

Ατομική Διπλωματική Εργασία

**ANYPLACE: ΕΝΑ ΟΛΟΚΛΗΡΩΜΕΝΟ ΣΥΣΤΗΜΑ ΕΣΩΤΕΡΙΚΗΣ
ΓΕΩ-ΤΟΠΟΘΕΤΗΣΗΣ ΚΑΙ ΠΛΟΗΓΗΣΗΣ ΑΠΟ
ΜΕΓΑΛΑ ΔΕΔΟΜΕΝΑ**

Λάμπρος Πέτρου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

***ANYPLACE*: ΕΝΑ ΟΛΟΚΛΗΡΩΜΕΝΟ ΣΥΣΤΗΜΑ ΕΣΩΤΕΡΙΚΗΣ ΓΕΩ-ΤΟΠΟΘΕΤΗΣΗΣ ΚΑΙ ΠΛΟΗΓΗΣΗΣ ΑΠΟ ΜΕΓΑΛΑ ΔΕΔΟΜΕΝΑ**

Λάμπρος Πέτρου

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

Αρχικά θα ήθελα να ευχαριστήσω θερμά τον κ. Ζεϊναλιπούρ Δημήτρη επιβλέπων καθηγητή της παρούσας διπλωματικής εργασίας για την υποστήριξη, καθοδήγηση και συμβουλές που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας.

Ακολούθως θα ήθελα να ευχαριστήσω τους φίλους μου εντός και εκτός πανεπιστημίου που μου συμπαραστάθηκαν όλο αυτό τον καιρό αλλά και ένα τεράστιο ευχαριστώ στην οικογένεια μου που συνεχώς με ενθάρρυνε να συνεχίσω και να μην πτοούμαι από δυσκολίες και εμπόδια που έβρισκα μπροστά μου. Ειδικότερα, θα ήθελα να ευχαριστήσω την μητέρα μου για την ηθική και οικονομική υποστήριξη που μου πρόσφερε όχι μόνο κατά τη διάρκεια της εκπόνησης αυτής της διπλωματικής εργασίας αλλά και όλη τη διάρκεια των σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω τον γνωστό οίκο μόδας Victorias Secret αφού μου πρόσφερε αμέτρητες ώρες οπτικοακουστικού υλικού ώστε να χαλαρώνω τις πολύωρες ώρες που αφιέρωνα στη μελέτη μου καθώς με βοηθούσαν να σκεφτώ out-of-the box σε κρίσιμες στιγμές.

Αναγνώριση

Επιδείξεις του συστήματος AnyPlace που περιγράφεται σε αυτήν τη διπλωματική εργασία έχουν παρουσιαστεί με επιτυχία στα ακόλουθα διεθνή συνέδρια:

1. *"Crowdsourced Indoor Localization and Navigation with Anyplace"*, **L. Petrou**, G. Larkou, C. Laoudias, D. Zeinalipour-Yazti and C.G. Panayiotou
"Proceedings of the 13th International Conference on Information Processing in Sensor Networks" (IPSN'14), Berlin, Germany, April 15-17, 2014
2. *"Anyplace: Indoor Positioning and Navigation in the Big-Data Era"*, **L. Petrou**, G. Larkou, C. Laoudias, D. Zeinalipour-Yazti and C. G. Panayiotou
"Proceedings of the 4th International Conference on Indoor Positioning and Indoor Navigation" (IPIN'13), Montbeliard-Belfort, France, October 28-31, 2013

Περίληψη

Σε αυτή τη διπλωματική εργασία παρουσιάζεται ένα ολοκληρωμένο σύστημα γεωπλοήγησης και γεωτοποθέτησης σε εσωτερικούς χώρους με Μεγάλα Δεδομένα, το οποίο φέρει το όνομα AnyPlace.

Τα τελευταία χρόνια παρουσιάστηκε τεράστια εξάπλωση συστημάτων τοποθέτησης τα οποία βασίζονται στις ενδείξεις γειτονικών σημείων πρόσβασης ασύρματης δικτύωσης (Wi-Fi Access Points), καθώς το GPS (Global Positioning System) δεν είναι διαθέσιμο εσωτερικά των κτιρίων και η ενέργεια που απαιτεί το καθιστά ακατάλληλο για παρατεταμένη χρήση. Επιπρόσθετα, συνεπακόλουθο της εξάπλωσης των έξυπνων συσκευών και της τρομερής αύξησης των πληροφοριών που παράγονται ήταν να εμφανιστεί ο όρος *Μεγάλα Δεδομένα*. Δεδομένα τα οποία φτάνουν μεγέθη που πλέον είναι δύσκολο να διαχειριστούν τα υφιστάμενα συστήματα διαχείρισης δεδομένων.

Σε αυτή την εργασία, θα παρουσιαστεί το AnyPlace, ένα ολοκληρωμένο σύστημα γεωτοποθέτησης (ανίχνευση του χρήστη σε πραγματικό χρόνο) και γεωπλοήγησης (παροχή οδηγιών στον χρήστη για την μετάβαση του σε διάφορα σημεία ενδιαφέροντος) πάνω από μεγάλα δεδομένα. Έχει αναπτυχθεί μια ολόκληρη αρχιτεκτονική που υποστηρίζει το AnyPlace με στόχο την αποδοτική διαχείριση μεγάλων δεδομένων, πληροφοριών που συλλέγονται από τις συσκευές των χρηστών. Έχει αναπτυχθεί ένα σύστημα διαχείρισης των *RSS (Received Signal Strength)* μετρήσεων το οποίο μπορεί να επεξεργαστεί αποδοτικά τεράστιες ποσότητες δεδομένων και εκτελείται πάνω από ένα κατανεμημένο σύστημα βασισμένο στο πλαίσιο Hadoop.

Επιπρόσθετα, έχουν αναπτυχθεί διάφορες εφαρμογές για έξυπνα κινητά Android αλλά και εφαρμογές ιστού μέσω των οποίων μπορεί κανείς να αποκτήσει πρόσβαση και να περιηγηθεί στα χαρτογραφημένα κτίρια του συστήματος και να ζητήσει δρομολόγηση προς και από συγκεκριμένα σημεία ενδιαφέροντος. Με την χρήση μιας κινητής συσκευής επιτρέπεται επίσης ανίχνευση του χρήστη σε πραγματικό χρόνο. Η χαρτογράφηση των κτιρίων μπορεί να γίνει μέσω μιας ειδικής εφαρμογής ιστού που αναπτύχθηκε στα πλαίσια αυτής της εργασίας από οποιονδήποτε περιηγητή.

Περιεχόμενα

Ευχαριστίες.....	iii
Αναγνώριση	iv
Περίληψη	v
Περιεχόμενα.....	vi
Κεφάλαιο 1 Εισαγωγή	1
1.1 Υποκίνηση εργασίας.....	1
1.2 Ανασκόπηση συστήματος AnyPlace	4
1.3 Συνεισφορές.....	5
1.4 Περίγραμμα εργασίας	6
Κεφάλαιο 2 Σχετική Βιβλιογραφία	7
2.1 Εσωτερική γεωτοποθέτηση βασισμένη σε WLAN RSS	7
2.2 Σχετικά συστήματα γεωτοποθέτησης για εσωτερικούς χώρους.....	11
2.2.1 RADAR	12
2.2.2 MazeMap	13
2.2.3 Airplace.....	15
2.3 Τεχνικές διαχείρισης αποτυπωμάτων σημείων πρόσβασης.....	19
2.4 Υπόβαθρο Παράλληλης – Κατανεμημένης Επεξεργασίας.....	24
2.4.1 Παράλληλα και Κατανεμημένα Συστήματα	24
2.4.2 Μοντέλα προγραμματισμού για κατανεμημένα συστήματα	27
Κεφάλαιο 3 Αρχιτεκτονική του AnyPlace	32
3.1 Υποδομή Ραχοκοκαλιάς	34
3.2 Εφαρμογές στον Παγκόσμιο Ιστό.....	34
3.3 Εφαρμογές για έξυπνα κινητά Android	35
3.3.1 AnyPlace Logger	35
3.3.2 AnyPlace Navigator	35
3.4 Ροή πληροφορίας στο AnyPlace & Περιπτώσεις Χρήσης	36
3.4.1 Χαρτογράφηση κτιρίου μέσω του AnyPlace Architect.....	36
3.4.2 Συλλογή RSS αποτυπωμάτων μέσω του AnyPlace Logger	37
3.4.3 Χρήση εφαρμογής AnyPlace Navigator & AnyPlace Viewer.....	37

Κεφάλαιο 4 Δημιουργία Ραδιοχάρτη σε Μεγάλα Δεδομένα – LPRadioScale	39
4.1 Θεωρητικό υπόβαθρο	39
4.2 Περιγραφή προβλήματος	44
4.3 Μοντέλο Συστήματος	46
4.4 Περιγραφή προτεινόμενου αλγόριθμου – <i>LPRadioScale</i>	48
4.5 Λεπτομερής Ανάλυση αλγόριθμου <i>LPRadioScale</i>	49
4.5.1 Συλλογή στατιστικών στοιχείων για κάθε σημείο πρόσβασης.....	49
4.5.2 Φιλτράρισμα των σημείων πρόσβασης.....	52
4.5.3 Δημιουργία Ραδιοχάρτη	52
Κεφάλαιο 5 AnyPlace Mobile – Εφαρμογές για Android συσκευές.....	55
5.1 AnyPlace Navigator	55
5.1.1 Ανίχνευση και τοποθέτηση σε πραγματικό χρόνο (Positioning).....	57
5.1.2 Πλοήγηση & Δρομολόγηση (Navigation)	59
5.1.3 Αναζήτηση σημείων ενδιαφέροντος.....	60
5.2 AnyPlace Logger	61
Κεφάλαιο 6 AnyPlace Web – Υποδομή Ραχοκοκαλιάς και Συστατικά	63
6.1 Επίπεδο Εξυπηρετητών – Web-server Layer	63
6.2 Επίπεδο Δεδομένων – Data Layer	66
6.3 AnyPlace Viewer	69
6.4 AnyPlace Architect	73
6.5 AnyPlace Developers.....	74
Κεφάλαιο 7 Πειραματική Αποτίμηση Συστήματος.....	76
7.1 Μεθοδολογία.....	76
7.2 Αριθμός εμφανίσεων ενός σημείου πρόσβασης στη συλλογή μετρήσεων.....	77
7.3 Αριθμός μοναδικών τιμών ένδειξης ισχύς για κάθε σημείο πρόσβασης	78
7.4 Τυπική απόκλιση των ενδείξεων ισχύς κάθε σημείου πρόσβασης.....	80
7.5 Σύγκριση όλων των φίλτρων	82
Κεφάλαιο 8 Συμπεράσματα και Μελλοντικές Επεκτάσεις	84
8.1 Συμπεράσματα	84
8.2 Μελλοντικές επεκτάσεις	85
Βιβλιογραφία	87

Κεφάλαιο 1

Εισαγωγή

1.1	Υποκίνηση εργασίας	1
1.2	Ανασκόπηση συστήματος AnyPlace	4
1.3	Συνεισφορές	5
1.4	Περίγραμμα εργασίας	6

1.1 Υποκίνηση εργασίας

Σύμφωνα με πρόσφατες στατιστικές έρευνες από την Strategy Analytics [1] φαίνεται ότι οι άνθρωποι περνούν το 80-90% του χρόνου τους σε εσωτερικούς χώρους όπως εμπορικά κέντρα, βιβλιοθήκες, αεροδρόμια ή πανεπιστημιακούς χώρους. Επίσης, η ίδια έρευνα έδειξε ότι το 70% των τηλεφωνημάτων αλλά και η ασύρματη διακίνηση δεδομένων από κινητές συσκευές έγινε από εσωτερικούς χώρους.

Παρόμοιες έρευνες έδωσαν το έναυσμα και αύξησαν το ενδιαφέρον για την δημιουργία και εξέλιξη υπηρεσιών για εσωτερικές τοποθεσίες (location-based services) όπως επίσης και την ανάπτυξη εφαρμογών που λαμβάνουν υπόψη την τοποθεσία του χρήστη (location-aware applications). Απλά παραδείγματα εφαρμογών είναι η καθοδηγούμενη πλοήγηση ή η αναζήτηση αντικειμένων εντός κτιρίων. Για να γίνει δυνατή η ανάπτυξη τέτοιου είδους εφαρμογών αλλά και να γίνουν γενικά αποδεκτές και να τύχουν χρήσης ευρέως από το κοινό έπρεπε να επινοηθούν νέες μέθοδοι για την παροχή ακριβέστατων και αξιόπιστων ενδείξεων της τοποθεσίας του χρήστη, καθώς το ευρέως διαδεδομένο

GPS παρέχει μειωμένη διαθεσιμότητα εντός κτιρίων λόγω της απόφραξης και της εξασθένησης των σημάτων, με την ύπαρξη διάφορων αντικειμένων (τοιχοί, πόρτες, έπιπλα, κλπ.).

Μετά από πολλές έρευνες τα τελευταία 15 χρόνια πολλά συστήματα προσπάθησαν να λύσουν αυτό το πρόβλημα με την χρήση διαφόρων επιπλέον συσκευών όπως Bluetooth, RFID (Radio-Frequency Identification), βιντεοκαμερών, αισθητήρων φωτός και υπέρυθρων και πολλών άλλων. Λόγω κόστους και αδυναμίας συντήρησης, αυτές οι μέθοδοι παρέμειναν σε ερευνητικό επίπεδο.

Τα τελευταία χρόνια έχει επικρατήσει η μέθοδος καταγραφής των σημάτων από ασύρματα σημεία πρόσβασης (RSS signals - Received Signal Strength) από τα κοινά ήδη εγκατεστημένα σημεία πρόσβασης ασύρματης δικτύωσης (WLAN Access Points). Τα πλείστα συστήματα βασισμένα στα RSS σήματα έχουν δύο φάσεις για να γίνει η τοποθέτηση και η ανίχνευση του χρήστη [2]. Κατά την πρώτη φάση (offline-phase ή fingerprinting phase) πρέπει να παρθούν μετρήσεις σε διάφορες τοποθεσίες και να αποθηκευτούν οι ακριβή συντεταγμένες μαζί με τα WLAN APs (48-bit MAC διευθύνσεις) που εισακούονται και την αντίστοιχη ένταση ισχύς του σήματος. Αυτός ο συνδυασμός τοποθεσίας και WLAN AP αποτελεί ένα αποτύπωμα RSS (RSS fingerprint). Ακολούθως, στην δεύτερη φάση, φάση ανίχνευσης (online-phase ή tracking-phase), ο χρήστης κινείται στον χώρο στέλνοντας στο σύστημα τα MACs που εισακούει με στόχο να γίνει κάποιο ταίριασμα των μετρήσεων που λαμβάνει με τα αποθηκευμένα σήματα και να υπολογιστεί η θέση του χρήστη. Εδώ, στην δεύτερη φάση, είναι που διαφέρουν τα πλείστα συστήματα αφού χρησιμοποιούν διαφορετικούς αλγορίθμους και μεθόδους υπολογισμού της θέσης. Επίσης, δεν ήταν πολύς καιρός που άρχισε να μελετιέται και το θέμα διαχείρισης αυτών των αποτυπωμάτων για πιο αποδοτική επεξεργασία.

Όπως μπορεί κάποιος εύκολα να αντιληφθεί, η ραγδαία αύξηση των κινητών συσκευών αλλά και της συνεχόμενης εξάπλωσης και εγκατάστασης σημείων πρόσβασης ασύρματης δικτύωσης έχει ως αποτέλεσμα τα δεδομένα που πρέπει να διαχειρίζεται ένα σύστημα εσωτερικής τοποθέτησης φτάνουν μεγάλες ποσότητες. Πριν από μερικά χρόνια έγινε πρώτη αναφορά στα Μεγάλα Δεδομένα, Big Data. Με τον όρο Big Data αναφερόμαστε σε δεδομένα με μεγέθη πέραν του σύνηθες μεγέθους που μπορούν να χειριστούν τα κοινά συστήματα μέχρι τώρα ώστε να προσφέρουν τις υπηρεσίες τους με

χαμηλό χρόνο ανταπόκρισης. Τα μεγέθη αυτά συνήθως κυμαίνονται από μερικά Terabytes σε πολλαπλά Petabytes και συνεχώς αυξάνονται. Το 2001 ο αναλυτής Doug Laney (τόρα Gartner) έχει δώσει τον ορισμό των μεγάλων δεδομένων ως τα 3-V, Volume – Velocity – Variety, χαρακτηρίζοντας έτσι ότι τα μεγάλα δεδομένα αφορούν δεδομένα τεραστίων ποσοτήτων (Volume), τα οποία παράγονται σε απίστευτα γρήγορους ρυθμούς λόγω της τεχνολογίας σήμερα, αισθητήρες, κινητές συσκευές κλπ. (Velocity), αλλά επίσης δεν έχουν συγκεκριμένη δομή και μπορεί να είναι οτιδήποτε, από δομημένη μορφή αρχείων και σχημάτων βάσης μέχρι αδόμητα δεδομένα όπως οπτικοακουστικό υλικό και στατιστικές μετρήσεις (Variety). Αργότερα η εταιρεία SAS, πρόσθεσε στον ορισμό ακόμα δύο χαρακτηριστικά: α) μεταβλητότητα (Variability) εννοώντας την μη-συνοχή στην ταχύτητα παραγωγής των δεδομένων ανά χρονική περίοδο και β) πολυπλοκότητα (Complexity) εννοώντας ότι σήμερα τα δεδομένα έρχονται από απίστευτη ποικιλότητα συσκευών, συνεπώς η διαχείριση τους και ο συνδυασμός τους αποτελεί δύσκολο έργο.

Η εξέλιξη της τεχνολογίας και η έκρηξη υπηρεσιών και εφαρμογών έχει ανοίξει τις πόρτες για τους ερευνητές και όσους ασχολούνται με την Τεχνολογία. Ακόμα ένας τομέας που παρουσιάζει απίστευτη εξέλιξη και διαδίδεται με ταχύτητες φωτός είναι ο πληθοπορισμός (Crowdsourcing). Το Crowdsourcing είναι μια νέα τάση που επικρατεί όπου κάποιο πρόβλημα γίνεται γνωστό στο ευρύ κοινό με στόχο να γίνει πιο εύκολη η επίλυση του αφού θα μπορούν πολλοί άνθρωποι να δουλέψουν μαζί.

Συνοψίζοντας, μπορούμε να δούμε την σχέση που υπάρχει στους τρεις τομείς που έχω αναφέρει μέχρι τώρα, εσωτερική γεωτοποθέτηση, μεγάλα δεδομένα και πληθοπορισμός. Ένα σύστημα διαχείρισης δεδομένων για εσωτερική τοποθέτηση και πλοήγηση που θέλει να επικρατήσει και να εδραιωθεί στις μέρες μας πρέπει να έχει την κατάλληλη υποδομή ώστε να μπορεί να χειριστεί μεγάλα δεδομένα. Επιπλέον, ένας τρόπος με τον οποίο μπορεί να διεκπεραιωθεί η πρώτη φάση, φάση συλλογής αποτυπωμάτων, είναι μέσω του Crowdsourcing αφού αποτελεί μια από τις πιο χρονοβόρες και φορτικές διαδικασίες και αποτελεί ιδανικό πρόβλημα για τον πληθοπορισμό.

1.2 Ανασκόπηση συστήματος AnyPlace

Σε αυτή τη ατομική διπλωματική εργασία θα παρουσιαστεί το AnyPlace, ένα ολοκληρωμένο σύστημα γεωτοποθέτησης και γεωπλοήγησης σε εσωτερικούς χώρους με υποδομή για Μεγάλα Δεδομένα. Επιπλέον, έχει αναπτυχθεί τεχνική διαχείρισης των RSS αποτυπωμάτων ώστε να παράγεται μικρότερος ραδιοχάρτης (RSS Radiomap – δυσδιάστατος πίνακας με την μια διάσταση όλες τις τοποθεσίες και την άλλη διάσταση όλα τα WLAN APs) σε αριθμό αποτυπωμάτων και ασύρματων σημείων πρόσβασης, διατηρώντας ωστόσο την ακρίβεια και την ποιότητα της πρόβλεψης τοποθεσίας του χρήστη.

Συγκεκριμένα, έχει σχεδιαστεί και υλοποιηθεί υποδομή μεγάλων δεδομένων όπου υπάρχει ένα συγκρότημα εξυπηρετητών που παρέχουν μια διεπαφή προγραμματισμού εφαρμογών (Web2.0 API) μέσω της οποίας μπορεί να επικοινωνήσουν πελάτες με την υπηρεσία μας. Επιπλέον, το σύστημα υποστηρίζεται από NoSQL βάση δεδομένων (Couchbase) και ένα κατανεμημένο σύστημα διαχείρισης αρχείων (GlusterFS).

Εκτός από το back-end κομμάτι του συστήματος έχουν αναπτυχθεί δύο (2) εφαρμογές για κινητές συσκευές Android. Η πρώτη, *AnyPlace Logger*, χρησιμοποιείται κατά την πρώτη φάση γεωτοποθέτησης και αφορά την συλλογή RSS αποτυπωμάτων από οποιονδήποτε χρήστη διαθέτει την εφαρμογή, εφαρμόζοντας έτσι το μοντέλο του Crowdsourcing. Η δεύτερη, *AnyPlace Navigator*, είναι η κύρια εφαρμογή και προσφέρει γεωπλοήγηση και γεωτοποθέτηση στους χρήστες σε πραγματικό χρόνο εντός των κτιρίων που είναι ήδη χαρτογραφημένα στο σύστημα.

Επιπρόσθετα, έχουν αναπτυχθεί τρεις (3) διαδικτυακές εφαρμογές και απαιτούν τη χρήση ενός περιηγητή με υποστήριξη του προτύπου HTML5. Αρχίζοντας με την εφαρμογή *AnyPlace Developers*, η οποία παρέχει την δυνατότητα σε κάποιον να δοκιμάσει την διεπαφή προγραμματισμού που παρέχει το AnyPlace (API). Η χαρτογράφηση των κτιρίων γίνεται εξ' ολοκλήρου διαμέσου της εφαρμογής *AnyPlace Architect* και η τρίτη εφαρμογή, *AnyPlace Viewer*, αποτελεί την αντίστοιχη εφαρμογή του *AnyPlace Navigator* για το διαδίκτυο αφού μέσω της κάποιος μπορεί να προσπελάσει όλα τα χαρτογραφημένα κτίρια, να αναζητήσει αντικείμενα και αίθουσες όπως επίσης και να πλοηγηθεί ανάμεσα στα διάφορα σημεία ενδιαφέροντος.

1.3 Συνεισφορές

Μέσω αυτής της ατομικής διπλωματικής εργασίας έχω κάνει τρεις σημαντικές συνεισφορές:

1. Πρότεινα, σχεδίασα και υλοποίησα μια ολοκληρωμένη αρχιτεκτονική για σύστημα γεωτοποθέτησης και γεωπλοήγησης σε εσωτερικούς χώρους με τεχνικές υποστήριξης για διαχείριση μεγάλων δεδομένων.
2. Μοντελοποίησα το πρόβλημα της δημιουργίας ραδιοχάρτη αποτυπωμάτων από ενδείξεις ασυρμάτων σημείων πρόσβασης σε μεγάλα δεδομένα και πρότεινα την τεχνική *LPRadioScale* η οποία αποτιμήθηκε στο κεφάλαιο 7.
3. Υλοποιήθηκαν εφαρμογές που παρέχουν τις υπηρεσίες που προσφέρει το AnyPlace για έξυπνα κινητά Android (AnyPlace Logger, AnyPlace Navigator) όπως επίσης και για τον παγκόσμιο ιστό μέσω οποιουδήποτε περιηγητή που υποστηρίζει HTML5 (AnyPlace Viewer, AnyPlace Architect, AnyPlace Developers).

1.4 Περίγραμμα εργασίας

Στο πρώτο κεφάλαιο παρουσιάσαμε συνοπτικά την υποκίνηση αυτής της διπλωματικής εργασίας και αναφέραμε κάποιες βασικές έννοιες που θα αναφέρονται συχνά στο υπόλοιπο του κειμένου.

Στο δεύτερο κεφάλαιο θα προχωρήσουμε σε περισσότερες έννοιες και ορισμούς και θα παρουσιάσουμε συστήματα εσωτερικής τοποθέτησης και πλοήγησης παρόμοια με το AnyPlace αναφέροντας και τις κυριότερες διαφορές μας από αυτά. Επίσης, θα γίνει αναφορά σε σχετική βιβλιογραφία όσο αφορά τεχνικές διαχείρισης των αποτυπωμάτων RSS όπως και μια μικρή περιγραφή κατανεμημένων και παράλληλων συστημάτων και του μοντέλου προγραμματισμού MapReduce[3] ώστε να γίνει εύκολη η κατανόηση της δικής μας τεχνικής.

Στο τρίτο κεφάλαιο θα γίνει μια αναλυτική περιγραφή της αρχιτεκτονικής του AnyPlace παρουσιάζοντας τα επιμέρους κομμάτια που την αποτελούν, καθώς επίσης και μια συνοπτική αναπαράσταση της ροής της πληροφορίας από τη στιγμή που εισέρχεται στο σύστημα μέχρι την προβολή της στον χρήστη.

Στο τέταρτο κεφάλαιο περιγράφεται η τεχνική διαχείρισης αποτυπωμάτων *LPRadioScale* για μεγάλα δεδομένα αφού πρώτα γίνει μια εισαγωγή στο πλαίσιο προγραμματισμού Hadoop[4] στο οποίο βασίζεται και η τεχνική μας.

Ακολούθως, στο πέμπτο κεφάλαιο θα γίνει εκτενής περιγραφή των εφαρμογών για έξυπνες κινητές συσκευές Android[5] ενώ στο έκτο κεφάλαιο θα αναλυθεί η υποδομή που παρέχει τις διάφορες υπηρεσίες και οι διάφορες διαδικτυακές εφαρμογές.

Στη συνέχεια, στο κεφάλαιο 7 θα γίνει μια πειραματική αξιολόγηση του συστήματος, σχετικά όμως μόνο με την τεχνική διαχείρισης των αποτυπωμάτων και την δημιουργία ραδιοχάρτη για μεγάλα δεδομένα.

Στο όγδοο και τελευταίο κεφάλαιο θα παραθέσουμε τα τελικά συμπεράσματα σχετικά με το όλο σύστημα που αναπτύχθηκε καθώς επίσης και μελλοντικές επεκτάσεις και βελτιστοποιήσεις που μπορούν να γίνουν.

Κεφάλαιο 2

Σχετική Βιβλιογραφία

2.1	Εσωτερική γεωτοποθέτηση βασισμένη σε WLAN RSS	7
	Σχετικά συστήματα γεωτοποθέτησης για εσωτερικούς χώρους	10
	2.2.1 RADAR	12
2.2	2.2.2 MazeMap	13
	2.2.3 Airplace	15
	Τεχνικές διαχείρισης αποτυπωμάτων σημείων πρόσβασης	18
	Υπόβαθρο Παράλληλης – Κατανεμημένης Επεξεργασίας	24
2.3	2.4.1 Παράλληλα και Κατανεμημένα Συστήματα	24
2.4	2.4.2 Μοντέλα προγραμματισμού για κατανεμημένα συστήματα	27

2.1 Εσωτερική γεωτοποθέτηση βασισμένη σε WLAN RSS

Σε αυτό το κεφάλαιο θα εξηγηθούν διάφοροι όροι και έννοιες σχετικές με γεωτοποθέτηση σε εσωτερικούς χώρους.

2.1.1 Γενικές πληροφορίες για WLAN & RSS

Το πρότυπο WLAN IEEE 802.11b/g είναι το σύνηθες πρότυπο που χρησιμοποιείται για να παρέχεται ασύρματη πρόσβαση στο διαδίκτυο σε εσωτερικούς κυρίως χώρους ή καλύτερα σε μικρή έκταση γύρω από το σημείο πρόσβασης. Λειτουργεί στα 2.4GHz

και έχει δυναμικότητα μέχρι και 50m. Η ένδειξη RSS όπως προαναφέρθηκε μπορεί να ληφθεί χρησιμοποιώντας οποιαδήποτε συσκευή που υποστηρίζει το WLAN πρότυπο, κάτι το οποίο συμβαίνει με αν όχι όλες, τις πλείστες συσκευές σήμερα και κυρίως τα έξυπνα τηλέφωνα (Android, iPhone, Windows Phone κλπ.).

Η ένδειξη RSS (πιο σωστά RSSI – Received Signal Strength Indication) δείχνει την ένταση του σήματος από ένα συγκεκριμένο σημείο πρόσβασης και μετριέται σε dB με αρνητική τιμή μέτρησης, όπου η μεγαλύτερη τιμή δείχνει και πιο δυνατό σήμα.

Η χρήση των RSS έχει πλέον εδραιωθεί από τους ερευνητές ως η λύση για συστήματα τοποθέτησης. Οι κυριότερες τεχνικές για τον υπολογισμό της τοποθεσίας βάση των ενδείξεων RSS είναι οι παρακάτω.

1. *Triangulation*: Η ένδειξη RSS μπορεί να μετατραπεί σε απόσταση από ένα συγκεκριμένο σημείο πρόσβασης βάση θεωρητικών ή εμπειρικά αποδεδειγμένων μοντέλων διάδοσης του σήματος. Ακολουθώντας, χρησιμοποιώντας μετρήσεις από τουλάχιστον τρία (3) APs με γνωστές θέσεις μπορούμε να εκτιμήσουμε την τοποθεσία του χρήστη. Η μέθοδος αυτή δεν είναι και πολύ αποτελεσματική αφού η διάδοση κυμάτων σε εσωτερικούς χώρους είναι πολύ απρόβλεπτη με αποτέλεσμα να καθιστά δύσκολη την εύρεση ενός αξιόπιστου μοντέλου.
2. *Proximity*: Η μέθοδος αυτή βρίσκει το πιο δυνατό σήμα RSS από ένα συγκεκριμένο AP και εκτιμά ότι ο χρήστης βρίσκεται εντός της εμβέλειας αυτού του AP. Δεν είναι πολύ fine-grained (ακριβής) η μέθοδος αυτή αλλά είναι πολύ εύκολη στην υλοποίηση της.
3. *Scene Analysis*: Κατά την μέθοδο αυτή πρώτα συλλέγονται RSS μετρήσεις σε αρκετές γνωστές τοποθεσίες, τις οποίες ονομάζουμε αποτυπώματα (fingerprints), στην περιοχή που θέλουμε να γίνεται η ανίχνευση του χρήστη. Ακολουθώντας, ενώ ο χρήστης κινείται στον χώρο η εκτίμηση θέσης γίνεται με την σύγκριση των εκάστοτε ενδείξεων RSS από τα κοντινά σημεία πρόσβασης (APs) βάση τεχνικών αναγνώρισης μοτίβων. Αυτή η τεχνική έχει την

μεγαλύτερη ακρίβεια και εξαρτάται από τον αλγόριθμο σύγκρισης. Επίσης είναι η μέθοδος που χρησιμοποιούμε και στο σύστημα AnyPlace.

Η χρήση των RSS είναι πλέον η πιο διαδεδομένη τεχνική και η μόνη που μπορεί να παρέχει μεγάλη ακρίβεια στην εκτίμηση της θέσης του χρήστη. Όμως δεν είναι χωρίς τα προβλήματα της. Πιο κάτω παραθέτουμε κάποιους από τους λόγους που μπορεί να επηρεαστεί η ποιότητα της εκτίμησης της τοποθεσίας λόγω των RSS ενδείξεων.

