

Ατομική Διπλωματική Εργασία

**ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΜΑΖΙΚΩΝ ΠΡΟΣΟΜΟΙΩΣΕΩΝ ΣΕ
ΕΞΥΠΝΕΣ ΚΙΝΗΤΕΣ ΣΥΣΚΕΥΕΣ**

Μάριος Μίντζης

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σύστημα Διαχείρισης Μαζικών Προσομοιώσεων σε Έξυπνες Κινητές Συσκευές

Μάριος Μίντζης

Επιβλέπων Καθηγητής

Δημήτρης Ζεϊναλιπούρ

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2014

Ευχαριστίες

Θα ήθελα αρχικά να ευχαριστήσω τον κύριο Δημήτρη Ζεϊναλιπούρ, ο οποίος μου προσέφερε αυτή την πολύ ενδιαφέρουσα και ερευνητική εργασία, καθώς και για την πλήρη κατανόηση και υποστήριξη που είχα από αυτόν, κατά τη διάρκεια της διπλωματικής μου εργασίας. Πρόκειται για ένα σπουδαίο καθηγητή και ερευνητή με πολλές γνώσεις σε όλους του τομείς της επιστήμης της Πληροφορικής, αλλά πάνω απ' όλα για ένα εξαιρετικό άνθρωπο.

Θα ήθελα επίσης να ευχαριστήσω το Τμήμα της Πληροφορικής του Πανεπιστημίου Κύπρου για την αγορά και διάθεση όλων των συσκευών που ήταν απαραίτητα για την ολοκλήρωση της παρούσας διπλωματικής εργασίας.

Ακόμη θα ήθελα να ευχαριστήσω το Γεώργιο Λάρκου, μεταπτυχιακό φοιτητή του Πανεπιστημίου Κύπρου, για όλη τη καθοδήγηση και τη βοήθεια που μου πρόσφερε κατά τη διάρκεια υλοποίησης της διπλωματικής μου εργασίας.

Τέλος οφείλω να πω ένα μεγάλο ευχαριστώ στους φίλους μου εντός και εκτός Πανεπιστημίου οι οποίοι μου συμπαραστάθηκαν όλο αυτό τον καιρό αλλά και ένα τεράστιο ευχαριστώ στην οικογένειά μου και ιδιαίτερα στους γονείς μου Σταύρο και Γεωργία για την ηθική και οικονομική υποστήριξη που μου πρόσφεραν όχι μόνο κατά τη διάρκεια εκπόνησης της παρούσας διπλωματικής εργασίας αλλά και καθ' όλη τη διάρκεια των σπουδών μου.

Αναγνώριση

Επίδειξη του συστήματος προσομοιώσεων της διπλωματικής αυτής εργασίας έχει παρουσιαστεί στο ακόλουθο συνέδριο:

- **“Sensor Mockup Experiments with SmartLab (Demo)”**, G. Larkou, M. Mintzis, S. Taranto, A. Konstantinidis, P.G. Andreou, D. Zeinalipour-Yazti, "Proceedings of the 13th international conference on Information processing in sensor networks" (IPSN '14) , Berlin Germany April 15-17, **2014**

Το υποσύστημα αυτό έχει επίσης συμβάλει στη δημιουργία μιας μελέτης η οποία έχει υποβληθεί για δημοσίευση και στο παρόν στάδιο είναι υπό αξιολόγηση.

- **“Managing Big Data Experiments on Smartphones”**, G Larkou, M Mintzis, P.G. Andreou, A Konstantinidis, D Zeinalipour-Yazti

Περίληψη

Στις μέρες μας υπάρχει μια εκρηκτική αύξηση στις δυνατότητες αίσθησης και υπολογισμού στις έξυπνες κινητές συσκευές. Ο επαναπρογραμματισμός αυτών για τον έλεγχο μιας εφαρμογής και της συλλογής δεδομένων σε κλίμακα είναι σήμερα μία επίπονη και χρονοβόρα διαδικασία που θέτει σημαντικές υλικοτεχνικές προκλήσεις.

Η παρούσα διπλωματική εργασία αφορά το σχεδιασμό και την υλοποίηση ενός συστήματος προσομοιώσεων (mock-up system), το οποίο θα προσφέρει στους προγραμματιστές εφαρμογών Android την δυνατότητα καταγραφής και εξομοίωσης των αισθητήρων που χρησιμοποιούνται από τη κινητή συσκευή στην εφαρμογή τους. Παρουσιάζουμε την αρχιτεκτονική του συστήματος καθώς επίσης και της δικής μας βιβλιοθήκης για κινητές συσκευές Android, η οποία ενσωματώνεται στην εφαρμογή κάθε χρήστη με ελάχιστες αλλαγές, αφού ακολουθεί το υφιστάμενο πρωτόκολλο των αισθητήρων του Android, και χρησιμοποιείται κυρίως στη διαδικασία εξομοίωσης αφού παρέχει τη δυνατότητα να τροφοδοτεί τη κινητή συσκευή με δεδομένα αισθητήρων προερχόμενα από εξωτερική βάση δεδομένων NoSQL .

Μέσω διεξαγωγής πειραμάτων έχει διαφανεί ότι το προτεινόμενο σύστημα μπορεί να προσφέρει διευκόλυνση στην ερευνητική κοινότητα για διεξαγωγή πειραμάτων σε κινητές συσκευές. Παρουσιάζεται επίσης μια αρχιτεκτονική για διαχείριση αυτών των πειραμάτων εξομοίωσης σε πραγματικές έξυπνες κινητές συσκευές. Έχοντας συλλέξει δεδομένα από τους αισθητήρες των έξυπνων κινητών συσκευών από το πλήθος και οργανώνοντάς τα στο αποθετήριο μεγάλων δεδομένων μας παρουσιάζουμε την εφαρμογή αυτής της αρχιτεκτονικής. Ειδικότερα, παρουσιάζουμε τα πλεονεκτήματα αυτής της αρχιτεκτονικής μέσω μιας μελέτης περιπτώσεων που περιλαμβάνει την αξιολόγηση επιμέρους στοιχείων που αποτελούν μέρος ενός περίπλοκου συστήματος εσωτερικού εντοπισμού θέσης για έξυπνες κινητές συσκευές. Η δεδομένη μελέτη δείχνει πώς η προτεινόμενη αρχιτεκτονική, μας επιτρέπει να αποκομίσουμε νέες γνώσεις όσον αφορά την απόδοση συγκεκριμένων αλγορίθμων και εφαρμογών εσωτερικής τοποθέτησης μέσω απλούστευσης της διαχείρισης των δεδομένων μεγάλης κλίμακας.

Περιεχόμενα

| | |
|---|------------|
| Ευχαριστίες | i |
| Αναγνώριση | ii |
| Περίληψη | iii |
| | |
| Κεφάλαιο 1 | 1 |
| 1.1 Εισαγωγή | 1 |
| 1.2 Υποκίνηση | 1 |
| 1.3 Εισαγωγή στον Πληθοπορισμό | 2 |
| 1.4 Εισαγωγή στο σύστημα SmartLab [1] | 3 |
| 1.5 Υποσύστημα Προσομοιώσεων | 5 |
| Κεφάλαιο 2 | 7 |
| 2.1 Εισαγωγή | 7 |
| 2.2 Wireless Smartphone Testbeds | 8 |
| 2.3 People Centric Testbeds | 12 |
| 2.4 Android Emulators/Simulators | 14 |
| 2.5 Android Sensors (SDK) [21] | 17 |
| Κεφάλαιο 3 | 20 |
| 3.1 Εισαγωγή κεφαλαίου | 20 |
| 3.2 Υφιστάμενη αρχιτεκτονική SmartLab | 20 |
| 3.3 Υποσύστημα RM Mockups ως επέκταση του Device Server (Σχήμα 3.2.1 A) | 22 |
| 3.4 SLSensorManager library (Σχήμα 3.2.1 B) | 22 |
| 3.5 Αποθετήριο Μεγάλων δεδομένων (Σχήμα 3.2.1 C) | 23 |
| Κεφάλαιο 4 | 25 |
| 4.1 Εισαγωγή | 25 |
| 4.2 SmartLab Device Server σε Java | 26 |
| 4.3 Σύστημα προσομοιώσεων πάνω από το Device Server του SmartLab | 27 |
| 4.3.1 TCPServer (Καταγραφή) | 27 |
| 4.3.2 EmulationServer (Εξομοίωση) | 28 |
| 4.3.3 EmulationClient (Εξομοίωση) | 29 |
| 4.4 Παράρτημα Συστήματος στην ιστοσελίδα SmartLab | 30 |

| | | |
|-------------------------|---|-----------|
| 4.5 | SmartLabRecorder (Καταγραφή) | 31 |
| 4.6 | Βιβλιοθήκη SLSensorManager (Σχήμα 4.6) | 31 |
| 4.7 | Σύνοψη διαδικασίας καταγραφής και εξομοίωσης. | 34 |
| 4.8 | Προβλήματα που προέκυψαν και αντιμετώπισή τους | 36 |
| Κεφάλαιο 5 | | 38 |
| 5.1 | Εισαγωγή | 38 |
| 5.2 | Πειραματική Υποδομή | 39 |
| 5.3 | Πειραματική Σειρά 1: Ανάπτυξη Πειραμάτων μέσω του υποσυστήματος προσομοιώσεων | 41 |
| 5.4 | Πειραματική σειρά 2 : Αλληλεπίδραση με τη βιβλιοθήκη SLSensorManager | 43 |
| 5.5 | Πειραματική σειρά 3 : Παρουσίαση νέων κατευθύνσεων έρευνας | 45 |
| Κεφάλαιο 6 | | 47 |
| 6.1 | Συμπεράσματα | 47 |
| 6.2 | Μελλοντικές Επεκτάσεις | 48 |
| Βιβλιογραφία | | 49 |
| Παράρτημα Α | | 52 |

Κεφάλαιο 1

Εισαγωγή

| | | |
|-----|-------------------------------|---|
| 1.1 | Εισαγωγή | 1 |
| 1.2 | Υποκίνηση | 1 |
| 1.3 | Εισαγωγή στον Πληθοπορισμό | 2 |
| 1.4 | Εισαγωγή στο σύστημα SmartLab | 3 |
| 1.5 | Υποσύστημα Προσομοιώσεων | 5 |

1.1 Εισαγωγή

Στα επόμενα υποκεφάλαια θα γίνει μια αναφορά στην υποκίνηση της διπλωματικής εργασίας αυτής. Μετέπειτα, θα γίνει μια αναφορά στο πληθοπορισμό, μια έννοια βασική για ορισμένα πειράματα στο κεφάλαιο 5, όπου μέσω πληθοπορισμού συλλέχτηκαν τα δεδομένα που χρησιμοποιήθηκαν στις πειραματικές μελέτες. Ακολούθως γίνεται μια εκτενής αναφορά στο SmartLab στο οποίο ενσωματώθηκε το υποσύστημα προσομοιώσεων. Τέλος γίνεται μια αναφορά στο ίδιο το υποσύστημα προσομοιώσεων.

1.2 Υποκίνηση

Οι έξυπνες κινητές συσκευές έχουν γοητεύσει τους ανθρώπους με τις εκπληκτικές δυνατότητες που μας παρέχουν. Δυνατότητες αίσθησης της θερμότητας ενός θερμικού σώματος, της ατμοσφαιρικής πίεσης, της ταχύτητας ενός οχήματος, του χωρικού προσανατολισμού ενός σταθερού αντικειμένου, της μαγνητικής ακτινοβολίας σε μια περιοχή εξόρυξης, του εντοπισμού της γεωγραφικής θέσης σε απομακρυσμένα μέρη της

Γής και πολλά άλλα. Το τελευταίο καιρό οι δυνατότητες αυτές έχουν γίνει αναπόσπαστο κομμάτι της ζωής τους λόγω της απεριόριστης χρησιμότητας τους προς αυτούς στις καθημερινές τους δραστηριότητες. Η εικονικοποίηση των έξυπνων κινητών συσκευών (Virtualization) είναι μια ενδιαφέρουσα τεχνολογία η οποία έχει την ικανότητα για το ίδιο κομμάτι υλικού να φιλοξενήσει διαφορετικές πλατφόρμες χωρίς επιπλέον κόστος. Η ικανότητα να τρέχουν τέτοιες εικονικοποιημένες έξυπνες κινητές συσκευές σε επιτραπέζιους υπολογιστές ή μεγάλους διακόμισες ανοίγει νέους ορίζοντες σε εφαρμογές που βασίζονται στη χρήση αισθητήρων. Αυτό συμβαίνει γιατί δίνεται η ευκαιρία να τρέχουν εκατοντάδες έξυπνες κινητές συσκευές εφαρμογές που χρησιμοποιούν αισθητήρες ταυτόχρονα στη ίδια μηχανή.

Λαμβάνοντας υπόψη παλαιότερες προσεγγίσεις όσων αφορά τη προσομοίωση και εξομοίωση δεδομένων των αισθητήρων των έξυπνων κινητών συσκευών υλοποιήθηκε στα πλαίσια της παρούσας διπλωματικής ένα καινοτόμο σύστημα (RM Sensor Mockups).

Βασικό κίνητρο υπήρξαν οι τεράστιες δυνατότητες αίσθησης που μπορούν να προσφέρουν στις μέρες μας οι έξυπνες κινητές συσκευές αλλά και η μη επαρκής αξιοποίηση τους από προηγούμενες προσεγγίσεις. Βασική έλλειψη στις προηγούμενες προσεγγίσεις ουσιαστικά ήταν η δυνατότητα της επαναληψιμότητας, δηλαδή να δοθεί η δυνατότητα στο χρήστη να προσομοιώσει ξανά και ξανά κάποιες συγκεκριμένες συνθήκες για να μπορέσει να επαληθεύσει συγκεκριμένες υποθέσεις. Ως γνωστόν, οι αισθητήρες των κινητών συσκευών παρέχουν μεγάλη ακρίβεια και θα ήταν σχεδόν αδύνατο να παραχθούν από την αρχή ακριβώς οι ίδιες ενδείξεις και αυτό δεν θα επέτρεπε τον έλεγχο συγκεκριμένων καταστάσεων που χρειάζονται την αναπαράσταση των ίδιων ακριβώς μετρήσεων.

1.3 Εισαγωγή στον Πληθοπορισμό

Οι χρήστες έξυπνων κινητών συσκευών είναι συνεχώς σε κίνηση δημιουργώντας έτσι μεγάλες ποσότητες δεδομένων προερχόμενων από τους αισθητήρες των συσκευών που συμβάλλουν στην εξέλιξη νέων υπηρεσιών και εφαρμογών. Το μοντέλο αυτό ουσιαστικά ονομάζεται πληθοπορισμός το οποίο σταδιακά καθίσταται ως το επικρατέστερο μέσο για συλλογή δεδομένων από μεγάλες ομάδες ανθρώπων αλλά και επεξεργασία αυτών.

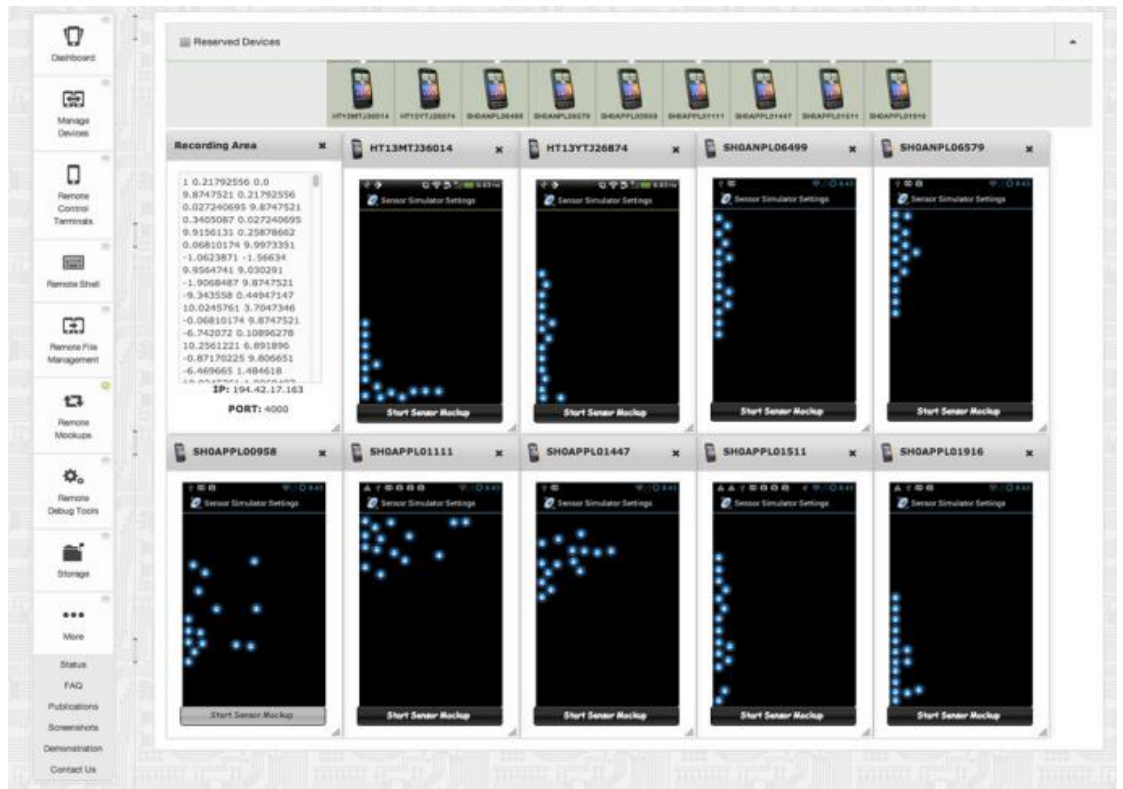
Ο όρος πληθοπορισμός (crowdsourcing) αναφέρεται σε ένα νέο καταναμημένο, κοινωνικό, συνεργατικό μοντέλο επίλυσης προβλημάτων στο οποίο ένα πλήθος απροσδιορίστου μεγέθους δεσμεύεται να λύσει ένα περίπλοκο πρόβλημα μέσω μίας ανοικτής κλήσης [2].

