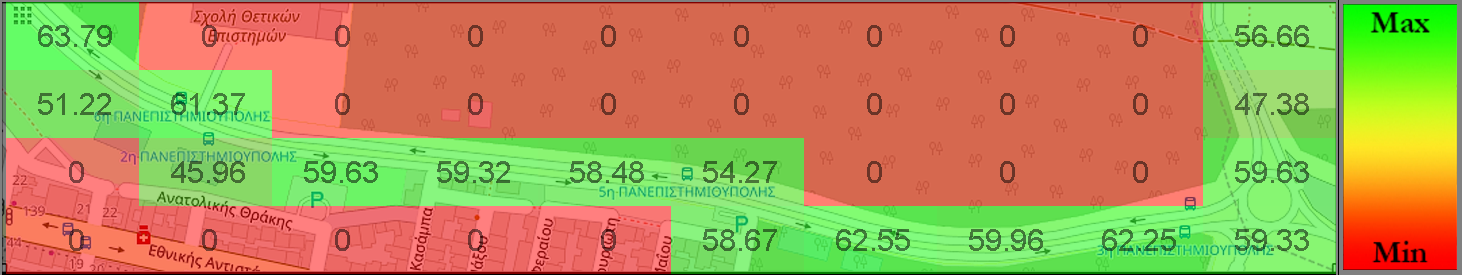
Η λειτουργία αυτή, όπως περιγράφεται στην εκφώνηση, διαβάζει από τα SUMO xml αρχεία για κάθε datapoint τα “timestep, id, lat, long, angle, speed”, δημιουργεί “RSSI, throughput” με τον τρόπο που περιγράφεται στην εκφώνηση και τα γράφει σε καινούργια αρχεία csv, αντίστοιχα για κάθε xml.

* Heatmap generation ( *-g* )

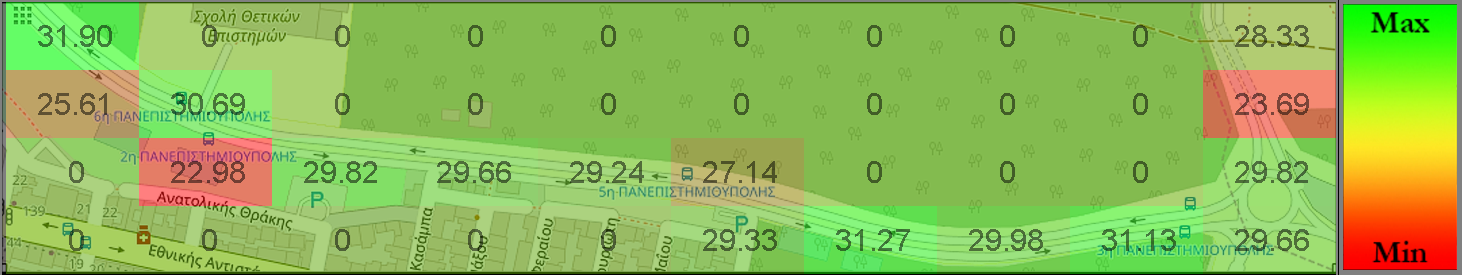
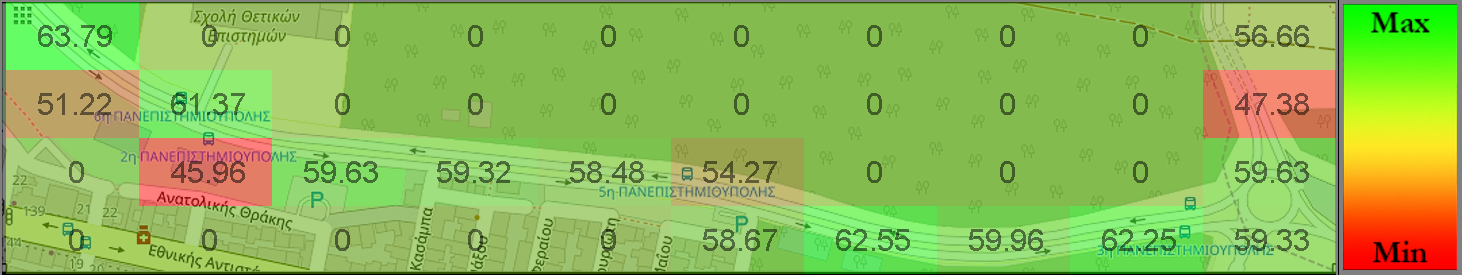
Εδώ αρχικά διαβάζονται τα αρχεία csv που δημιουργήθηκαν παραπάνω προκειμένου να δημιουργηθούν 2 δισδιάστατοι πίνακες (*rssiCellMap* και *throughputCellMap*) με διαστάσεις όσες και του ζητούμενου grid επί του δοθέντος χάρτη *Map.jpg* (default 4x10). Το κάθε κελί αυτών περιέχει το μέσο όρο των τιμών – RSSI και throughput αντίστοιχα – που βρίσκονται στις συντεταγμένες που αναλογούν στο εκάστοτε κελί.

