

## Τμήμα Πληροφορικής και Τηλεπικοινωνιών

### Ανάπτυξη Λογισμικού για Δίκτυα και Τηλεπικοινωνίες 2017-2018

#### Δεύτερο Παραδοτέο

Βαρβαγιάννης Μάρκος  
Χαραμής Ιωάννης

1115201400017  
1115201400220

#### Περιγραφή:

Οι εφαρμογές έχουν υλοποιηθεί σύμφωνα με τα ζητούμενα της εκφώνησης.

Πιο συγκεκριμένα, έχουν αναπτυχθεί σε γλώσσα Java ένας **MQTT Publisher** για την αποστολή μηνυμάτων σε έναν broker και στον κατάλληλο client (στην περίπτωση μας ένα android τερματικό).

Επίσης έχει υλοποιηθεί μία εφαρμογή για Android που θα είναι ο **MQTT Subscriber**, θα λαμβάνει τα μηνύματα και θα εκτελεί την κατάλληλη ενέργεια.

Οι εφαρμογές αναπτύχθηκαν στο περιβάλλον **IntelliJ IDEA** και **Android Studio**, αντίστοιχα.

#### Λήψη Συχνότητας:

Έχει προβλεφθεί ο JavaClient να λαμβάνει τη συχνότητα είτε ανοίξει πριν, είτε μετά την android εφαρμογή. Για το σκοπό αυτό, όταν ανοίγει ο JavaClient, κάνει publish ένα μήνυμα με context "Frequency", με σκοπό να το λάβει το android και να του απαντήσει ποια πρέπει να είναι η συχνότητα. Αντίστοιχα, όταν ανοίγει πρώτα ο Android Client, κάνει Publish τη συχνότητα ώστε να τη λάβει ο JavaClient, σε περίπτωση που είναι ήδη ανοιχτή.

#### Λειτουργικότητα της Java Εφαρμογής:

Χρησιμοποιούνται τα εξής πακέτα:

- **entropy**: Περιέχει τις κλάσεις Experiment, CalcEntropy και ProbabilityState: Η κλάση *Experiment* αναπαριστά ένα πείραμα με το όνομά του και έναν vector 14 στοιχείων που περιέχει την εντροπία για κάθε έναν από τους 14 αισθητήρες.
- **mqtt**: Περιέχει τις κλάσεις MqttPublish και MqttSubscribe. Στην κλάση *MqttPublish* περιέχονται όλα τα απαραίτητα στοιχεία για τη σύνδεση σε έναν Mqtt broker και την αποστολή ενός μηνύματος στον client. Για το σκοπό αυτό, χρησιμοποιούμε τη συνάρτηση PublishMessage. Αντίστοιχα, στην κλάση *MqttSubscribe*, υλοποιείται ένας Subscriber, ώστε να λαμβάνει τη συχνότητα που ορίζεται στην Android εφαρμογή.
- **shared\_buffer**: Περιέχει τις κλάσεις Producer, Consumer και SharedBuffer. Στην τελευταία υλοποιείται ένας buffer, τον οποίο μοιράζονται ο producer και ο consumer, αλλά και οι συναρτήσεις πρόσβασης σε αυτόν, put και get. Στην κλάση του Producer αποθηκεύονται και το test set, αλλά και το X (που είναι το σύνολο των κατηγοριοποιημένων διανυσμάτων). Στην

κλάση του Consumer αποθηκεύεται ένα αντικείμενο MqttPublisher, ώστε, καθώς παίρνει τις εντολές από τον buffer, να τις στέλνει απευθείας στον broker, με τελικό παραλήπτη το τερματικό Android. Έχει προβλεφθεί ο Consumer να στέλνει τις εντολές με τη συχνότητα που έχει οριστεί από την εφαρμογή Android.

Περισσότερες επεξηγήσεις και λεπτομέρειες υλοποίησης, περιέχονται σαν σχόλια στα πηγαία αρχεία.

## Δομές που χρησιμοποιούνται στον Java Client

Για την υλοποίηση του αλγορίθμου κατηγοριοποίησης, ήταν απαραίτητο να χρησιμοποιηθούν κάποιες δομές. Πιο συγκεκριμένα, το σύνολο των ήδη κατηγοριοποιημένων διανυσμάτων X, αλλά και το test set, με τα διανύσματα προς κατηγοριοποίηση υλοποιούνται με λίστες. Προτιμήθηκαν οι απλά συνδεδεμένες λίστες (LinkedList), καθώς τα δεδομένα προσπελάσσονται εν γένει σειριακά και όχι τυχαία.

Επιπρόσθετα, ο SharedBuffer υλοποιείται με έναν Vector και έχει default μέγεθος 4.

## Έλεγχος σωστής κατηγοριοποίησης και ποσοστό επιτυχίας

Στο πρόγραμμα έχει υλοποιηθεί, όπως αναφέρεται στην εκφώνηση, έλεγχος σχετικά με το αν είναι σωστή η κατηγοριοποίηση και εμφανίζεται στην κονσόλα Success ή Not Success, αντίστοιχα. Ωστόσο, επειδή τα δεδομένα είναι πολλά και δεν είναι ρεαλιστικό να ολοκληρωθούν σε ένα εύλογο χρονικό διάστημα, εμφανίζεται περιοδικά στην κονσόλα (για κάθε 5 διανύσματα που κατηγοριοποιούνται) το ποσοστό επιτυχίας του αλγορίθμου τη δεδομένη χρονική στιγμή.