Τυπικά παραδείγματα εφαρμογών πληθοπορισμού περιλαμβάνουν εμπορικές υπηρεσίες όπως το παιχνίδι αναγνώρισης από φωτογραφία (Gwap.nets ESP), η υπηρεσία διόρθωσης βιβλίου (reCAPTCHA.nets), καθώς επίσης και εξειδικευμένες αγορές για την ανάθεση πληθοποριστικών καθηκόντων (π.χ. Amazons Mechanical Turk) και ακαδημαϊκές τεχνικές διαχείρισης δεδομένων [3,4], η διαχείριση δικτύου [5], η ανάπτυξη πηγαίου κώδικα [6], υπολογιστική γλωσσολογία [7] και η ενεργός μάθηση [8].

Οι εφαρμογές αυτές συχνά έχουν την τάση να ανταλλάσσουν πληροφορίες που έχουν αποκτηθεί από ενσωματωμένους στη κινητή συσκευή αισθητήρες (π.χ. κάμερα, GPS, κλπ.) με αποτέλεσμα την έκρηξη των γεωγραφικά αναγνωρισμένων χρονικών δεδομένων που μεταδίδονται μέσα από το Διαδίκτυο.

1.4 Εισαγωγή στο σύστημα SmartLab [1]

Ο επαναπρογραμματισμός των έξυπνων κινητών συσκευών για τον έλεγχο μιας εφαρμογής καθώς επίσης και η συλλογή δεδομένων σε κλίμακα είναι σήμερα μια επίπονη, χρονοβόρα διαδικασία που θέτει σημαντικές υλικοτεχνικές προκλήσεις. Επιπλέον, αν επιτραπεί η απρόσκοπτη πειραματική επαναληψιμότητα και η τυποποίηση, θα προκαλέσει δυσκολίες στις έρευνες προσανατολισμένες σε έξυπνες κινητές συσκευές. Κοιτάζοντας άλλους τομείς έρευνας, συνειδητοποιούμε ότι τα ανοικτά σύνολα δεδομένων συγκριτικής αξιολόγησης και τα συναφή σύνολα κατάλληλων αντικειμενικών δεδομένων έχουν διαδραματίσει σημαντικό ρόλο στην ακαδημαϊκή και βιομηχανική έρευνα κατά τις τελευταίες δεκαετίες. Για παράδειγμα το συνέδριο TREC το οποίο συγχρηματοδοτείται από το Εθνικό Ινστιτούτο Προτύπων και τεχνολογίας (NIST) του Υπουργείου Εμπορίου των ΗΠΑ ξεκίνησε από τη κοινότητα



Σχήμα 1.1

Ένα παράδειγμα πειράματος του συστήματος εξομοίωσης αυτής της εργασίας όπου ο χρήστης τροφοδοτεί τις πραγματικές έξυπνες κινητές συσκευές με μετρήσεις αισθητήρων από το χώρο αποθήκευσης δεδομένων και εποπτεύει τα αποτελέσματα μέσω ενός Web 2.0 περιβάλλοντος εργασίας χρήστη.

ανάκτησης πληροφοριών, το TCP (το συμβούλιο απόδοσης επεξεργασίας συναλλαγών) μια μη κερδοσκοπική εταιρία, η οποία ιδρύθηκε για να καθορίσει την επεξεργασία συναλλαγών και δεδομένων μεταφοράς, ξεκίνησε και αυτή σε μεγάλο βαθμό από τη κοινότητα διαχείρισης δεδομένων.

Στη συνέχεια γίνεται μια εκτενής περιγραφή του Smartlab[1] στο οποίο ενσωματώθηκε το υποσύστημα της καταγραφής και εξομοίωσης των αισθητήρων των έξυπνων κινητών συσκευών. Μέσω αυτού επίσης έγιναν τα όποια πειράματα εσωτερικής τοποθέτησης. Το SmartLab ουσιαστικά είναι μια ολοκληρωμένη αρχιτεκτονική για τη διαχείριση ενός συμπλέγματος πραγματικών συσκευών Android (ARDs) καθώς επίσης και εικονικών συσκευών Android (AVDs), το οποίο μπορεί ο χρήστης να διαχειριστεί μέσω μιας διαισθητικής web-based διεπαφής. Μεταξύ άλλων η

τρέχουσα αρχιτεκτονική είναι ιδανική για τη διεξαγωγή ενός λεπτομερούς και χαμηλού επιπέδου ελέγχου των διαφόρων αλγορίθμων εσωτερικής τοποθέτησης πάνω σε πραγματικές ετερογενείς έξυπνες κινητές συσκευές. Επίσης είναι ιδανική για τη συλλογή και διαχείριση δεδομένων τα οποία συλλέχτηκαν από το πλήθος (π.χ. τα δεδομένα WIFI RSS[9]) και την αποθήκευση τους στην υποδομή μεγάλων δεδομένων, καθώς και για τη διενέργεια πειραμάτων εξομοίωσης των αισθητήρων των έξυπνων κινητών συσκευών (Σχήμα 1.1).

Το τρέχον υλικό του SmartLab αποτελείται από 40 συσκευές Android συνδεδεμένες μέσω διαφόρων μέσων (δηλαδή ενσύρματα, ασύρματα και εικονικά) σε ένα προσωπικό νέφος (datacenter). Μέσα από μία διαισθητική web-based διεπαφή, οι χρήστες μπορούν να εκτελέσουν διάφορες λειτουργίες οι οποίες βοηθούν περαιτέρω την διαδικασία χρήσης του συστήματος εξομοίωσης. Μπορούν ταυτόχρονα να φορτώσουν και να εγκαταστήσουν το Android APK (εκτελέσιμο) για ένα αριθμό κινητών συσκευών, μπορούν να καταγράψουν στιγμιότυπα της οθόνης, να μεταφέρουν αρχεία και να εκτελέσουν εντολές κελύφους UNIX. Το σημαντικότερο νέο χαρακτηριστικό είναι η δυνατότητα τροφοδότησης δεδομένων GPS και δεδομένων των αισθητήρων συσκευών Android στις κινητές συσκευές του SmartLab.

1.5 Υποσύστημα Προσομοιώσεων

Το βασικό χαρακτηριστικό και υποκατηγορία του SmartLab που δημιουργήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας ήταν το RM Sensor Mockups.

Στο πλαίσιο των έξυπνων κινητών συσκευών, το mockup αναφέρεται στη διαδικασία επέκτασης των αισθητήρων κάποιας εικονικής ή πραγματικής συσκευής Android με προσαρμοσμένες τιμές καθώς και η υποστήριξη της προσθήκης αισθητήρων που μπορεί να μην παρέχονται από το υλικό μιας πραγματικής συσκευής (π.χ. NFC). Προκειμένου να μπορεί το σύστημα να υποστηρίξει τους αισθητήρες GPS και άλλους αισθητήρες (π.χ. επιταχυνσιόμετρο), τόσο στις πραγματικές όσο και στις εικονικές συσκευές Android δημιουργήθηκε η βιβλιοθήκη SLSensorManager. Η συγκεκριμένη βιβλιοθήκη είναι εύκολη στην εγκατάσταση και χρήση και μπορεί να χρησιμοποιηθεί σε εκδόσεις Android 4.0 και άνω. Μπορεί να ενσωματωθεί σε οποιαδήποτε εφαρμογή Android επιτρέποντας διαδραστικότητα με το υποσύστημα του SmartLab (RM) που βρίσκεται στο Device Server. Στη πραγματικότητα, η εν λόγω βιβλιοθήκη έχει την ίδια ακριβώς

διασύνδεση με το Android SDK SensorManager και κατά συνέπεια ο χρήστης μπορεί να παρακάμψει την προεπιλεγμένη συμπεριφορά του SensorManager. Η αρχιτεκτονική του υποσυστήματος αποτελείται από πολλά συναρπαστικά εξαρτήματα για τον επαναπρογραμματισμό των κινητών συσκευών ως προς τον έλεγχο μιας εφαρμογής και συλλογή δεδομένων με εύκολο τρόπο, καθώς και την εκτέλεση διαφόρων πειραμάτων όπου οι συσκευές τροφοδοτούνται από δεδομένα αισθητήρων RSS fingerprints. Βασικό κομμάτι της αρχιτεκτονικής είναι η αποθήκευση των δεδομένων σε NoSQL αποθετήριο δεδομένων σε μορφή JSON. Τα δοθέντα σύνολα δεδομένων μπορούν να αναπαραχθούν στη πλατφόρμα δοκιμών που περιλαμβάνει 40 κινητές συσκευές προσβάσιμες μέσω μιας WEB 2.0 διεπαφής χρήστη.

Κεφάλαιο 2

Υπόβαθρο και Σχετική Δουλειά

| | | |
|-----|------------------------------|----|
| 2.1 | Εισαγωγή | 7 |
| 2.2 | Wireless Smartphone Testbeds | 8 |
| 2.3 | People Centric Testbeds | 12 |
| 2.4 | Android Emulators/Simulators | 14 |
| 2.5 | Android Sensors (SDK) | 17 |

2.1 Εισαγωγή

Η έμπνευση της δημιουργίας του SmartLab προήλθε κυρίως από το PlanetLab [10] και το Emulab [11], τα οποία υπήρξαν πρωτοπόρα στη παγκόσμια έρευνα αισθητήρων μέσω δικτύου. Παροχή έμπνευσης επίσης ήταν και το MoteLab [12] και Amazon Elastic Compute Cloud (EC2). Οι προαναφερθείσες προσπάθειες δεν επικεντρώθηκαν στις έξυπνες κινητές συσκευές και ως εκ τούτου αυτές οι κλίνες (testbeds) δοκιμών είχαν ριζικά διαφορετικές αρχιτεκτονικές και επιθυμίες.

Στις πιο κάτω υποενότητες θα γίνει μια εκτενής επισκόπηση έργων που σχετίζονται με εξομοιωτές Android και των testbeds που σχετίζονται με το SmartLab. Στη συνέχεια, θα γίνει μια αναφορά σε προηγούμενες προσεγγίσεις συστημάτων ή εφαρμογών που χρησιμοποιούσαν αισθητήρες και θα μελετηθούν πιθανόν μειονεκτήματα και πλεονεκτήματα των προσεγγίσεων αυτών και θα διαφανούν οι διαφορές ως προς το υποσύστημα προσομοιώσεων που δημιουργήθηκε στα πλαίσια της παρούσας



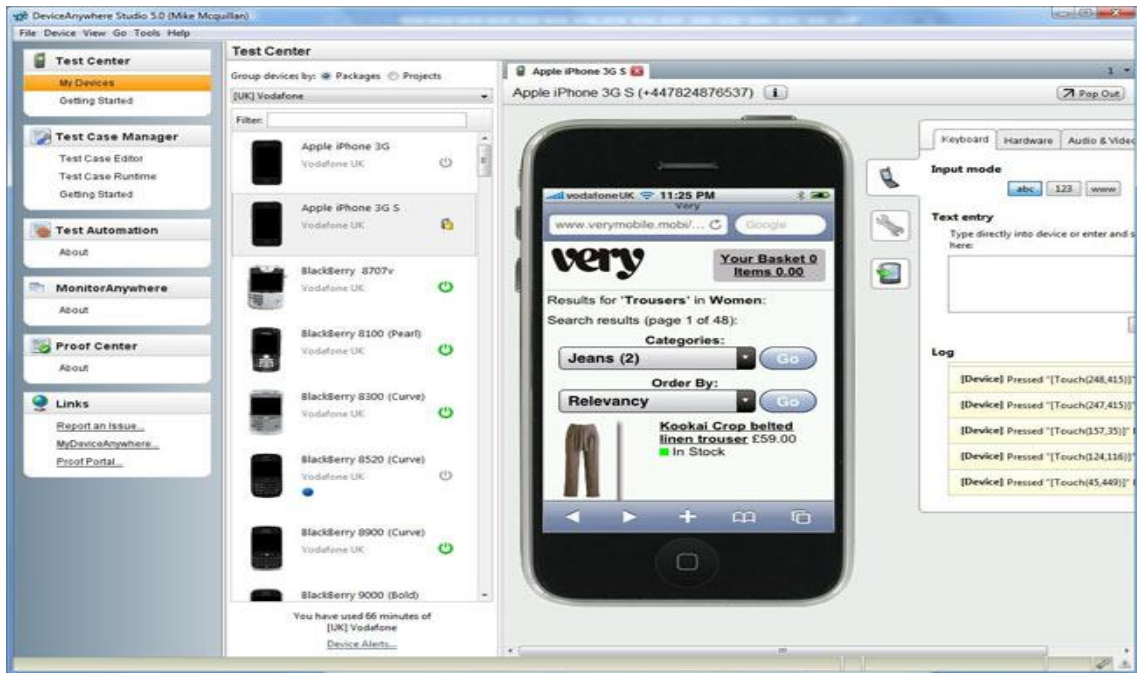
Σχήμα 2.2.1 Perfecto Mobile: Διεπαφή μέσω της οποίας οι χρήστες μπορούν να διαχειριστούν τις συσκευές

διπλωματικής. Τέλος γίνεται μια αναφορά στο Android SDK και συγκεκριμένα στους αισθητήρες που παρέχονται και στη λειτουργία του Android SensorManager.

2.2 Wireless Smartphone Testbeds

Υπάρχουν επί του παρόντος πολλές εμπορικές πλατφόρμες που παρέχουν εξ αποστάσεως πρόσβαση σε πραγματικές έξυπνες κινητές συσκευές, συμπεριλαμβανομένης και της Samsung's Remote Test Lab [13], PerfectoMobile [14], Device Anywhere [15] και AT&T ARO [16].

PerfectoMobile (Σχήμα 2.2.1): Μια ασφαλής πλατφόρμα που επιτρέπει την απομακρυσμένη πρόσβαση σε πραγματικές κινητές συσκευές με ευέλικτες επιλογές ανάπτυξης. Παρέχει στους χρήστες τη δυνατότητα διαδραστικότητας με τις κινητές



Σχήμα 2.2.2 DeviceAnyware :Διεπαφή μέσω της οποίας οι χρήστες μπορούν να διαχειριστούν τις συσκευές

συσκευές που είναι συνδεδεμένες στη νεφέλη και επιτρέπει την απομακρυσμένη πραγματοποίηση πειραμάτων σε αυτές.

Device Anyware (Σχήμα 2.2.2): Μία εύχρηστη πλατφόρμα για τη δοκιμή κινητών εφαρμογών καθώς επίσης και τη παρακολούθηση της ευχρηστίας, της απόδοσης και της διαθεσιμότητας των κινητών εφαρμογών. Παρέχει ευέλικτες λύσεις όσον αφορά την εκτέλεση και έλεγχο των εφαρμογών των χρηστών σε πραγματικό χρόνο. Προσφέρει αυτοματοποιημένα εργαλεία για την παρακολούθηση της οθόνης και εξασφαλίζει στο χρήστη την εύκολη και ποιοτική χρήση των έξυπνων κινητών συσκευών.