1. Η συχνότητα εκπομπής του προτύπου WLAN IEEE 802.11 όπως αναφέρθηκε είναι τα 2.4GHz με αποτέλεσμα άλλες συσκευές που εκπέμπουν στις ίδιες συχνότητες, όπως κινητά τηλέφωνα, φούρνοι μικροκυμάτων και πολλές άλλες, να παρεμβάλουν τα σήματα και να προκαλούν θόρυβο στις μετρήσεις οι οποίες θα οδηγήσουν σε λανθασμένη εκτίμηση θέσεις. [6], [7]
2. Η θέση του χρήστη καθώς και η κατεύθυνση του μπορεί να επηρεάσουν αρνητικά τις ενδείξεις που λαμβάνει από κάθε σημείο πρόσβασης αφού το ανθρώπινο σώμα αποτελεί εμπόδιο στο πέρασμα των ράδιο κυμάτων.
3. Η ακρίβεια του συστήματος βασίζεται στις καταχωρημένες ενδείξεις με αποτέλεσμα τυχόν αλλαγές στις εγκαταστάσεις των κτιρίων (αλλαγή ασύρματων σημείων πρόσβασης) θα προκαλέσει σφάλματα και λάθος εκτιμήσεις από το σύστημα. Γι' αυτό και πρέπει να λαμβάνονται επαρκή αποτυπώματα πριν γίνει η ανίχνευση του χρήστη όπως επίσης και συλλογή αποτυπωμάτων μετά από κάθε μεγάλη αλλαγή στα εγκατεστημένα σημεία πρόσβασης.
4. Επίσης, οι κλιματολογικές συνθήκες και ο χρόνος μπορεί να επηρεάσουν τις RSS ενδείξεις στην ίδια τοποθεσία όπως δείχθηκε από προηγούμενες δουλειές [8], [9]

Παρόλα αυτά τα πλείστα συστήματα συνεχίζουν να χρησιμοποιούν RSS-βασισμένες μεθόδους, όπως και εμείς μέσα από το AnyPlace.

Προτού προχωρήσουμε στην σχετική βιβλιογραφία για παρόμοια συστήματα γεωτοποθέτησης σε εσωτερικούς χώρους, θα εξηγήσουμε τι είναι ο ραδιοχάρτης, λόγω του ότι θα γίνεται συχνή αναφορά σ' αυτόν στη συνέχεια αυτής της εργασίας.

2.1.2 Ραδιοχάρτης (Radio Map)

Είναι ένας δυσδιάστατος πίνακας του οποίου η κάθε σειρά εκφράζει την σχέση μεταξύ μιας τοποθεσίας (x,y) και τις ενδείξεις RSS (σε dBs) για κάθε ένα από τα AP που εισακούονται στο σημείο (x,y).

Με απλά λόγια είναι ένας τεράστιος πίνακας που έχει μια σειρά για κάθε μοναδική τοποθεσία για την οποία έχουμε μετρήσεις RSS και έχουμε μια στήλη για κάθε μοναδικό AP (MAC address) που εισακούστηκε σε τουλάχιστον μια τοποθεσία.

Για παράδειγμα το πιο κάτω αποτελεί ένα παράδειγμα ραδιοχάρτη:

Location	AP-1	AP-2	...	AP-N
X,Y – 1	-110	-60		-22
X,Y – 2	-78	-56		-34
...				-110
X,Y – M	-90	-110		-110

Πίνακας 2.1 : Παράδειγμα ραδιοχάρτη (Radio Map)

Στο πιο πάνω σχήμα έχουμε ένα ραδιοχάρτη που συσχετίζει τις τοποθεσίες X,Y-1 μέχρι και την τοποθεσία X,Y-M μαζί με τα σημεία πρόσβασης AP-1 μέχρι και AP-N. Στο κάθε κελί αναγράφεται η ένδειξη σήματος για την εκάστοτε τοποθεσία και το εκάστοτε σημείο πρόσβασης, με τις τιμές -110 να δείχνουν μη ύπαρξη σήματος του AP στην συγκεκριμένη τοποθεσία.

2.2 Σχετικά συστήματα γεωτοποθέτησης για εσωτερικούς χώρους

Σε αυτή την ενότητα θα μελετήσουμε διάφορα συστήματα εσωτερικής γεωτοποθέτησης και πλοήγησης παρόμοια με το σύστημα που παρουσιάζουμε σε αυτή τη διπλωματική εργασία. Επίσης στο τέλος θα γίνει μια συγκριτική σύνοψη των υπηρεσιών που παρέχουν αλλά και τι είναι αυτό που χρειάζεται και ακόμα λείπει από τα υπάρχον συστήματα.

Μέχρι το 2000, η βιβλιογραφία που υπήρχε αφορούσε συστήματα που χρησιμοποιούσαν επιπλέον συσκευές ή εξαρτήματα ώστε να επιτρέπουν την ανίχνευση του χρήστη εντός κτιρίων σε πραγματικό χρόνο. Για παράδειγμα, το *Active Badge*[10] το οποίο με τη σειρά του πρωτοπόρησε στον χώρο των υπηρεσιών χρήσης γεωγραφικής τοποθεσίας (location-aware services) χρησιμοποιούσε ειδικά εμβλήματα που εκπέμπαν υπέρυθρα σήματα (IR signals) ώστε ειδικοί αισθητήρες τοποθετημένοι σε διάφορες τοποθεσίες να μπορούν να ανιχνεύουν τον χρήστη ανά πάσα στιγμή.

Άλλο σύστημα που χρησιμοποιούσε επιπλέον εξαρτήματα ήταν το [11] στο οποίο ειδικοί πομποί υπέρυθρων (IR transmitters) ήταν τοποθετημένοι στο ταβάνι των κτιρίων. Ο χρήστης θα έπρεπε να φέρει μαζί του διαμορφωμένο οπτικό αισθητήρα στο κεφάλι του ώστε να λαμβάνει τα υπέρυθρα σήματα και να ενημερώνει το λογισμικό ανάλογα ώστε να γίνεται επιτρεπτή η ανίχνευση του σε πραγματικό χρόνο.

Αυτές οι προσεγγίσεις αν και αρκετά αποτελεσματικές έχουν πολλά προβλήματα και γι' αυτό και δεν ανελίχθηκαν πέρα της ερευνητικής περιοχής. Μερικά από τα σημαντικότερα προβλήματα τους είναι:

1. Δεν είναι επεκτάσιμες λύσεις λόγω του ότι οι υπέρυθρες ακτίνες δεν έχουν εμβέλεια σε μεγάλες αποστάσεις, άρα είτε θα έπρεπε να υπάρχουν πολύ περισσότεροι αισθητήρες ή θα είχαμε ανίχνευση του χρήστη σε μικρότερη γεωγραφικά περιοχή.
2. Υπάρχει σημαντικό κόστος στην εγκατάσταση όλων των απαιτούμενων συσκευών και αισθητήρων ώστε να παρέχεται επαρκής κάλυψη περιοχής αλλά

και στην συντήρηση και επιδιόρθωση όλων αυτών των συσκευών και εξαρτημάτων.

3. *Παρουσιάζει μειωμένη επίδοση* στην παρουσία ηλιακού φωτός λόγω της φύσης των υπέρυθρων ακτινών άρα δεν μπορεί να χρησιμοποιηθεί σε εσωτερικούς χώρους με πολλά ή μεγάλα παράθυρα.

Αυτοί είναι μόνο μερικοί από τους λόγους που τα εν λόγω συστήματα αλλά και πολλά άλλα της ίδιας φιλοσοφίας δεν πέτυχαν ποτέ.

Στη συνέχεια, θα παρουσιάσουμε κάποια συστήματα παρόμοια με το AnyPlace τα οποία είτε έχουν πετύχει είτε αποτέλεσαν πρωτεργάτες στον χώρο της γεωτοποθέτησης και πλοήγησης σε εσωτερικούς χώρους.

2.2.1 RADAR

Το RADAR [2] μπορεί να χαρακτηριστεί ως το σύστημα πρωτεργάτης στον χώρο της γεωτοποθέτησης με τη χρήση ασυρμάτων σημείων πρόσβασης (Wi-Fi Access Points) αφού τον καιρό που δημοσιεύτηκε δεν υπήρχε σχετική βιβλιογραφία και ήταν ένα από τα καινοτόμα συστήματα της εποχής.

Το σύστημα RADAR δημοσιεύτηκε το 2000 από το τμήμα ερευνών της Microsoft (Microsoft Research department) και αποτελεί το πρώτο σύστημα εύρεσης και τοποθέτησης του χρήστη σε πραγματικό χρόνο με τη χρήση ράδιο-συχνοτήτων από WLAN APs (Wireless Local Area Network Access Points).

Υπάρχουν δύο φάσεις στην λειτουργία του RADAR, η φάση εκπαίδευσης και η φάση ανίχνευσης. Κατά την φάση ανίχνευσης το σύστημα μαθαίνει για διάφορες τοποθεσίες τις ενδείξεις που επικρατούν και κατά τη φάση ανίχνευσης γίνεται σύγκριση των εκάστοτε ενδείξεων με αυτές που έχει γίνει η εκπαίδευση ώστε να υπολογιστεί η ακριβής τοποθεσία του χρήστη, ή καλύτερα να εκτιμηθεί.

Ακριβέστερα, το RADAR χρησιμοποιεί ασύρματους σταθμούς βάσης (base-stations) τοποθετημένους σε κάποια σημεία (3 στην δημοσίευσή τους) ώστε να καλύπτουν όλο τον εσωτερικό χώρο και να υπάρχει επικάλυψη μεταξύ των σταθμών. Ακολούθως, κατά την φάση εκπαίδευσης (offline-training phase) ο χρήστης χρησιμοποιώντας ένα

Windows-based κινητό διακομιστή αναμεταδίδει (broadcasts) πακέτα (beacons) ώστε οι σταθμοί βάσης να συλλέγουν και να υπολογίζουν την απόσταση από τον χρήστη. Επίσης, μαζί με τα πακέτα ο χρήστης σημειώνει στον διακομιστή την ακριβή του τοποθεσία πάνω στο χάρτη του κτιρίου η οποία και αποθηκεύεται μαζί με τις μετρήσεις από τους 3 σταθμούς.

Στην φάση ανίχνευσης (online-tracking phase), ο χρήστης κινείται στον χώρο και βάση των αναμεταδόσεων που κάνει ο διακομιστής, εκτελούνται κάποιοι αλγόριθμοι ώστε να υπολογιστεί ή να εκτιμηθεί η θέση του. Χρησιμοποιούνται triangulation-algorithms αλλά και K-Nearest Neighbors αλγόριθμοι όπου βρίσκουν τις K πιο κοντινές τοποθεσίες του χρήστη και κάνουν ένα μέσο όρο της τελικής εκτίμησης. Όπως φάνηκε στα πειράματα που έγιναν καλύτερα αποτελέσματα παρατηρούνται για μικρό K, λογικά, ώστε να μην λαμβάνονται τοποθεσίες απομακρυσμένες του χρήστη.

Επιπλέον, στη δημοσίευση του RADAR γίνεται αναφορά στην σημαντικότητα της κατεύθυνσης του χρήστη αφού τα σήματα ράδιο-συχνοτήτων πολλές φορές παρουσιάζονται με διαφορετική ένταση ανάλογα με το που βλέπει ο χρήστης για τον προφανή λόγο ότι το σώμα μπορεί να αποτελεί εμπόδιο για τα κύματα σήματος με αποτέλεσμα να φτάνει με μειωμένη ένταση στον διακομιστή, παρόμοια και στους σταθμούς βάσης. Επίσης, εφαρμόζονται κάποια μοντέλα προσέγγισης και πρόβλεψης της κίνησης του χρήστη ώστε να γίνεται ακριβέστερη εκτίμηση της τοποθεσίας του σε πραγματικό χρόνο.

2.2.2 MazeMap

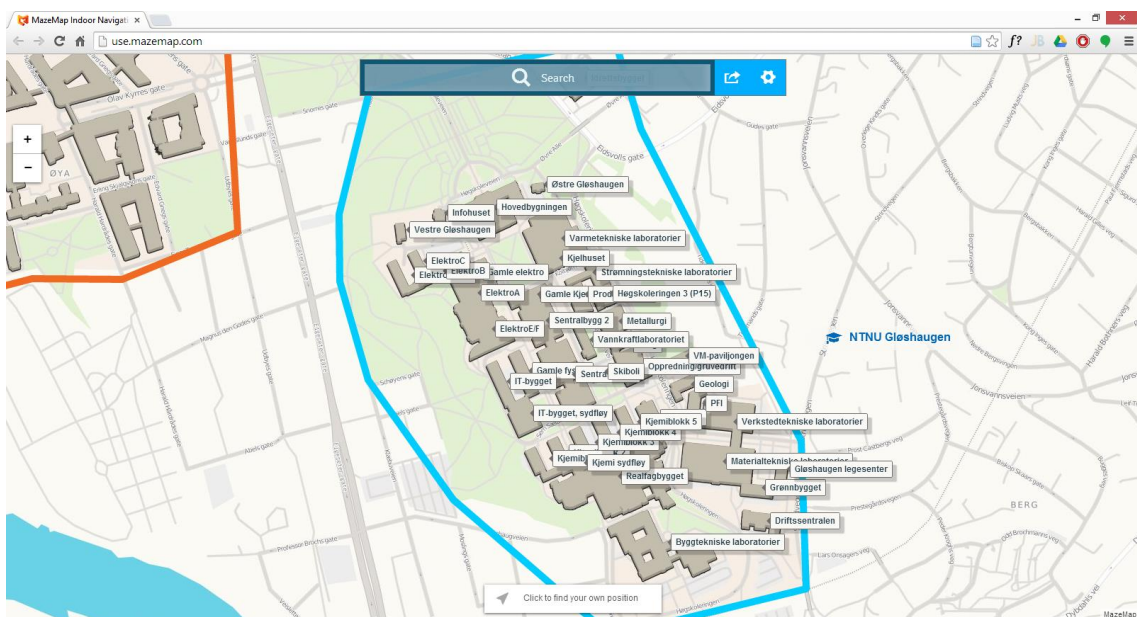
Το MazeMap [12] αποτελεί ένα από τα τελευταία συστήματα τοποθέτησης και πλοήγησης σε εσωτερικούς χώρους. Πρωτοδημοσιεύτηκε το 2011 ως CampusGuide από το NTNU – Norwegian University of Science and Technology και αργότερα μετονομάστηκε σε MazeMap.

Το MazeMap βασικά δημιουργήθηκε ώστε να γίνει πιο εύκολη η πλοήγηση των φοιτητών, καθηγητών και επισκεπτών του πανεπιστημίου NTNU αφού διαθέτει μεγάλο αριθμό αιθουσών και κτιρίων (350 000 m² έκταση με περίπου 60 κτίρια και 13 000

αίθουσες) με αποτέλεσμα να είναι δύσκολη η διακίνηση νεοφερμένων στο χώρο (περίπου 5000 άτομα το χρόνο).

Οι υπηρεσίες που παρέχει το MazeMap είναι:

1. Προβολή όλων των κτιρίων του πανεπιστημίου
2. Ανίχνευση χρήστη μέσα στα κτίρια
3. Αναζήτηση κτιρίου, δωματίου ή και αντικειμένων (χώρους υγιεινής κλπ.)
4. Παροχή οδηγιών πλοήγησης από την τοποθεσία του χρήστη στο αναζητηθέν σημείο



Εικόνα 2.1 : MazeMap εφαρμογή στο διαδίκτυο

Το πλεονέκτημα που έχει το MazeMap έναντι άλλων συστημάτων είναι ότι είναι διαθέσιμο σε όλους μέσω διαδικτύου με την ειδική εφαρμογή που τρέχει σε κάποιον περιηγητή (internet browser) αλλά επίσης και με τις διαθέσιμες εφαρμογές για έξυπνα κινητά Android και Apple iOS οι οποίες μπορούν να καταφορτωθούν δωρεάν από τους χρήστες.

Η χαρτογράφηση των κτηρίων γίνεται από τους διαχειριστές της υπηρεσίας και η εκτίμηση της τοποθεσίας του χρήστη σε πραγματικό χρόνο γίνεται με την χρήση του πυκνού δικτύου από τα ασύρματα σημεία πρόσβασης (Wi-Fi Access Points) που είναι ήδη τοποθετημένα στην πανεπιστημιούπολη.

Το σύστημα δεν παρέχει κάποιο εξειδικευμένο αλγόριθμο τοποθέτησης αφού χρησιμοποιεί απλούς αλγορίθμους *trilateration* οι οποίοι βάση των ενδείξεων στα γύρω σημεία πρόσβασης υπολογίζουν την θέση του χρήστη. Η εκτιμώμενη τοποθεσία έχει περίπου 5-10 μέτρα απόκλιση και επηρεάζεται από τον τρόπο που είναι τοποθετημένα τα σημεία πρόσβασης αλλά και την ισχύ των σημάτων τους.

Το MazeMap δεν έχει σκοπό την επινόηση περίτεχνων αλγορίθμων για τοποθέτηση του χρήστη σε εσωτερικούς χώρους με ασύρματα μέσα, αλλά την μελέτη και την συλλογή στατιστικών στοιχείων για συστήματα εσωτερικής τοποθέτησης και πλοήγησης που χρησιμοποιούνται σε πραγματικούς χώρους (όχι ερευνητικά πειράματα) αλλά επίσης και την εύρεση προτύπων για τη χρήση των εν λόγω συστημάτων από τους χρήστες, στη συγκεκριμένη περίπτωση μας χρήστες σε πανεπιστημιακό χώρο.

Όπως ανέφερα πιο πριν το MazeMap δεν έχει ερευνητική προσέγγιση στην εσωτερική τοποθέτηση όσο στα στατιστική χρήσης του συστήματος. Ο λόγος που το θέτω στην σχετική βιβλιογραφία όμως είναι διότι αποτελεί ένα από τα σύγχρονα συστήματα που αφορούν εσωτερική τοποθέτηση και πλοήγηση σε πραγματικό χώρο και κατά τη γνώμη μου παρέχει αρκετά εύχρηστη διεπαφή στους χρήστες μέσω των εφαρμογών του στο διαδίκτυο αλλά και στα έξυπνα τηλέφωνα [13].

2.2.3 Airplace

Το Airplace[14] αποτελεί τον πρόγονο του AnyPlace και δημοσιεύτηκε το 2012 από το Πανεπιστήμιο Κύπρου σε συνεργασία με το ερευνητικό κέντρο KIOS στα πλαίσια του διεθνές συνεδρίου Mobile Data Management (MDM '12) όπου και κέρδισε τον τίτλο του καλύτερου demo (Best demo award).

Το Airplace ήταν ένα απλό σύστημα εσωτερικής γεωτοποθέτησης που χρησιμοποιούσε και αυτό την τεχνική ανάλυσης περιοχής (Scene Analysis) συλλέγοντας αποτυπώματα RSS από τα γύρω κοντινά σημεία πρόσβασης. Σκοπός του ήταν η μειωμένη επικοινωνία με τον εξυπηρετητή αλλά και η δυνατότητα ανίχνευσης της θέσης του χρήστη εξ' ολοκλήρου από την συσκευή αφού ολόκληρος ο ραδιοχάρτης φορτωνόταν στην συσκευή έτσι αποφεύγονταν περαιτέρω μηνύματα με στον εξυπηρετητή.

Υλοποίηση του Airplace έγινε στα έξυπνα κινητά Android ενώ ο εξυπηρετητής ήταν γραμμένος σε Java και έτρεχε σε κοινούς υπολογιστές.

Η αρχιτεκτονική του Airplace αποτελείται από:

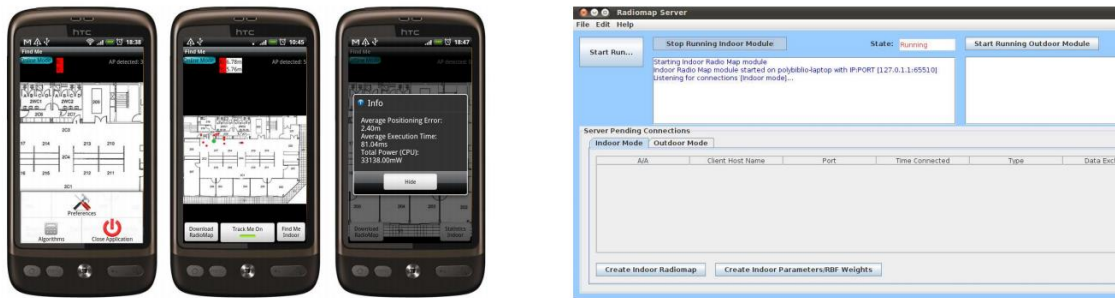
1. *RSS Logger Application*: Εφαρμογή που αναπτύχθηκε πάνω από το Android RSS API και χρησίμευε στην συλλογή των RSS αποτυπωμάτων σε συγκεκριμένες τοποθεσίες σε προκαθορισμένα χρονικά διαστήματα. Οι ενδείξεις καταγράφονταν σε αρχείο το οποίο αργότερα αποστέλλεται στον εξυπηρετητή διανομής ώστε να δημιουργηθεί ο ραδιοχάρτης.
2. *Find Me Application*: Η κύρια εφαρμογή που κρατούσε ο χρήστης σε μια Android συσκευή. Η εφαρμογή αυτή ζητούσε από τον εξυπηρετητή να τις αποσταλεί ο ραδιοχάρτης της περιοχής και πλέον δεν υπήρχε άλλη επικοινωνία ή μετάδοση πληροφοριών του χρήστη. Αφού ο ραδιοχάρτης υπήρχε στην συσκευή πλέον ο χρήστης μπορούσε να επιλέξει έναν από τους διάφορους αλγόριθμους τοποθέτησης που παρείχε η εφαρμογή ώστε βάση των εκάστοτε ενδείξεων RSS καθώς κινείται στον χώρο να γίνεται εκτίμηση της θέσης του.
3. *Εξυπηρετητής Διανομής*: Ο εξυπηρετητής αυτός ήταν υπεύθυνος για την διανομή του ραδιοχάρτη στην εφαρμογή *Find Me* μετά από το κατάλληλο αίτημα. Επίσης με το ανέβασμα του αρχείου με της μετρήσεις RSS από τον *Logger* δημιουργούσε τον ραδιοχάρτη της περιοχής.

Επίσης υπάρχει και κάποια ειδική έκδοση του Airplace συστήματος με την Ταϊβανέζικη εταιρεία Cywee Corporation Ltd – Taipei, Taiwan. Αυτή η έκδοση Airplace χρησιμοποιεί επιπλέον εξειδικευμένους αισθητήρες για πιο ακριβή εκτίμηση της τοποθεσίας του χρήστη με υπολογισμό της θέσης του σε πραγματικό χρόνο. Η έκδοση αυτή έχει κερδίσει την 2^η θέση στο συνέδριο IPSN'14 (Information Processing and Sensor Networks) στην κατηγορία τοποθέτησης σε εσωτερικούς χώρους. Στη συνέχεια παρουσιάζεται ένα απλό σενάριο χρήσης του Airplace.

Φάση συλλογής αποτυπωμάτων:

1. Ο χρήστης εισέρχεται στο κτίριο χαρτογράφησης και μέσω της εφαρμογής *Logger* επιλέγει από το σχέδιο ορόφου (φορτώνεται offline πάνω στην δευτερεύουσα μνήμη του κινητού)

2. Ακολούθως κινείται στον χώρο και σημειώνοντας πάνω στον χάρτη ορόφου την ακριβή του τοποθεσία παίρνει τις κατάλληλες μετρήσεις από τα γύρω σημεία πρόσβασης, καταγράφοντας τα στο αρχείο (rss-log)
3. Αφού τελειώσει με τις μετρήσεις του σε όλο το χώρο του ορόφου, ανεβάζει το αρχείο με τις μετρήσεις στον εξυπηρετητή



Εικόνα 2.2 : Airplace FindMe & Logger (αριστερά), Εξυπηρετητής διανομής (δεξιά)

Ακολούθως, εκτελούμε στον εξυπηρετητή την διαδικασία δημιουργίας του ραδιοχάρτη ώστε να λάβει υπόψη του τα νέα αποτυπώματα που ανεβάζτηκαν.

Φάση ανίχνευσης και τοποθέτησης:

1. Ο χρήστης χρησιμοποιώντας την εφαρμογή *Find Me* εισέρχεται στο κτίριο και ζητά από τον εξυπηρετητή να του στείλει ολόκληρο τον ραδιοχάρτη
2. Ακολούθως φορτώνει από την δευτερεύουσα μνήμη της συσκευής τον χάρτη ορόφου και επιλέγει έναν από τους αλγορίθμους τοποθέτησης που ήταν υλοποιημένοι
3. Κινείται στο χώρο και η εφαρμογή υπολογίζει ανάλογα την θέση του χρήστη βάση των σημείων πρόσβασης που ακούει

Το Airplace δίνει περισσότερη έμφαση στην ακρίβεια τοποθέτησης και στην ανωνυμία του χρήστη, παρά στην ευκολία χρήσης της εφαρμογής και της διαχείρισης των δεδομένων. Εννοώντας, το φόρτωμα του χάρτη απευθείας στην συσκευή αλλά και το κατέβασμα ολόκληρου του ραδιοχάρτη στην συσκευή. Αυτό για τεράστιου μεγέθους ραδιοχάρτη δεν θα ήταν και πολύ καλό αφού θα γινόταν πολύ αργή η επεξεργασία και ο υπολογισμός της τοποθεσίας του χρήστη.

Όλα τα σχετικά συστήματα γεωτοποθέτησης σε εσωτερικούς χώρους έχουν τα θετικά τους και τα αρνητικά τους. Κάποια σημεία που υστερούν όλα είναι η διαχείριση της πληροφορίας που συλλέγεται, των αποτυπωμάτων για παράδειγμα. Επίσης, η ευκολία συλλογής αποτυπωμάτων σε κάθε κτίριο και η παροχή σχεδίου ορόφου είναι κάτι που δεν υπάρχει σε όλα τα συστήματα. Το RADAR για παράδειγμα απαιτεί σταθμούς πρόσβασης σε προκαθορισμένα σημεία, αντίθετα με τα υπόλοιπα συστήματα που εκμεταλλεύονται τα ήδη υπάρχον WLAN APs.

Το σύστημα AnyPlace έρχεται για να γεφυρώσει το κενό ανάμεσα στα διάφορα συστήματα και να βελτιώσει τομείς που υστερούν. Έχοντας παρόμοια λειτουργία με τον *Airplace Logger* επιτρέπει την συλλογή αποτυπωμάτων από τον οποιοδήποτε (Crowdsourcing) και κάνει το φόρτωμα των ορόφων και των κτιρίων πιο αυτοματοποιημένο μέσω της υπηρεσίας που παρέχεται από το AnyPlace API. Επίσης, έχει εξελιγμένη τεχνική διαχείρισης δεδομένων με Big Data υποδομή και τεχνικές φιλτραρίσματος των αποτυπωμάτων για δημιουργία μικρότερων ραδιοχάρτων. Επίσης, παρέχει εφαρμογές για συσκευές Android και εφαρμογές για τον παγκόσμιο ιστό όπως κάνει και το MazeMap προσφέροντας όμως και μια επιπλέον εφαρμογή, τον AnyPlace Architect, που κάνει την διαδικασία χαρτογράφησης εύκολη και προσιτή για τον οποιοδήποτε.

Πιο κάτω βλέπουμε ένα συγκριτικό πίνακα για όλα τα προαναφερθέν συστήματα.

	Positioning Client	Crowdsourced RSS logging	Mapping Tools	Web Client	RSS Management techniques	Sophisticated Positioning
AnyPlace	Smartphone	Yes	Yes	Yes	Yes	Yes
RADAR	PC	No	No	No	No	No
MazeMap	Smartphone	No	No	Yes	No	No
Airplace	Smartphone	Yes	No	No	No	Yes

Πίνακας 2.2 : Συγκριτικός πίνακας των διάφορων συστημάτων εσωτερικής τοποθέτησης

2.3 Τεχνικές διαχείρισης αποτυπωμάτων σημείων πρόσβασης

Στην προηγούμενη ενότητα αναφέραμε σχετική βιβλιογραφία όσο αφορά συστήματα εσωτερικής γεωτοποθέτησης και πλοήγησης. Μέχρι τώρα σχεδόν όλα τα συστήματα επικεντρώνονταν στους αλγόριθμους και τις τεχνικές εκτίμησης της τοποθεσίας του χρήστη με τη χρήση του απλού ραδιοχάρτη (όλες οι τοποθεσίες και όλα τα σημεία πρόσβασης). Κανένα σύστημα δεν ασχολήθηκε με τον τρόπο που διαχειρίζονται σωστά τα αποτυπώματα που συλλέγονται ώστε το σύστημα να είναι επεκτάσιμο και συνεπές με το πέρασμα του χρόνου και την αύξηση των δεδομένων.

Σε αυτή την ενότητα θα επικεντρωθούμε σε σχετική δουλειά που έγινε πρόσφατα, μόλις πέρυσι το 2013, και αφορά διάφορες τεχνικές διαχείρισης των αποτυπωμάτων ώστε να είναι εφικτή η επεκτασιμότητα (scalability) και η βιωσιμότητα (viability) του συστήματος με τη συλλογή αποτυπωμάτων από πολλούς χρήστες σε μεγάλες χρονικές περιόδους. Επίσης, έμφαση δίνεται και στην διατήρηση της συνέπειας (consistency) όσο αφορά τις εκτιμήσεις για την τοποθεσία του χρήστη ανεξάρτητα συσκευής, χρόνου και κατεύθυνσης.

Όπως ανέφερα προηγουμένως πολλά συστήματα αρχίζουν να κατευθύνονται προς τον πληθοπωρισμό με αποτέλεσμα το μέγεθος των δεδομένων, αριθμός αποτυπωμάτων, να φτάνει μεγάλες ποσότητες. Με την εξάπλωση του Wi-Fi σήμερα δεν είναι απίθανο σε κάθε τοποθεσία να εισακούονται περισσότερα από 10-20 APs και σε ένα ολόκληρο κτίριο να εντοπίζονται συνολικά πέραν των 50-100 APs. Αυτό, σε συνδυασμό με την έκταση μεγάλων κτιρίων που συνεπάγεται πολλές τοποθεσίες για να συλλεχθούν αποτυπώματα, οδηγεί σε τεράστιους ραδιοχάρτες.

Λόγω των παραπάνω έχει αρχίσει να μελετάται το θέμα Επεκτασιμότητας ενάντια της Συνέπειας, *Scalability vs Consistency*. Οι ερευνητές έχουν μελετήσει διάφορες τεχνικές ώστε να περιορίσουν το μέγεθος του ραδιοχάρτη διατηρώντας ωστόσο την ποιότητα των εκτιμήσεων της τοποθεσίας του χρήστη όσο γίνεται ανεξάρτητα του πλαισίου του (συσκευή, χρόνος, καιρός, κλπ).

Παρακάτω παρατίθενται δύο δουλειές από σχετικά συνέδρια το 2013.

2.3.1 Scalable and consistent radio map management scheme for participatory sensing-based Wi-Fi fingerprinting (IPSN '13)

Αυτή η δουλειά [15] έχει σκοπό την μείωση του αριθμού των αποτυπωμάτων ανά τοποθεσία που συλλέχθηκαν αποτυπώματα, ώστε να μικραίνει το μέγεθος του ραδιοχάρτη από άποψη σειρών, συνεπώς και υπολογισμών κατά την εκτίμηση τοποθεσίας.

Αναφέρεται έντονα και πάλι η σχέση μεταξύ επεκτασιμότητας και συνέπειας δείχνοντας με πειραματικά πειράματα ότι καθώς αυξάνεται το μέγεθος του ραδιοχάρτη (αριθμός αποτυπωμάτων ανά τοποθεσία σε διάφορα χρονικά διαστήματα) αυξάνεται και η συνέπεια των αποτελεσμάτων των αλγορίθμων τοποθέτησης, επηρεάζοντας αρνητικά όμως την επεκτασιμότητα του συστήματος.

Η τεχνική που προτείνουν αφορά ομαδοποίηση των αποτυπωμάτων και επιλογή μόνο ενός αποτυπώματος που θα αποτελεί τον αντιπρόσωπο κάθε ομάδας. Οι δύο φάσεις της τεχνικής περιγράφονται συνοπτικά πιο κάτω.

Στην πρώτη φάση αυτή δημιουργείται ένας δυσδιάστατος πίνακας T_s , στον οποίο κάθε σειρά αντιπροσωπεύει ένα αποτύπωμα και κάθε στήλη αντιπροσωπεύει όλα τα υπόλοιπα αποτυπώματα. Η τιμή σε κάθε κελί (row I , column J) του πίνακα είναι η ομοιότητα του αποτυπώματος I σε σχέση με το αποτύπωμα J . Για τον υπολογισμό της ομοιότητας χρησιμοποιείται το *Tanimoto coefficient*, το οποίο αποδίδει το *Jaccard coefficient* δεδομένων δύο διανυσμάτων.