## Επικοινωνία μεταξύ Java και Android Client:

Εκτελούμε το JavaClient, στη συνέχεια εισάγουμε τα κατάλληλα στοιχεία σύνδεσης στο παράθυρο που εμφανίζεται και τέλος πατάμε το κουμπί Save.

Πιο συγκεκριμένα:

- Το πεδίο **"Server URI"** πρέπει να εισαχθεί χωρίς το "tcp://", καθώς συμπληρώνεται αυτόματα.
- Για το πεδίο **Content** υπάρχουν οι εξής 9 εντολές:

1. Frequency	Η εφαρμογή Android στέλνει πίσω την τιμή της συχνότητας
2. Sound	Ενεργοποιεί τον προεπιλεγμένο ήχο ειδοποίησης για default sec.
3. Sound t (sec)	Ενεργοποιεί τον προεπιλεγμένο ήχο ειδοποίησης για t sec.
4. Flash	Ενεργοποιεί τον φακό της συσκευής (αν υπάρχει) για default sec.
5. Flash t (sec)	Ενεργοποιεί τον φακό της συσκευής (αν υπάρχει) για t sec.
6. Stop Sound	Απενεργοποιεί την ηχητική ειδοποίηση
7. Stop Flash	Απενεργοποιεί την οπτική ειδοποίηση
8. Start	Ενεργοποιεί όλες τις ειδοποιήσεις για default sec.
9. Stop All	Απενεργοποιεί όλες τις ειδοποιήσεις

, όπου default είναι ένα ορισμένο χρονικό διάστημα στην εφαρμογή (5sec).

Αντίστοιχα, εισάγουμε τις ίδιες παραμέτρους στο πεδίο Settings της Android εφαρμογής και συνδεόμαστε στον Broker πατώντας το πλήκτρο Login. Να σημειωθεί πως δεν έχει καμία σημασία ποια εκ των δύο εφαρμογών θα συνδεθεί πρώτη με τον Broker, καθώς έχουν προβλεφθεί και τα δύο σενάρια.

## Λειτουργικότητα της Android Εφαρμογής

Η εφαρμογή Android, μετά το Subscribe σε ένα topic, δέχεται τις εντολές στη μορφή που περιγράφηκε παραπάνω και τις μεταφράζει ώστε να πραγματοποιήσει την κατάλληλη λειτουργία. Όπως και στην Java, στο πεδίο Server URI πρέπει να παραλειφθεί το “tcp://” καθώς προστίθεται αυτόματα. Επίσης έχει υλοποιηθεί ένα Service, ώστε η εφαρμογή να παραμένει λειτουργική και μετά το πάτημα του “home button”, και ένα Thread, προκειμένου να ελέγχεται η συνδεσιμότητα της κινητής συσκευής με το διαδίκτυο και να προτείνονται τρόποι επανεγκαθίδρυσής της. Ακόμη, έχει υλοποιηθεί και η δυνατότητα παραμετροποίησης της συχνότητας με την οποία η Java εφαρμογή στέλνει τις εντολές. Η συχνότητα αυτή, έχει τη μορφή δευτερολέπτων.

Έχουν προβλεφθεί όλες οι λειτουργίες που περιγράφονται στην εκφώνηση και επιπλέον:

- Με το swipe από το μενού των running apps, απενεργοποιείται η εφαρμογή καθώς και το αντίστοιχο Service και Thread.
- Όταν έχοντας συνδεθεί σε ένα topic, πατήσουμε το back button (είτε του κινητού, είτε αυτό που βρίσκεται πάνω αριστερά, γίνεται αποσύνδεση και κλείνουν όλες οι ηχητικές και οπτικές ειδοποιήσεις.
- Δυνατότητα αποθήκευσης των παραμέτρων σύνδεσης, ώστε να μην είναι αναγκαία η εισαγωγή τους σε κάθε εκκίνηση της εφαρμογής.
- Ξεχωριστά πλήκτρα για την ενεργοποίηση/απενεργοποίηση της ηχητικής και της οπτικής ειδοποίησης αντίστοιχα, τα οποία είναι διαδραστικά, παρέχοντας feedback για την κατάσταση λειτουργίας τους.
- Κατά την απώλεια σύνδεσης με το διαδίκτυο, η αντίστοιχη ειδοποίηση που εμφανίζεται στο χρήστη αποτελεί συντόμευση στις ρυθμίσεις της συσκευής, ώστε να διευκολύνεται η επανεγκαθίδρυση της σύνδεσης.
- Δημοσίευση της συχνότητας αποστολής μηνυμάτων του Java Client, σε διάφορες φάσεις της επικοινωνίας μεταξύ των εφαρμογών, ώστε να εξασφαλιστεί η λήψη της από τον Java Client σε κάθε περίπτωση.

Περισσότερες επεξηγήσεις και λεπτομέρειες υλοποίησης, περιέχονται σαν σχόλια στα πηγαία αρχεία.