Samsung's Remote Test Lab (Σχήμα 2.2.3): Το RTL βοηθάει τους κατασκευαστές εφαρμογών να δοκιμάσουν τις εφαρμογές τους σε πραγματικές κινητές συσκευές εξ αποστάσεως. Επί του παρόντος υπάρχουν εξυπηρετητές στη Κορέα, στο Ηνωμένο Βασίλειο, στη Πολωνία, στις ΗΠΑ και στην Ινδία. Το σύστημα αυτό προσφέρει τον έλεγχο της

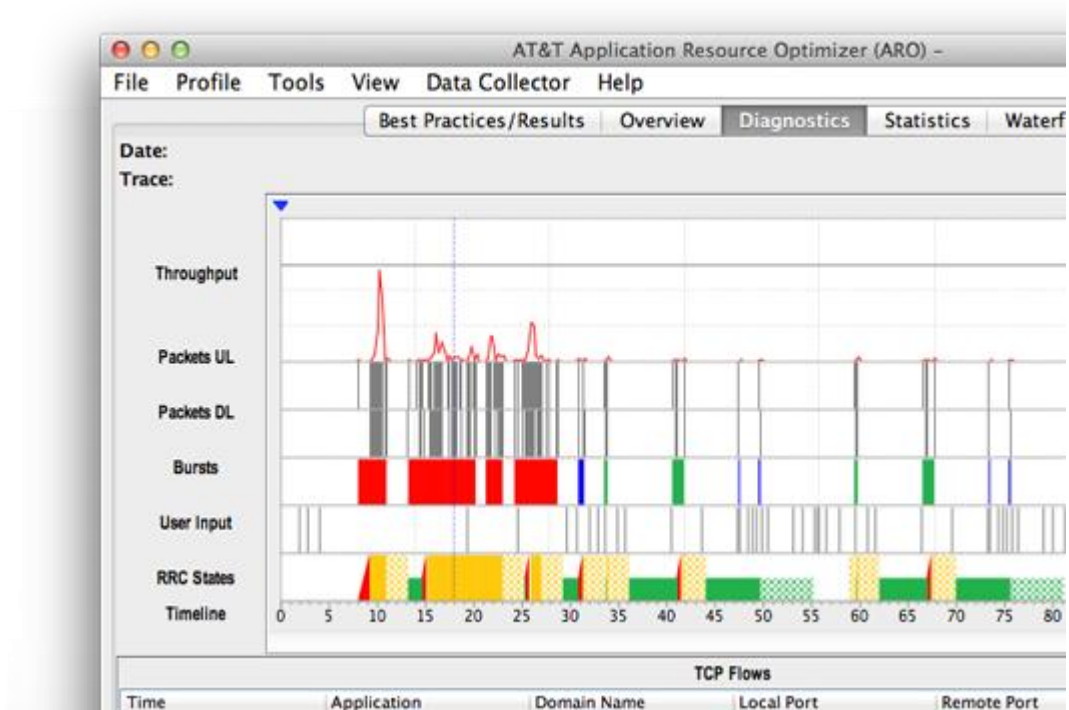
Device List

Remote Test Lab(RTL) allows users to install and test applications over the web.

Σχήμα 2.2.3 Samsung RMTL : Διαπαφή μέσω της οποίας οι χρήστες μπορούν να διαχειριστούν τις συσκευές

συσκευής ενεργοποιώντας τη χρήση των κουμπιών της κινητής συσκευής και παρακολούθησης της οθόνης. Οι χρήστες μπορούν να εγκαταστήσουν και να διαχειριστούν εφαρμογές στις απομακρυσμένες κινητές συσκευές επιτρέποντας την επαναλαμβανόμενη αυτοματοποιημένη δοκιμή σε αυτές.

AT & T Εφαρμογή Βελτιστοποίησης Πόρων (ARO) (Σχήμα 2.2.4): Είναι μια εφαρμογή που παρέχει συστάσεις που επιτρέπουν στους χρήστες τη βελτιστοποίηση της απόδοσης των κινητών εφαρμογών στο διαδίκτυο, τη καλύτερη και πιο αποτελεσματική χρήση της μπαταρίας, και τη μείωση των επιπτώσεων του δικτύου. Η AT & T ARO αποτελείται από δύο μέρη, το Data Collector και το Data Analyzer. Ο Data Collector μαζεύει δεδομένα από το κινητό που χρησιμοποιεί ο χρήστης όσον αφορά την εφαρμογή που δημιούργησε ο ίδιος. Στη συνέχεια τα δεδομένα δίδονται στον αναλυτή



Σχήμα 2.2.4 AT & ARO : Παράδειγμα διάγνωσης της κατανάλωσης διαφόρων παραμέτρων

δεδομένων ο οποίος εξετάζει το πώς η εφαρμογή του χρήστη χειρίζεται την προσωρινή αποθήκευση, και πώς διαχειρίζεται τις συνδέσεις δικτύου.

Αυτές οι πλατφόρμες διαφέρουν από το SmartLab για τους εξής λόγους : 1) είναι κυρίως προσανατολισμένες σε δοκιμές σεναρίων εφαρμογής σε μεμονωμένες έξυπνες κινητές συσκευές και 2) είναι κλειστές και συνεπώς, δεν παρέχουν οποιαδήποτε πληροφορία για το πώς μπορεί ένας χρήστης να κτίσει και να τρέξει αποτελεσματικά εφαρμογές στις έξυπνες κινητές συσκευές σε κλίμακα (σχήμα 2.1). Έγινε ουσιαστικά μια προσπάθεια ώστε να μπορούν να χρησιμοποιούν πολλές συσκευές ταυτόχρονα χωρίς όμως να στηρίζουν το ευρύ φάσμα των λειτουργιών που πλέον παρέχονται από το SmartLab όπως οι αισθητήρες και η εξομοίωση αυτών στις πραγματικές κινητές συσκευές .

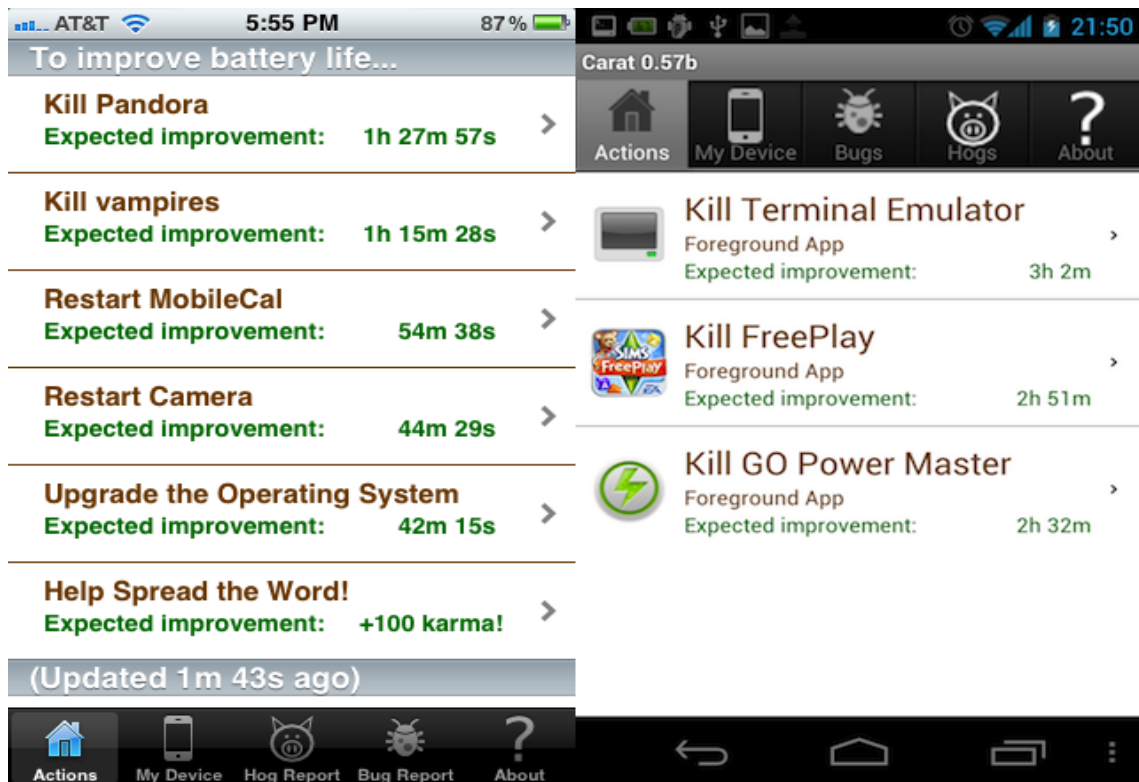
| | Application Installation | Screen Capture | Scalable | Remote Access | Sensor Mockups |
|--------------------------|--------------------------|----------------|----------|---------------|----------------|
| Perfecto Mobile | ✓ | ✓ | ✓ | ✓ | ✓ |
| Samsung's Remote TestLab | ✓ | ✓ | ✓ | ✓ | ✗ |
| Device Anyware | ✓ | ✓ | ✓ | ✓ | ✗ |
| AT&T ARO | ✓ | ✗ | ✗ | ✗ | ✗ |

Σχήμα 2.2.5: Υπηρεσίες που παρέχονται από κάθε πλατφόρμα

2.3 People Centric Testbeds

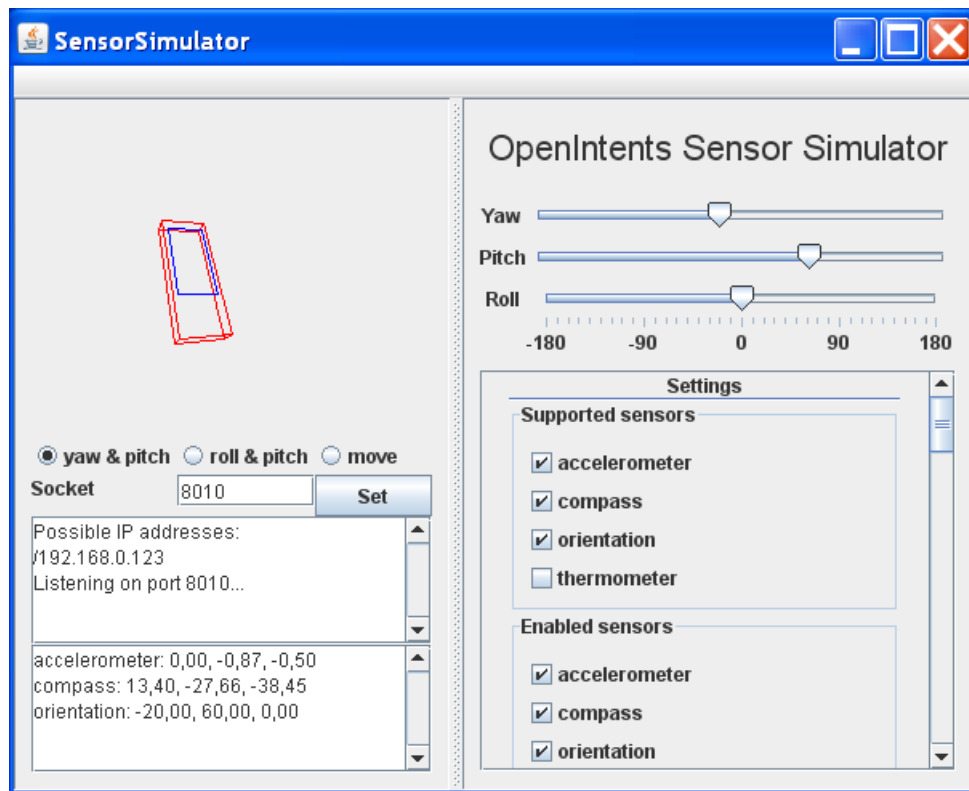
Ένα ανθρωποκεντρικό σύστημα που χρησιμοποιεί αισθητήρες και εστιάζεται σε δεδομένα αισθητήρων που μαζεύονται από ανθρώπους, αντί μηχανές και συνήθως η πληροφορία που προκύπτει είναι προς όφελος κυρίως των απλών πολιτών. Υπάρχουν αρκετά testbeds αυτού του τύπου συμπεριλαμβανομένου και των PhoneLab, Carat project.

PhoneLab: Είναι μια δημόσια πλατφόρμα δοκιμών από κινητά Android που έχει σχεδιαστεί για να απλοποιήσει τα πειράματα μεγάλης κλίμακας στις έξυπνες κινητές συσκευές. Ο στόχος του PhoneLab κυρίως είναι η αντιμετώπιση μερικών κοινών εμποδίων που κάνουν τον πειραματισμό πρόκληση, η προσέλκυση συμμετεχόντων και η συλλογή πολλών δεδομένων (μεγάλης κλίμακας). Ένας σημαντικός περιορισμός είναι ότι η συλλογή δεδομένων έπρεπε να γίνει offline. Παρόλαυτα για την εγκατάσταση μιας εφαρμογής η οποία θα μπορούσε να συλλέγει δεδομένα αισθητήρων που είναι και το θέμα που μας αφορά θα έπρεπε να εξεταστεί από το υπεύθυνο συμβούλιο για να αποφασιστεί αν ο σκοπός και η λειτουργία της εφαρμογής παρέχει οποιαδήποτε θέματα όσον αφορά τη προστασία των προσωπικών δεδομένων των χρηστών.



Σχήμα 2.3.1 Γραφικό περιβάλλον του Carat Project σε Android

Carat project (Σχήμα 2.3.1): Παρόμοια προσπάθεια έγινε και από το Πανεπιστήμιο του Berkeley όπου μέσω δεδομένων που συλλέγονταν από το πλήθος μπορούσαν να πληροφορήσουν το χρήστη για τη κατανάλωση ενέργειας της κινητής συσκευής. Υπάρχει μια ισχυρή υποδομή για τη διατήρηση αυτής της εφαρμογής παρόλαυτα δεν προσφέρεται καταγραφή ούτε εξομοίωση αισθητήρων.



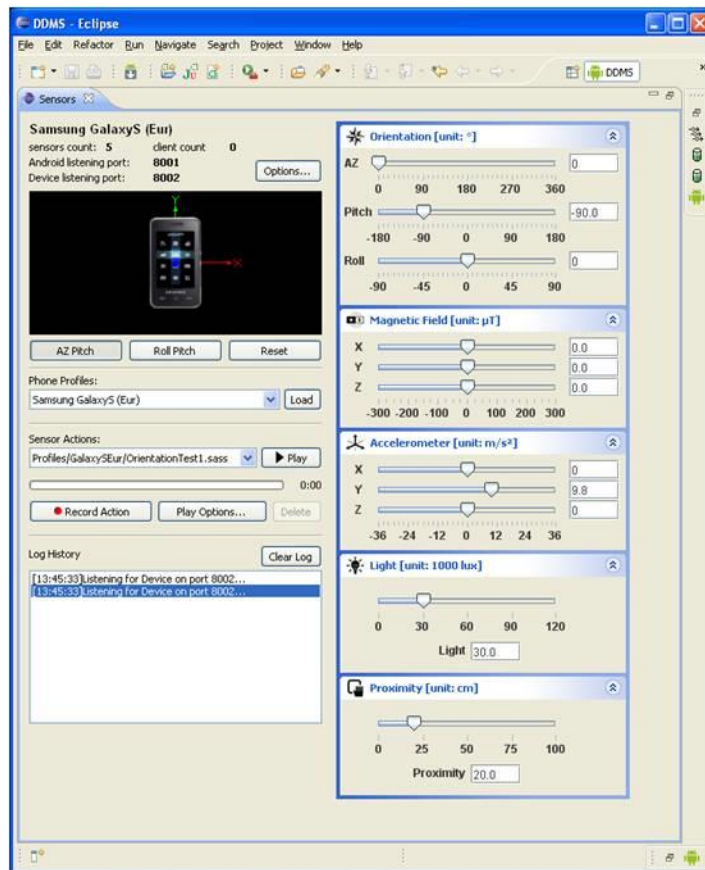
Σχήμα 2.2.1: Το γραφικό περιβάλλον του εξυπηρετητή στο OpenIntents Sensor Simulator

2.4 Android Emulators/Simulators

Παρακάτω είναι τα έργα τα οποία σχετίζονται με Εξομοιωτές και προσομοιωτές Android και συζητούνται για τη στήριξη των αισθητήρων ή την απουσία αυτής.

Sensor Simulator (σχήμα 2.2.1) [17]: Μία Android εφαρμογή που παρέχει τεχνητές μετρήσεις δεδομένων βασισμένες σε συσκευές εισόδου. Ωστόσο αυτή η εφαρμογή δε μπορεί να παράξει κάποια αντιστοίχιση της εικονικής συσκευής με κάποια πραγματική σε πραγματικό χρόνο, αν και πρόσφατα έχει εγκατασταθεί ένα σύστημα καταγραφής και εξομοίωσης στην εφαρμογή. Κάθε εφαρμογή που θέλει να χρησιμοποιήσει τις ενδείξεις

από αυτή τη προσομοίωση πρέπει να τροποποιηθεί έτσι ώστε να μπορεί να επικοινωνεί με αυτό το stand-alone προσομοιωτή για να λάβει τις μετρήσεις που δημιουργούνται.



Σχήμα 2.2.2: Το γραφικό περιβάλλον του εξυπηρετητή στο Samsung Sensor Simulator

SamsungSensorSimulator (Σχήμα 2.2.2) [18]: Είναι ένας προσομοιωτής μόνο για λειτουργικά Windows , Mac. Η προσέγγιση αυτή υποστηρίζει όλους τους αισθητήρες , ίσως και περισσότερους από τη τρέχουσα εργασία, σε πραγματικές συσκευές απομακρυσμένες και συνδεδεμένες με το προσομοιωτή μέσω WIFI. Παρόλα αυτά μελετώντας το σύστημα σε λεπτομέρεια παρατηρήθηκε ότι υπάρχουν αρκετοί περιορισμοί. Δεν αναφέρεται πουθενά αν είναι δυνατόν κάποια εφαρμογή να χρησιμοποιήσει τις μετρήσεις των αισθητήρων που παράγονται (τεχνικές ή από κάποια συσκευή). Αν πράγματι αυτό είναι δυνατό δεν αναφέρεται αν μπορεί να συμβεί σε πραγματικό χρόνο. Επιπλέον επιβάλλει η έκδοση της κινητής συσκευής να είναι Android 2.2 και το λειτουργικό σύστημα του ηλεκτρονικού υπολογιστή όπου θα εγκατασταθεί ο προσομοιωτής να είναι Windows XP Sp2, Windows 7 ή Mac OS 10.6 . Αν είναι δυνατόν για κάποια εφαρμογή που χρησιμοποιεί αισθητήρες να χρησιμοποιήσει δεδομένα αισθητήρων μιας συνδεδεμένης στο σύστημα συσκευής , δεν

είναι σαφές εάν περισσότερες από μια τέτοιες εφαρμογές μπορούν ταυτόχρονα να χρησιμοποιήσουν τις ενδείξεις αυτές σε πραγματικό χρόνο.