Στην δεύτερη φάση της τεχνικής εφαρμόζεται η ομαδοποίηση των αποτυπωμάτων βάση του πίνακα ομοιότητας που παράχθηκε στην 1^η φάση. Ο αλγόριθμος ομαδοποίησης που χρησιμοποιείται στην δουλειά αυτή είναι ο DBSCAN [16] ο οποίος δεν έχει προκαθορισμένο αριθμό ομάδων και μπορεί να φιλτράρει στοιχεία θορύβου (noisy objects).

Ο DBSCAN απαιτεί δύο παραμέτρους: α) την τιμή ε και β) την τιμή $MinPts$. Η τιμή ε καθορίζει την απόσταση πυκνότητας δύο σημείων (density-reachable distance) και λόγω του ότι χρησιμοποιείται ομοιότητα σημείων και όχι απόσταση, καθορίζεται ότι δύο σημεία είναι προσβάσιμα (density-reachable) εάν η ομοιότητα τους είναι μεγαλύτερη του ε ($Similarity(a,b) > \varepsilon$). Η τιμή $MinPts$ καθορίζει τον ελάχιστο αριθμό στοιχείων που χρειάζονται ώστε να σχηματιστεί μια ομάδα.

Ακολούθως αφού γίνει το clustering επιλέγεται ένα αποτύπωμα από κάθε ομάδα (cluster) το οποίο θα αντιπροσωπεύει την ομάδα στον τελικό ραδιοχάρτη. Η επιλογή του αποτυπώματος αντιπροσώπου γίνεται βάση του αθροίσματος των ομοιοτήτων κάθε αποτυπώματος σε σχέση με τα άλλα της ομάδας του. Το αποτύπωμα με το μεγαλύτερο άθροισμα είναι και αυτό που θα επιλεγεί για την ομάδα. Τέλος, ο ραδιοχάρτης που δημιουργείται αποτελείται από τους αντιπροσώπους κάθε ομάδας.

Με πειραματικές δοκιμές κατέληξαν ότι για το δικό τους σκηνικό οι πιο αποδοτικές τιμές για τις παραμέτρους του DBSCAN είναι $\varepsilon=0.95$ και $MinPts=3$, ενώ τιμές του $MinPts$ πέραν του 4 δεν ήταν λογικές αφού δεν δημιουργούσε αρκετές ομάδες και χειροτέρευε τις εκτιμήσεις. Επίσης, με $MinPts=2$ η ακρίβεια δεν ήταν πολύ καλύτερη όμως ο αριθμός των ομάδων ήταν πολύ μεγαλύτερος με αποτέλεσμα να αποτελεί πρόβλημα επεκτασιμότητας.

Επίσης, σημειώνεται ότι αντί του DBSCAN μπορεί να χρησιμοποιηθεί οποιοσδήποτε άλλος αλγόριθμος ομαδοποίησης (clustering algorithm).

2.3.2 Removing Useless APs and Fingerprints from Wi-Fi Indoor Positioning Radio Maps (IPIN '13)

Η δεύτερη δουλειά [17] που θα περιγράψουμε είναι από το University of Minho στην Πορτογαλία και δημοσιεύτηκε πέρυσι στο διεθνές συνέδριο Indoor Positioning and Navigation 2013, όπου δημοσιεύτηκε και το πρωτότυπο του δικού μας συστήματος AnyPlace.

Σε αυτήν την δουλειά σκοπός είναι η αφαίρεση σημείων πρόσβασης από τον ραδιοχάρτη ώστε να μειωθεί το μέγεθος του και πάλι αλλά προχωρούν και σε μια δεύτερη πρόταση όπου προσπαθούν να αφαιρέσουν ασήμαντα ή «κακά» αποτυπώματα ακραίων περιπτώσεων (bad outliers).

Φιλτράρισμα σημείων πρόσβασης (APs)

Μέσα από διάφορες πειραματικές μετρήσεις που έγιναν στην δουλειά αυτή φάνηκε ότι δεν είναι χρήσιμες όλες οι μετρήσεις από όλα τα σημεία πρόσβασης και πολλές φορές το μοναδικό πράγμα που πετυχαίνουν είναι την αύξηση το υ χρόνου επεξεργασίας λόγω επιπλέον πολυπλοκότητας του ραδιοχάρτη.

Έτσι λοιπόν παρουσιάζεται μια τεχνική η οποία αφαιρεί κάποια σημεία πρόσβασης ώστε να μείνουν μόνο όσα είναι αντιπροσωπευτικά του χώρου που εξετάζουμε.

Αναφέρονται τρία φιλτραρίσματα στην τεχνική αυτή και παρουσιάζονται παρακάτω:

1. Αριθμός Μοναδικών RSSI ενδείξεων (Number of distinct RSSI values)

Εκτιμάται ότι ένα σημείο πρόσβασης αποτελεί «καλό» όταν έχει καλή κατανομή ενδείξεων στις τοποθεσίες που εισακούεται. Έτσι, όσα APs έχουν λιγότερες από 10 διαφορετικές ενδείξεις RSSI απορρίπτονται.

2. Ποσοστό ελλειπόντων RSSI ενδείξεων (Percentage of missing RSSI values)

Υπάρχουν πολλά σημεία πρόσβασης τα οποία δεν εισακούονται από όλες τις τοποθεσίες, όταν όμως η παρουσία τους είναι μηδαμινή το μοναδικό αποτέλεσμα είναι η αυξημένη πολυπλοκότητα. Έτσι, όσα σημεία πρόσβασης δεν έχουν ενδείξεις για μεγαλύτερο του 80% του αριθμού των αποτυπωμάτων, απορρίπτονται.

3. Καθολική κανονική απόκλιση (STDEV-overall standard deviation)

Άλλο κριτήριο που εφαρμόζεται βασίζεται στο ότι θέλουμε σημεία πρόσβασης που έχουν καλή κατανομή τιμών, αφού μεγάλη κανονική απόκλιση συνεπάγεται πιο ποικίλες ενδείξεις σε διαφορετικές τοποθεσίες κάνοντας πιο γρήγορη την

εκτίμηση τοποθεσίας. Ενώ αντίθετα, ένα σημείο πρόσβασης που έχει όλες τις ενδείξεις του παρόμοιες δεν προσφέρει κάτι ουσιαστικό την επεξεργασία και δυσχεράνει τον υπολογισμό της θέσης του χρήστη. Έτσι, όσα APs έχουν μικρότερη κανονική απόκλιση από 5, απορρίπτονται.

<i>Scheme</i>	<i>#Distinct values</i>	<i>% missing values</i>	<i>STDEV</i>	<i>Useful APs</i>
A	5	95	2	21
B	10	80	5	16
C	15	90	3	20

Πίνακας 2.3 : Αποτελέσματα τελικών AP μετά το φιλτράρισμα των 50 αρχικών, όπως φαίνεται στο συγκεκριμένο paper

Επίσης, εκτός από το φιλτράρισμα των ασήμαντων σημείων πρόσβασης, δοκιμές έγιναν και για την τιμή που πρέπει να δίνετε στα σημεία πρόσβασης που δεν εισακούονται από κάποια τοποθεσία. Δοκιμάστηκαν οι τιμές -85, -95, -100, -110 και φάνηκε ότι η τιμή -85 είχε τα καλύτερα αποτελέσματα όσο αφορά την εκτίμηση της θέσης του χρήστη.

Dataset	Before filtering	After Aps Filtering	Gain
-85	68.8%	72.5%	3.7
-95	70.9%	70.3%	-0.6
-100	70.9%	70.3%	-0.6
-110	70.9%	70.3%	-0.6
NaN	66.3%	65.8%	-0.5

Πίνακας 2.4 : Αποτελέσματα με διαφορετικές προκαθορισμένες τιμές για APs που δεν εισακούονται

Φιλτράρισμα αποτυπωμάτων (RSS fingerprints)

Το φιλτράρισμα των αποτυπωμάτων γίνεται με τη χρήση ειδικών αλγορίθμων κατάταξης (classification), συγκεκριμένα του J48 αλγορίθμου μέσω της σουίτας προγραμμάτων WEKA. Συνοπτικά, κατατάσσουν τα αποτυπώματα στα δωμάτια που

αντιστοιχούν και τα αποτυπώματα που δεν εμπίπτουν σε κάποιο δωμάτιο απορρίπτονται.

Όπως αντιλαμβάνεστε, η ερευνητική δράση στην περιοχή διαχείρισης των αποτυπωμάτων είναι ακόμα πρώιμη με μόλις πέρυσι να γίνεται δημοσίευση δουλειών με ουσιαστικά αποτελέσματα. Στο AnyPlace, υλοποιήσαμε παρόμοια τεχνική όπως αυτές που παρουσιάστηκαν η οποία θα αναλυθεί στο κεφάλαιο 4.

2.4 Υπόβαθρο Παράλληλης – Κατανεμημένης Επεξεργασίας

Σε αυτή την ενότητα θα δούμε συνοπτικά διάφορες έννοιες ώστε να αποκτηθεί το απαραίτητο θεωρητικό υπόβαθρο που θα διευκολύνει την κατανόηση του αλγορίθμου αυτής της διπλωματικής εργασίας και το μοντέλο συστήματος που επιλέχτηκε.

2.4.1 Παράλληλα και Κατανεμημένα Συστήματα

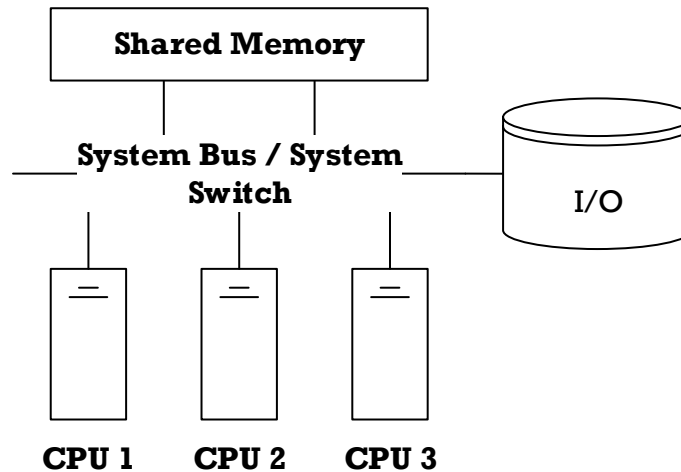
Με το πέρασμα των χρόνων ο αριθμός των δεδομένων που επεξεργάζονται οι διάφοροι οργανισμοί αυξάνεται με εκθετικούς ρυθμούς με συνεπακόλουθο και ο χρόνος εκτέλεσης των προγραμμάτων να φτάνει απαγορευτικά επίπεδα ή ακόμα χειρότερα κάποια προγράμματα να μην τερματίζουν ποτέ.

Αυτό είχε ως αποτέλεσμα την στροφή της επιστήμης προς την Παράλληλη επεξεργασία. Παράλληλη επεξεργασία σημαίνει ότι μοιράζουμε το πρόβλημα σε μικρότερα υπό-προβλήματα τα οποία εκτελούνται παράλληλα, ταυτόχρονα, ώστε να μειώσουμε τον ολικό χρόνο εκτέλεσης. Υπάρχουν διάφορες κατηγορίες παράλληλης επεξεργασίας οι οποίες διαφέρουν στον τρόπο επικοινωνίας και την απόσταση μεταξύ των υπολογιστικών μονάδων που συντελούν το σύστημα. Τα δύο είδη συστημάτων που εδραιώθηκαν μέχρι σήμερα παρουσιάζονται πολύ επιγραμματικά στη συνέχεια.

Παράλληλα Συστήματα κοινής μνήμης (Shared memory Systems)

Τα παράλληλα συστήματα είναι υπολογιστικές μονάδες συνδεδεμένες μεταξύ τους σε πολύ μικρές αποστάσεις ($< \sim 0.5m$) οι οποίες μοιράζονται διάφορους υπολογιστικούς πόρους, και ειδικά την κύρια μνήμη (shared memory) η οποία είναι και το μέσο επικοινωνίας μεταξύ τους. Παραδείγματα παράλληλων συστημάτων είναι οι πολυπύρηντοι υπολογιστές όπου έχουμε πολλούς πυρήνες (cores) στο ίδιο επεξεργαστή (processor-cpu) ή ακόμα και οι υπέρ-υπολογιστές οι αποτελούνται από πολλούς πολυπύρηντες επεξεργαστές συνδεδεμένους έχοντας ενιαία κοινή κύρια μνήμη.

Κύρια μονάδα επεξεργασίας σε αυτά τα συστήματα είναι ο πυρήνας (processor core), είτε λογικός είτε φυσικός. Μέσο επικοινωνίας μεταξύ των πυρήνων είναι η κύρια μνήμη. Αφού όλοι οι πυρήνες έχουν πρόσβαση σε όλη την έκταση της μνήμης, για να επικοινωνήσει ο πυρήνας A με τον πυρήνα B, θα γράψει κάτι σε ένα κελί διεύθυνσης στην μνήμη ο A και ακολούθως ο B θα διαβάσει από την ίδια διεύθυνση ώστε να μάθει την τιμή που έγραψε ο A.



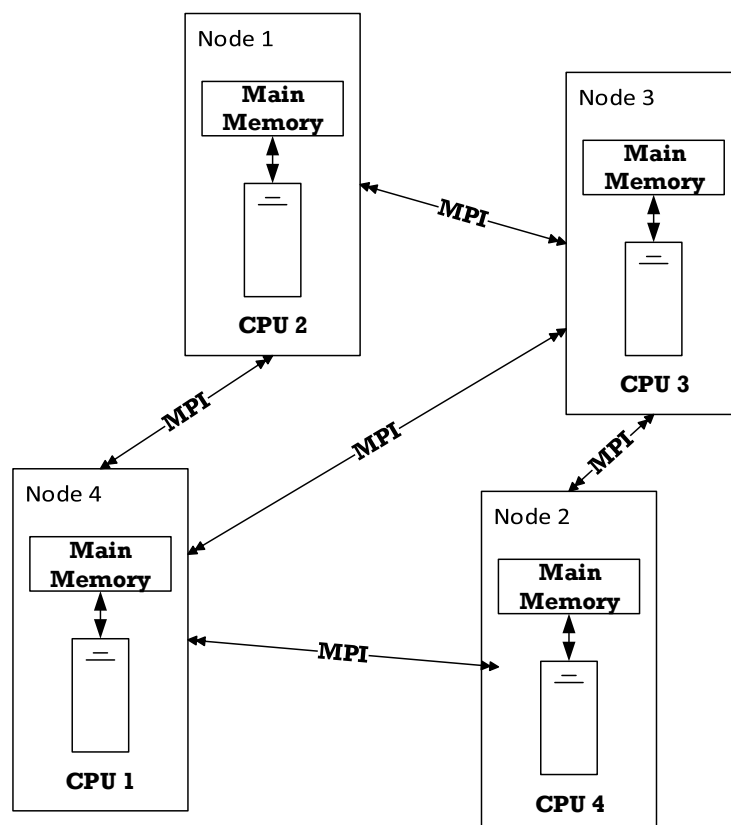
Σχήμα 2.1 : Ένα Shared Memory σύστημα όπου η επικοινωνία γίνεται με εγγραφές σε κοινή κύρια μνήμη

Κατανεμημένα Συστήματα (Distributed Systems)

Τα κατανεμημένα συστήματα είναι τα συστήματα όπου οι υπολογιστικές μονάδες που συντελούν το σύστημα δεν είναι κατ' ανάγκη κοντά η μια στην άλλη, και συνήθως

βρίσκονται αρκετά μακριά ακόμα και σε διαφορετικές χώρες και είναι συνδεδεμένες μεταξύ τους με κοινούς τρόπους δικτύωσης (ως επί των πλείστων TCP). Με αυτό τον τρόπο μπορούμε να έχουμε διάσπαρτες υπολογιστικές μονάδες, συνήθως πολυπύρηνες επεξεργαστές οι οποίες επικοινωνούν μεταξύ τους και δημιουργούν ένα μεγάλης ισχύς υπολογιστικό σύστημα. Είναι προφανές ότι στα κατακεκομμένα συστήματα δεν μπορεί να γίνει χρήση κοινής μνήμης σαν μέσω επικοινωνίας αφού ένας υπολογιστής δεν έχει απαραίτητα πρόσβαση σε πόρους άλλου υπολογιστή του συστήματος. Ο τρόπος επικοινωνίας μεταξύ των υπολογιστικών μονάδων στα συστήματα αυτά είναι η ανταλλαγή μηνυμάτων διαμέσου του δικτύου επικοινωνίας.

Για παράδειγμα, για να επικοινωνήσει κάποιος υπολογιστικός κόμβος A με τον υπολογιστικό κόμβο B, θα στείλει ένα μήνυμα μέσω του πρωτοκόλλου TCP στον B. Κατά κάποιον τρόπο όλες οι υπολογιστικές μονάδες εφαρμόζουν το μοντέλο πελάτη-εξυπηρετητή (client-server) εκτελώντας και τις δύο λειτουργίες, δηλαδή όλοι λαμβάνουν και αποστέλλουν μηνύματα.



Σχήμα 2.2 : Ένα Distributed Memory σύστημα όπου η επικοινωνία γίνεται με ανταλλαγή μηνυμάτων

Όπως είπαμε πιο πάνω το μεγαλύτερο πρόβλημα στην επεξεργασία μεγάλων δεδομένων είναι ότι δεν χωράνε στην μνήμη ταυτόχρονα όλα τα δεδομένα. Επίσης, τα συστήματα που έχουν πολλή υπολογιστική ισχύ και τεράστιες ποσότητες μνήμης είναι αρκετά ακριβά (υπέρ-υπολογιστές) γι' αυτό πολλοί μεγάλοι οργανισμοί και οι πλείστες εταιρείες καταφεύγουν σε λύσεις που εφαρμόζονται σε καταναμημένα συστήματα. Το μεγάλο πλεονέκτημα των καταναμημένων συστημάτων είναι ότι δεν είναι απαραίτητο οι υπολογιστικές μονάδες να είναι του ίδιου τύπου ή ακόμα και των ίδιων προδιαγραφών. Συνεπώς, με λιγότερο κόστος μπορούμε να έχουμε ένα υπολογιστικό σύστημα μεγάλης υπολογιστικής ισχύς. Επίσης, πλέον στις μέρες μας υπάρχει πληθώρα υπηρεσιών που προσφέρουν υποδομές καταναμημένων συστημάτων προς ενοικίαση, όπως Amazon Web Services[18] και Google Cloud Platform[19], με αποτέλεσμα να είναι πιο εύκολη η εφαρμογή ενός αλγορίθμου σχεδιασμένου για καταναμημένα συστήματα και η άμεση χρήση του.

Επιπρόσθετα, το πρόβλημα μνήμης όπως αναφέρθηκε μπορεί να επιλυθεί εντελώς με την χρήση αλγορίθμων που χρησιμοποιούν τον δίσκο ως ενδιάμεσο χώρο αποτελεσμάτων. Ως εκ τούτου ο αλγόριθμος που θα παρουσιαστεί σε αυτό το κεφάλαιο είναι προορισμένος για καταναμημένα συστήματα και στοχεύει στην εξάλειψη της αναγκαιότητας για διαθέσιμη μνήμη ανάλογη με το μέγεθος των δεδομένων, κάτι που θα του επιτρέπει να χειρίζεται υπέρογκο πλήθος δεδομένων.

2.4.2 Μοντέλα προγραμματισμού για καταναμημένα συστήματα

Σε αυτή την ενότητα θα μελετήσουμε συνοπτικά τα κυριότερα μοντέλα προγραμματισμού στα καταναμημένα συστήματα. Ειδικότερα μας αφορούν τα μοντέλα που χρησιμοποιούνται κατά κόρων για την επεξεργασία Μεγάλων Δεδομένων (Big Data).

Συγκεκριμένα θα περιγράψουμε τα δύο de-facto μοντέλα των τελευταίων χρόνων, το MPI [20] (Διεπαφή ανταλλαγής μηνυμάτων) και το Hadoop framework (σύστημα βασισμένο στο MapReduce μοντέλο) με το δεύτερο να έχει πιο αναλυτική περιγραφή

λόγω του ότι είναι και το σύστημα στο οποίο βασίζεται ο αλγόριθμος που θα παρουσιαστεί αργότερα.

Πλαίσιο προγραμματισμού MPI

Το ακρώνυμο MPI συμβολίζει την φράση Message Passing Interface, «Διεπαφή ανταλλαγής μηνυμάτων» και προέρχεται από το γεγονός ότι το κύριο μέσο επικοινωνίας στο μοντέλο αυτό είναι μηνύματα διαμέσου του δικτύου. Συγκεκριμένα, γίνεται χρήση του πρωτοκόλλου TCP για την επίτευξη της επικοινωνίας μεταξύ των υπολογιστικών μονάδων.

Προγραμματιστικό Μοντέλο Map Reduce (Google)

Το MapReduce είναι ένα προγραμματιστικό μοντέλο το οποίο είναι ιδανικό για επεξεργασία και δημιουργία τεράστιου όγκου δεδομένων. Έχει επινοηθεί από την Google το 2004 και είναι βασισμένο στο συναρτησιακό είδος προγραμματισμού επιτρέποντας του να είναι εξαιρετικά παράλληλο (massively parallel) και να τρέχει σε συμπλέγματα μεγάλου αριθμού υπολογιστικών κόμβων (cluster of processors / computers).

Pseudocode 2.1: Map Reduce Paradigm – Main functions

```
Map
map (in_key, in_value) -> list(out_key, intermediate_value)

Reduce
reduce (out_key, list(intermediate_value)) -> list(out_value)
```

Η βασική λειτουργία του MapReduce βασίζεται στη χρήση δύο συναρτήσεων. Η πρώτη συνάρτηση είναι η *Map* η οποία παίρνει σαν είσοδο ένα ζευγάρι *Κλειδί – Τιμή* (key-value pair) όπου κάθε τιμή συνήθως είναι μια γραμμή από τα δεδομένα εισόδου και μετά από επεξεργασία παράγει ένα ή και περισσότερα ενδιάμεσα ζευγάρια τύπου *Κλειδί – Τιμή*. Τα ενδιάμεσα αυτά αποτελέσματα τροφοδοτούνται στην δεύτερη συνάρτηση, *Reduce*, η οποία παραλαμβάνει όλα τα ενδιάμεσα ζευγάρια που παράχθηκαν με το ίδιο

κλειδί και μετά από επεξεργασία παράγει τα τελικά αποτελέσματα συνδυάζοντας τις διάφορες τιμές κάθε κλειδιού.

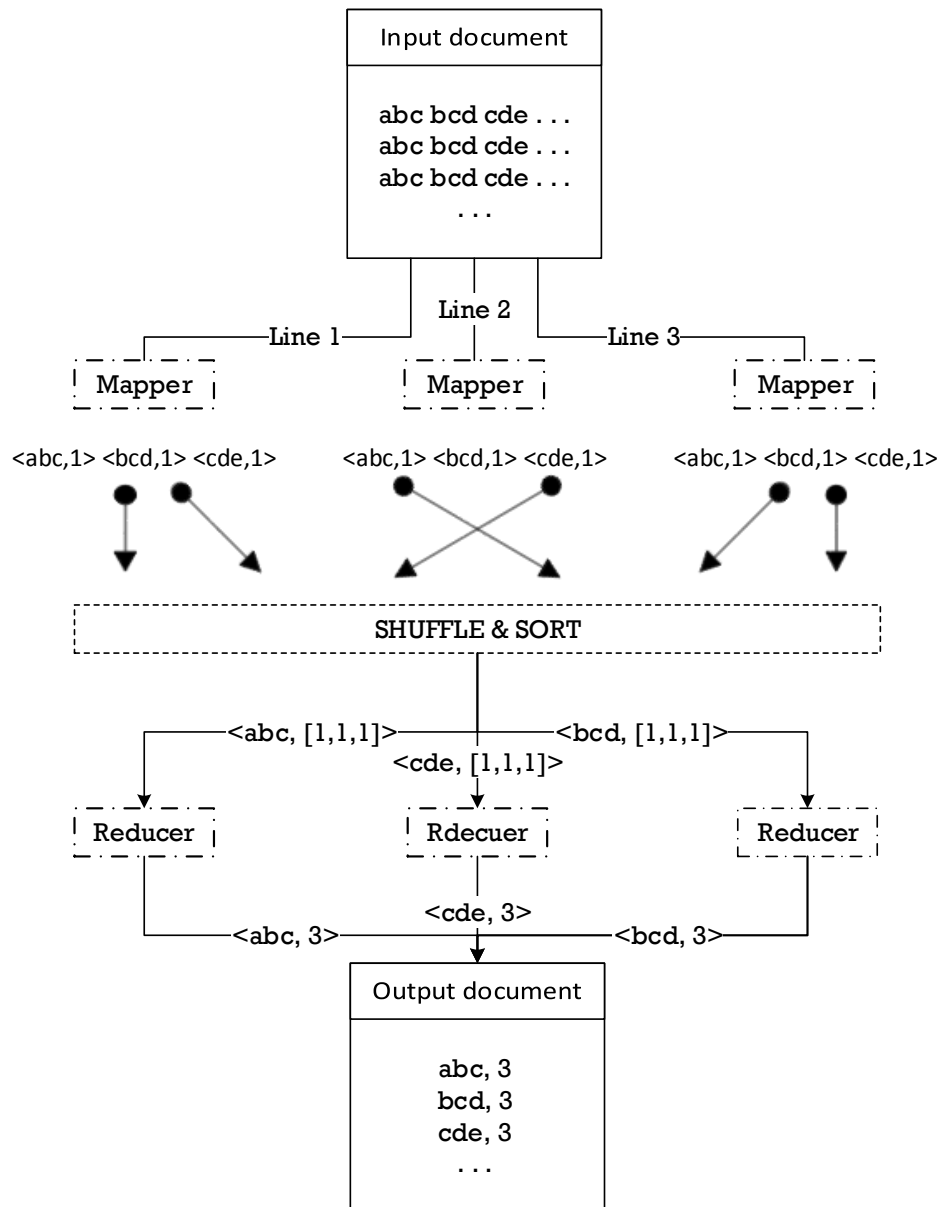
Algorithm 2.1: Όπως περιγράφεται σε παράδειγμα στις διαφάνειες του Map Reduce από την Google[21]

```
1:  map(String input_key, String input_value):
2:      // input_key: document name
3:      // input_value: document contents
4:      for each word w in input_value:
5:          EmitIntermediate(w, '1');

1:  reduce(String output_key, Iterator intermediate_values):
2:      // output_key: a word
3:      // output_values: a list of counts
4:      int result = 0;
5:      for each v in intermediate_values:
6:          result += ParseInt(v);
7:      Emit(AsString(result));
```

Ένα απλό παράδειγμα είναι αυτό όπου μετράμε τις εμφανίσεις κάθε λέξης στην είσοδο μας όπως περιγράφεται στον *Algorithm 2.1* και όπως φαίνεται στο *σχήμα 2.3*. Η συνάρτηση *Map* παίρνει Κλειδί το κείμενο και Τιμή κάθε φορά είναι μια γραμμή κειμένου. Ακολούθως για κάθε λέξη της εισόδου παράγει το ζευγάρι <Λέξη, 1>. Ακολούθως η συνάρτηση *Reduce* θα πάρει ως κλειδί την κάθε λέξη και ως *intermediate_values* (λίστα – δεύτερη παράμετρος στον *Algorithm 2.1*) όλες τις ενδιάμεσες τιμές των συναρτήσεων *Map*, δηλαδή για κάθε φορά που εμφανίστηκε η λέξη πρέπει να υπάρχει και ένας άσος στα *intermediate_values*.

Συνεπώς στο *Reduce* απλά μετράμε πόσες εμφανίσεις έχει η κάθε λέξη και παράγουμε το ζευγάρι <Λέξη, Συνολικές Εμφανίσεις>.



Σχήμα 2.3 : Ροή εκτέλεσης του παραδείγματος για καταμέτρηση των εμφανίσεων κάθε λέξης σε ένα κείμενο.

Πρέπει να σημειώσουμε ότι για κάθε γραμμή στο κείμενο θα γίνει ένα κάλεσμα της συνάρτησης *Map*, χωρίς όμως να σημαίνει ότι κάθε κάλεσμα της θα εκτελεστεί από διαφορετικό *Mapper*. Συνήθως οι *Mapper* είναι πολύ λιγότεροι από τα καλέσματα της *Map* και είναι πιο κοντά στον αριθμό των κόμβων που έχουμε (κάποτε ακριβώς ίδιος). Τα προγράμματα που γράφονται με αυτό το τύπο είναι αυτόματα εύκολο να παραλληλοποιηθούν και να εκτελεστούν σε συγκροτήματα χιλιάδων υπολογιστών κοινού τύπου (commodity PCs). Το μοντέλο αυτό επιτρέπει στους προγραμματιστές να γράφουν προγράμματα έτοιμα για εκτέλεση σε κατακευματισμένα συστήματα χωρίς να έχουν τεχνική γνώση για τις λεπτομέρειες κατακευματισμένων συστημάτων, κάτι πολύ

σημαντικό και ένας από τους παράγοντες που είχαν ως αποτέλεσμα την εδραίωση του μοντέλου αυτού.

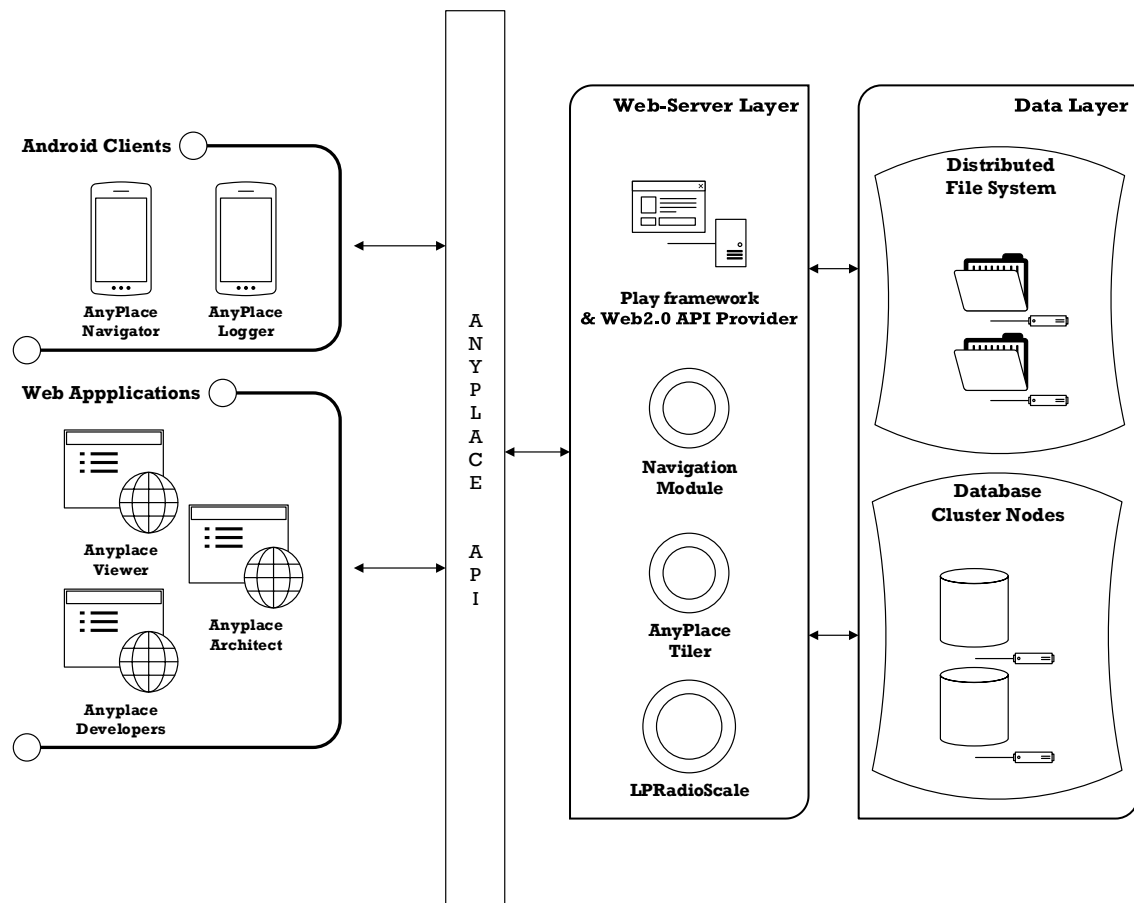
Το MapReduce έχει εξαπλωθεί με γοργούς ρυθμούς από την μέρα δημοσίευσής του και χρησιμοποιείται κατά κόρον σε πολλά προγράμματα της Google με πιο γνωστό τον αλγόριθμο PageRank ο οποίος χρησιμοποιείται για τη ταξινόμηση των αποτελεσμάτων αναζήτησης. Άλλα παραδείγματα χρήσης του MapReduce είναι Distributed Grep (κατανεμημένη υλοποίηση της εντολής grep) και Reverse Web-Link Graph (εύρεση όλων των σελίδων που δείχνουν σε ένα υπερσύνδεσμο).