Στη συνέχεια, για καθένα από τους 2 αυτούς πίνακες κατασκευάζεται από 1 heatmap με χρήση της βιβλιοθήκης [JHeatChart](http://www.javaheatmap.com/). Το κάθε heatmap τοποθετείται πάνω από την εικόνα του χάρτη (overlay), με τις τιμές του κάθε κελιού να αναγράφονται στο εσωτερικό τους.

Ένα πρόβλημα που συναντήσαμε στο κομμάτι των heatmaps ήταν το ότι για πολλά κελία, συγκεκριμένα όσα δεν αντιστοιχούσαν σε δρόμους, όπως ήταν λογικό δεν υπήρχαν διαθέσιμες τιμές στα δεδομένα μας και συνεπώς ο μέσος όρος των RSSI/throughput για αυτά ήταν 0. Αυτό όμως είχε σαν αποτέλεσμα η JHeatChart να χρωματίζει αυτά κόκκινα και όλα τα υπόλοιπα – οι διακυμάνσεις μεταξύ των οποίων είναι προφανώς μικρές – να τα χρωματίζει όλα πράσινα. Προκειμένου να το αντιμετωπίσουμε αυτό σκεφτήκαμε να υπολογίσουμε το μέσο όρο όλων των τιμών του κάθε πίνακα και να περάσουμε αυτή την τιμή αντί για 0 στη JHeatChart στα κελιά των οποίων η τιμή είναι 0 με σκοπό αυτά να χρωματιστούν με το πιο «ουδέτερο» χρώμα, μεταξύ κόκκινου και πράσινου. Η λειτουργικότητα αυτή ενεργοποιείται μέσω του πεδίου *Heatmap.normalizeFlag* και τα διαφορετικά αποτελέσματα για ενδεικτικά csv φαίνονται παρακάτω:



Heatmaps RSSI (πάνω) και throughput (κάτω) με *normalizeFlag=false*



Heatmaps RSSI (πάνω) και throughput (κάτω) με *normalizeFlag=true*

* Vehicle MQTT topics subscription ( *-s* )

Εδώ απλά γίνεται subscribe στα MQTT topics στα οποία τα android terminals στέλνουν τα datapoints τους. Κάθε datapoint που φτάνει στον Edge Server εκτυπώνεται στο stdout. Πατώντας “Enter” ο ES αποσυνδέεται από τα topics αυτά και τερματίζει.

**Android Terminals:**

Η ταυτότητα του terminal, αν δηλαδή διαβάζει το *vehicle\_26.csv* ή το *vehicle\_27.csv* και, αντίστοιχα, σε ποιο MQTT topic θα επικοινωνεί με τον ES ρυθμίζεται μέσω της μεταβλητής *terminalId*.

Τα terminals κάνουν περιοδικούς ελέγχους κάθε *onlineCheckFrequency* (default 5’’) για σύνδεση στο internet. Σε περίπτωση που αυτή δεν υπάρχει παραπέμπουν μέσω Alert στην ενεργοποίησή του.

* Main Activity

Το κεντρικό activity αποτελείται από κουμπιά “Start” και “Stop” για εκκίνηση και τερματισμό της αποστολής των datapoints (γραμμές του csv) του android terminal στον ES, έναν πίνακα με τα δεδομένα και μια μπάρα προόδου για παρακολούθηση των δεδομένων που στέλνονται σε πραγματικό χρόνο.

Η αποστολή γίνεται με σταθερό ρυθμό ένα datapoint ανά δευτερόλεπτο και εάν σταματήσει (μέσω του κουμπιού “Stop” ή λόγω διακοπής της πρόσβασης στο Internet) η επανέναρξή της (με το κουμπί “Start”) γίνεται από την αρχή. Το πλήθος των datapoints που αποστέλλεται εξαρτάται από την τιμή Runtime των ρυθμίσεων (default το σύνολο των γραμμών του csv).

* Settings Activity

Εδώ υπάρχουν επιλογές για είσοδο IP Address, Port και Runtime, όπως και κουμπί “Exit” για έξοδο από την εφαρμογή κατόπιν σχετικού Alert για επιβεβαίωση. Για όλες τις εισόδους πραγματοποιούνται έλεγχοι εγκυρότητας. Σημειώνεται ότι οι ρυθμίσεις αυτές αποθηκεύονται ακόμα και μετά από κλείσιμο τη εφαρμογής.

**MySQL Database:**

Περιλαμβάνει απλώς 2 tables (*datapoints* και *predicted\_datapoints*) τα οποία περιέχουν τις ζητούμενες από την εκφώνηση στήλες. Οδηγίες για σύνδεση στη βάση περιλαμβάνονται στο *./mysql/DB\_Readme.txt* .