QEMU (Quick Emulator) [19]: Ουσιαστικά είναι μια εικονική μηχανή η οποία μιμείται τη κεντρική μονάδα επεξεργασίας μέσω δυναμικής δυαδικής μετάφρασης και επιτρέπει την εκτέλεση ποικιλίας μη τροποποιημένων λειτουργικών συστημάτων, όπως και του Android. Παρέχει επίσης μια επιταχυνόμενη λειτουργία για την υποστήριξη ενός μίγματος δυαδικής μετάφρασης (για το κώδικα του πυρήνα) και της μητρικής εκτέλεσης(για το κώδικα του χρήστη), με τον ίδιο τρόπο όπως το VMware Workstation και το VirtualBox. Το QEMU μπορεί επίσης να χρησιμοποιηθεί αποκλειστικά για την εξομοίωση της κεντρικής μονάδας επεξεργασίας για διεργασίες επιπέδου του χρήστη, επιτρέποντας εφαρμογές που έχουν μεταγλωττιστεί για μία αρχιτεκτονική να τρέχουν σε κάποια άλλη.

SensorEmulator [20]: Είναι μια ενσύρματη προσέγγιση που βασίζεται στην συνδεσιμότητα USB. Δεν μπορεί να λειτουργήσει με απομακρυσμένες κινητές συσκευές, δηλαδή μέσω δικτύου χρησιμοποιώντας WIFI. Επιπλέον, σύμφωνα με τον επικεφαλής του έργου δεν υπήρξε καμία εργασία για μεγάλο χρονικό διάστημα, υπάρχει μια καθυστέρηση στο κομμάτι τις εξομοίωσης, αναπαράστασης των δεδομένων. Επιπλέον αυτός ο εξομοιωτής αισθητήρων υποστηρίζει μόνο επιταχυνσιόμετρο και πυξίδα.



Σχήμα 2.3.1 Οι τρεις άξονες μέτρησης των αισθητήρων κίνησης

2.5 Android Sensors (SDK) [21]

Οι “έξυπνες” κινητές συσκευές έχουν την ικανότητα ανίχνευσης της κίνησης, του προσανατολισμού και αναγνώρισης των πτυχών του περιβάλλοντος (θερμοκρασία ή υγρασία). Ανάλογα με την εφαρμογή υπάρχει τεράστια ποικιλία ως προς τη χρήση αυτών των ικανοτήτων. Ένα παιχνίδι μπορεί να χρησιμοποιήσει τη κλίση της συσκευής για να αλλάξει τη ροή μιας μπάλας, μια εφαρμογή σχεδίασης μπορεί να χρησιμοποιήσει το κούνημα για τον καθαρισμό της τρέχουσας οθόνης. Όλο αυτό επιτυγχάνεται μέσω ενός διαχειριστή των αισθητήρων που είναι εγκατεστημένες στη συσκευή και για το Android αυτό εννοεί τον `SensorManager`. Είναι υπεύθυνος για τη διαχείριση των διαφόρων αναγνώσεων των αισθητήρων και στη συνέχεια τις καθιστά διαθέσιμες για χρήση από προγραμματιστές στις εφαρμογές τους.

Υπάρχουν τρεις μεγάλες κατηγορίες αισθητήρων που υποστηρίζονται από τη πλατφόρμα Android.

Αισθητήρες κίνησης: Επιταχυνσιόμετρο, αισθητήρες βαρύτητας, γυροσκόπιο και αισθητήρες περιστροφής φορέα. Χρησιμοποιούνται για τη μέτρηση δύναμης της επιτάχυνσης και των περιστροφικών δυνάμεων κατά μήκος τριών αξόνων.

Περιβαλλοντικοί αισθητήρες: Βαρόμετρο, φωτόμετρο και θερμόμετρο. Χρησιμοποιούνται κυρίως για τη μέτρηση των περιβαλλοντικών πτυχών, όπως η θερμοκρασία, η πίεση, ο φωτισμός, η υγρασία.

Αισθητήρες θέσης: Αισθητήρες προσανατολισμού και μαγνητόμετρο. Χρησιμοποιούνται για το καθορισμό της φυσικής θέσης της συσκευής.

Εύρος υποστηριζόμενων αισθητήρων

Οι παρακάτω αισθητήρες υποστηρίζονται από το Android, από το Jelly Bean 2 δηλαδή από τις εκδόσεις 4.2 και άνω.

- Επιταχυνσιόμετρο
- Θερμοκρασία περιβάλλοντος
- Βαρύτητα
- Γυροσκόπιο
- Φώς
- Γραμμική επιτάχυνση
- Μαγνητικό πεδίο
- Προσανατολισμός
- Πίεση
- Εγγύτητα
- Υγρασία
- Διάνυσμα περιστροφής

Sensor Rates

Η καθυστέρηση των δεδομένων (ή ρυθμός δειγματοληψίας) ελέγχει το χρονικό διάστημα κατά το οποίο οι εκδηλώσεις του αισθητήρα αποστέλλονται με την αίτηση του χρήστη μέσω της μεθόδου `onSensorChanged()`. Η προεπιλεγμένη καθυστέρηση λήψης των δεδομένων είναι κατάλληλη για τη παρακολούθηση των τυπικών αλλαγών προσανατολισμού της οθόνης και χρησιμοποιεί μια καθυστέρηση των 2000 μικροδευτερολέπτων. Ο προγραμματιστής μπορεί να καθορίσει άλλες καθυστερήσεις δεδομένων όπως:

- `SENSOR_DELAY_GAME` (20.000 μικροδευτερόλεπτα καθυστέρηση)
- `SENSOR_DELAY_UI` (60.000 μικροδευτερόλεπτα καθυστέρηση)
- `SENSOR_DELAY_FASTEST` (0 καθυστέρηση)

Η καθυστέρηση που καθορίζεται είναι μόνο μια προτεινόμενη καθυστέρηση. Το σύστημα Android και άλλες εφαρμογές μπορούν να αλλάξουν αυτή τη καθυστέρηση. Ως βέλτιστη πρακτική, προτείνεται στο προγραμματιστή να καθορίζει τη μεγαλύτερη καθυστέρηση γιατί το σύστημα χρησιμοποιεί συνήθως μια μικρότερη καθυστέρηση από αυτό που ορίζει ο ίδιος. Χρησιμοποιώντας μια μεγαλύτερη καθυστέρηση επιβάλλει ένα μικρότερο φορτίο στον επεξεργαστή και ως εκ τούτου χρησιμοποιεί λιγότερη ενέργεια.

Κεφάλαιο 3

Αρχιτεκτονική Υποσυστήματος

| | | |
|-----|---|----|
| 3.1 | Εισαγωγή κεφαλαίου | 20 |
| 3.2 | Υφιστάμενη αρχιτεκτονική SmartLab | 20 |
| 3.3 | Υποσύστημα RM Mockups ως επέκταση του Device Server | 22 |
| 3.4 | SLSensorManager library | 22 |
| 3.5 | Αποθετήριο Μεγάλων δεδομένων | 23 |

3.1 Εισαγωγή κεφαλαίου

Στο κεφάλαιο αυτό θα γίνει μια περιγραφή στην υφιστάμενη αρχιτεκτονική του SmartLab. Στη συνέχεια για κάθε κομμάτι της αρχιτεκτονικής του SmartLab θα αναφερθεί η επέκταση που έγινε για τις ανάγκες της εγκατάστασης και λειτουργίας του υποσυστήματος προσομοιώσεων.

3.2 Υφιστάμενη αρχιτεκτονική SmartLab

Το SmartLab είναι μια καινοτόμα πλατφόρμα δοκιμών για έξυπνες κινητές συσκευές που επιτρέπει τον πειραματισμό χρησιμοποιώντας εικονικές και πραγματικές συσκευές. Ξεκινάμε με μια σύντομη περιγραφή της υποδομής του σε υψηλό επίπεδο (διεπαφή χρήστη (user interface), device server και υλικού (hardware)).

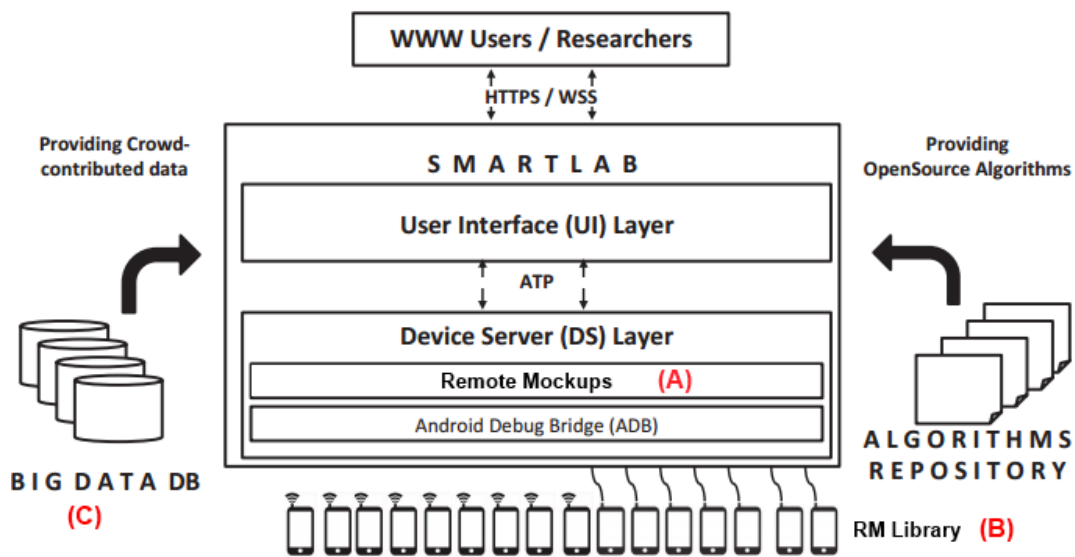


Σχήμα 3.2.1: Η διεπαφή χρήστη του SmartLab που διευκολύνει το χρήστη στη χρήση όλων των υποσυστημάτων καθώς επίσης και του υποσυστήματος προσομοιώσεων

User interface: Το SmartLab χρησιμοποιεί μια σαφή, συνοπτική και εύχρηστη διεπαφή που επιτρέπει στο χρήστη διάφορους τρόπους αλληλεπίδρασης με τις συνδεδεμένες κινητές συσκευές όπως απεικονίζεται στο σχήμα 3.2.1. Αυτή η αλληλεπίδραση γίνεται είτε μέσω WebSockets για υπηρεσίες υψηλής συχνότητας είτε μέσω AJAX based επικοινωνίας για μη συχνές υπηρεσίες. Για το σκοπό αναπαράστασης και διαδραστικότητας του χρήστη με το υποσύστημα που δημιουργήθηκε για τη παρούσα διπλωματική προστέθηκε το παράρτημα RM Mockups στη διεπαφή χρήστη το οποίο διατηρεί μια επικοινωνία μέσω WebSockets με το υπόλοιπο υποσύστημα εξομοίωσης .

Device Server Layer: Ο διακομιστής συσκευών φιλοξενεί όλα τα υποσυστήματα του SmartLab και διευκολύνει τη σύνδεση των πραγματικών ή εικονικών συσκευών με το περιβάλλον χρήστη. Σήμερα χρησιμοποιείται το λειτουργικό CentOS 6.5 X64 με 4x2,4GHz εικονικούς επεξεργαστές ,8GB RAM και 80 GB σκληρό δίσκο.

Hardware Layer: Οι κινητές συσκευές και το Datacenter αποτελούν το βασικό υλικό του SmartLab. Το τελευταίο περιλαμβάνω πάνω από 16 TB μνήμη αποθήκευσης από RAID-5 /SSD σε ένα υπολογιστή IBM X3550 καθώς και 320 GB κύρια μνήμη στα 5 racks πολλαπλών IBM/HP .



Σχήμα 3.2.2 :Αρχιτεκτονική SmartLab και επέκταση αυτής με την εγκατάσταση του υποσυστήματος προσομοιώσεων

3.3 Υποσύστημα RM Mockups ως επέκταση του Device Server (Σχήμα 3.2.1 A)

Ένα ανεξάρτητο κομμάτι λογισμικού προστέθηκε στον Device Server το οποίο είναι υπεύθυνο για τη λειτουργία καταγραφής και εξομοίωσης και επικοινωνίας του συστήματος με τις συσκευές που είναι ενσωματωμένες στο SmartLab. Η πρόθεση του συστήματος όσον αφορά τη καταγραφή των αισθητήρων είναι να διαβάζει γρήγορα, αποδοτικά και αξιόπιστα τις μετρήσεις σε πραγματικό χρόνο. Υποστηρίζει πολυνηματική καταγραφή αισθητήρων από πολλές κινητές συσκευές ταυτόχρονα. Σε περιπτώσεις σφαλμάτων επικοινωνίας το σύστημα έχει σχεδιαστεί έτσι ώστε να μπορεί να επαναφέρει τον RM server ,επέκταση του Device Server σε μια υγιή κατάσταση διατηρώντας έτσι μια ομαλή λειτουργία στα υπόλοιπα συστήματα του SmartLab

3.4 SLSensorManager library (Σχήμα 3.2.1 B)

Στο πλαίσιο του λειτουργικού Android ,το Mockup (όπως ανέφερα σε προηγούμενο κεφάλαιο) αναφέρεται στη διαδικασία επέκτασης ενός AVD ή συγκεκριμένου αισθητήρα ή GPS ARD με προσαρμοσμένες τιμές. Επιπρόσθετα ένα σημαντικό

προνόμιο του συστήματος των Mockups (κυρίως της εξομοίωσης) είναι το ότι μπορεί να υποστηρίξει την εξομοίωση αισθητήρων που μπορεί το υλικό να μην προσφέρει στη συγκεκριμένη συσκευή ARD (π.χ. NFC, Wifi Direct, κτλ). Προκειμένου να παρέχει τη δυνατότητα υποστήριξης τόσο αισθητήρων GPS όσο και άλλων αισθητήρων δημιουργήθηκε μια προσαρμοσμένη ενότητα(module) η SLSensorManager. Η SLSensorManager είναι μια βιβλιοθήκη η ενσωματώνεται στην εφαρμογή η οποία επιθυμεί να κάνει εξομοίωση αισθητήρων μέσω σύνδεσης υποδοχών (socket connection) με τον RM Server (υποσύστημα του Device server). Μέσω της σύνδεσης αυτής ο RM Server θα τροφοδοτεί τις συσκευές με GPS ή μετρήσεις άλλων αισθητήρων κωδικοποιημένες σε μορφή JSON και στη συνέχεια τις αποθηκεύει σε μία μεγάλη αποθήκη δεδομένων . Δεδομένου ότι αυτή η λειτουργία είναι εντελώς έξω από το ρεύμα αλληλεπίδρασης του ADB παρέχουμε σε κάθε προγραμματιστή τη βιβλιοθήκη αυτή. Με τα εργαλεία Android r18 και λειτουργικό Android 4.0, οι προγραμματιστές στις μέρες μας έχουν την ευκαιρία να ανακατευθύνουν πραγματικές μετρήσεις αισθητήρων , που παράγονται από πραγματικές έξυπνες κινητές συσκευές προς τις εικονικές συσκευές για περαιτέρω επεξεργασία. Είναι σημαντικό να αναφερθεί ότι αυτή η λειτουργικότητα είναι το αντίθετο από αυτό που προσφέρεται από το υποσύστημα RM Mockups. Στη περίπτωση μας, θέλουμε να είμαστε σε θέση να ανακατευθύνουμε δεδομένα από ένα αρχείο κειμένου με μια ARD, έτσι ώστε ένα δεδομένο πείραμα πάνω σε πραγματικές συσκευές ή εικονικές να χρησιμοποιεί δεδομένα αισθητήρων από το αρχείο . Η καταγραφή των δεδομένων των αισθητήρων και αποθήκευση τους σε αρχεία κειμένου ή σε βάση δεδομένων μπορεί να πραγματοποιηθεί πολύ εύκολα με μια ποικιλία εργαλείων.

3.5 Αποθετήριο Μεγάλων δεδομένων (Σχήμα 3.2.1 C)

Για την αποθήκευση αλλά και γρήγορη ανάκτηση των δεδομένων των αισθητήρων χρησιμοποιήθηκε η Couchbase ως NOSQL βάση δεδομένων. Μία βάση δεδομένων NOSQL ή Not only SQL παρέχει μηχανισμούς για την αποθήκευση και την ανάκτηση δεδομένων διαφορετικούς από αυτούς που προσφέρουν οι σχεσιακές βάσεις δεδομένων. Κίνητρο για τη χρησιμοποίηση τέτοιου είδους βάσης δεδομένων ήταν η απλότητα του σχεδιασμού, η επεκτασιμότητα και ο καλύτερος έλεγχος διαθεσιμότητας που παρέχει. Η δομή των δεδομένων διαφέρει πολύ αυτή των σχεσιακών βάσεων αφού είναι κλειδί-

τιμή (key-value) οπότεν δεν θα χρειαζόταν καμία αλλαγή η μετατροπή των δεδομένων μας για εισαγωγή τους στη βάση, αντίθετα με τις σχεσιακές βάσεις που θα χρειαζόταν τροποποίηση των δεδομένων κατά την εισαγωγή και ανάκτηση των δεδομένων από του πίνακες.