Κεφάλαιο 3

Αρχιτεκτονική του AnyPlace

3.1	Υποδομή Ραχοκοκαλιάς	34
3.2	Εφαρμογές στον Παγκόσμιο Ιστό	34
3.3	Εφαρμογές για έξυπνα κινητά Android	35
3.3.1	AnyPlace Logger	35
3.3.2	AnyPlace Navigator	35
3.4	Ροή πληροφορίας στο AnyPlace & Περιπτώσεις Χρήσης	36
3.4.1	Χαρτογράφηση κτιρίου μέσω του AnyPlace Architect	36
3.4.2	Συλλογή RSS αποτυπωμάτων μέσω του AnyPlace Logger	37
3.4.3	Χρήση εφαρμογής AnyPlace Navigator & AnyPlace Viewer	37

Σε αυτή την ενότητα θα μελετήσουμε την αρχιτεκτονική του συστήματος ως σύνολο, όπως φαίνεται στο *Σχήμα 3.1* και θα την περιγράψουμε επιγραμματικά. Ακολούθως θα γίνει μια σύντομη περιγραφή της χρήσης του συστήματος και πως ρέει η πληροφορία μέσα του από τη στιγμή που εισέρχεται μέχρι τη στιγμή που προβάλλεται στον χρήστη.



Σχήμα 3.1 : Η αρχιτεκτονική του συστήματος AnyPlace

Στο σχήμα 3.1 βλέπουμε την όλη αρχιτεκτονική του συστήματος AnyPlace και όλα τα επιμέρους κομμάτια που την αποτελούν. Ανάλυση της κάθε μονάδας θα γίνει ξεχωριστά σε κάθε υπό-ενότητα αυτού του κεφαλαίου.

Όπως φαίνεται στο σχήμα έχουμε χρήστες κινητών συσκευών Android μέσω των εφαρμογών Navigator και Logger, και χρήστες των διαδικτυακών εφαρμογών Viewer, Architect και Developers. Όλοι οι χρήστες επικοινωνούν με το σύστημα χρησιμοποιώντας το API (διεπαφή προγραμματισμού εφαρμογής) που παρέχεται. Το API παρέχεται από τους εξυπηρετητές που φαίνονται στο Web-Server Layer και τα δεδομένα αποθηκεύονται ανάλογα του είδους είτε στην κατανεμημένη βάση δεδομένων είτε στο κατανεμημένο σύστημα αρχείων, Data Layer.

3.1 Υποδομή Ραχοκοκαλιάς

Η ραχοκοκαλιά του AnyPlace συνδυάζει πολλές τεχνολογίες και πλαίσια προγραμματισμού (programming frameworks) ώστε να υποστηρίζει τις παρεχόμενες υπηρεσίες και ταυτόχρονα να εγγυάται επεκτασιμότητα και επίδοση.

Ένας HA-Proxy διαμοιραστής φόρτου εργασίας (load-balancer) είναι υπεύθυνος για να διαχειρίζεται τον φόρτο εργασίας από τις δύο κύριες διαδικτυακές εφαρμογές, AnyPlace Viewer & AnyPlace Architect. Επιπλέον, ένα συγκρότημα κόμβων τρέχει το πλαίσιο προγραμματισμού Play! [22] το οποίο αναλαμβάνει την εκτέλεση όλων των αιτημάτων πάνω στο AnyPlace API. Τέλος, ακόμα ένα συγκρότημα κόμβων παρέχει ένα επεκτάσιμο επίπεδο όπου λειτουργεί το σύστημα διαχείρισης αρχείων και οι βάση δεδομένων.

Τα δεδομένα που ρέουν από το σύστημα μπορούν να χωριστούν σε δύο κατηγορίες:

1. Δεδομένα που έχουν καθορισμένη δομή και μοντελοποιούνται ως JSON [23] κείμενα (RSS fingerprints, Point of Interest information, κλπ.)
2. Δεδομένα δυαδικής μορφής, binary, που δεν μπορούν να αναπαρασταθούν ως κείμενο (αρχιτεκτονικά σχέδια ορόφων, κλπ.)

Τα δεδομένα της 1^{ης} κατηγορίας διαχειρίζονται από ένα συγκρότημα κόμβων που τρέχει την βάση δεδομένων Couchbase Server 2.x [24] ενώ της 2^{ης} από άλλο συγκρότημα που τρέχει το κατακευκτικό σύστημα διαχείρισης αρχείων GlusterFS [25], και τα δύο θα εξηγηθούν περαιτέρω στο κεφάλαιο 6.

Το παρόν πρωτότυπο σύστημα του AnyPlace φέρει 16TB από RAID-5 / SSD αποθηκευτικό χώρο σε ένα εξυπηρετητή IBM X3550 και μπορεί να χρησιμοποιήσει μέχρι και 320GB κύριας μνήμης. Κάθε εικονικός υπολογιστής του συστήματος λειτουργεί σύστημα CentOS 6.4 x64-bits.

3.2 Εφαρμογές στον Παγκόσμιο Ιστό

Το AnyPlace παρέχει δύο εφαρμογές διαδικτύου, τον Architect και τον Viewer. Ο πρώτος παρέχει απλή διεπαφή μέσω της οποίας γίνεται η χαρτογράφηση των κτιρίων

πάνω από το σύστημα παροχής χαρτών Google Maps και ο καθορισμός των σχεδίων για κάθε όροφο. Ο μεν Viewer λειτουργεί σαν περιηγητής μέσα στα χαρτογραφημένα κτίρια του συστήματος. Επιτρέπει στον χρήστη να αναζητήσει και να δει λεπτομέρειες για κτίρια ή σημεία ενδιαφέροντος εντός του κτιρίου και επίσης παρέχει οδηγίες πλοήγησης ανάμεσα τους. Ο Viewer αναπτύχθηκε με την χρήση των πλαισίων ανάπτυξης εφαρμογών Dart [26] και AngularDart [27] από την Google ενώ ο Architect με τη χρήση HTML5 και του πλαισίου εφαρμογών AngularJS [28] που υποστηρίζεται και πάλι από την Google.

3.3 Εφαρμογές για έξυπνα κινητά Android

Σε αυτή την ενότητα θα δούμε συνοπτικά τις δύο εφαρμογές που παρέχει το AnyPlace για έξυπνες κινητές συσκευές Android οι οποίες υποστηρίζουν όλες τις εκδόσεις Android 2.3.3+ χωρίς την απαίτηση δικαιωμάτων διαχειριστή (root access), ενώ αναλυτική περιγραφή τους θα γίνει στο κεφάλαιο 5.

3.3.1 AnyPlace Logger

Η εφαρμογή AnyPlace Logger είναι υπεύθυνη για την συλλογή των RSS αποτυπωμάτων βάση των κοντινών WLAN σημείων πρόσβασης και η καταγραφή τους στην δευτερεύουσα μνήμη της συσκευής. Ακολουθώντας, η αποστολή τους στην υπηρεσία AnyPlace ώστε να αποθηκευτούν στο σύστημα και να ενημερωθεί ο ραδιοχάρτης του κτιρίου.

3.3.2 AnyPlace Navigator

Η εφαρμογή AnyPlace Navigator είναι μια αυτόνομη εφαρμογή που επιτρέπει την γεωτοποθέτηση και πλοήγηση σε εσωτερικούς χώρους για συσκευές Android δίχως να έχει γίνει κάποια ρύθμιση προηγουμένως ή χρήση επιπλέον εξαρτημάτων. Εμφανίζει ένα απλό γεωγραφικό χάρτη μέσω μιας απλοϊκής διεπαφής και επιτρέπει στον χρήστη να τοποθετηθεί στον χώρο καθώς κινείται σε πραγματικό χρόνο καθώς και να

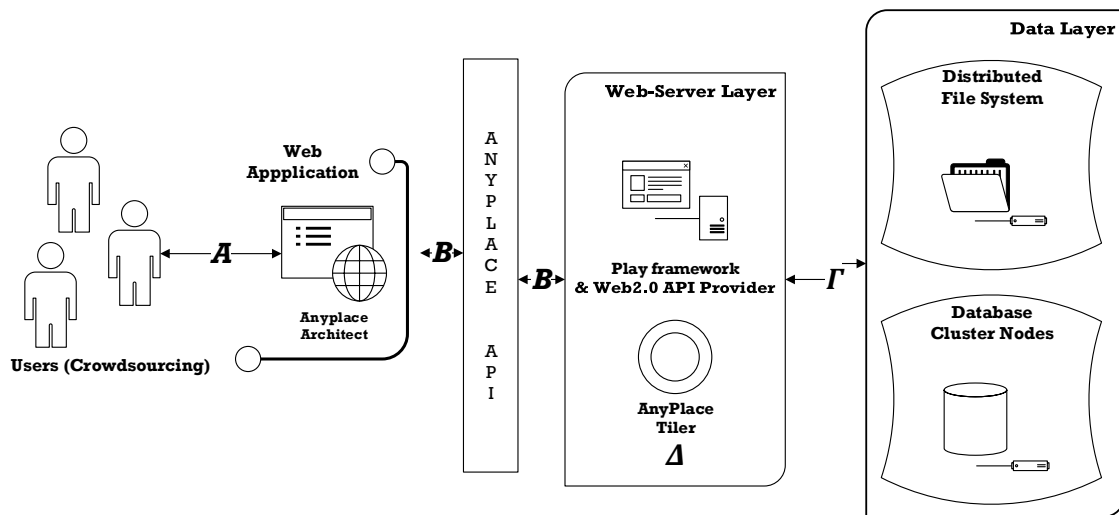
πλοηγηθεί στα διάφορα σημεία ενδιαφέροντος που υπάρχουν στο κτίριο βάση της εκάστοτε τοποθεσίας του.

3.4 Ροή πληροφορίας στο AnyPlace & Περιπτώσεις Χρήσης

Σε αυτή την υπό-ενότητα θα μελετηθούν οι διάφορες περιπτώσεις χρήσης (use-cases) του συστήματος και πως αυτή η πληροφορία ρέει μέσα από τα διάφορα επιμέρους συστατικά του.

3.4.1 Χαρτογράφηση κτιρίου μέσω του AnyPlace Architect

Εδώ θα δούμε ποιες μονάδες λαμβάνουν μέρος στην χαρτογράφηση ενός κτιρίου.

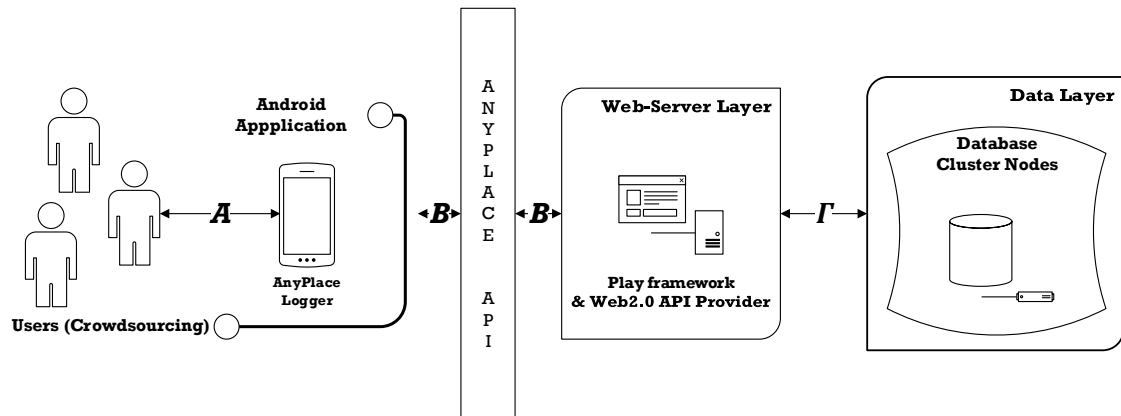


Σχήμα 3.2 : Ροή πληροφορίας και εμπλεκόμενα συστατικά κατά την χαρτογράφηση ενός κτιρίου

Αρχικά, μέσω του AnyPlace Architect ο χρήστης συνδέεται στην υπηρεσία AnyPlace (σημείο *A*). Ακολούθως, δημιουργεί το κτίριο, τους ορόφους και σημειώνει σημεία ενδιαφέροντος τα οποία αν θέλει μπορεί να συνδέσει ώστε να γίνεται δυνατή η πλοήγηση. Όλες αυτές οι πληροφορίες αποστέλλονται στο AnyPlace API (σημείο *B*) και αποθηκεύονται ανάλογα είτε στο σύστημα διαχείρισης αρχείων είτε στην βάση δεδομένων όπως περιεγράφηκαν στο 3.1 (σημείο *Γ*). Επίσης, με το που αποστέλλεται το αρχιτεκτονικό σχέδιο ορόφου τυγχάνει επεξεργασίας από το συστατικό AnyPlace Tiler (σημείο *Δ*).

Στο σημείο αυτό είναι εμφανής η προσέγγιση του συστήματος μας στην υποστήριξη πληθωρισμού καθώς το δύσκολο έργο της χαρτογράφησης κτιρίων πλέον διαχέεται στον κάθε εγκεκριμένο χρήστη.

3.4.2 Συλλογή RSS αποτυπωμάτων μέσω του AnyPlace Logger



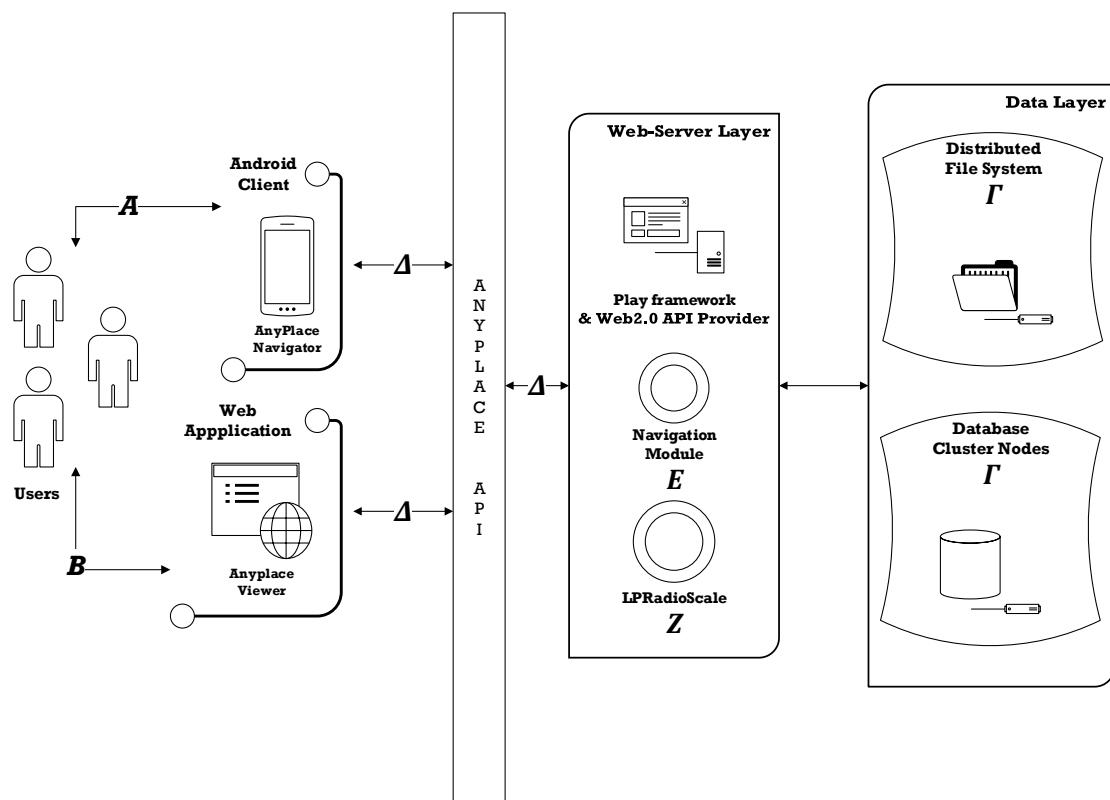
Σχήμα 3.3 : Ροή πληροφορίας και εμπλεκόμενα συστατικά κατά τη συλλογή και αποθήκευση των WLAN RSS fingerprints

Εκτός από τα δεδομένα χαρτογράφησης το σύστημα AnyPlace χειρίζεται και αποτυπώματα από γειτονικά WLAN σημεία πρόσβασης. Λόγω του ότι αποτελεί μια χρονοβόρα και φορτική εργασία και πάλι εφαρμόζεται πληθωριστικό μοντέλο για την συλλογή των αποτυπωμάτων μέσω του Logger (σημείο A). Αρχικά ο Logger, συλλέγει τις ενδείξεις από τα APs και ακολούθως τα αποστέλλει στην υπηρεσία μέσω του API (σημείο B) για αποθήκευση και επεξεργασία όταν χρειαστεί να δημιουργηθεί ο ραδιοχάρτης. Τα αποτυπώματα RSS μοντελοποιούνται ως JSON documents και αποθηκεύονται στο κατακευκτικό σύστημα βάσης δεδομένων (σημείο Γ).

3.4.3 Χρήση εφαρμογής AnyPlace Navigator & AnyPlace Viewer

Ο Navigator (σημείο A) και ο Viewer (σημείο B) είναι οι δύο κύριες εφαρμογές που τυγχάνουν χρήσης από τον απλό χρήστη του συστήματος. Βασικά είναι αυτές που αντλούν όλες τις πληροφορίες του συστήματος (σημείο Γ) μέσω του AnyPlace API (σημείο Δ) και τις προβάλλουν στους χρήστες. Η μόνη διαφορά τους είναι ότι ο Viewer

δεν μπορεί να παρέχει στον χρήστη τοποθέτηση πραγματικού χρόνου εκτός από την εκτίμηση θέσης του περιηγητή του.



Σχήμα 3.4 : Ροή πληροφορίας και εμπλεκόμενα συστατικά κατά τη χρήση του Navigator (Android) και του Viewer (Web)

Ο χρήστης μπορεί να αναζητήσει κτίρια ή σημεία ενδιαφέροντος εντός του κτιρίου και να πλοηγηθεί (σημείο *E*) σε αυτά είτε βάση της εκάστοτε θέσης του (Navigator) είτε βάση της θέσης του περιηγητή του ή της επιλογής του (Viewer). Στον Navigator, εάν επιλέξει ο χρήστης την ανίχνευση της θέσης του σε πραγματικό χρόνο τότε στην ροή εκτέλεσης θα εμπλακεί και το συστατικό του *LPRadioScale* (σημείο *Z*) το οποίο θα δημιουργήσει εάν χρειάζεται ενημέρωση τον ραδιοχάρτη που ζητήθηκε.

Κεφάλαιο 4

Δημιουργία Ραδιοχάρτη σε Μεγάλα Δεδομένα – *LPRadioScale*

4.1	Θεωρητικό υπόβαθρο	39
4.2	Περιγραφή προβλήματος	44
4.3	Μοντέλο Συστήματος	46
4.4	Περιγραφή προτεινόμενου αλγόριθμου – <i>LPRadioScale</i>	48
4.5	Λεπτομερής Ανάλυση αλγορίθμου <i>LPRadioScale</i>	49
	4.5.1 Συλλογή στατιστικών στοιχείων για κάθε σημείο πρόσβασης	49
	4.5.2 Φιλτράρισμα των σημείων πρόσβασης	52
	4.5.3 Δημιουργία Ραδιοχάρτη	52

4.1 Θεωρητικό υπόβαθρο

Σε αυτή την ενότητα θα αναφερθούμε σε κάποιες βασικές έννοιες και θα παρουσιάσουμε το βασικό υπόβαθρο που χρειάζεται για την κατανόηση της τεχνικής μας, *LPRadioScale*.

4.1.1 Hadoop Framework – Apache Foundation

Το Hadoop είναι ένα πλαίσιο προγραμματισμού που διατηρείται από την Apache και είναι δημοσιευμένο κάτω από άδεια ανοικτού κώδικα. Επιτρέπει την κατανομή

επεξεργασία τεράστιων δεδομένων πάνω από συγκροτήματα κοινών υπολογιστικών μονάδων. Είναι σχεδιασμένο ώστε να επεκτείνεται από ένα μηχάνημα σε χιλιάδες κόμβους με πολύ μεγάλη ανεκτικότητα σε σφάλματα. Επιπλέον, αντί να στηρίζεται σε μεγάλης ισχύς μηχανήματα για να πετύχει αυτή την ανεκτικότητα και ελαστικότητα, στηρίζεται στην ικανότητα του λογισμικού να εντοπίσει και να χειριστεί αποτυχίες σε επίπεδο εφαρμογής.

Αποτελεί υλοποίηση του μοντέλου MapReduce της Google και του GFS, Google File System, και πάλι από την Google τα οποία αποτελούν και τα δύο κύρια του συστατικά:

1. HDFS

Το Hadoop Distributed File System αποτελεί την ανοικτού κώδικα υλοποίηση του GFS – Google File System [29] και πάνω του στηρίζεται όλο το Hadoop framework. Είναι ένα καταναμημένο σύστημα διαχείρισης αρχείων το οποίο συγκεντρώνει όλο τον αποθηκευτικό χώρο ενός συγκροτήματος υπολογιστικών μονάδων και τον ενοποιεί ώστε να συμπεριφέρεται σαν ένα τεράστιο σύστημα αρχείων (big file system). Τα αρχεία είναι αποθηκευμένα σε μικρά τεμάχια (chunks) συνήθως μεγέθους 64MB. Η ανεκτικότητα σε σφάλματα στο HDFS επιτυγχάνεται με την αντιγραφή των δεδομένων σε πολλαπλούς κόμβους, κάτι που γίνεται πιο εύκολα και λόγω των file chunks.

2. MapReduce Engine

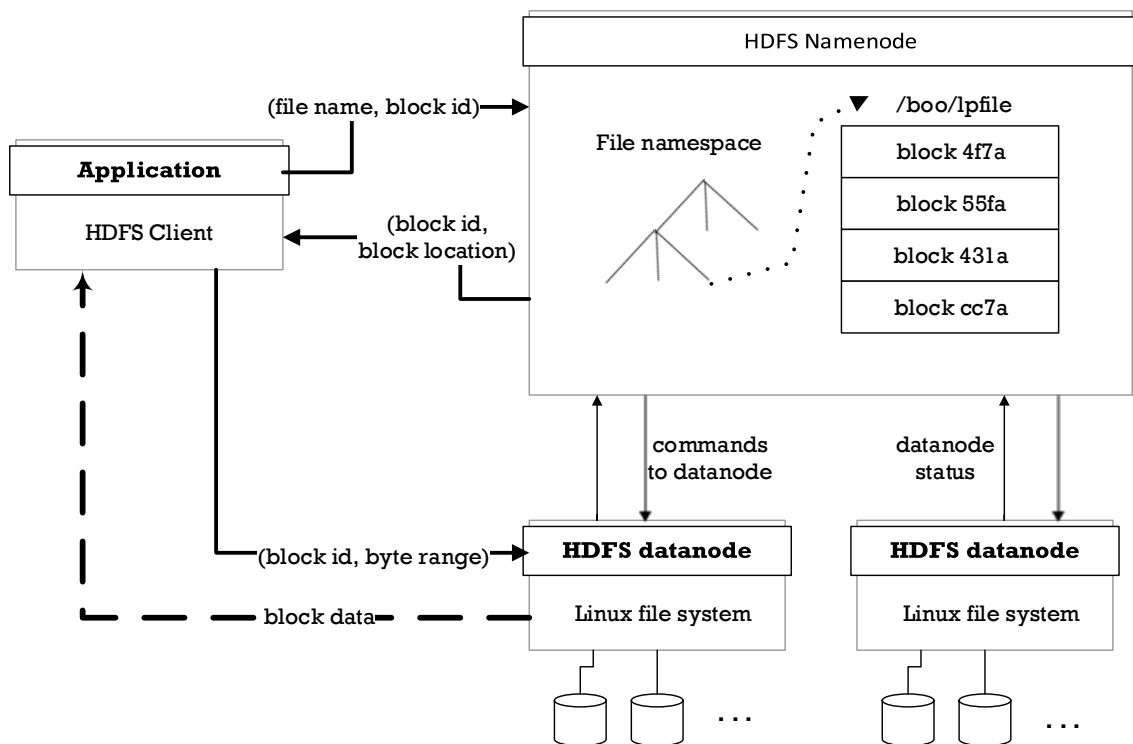
Το MapReduce engine του Hadoop είναι η υλοποίηση του MapReduce μοντέλου και η ενσωμάτωση του με το HDFS. Ο συνδυασμός των δύο επιτρέπει γρήγορη επεξεργασία λόγω του ότι η επεξεργασία των δεδομένων γίνεται τοπικά σε κάθε κόμβο επεξεργασίας, κάτι που έπεται από την ύπαρξη των δεδομένων σε περισσότερους από ένα κόμβους. Επίσης, κληρονομούνται όλες οι ιδιότητες του μοντέλου όπως την επεκτασιμότητα και την ευκολία ανάπτυξης προγραμμάτων.

Η αρχιτεκτονική του HDFS είναι παρόμοια με το μοντέλο master-slave. Συγκεκριμένα, υπάρχει ένα κεντρικός κόμβος που ονομάζεται Namenode ο οποίος είναι υπεύθυνος ώστε να διατηρεί μόνο μετά-πληροφορίες σχετικά με το που βρίσκεται κάθε κομμάτι

(file-chunk) οποιουδήποτε αρχείου. Εκτός από το Namenode υπάρχουν και τα Datanodes τα οποία είναι υπεύθυνα για την αποθήκευση των κομματιών-αρχείων. Σχετική απεικόνιση του HDFS παρουσιάζεται στο *σχήμα 4.1*.

Η διαδικασία με την οποία γίνεται η ανάληψη ενός αρχείου A είναι η εξής:

- Θα γίνει αίτημα στο Namenode για το αρχείο A
- Θα επιστραφούν οι πληροφορίες σχετικά με τους κόμβους που φυλάσσουν τα κομμάτια του αρχείου (chunks)
- Θα γίνει ένα αίτημα για κάθε κομμάτι αρχείου στον αντίστοιχο Datanode και θα επιστραφεί το αρχείο



Σχήμα 4.1 : Αρχιτεκτονική του Hadoop File System - HDFS

Παρακάτω δίνετε μια απλή περιγραφή ενός παραδείγματος εκτέλεσης μιας δουλειάς MapReduce.

Η είσοδος μας είναι ένα αρχείο όπου κάθε γραμμή είναι της μορφής:

UserId, movie1, movie2, . . . , movieN

όπου κάθε γραμμή αντιπροσωπεύει τις αγαπημένες ταινίες ενός χρήστη. Εμείς θέλουμε να μετρήσουμε για κάθε ταινία τους χρήστες που την σημείωσαν ως αγαπημένη.

Algorithm 4.1: Movie Counters - Map function

Input: String “user, movie1, movie2, ... , movieN”

Output: <movieK, 1>

```
1: Map( String line, Context ctx ) {
2:     String[] segments = line.split(',');
3:     For( int i=1; i<segments.length; i++ ){
4:         String movie = segments[i];
5:         ctx.write( movie, 1 );
6:     }
7: }
```

Η διαδικασία εκτέλεσης έχει ως εξής:

1. Το Hadoop διαβάζει από το HDFS το αρχείο και καλεί για κάθε γραμμή την συνάρτηση *Map* που θα δηλώσουμε στο πρόγραμμα μας. Εδώ η εκτέλεση γίνεται ανεξάρτητα και παράλληλα σε κάθε κόμβο.
2. Συνάρτηση *Map* όπως φαίνεται στο *Algorithm 4.1*
3. Τα ενδιάμεσα αποτελέσματα της πρώτης φάσης είναι ένα ζευγάρι <Ταινία, 1> για κάθε αγαπημένη ταινία κάθε χρήστη.
4. Ακολούθως, γίνεται μια ενδιάμεση φάση από το Hadoop όπου τα ενδιάμεσα αποτελέσματα ομαδοποιούνται και ταξινομούνται ώστε ζευγάρια με ίδιο κλειδί να φτάσουν στο ίδιο *Reduce* κάλεσμα μαζί.
5. Η συνάρτηση *Reduce* το μόνο που χρειάζεται να κάνει είναι να μετρήσει τα ζευγάρια που έχει για την κάθε ταινία, *Algorithm 4.2*
6. Άρα τα τελικά αποτελέσματα είναι για κάθε ταινία ένα ζευγάρι <Ταινία, Εμφανίσεις >

Algorithm 4.2: Movie Counters - Reduce function

Input: <movieK, List<Int> Counters>

Output: <movieK, TotalCounter>

```
1: Reduce( String movie, Iterator values, Context ctx ) {
2:     int count = 0;
3:     for( int value : values ){
4:         Count += value;
5:     }
6:     ctx.write( movie, count );
7: }
```

Να σημειώσουμε ότι οι υπολογιστικοί κόμβοι που θα χρησιμοποιηθούν είναι πολύ λιγότεροι από τους *Mapper* (υπεύθυνοι για να καλέσουν την συνάρτηση *Map*) και τους *Reducer* (υπεύθυνοι για να καλέσουν την συνάρτηση *Reduce*). Κάθε κόμβος θα καλέσει πολλαπλές φορές την κάθε συνάρτηση ανάλογα με την εξέλιξη της εκτέλεσης. Με αυτό τον τρόπο επιτυγχάνεται καλός διαμοιρασμός φόρτου εργασίας στους υπολογιστικούς κόμβους αφού μπορεί κάποιος κόμβος μπορεί να καθυστερήσει σε κάποια από τις εκτελέσεις του, έτσι για αποφυγή αύξηση του χρόνου εκτέλεσης, άλλοι κόμβοι θα εκτελέσουν περισσότερα καλέσματα των συναρτήσεων μέχρι να τελειώσουν όλα τα δεδομένα.

Σύνοψη για το σύστημα Hadoop [30]–[32]

Πως αποθηκεύονται τα δεδομένα: Τα διάφορα δεδομένα, αρχεία, αποθηκεύονται διάσπαρτα στους διάφορους υπολογιστικούς κόμβους που συντελούν το σύστημα με αντιγραφή σε πολλαπλούς κόμβους. Ως αποτέλεσμα πολύ μεγάλα αρχεία πλέον μπορούν να τύχουν επεξεργασίας εύκολα. Συχνά χρησιμοποιείται η φράση ότι το Hadoop βασίζεται στο απλό μοντέλο δεδομένων « *any data will fit* », δηλαδή όπως και να είναι τα δεδομένα, το Hadoop μπορεί να τα διαχειριστεί και να τα επεξεργαστεί.

Πως γίνεται η επεξεργασία των δεδομένων: Με τη χρήση του MapReduce η επεξεργασία γίνεται πολύ πιο εύκολη και πιο γρήγορη αφού αντίθετα με τις κοινές τεχνικές όπου γίνεται μεταφορά δεδομένων στους κόμβους για να τύχουν επεξεργασίας, εδώ γίνεται μεταφορά της επεξεργασίας στους κόμβους που ήδη έχουν τα δεδομένα τοπικά αποφεύγοντας μεγάλη διακίνηση δεδομένων μέσω του δικτύου. Αυτό γίνεται με τη σωστή ανάθεση των Map και Reduce συναρτήσεων στους υπολογιστικούς κόμβους.

Χαρακτηριστικά:

- **Επεκτάσιμο:** Υπολογιστικοί κόμβοι μπορούν να προστεθούν χωρίς να αλλάξει η υπάρχουσα υποδομή, η μορφή των δεδομένων εισόδου, η φόρτωση των δεδομένων και η ανάπτυξη προγραμματιστικών εργασιών.
- **Πρακτική οικονομική λύση:** Το Hadoop επιτρέπει τεράστια παράλληλη επεξεργασία σε κοινούς υπολογιστές με αποτέλεσμα την μείωση κόστους ανά

GB αποθηκευτικού χώρου, το οποίο με τη σειρά του επιτρέπει την μοντελοποίηση των δεδομένων όπως θέλουμε.