Κεφάλαιο 4

Υλοποίηση Υποσυστήματος Προσομοιώσεων

| | | |
|-------|--|----|
| 4.1 | Εισαγωγή | 25 |
| 4.2 | SmartLab Device Server σε Java | 26 |
| 4.3 | Σύστημα προσομοιώσεων πάνω από το Device Server του SmartLab | 27 |
| 4.3.1 | TCPServer (Καταγραφή) | 27 |
| 4.3.2 | EmulationServer (Εξομοίωση) | 28 |
| 4.3.3 | EmulationClient (Εξομοίωση) | 29 |
| 4.4 | Παράρτημα Συστήματος στην ιστοσελίδα SmartLab | 30 |
| 4.5 | SmartLabRecorder (Καταγραφή) | 31 |
| 4.6 | Βιβλιοθήκη SLSensorManager (Σχήμα 4.6) | 31 |
| 4.7 | Σύνοψη διαδικασίας καταγραφής και εξομοίωσης. | 34 |
| 4.8 | Προβλήματα που προέκυψαν και αντιμετώπισή τους | 36 |

4.1 Εισαγωγή

Το κεφάλαιο αυτό επικεντρώνεται στις βασικές υποενότητες του υποσυστήματος.

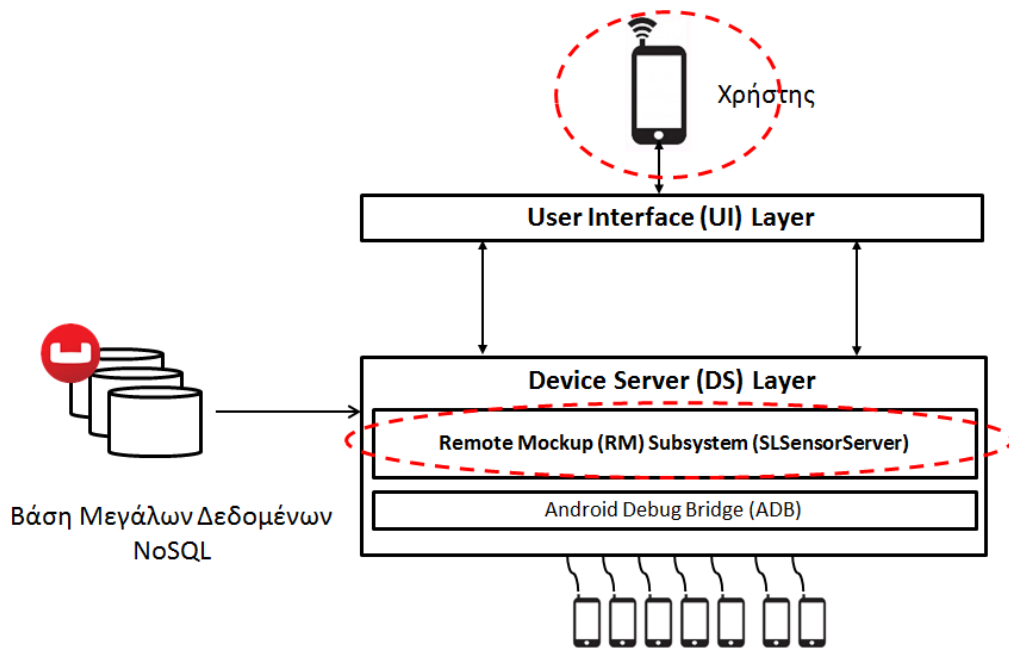
Γίνεται μια αναφορά στην υλοποίηση του εξυπηρετητή που βρίσκεται μέσα στον Device Server του SmartLab, ο οποίος στην ουσία είναι ο συντονιστής των περισσότερων διαδικασιών που αφορούν την καταγραφή αλλά και την εξομοίωση. Στη συνέχεια γίνεται μία ανάλυση της εφαρμογής καταγραφής των αισθητήρων η οποία εγκαθιστάτε στη κινητή συσκευή Android. Έπειτα γίνεται μία αναφορά στη βιβλιοθήκη η οποία εγκαθιστάτε στις κινητές συσκευές που θα τροφοδοτούνται από δεδομένα αισθητήρων και είναι το βασικότερο κομμάτι λογισμικού που αναπτύχθηκε στα πλαίσια

της παρούσας διπλωματικής. Τέλος γίνεται μια σύνοψη των ενεργειών ώστε να πετύχει ο χρήστης καταγραφή αλλά και εξομοίωση αισθητήρων.

4.2 SmartLab Device Server σε Java

Ο εξυπηρετητής έχει γραφτεί στη γλώσσα προγραμματισμού JAVA. Ένα από τα βασικά πλεονεκτήματα της JAVA έναντι των περισσότερων άλλων γλωσσών είναι η ανεξαρτησία του λειτουργικού συστήματος και πλατφόρμας. Τα προγράμματα που είναι γραμμένα σε JAVA τρέχουν ακριβώς το ίδιο σε Windows, Linux, Unix και Macintosh και δεν χρειάζεται να ξαναγίνει μεταγλώττιση ή να αλλάξει ο πηγαίος κώδικας για κάθε διαφορετικό λειτουργικό σύστημα. Αυτό επιτυγχάνεται με τη Εικονική μηχανή της JAVA (JVM) η οποία μεταφράζει σε συμβολικό κώδικα (assembly) το κώδικα υψηλού επιπέδου ανάλογα με τη ΚΜΕ του συγκεκριμένου υπολογιστή. Ωστόσο για να το πετύχει αυτό η JAVA είναι πιο αργή σε σχέση με άλλες προγραμματιστικές γλώσσες υψηλού επιπέδου (high level) όπως C, C++. Παρόλ' αυτά οι τελευταίες εκδόσεις του javac με τη χρήση της τεχνολογίας Hot Spot έχουν καταφέρει αξιόλογες επιδόσεις που πλησιάζουν ή και ξεπερνούν σε μερικές περιπτώσεις τον εγγενή κώδικα. Επίσης η JAVA έχει τη δυνατότητα ανάπτυξης πολυνηματικών προγραμμάτων όπου διάφορες λειτουργίες μπορούν να εκτελούνται ταυτόχρονα στο πλαίσιο ενός προγράμματος. Σε σύγκριση με τις υπόλοιπες υψηλού επιπέδου γλώσσες προγραμματισμού η χρήση νημάτων στη JAVA είναι πολύ πιο εύκολη.

Ο συγκεκριμένος εξυπηρετητής είναι υπεύθυνος για όλη την επικοινωνία μεταξύ κινητών και της ιστοσελίδας SmartLab μέσω της γέφυρας συσκευών Android (adb) καθώς επίσης και άλλων αναγκαίων λειτουργιών που αφορούν το SmartLab.



Σχήμα 4.3 Βασικά στοιχεία του υποσυστήματος προσομοιώσεων

4.3 Σύστημα προσομοιώσεων πάνω από το Device Server του SmartLab

Το Remote Mockup σύστημα έχει ενσωματωθεί στον συγκεκριμένο εξυπηρετητή εκτελώντας τις απαραίτητες λειτουργίες του ανεξαρτήτως των άλλων λειτουργιών του εξυπηρετητή αυτού. Για τις ανάγκες του πρωτοκόλλου γίνονται reserved είκοσι θύρες (10000-10020 π.χ.) για τη φάση της καταγραφής των αισθητήρων καθώς επίσης και είκοσι θύρες (10021-10041 π.χ.) για τη φάση της εξομοίωσης κάθε διαφορετικής καταγραφής.

4.3.1 TCPServer (Καταγραφή)

Για τις ανάγκες της καταγραφής των διαφόρων αισθητήρων έχει υλοποιηθεί μια κλάση σε JAVA, η TCPServer. Κάθε διαφορετική καταγραφή αισθητήρων από χρήστες του SmartLab εκτελείται σε ξεχωριστό νήμα. Αρχικά γίνεται μια αναγνώριση της συμμόρφωσης του δικτύου και ανάλογα καθορίζεται και το μέγεθος του παραθύρου,

```

{
  'phone_id': 'SHOAPPL00803',
  'user_id': 'smartlab'
  [
    {
      't': '1391198355678',
      's': 'proximity',
      'v': 0
    },
    {
      't': '1391198355678',
      's': 'accelerometer',
      'v': [ '9.00', '1.23', '4.65' ]
    },
    {
      't': '1391198355678',
      's': 'gravity',
      'v': [ '0.00', '0.00', '0.00' ]
    },
    {
      't': '1391198355678',
      's': 'orientation',
      'v': [ '31.0', '6.0', '60.0' ]
    },
    {
      't': '1391198355678',
      's': 'rotation vector',
      'v': 0
    },
    {
      't': '1391198355678',
      's': 'light',
      'v': 0
    },
    {
      't': '1391198355678',
      's': 'gyroscope',
      'v': [ '0.0', '0.0', '0.0' ]
    },
    {
      't': '1391198355680',
      's': 'gravity',
      'v': [ '0.00', '0.00', '0.00' ]
    },
    {
      't': '1391198355678',
      's': 'magnetic field',
      'v': [ '-42.76', '27.27', '-8.58' ]
    },
    {
      't': '1391198355678',
      's': 'temperature',
      'v': 0
    }
  ]
}

```

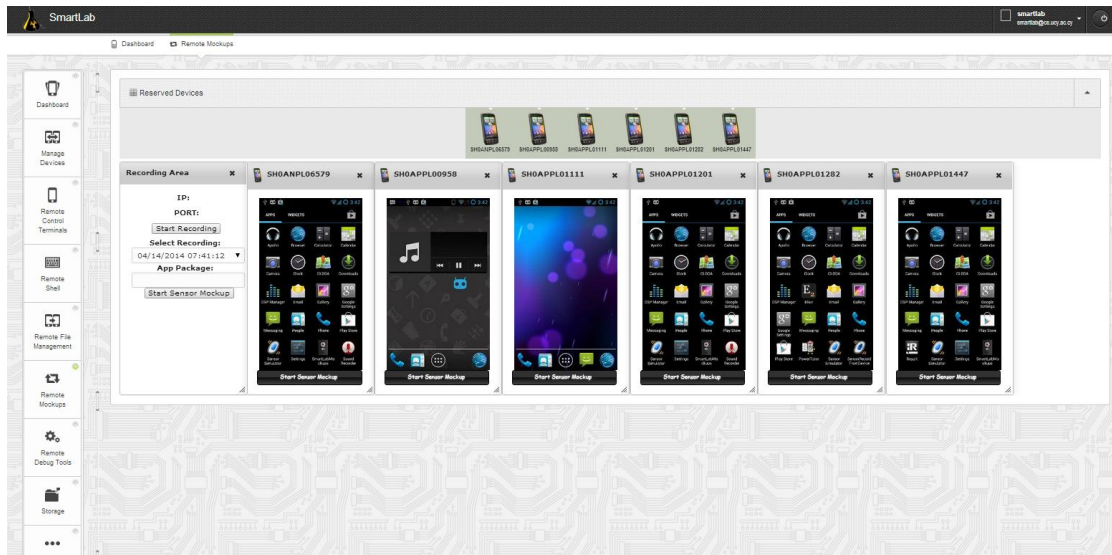


Σχήμα 4.3.1 : Κωδικοποίηση των δεδομένων των αισθητήρων σε μορφή JSON

δηλαδή πόσα δεδομένα θα αποστέλλονται ανά πάσα στιγμή στη βάση Couchbase για καταχώρηση. Αυτό έχει ως αποτέλεσμα σε κάθε καταγραφή να μην χρησιμοποιείτε μνήμη περισσότερη απ' όσο το συγκεκριμένο παράθυρο, πράγμα πολύ σημαντικό αφού σε περίπτωση κατάχρησης της μνήμης που διαθέτει το λειτουργικό σύστημα για το συγκεκριμένο πρόγραμμα προς εκτέλεση και ξεπέρασης αυτής το λειτουργικό σύστημα θα τερματίσει τον εν λόγω πρόγραμμα. Οπότε όταν έχουν καταγραφεί δεδομένα όσο το μέγεθος του παραθύρου δημιουργούνται JSON objects της μορφής (Σχήμα 4.3.1) και αποστέλλονται στη βάση για αποθήκευση. Στο τέλος της καταγραφής το συγκεκριμένο νήμα τερματίζει απελευθερώνοντας τη θύρα την οποία κρατούσε.

4.3.2 EmulationServer (Εξομοίωση)

Μία συγκεκριμένη καταγραφή μπορεί να εξομοιωθεί σε όσα κινητά έχει κρατήσει (reserved) ένας χρήστης του SmartLab. Η κλάση λοιπόν αυτή που είναι υπεύθυνη για την ανάθεση μιας καταγραφής στις διάφορες κινητές συσκευές είναι η EmulationServer κλάση. Έχοντας πάρει τις απαραίτητες πληροφορίες για να μπορεί να ανακτήσει τα δεδομένα που χρειάζονται για μια συγκεκριμένη καταγραφή (recordid,userid) από τη βάση δεδομένων τότε είναι σε αναμονή περιμένοντας τις όποιες κινητές συσκευές έχει



Σχήμα 4.4.1: Γραφικό περιβάλλον συστήματος προσομοίωσης στην ιστοσελίδα του SmartLab

κρατήσει από το SmartLab ο συγκεκριμένος χρήστης να κάνουν μια σύνδεση μέσω sockets προς αυτήν.

4.3.3 EmulationClient (Εξομοίωση)

Κάθε κινητή συσκευή η οποία θα εξομοιώσει μια συγκεκριμένη καταγραφή, βάση του πρωτοκόλλου αρχικά θα δημιουργήσει μια TCP σύνδεση μέσω sockets με τον εξυπηρετητή κινητών συσκευών. Ο εν λόγω εξυπηρετητής θα εκτελέσει μια εντολή προς τον adb αναθέτοντας στη συγκεκριμένη κινητή συσκευή το ip ,port στα οποία θα πρέπει να δημιουργήσει τη σύνδεση καθώς επίσης και το πακέτο ,δηλαδή ποιο πρόγραμμα θα εκκινήσει .Όπως έχω προαναφέρει η σύνδεση αυτή θα γίνει με τον EmulationServer ο οποίος έχει ενσωματωθεί μέσα στον εξυπηρετητή κινητών συσκευών. Η κλάση EmulationClient τότε είναι υπεύθυνη για την ανάκτηση των δεδομένων των αισθητήρων από τη βάση δεδομένων και την αποστολή αυτών στη κινητή συσκευή με την οποία έχει κάνει τη σύνδεση.



Σχήμα 4.4.2: Διαδικασία που ακολουθείται για να γίνει καταγραφή δεδομένων των αισθητήρων από τη κινητή συσκευή του χρήστη

4.4 Παράρτημα Συστήματος στην ιστοσελίδα SmartLab

Το συγκεκριμένο τμήμα της σελίδας SmartLab έχει δημιουργηθεί για να εξυπηρετήσει το όλο σύστημα καταγραφής και αναπαράστασης της εξομοίωσης από τις κινητές συσκευές. Στο συγκεκριμένο τμήμα εμφανίζονται οι οθόνες των κινητών τηλεφώνων που έχει κρατήσει από το SmartLab ο συγκεκριμένος χρήστης και έχει τη δυνατότητα να εκτελέσει καταγραφή αισθητήρων από το κινητό του τηλέφωνο καθώς επίσης να εξομοιώσει σε οποιοδήποτε από τις κρατημένες κινητές συσκευές κάποια δική του καταγραφή . Όλες οι καταγραφές που έχει αποθηκεύσει ο συγκεκριμένος χρήστης παλαιότερα εμφανίζονται σε αναπτυσσόμενη λίστα και έχουν ως όνομα την ημερομηνία και ώρα που πραγματοποιήθηκε η εγγραφή. Ο χρήστης έχει τη δυνατότητα να αναπαραστήσει οποιαδήποτε από τις αποθηκευμένες καταγραφές σε οποιαδήποτε κινητή συσκευή επιθυμεί ,ταυτόχρονα, χρησιμοποιώντας και οποιαδήποτε διαφορετική εφαρμογή αν και εφόσον η κινητή συσκευή έχει εγκατεστημένη την εν λόγω εφαρμογή.

4.5 SmartLabRecorder (Καταγραφή)

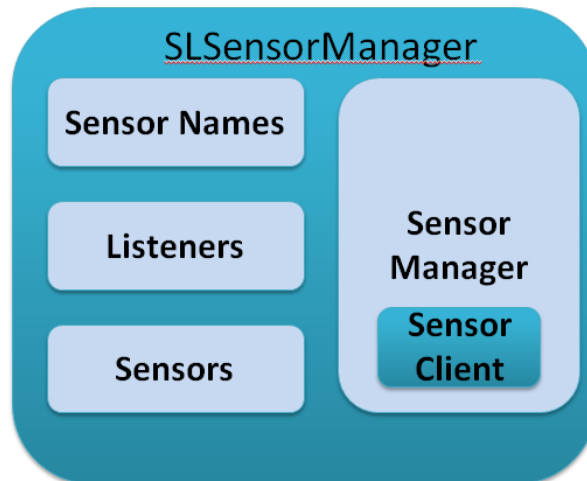
Η **SmartLabRecorder** είναι μία εφαρμογή γραμμένη σε JAVA για λειτουργικό Android. Ο χρήστης καταχωρεί τη διεύθυνση ip και θύρα στην οποία θα αποστέλλει τα δεδομένα των αισθητήρων που έχει επιλέξει προς καταγραφή. Όπως φαίνεται και από το σχήμα η ιστοσελίδα του SmartLab παρέχει στο χρήστη μια διεύθυνση IP και τη θύρα στην οποία πρέπει να συνδεθεί η κινητή συσκευή ώστε να ξεκινήσει η καταγραφή των αισθητήρων.

Αισθητήρες που υποστηρίζει η εν λόγω εφαρμογή

- Επιταχυνσιόμετρο
- Βαρύτητα
- Γυροσκόπιο
- Φώς
- Γραμμική επιτάχυνση
- Μαγνητικό πεδίο
- Προσανατολισμός
- Πίεση
- Εγγύτητα
- Υγρασία

4.6 Βιβλιοθήκη SLSensorManager (Σχήμα 4.6)

Η συγκεκριμένη βιβλιοθήκη είναι το βασικότερο στοιχείο για τη πραγματοποίηση της εξομοίωσης. Έχει υλοποιηθεί σε JAVA για περιβάλλον Android. Η εν λόγω βιβλιοθήκη ενσωματώνεται στην εφαρμογή η οποία θα χρησιμοποιηθεί για εξομοίωση και χωρίς να χρειάζονται ιδιαίτερες αλλαγές στο κώδικα της εφαρμογής, όσοι αισθητήρες έχουν εγγραφεί τότε θα λαμβάνουν δεδομένα από το Remote Mockup System μέσω της σύνδεσης TCP που έχει καθιερωθεί από τη κινητή συσκευή προς τον EmulationClient. Υποστηρίζει όλους του αισθητήρες που υποστηρίζει και η εφαρμογή καταγραφής με τη διαφορά πως ακόμα και αν η συγκεκριμένη συσκευή δεν υποστηρίζει κάποιο από τους αισθητήρες που εξομοιώνονται αυτό δεν επηρεάζει τη λειτουργία αφού η εξομοίωση θα γίνει κανονικά και τα δεδομένα θα χρησιμοποιηθούν από την εφαρμογή χωρίς κανένα



Σχήμα 4.6.1: Τα βασικά στοιχεία της βιβλιοθήκης η οποία ενσωματώνεται στις κινητές συσκευές που θα χρησιμοποιηθούν για εξομοίωση

πρόβλημα αφού στα πλαίσια της εξομοίωσης η εφαρμογή θα υποστηρίζει όλους τους αισθητήρες γιατί χρησιμοποιείται ο SLSensorManager αντί ο αντίστοιχος του Android.

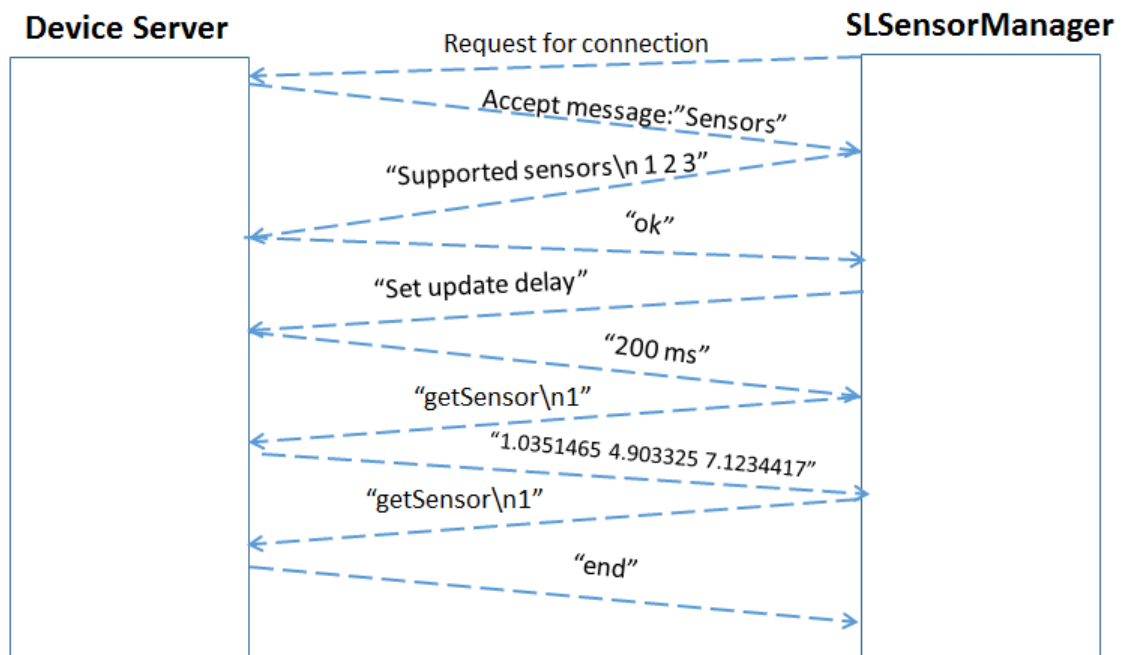
Βασικά στοιχεία (components) της βιβλιοθήκης.

SensorManagerSimulator

Η κλάση αυτή αντικαθιστά τον SensorManager του λειτουργικού Android. Είναι το βασικό στοιχείο του πρωτοκόλλου της εξομοίωσης αφού είναι υπεύθυνη για το registration των αισθητήρων , τη δημιουργία ενός αντικειμένου SensorSimulationClient υπεύθυνο για την δημιουργία σύνδεσης TCP προς τον EmulationClient.

SensorSimulationClient

Ο SensorSimulationClient είναι υπεύθυνος για τη σύνδεση και αποσύνδεση του SensorManagerSimulator. Χειρίζεται επίσης όλες τις εισόδους και εξόδους και δημιουργεί μια TCP σύνδεση μεταξύ της κλάσης αυτής και του EmulationClient. Η κλάση αυτή διαθέτει όλες τις μεθόδους που χρειάζονται για αποστολή και λήψη δεδομένων. Επίσης μέσα της υλοποιείται και μια δική μας κλάση Listener. Κάθε φορά που ένας νέος αισθητήρας ενεργοποιείται δημιουργείται ένα νέο instance αυτής της κλάσης και αποθηκεύεται μέσα στον SensorSimulationClient. Με αυτό τον τρόπο



Σχήμα 4.6.2: Το πρωτόκολλο που ακολουθείται από τη κινητή συσκευή προς τον εξυπηρετητή που βρίσκεται στο Device Server του SmartLab κατά τη διάρκεια της εξομοίωσης

γνωρίζουμε ποιοι αισθητήρες έχουν registered listeners και ποιος listener αντιπροσωπεύει ποιο αισθητήρα.

Βασικές μέθοδοι του πρωτοκόλλου εξομοίωσης

Connect()

Μέθοδος του SensorSimulationClient η οποία δοσμένου σαν είσοδο το IP και port στην εφαρμογή θα δημιουργήσει μια TCP σύνδεση από την εφαρμογή ουσιαστικά προς τον EmulationClient.

GetSensors()

Μέθοδος στον SensorSimulationClient η οποία χρησιμοποιείται μετά την εδραίωση της σύνδεσης στη λήψη της ένδειξης για το ποιοι αισθητήρες χρησιμοποιούνται στην εν λόγω καταχώρηση η οποία πρόκειται να εξομοιωθεί έτσι ώστε να έχουμε γνώση για το από ποιους αισθητήρες αναμένουμε λήψη δεδομένων.

ReadSensor()

Μέθοδος η οποία λαμβάνει τα δεδομένα ενός αισθητήρα από το δίκτυο.

SetSensorUpdateDelay()

Μέθοδος η οποία καθορίζει τη καθυστέρηση με την οποία θα ζητά η βιβλιοθήκη δεδομένα για τον εν λόγω αισθητήρα από τον EmulationClient

UnsetSensorUpdateRate()

Η μέθοδος που χρησιμοποιείται στον SensorSimulationClient σαν ένδειξη του πότε πρέπει να τερματίσει τη σύνδεση αφού τα δεδομένα που είχε αποθηκευμένα η καταγραφή η οποία τροφοδοτούσε τους αισθητήρες έχουν αποσταλεί και δεν υπάρχουν επιπλέον δεδομένα.

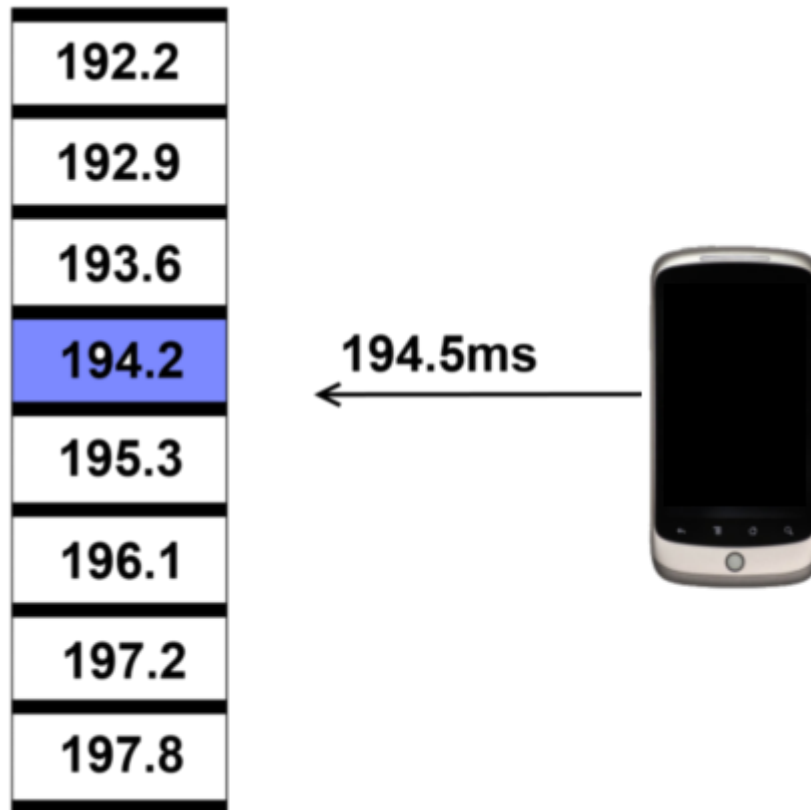
4.7 Σύνοψη διαδικασίας καταγραφής και εξομοίωσης.

Διαδικασία καταγραφής

- i. Ο χρήστης λαμβάνει από τη ιστοσελίδα τη διεύθυνση IP και PORT
- ii. Καταχωρεί στην εφαρμογή SmartLab Recorder το IP και PORT
- iii. Επιλέγει τους αισθητήρες που επιθυμεί να καταγράψει
- iv. Ξεκινά τη καταγραφή
- v. Σταματά τη καταγραφή

Διαδικασία εξομοίωσης

- i. Ο χρήστης επιλέγει από την ιστοσελίδα τη καταγραφή που επιθυμεί να εξομοιώσει.
- ii. Πληκτρολογεί το πακέτο της εφαρμογής (δηλαδή ποια εφαρμογή επιθυμεί να τρέξει στις κινητές συσκευές η οποία χρησιμοποιεί δεδομένα αισθητήρων και έχει εγκατεστημένη τη βιβλιοθήκη)
- iii. Πατά το κουμπί “Start Sensor Mockup” κάτω από τη κινητή συσκευή στην ιστοσελίδα ώστε να ξεκινήσει η εξομοίωση.



Σχήμα 4.8: Αναπαράσταση της χρήσης αλγορίθμου εύρεσης κοντινότερων τιμών σε σχέση με τις τιμές που ζητούνται

4.8 Προβλήματα που προέκυψαν και αντιμετώπισή τους

Πλέον οι κινητές συσκευές λαμβάνουν με πολύ γρήγορους ρυθμούς δεδομένα από τους αισθητήρες. Αν σκεφτούμε ότι σε 1 δευτερόλεπτο έχουμε δεδομένα 1000 μετρήσεων και κάθε μέτρηση είναι έχοντας όλη τη πληροφορία γύρω στα 0.3 kb τότε για μία κινητή συσκευή έχουμε 15 mb δεδομένα το λεπτό. Αν καταγράψουμε μετρήσεις για μια ώρα και 10 κινητές συσκευές ταυτόχρονα τότε πρέπει να διαχειριστούμε περίπου 10 GB δεδομένα. Δεν θα μπορούσε ο εξυπηρετητής να αντεπεξέλθει σε τόσο μεγάλο αριθμό δεδομένων και αυτό ήταν ένα σοβαρό πρόβλημα που προέκυψε στη διαδικασία καταγραφής και εξομοίωσης. Για την αντιμετώπιση του προβλήματος αυτού υλοποιήθηκε μια τεχνική η οποία εγγυόταν πως ανά πάσα χρονική στιγμή είτε στην καταγραφή είτε στην εξομοίωση ο εξυπηρετητής θα έπρεπε να διαχειριστεί συγκεκριμένο και ανεκτό αριθμό δεδομένων. Η τεχνική αυτή αρχικά μετρούσε τη πιθανή

καθυστέρηση του δικτύου και έθετε μια σταθερά η οποία καθόριζε το μέγεθος των δεδομένων που θα μπορούσαν ανά πάσα χρονική στιγμή να βρίσκονται στον εξυπηρετητή. Στη συνέχεια στη περίπτωση της εξομοίωσης τα δεδομένα ανακτώνταν από τη βάση τοποθετούνταν σε δομές και ακολούθως αποστέλλονταν στη κινητή συσκευή. Όταν περίπου τα μισά δεδομένα είχαν αποσταλεί τότε ένα διαφορετικό νήμα ήταν υπεύθυνο να ανακτήσει το επόμενο κομμάτι δεδομένων ίδιου μεγέθους. Αφού αποστέλλονταν όλα τα δεδομένα του αρχικού νήματος διαγράφονταν και χρησιμοποιούνταν τα νέα δεδομένα, και ούτω καθεξής. Προτού υλοποιηθεί αυτή η τεχνική τα δεδομένα των αισθητήρων κατατάσσονταν σε KDTree ένα δέντρο που βοηθούσε στη γρήγορη αναζήτηση κοντινών γειτόνων (στη περίπτωση μας τη κοντινότερη χρονοσφραγίδα). Παρόλαυτα εφόσον πλέον τα δεδομένα που έπρεπε να διαχειριστεί ο εξυπηρετητής θα ήταν πολύ λιγότερα ανά πάσα χρονική στιγμή υπήρχε μεγάλη καθυστέρηση στο κτίσιμο του δέντρου πράγμα που πλέον θα γινόταν συχνότερα και θα χρησιμοποιούταν για πολύ λιγότερο χρόνο. Οπότε υλοποιήθηκε ένας brute-force αλγόριθμος για την αναζήτηση της κοντινότερης χρονοσφραγίδας από αυτή που ζητήθηκε.

Κεφάλαιο 5

Πειραματική Μελέτη

| | | |
|-----|---|----|
| 5.1 | Εισαγωγή | 38 |
| 5.2 | Πειραματική Υποδομή | 39 |
| 5.3 | Πειραματική Σειρά 1: Ανάπτυξη Πειραμάτων μέσω του υποσυστήματος προσομοιώσεων | 41 |
| 5.4 | Πειραματική σειρά 2 : Αλληλεπίδραση με τη βιβλιοθήκη SLSensorManager | 43 |
| 5.5 | Πειραματική σειρά 3 : Παρουσίαση νέων κατευθύνσεων έρευνας | 45 |

5.1 Εισαγωγή

Σε αυτή την ενότητα, παρουσιάζουμε μια πειραματική μελέτη που δείχνει πώς το υποσύστημα προσομοιώσεων διευκολύνει τα πειράματα μεγάλων δεδομένων με αποδοτικό και αποτελεσματικό τρόπο. Περιγράφεται η πειραματική μεθοδολογία η οποία περιλαμβάνει ένα πολύπλοκο σύστημα εσωτερικής τοποθέτησης που αναπτύχθηκε στο τμήμα πληροφορικής του Πανεπιστημίου Κύπρου, το Anyplace, το οποίο μας επιτρέπει να δοκιμάσουμε πολλά από τα συναρπαστικά χαρακτηριστικά που πλέον διαθέτει το SmartLab εξαιτίας της ενσωμάτωσης του υποσυστήματος προσομοιώσεων.