- **Ελαστικό:** Λόγω της έλλειψης ενός συγκεκριμένου σχήματος δεδομένων μπορεί να χρησιμοποιηθεί σε πολλές περιπτώσεις αντλώντας δεδομένα από πολλαπλές πηγές επιτρέποντας την συνένωση τους και ταυτόχρονη επεξεργασία τους.
- **Ανεκτικό σε σφάλματα:** Μεγάλη σημασία δίνεται στην παροχή ανεκτικότητας στα σφάλματα ώστε η λειτουργία των προγραμμάτων να συνεχίζει χωρίς να διακόπτεται και κυρίως χωρίς επανεκκίνηση της. Αυτό επιτυγχάνεται με το να εκτελεί την εργασία του εσφαλμένου κόμβου σε ένα άλλο σωστό κόμβο που διαθέτει τα απαιτούμενα δεδομένα.

4.2 Περιγραφή προβλήματος

Σε αυτή την ενότητα θα περιγράψουμε το πρόβλημα που θέλουμε να λύσουμε και ακολούθως, στην επόμενη ενότητα θα εξηγήσουμε πως η τεχνική μας LPRadioScale προσπαθεί να το αντιμετωπίσει.

Η διαδικασία δημιουργίας ραδιοχάρτη από αποτυπώματα RSS δεν είναι δύσκολη και μέχρι πρόσφατα δεν υπήρχε βιβλιογραφία για τον τρόπο που διεξάγεται, αφού οι κοινές απλές προσεγγίσεις δούλευαν μια χαρά. Μέχρι τώρα σε καμιά βιβλιογραφία δεν χρησιμοποιήθηκαν τεράστια μεγέθη δεδομένων και η απλή σειριακή προσέγγιση της δημιουργίας του ραδιοχάρτη χρειαζόταν μερικά δευτερόλεπτα.

Πλέον, όπως αναφέρετε και σε προηγούμενα σημεία αυτής της διπλωματικής εργασίας έχουμε μπει στην εποχή των Μεγάλων Δεδομένων, όπου το μέγεθος των δεδομένων δεν μας επιτρέπει να χρησιμοποιήσουμε τις μέχρι τώρα λύσεις (είτε σειριακής εκτέλεσης είτε παράλληλης), κυρίως λόγω λιγοστής μνήμης.

Το πρόβλημα φαίνεται πιο έντονο μέσα από το παρακάτω απλό παράδειγμα. Έστω ότι συλλέξαμε αποτυπώματα RSS από ένα μεγάλο κτίριο, τύπου κρατικής έκθεσης, και έχουμε συνολικά $1 \cdot 10^{10}$ αποτυπώματα και 1000 διαφορετικά σημεία πρόσβασης ασύρματης δικτύωσης. Η ελάχιστη μνήμη που απαιτείται για την δημιουργία του ραδιοχάρτη, δεδομένου ότι δεν έχουμε τα δεδομένα μας σε κάποια συγκεκριμένη μορφή ή σειρά προσέλευσης, είναι περίπου ίση με $1 \cdot 10^{13} \cdot \text{sizeof}(\text{short int}) \approx 40 \text{ TB}$.

Ακόμα και μεγάλης δυναμικότητας εξυπηρετητές δεν μπορούν να χειριστούν τόσα πολλά δεδομένα ταυτόχρονα άρα πολλές υφιστάμενες λύσεις δεν είναι εφαρμόσιμες. Το πρόβλημα αυτό καθίσταται ακόμα πιο έντονο όταν θελήσουμε να επεξεργαστούμε αυτά τα αποτυπώματα με στόχο το φιλτράρισμα τους ή την στατιστική ανάλυση τους.

Ως εκ τούτου, τα πλείστα προβλήματα που αφορούν μεγάλα δεδομένα βρίσκουν λύσεις σχεδιασμένες για εκτέλεση σε κατανεμημένα συστήματα τα οποία χρησιμοποιούν πολλές υπολογιστικές μονάδες ώστε να έχουν περισσότερη συνολική μνήμη και αυξημένη υπολογιστική ισχύ. Σε πολλές περιπτώσεις όμως το πρόβλημα της μνήμης συνεχίζει να υφίσταται με αποτέλεσμα να οδηγούμαστε σε χρήση αλγορίθμων που χρησιμοποιούν το δίσκο για ενδιάμεσα ή για προσωρινά αποτελέσματα τα οποία θα χρησιμοποιηθούν σε μεταγενέστερο στάδιο του αλγορίθμου.

Στη συνέχεια θα παρουσιάσουμε την τεχνική *LPRadioScale* η οποία επιλύει το πρόβλημα της δημιουργίας ραδιοχάρτη με Μεγάλα Δεδομένα. Ο αλγόριθμος μας τρέχει σε κατανεμημένο σύστημα Hadoop και δεν εξαρτάται καθόλου από την διαθέσιμη μνήμη των υπολογιστικών κόμβων σε σχέση με το μέγεθος των δεδομένων αφού δεν διατηρείται οποιαδήποτε δομή στην μνήμη κατά τη διάρκεια της εκτέλεσης, παρά μόνο δομές που διατηρούν το σύνολο όλων των έγκυρων σημείων πρόσβασης, που είναι απίθανο να ξεπερνούν την διαθέσιμη μνήμη των υπολογιστικών μας κόμβων. Ο λόγος που προτιμήθηκε το Hadoop σε σχέση με άλλα frameworks είναι ότι το πρόβλημα μας δεν είναι πρόβλημα που χρειάζεται επίλυση σε πραγματικό χρόνο και μπορεί να γίνει offline πριν γίνει χρήση του ραδιοχάρτη, άρα μοιάζει αρκετά με στατιστικές εφαρμογές που επεξεργάζονται τεράστια δεδομένα στο παρασκήνιο και δεν χρειάζεται απαραίτητα

να είναι η πιο γρήγορη προσέγγιση. Στην κατηγορία αυτή είναι που διαπρέπει το Hadoop και γενικά τα frameworks Map Reduce. Είναι βασικά ένας συμβιβασμός μεταξύ χρόνου εκτέλεσης και υποστηριζόμενο μέγεθος δεδομένων.

Επίσης, κατά τη διάρκεια της δημιουργίας του ραδιοχάρτη εφαρμόζουμε κάποια φίλτρα τα οποία θα συμβάλουν στη μείωση των ενεργών σημείων πρόσβασης που θα παραμείνουν στον τελικό ραδιοχάρτη με στόχο την μείωση του μεγέθους του. Μειωμένο μέγεθος του ραδιοχάρτη σημαίνει ότι όλοι οι αλγόριθμοι τοποθέτησης του χρήστη σε πραγματικό χρόνο θα έχουν μικρότερους χρόνους εκτέλεσης παρέχοντας έτσι υπηρεσίες ανίχνευσης του χρήστη με χαμηλό χρόνο ανταπόκρισης.

Υπάρχει αρκετή βιβλιογραφία σχετικά με την εκτίμηση κάποιων κελιών του ραδιοχάρτη, για παράδειγμα με χρήση προσεγγιστικών μοντέλων ραδιοκυμάτων, όμως δεν υπάρχει βιβλιογραφία για την διαχείριση υπέρογκων πληροφοριών και φιλτράρισμα τους. Είναι σημαντικό να σημειωθεί ότι από τη μια θέλουμε να μειώσουμε τα ενεργά αποτυπώματα και σημεία πρόσβασης ώστε να μειωθεί το μέγεθος του ραδιοχάρτη και να γίνεται πιο γρήγορη επεξεργασία του, αλλά από την άλλη δεν θέλουμε να αφαιρέσουμε ή να παραλείψουμε δεδομένα τα οποία θα αυξήσουν σε μεγάλο βαθμό την απόκλιση των εκτιμήσεων θέσης του χρήστη.

4.3 Μοντέλο Συστήματος

Σε αυτή την ενότητα θα περιγράψουμε το μοντέλο συστήματος που εφαρμόζεται η τεχνική *LPRadioScale* πιο αυστηρά. Έστω ότι θεωρούμε μια περιοχή A η οποία καλύπτεται από ένα σύνολο ασύρματων σημείων πρόσβασης (Wi-Fi Access Points) $C=\{AP_1, AP_2, \dots, AP_N\}$ αλλά δεν είναι απαραίτητο η περιοχή A να είναι συνεχόμενη και επίσης θεωρούμε ότι τα σημεία πρόσβασης είναι κατανεμημένα άτακτα, δηλαδή χωρίς κάποια συγκεκριμένη κατανομή. Το κάθε σημείο πρόσβασης AP_i από το σύνολο C έχει μοναδική διεύθυνση MAC (BSSID/MAC address) η οποία είναι διαθέσιμη στο κοινό.

Οποιοσδήποτε χρήστης u μέσω του Logger μπορεί να συλλέξει μετρήσεις RSS και να τις αποστείλει στην υπηρεσία AnyPlace για να συμπεριληφθούν στον ραδιοχάρτη RM που θα δημιουργηθεί για την περιοχή A. Θεωρούμε ότι ο αλγόριθμος *LPRadioScale* (*LPRS* - *συντομογραφία*) θα τρέξει με τα δεδομένα της περιοχής A.

Ο *LPRS* λαμβάνει σαν είσοδο όλες τις μετρήσεις που συλλέχθηκαν για την περιοχή A υπό την μορφή που φαίνεται πιο κάτω.

Input: RSS-log file

<i>X</i>	<i>Y</i>	<i>Timestamp</i>	<i>Heading</i>	<i>MAC</i>	<i>dB</i>
----------	----------	------------------	----------------	------------	-----------

Η κάθε γραμμή του αρχείου *RSS-log* είναι της μορφής πιο πάνω, όπου το *X,Y* καθορίζουν την τοποθεσία που έγινε η μέτρηση ενώ το *Timestamp* και το *Heading* καθορίζουν τον χρόνο της μέτρησης (σε milliseconds) και την κατεύθυνση του χρήστη σε μοίρες. Το *MAC* και το *dB* αντιπροσωπεύουν ένα σημείο πρόσβασης που υπάρχει στην μέτρηση και την τιμή έντασης σήματος αντίστοιχα. Να σημειωθεί ότι όταν αναφερόμαστε σε κάποιο αποτύπωμα (fingerprint) αναφερόμαστε σε συγκεκριμένο συνδυασμό $\langle X-Y-Timestamp-Heading \rangle$, έστω *FP*. Κάθε αποτύπωμα έχει μια γραμμή στο αρχείο *RSS-log* για κάθε ένα από τα AP που λήφθηκαν στην συγκεκριμένη μέτρηση.

Output: Radio Map RM

Αφού ο αλγόριθμος διαβάσει το *RSS-log* θα κάνει εάν χρειάζεται κάποιο φιλτράρισμα και ακολούθως θα δημιουργήσει τον ραδιοχάρτη RM (όπως περιγράφηκε στην ενότητα 2.1.2) ο οποίος θα περιέχει μόνο AP από το σύνολο C' , όπου C' το σύνολο των εναπομείναν APs. Στον ραδιοχάρτη κανονικά έχουμε μια γραμμή για κάθε αποτύπωμα και για κάθε αποτύπωμα έχουμε μια στήλη για κάθε AP με τιμή την ένταση σήματος του. Ο *LPRS* συνδυάζει τα αποτυπώματα που έχουν κοινό $\langle X-Y-Heading \rangle$ και οι τιμές για κάθε στήλη AP είναι ο μέσος όρος μεταξύ των αποτυπωμάτων.

4.4 Περιγραφή προτεινόμενου αλγόριθμου – *LPRadioScale*

Ο *LPRadioScale* όπως αναφέραμε έχει σαν είσοδο το RSS-log file και σαν έξοδο τον ραδιοχάρτη RM. Το φιλτράρισμα που εφαρμόζει στηρίζεται στην βιβλιογραφία που περιγράφεται στην ενότητα 2.3.2, δηλαδή στην ιδέα του να μειώνεται ο αριθμός των σημείων πρόσβασης ώστε να μειώνεται και το μέγεθος του RM.

Τα φίλτρα που εφαρμόζονται είναι τα τρία που περιεγράφηκαν στην αντίστοιχη ενότητα και παρατίθενται στα γρήγορα πιο κάτω.

1. *Αριθμός μοναδικών τιμών έντασης σήματος (Number of unique dB values)*
Ένα AP πρέπει να έχει εμφανιστεί στη συλλογή αποτυπωμάτων με τουλάχιστον *THRESHOLD_UNIQUE_DB* τιμές
2. *Ποσοστό εμφανίσεων στη συλλογή αποτυπωμάτων (Percentage of appearances)*
Ένα AP πρέπει να έχει ποσοστό εμφανίσεων στη συλλογή αποτυπωμάτων τουλάχιστον *THRESHOLD_APPEARANCES*
3. *Κατανομή τυπικής απόκλισης στις ενδείξεις RSS (Standard deviation distribution on RSS)*
Οι ενδείξεις RSS (dB values) ενός AP πρέπει να έχουν τυπική απόκλιση τουλάχιστον *THRESHOLD_STDEV*

Επίσης, όπως ανέφερα στην προηγούμενη ενότητα ο *LPRadioScale* συνδυάζει αποτυπώματα βάση X-Y-Heading ώστε να έχει μόνο μια σειρά για κάθε αποτύπωμα στον RM και χρησιμοποιεί τον αριθμό των διαφορετικών Timestamp του αποτυπώματος ώστε να υπολογίσει τον μέσο όρο κάθε κελιού του RM.

Ολόκληρος ο αλγόριθμος αποτελείται από δύο φάσεις, την φάση προ-επεξεργασίας και την φάση κύριας επεξεργασίας και δημιουργίας του RM. Η πρώτη φάση αποτελείται από ένα MapReduce job και ακόμα ένα job μόνο με το Map όμως, ενώ η δεύτερη φάση αποτελείται από ένα μόνο MapReduce job. Ο *LPRS* τρέχει πάνω από το Hadoop.

Αρχίζοντας με την φάση *προ-επεξεργασίας (pre-processing phase)*, διαβάζεται το RSS-log από το *MapReduce1* και γίνονται οι απαραίτητοι υπολογισμοί ώστε για κάθε AP να έχουμε τις πληροφορίες που απαιτούνται για να κάνουμε τα τρία φιλτραρίσματα. Στη συνέχεια, τα στατιστικά αποτελέσματα τροφοδοτούνται στο *MapReduce2* και ανάλογα των στοιχείων του κάθε AP γίνεται σύγκριση με τα όρια που δόθηκαν του αλγορίθμου και φιλτράρονται και αφαιρούνται όσα AP δεν ικανοποιούν τα κριτήρια μας.

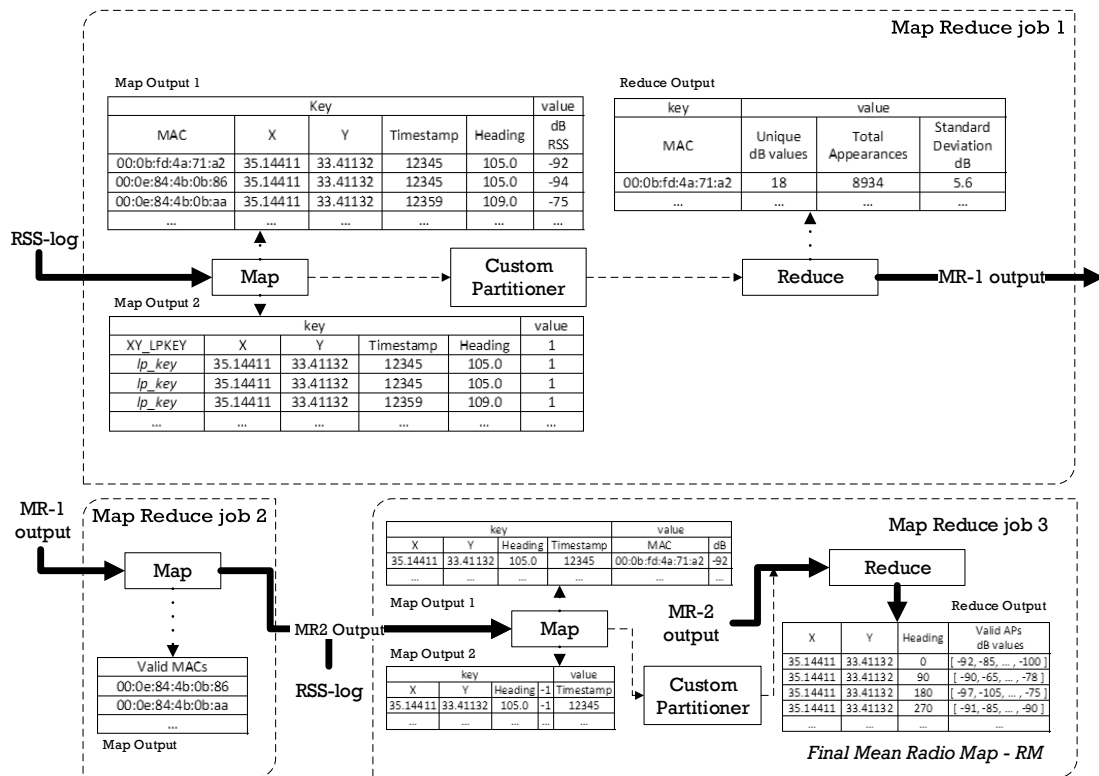
Στην δεύτερη και κύρια φάση *επεξεργασίας (main processing phase)*, πριν αρχίσει το διάβασμα του RSS-log ξανά και η κατασκευή του ραδιοχάρτη από το *MapReduce3*, διαβάζονται τα AP που πέρασαν το φιλτράρισμα (έξοδος του *MapReduce2*), αποτελούν το σύνολο *VAP*. Με αυτό τον τρόπο επιτυγχάνεται και μείωση των ενδιάμεσων αποτελεσμάτων στο *MapReduce3* αφού οι εγγραφές με AP που ανήκουν στο *VAP* θα αγνοηθούν. Με το τέλος της δεύτερης φάσης έχουμε έτοιμο τον ραδιοχάρτη όπου κάθε αποτύπωμα *<X-Y-Heading>* έχει τον μέσο όρο ενδείξεων RSS για κάθε AP και την τιμή *DEFAULT_UNSEEN_DB* για κάθε AP που δεν εμφανίστηκε καθόλου.

4.5 Λεπτομερής Ανάλυση αλγόριθμου *LPRadioScale*

Στην ενότητα αυτή θα γίνει λεπτομερής περιγραφή του αλγόριθμου *LPRadioScale* και της ακριβούς λειτουργίας της κάθε εργασίας *MapReduce*. Η ροή της εκτέλεσης περιγράφεται συνοπτικά από το *σχήμα 4.1* όπου φαίνονται η είσοδος και η έξοδος της κάθε *MapReduce* job. Για να γίνει πιο εύκολη κατανόηση του *LPRadioScale* μαζί με την περιγραφή θα παρατίθεται και ψευδοκώδικας.

4.5.1 Συλλογή στατιστικών στοιχείων για κάθε σημείο πρόσβασης

Στο πρώτο *MapReduce* job (*Algorithm 4.3*) έχουμε την συλλογή στατιστικών στοιχείων για κάθε AP τα οποία θα μας βοηθήσουν να ελέγξουμε αν το AP ικανοποιεί τα κριτήρια στα τρία φίλτρα που έχουμε στην πρώτη φάση του αλγορίθμου.



Σχήμα 4.2 : LPRadioScale algorithm - MapReduce jobs

Είσοδος στο *MR-1* είναι το *RSS-log* αρχείο επομένως σε κάθε *Map function* έχουμε την επεξεργασία κάθε γραμμής του αρχείου, δηλαδή ένα *AP* με την ένδειξη ισχύος του σήματος του σε κάποιο αποτύπωμα. Υπάρχουν δύο είδη εξόδου από το πρώτο *Map function*. Συγκεκριμένα το πρώτο που θέλουμε είναι ένα ζεύγος *κλειδιού-τιμής* όπου κλειδί είναι το αποτύπωμα σε συνδυασμό με το *MAC* του *AP* και τιμή είναι η ένδειξη ισχύς (θα χρησιμοποιηθεί για τα φίλτρα 1 και 3). Ακολούθως, παράγεται ακόμα ένα ζεύγος *κλειδιού-τιμής* το οποίο έχει κλειδί το αποτύπωμα σε συνδυασμό με ένα δικό μου μοναδικό αναγνωριστικό (θα χρησιμοποιηθεί ώστε να μετρήσουμε τον ολικό αριθμό των αποτυπωμάτων) και τιμή τον αριθμό 1. Επίσης υπάρχει ένας *Custom Partitioner* ο οποίος διαμοιράζει τα *key-value pairs* της πρώτης ομάδας στους *Reducers* μόνο βάση του *MAC* κομματιού από το κλειδί. Ενώ τα ζευγάρια της δεύτερης ομάδας κατανέμονται κανονικά βάση ολόκληρου του κλειδιού.

Ο λόγος που γίνεται αυτή η ειδική κατανομή είναι έτσι ώστε ζευγάρια του ίδιου *MAC* να καταλήξουν στον ίδιο *Reducer*, και επειδή το κλειδί τους περιέχει και το αποτύπωμα θα φτάσουν ταξινομημένα βάση του αποτυπώματος. Με αυτό επιτυγχάνουμε βελτιστοποιήσεις στη κύρια μνήμη των *Reducers*, όπως θα εξηγηθεί παρακάτω.

Οι Reducer λαμβάνουν τα key-value ζευγάρια ταξινομημένα βάση κλειδιού. Άρα, με την χρήση του Custom Partitioner μεταξύ του Map και του Reduce πετύχαμε όλα τα αποτυπώματα όπου εμφανίστηκε κάποιο MAC να έρθουν το ένα μετά το άλλο δίνοντας μας την δυνατότητα έτσι να μην αποθηκεύουμε πληροφορίες για άλλα APs παρά μόνο το εκάστοτε AP που υπάρχει αυτή την στιγμή στον Reducer. Κάθε φορά που παραλαμβάνεται ένα καινούριο MAC, αποθηκεύουμε τις πληροφορίες που έχουμε για το προηγούμενο AP και αρχίζουμε την αποθήκευση πληροφοριών για το τρέχων AP. Το τελευταίο AP θα αποθηκευτεί στην συνάρτηση *Cleanup* η οποία καλείται μετά την εκτέλεση όλων των *Reduce* καλεσμάτων. Όσο αφορά τα ζευγάρια της δεύτερης ομάδας, για κάθε ένα ζεύγος αυξάνουμε ένα ενιαίο μετρητή (παρέχεται από το Hadoop).

Algorithm 4.3: Collecting statistics for each Access Point

```

1: ----- Class MAPPER -----
2: Function MAP(k, v)
3:     key = getKey(v.MAC, v.X, v.Y, v.T, v.H);
4:     Emit(key, v.DB);    // group 1
5:     key = getSpecialLPKey(v.X, v.Y, v.T, v.H);
6:     Emit(key, 1);      // group 2

7: ----- Class PARTITIONER -----
8: Function LPCUSTOMPARTITIONER(k, v, num_of_reducers)
9:     if(Key is SpecialLPKey):    // group 2 key-value pair
10:         assign to Reducer# using k;
11:     else
12:         assign to Reducer# using only k.MAC

13: ----- Class REDUCER -----
14:     Array MAC_dbs[200];    // size to hold all possible dBs
15:     Var previousMAC = null; previousFingerprint = null;
16:     Var totalFingerprints = 0, totalDBSum = 0;
17: Function REDUCE(k, v)
18:     if(k is SpecialLPKey):
19:         ++HadoopCounters.UniqueFingerprints;
20:         Exit;
21:     if(k.MAC != previousMAC):
22:         handlePreviousMAC();
23:         Array.Fill(MAC_dbs, 0);
24:         previousFingerprint = null;
25:         totalFingerprints = 0, totalDBSum = 0;
26:     if(k.fingerprint is new):
27:         ++totalFingerprints;
28:         for each db in v do:
29:             ++MAC_dbs[-db];
30:             totalDBSum += db;
31:         previousFingerprint = k.fingerprint;
32:         previousMAC = k.MAC;

34: Function CLEANUP()
35:     handlePreviousMAC();    // process and stores the statistics
                             // for uniqueDBs, totalFingers, stdev

```

4.5.2 Φιλτράρισμα των σημείων πρόσβασης

Το δεύτερο MapReduce, *MR-2*, είναι πολύ απλό, αφού απλά διαβάζει τα στατιστικά στοιχεία που παρήγαγε το *MR-1* και βγάζει στην έξοδο μόνο όσα APs ικανοποιούν τα κριτήρια (thresholds) που έχουμε για κάθε ένα από τα τρία φίλτρα. Περιγράφεται σε ψευδοκώδικα στον *Algorithm 4.4*.

Algorithm 4.4: Filtering Access Points – MACs

```
1: ----- Class MAPPER -----  
2: Function MAP(k, v)  
3:     if(v.UniqueDBs >= THRESHOLD_UNIQUE_DBS  
4:         && v.AppearancePercentage >= THRESHOLD_APPEARANCES  
5:         && v.Stdev >= THRESHOLD_STDEV):  
6:         Emit(k, v);
```

4.5.3 Δημιουργία Ραδιοχάρτη

Στο τρίτο και τελευταίο MapReduce job, *MR-3* – *Algorithm 4.5*, έχουμε την κατασκευή του ραδιοχάρτη RM ο οποίος όμως θα συμπεριλαμβάνει μόνο τα σημεία πρόσβασης που έβγαλε στην έξοδο του το *MR-2*.

Αρχικά, πριν αρχίσει η εκτέλεση της Map συνάρτησης διαβάζεται από όλους τους κόμβους (Mappers) το τελικό αποτέλεσμα του *MR-2* αποθηκεύουν στην κύρια μνήμη τους όλα τα έγκυρα APs σε ένα σύνολο VAP. Ακολούθως αρχίζει κανονικά η εκτέλεση του *MR-3* όπου σαν είσοδος είναι και πάλι το αρχείο *RSS-log*. Ακολούθως για κάθε γραμμή του αρχείου παράγονται δύο ζευγάρια κλειδιού-τιμής. Το πρώτο ζεύγος, έχει κλειδί το αποτύπωμα και τιμή τον συνδυασμό MAC και dB (ένδειξη ισχύς), ενώ το δεύτερο έχει κλειδί το αποτύπωμα αλλά με τιμή τον αριθμό -1.

Ακολουθεί και πάλι ένας Custom Partitioner ο οποίος αυτή τη φορά στην ανάθεση των key-value pairs λαμβάνει υπόψη του το κομμάτι του αποτυπώματος που δεν περιέχει την χρονοσφραγίδα (Timestamp). Άρα με αυτό τον τρόπο όλα τα ζευγάρια του ίδιου αποτυπώματος ανεξαρτήτως χρόνου θα καταλήξουν στον ίδιο Reducer, αφού όπως είπαμε πριν ο *LPRadioScale* στον ραδιοχάρτη περιέχει μια φορά το κάθε αποτύπωμα και οι τιμές για κάθε AP είναι ο μέσος όρος όλων των μετρήσεων που έχουμε γι' αυτό. Επίσης, στο δεύτερο ζευγάρι επιλέχθηκε η τιμή -1 έτσι ώστε όταν ταξινομηθούν τα ζευγάρια πριν σταλούν στους Reducers να βρίσκεται ακριβώς πριν από τα ζευγάρια της πρώτης ομάδας. Πιο συγκεκριμένα, ο κάθε Reducer θα λαμβάνει πρώτα το αποτύπωμα

Fi το οποίο περιέχει την τιμή -1 στο κλειδί του και σαν τιμή θα είναι η λίστα με όλες τις χρονοσφραγίδες του αποτυπώματος. Μετά θα ακολουθήσουν όλα τα ζευγάρια που αφορούν το *Fi* και σαν τιμή θα έχουν λίστα με ζευγάρια τύπου $\langle \text{MAC}, \text{dB} \rangle$. Με αυτό τον τρόπο οι Reducer θα μπορούν όπως και πριν να αποφύγουν τη χρήση υπέρογκης κύριας μνήμης αφού κάθε φορά κρατάνε στοιχεία μόνο για ένα αποτύπωμα. Ο κάθε Reducer μόλις λάβει το τελευταίο ζευγάρι του αποτυπώματος *Fi* θα υπολογίσει το μέσο όρο για κάθε AP που έχει εισακουστεί και θα βγάλει στην έξοδο την γραμμή του ραδιοχάρτη που αφορά το *Fi*. Όσα AP που ανήκουν στο VAP δεν έχουν τιμή στο αποτύπωμα τους ανατίθεται η τιμή *DEFAULT_UNSEEN_DB* που δίνεται σαν παράμετρος στον αλγόριθμο.

Algorithm 4.5: Collecting statistics for each Access Point – MAC

```

1: ----- Class MAPPER -----
2:   Set ValidMACs;
3:   Function SETUP()
4:     for each MAC in MR-2 output:
5:       ValidMACs.put(MAC);
6:   Function MAP(k, v)
7:     if(!ValidMACs.contains(v.MAC)):
8:       exit;
9:     key = getKey(v.X, v.Y, v.H, v.T);
10:    Emit(key, <v.MAC, v.DB> );           // group 1
11:    key = getKey(v.X, v.Y, v.H, -1);
12:    Emit(key, v.T);                       // group 2

13: ----- Class PARTITIONER -----
14: Function LPCUSTOMPARTITIONER(k, v, num_of_reducers)
15:   assign to Reducer# using only k.X,k.Y,k.H

16: ----- Class REDUCER -----
17:   Set ValidMACs;
18:   TreeMap<> currentXYMacs; HashMap<> currentXYCounters;
19:   Var currentBatchSize = -1; consumedBatchSize = -1;
20:   Function SETUP()
21:     for each MAC in MR-2 output:
22:       ValidMACs.put(MAC);

23:   Function REDUCE(k, v)
24:     if(k.T == -1):                       // group 2
25:       currentXYMacs.clear();currentXYCounters.clear();
26:       currentBatchSize = UniqueTimestampsInList(v);
27:       consumedBatchSize = 0;
28:       Exit;
29:     for each <mac,db> in v do:           // group 1
30:       currentXYCounters[mac] += db;
31:     ++consumedBatchSize;
32:     if(consumedBatchSize == currentBatchSize):
33:       for each M in ValidMACs:
34:         calculate average for M and append to radio map line

```

Όπως βλέπετε ο αλγόριθμος *LPRadioScale* είναι αρκετά απλός όμως επιλύει το πρόβλημα της δημιουργίας ραδιοχάρτη για μεγάλα δεδομένα αποτελεσματικά και χωρίς προβλήματα μνήμης αφού οι απαιτήσεις του για κύρια μνήμη είναι πολύ μικρές λόγω της χρήσης εξειδικευμένων partitioners που διανέμουν τα δεδομένα με τέτοια σειρά ώστε οι Reducers να μπορούν χωρίς να κρατούν πολλά στοιχεία να κάνουν τους απαραίτητους υπολογισμούς. Επίσης εφαρμόζουμε τρία είδη φίλτρων έτσι ώστε το μέγεθος του ραδιοχάρτη να μειωθεί ακόμα περισσότερο. Τα τρία κριτήρια (thresholds) που χρησιμοποιεί ο αλγόριθμος δίνονται στο πρόγραμμα ως παράμετροι, όπως επίσης και η τιμή για την ισχύ σήματος των σημείων πρόσβασης που δεν έχουν ενδείξεις σε κάποιο αποτύπωμα.

Να σημειώσουμε ξανά ότι ίσως ο αλγόριθμος να διέφερε εάν θέλαμε να έχουμε κέρδος καθαρά σε επίδοση και μείωση του χρόνου εκτέλεσης, ακόμα και σε επίπεδο framework. Εμείς όμως ενδιαφερόμαστε για διαχείριση Μεγάλων Δεδομένων και δώσαμε έμφαση στο να δουλεύει ανεξάρτητα του μεγέθους τους και δεν μας ενδιαφέρει το αν θα τερματίσει σε 3 ώρες ή 3.5 ώρες.