Τα πειράματα επικεντρώνονται σε δυο ποιοτικές και ποσοτικές διαστάσεις :

- i. Παρακολούθηση και διαχείριση χαρακτηριστικών των εφαρμογών των χρηστών (δηλαδή , χρόνος εκτέλεσης, χρήση της ΚΜΕ και κατανάλωσης της μπαταρίας κατά τη σύγκριση αλγορίθμων εσωτερικής τοποθέτησης)

- ii. Πειραματισμός με τη χρήση δεδομένων αισθητήρων χρησιμοποιώντας τη βιβλιοθήκη SLSensorManager.

Τέλος παρουσιάζεται πως το υποσύστημα προσομοιώσεων διευκολύνει νέες κατευθύνσεις έρευνας μέσα από ένα πείραμα που χρησιμοποιεί δεδομένα που μαζεύτηκαν από το πλήθος για την επαλήθευση/ φιλτράρισμα πριν την αποθήκευση τους στο αποθετήριο μεγάλων δεδομένων.

5.2 Πειραματική Υποδομή

Anyplace [22]

Το Anyplace είναι μια πλατφόρμα εσωτερικής τοποθέτησης και πλοήγησης που λειτουργεί πάνω από τους χάρτες της Google με μια υπηρεσία Web 2.0 διαχείρισης μεγάλων δεδομένων στο παρασκήνιο. Το Anyplace εκμεταλλεύεται την ισχύ του λαμβανόμενου σήματος (RSS) εξάγοντας τις τιμές αυτές μέσω σάρωσης των πακέτων που μεταδίδονται από γειτονικά σημεία πρόσβασης. Για την αντιμετώπιση των δύσκολων συνθηκών διάδοσης του σήματος σε εσωτερικούς χώρους λόγω αντανακλάσεων και διαφορετικών κατευθύνσεων, οι ενδείξεις των RSS (δηλαδή διανύσματα που περιέχουν το σημείο πρόσβασης και την ισχύ του σήματος από αυτό) μαζεύονται εκ των προτέρων σε προκαθορισμένα σημεία αναφοράς. Κάθε αποτύπωμα συνδέεται με την αντίστοιχη θέση αναφοράς και αποθηκεύεται στο λεγόμενο Radiomap το οποίο καλύπτει ολόκληρη τη περιοχή ενδιαφέροντος. Ουσιαστικά, το Radiomap είναι μια χαρτογράφηση από το πολυδιάστατο χώρο ενδείξεων RSS σε φυσικές συντεταγμένες.

Η εφαρμογή για Android του Anyplace έχει δυο τρόπους λειτουργίας, είτε για καταγραφή ενδείξεων είτε για περιήγηση. Στη λειτουργία καταγραφής οι χρήστες μπορούν να καταγράψουν πληροφορίες για την ισχύ του σήματος από τα κοντινά σημεία πρόσβασης WIFI και μέσω πληθοπορισμού αποστέλλονται στο διακομιστή του Anyplace μέσω του API. Αυτά τα δεδομένα αποθηκεύονται στη βάση δεδομένων όπου και δημιουργείται το Radiomap, το οποίο αργότερα χρησιμοποιείται για τη παροχή πληροφοριών για τον εντοπισμό της θέσης του χρήστη. Στη λειτουργία περιήγησης οι χρήστες μπορούν να δουν την τρέχουσα θέση τους και να περιηγηθούν μεταξύ των σημείων ενδιαφέροντος στο εσωτερικό των κτηρίων. Οι ενσωματωμένοι αισθητήρες των έξυπνων κινητών συσκευών (δηλαδή , επιταχυνσιόμετρο, γυροσκόπιο και ψηφιακή

πυξίδα) χρησιμοποιούνται επίσης για να εξομαλύνουν τις θέσης των WIFI σημείων πρόσβασης έτσι ώστε να ενισχυθεί η πλοήγηση.

Έχει επιλεγεί το Anyplace ως βάση για τη πειραματική μελέτη της παρούσας διπλωματικής εφόσον, όπως λέχθηκε προηγουμένων έχει τη δυνατότητα να παρέχει πρόσβαση σε μεγάλα ποσά δεδομένων (π.χ., περίπου 10GB μετρήσεων RSS για τη χαρτογράφηση κάθε κτηρίου). Επιπλέον, η αποσφαλμάτωση και η δοκιμή μιας εφαρμογής σαν το Anyplace είναι χρονοβόρα και απαιτεί πολλούς πόρους όσον αφορά το προσωπικό και τον προϋπολογισμό. Ομοίως η αξιολόγηση διαφορετικών αλγορίθμων, προσεγγίσεων απαιτεί πολύ χρόνο και την ικανότητα επανειλημμένης εκτέλεσης της ίδιας διαδικασίας ενώ ταυτόχρονα να απομονώνεται και να επιτρέπεται λεπτομερής έλεγχος των άλλων μεταβλητών (π.χ., της κάρτας WIFI της κινητής συσκευής, ο αριθμός των σημείων πρόσβασης γύρω από τη θέση του χρήστη).

Αλγόριθμοι εσωτερικής τοποθέτησης

Για την διεκπεραίωση της πειραματικής μελέτης χρησιμοποιήθηκαν ορισμένοι αλγόριθμοι εσωτερικής τοποθέτησης, συμπεριλαμβανομένου του ντετερμινιστικού K Nearest Neighbor (KNN) [23,24], καθώς και του πιθανοτικού αλγορίθμου (MAP) και του ελαχίστου μέσου σφάλματος (MMSE) [25,26]. Αυτοί οι αλγόριθμοι έχουν συσκευαστεί εκ νέου σε Android βιβλιοθήκες με τη συναφή τεκμηρίωση και τοποθετήθηκαν στο αποθετήριο αλγορίθμων του SmartLab. Ομοίως, ο αλγόριθμος Dead Reckoning (DR) που χρησιμοποιείται στη δεύτερη πειραματική σειρά έχει επίσης τοποθετηθεί στο αποθετήριο αλγορίθμων κατά παρόμοιο τρόπο.

Σύνολα Δεδομένων

Στη παρούσα πειραματική μελέτη χρησιμοποιούνται τα ακόλουθα σύνολα δεδομένων.

UCY RadioMap: Αυτό το RadioMap έχει σχεδιαστεί από δεδομένα που συλλέχθηκαν σε ένα τυπικό κτήριο του τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου. Χρησιμοποιήθηκαν πολλές και ποικίλες συσκευές Android (HTC Hero, HTC Desire, Samsung Nexus S, Motorola Xoom, HTC One X, κλπ) για τη συλλογή 30 αποτυπωμάτων RSS (δηλαδή τιμές RSS των σημείων πρόσβασης σε μια θέση αναφοράς) σε 1,500 διακριτές περιοχές για ένα σύνολο από 45.000 αποτυπώματα

αναφοράς. Υπάρχουν 120 σημεία αναφοράς WLAN εγκατεστημένα σε 4 ορόφους του κτηρίου, συμπεριλαμβανομένων των σημείων αναφοράς των γειτονικών κτηρίων που μπορεί εν μέρει να ανιχνευτούν σε κάποια τμήματα του κτηρίου του τμήματος Πληροφορικής. Κατά μέσο όρο 10.6 σημεία αναφοράς ανιχνεύονται ανά θέση. Συλλέχθηκαν δεδομένα από το περπάτημα σε μια διαδρομή που αποτελείται από 2900 περιοχές.

UCY Sensory Readings: Τα δεδομένα των αισθητήρων που αποθηκεύτηκαν στο αποθετήριο μεγάλων δεδομένων συλλεχτήκαν από διάφορες τροχιές στο κτήριο Πληροφορικής. Όπως και στο UCY RadioMap χρησιμοποιήθηκαν ποικίλες συσκευές Android και χρησιμοποιήθηκαν τροχιές που μαζεύτηκαν όλα αυτά τα χρόνια από τους φοιτητές με τα πόδια (γύρω στα 20 km).

Κινητές συσκευές

Ο πίνακας 2 παρουσιάζει μια ολοκληρωμένη λίστα και κατηγοριοποίηση, ανάλογα με τον αριθμό των πυρήνων, όλες τις συσκευές που χρησιμοποιήθηκαν κατά τη φάση του πειραματισμού.

5.3 Πειραματική Σειρά 1: Ανάπτυξη Πειραμάτων μέσω του υποσυστήματος προσομοιώσεων

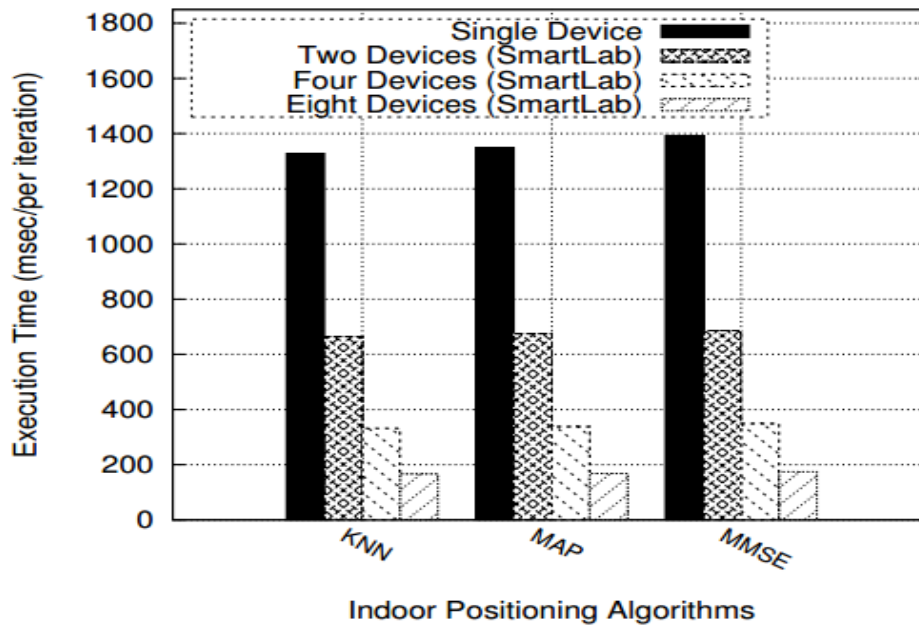
Προκειμένου να εξετάσουμε πώς το υποσύστημα προσομοίωσης μπορεί να διευκολύνει τον εκ νέου προγραμματισμό και επαναληψιμότητα των πειραμάτων σε ένα μεγάλο αριθμό κινητών (ετερογενή ή όχι) έχουμε διεξάγει ένα micro benchmark το οποίο υπολογίζει τον απαιτούμενο χρόνο για την εκτέλεση ποικίλων αλγορίθμων εσωτερικής γεωτοποθέτισης σε πολλαπλές συσκευές μέσω του SmartLab. Επιπλέον θέλαμε να εξετάσουμε πώς η διαφορετικότητα των κινητών συσκευών θα επηρεάσουν τους αλγόριθμους εσωτερικής τοποθέτισης. Ομοίως με τις προηγούμενες πειραματικές ρυθμίσεις, μετράμε το χρόνο, σε δευτερόλεπτα, που απαιτείται για την εκτέλεση ενός αλγορίθμου εσωτερικής τοποθέτισης (δηλαδή, την ανάκτηση του geo-location του

| Category | Target Device Model | CPU | Android OS version |
|--------------------|---------------------|-----------------------------|----------------------|
| Single-core (1GHz) | HTC Desire | Qualcomm QSD8250 Snapdragon | Jelly Bean (v 4.2.2) |
| Dual-core (1GHz) | Motorola Xoom | Nvidia Tegra 2 T20 | Ice Cream (v.4.0.4) |
| Quad-core (1.5GHz) | HTC One X | Nvidia Tegra 3 | Jelly Bean (v4.2.2) |
| | Google Nexus 7 | Qualcomm Snapdragon S4Pro | Kit Kat (v4.4.2) |

Σχήμα 6.2.1 : Κινητές συσκευές που χρησιμοποιήθηκαν για τη διεκπεραίωση των πειραμάτων

χρήστη) σε διαφορετικό αριθμό κινητών συσκευών. Κάθε πείραμα εκτελέστηκε 5 φορές οι οποίες αποτελούνται από δέσμες των 450 επαναλήψεων σε μια μονοπύρηνη συσκευή (π.χ. HTC DESIRE) και καταγράφεται η μέση τιμή για κάθε δέσμη. Τα αποτελέσματα απεικονίζονται στο σχήμα 5.3.1 το οποίο απεικονίζει καθαρά το πλεονέκτημα αξιοποίησης του SmartLab στο πλαίσιο αυτού του τύπου πειραμάτων , δεδομένου ότι επιτρέπει το μαζικό παραλληλισμό του αλγορίθμου εσωτερικής τοποθέτησης σε πολλαπλές κινητές συσκευές, με αποτέλεσμα σημαντική μείωση του χρόνου που χρειάζεται για πειραματισμό. Επιπλέον, μια άλλη παρατήρηση όσον αφορά τη βιβλιοθήκη εξομοίωσης, ήταν η δυνατότητα της να τροφοδοτήσει το ακριβές Wi-Fi fingerprint από το βάση δεδομένων στις κινητές συσκευές, προκειμένου να διασφαλιστεί η πανομοιότυπη εκτέλεση των πειραμάτων και να μειωθούν τυχόν παρενέργειες (side effects).Όσον αφορά το συγκεκριμένο πείραμα χρησιμοποιήθηκαν 8 συσκευές ίδιου τύπου οπότεν τα πειράματα μπορούσαν να εκτελεστούν 8 φορές πιο γρήγορα. Συμπεραίνουμε λοιπόν πως μέσω του SmartLab αλλά και της βιβλιοθήκης SLSensorManager επιτυγχάνεται η εκτέλεση πειραμάτων (*i όπου i οι συσκευές που διαθέτουμε).

Χρόνος εκτέλεσης για διαφορετικούς αλγόριθμους εσωτερικής τοποθέτησης
Μονοπύρηνη Συσκευή (HTC DESIRE)

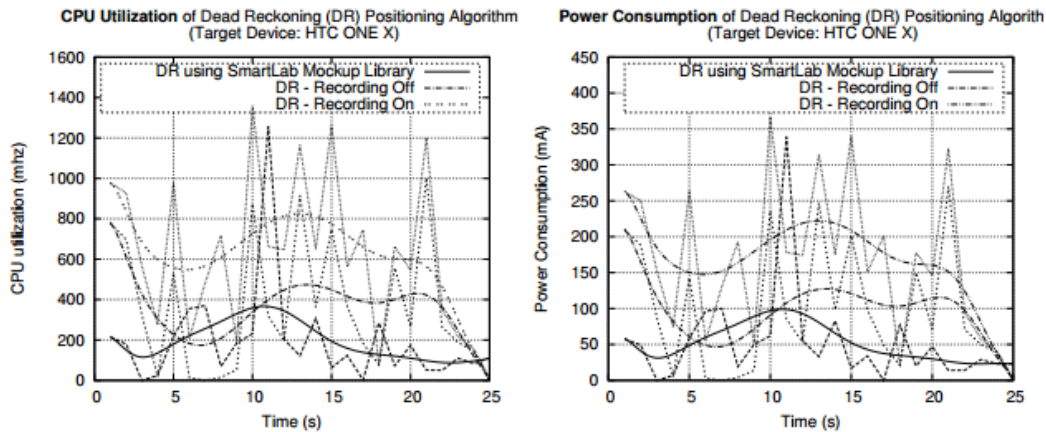


Σχήμα 5.3.1

Micro benchmark επαναληψιμότητας και επαναπρογραμματισμού. Αξιολογώντας το μέσο χρόνο για την ανάκτηση 450 φορές της θέσης του χρήστη από διαφορετικού τύπου συσκευές χρησιμοποιώντας διαφορετικούς αλγόριθμους εσωτερικής τοποθέτησης.

5.4 Πειραματική σειρά 2 : Αλληλεπίδραση με τη βιβλιοθήκη SLSensorManager

Σε αυτή τη πειραματική σειρά δείχνουμε τη χρησιμοποίηση της βιβλιοθήκης SLSensorManager με απρόσκοπτη ενσωμάτωση των μετρήσεων RSS στο Tracker application του Anyplace. Χρησιμοποιείται η ίδια μεθοδολογία που έχει περιγραφή αλλά παρουσιάζονται τα αποτελέσματα από τη κινητή συσκευή HTC ONE X αντί του HTC DESIRE.



Σχήμα 5.4.1: Φαίνονται διάφορες άλλες μετρήσεις για τα διάφορα κινητά καθώς επίσης και την επιβάρυνση της βιβλιοθήκης

Ο κύριος στόχος είναι να δείξουμε πως η βιβλιοθήκη SLSensorManager τροφοδοτεί τον Anyplace Tracker με ενδείξεις αισθητήρων που έχουν συλλεχτεί προηγουμένως προκειμένου να αξιολογήσει την απόδοση του αλγορίθμου Dead Reckoning (DR) ο οποίος χρησιμοποιείται για εσωτερική τοποθέτηση μέσω αισθητήρων. Στο συγκεκριμένο πείραμα εξετάζουμε τη χρήση του CPU καθώς επίσης και την κατανάλωση ενέργειας του συγκεκριμένου αλγορίθμου σε τετραπύρηνη συσκευή (π.χ. HTC ONE X).

Κάθε πείραμα εκτελέστηκε 10 φορές χρησιμοποιώντας τρεις διαφορετικούς τρόπους:

- i. εκτέλεση αλγόριθμου DR με τροφοδότηση αισθητήρων από τη βιβλιοθήκη SLSensorManager,
- ii. εκτέλεση του αλγορίθμου όπως διατίθεται στον Anyplace Tracker (DR-Recording Off)
- iii. εκτέλεση αλγορίθμου κατά την εγγραφή και τη συμβολή των αισθητικών αναγνώσεις στο μεγάλο δεδομένων αποθετήριο (DR-Recording On).

Τα αποτελέσματα παρουσιάζονται στο σχήμα 5.4.1 και απεικονίζουν τη χρήση του CPU και την κατανάλωση ενέργειας για κάθε mode. Τα αποτελέσματα δείχνουν ότι η χρήση των αισθητήρων έναντι της ανάκτησης των δεδομένων αυτών από τη βάση δεδομένων χρησιμοποιεί λιγότερη ισχύ και καταναλώνει λιγότερη ενέργεια (γύρω στο 2%). Τέλος, όπως και στο προηγούμενο πείραμα, όλα τα αποτελέσματα(δηλαδή ίχνη

| Algorithms | Average Positioning Error | Standard Deviation |
|----------------|---------------------------|--------------------|
| KNN Algorithm | 10.12 meters | 5.50 meters |
| MAP Algorithm | 25.73 meters | 25.57 meters |
| MMSE Algorithm | 9.94 meters | 9.71 meters |

Σχήμα 5.5.1 Υπολογιζόμενη Μέση θέση λάθους για Διαθέσιμους Αλγόριθμους Indoor Positioning

χρήσης του CPU και κατανάλωσης ενέργειας) προσφέρθηκαν στο “μεγάλο” χώρο αποθήκευσης δεδομένων.

5.5 Πειραματική σειρά 3 : Παρουσίαση νέων κατευθύνσεων έρευνας

Σε αυτή την πειραματική σειρά , παρουσιάζουμε μια νέα κατεύθυνση έρευνας που έχει ως στόχο την επαλήθευση των δεδομένων εσωτερικής τοποθέτησης προτού αποθηκευτούν στη βάση δεδομένων. Για να το πετύχουμε αυτό , έχουμε διεξάγει ένα πείραμα που αποτελείται από δυο φάσεις. ι) Ο υπολογισμός μέσου όρου σφάλματος θέσης και της τυπικής απόκλισης για κάθε διαθέσιμο αλγόριθμο, ιι) χρήση των προηγούμενων υπολογισμών προκειμένου να φιλτραριστούν πιθανές κακόβουλες εισφορές σκόπιμα ή κατά λάθος κακόβουλες ενδείξεις για τη θέση του συγκεκριμένου χρήστη).

Πρώτη φάση – Υπολογισμός μέσης τιμής σφάλματος θέσης

Προκειμένου να υπολογιστεί η μέση τιμή σφάλματος θέσης για κάθε αλγόριθμο εντοπισμού, αξιολογήσαμε τα σφάλματα που προέκυψαν από τον υπολογισμό της Ευκλείδειας απόστασης σχετικά με τη θέση που επέστρεφε ο αλγόριθμος με τη πραγματική θέση του χρήστη. Δίνονταν σε κάθε αλγόριθμο κάποιες διευθύνσεις MAC με τα RSS σήματα τους σαν είσοδος, τα οποία ανήκαν σε ένα πραγματικό συγκεκριμένο σημείο (x,y). Οπότεν μέσω της Ευκλείδειας απόστασης βλέπαμε πόση απόκλιση είχε κάθε φορά ο κάθε αλγόριθμος από το πραγματικό σημείο. Τα αποτελέσματα της φάσης αυτής φαίνονται στο πίνακα 5.5.1 και δείχνουν τη μέση τιμή σφάλματος θέσης καθώς επίσης και τη τυπική απόκλιση.

| Dataset | # of Fingerprints Contributed | Correctness |
|-------------------|-------------------------------|-------------|
| Verified dataset | 41 out of 45 | 91 % |
| Malicious dataset | 6 out of 45 | 86 % |

Σχήμα 5.5.2 : Τα αποτελέσματα που φαίνονται στο σχήμα 6.4.2, δείχνουν πως το SmartLab μπορεί να παίξει σημαντικό ρόλο στην επικύρωση crowd-sourced δεδομένων χρησιμοποιώντας τη νέα αυτή μέθοδο που έχουμε προτείνει.

Δεύτερη Φάση – Φιλτράρισμα κακόβουλων συνεισφορών

Προκειμένου να εξεταστεί η αποδοτικότητα της προτεινόμενης τεχνικής αποφασίσαμε να τροφοδοτήσουμε τα κινητά του SmartLab με δεδομένα χρηστών είτε κακόβουλων είτε όχι και το SmartLab ήταν υπεύθυνο σύμφωνα με τη μέση τιμή σφάλματος θέσης που καθορίσαμε να μπορεί να αποκλείσει τα δεδομένα των κακόβουλων χρηστών. Συλλέξαμε 90 fingerprints και τα χωρίσαμε σε 2 κατηγορίες. Τα μισά από αυτά θεωρούνταν ως “verified” και τα υπόλοιπα ως “malicious”. Με τον όρο verified εννοούνται τα αποτυπώματα τα οποία ήτα ορθά και σύμφωνα με τη πρώτη φάση η απόκλιση τους ήταν ανεκτή. Malicious ήταν αποτυπώματα που εσκεμμένα μαζέψαμε από διαφορετικά σημεία έχοντας όμως εισάγει λάθος τοποθεσία σε σχέση με τα σημεία πρόσβασης από τα οποία λαμβάναμε σήμα. Σαν τελικό βήμα τροφοδοτήσαμε το SmartLab με αυτά τα δεδομένα και παρατηρήσαμε τη διαδικασία επικύρωσης.

Αποτελέσματα

Τα αποτελέσματα που απεικονίζονται στον πίνακα 5.52 δείχνουν πως το υποσύστημα που αναπτύχθηκε στα πλαίσια της διπλωματικής αυτής μπορεί να διαδραματίσει καθοριστικό ρόλο για την επαλήθευση δεδομένα προερχόμενα από το πλήθος. Τα αποτελέσματα δείχνουν πως μόνο 41 από τα 45 (περίπου 91%) αποτυπώματα RSS από το σύνολο των επικυρωμένων δεδομένων πολύ σωστά πέρασαν από τον έλεγχο και αποθηκεύτηκαν στο αποθετήριο μεγάλων δεδομένων ενώ μόνο 4 από τα 45 (9%) λανθασμένα απορρίφθηκαν. Παρόμοια παρατήρηση προέκυψε από τα αποτελέσματα των κακόβουλων δεδομένων αφού μόνο 6 από τα 45 (14%) των αποτυπωμάτων λανθασμένα αποθηκεύτηκαν στο χώρο αποθήκευσης ενώ 39 από τα 45 (86%) πολύ σωστά απορρίφθηκαν.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντικές Επεκτάσεις

| | | |
|-----|------------------------|----|
| 6.1 | Συμπεράσματα | 47 |
| 6.2 | Μελλοντικές Επεκτάσεις | 48 |

6.1 Συμπεράσματα

Στην εργασία αυτή παρουσιάστηκε ένα σύστημα προσομοιώσεων δεδομένων αισθητήρων σε πραγματικές κινητές συσκευές και αποδείχτηκε πως μπορεί να διαχειριστεί με ευκολία την πολυπλοκότητα των πειραμάτων διαχείρισης δεδομένων μεγάλης κλίμακας σε πραγματικές κινητές συσκευές. Έχουν παρουσιαστεί και επικυρωθεί τα ποιοτικά χαρακτηριστικά της αρχιτεκτονικής του συστήματος μέσω πειραμάτων σε ένα πολύπλοκο σύστημα εσωτερικής γεωτοποθέτησης για έξυπνες κινητές συσκευές και έχει διαφανεί πως οι ερευνητές μπορούν να εκτελέσουν τα πειράματά τους με ένα πιο αποτελεσματικό και αποδοτικό τρόπο μέσω του συστήματος αυτού. Οι αρχικοί στόχοι έχουν επιτευχθεί αφού το σύστημα επιτρέπει προσομοίωση δεδομένων αισθητήρων σε πραγματικές και εικονικές κινητές συσκευές ετερογενείς και μη μέσω της βιβλιοθήκης SLSensorManager η οποία δεν επηρεάζεται από το υλικό της

συσκευής αφού τα δεδομένα λαμβάνονται από το δίκτυο και όχι από τους πραγματικούς αισθητήρες της κινητής συσκευής. Παρέχεται ακρίβεια στην αναπαράσταση ενός σεναρίου καταγραφής αφού χρησιμοποιείται αλγόριθμος κοντινότερων γειτόνων για την ανάκτηση της κοντινότερης τιμής σε αυτή που ζητείται. Επίσης το σύστημα μπορεί να χειριστεί μεγάλα δεδομένα αφού χρησιμοποιείται η NoSQL βάση δεδομένων που προσφέρει οριζόντια κλιμάκωση και υψηλή διαθεσιμότητα και περιλαμβάνει ένα ενσωματωμένο επίπεδο memcache για την εξυπηρέτηση των συχνών αιτημάτων έτσι ο χρόνος απόκρισης για τα μεγάλα δεδομένα μειώνεται.

6.2 Μελλοντικές Επεκτάσεις

Το σύστημα έχει τη δυνατότητα να προσομοιώνει σενάρια από δεδομένα αισθητήρων σε πραγματικές κινητές συσκευές Android. Στο μέλλον θα μπορούσε να γίνει μια επέκταση της βιβλιοθήκης η οποία εγκαθιστάτε στις κινητές συσκευές ώστε να μπορεί το σύστημα να εξομοιώσει δεδομένα σε κινητές συσκευές IOS7 και Windows. Επίσης θα μπορούσε να προστεθεί ένα ανασυρόμενο παράθυρο παρόμοιο με αυτό που υπάρχει στον εξυπηρετητή του συστήματος το οποίο να χρησιμοποιείται για caching αλλά και prefetching ώστε να γίνει πιο ομαλή και περισσότερο ακριβής η εξομοίωση. Το σύστημα έχει υλοποιηθεί με τρόπο που να επιτρέπει πολύ εύκολα, με λίγες αλλαγές στην υλοποίηση, τη προσομοίωση του σεναρίου σε πραγματικό χρόνο την ώρα της καταγραφής πράγμα που θα προσφέρει ένα πιο ελκυστικό τρόπο αναπαράστασης των δεδομένων προς το χρήστη κατά τη φάση της καταγραφής. Επίσης θα μπορούσε να βελτιωθεί ο τρόπος με τον οποίο υπολογίζεται το μέγεθος του παραθύρου στον εξυπηρετητή αν ληφθούν υπόψη περισσότερες μεταβλητές και όχι μόνο η συμφόρηση του δικτύου. Τέλος, μια πολύ σημαντική επέκταση είναι να προστεθεί κάποιο είδος caching στον εξυπηρετητή κατά τη διάρκεια της εξομοίωσης έτσι ώστε δεδομένα τα οποία ήδη υπάρχουν στον εξυπηρετητή να μην ξαναζητούνται από το αποθετήριο μεγάλων δεδομένων.

Βιβλιογραφία

- [1]. Georgios Larkou, Constantinos Costa, Panayiotis G. Andreou, Andreas Konstantinidis and Demetrios Zeinalipour-Yazti. “Managing Smartphone Testbeds with Smartlab”, In Proceedings of the 27th International Conference on Large Installation System Administration, USENIX Association, LISA'13, 115-132, 2013.
- [2]. Georgios Chatzimilioudis, Andreas Konstantinidis, Christos Laoudias, and Demetrios Zeinalipour-Yazti, “Crowdsourcing with Smartphones”, IEEE Internet Computing, vol. 16, no. 5, pp. 36–44, Sep. 2012.
- [3]. Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh and Reynold Xin, “CrowdDB: answering queries with crowdsourcing.” In SIGMOD, 2011.
- [4]. Adam Marcus, Eugene Wu, Samuel Madden and Robert C. Miller, “Crowdsourced Databases: Query Processing with People,” In CIDR, 2011.
- [5]. David R. Chones, Fabian Bustamante and Zihui Ge, “Crowdsourcing service-level network event monitoring,” In SIGCOMM, 2011.
- [6]. Archak Money, “Glory and Cheap Talk: Analyzing Strategic Behavior of Contestants in Simultaneous Crowdsourcing Contests on TopCoder.com” In WWW, 2011.
- [7]. Omar F. Zaidan and Chris Callison-Burch, “Crowdsourcing Translation: Professional Quality from Non-Professionals.” In ACL, 2011.
- [8].41. Anthony Brew, Derek Greene and Pdraig Cunningham, “Using Crowdsourcing and Active Learning to Track Sentiment in Online Media,” In em ECAI, 2010
- [9]. Christos Laoudias, George Constantinou, Marios Constantinides, Silouanos Nicolaou, Demetrios Zeinalipour-Yazti, and Christos G. Panayiotou. “The Airplace Indoor Positioning Platform for Android Smartphones”, In Proceedings of the 13th IEEE International Conference on Mobile Data Management (MDM'12), IEEE Computer Society, Washington, DC, USA, 312-315, 2012.



- [10]. Larry Peterson, Tom Anderson, David Culler, and Timothy Roscoe. "A Blueprint for Introducing Disruptive Technology into the Internet", A blueprint for introducing disruptive technology into the Internet. SIGCOMM Comput. Commun. Rev. 33, 1 (January 2003), 59-64, 2003
- [11]. David Johnson, Tim Stack, Russ Fish, Daniel Montrallos Flickinger, Leigh Stoller, Robert Ricci, and Jay Lepreau. "Mobile Emulab: A Robotic Wireless and Sensor Network Testbed", In Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM'06), IEEE Computer Society, Washington, DC, USA, 1-12, 2006
- [12]. Georey Werner-Allen, Patrick Swieskowski, and Matt Welsh. "MoteLab: a wireless sensor network testbed", In Proceedings of the 4th international symposium on Information processing in sensor networks (IPSN'05). IEEE Press, Piscataway, NJ, USA, Article 68, 2005.
- [13]. Samsung, Remote Test Lab: <http://goo.gl/p7SNU>
- [14]. Perfecto Mobile, <http://goo.gl/DS1P9>.
- [15]. Keynote Systems Inc., Device Anywhere: <http://goo.gl/mCxFt>
- [16]. AT&T Application Resource Optimizer (ARO), Free Diagnostic Tool:<http://goo.gl/FZnXS>
- [17]. OpenIntents Sensor Simulator <http://www.openintents.org/node/6>
- [18]. Samsung Sensor Simulator <http://developer.samsung.com/android/tools-sdks/Samsung-Sensor-Simulator>
- [19]. Qemu Project http://wiki.qemu.org/Main_Page
- [20]. AndroidEmulator
<http://developer.android.com/tools/devices/emulator.html>
- [21]. Android Sensor Manager
<http://developer.android.com/reference/android/hardware/SensorManager.html>
- [22]. Anyplace, <http://anyplace.cs.ucy.ac.cy/>
- [23]. Paramvir Bahl and Venkata Padmanabhan, "RADAR: an in-building RF-based user location and tracking system", in 19th IEEE Conference on Computer Communications,(Infocom'00), 2000, 775-784.
- [24]. Binghao Li, James Salter, Andrew G. Dempster, and Chris Rizos, "Indoor positioning techniques based on wireless LAN", In Proceedings of the 1st IEEE

International Conference on Wireless Broadband and Ultra Wideband Communications, 2006.

- [25]. Moustafa Youssef and Ashok Agrawala, "The Horus WLAN location determination system," In Proceedings of the 3rd international conference on Mobile systems, applications, and services (MobiSys'05). ACM, Seattle, WA, USA, 205-218, 2005.
- [26]. Teemu Roos, Petri Myllymaki, Henry Tirri, Pauli Misikangas, and Juha Sievanen, "A probabilistic approach to WLAN user location estimation", International Journal of Wireless Information Networks, vol. 9, no. 3, pp. 155-164, Jul. 2002.

Παράρτημα Α

Ο σχετικός κώδικας που αφορά την υλοποίηση της παρούσας διπλωματικής βρίσκεται μέσα στο cd στους εξής φακέλους.

| | | |
|--|------------------|-------------|
|  android | 2/6/2014 1:27 μμ | File folder |
|  SmartLabDeviceServer | 2/6/2014 1:27 μμ | File folder |

Στο φάκελο android βρίσκεται ο κώδικας και το εκτελέσιμο για την εφαρμογή που χρησιμοποιείται για καταγραφή. Επίσης βρίσκεται και ο κώδικας υλοποίησης της βιβλιοθήκης η οποία χρησιμοποιείται για εξομοίωση.

Στο φάκελο SmartLabDeviceServer βρίσκεται ο κώδικας που αφορά τον εξυπηρετητή ο οποίος διαχειρίζεται τη σύνδεση με τις κινητές συσκευές και όλα τα υποσυστήματα συμπεριλαμβανομένου και του υποσυστήματος προσομοιώσεων. Οι κλάσεις TCP Server, EmulationClnet, EmulationServer, Couchbase, CouchbaseWindow είναι οι κλάσεις που υλοποιήθηκαν στη παρούσα διπλωματική εργασία.