Κεφάλαιο 5

AnyPlace Mobile – Εφαρμογές για Android συσκευές

5.1	AnyPlace Navigator	55
5.1.1	Ανίχνευση και τοποθέτηση σε πραγματικό χρόνο (Positioning)	57
5.1.2	Πλοήγηση & Δρομολόγηση (Navigation)	59
5.1.3	Αναζήτηση σημείων ενδιαφέροντος (Search Points of Interest)	60
5.2	AnyPlace Logger	61

Σε αυτή την ενότητα θα μελετήσουμε αναλυτικά τις εφαρμογές που αναπτύχθηκαν για έξυπνες συσκευές Android, αρχικά τον Navigator στην υπό ενότητα 5.1 και ακολούθως τον Logger στην υπό ενότητα 5.2

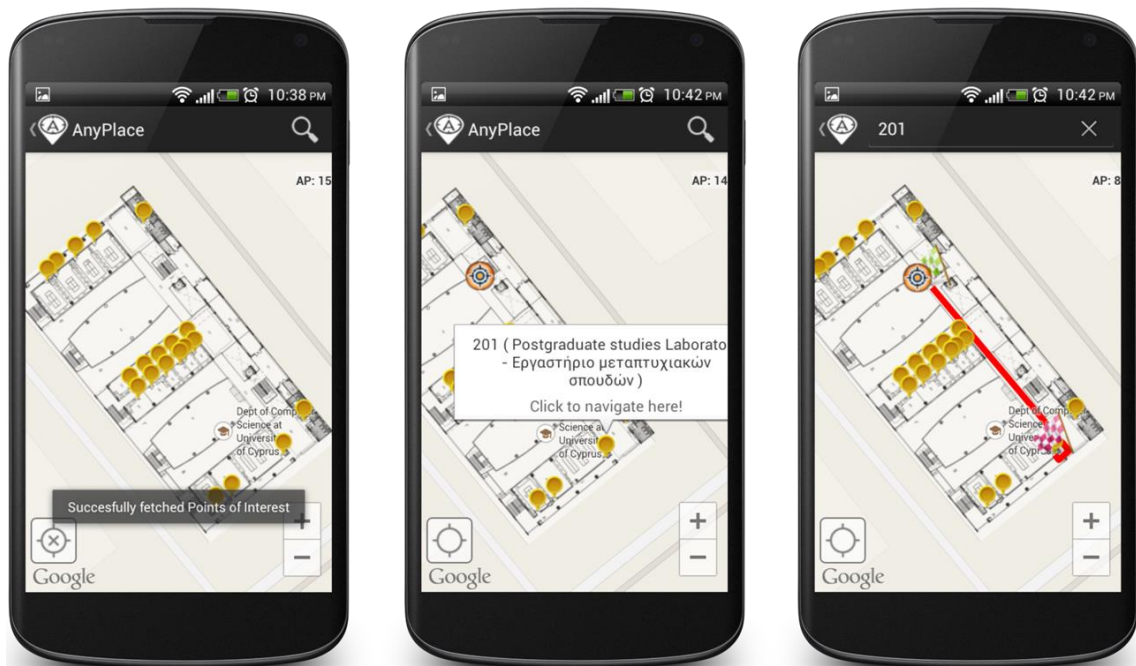
5.1 AnyPlace Navigator

Ο AnyPlace Navigator είναι η κύρια εφαρμογή του AnyPlace για κινητές συσκευές Android με έκδοση λογισμικού 2.3.3 ή νεότερη, η οποία παρέχει στον χρήστη τοποθέτηση και πλοήγηση σε εσωτερικούς χώρους σε πραγματικό χρόνο.

Η εφαρμογή έχει αναπτυχθεί με modular τρόπο ώστε να μπορεί εύκολα να επεκταθεί η λειτουργία του με ανεξάρτητες λειτουργίες.

Με την εκκίνηση της εφαρμογής, γίνεται αυτόματη ανάκτηση της τοποθεσίας του χρήστη και εμφανίζονται τα κοντινότερα κτίρια γύρω του ώστε να επιλέξει που βρίσκεται. Εάν τα ανακτηθέντα κτίρια είναι μόνο ένα τότε επιλέγεται αυτόματα αυτό

και ο χρήστης κλίνεται να επιλέξει τον όροφο του (εάν και πάλι δεν είναι μοναδικός όπου θα επιλεγεί αυτόματα).



Εικόνα 5.1 : Στιγμιότυπα λειτουργίας του AnyPlace Navigator, α) φόρτωση σημείων ενδιαφέροντος (POIs) – αριστερά, β) ανίχνευση χρήστη σε πραγματικό χρόνο – μεσαία, γ) προβολή οδηγιών δρομολόγησης – δεξιά. Το κουμπί κάτω αριστερά ενεργοποιεί την ανίχνευση του χρήστη σε πραγματικό χρόνο.

Η επιλογή ορόφου αμέσως αρχίζει την ανάκτηση του αρχιτεκτονικού σχεδίου και των σημείων ενδιαφέροντος που έχουν χαρτογραφηθεί μέσω του Architect (κεφάλαιο 6.4). Ολόκληρη η διεπαφή της εφαρμογής βασικά αποτελείται από τον χάρτη όπου τοποθετείται αυτόματα το αρχιτεκτονικό σχέδιο του ορόφου που επιλέγεται. Ο χρήστης μπορεί να επιλέξει κάποιο σημείο ενδιαφέροντος πάνω στον χάρτη ώστε να δει λεπτομερής περιγραφή του ή να λάβει οδηγίες πλοήγησης προς αυτό (ενότητα 5.1.2), εικόνα 5.1 (β).

Εκτός από τον χάρτη, στην κορυφή της εφαρμογής υπάρχει μια μπάρα αναζήτησης όπου ο χρήστης έχει την ικανότητα να αναζητήσει βάση λέξεων κλειδιά συγκεκριμένα σημεία ενδιαφέροντος. Επιπλέον, εάν ο χρήστης έχει ενεργοποιημένη την ανίχνευση του σε πραγματικό χρόνο, λαμβάνουμε υπόψη την μετακίνηση του στον χώρο ώστε να γίνεται αυτόματη τοποθέτηση του στον χάρτη, μέσω του Tracker module (ενότητα 5.1.1).

Στη συνέχεια, θα εξηγηθούν αναλυτικά όλα τα επιμέρους κομμάτια (modules) του AnyPlace Navigator.

5.1.1 Ανίχνευση και τοποθέτηση σε πραγματικό χρόνο (Positioning)

Ο *AnyPlace Tracker* είναι η επιμέρους μονάδα υπεύθυνη για την ανίχνευση του χρήστη και η τοποθέτηση του πάνω στον χάρτη σε πραγματικό χρόνο καθώς κινείται.

Αυτή τη στιγμή παρέχονται δύο διαφορετικές μέθοδοι ανίχνευση του χρήστη βάση της ακρίβειας που εκτιμάται η θέση του: α) Τοποθέτηση μεγάλης ακρίβειας (Fine positioning) όπου η εκτίμηση της θέσης του χρήστη γίνεται βάση αλγορίθμων που αναπτύχθηκαν από ερευνητές του εργαστηρίου Διαχείρισης Δεδομένων και Συστημάτων στο Πανεπιστήμιο Κύπρου (DMSL@UCY) και β) Τοποθέτηση μικρής ακρίβειας (Coarse positioning) όπου χρησιμοποιούνται εξωτερικές πηγές. Και οι δύο μέθοδοι θα αναλυθούν πιο κάτω.

5.1.1.1. Τοποθέτηση Μεγάλης Ακρίβειας (Fine Positioning)

Ο AnyPlace Navigator χρησιμοποιεί το *Wi-Fi Tracker* ως την κύρια μονάδα εκτίμησης της τοποθεσίας του χρήστη. Το module αυτό υλοποιεί ένα δικό μας αλγόριθμο [33] που βασίζεται στα δίκτυα RBF (Radial Basis Function networks). Ο αλγόριθμος αυτός εκμεταλλεύεται της RSS μετρήσεις γειτονικών σημείων πρόσβασης έτσι ώστε με κάποια επεξεργασία να ταυτίζει τις διάφορες ενδείξεις RSS σε πραγματικές συντεταγμένες τοποθεσίας πάνω στον χάρτη.

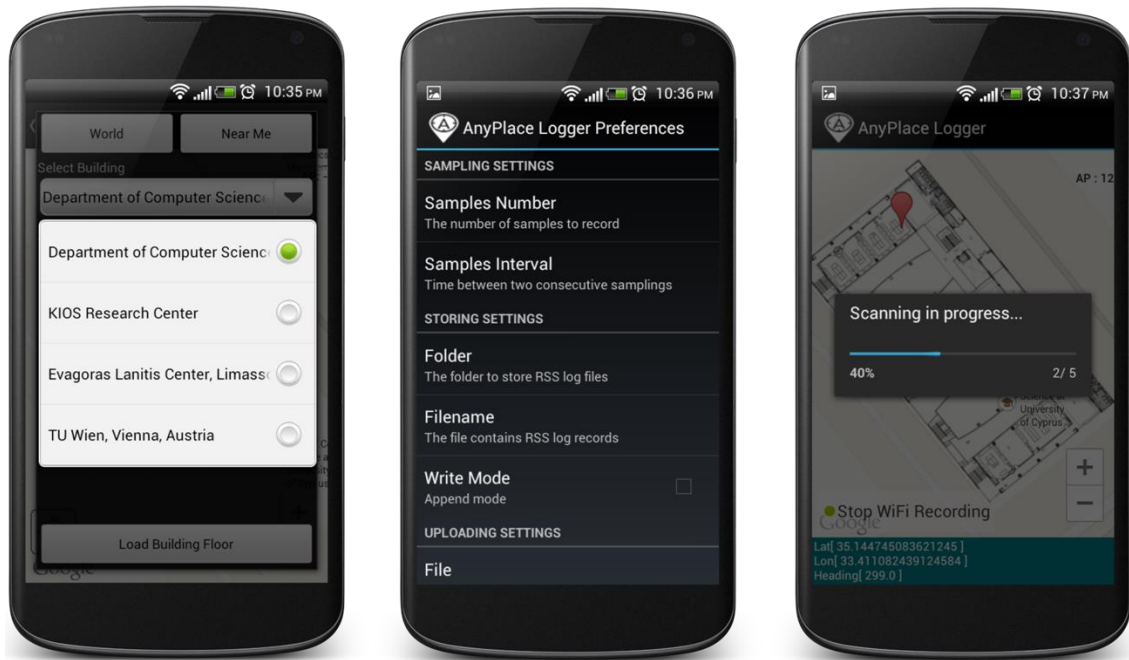
Αφού ο χρήστης ενεργοποιήσει αυτή την υπηρεσία (με το να πατήσει το κουμπί κάτω αριστερά στην *εικόνα 5.1*) η εφαρμογή ζητά από το AnyPlace API να καταφορτώσει ραδιοχάρτη που αντιστοιχεί στον όροφο που βρίσκεται και αμέσως ξεκινά να τρέχει ο αλγόριθμος. Με κάθε νέα εκτίμηση της τοποθεσίας του χρήστη γίνεται ενημέρωση της και πάνω στον χάρτη μέσω της ειδικής «*κουκίδας του χρήστη*», *εικόνα 5.1 (α)* με χρώμα πορτοκαλί.

Να σημειωθεί ότι αφού κατεβεί ο ραδιοχάρτης του ορόφου, πλέον δεν χρειάζεται οποιαδήποτε επικοινωνία της εφαρμογής με το AnyPlace API και γενικά δεν χρειάζεται σύνδεση στο διαδίκτυο για κάτι. Ως εκ τούτου, αποφεύγουμε την παρακολούθηση του χρήστη από τρίτους ή και ακόμα από την υπηρεσία του AnyPlace αυτή καθαυτή. Επιπλέον, με την αποφυγή επιπρόσθετης επικοινωνίας αποφεύγεται αχρείαστη σπατάλη ενέργειας της συσκευής αφού ολόκληρη η επεξεργασία για εκτίμηση της θέσης γίνεται στο κινητό.

5.1.1.2. Τοποθέτηση μικρής ακρίβειας (Coarse Positioning)

Κάποτε, η εκτίμηση της θέσης του χρήστη πρέπει να γίνεται γρήγορα άσχετα του αν δεν εγγυάται μεγάλη ακρίβεια. Για παράδειγμα, με την εκκίνηση της εφαρμογής απλά θέλουμε την θέση του χρήστη ώστε να εμφανίσουμε κοντινά κτίρια, όπου η ύπαρξη απόκλισης μέχρι και μερικά μέτρα είναι αποδεκτή. Για να έχουμε γρήγορη εκτίμηση θέσης του χρήστη γρήγορα χρησιμοποιούμε εξωτερικές πηγές, όπως την διεπαφή τοποθέτησης από την Google (Google Location API), η οποία είναι ήδη προ-εγκατεστημένη και ενεργή στις πλείστες κινητές συσκευές με λειτουργικό σύστημα Android. Λόγω του ότι η υπηρεσία αυτή χρησιμοποιεί τα κύματα και της ραδιοσυχνότητες από τα δίκτυα κινητής τηλεφωνίας αλλά και των γειτονικών σημείων πρόσβασης μπορεί να μας παρέχει εκτίμηση της θέσης του χρήστη σε συντεταγμένες χάρτη σε πολύ μικρό χρονικό διάστημα όμως με μικρή ακρίβεια. Το αρνητικό αυτής της υπηρεσίας είναι ότι χρειάζεται σύνδεση στο διαδίκτυο ώστε να μπορεί η συσκευή να επικοινωνεί με την υπηρεσία της Google [34].

Εναλλακτικά, σε περίπτωση που ο χρήστης δεν επιθυμεί να αποστέλλονται οι πληροφορίες του σε τρίτους απενεργοποιείται η αυτόματη επιλογή κτιρίων και ο χρήστης μπορεί να επιλέξει μόνος του το κτίριο που βρίσκεται μέσω της εύκολης διεπαφής αναζήτησης που παρέχει ο *Navigator*, εικόνα 5.2 (α).



Εικόνα 5.2 : Στιγμιότυπα λειτουργίας του AnyPlace Logger, α-αριστερά) χειροκίνητη επιλογή κτιρίου, β-μέση) οθόνη ρυθμίσεων, γ-δεξιά) συλλογή RSS αποτυπωμάτων.

5.1.2 Πλοήγηση & Δρομολόγηση (Navigation)

Ο AnyPlace Navigator εκτός από την λειτουργία τοποθέτησης περιέχει μια μονάδα υπηρεσίας (module) η οποία παρέχει στον χρήστη την δυνατότητα πλοήγησης σε κάποιο σημείο ενδιαφέροντος εντός κτιρίου μέσω διαδρομής πάνω στο χάρτη.

Υπάρχουν πολλοί τρόποι να αιτηθεί ο χρήστης οδηγίες πλοήγησης .

- Ο χρήστης μπορεί μέσω της διεπαφής αναζήτησης να βρει κάποιο σημείο ενδιαφέροντος το οποίο επιλέγοντας το από τη λίστα αποτελεσμάτων αυτόματα ζητά οδηγίες δρομολόγησης από την υπηρεσία AnyPlace, μέσω του AnyPlace API
- Πάνω στον χάρτη αφού επιλεγεί ο όροφος που βρίσκεται ο χρήστης, φορτώνονται και σημειώνονται όλα τα σημεία ενδιαφέροντος που χαρτογραφήθηκαν μέσω του Architect. Ακολουθώντας, επιλέγοντας ένα σημείο ενδιαφέροντος υπάρχει επιλογή να ζητηθεί δρομολόγηση προς αυτό το σημείο.

Αφού γίνει αίτημα για την δρομολόγηση προς το σημείο ενδιαφέροντος η υπηρεσία του AnyPlace απαντά με το δρομολόγιο μέσα στο κτίριο από την εκάστοτε θέση του

χρήστη προς το σημείο που επιλέγηκε και η εφαρμογή σχεδιάζει στον χάρτη την διαδρομή ώστε να μπορεί ο χρήστης να την ακολουθήσει καθώς κινείται.

Το σύστημα αυτή τη στιγμή παρέχει υπηρεσία δρομολόγησης μόνο μεταξύ των ορόφων ενός κτιρίου.

Παράδειγμα χρήσης της πλοήγησης φαίνεται στην *εικόνα 5.1 (γ)*.

5.1.3 Αναζήτηση σημείων ενδιαφέροντος

Το συστατικό αναζήτησης (search component) επεκτείνει τον Navigator επιτρέποντας την αναζήτηση σημείων ενδιαφέροντος. Το module αυτό βάσει του εκάστοτε επιπέδου μεγέθυνσης κάνει διαφορετικό είδος αναζήτησης.

Η *αναζήτηση κτιρίου* παρέχει την δυνατότητα στον χρήστη να αναζητήσει βάση μιας γραμματοσειράς (keywords) κάποιο σημείο πρόσβασης εντός κτιρίου. Για παράδειγμα, μια γραμματοσειρά όπως «γραφείο» θα επιστρέψει όλα τα αποτελέσματα των σημείων που έχουν τη λέξη «γραφείο» είτε στο όνομα τους είτε στην περιγραφή τους, λογικά θα συμπεριλαμβάνονται όλα τα γραφεία καθηγητών εάν είμαστε στο κτίριο ενός πανεπιστημίου. Η επιλογή οποιουδήποτε αποτελέσματος αυτόματα ενεργοποιεί την μονάδα πλοήγησης η οποία θα φέρει οδηγίες δρομολόγησης προς αυτό το σημείο. Η αναζήτηση των σημείων ενδιαφέροντος γίνεται εξ' ολοκλήρου από την συσκευή και δεν υπάρχει επικοινωνία με την υπηρεσία μας.

Η *αναζήτηση μεγάλης ακτίνας* αφορά αναζήτηση τοπίων ή και σημείων ενδιαφέροντος, πλέον όχι μέσω της υπηρεσίας AnyPlace αλλά μέσω της υπηρεσίας Τοποθεσιών της Google (Google Places API [34]). Συνεπώς, για τη χρήση αυτής της λειτουργίας απαιτείται σύνδεση στο διαδίκτυο. Η επιλογή ενός από τα αποτελέσματα σε αυτού του είδους την αναζήτηση θα έχει ως αποτέλεσμα την μεταφορά του χάρτη σε εκείνη την τοποθεσία και εάν εκείνος ο χώρος έχει χαρτογραφηθεί εσωτερικά μέσω του Architect θα υπάρξει η κατάλληλη ένδειξη ώστε να κατεβάσει τον χάρτη κτιρίου.

Όπως ανέφερα πιο πάνω, η αναζήτηση χρησιμοποιεί το επίπεδο μεγέθυνσης του χάρτη τη στιγμή που γίνεται η αναζήτηση από το χρήστη. Συγκεκριμένα, εάν το επίπεδο

μεγέθυνσης είναι μεγαλύτερο ή ίσο με 19 γίνεται αναζήτηση κτιρίου, ενώ εάν είναι μικρότερο του 19 γίνεται αναζήτηση μεγάλης ακτίνας.

5.2 AnyPlace Logger

Το AnyPlace είναι απόγονος και η συνέχεια – επέκταση- του βραβευμένου συστήματος *Airplace*, και ο AnyPlace Logger αποτελεί το συστατικό που κληρονόμησε μεγάλο ποσοστό της λειτουργίας του από το *Airplace* σχετικά με την συλλογή των RSS αποτυπωμάτων. Είναι μια εύκολη στη χρήση εφαρμογή η οποία αναπτύχθηκε με βάση το RSS API του λειτουργικού συστήματος Android με βασική λειτουργία την συλλογή και αποθήκευση RSS ενδείξεων σε προκαθορισμένα και μη χρονικά διαστήματα σε επιλεγμένες τοποθεσίες. Σκοπός του Logger είναι η συλλογή των αποτυπωμάτων RSS με τα οποία θα γίνει η δημιουργία του ραδιοχάρτη αργότερα ο οποίος τυγχάνει χρήσης από τους αλγόριθμους τοποθέτησης του χρήστη σε πραγματικό χρόνο.

Με την εκκίνηση της εφαρμογής, ο χρήστης επιλέγει το κτίριο που βρίσκεται μέσω της διεπαφής που παρέχει ο Logger όπου εμφανίζονται τα κτίρια που είναι χαρτογραφημένα στην υπηρεσία AnyPlace, *εικόνα 5.2 (α)*. Ακολούθως, αφού γίνει επιλογή κτιρίου, ορόφου και φορτωθεί το αρχιτεκτονικό σχέδιο ο χρήστης μπορεί να κινήσει (pan) και να μεγεθύνει (zoom) τον χάρτη ώστε να έχει την ακρίβεια που θέλει για να βρει την ακριβή θέση του στον χάρτη. Αφού βρει την θέση του σημειώνει στον Logger πατώντας πάνω στον χάρτη το σημείο για το οποίο θέλει να πάρει μετρήσεις. Παρέχεται ειδική διεπαφή μέσω της οποίας ο χρήστης μπορεί να αλλάξει και να επισκοπήσει τις διάφορες ρυθμίσεις της εφαρμογής, όπως μετρήσεις που λαμβάνονται ανά σημείο, χρονικό διάστημα ανάμεσα σε κάθε μέτρηση, όνομα αρχείου αποθήκευσης κλπ. όπως φαίνεται στην *εικόνα 5.2 (β)*.

Η έναρξη των μετρήσεων για το καθορισμένο σημείο στο χάρτη γίνεται με το πάτημα του ειδικού κουμπιού «*Start Wi-Fi Recording*» και οι μετρήσεις αποθηκεύονται στην δευτερεύουσα μνήμη της συσκευής, *εικόνα 5.2 (γ)*. Οι πληροφορίες που συλλέγονται περιέχουν την MAC διεύθυνση (WLAN AP MAC address) την ένδειξη έντασης του σήματος για κάθε γειτονικό σημείο πρόσβασης, οι ακριβή συντεταγμένες του

επιλεγμένου σημείου στο χάρτη (latitude & longitude), το κτίριο και ο όροφος που βρίσκεται ο χρήστης και τέλος το είδος συσκευής, το οποίο αργότερα χρησιμοποιείται για επίλυση προβλημάτων ανάμεσα στις μετρήσεις από διαφορετικές συσκευές (device diversity issues). Με αυτό τον τρόπο, υπάρχει μια άμεση συσχέτιση 1-προς-1 των παραγόμενων αρχείων RSS-LOGS με κάθε κτίριο, έτσι μπορούμε να δημιουργήσουμε ξεχωριστό ραδιοχάρτη για κάθε κτίριο και να μειώσουμε τον αριθμό αποτυπωμάτων που επεξεργάζονται αφού αποτυπώματα άσχετων κτιρίων δεν θα συμπεριληφθούν στο ραδιοχάρτη. Τέλος, αφού ο χρήστης συλλέξει τα αποτυπώματα για τα σημεία που θέλει μπορεί να ανεβάσει το αρχείο RSS-LOG (όπου είναι καταγραμμένα όλα τα αποτυπώματα) μέσω της επιλογής «*Upload the RSS Log*» από το κυρίως μενού.

Ο AnyPlace Logger δίνει την δυνατότητα στον οποιοδήποτε να συμβάλει με τον τρόπο του στην χαρτογράφηση των κτιρίων και την συλλογή αποτυπωμάτων ανώνυμα και διάφανα, αφού δεν γίνεται καμιά συλλογή πληροφοριών για προσωπικά στοιχεία του χρήστη της συσκευής. Επιπλέον, η συλλογή αποτυπωμάτων είναι μια απλή διαδικασία που δεν δυσχεράνει τον χρήστη και δεν απαιτεί από το χρήστη να αφιερώσει πολύ χρόνο. Αντίθετα, η συλλογή αποτυπωμάτων από μια οντότητα μόνο είναι μια απίστευτα χρονοβόρα διαδικασία που απαιτεί όχι μόνο χρόνο αλλά προσωπικό, συνεπώς έχει περισσότερο κόστος. Αυτή άλλωστε είναι η αιτία που έχει εξαπλωθεί τόσο έντονα τα τελευταία χρόνια το μοντέλο του πληθωπορισμού (crowdsourcing).

Κεφάλαιο 6

AnyPlace Web – Υποδομή Ραχοκοκαλιάς και Συστατικά

6.1	Web-server Layer	63
6.2	Data Layer	66
6.3	AnyPlace Viewer	69
6.4	AnyPlace Architect	73
6.5	AnyPlace Developers	74

Σε αυτό το κεφάλαιο θα παρουσιάσουμε αναλυτικά την υποδομή ραχοκοκαλιάς που υποστηρίζει ολόκληρη την υπηρεσία AnyPlace. Συγκεκριμένα, θα αναφερθούμε στο επίπεδο των εξυπηρετητών (Web-Server Layer), στο επίπεδο δεδομένων (Data Layer) και τέλος μια μικρή περιγραφή των εφαρμογών διαδικτύου AnyPlace Viewer, Architect και Developers.

6.1 Επίπεδο Εξυπηρετητών – Web-server Layer

Η υποδομή του AnyPlace αποτελείται από πολλαπλά συστατικά που θα εξηγηθούν στη συνέχεια.

Όλοι οι εξυπηρετητές μας που χειρίζονται τα αιτήματα τρέχουν το πλαίσιο προγραμματισμού Play! το οποίο είναι ένα framework που ακολουθεί την αρχιτεκτονική MVC (Model-View-Controller)[35] και βασίζεται στις γλώσσες προγραμματισμού Java [36] και Scala [37]. Είναι ένα framework που απέδειξε την σταθερότητα του και ήδη υιοθετήθηκε από πολλές μεγάλες εταιρείες όπως LinkedIn [38] λαμβάνοντας μέρος σε μεγάλο ποσοστό της υποδομής τους.

Το Play βασίζεται σε μια αρχιτεκτονική που δεν διατηρεί την κατάσταση μεταξύ των αιτημάτων (stateless) και δεν μπλοκάρει σε κάθε αίτημα μέχρι να τελειώσει ώστε χειριστεί το επόμενο (non-blocking) χρησιμοποιώντας μια πλήρη ασύγχρονη υλοποίηση πάνω από το μοντέλο AKKA [39]. Ως εκ τούτου, μπορεί εύκολα να επεκταθεί γραμμικά με την εισαγωγή περισσότερων μηχανών αλλά επίσης παρέχει και ευκολία στην πρόβλεψη της συμπεριφοράς και επίδοσης του ανάλογα του αριθμού των αιτημάτων.

Επιπρόσθετα, είναι ένα framework που επιτρέπει την εύκολη δημιουργία ενός Web 2.0 API παρέχοντας έτοιμη λειτουργικότητα σχετικά με κάποιες σύνθετες ρυθμίσεις που θα έπρεπε να γίνουν με άλλες βιβλιοθήκες και frameworks έτσι μπορέσαμε να επικεντρωθούμε στην κύρια λειτουργία του API χωρίς σπατάλη χρόνου.

Η πλείστη λειτουργία μας είναι υλοποιημένη στην γλώσσα Java ενώ υπάρχουν κάποια μικρά modules υλοποιημένα σε άλλες γλώσσες όπως Python [40] και Bash [41] .

6.1.1 AnyPlace API

Το AnyPlace όπως αναφέρθηκε και προηγουμένως παρέχει ένα Web2.0 API (διεπαφή προγραμματισμού εφαρμογής) η οποία μπορεί να χρησιμοποιηθεί από τρίτους (3rd party) προγραμματιστές για την ανάπτυξη εφαρμογών που θέλουν να χρησιμοποιήσουν την υπηρεσία AnyPlace.

Το API αναπτύχθηκε από την αρχή με πλήρη υποστήριξη για JSON documents (μηνύματα κειμένου σε μορφή JSON – JavaScript Notation). Για αυτό το λόγο όλες οι αιτήσεις και απαντήσεις στο API φέρουν ένα JSON object που περιέχει όλες τις πληροφορίες που απαιτούνται από το εκάστοτε API κάλεσμα.

6.1.2 Συστατικά Εξυπηρετητή – Server modules

Εκτός από το API, στο επίπεδο εξυπηρετητών βρίσκονται και μερικά άλλα συστατικά που εξειδικεύονται σε μικρές συγκεκριμένες λειτουργίες σχετικά με την επεξεργασία των δεδομένων πριν και μετά την προβολή τους στον χρήστη αλλά και για την ομαλή λειτουργία των διαδικτυακών εφαρμογών.

Ο κύριος εξυπηρετητής που σερβίρει τις εφαρμογές διαδικτύου είναι ο JBoss Netty που παρέχεται μαζί με το Play framework. Ακολουθώντας, υπάρχει ένα module, ο *AnyPlace Tiler*, ο οποίος είναι υπεύθυνος για να επεξεργαστεί τα αρχιτεκτονικά σχέδια των ορόφων που ανεβάζονται μέσω του AnyPlace Architect ώστε να δημιουργηθούν τα απαραίτητα «πλακάκια» (map tiles) που θα εμφανίζονται στον χάρτη των εφαρμογών στα κινητά Android. Όλες οι εφαρμογές μας βασίζονται στους χάρτες από την Google (Google Maps) έτσι για να είναι πολύ πιο γρήγορη η φόρτωση μεγάλων σχεδίων ορόφων στις συσκευές έπρεπε να γίνει η δημιουργία των map tiles ώστε να υπάρχει δικός μας παροχέας των tiles (AnyPlace Tile Provider). Η Google απαιτεί τα πλακάκια να είναι διαστάσεων 256 x 256 pixels ώστε να ταιριάζουν και να μπορούν να εφαρμοστούν στους χάρτες τους.

Έτσι, αναπτύχθηκαν κάποια μικρά εργαλεία γραμμένα στις γλώσσες *Python* και *Bash* τα οποία χρησιμοποιούν άλλα προγράμματα επεξεργασίας εικόνας, *ImageMagick* [42] και *AdvPNG* [43], τα οποία φορτώνουν τα αρχιτεκτονικά σχέδια που αποστέλλονται μέσω του Architect και παράγουν κάνοντας κάποιους υπολογισμούς τα πλακάκια για κάθε επίπεδο μεγέθυνσης που θέλουμε να υποστηρίζουν οι εφαρμογές μας στις κινητές συσκευές Android. Αυτή τη στιγμή, παρέχονται tiles μόνο για τα επίπεδα μεγέθυνσης 19 μέχρι 21 συμπεριλαμβανομένων και ο λόγος που επιλέχθηκε αυτό το πεδίο είναι το ότι σε μικρότερα επίπεδα δεν είναι ευδιάκριτα τα δωμάτια μέσα στα κτίρια που έχουμε δοκιμάσει μέχρι τώρα έτσι δεν υπάρχει λόγος να παρέχονται tiles για zoom levels τα οποία δεν θα είναι χρήσιμα. Η επεξεργασία και η δημιουργία των tiles μπορεί να γίνει ανεξάρτητα και παράλληλα για κάθε όροφο αφού δημιουργείται ένας φάκελος

(directory) για τον κάθε ένα και όλα τα tiles τοποθετούνται μέσα ανάλογα του επιπέδου μεγέθυνσης.

Στη συνέχεια, υπάρχει το module της δημιουργίας ραδιοχαρτών – *LPRadioScale*, όπως περιεγράφηκε στο κεφάλαιο 4, το οποίο είναι υπεύθυνο για την δημιουργία του ραδιοχάρτη κάθε ορόφου που χρησιμοποιείται για την τοποθέτηση του χρήστη σε πραγματικό χρόνο.

Τέλος, έχουμε το module πλοήγησης και δρομολόγησης (*Navigation module*) το οποίο είναι υπεύθυνο να δημιουργεί και να επιστρέφει τις οδηγίες δρομολόγησης και πλοήγησης από ένα σημείο ενδιαφέροντος σε άλλο εντός ενός κτιρίου μέσω των ειδικών αιτημάτων του AnyPlace API. Ο αλγόριθμος που χρησιμοποιείται για την εύρεση της κοντινότερης δρομολόγησης είναι ο αλγόριθμος Εύρεσης Κοντινότερου Μονοπατιού από μια Πηγή Dijkstra [44] – Single Source Shortest Path Dijkstra Algorithm.

6.2 Επίπεδο Δεδομένων – Data Layer

Το επίπεδο δεδομένων είναι το δεύτερο κομμάτι της υποδομής μας το οποίο καλύπτει την αρχιτεκτονική μας για την αποθήκευση των δεδομένων – βάσης δεδομένων και σύστημα διαχείρισης αρχείων.

Στο AnyPlace διαχειρίζονται τριών ειδών δεδομένα:

1. Η πληροφορία χαρτογράφησης
2. Τα πλακάκια για τα σχέδια ορόφων (floor blueprint map tiles)
3. Τελικός ραδιοχάρτης (Radio Map)

Αυτά μπορούν να χωριστούν σε δύο κατηγορίες ανάλογα με τον τρόπο αποθήκευσης τους:

1. Τα δεδομένα που έχουν συγκριμένη δομή και μορφή και μοντελοποιούνται σε αντικείμενα-κείμενα JSON – *Δεδομένα Ι*

2. Τα δεδομένα που βρίσκονται σε δυαδική αναπαράσταση χωρίς συγκεκριμένη μορφή (binary files) – Δεδομένα 2,3

```
{
  "coordinates_lat": "52.52816874290241",
  "address": "-",
  "description": "IPSN 2014 Venue",
  "name": "Andel's Hotel Berlin",
  "buid": "building_02e4ebd2-413e-4b27-b3bf-6f7e2ca3640f_1397488807470",
  "coordinates_lon": "13.457137495279312",
  "is_published": "true",
  "url": "-",
  "geometry": {
    "coordinates": [
      52.52816874290241,
      13.457137495279312
    ],
    "type": "Point"
  }
}

{
  "timestamp": "1380292556560",
  "floor": "0",
  "rss": "-89",
  "MAC": "00:02:6f:75:75:a3",
  "heading": "179.0",
  "y": "33.04121721535921",
  "x": "34.67211790256155",
  "geometry": {
    "type": "Point",
    "coordinates": [
      34.67211790256155,
      33.04121721535921
    ]
  }
}
```

Εικόνα 6.1 : Αναπαράσταση σε JSON document ενός κτιρίου (αριστερά) και ενός αποτυπώματος RSS (δεξιά)

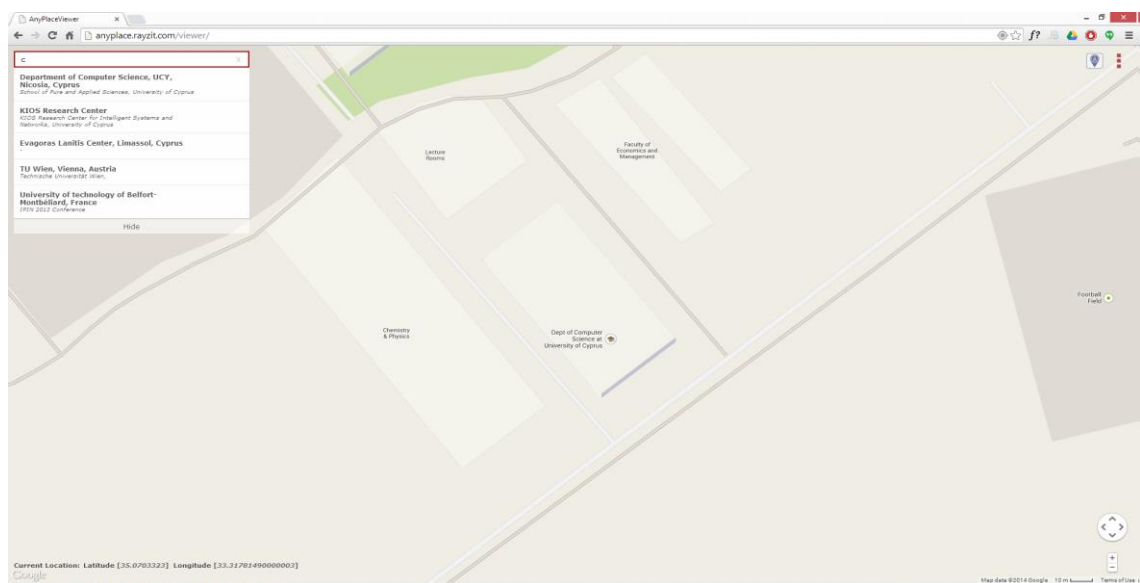
Τα δεδομένα 1 αποτελούνται από τις πληροφορίες που καταχωρούνται στον AnyPlace Architect, δηλαδή μέτα-πληροφορίες για τα κτίρια, τους ορόφους και τα σημεία ενδιαφέροντος όπου μοντελοποιούνται σε JSON αντικείμενα, *εικόνα 6.1*, αλλά και τα αποτυπώματα που καταχωρούνται από τον AnyPlace Logger, αφού το κάθε αποτύπωμα μοντελοποιείται σε ένα JSON αντικείμενο, *εικόνα 6.1*.

Λόγω της ιδιότητας ότι τα δεδομένα 1 αποτελούν κείμενα JSON αλλά και λόγω του ότι σε πολλά από τα αιτήματα που υποστηρίζει το AnyPlace API χρειάζεται κάποιο φιλτράρισμα βάση γεωγραφικών συντεταγμένων, το σύστημα βάσης δεδομένων που χρησιμοποιούμε στο AnyPlace είναι το Couchbase Server με έκδοση 2 ή νεότερη. Η Couchbase χρησιμοποιεί JSON αντικείμενα ως κύρια μονάδα αποθήκευσης της έτσι αποτελεί ιδανική για τον σκοπό μας. Επίσης, υποστηρίζει αιτήματα γεωχωρικής φύσεως (geospatial queries) χωρίς ειδική διαμόρφωση, κάτι που κάνει την ανταπόκριση στα συγκεκριμένα αιτήματα από το AnyPlace πιο γρήγορη. Επιπλέον, όπως είπαμε και πριν, η γραμμική επεκτασιμότητα και η επίδοση που παρέχει η Couchbase καθώς διαχειρίζεται τεράστιο όγκο δεδομένων, η οποία επεκτείνεται ακόμα περισσότερο με την χρήση του Memcache [45] το οποίο παρέχει δυνατότητα γρήγορης ανάκτησης των πιο πρόσφατων αντικείμενων αφού τα κρατά στη μνήμη, αποτελούν ιδανική και ικανοποιητική λύση στις απαιτήσεις μας.

Το δεύτερο είδος δεδομένων που διαχειρίζεται το AnyPlace είναι τα δεδομένα 2,3 τα οποία βασικά είναι συνηθισμένα αρχεία δυαδικής μορφής (binary files). Όσο αφορά τα floor tiles είναι πολύ μικρά αρχεία που το μέγεθος κάθε tile κυμαίνεται από μερικά bytes σε λιγότερο από 1MB, είναι εικόνες διαστάσεων 256 x 256 pixels. Από την εμπειρία μας, μέσω των κτιρίων που χαρτογραφήθηκαν μέχρι τώρα στο σύστημα, τα tiles είναι μικρότερα των 100KB ενώ ολόκληρο το σχέδιο ορόφου πριν την επεξεργασία δεν ξεπερνά τα 10MB. Ο τελικός ραδιοχάρτης είναι αρχεία απλού ASCII κειμένου των οποίων το μέγεθος εξαρτάται από το πλήθος των αποτυπωμάτων που υπάρχουν για τον εκάστοτε όροφο, το οποίο μπορεί να ελεγχθεί από το module LPRadioScale (κεφάλαιο 4).

Για την αποθήκευση και διαχείριση αυτών των δεδομένων σκεφτήκαμε είτε κάποια λύση κατανεμημένης βάσης που επιτρέπει αποθήκευση δυαδικών αρχείων, είτε ένα ολοκληρωμένο κατανεμημένο σύστημα διαχείρισης αρχείων. Για καλύτερη επεκτασιμότητα, αλλά και εγγύηση ότι θα μπορεί το σύστημα μας να δουλέψει με ραδιοχάρτες οποιουδήποτε μεγέθους επιλέξαμε το δεύτερο, σύστημα διαχείρισης αρχείων, αν και κάποιες βάσεις δεδομένων όπως MongoDB [46] και HyperTable [47] είχαν καλή προοπτική.

Επιλέξαμε να χρησιμοποιήσουμε το *GlusterFS*, ένα ανοικτού κώδικα κατανεμημένο σύστημα διαχείρισης αρχείων που διατηρείται από την Red Hat, Inc. Είναι ένα επεκτάσιμο σύστημα NAS (Network-attached storage filesystem) που δημιουργεί ένα



Εικόνα 6.2 : Αναζήτηση κτιρίου

παράλληλο δίκτυο συνενώνοντας πολλούς εξυπηρετητές μέσω του πρωτοκόλλου TCP/IP ή Infiniband RDMA, με αποτέλεσμα να δημιουργεί ένα ενιαίο σύστημα διαχείρισης αρχείων που παρέχεται κάτω από το ίδιο δέντρο (single global namespace) και επιτρέπει στους χρήστες να έχουν πρόσβαση στα αρχεία που διαχειρίζεται απλά με το να κάνουν mount στο σύστημα τους το σύστημα GlusterFS.

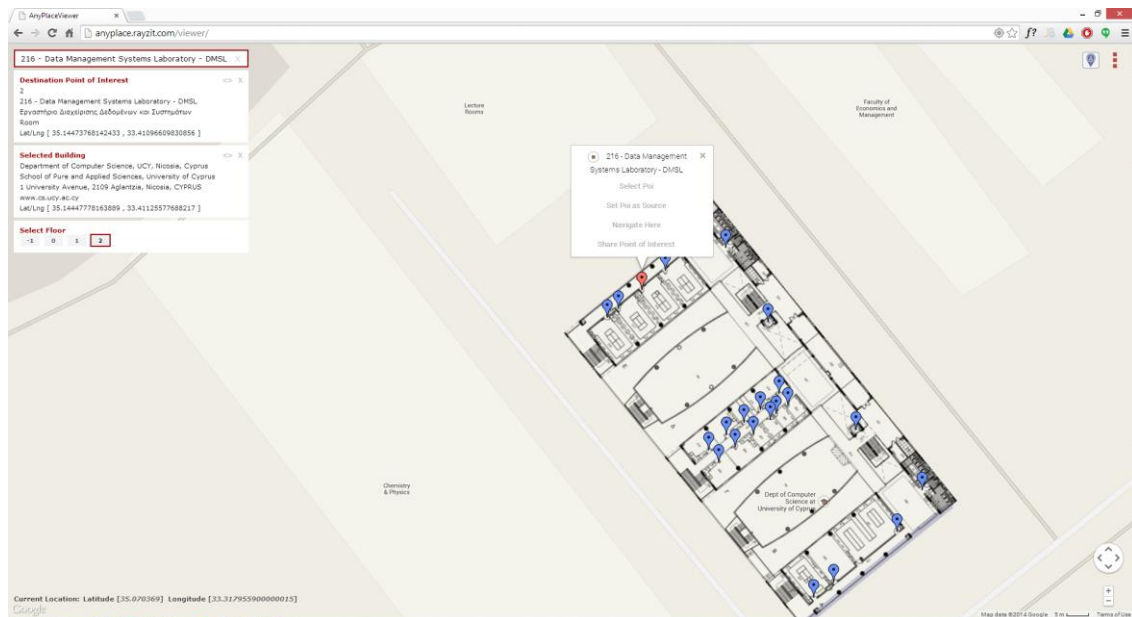
6.3 AnyPlace Viewer

Όπως αναφέρθηκε και προηγουμένως, ο AnyPlace Viewer είναι η κύρια εφαρμογή για τους χρήστες που θέλουν να χρησιμοποιήσουν και να δοκιμάσουν την υπηρεσία AnyPlace μέσω ενός περιηγητή (HTML5-compliance required).

Ο Viewer παρέχει τις πλείστες από τις λειτουργίες που παρέχει το AnyPlace μέσα από μια εύχρηστη και απλή διεπαφή που μοιάζει στην χρήση με άλλα συστήματα χαρτών που υπάρχουν σήμερα όπως Google Maps και Open Street Maps [48], με στόχο την άμεση εξοικείωση των χρηστών με την υπηρεσία.

Όπως βλέπετε στην *εικόνα 6.2* το μοναδικό μέσο εισαγωγής που υπάρχει στην εφαρμογή είναι ένα κουτί αναζήτησης (πάνω αριστερά).

Αρχικά, ο χρήστης μπορεί να αναζητήσει γράφοντας μια γραμματοσειρά το κτίριο στο οποίο θέλει να περιηγηθεί. Αφού επιλέξει από τη λίστα αποτελεσμάτων, *εικόνα 6.2*, το κτίριο επιθυμίας του, αυτόματα γίνεται ανάληψη όλων των πληροφοριών του κτιρίου συμπεριλαμβανομένων των ορόφων και των σημείων ενδιαφέροντος. Γίνεται αυτόματη επιλογή του πιο ψηλού ορόφου (η σύγκριση γίνεται βάση αλφαριθμητικής σύγκρισης του αριθμού του ορόφου) και εμφανίζεται στον χάρτη το αρχιτεκτονικό σχέδιο του μαζί με όλα τα σημεία ενδιαφέροντος του επιλεγμένου ορόφου, *εικόνα 6.3*.

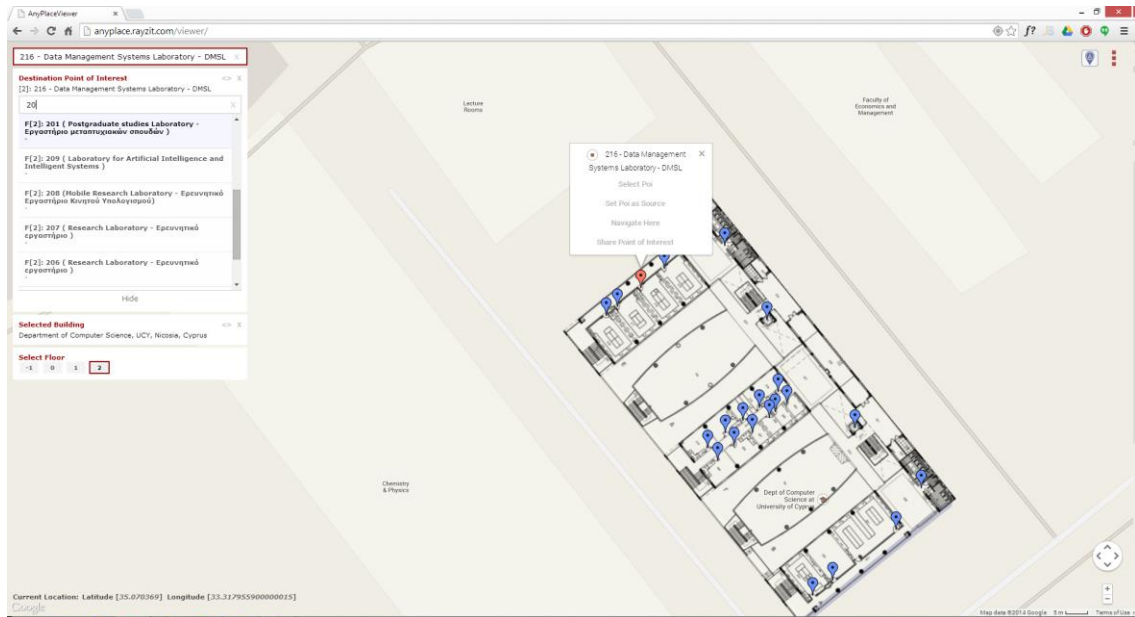


Εικόνα 6.3 : Ανεπτυγμένες λεπτομέρειες επιλεγμένου κτιρίου και ορόφου (πάνω αριστερά) ενώ βλέπουμε και τα σημεία ενδιαφέροντος του ορόφου (POIs)

Εάν υπάρχει επιλεγμένο κτίριο, οποιαδήποτε αναζήτηση αφορά σημεία ενδιαφέροντος. Η αναζήτηση ελέγχει και εμφανίζει αποτελέσματα των POI (Point of Interest – σημείο ενδιαφέροντος) τα οποία περιέχουν την γραμματοσειρά αναζήτησης είτε στον τίτλο, είτε στο όνομα είτε στην περιγραφή τους, παρέχοντας έτσι ένα εύκολο τρόπο εύρεσης του POI που θέλει ο χρήστης.

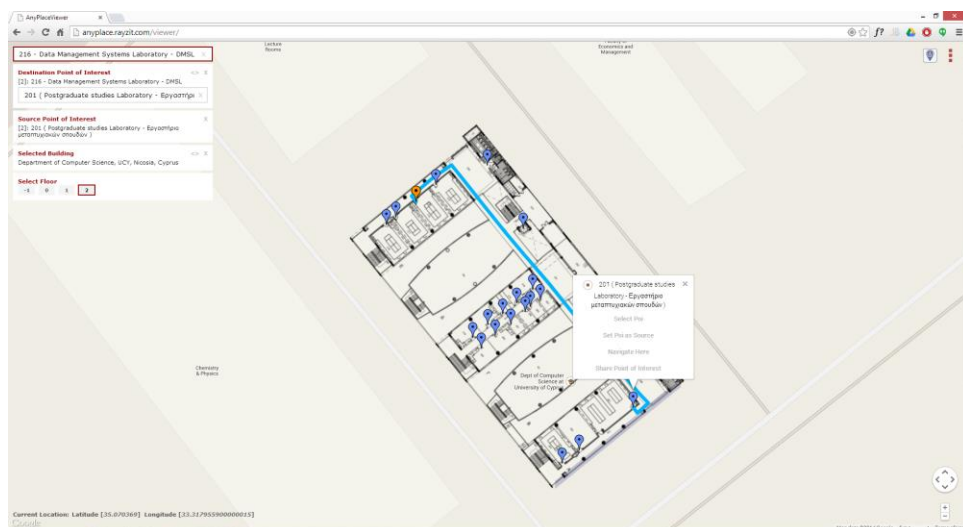
Εκτός από την περιήγηση μέσα στο κτίριο και την προβολή διάφορων πληροφοριών για τον επιλεγμένο όροφο και κτίριο (πληροφορίες πάνω αριστερά κάτω από το κουτί αναζήτησης) ο χρήστης μπορεί να πάρει οδηγίες πλοήγησης ή αλλιώς δρομολόγησης από ένα συγκεκριμένο σημείο ενδιαφέροντος σε άλλο, ή και ακόμα από οποιαδήποτε τοποθεσία πάνω στον χάρτη σε κάποιο POI.

Αρχικά γίνεται μέσω του κύριου κουτιού αναζήτησης η επιλογή του σημείου ενδιαφέροντος που θέλουμε να πάμε-προορισμός, *εικόνα 6.4*. Επιλογή ενός POI μπορεί να γίνει είτε όπως αναφέρθηκε προηγουμένως μέσω αναζήτησης είτε με την επιλογή του POI στο χάρτη και μετά την επιλογή «**Select Poi**».



Εικόνα 6.4 : Μετά από αναζήτηση και επιλογή σημείου ως προορισμός (πάνω κουτί αναζήτησης) βλέπουμε την αναζήτηση σημείου αφετηρίας (κάτω κουτί αναζήτησης)

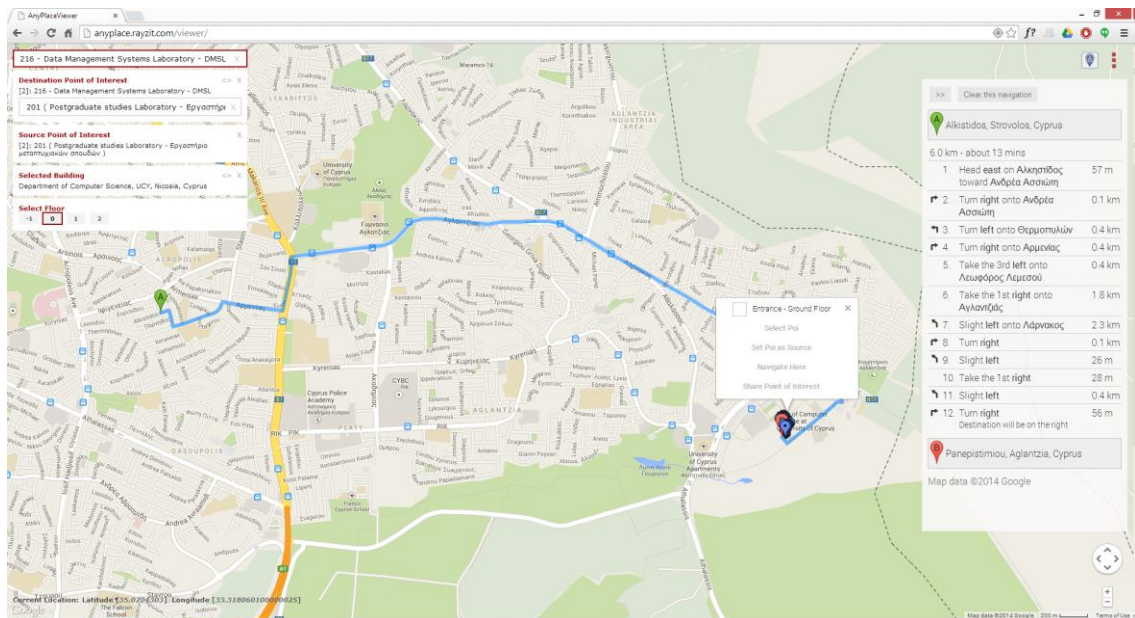
Έπειτα, εμφανίζεται ένα άλλο κουτί αναζήτησης κάτω από το κύριο, *εικόνα 6.4*, στην θέση με την πληροφορία για το επιλεγμένο POI. Η αναζήτηση σημείου ενδιαφέροντος μέσω εκείνου του search box επιλέγει το σημείο αφετηρίας (source POI) και αμέσως γίνεται κάλεσμα στο AnyPlace API για ανάληψη οδηγιών δρομολόγησης, από το σημείο αφετηρία στο σημείο προορισμός. Με την ολοκλήρωση του αιτήματος εμφανίζεται στον χάρτη η δρομολόγηση, *εικόνα 6.5*.



Εικόνα 6.5 : Προβολή οδηγιών δρομολόγησης εντός κτιρίου

Επιλέγοντας ένα σημείο ενδιαφέροντος από τον χάρτη υπάρχει η δυνατότητα να αλλάξει είτε η αφετηρία είτε ο προορισμός και να τεθεί σαν νέα τιμή το POI που επιλέχθηκε και αυτόματα θα αλλάξει και η δρομολόγηση εάν υπάρχει.

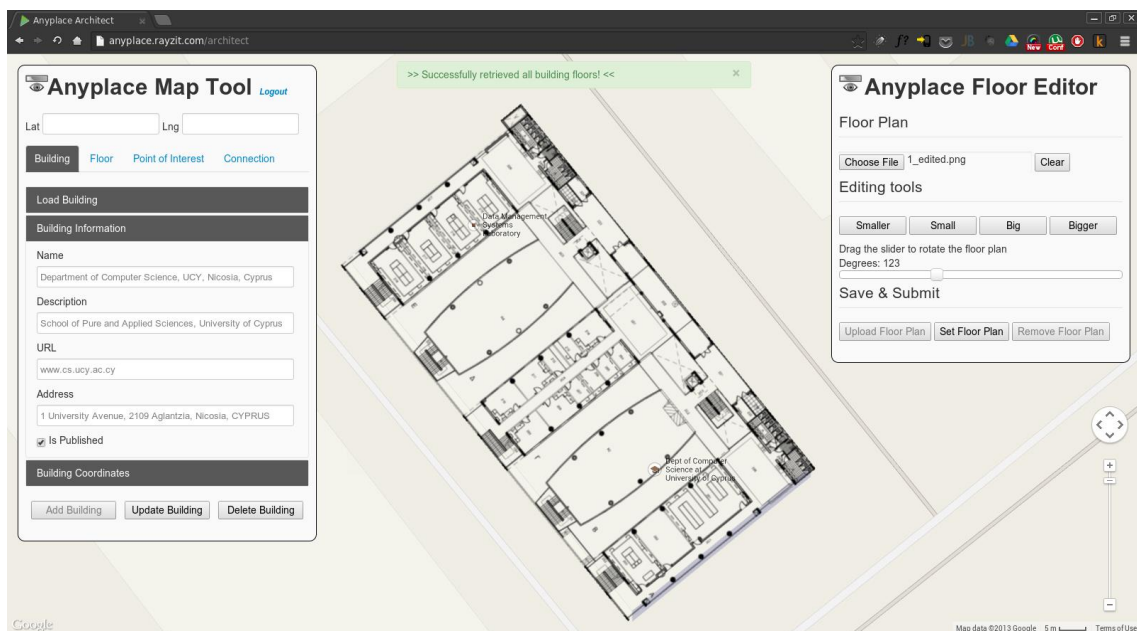
Κάνοντας δεξί κλικ – right click πάνω στον χάρτη μακριά έξω από το κτίριο υπάρχει η επιλογή «**Navigate from Here**», η οποία ζητά από το Google Directions API [49] οδηγίες δρομολόγησης από το σημείο που έγινε το δεξί κλικ μέχρι το κτίριο που είναι επιλεγμένο. Μετά, λόγω του ότι η δρομολόγηση της Google γίνεται σε επίπεδο δρόμων, η υπηρεσία AnyPlace θα συνεχίσει τη δρομολόγηση από την είσοδο του κτιρίου στο σημείο ενδιαφέροντος – προορισμός. Μαζί με την σχεδίαση της δρομολόγησης πάνω στον χάρτη εμφανίζονται και οι οδηγίες κειμένου στο πλάι δεξιά ώστε να μπορεί ο χρήστης να δει αναλυτικά την διαδρομή προς το κτίριο, *εικόνα 6.6*.



Εικόνα 6.6 : Προβολή οδηγιών δρομολόγησης με τη χρήση του Google Directions API σε συνδυασμό με το AnyPlace API. Δεξιά φαίνονται και λεκτικές οδηγίες.

Τέλος, μια άλλη λειτουργία που πρέπει να αναφέρουμε είναι ότι ανά πάσα στιγμή ο χρήστης μπορεί να μοιραστεί με άλλους τις οδηγίες δρομολόγησης ή ένα σημείο ενδιαφέροντος μέσω των ειδικά διαμορφωμένων υπέρ-συνδέσμων που παράγει ο Viewer. Εάν κάποιος χρήστης ανοίξει ένα AnyPlace σύνδεσμο τότε βλέπει τις πληροφορίες ακριβώς όπως ήταν στον χρήστη που δημιούργησε τον σύνδεσμο.

Για παράδειγμα, εάν είμαι καθηγητής στο Πανεπιστήμιο Κύπρου και θέλω να ενημερώσω τους φοιτητές μου για την ακριβή θέση του γραφείου μου, μπορώ να αναζητήσω το κτίριο του πανεπιστημίου και ακολούθως το σημείο ενδιαφέροντος που αντιπροσωπεύει το γραφείο μου. Εάν δημιουργήσω ένα υπέρ-σύνδεσμο και τον στείλω στους μαθητές μου θα μπορούν να δουν πάνω στον χάρτη αμέσως που είναι το γραφείο μου αφού με το που θα ανοίξουν τον σύνδεσμο, θα φορτωθεί ο χάρτης του ανάλογου ορόφου του πανεπιστημίου και θα επιλεγεί το γραφείο μου.



Εικόνα 6.7 : AnyPlace Architect - Floor Plan Editor

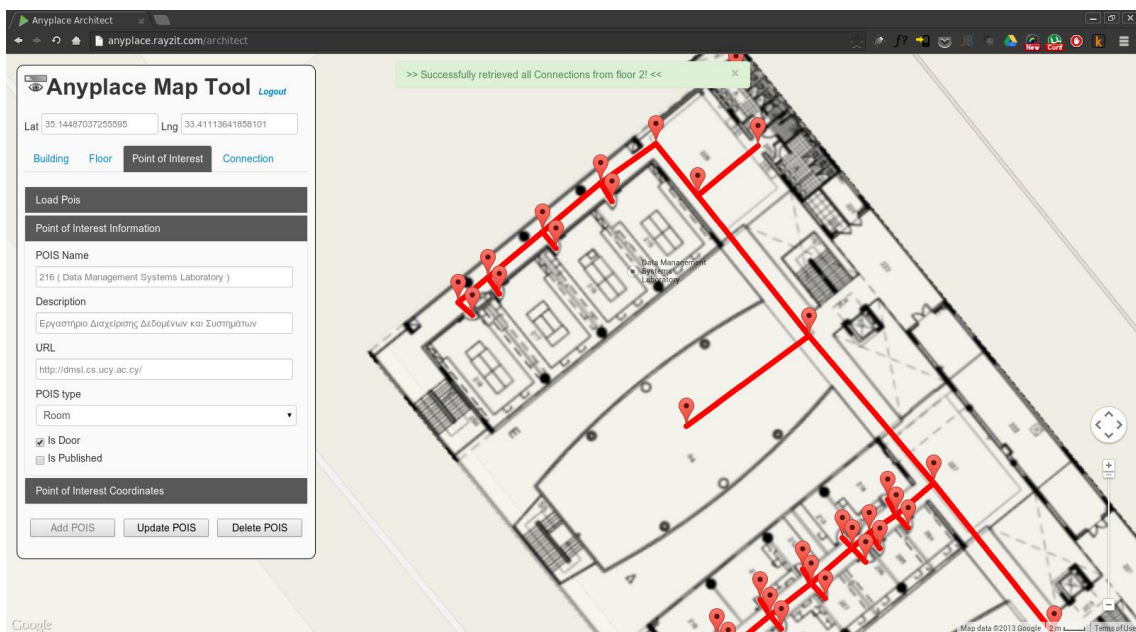
6.4 AnyPlace Architect

Η εφαρμογή AnyPlace Architect είναι η κύρια εφαρμογή χαρτογράφησης των κτιρίων. Μέσω του Architect μπορεί ο χρήστης να δημιουργήσει νέο κτίριο και να δηλώσει την θέση του. Ακολούθως, εισάγει τις πληροφορίες για τους ορόφους και ανεβάζει τα αρχιτεκτονικά σχέδια μέσω του ειδικού εργαλείου που παρέχεται από τον Architect το οποίο επιτρέπει την σμίκρυνση, μεγέθυνση και περιστροφή του σχεδίου ώστε να ταιριάζει όσο γίνεται με τους χάρτες της Google, εικόνα 6.7.

Αφού τοποθετηθεί και το σχέδιο ορόφου, πλέον ο χρήστης μπορεί να σημειώσει και να δώσει πληροφορίες για μερικά σημεία ενδιαφέροντος που έχουν σημασία μέσα στο κτίριο, όπως γραφεία, τουαλέτες, πόρτες εισόδου κλπ.

Τέλος, αφού σημειωθούν τα POI που πρέπει εντός του κτιρίου, μπορεί ο χρήστης να συνενώσει τα σημεία με συνδέσεις οι οποίες στην συνέχεια θα χρησιμοποιηθούν από το Navigation Module για να ανευρεθεί δρομολόγηση από το ένα σημείο στο άλλο, *εικόνα 6.8*.

Όλες αυτές οι πληροφορίες εισάγονται στο σύστημα μέσω της εξαιρετικά απλής διεπαφής του Architect στην οποία βασικά υπάρχουν 4 tabs, ένα για κάθε στοιχείο της χαρτογράφησης – κτίριο (building), όροφος (floor), σημεία ενδιαφέροντος (poi), συνδέσεις (connections) και ανάλογα σε πιο από αυτά βρίσκεται ο χρήστης μπορεί να κάνει και τις ανάλογες αλλαγές ή προσθήκες.



Εικόνα 6.8 : AnyPlace Architect - συνένωση σημείων ενδιαφέροντος

6.5 AnyPlace Developers

Η εφαρμογή AnyPlace Developers είναι μια πολύ απλή εφαρμογή η οποία έχει χρήση σαν σημείο αναφοράς για το AnyPlace API.

Βρίσκονται όλα τα αιτήματα στα οποία μπορεί να απαντήσει το AnyPlace API με τις παραμέτρους που πρέπει να σταλούν. Επίσης, υπάρχουν τα κατάλληλα πεδία για κάθε αίτημα με τις παραμέτρους του ώστε κάποιος προγραμματιστής ή κάποιος χρήστης που επιθυμεί να δει τον τρόπο λειτουργίας του AnyPlace API να μπορεί να εκτελέσει τα αιτήματα απευθείας από την εφαρμογή και να δει την απάντηση του API σε μια απλή και εργονομική φόρμα μαζί με τον χρόνο ανταπόκρισης.

The screenshot displays the AnyPlace Developers API documentation page. The page is organized into sections: Global Parameters, Navigation, and Mapping. Each section contains a table of API endpoints with their respective methods, names, URLs, and descriptions.

Global Parameters — These parameters will globally override method parameters if changed.

Parameter	Value	Type	Description
access_token	api_tester	text	OAuth2 Access Token

Navigation

Get navigation information between Points of Interest using AnyPlace services. [Show All](#)

Method	Endpoint Name	URL	Description
POST	POI Details	/navigation/pois/id	Get Point of interest details
POST	Navigation Route Coordinates-to-Poi	/navigation/route_xy	Get Navigation instructions from a given location to a POI
POST	Navigation Route Poi-to-Poi	/navigation/route	Get Navigation instructions between 2 POIs

Mapping

Getters | Mapping information and API calls to add building / floor / POI / connection. [Show All](#)

Method	Endpoint Name	URL	Description
POST	World Buildings	/mapping/building/all	Get all annotated buildings
POST	NearBy Buildings	/mapping/building/coordinates	Get annotated buildings near you (50 meters radius)
POST	All building floors	/mapping/floor/all	Get all floors of a building
POST	POIs by building	/mapping/pois/all_building	Get all POIs inside a building

Εικόνα 6.9 : AnyPlace Developers - όλα τα API endpoints κλειστά

Κεφάλαιο 7

Πειραματική Αποτίμηση Συστήματος

7.1	Μεθοδολογία	76
7.2	Αριθμός εμφανίσεων ενός σημείου πρόσβασης στη συλλογή μετρήσεων	77
7.3	Αριθμός μοναδικών τιμών ένδειξης ισχύς για κάθε σημείο πρόσβασης	79
7.4	Τυπική απόκλιση των ενδείξεων ισχύς κάθε σημείου πρόσβασης	80
7.5	Σύγκριση όλων των φίλτρων	83

Σε αυτή την ενότητα θα ασχοληθούμε με την πειραματική αξιολόγηση του συστήματος AnyPlace. Το σύστημα στο σύνολο του έχει παρουσιαστεί ήδη σε δύο διεθνή συνέδρια, IPIN'13 και IPSN'13, όπου έγιναν demos είτε των εφαρμογών στα κινητά είτε των διαδικτυακών εφαρμογών με σχετική επιτυχία. Γι' αυτό τον λόγο εδώ θα ασχοληθούμε με την αποτίμηση του αλγόριθμου *LPRadioScale*.

7.1 Μεθοδολογία

Τα δεδομένα που χρησιμοποιήσαμε για την αποτίμηση και αξιολόγηση του αλγόριθμου αφορούν το κτίριο Πληροφορικής στο Πανεπιστήμιο Κύπρου και συγκεκριμένα τον 2^ο όροφο για τον οποίο είχαμε τα περισσότερα αποτυπώματα.

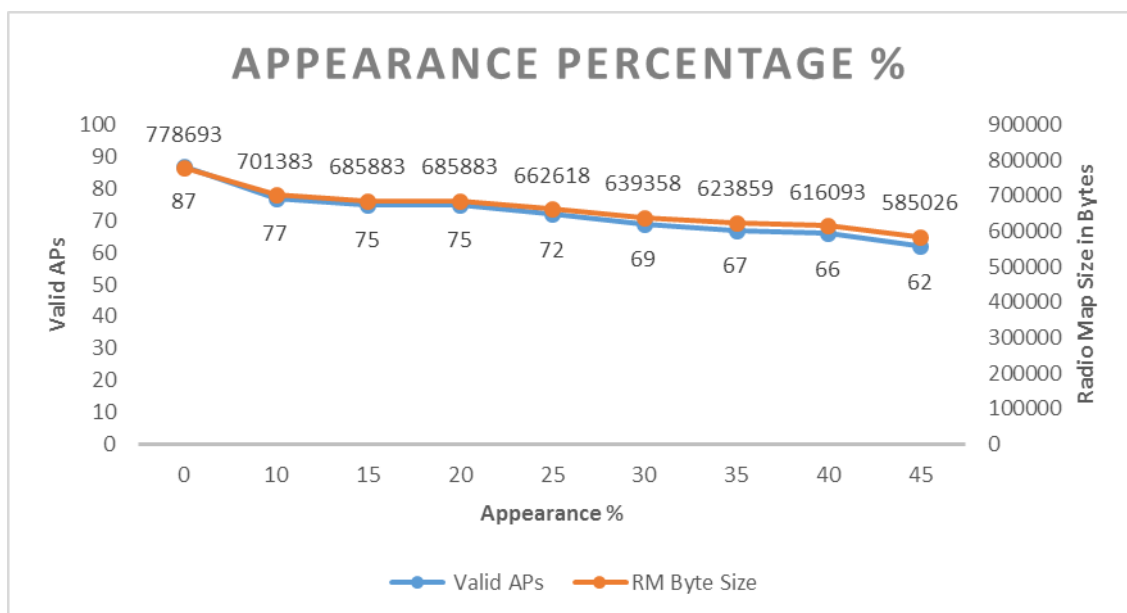
Χρησιμοποιήσαμε ένα *RSS-log* το οποίο έχει συνολικά 108138 ενδείξεις RSS (συνδυασμό αποτυπώματος, AP διεύθυνση MAC και dB ένδειξη ισχύς του σήματος)

που συλλέχθηκαν από 7910 σημεία στον χώρο, δηλαδή διαφορετικά αποτυπώματα. Συνολικά ο όροφος έχει διάσπαρτα ακανόνιστης κατανομής 87 σημεία πρόσβασης, δηλαδή έχουμε 87 διαφορετικά MACs. Η συλλογή όλων των μετρήσεων έγινε με την χρήση του AnyPlace Logger, κεφάλαιο 5.2 κατά τη διάρκεια 2 ημερών μεταξύ 10:00 και 21:00. Για την πειράματα μας ο αλγόριθμος έτρεχε σε ένα σύμπλεγμα 8 υπολογιστικών μονάδων που έτρεχαν Hadoop στην έκδοση 0.20.

Παρακάτω θα συγκρίνουμε τον τρόπο που επηρεάζουν οι τρεις παράμετροι που δίνονται στον αλγόριθμο όσο αφορά το φιλτράρισμα των σημείων πρόσβασης ξεχωριστά και ακολούθως θα δούμε την συμπεριφορά τους όταν συνδυάζονται. Οι τιμές που παρουσιάζονται σε κάθε πείραμα εδώ επιλέγηκαν μετά από εξαντλητικές δοκιμές και αποτελούν τις κύριες τιμές που είχαν ουσιαστική συνεισφορά.

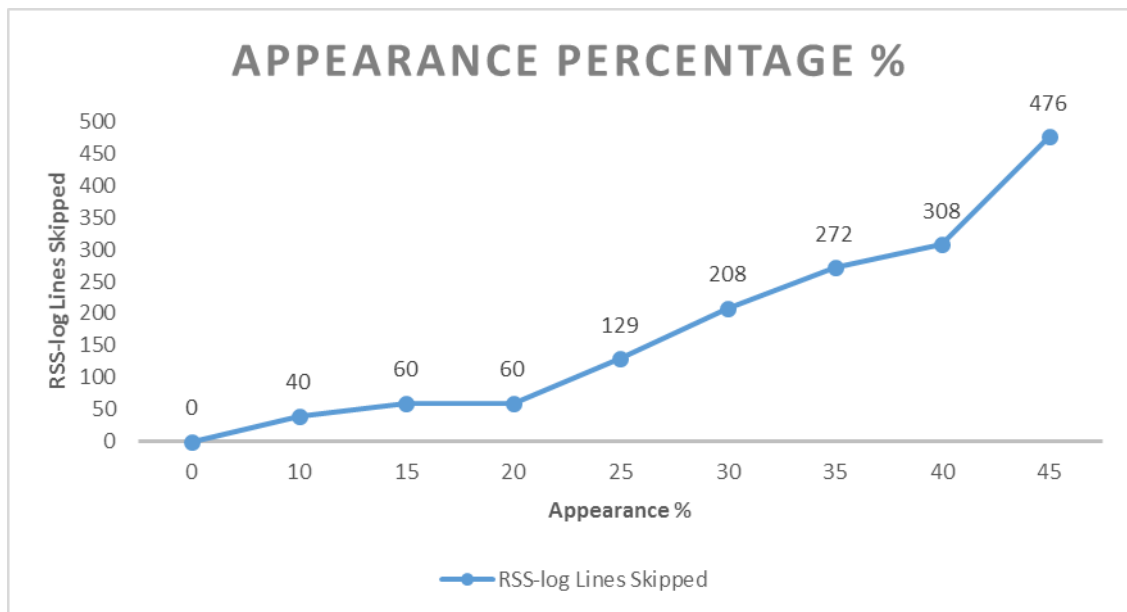
7.2 Αριθμός εμφανίσεων ενός σημείου πρόσβασης στη συλλογή μετρήσεων

Στην ενότητα αυτή θα δούμε πως επηρεάζονται τα σημεία πρόσβασης που καταλήγουν στον παραγόμενο ραδιοχάρτη χρησιμοποιώντας μόνο τον αριθμό εμφανίσεων κάθε σημείου πρόσβασης στις μετρήσεις μας. Δηλαδή, καθορίζεται ένα κριτήριο *THRESHOLD_APPEARANCES* το οποίο χρησιμοποιείται για να απορρίπτει σημεία πρόσβασης τα οποία δεν εμφανίστηκαν σε ποσοστό αποτυπωμάτων μεγαλύτερο του.



Εικόνα 7.1 : Έγκυρα σημεία πρόσβασης (αριστερός άξονας) και Μέγεθος ραδιοχάρτη (δεξιός άξονας) με τη χρήση φίλτρου 'Ποσοστό εμφανίσεων'

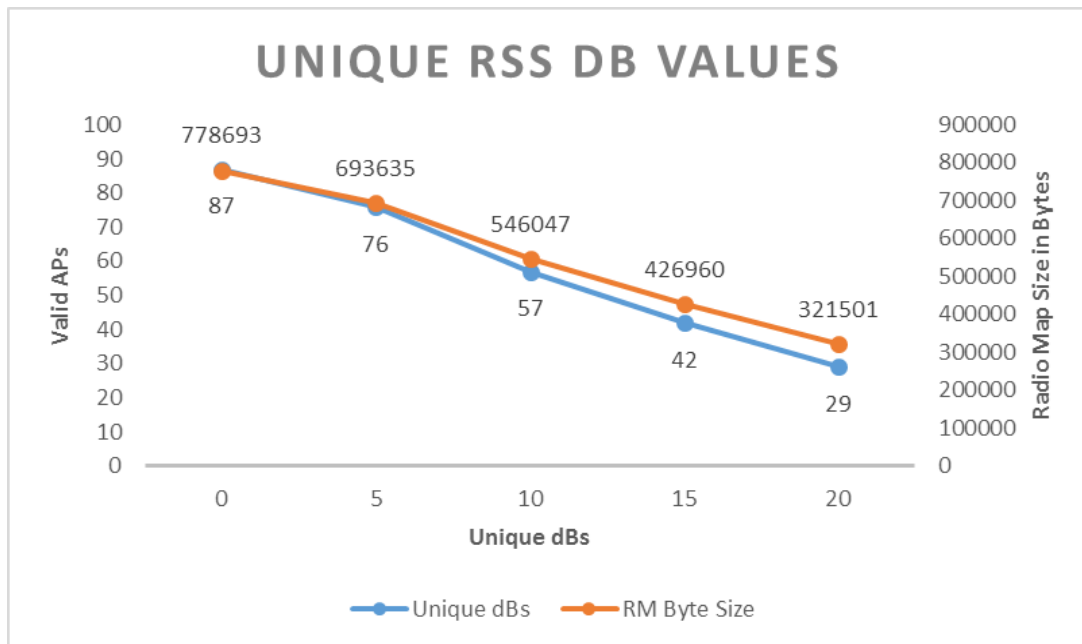
Όπως φαίνεται από την *εικόνα 7.1* τα έγκυρα σημεία πρόσβασης μετά το φιλτράρισμα αυτό μειώνονται αλλά σε πολύ αργούς ρυθμούς. Στην *εικόνα 7.2*, βλέπουμε ότι υπάρχει μια μικρή αύξηση στον αριθμό των γραμμών που αγνοούνται από το RSS-log αλλά δεν είναι μεγάλη αν λάβουμε υπόψη ότι έχουμε σύνολο 108138 εγγραφές. Επιπλέον, αφού δεν μειώνονται δραστικά τα σημεία πρόσβασης δεν μειώνεται ούτε το τελικό μέγεθος του ραδιοχάρτη.



Εικόνα 7.2 : Γραμμές που προσπεράστηκαν λόγω μη έγκυρου AP – φίλτρο 1

7.3 Αριθμός μοναδικών τιμών ένδειξης ισχύς για κάθε σημείο πρόσβασης

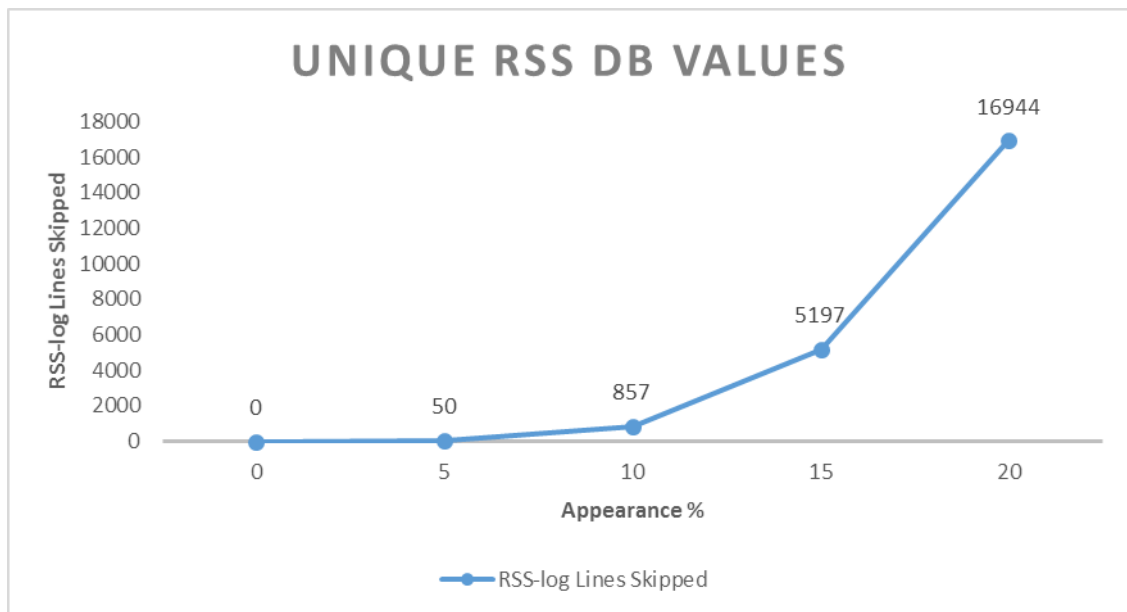
Το δεύτερο φίλτρο που εφαρμόζουμε είναι το κριτήριο των διαφορετικών τιμών της ένδειξης ισχύς του κάθε σημείου πρόσβασης. Δηλαδή, μας ενδιαφέρει ένα σημείο πρόσβασης να έχει πολλές διαφορετικές τιμές ένδειξης έτσι ώστε να αποτελεί «καλό» AP το οποίο θα έχει βοηθά μετέπειτα στην ανίχνευση του χάρτη. Ένα σημείο πρόσβασης το οποίο δεν έχει πολλές διαφορετικές τιμές δεν βοηθά ιδιαίτερα αφού σε όσα αποτυπώματα βρίσκεται δεν υπάρχει ξεκάθαρος διαχωρισμός της θέσης άρα δεν χρειάζεται να υπάρχει στον ραδιοχάρτη.



Εικόνα 7.3 : Έγκυρα σημεία πρόσβασης (αριστερός άξονας) και Μέγεθος ραδιοχάρτη (δεξιός άξονας) με τη χρήση φίλτρου 'Αριθμός μοναδικών ενδείξεων ισχύς'

Εδώ βλέπουμε ξεκάθαρα ότι το αυτό το φίλτρο έχει πολύ μεγαλύτερο αποτέλεσμα από το προηγούμενο αφού από 87 σημεία πρόσβασης φτάσαμε τα 29 όταν θέσαμε το κριτήριο *THRESHOLD_UNIQUE_DBs* ίσο με 20. Ανάλογη είναι και η μείωση του μεγέθους του ραδιοχάρτη αφού το μέγεθος του έφτασε το 50% του αρχικού.

Στην *εικόνα 7.4* μπορούμε να δούμε ότι η αύξηση των εγγραφών που αγνοούμε από το RSS-log καθώς αυξάνουμε το *THRESHOLD_UNIQUE_DBs* είναι εκθετική και έχει όντως αποτέλεσμα. Ακόμα και έτσι όμως φτάσαμε τα 29 εναπομείναντα σημεία πρόσβασης με το να θέσουμε το φίλτρο μας ίσο με 20 αλλά μόνο 16000 γραμμές προσπεράστηκαν.



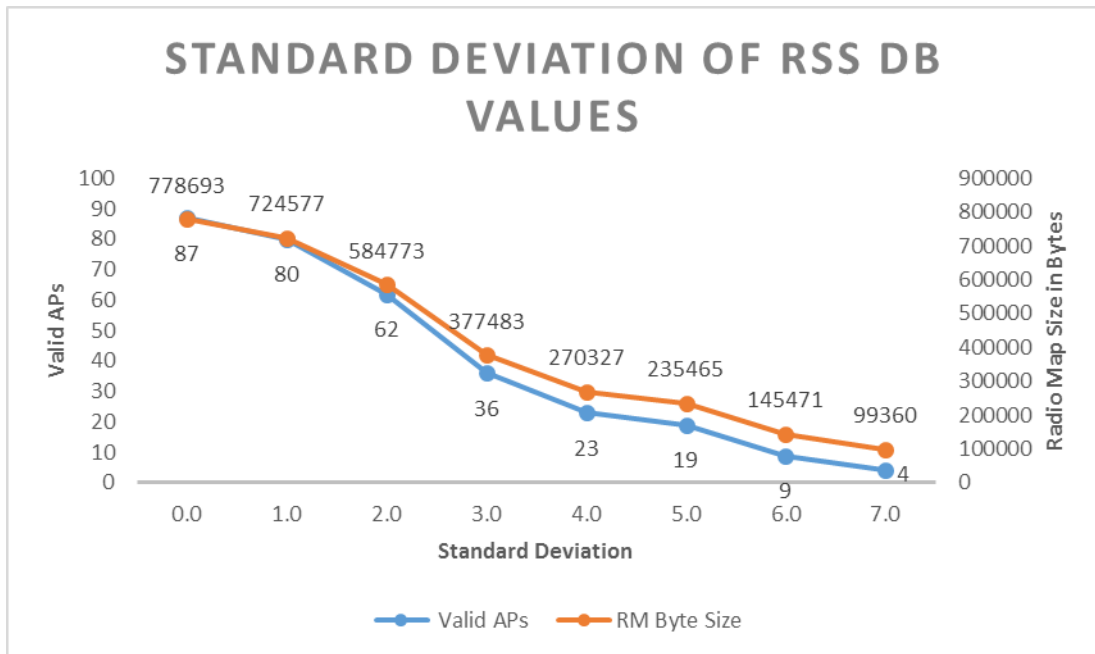
Εικόνα 7.4 : Γραμμές που προσπεράστηκαν λόγω μη έγκυρου AP – φίλτρο 2

7.4 Τυπική απόκλιση των ενδείξεων ισχύς κάθε σημείου πρόσβασης

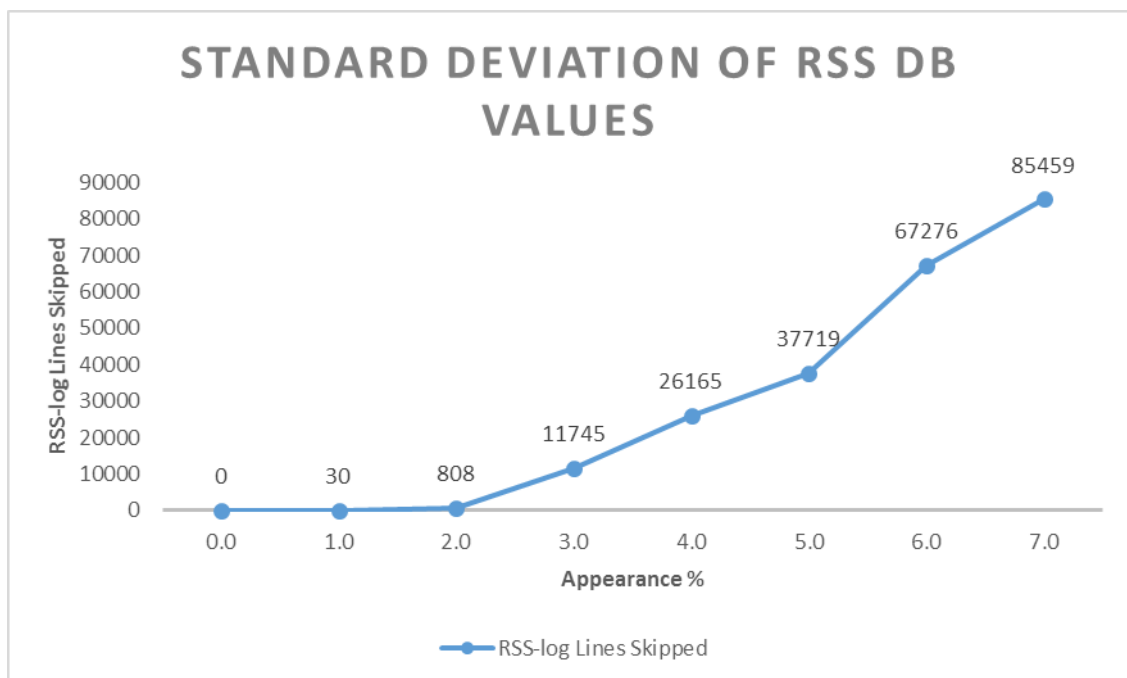
Το τελευταίο φίλτρο που εφαρμόζει ο *LPRadioScale* χρησιμοποιεί ξανά τις ενδείξεις ισχύς του σήματος κάθε σημείου πρόσβασης αλλά αυτή τη φορά θέλουμε η τυπική απόκλιση των ενδείξεων αυτών να είναι μεγαλύτερη του κριτηρίου *THRESHOLD_STDEV*. Το φίλτρο αυτό βασίζεται στο γεγονός ότι κάποιο σημείο πρόσβασης θα έχει θετική επιρροή στον υπολογισμό της θέσης του χρήστη εάν έχει αρκετά διάσπαρτη κατανομή ενδείξεων έτσι ώστε να αποτελεί σημείο πρόσβασης ορόσημο στους αλγορίθμους τοποθέτησης.

Όπως δείχνει η παράσταση της εικόνας 7.5 αυτό το φίλτρο έχει την μεγαλύτερη αποτελεσματικότητα στην μείωση των σημείων πρόσβασης που θα παραμείνουν στον ραδιοχάρτη. Αξίζει να σημειωθεί ότι με κριτήριο τυπικής απόκλισης 7,0 τα έγκυρα σημεία πρόσβασης έμειναν να είναι μόνο 4. Αυτό δεν είναι καλό και πρέπει να γίνει εκτενέστερη ανάλυση των τιμών του φίλτρου αυτού στο μέλλον περισσότερο σε συνδυασμό και με τα υπόλοιπα έτσι ώστε να έχουμε τον αριθμό σημείων πρόσβασης σε λογικά επίπεδα. Εάν φτάσει πολύ χαμηλά επίπεδα είναι χειρότερο από το να μην εφαρμόζαμε καθόλου το φίλτρο αφού η εκτίμηση της θέσης του χρήστη θα έχει τεράστια απόκλιση από την πραγματική. Στην εικόνα 7.6 παρατηρούμε όπως θα περιμέναμε ότι οι γραμμές που αγνοούνται από το RSS-log φτάνουν το 80% όλων των

εγγραφών, που ουσιαστικά μπορούμε να καταλάβουμε ότι αυτά τα 4 σημεία πρόσβασης που παρέμειναν εμφανίστηκαν σχεδόν σε όλα τα αποτυπώματα.

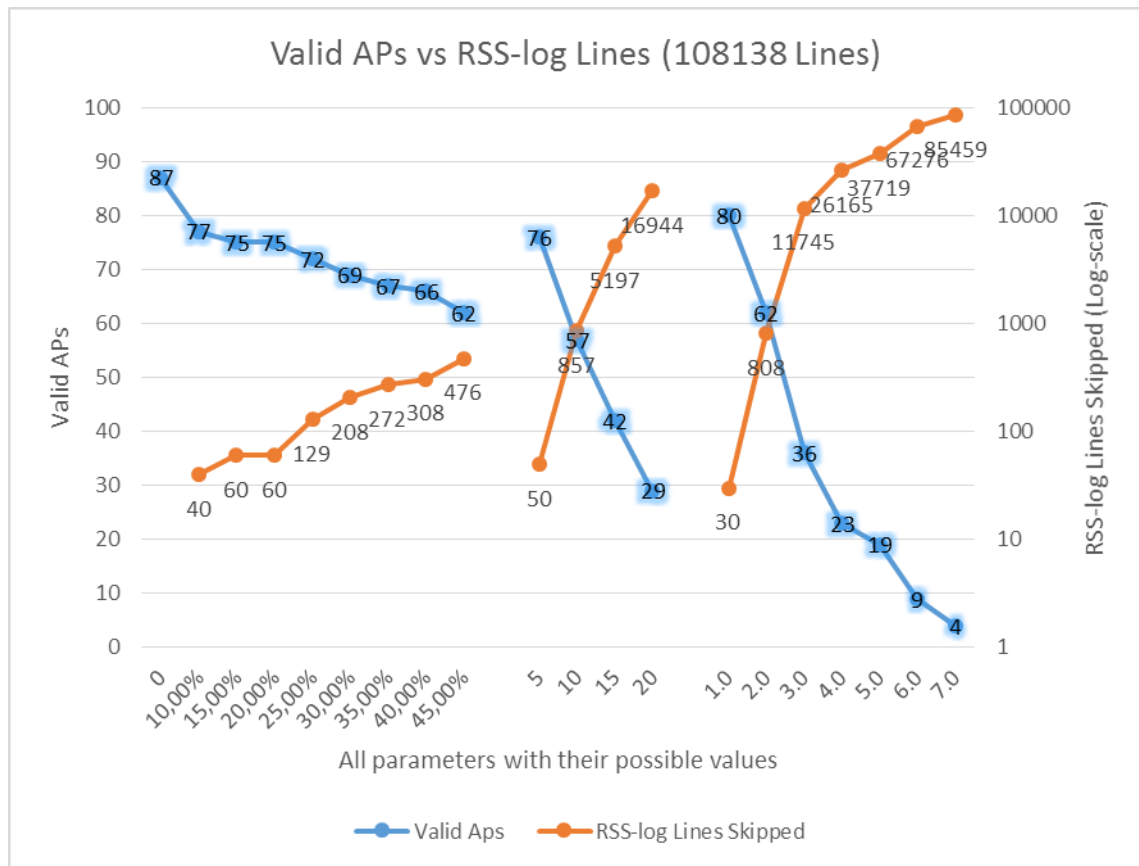


Εικόνα 7.5 : Έγκυρα σημεία πρόσβασης (αριστερός άξονας) και Μέγεθος ραδιοχάρτη (δεξιός άξονας) με τη χρήση φίλτρου 'Τυπική απόκλιση ενδείξεων ισχύς'



Εικόνα 7.6 : Γραμμές που προσπεράστηκαν λόγω μη έγκυρου AP – φίλτρο 3

7.5 Σύγκριση όλων των φίλτρων



Εικόνα 7.7 : Σύνοψη όλων των φίλτρων σχετικά με έγκυρα APs και γραμμές στο RSS-log που αγνοούνται

Εδώ βλέπουμε όλα τα φίλτρα και όλες τις δυνατές τιμές τους μαζί στην ίδια παράσταση. Είναι εμφανές το πόσο επηρεάζει το κάθε φίλτρο τον αριθμό των έγκυρων σημείων πρόσβασης αλλά και τις γραμμές που παραλείπονται από το RSS-log.

Το φίλτρο 1 έχει μικρή σχετικά επιρροή αλλά μπορεί να αφαιρέσει κάποια σημεία πρόσβασης που είναι εντελώς αχρείαστα και δεν συμβάλουν ουσιαστικά στον ραδιοχάρτη. Ακολουθεί το φίλτρο 2 με αρκετά καλό φιλτράρισμα και τέλος το φίλτρο 3 που ανάλογα τις τιμές του κριτηρίου που έχουμε μπορεί να φτάσει στο ανεπιθύμητο και να παράγει ραδιοχάρτη που θα οδηγεί σε εκτιμήσεις θέσης με τεράστια απόκλιση.

Είναι σημαντικό να εκτιμηθεί η διαφορά που υπάρχει μεταξύ των φίλτρων. Το κάθε ένα απορρίπτει διαφορετικά σημεία πρόσβασης με αποτέλεσμα να απορρίπτονται διαφορετικές ενδείξεις κάθε φορά. Επίσης ο αριθμός των σημείων πρόσβασης που απορρίπτονται δεν είναι ανάλογος με το πόσες ενδείξεις θα απορριφθούν αφού ένα AP

μεγάλης εμβέλειας σε έκταση, λογικά θα έχει περισσότερες εμφανίσεις, συνεπώς θα αγνοηθούν περισσότερες γραμμές από ένα AP που εισακούετε μόνο σε 2-3 σημεία.

Κεφάλαιο 8

Συμπεράσματα και Μελλοντικές Επεκτάσεις

8.1	Συμπεράσματα	84
8.2	Μελλοντικές επεκτάσεις	85

8.1 Συμπεράσματα

Σε αυτή την διπλωματική εργασία παρουσιάστηκε το AnyPlace, ένα ολοκληρωμένο σύστημα γεωτοποθέτησης και γεωπλοήγησης σε εσωτερικούς χώρους με υποδομή υποστήριξης για Μεγάλα Δεδομένα. Σχεδιάστηκε και αναπτύχθηκε από κάτω προς τα πάνω ολόκληρη η υποδομή παρέχοντας μια αρχιτεκτονική που μπορεί να επεκταθεί περαιτέρω με την προσθήκη διαφόρων συστατικών που θα επεκτείνουν ή θα βελτιώσουν τις λειτουργίες του.

Είναι εμφανές ότι πλέον φτάσαμε την εποχή όπου ο κάθε ένας από εμάς φέρει μαζί του έξυπνες συσκευές, συγκεκριμένα οι πλείστοι από εμάς πλέον έχουν τα λεγόμενα έξυπνα κινητά, smartphones. Αυτή η εξάπλωση των μικρών αυτών υπολογιστικών μονάδων έχει ωθήσει με εξωπραγματικές ταχύτητες προς ανάπτυξη και εξέλιξη όλους τους τομείς της Πληροφορικής και των επιστημών γενικότερα. Πολλές υπηρεσίες έχουν εδραιωθεί οι οποίες χρησιμοποιούν την εκάστοτε τοποθεσία του χρήστη είτε για να του παρέχουν βοήθεια και πληροφορίες για κοντινά σημεία ενδιαφέροντος είτε για να μεταβάλουν κάποιες πληροφορίες που του εμφανίζουν και να συνάδουν με την τοποθεσία του. Η τοποθεσία του χρήστη πλέον είναι ένα από τα μείζον στοιχεία που κάθε υπηρεσία θέλει να χρησιμοποιεί και η πλοήγηση και τοποθέτηση είναι ένας από

τους τομείς που εισήχθη από τις αρχές της Πληροφορικής. Δεν ήταν όμως μέχρι πρόσφατα που άρχισε να μελετιέται η τοποθέτηση και η δρομολόγηση του χρήστη εντός κτιρίων, λόγω ανεπάρκειας της τεχνολογίας και των υπάρχων υπολογιστικών μονάδων που είτε το μέγεθος τους δεν επέτρεπε την μεταφορά τους παντού ή η υπολογιστική τους δύναμη ήταν ανεπαρκής.

Το AnyPlace έρχεται δυναμικά σαν απόγονος ενός άλλου βραβευμένου συστήματος, Airplace, με πολλές όμως προσθήκες τόσο σε λειτουργίες όσο και σε αρχιτεκτονική. Οι διάφορες παρουσιάσεις που έγιναν σε διεθνή συνέδρια έδειξαν ότι υπάρχει προσοδοφόρο έδαφος στον τομέα αυτό και υπάρχουν πολλές προοπτικές ανέλιξης.

Οι διαδικτυακές εφαρμογές AnyPlace Viewer και Architect, οι δύο εφαρμογές για έξυπνα κινητά Android AnyPlace Navigator και Logger σε συνδυασμό με τον αλγόριθμο που αναπτύχθηκε, LPRadioScale, αποτελούν μια καλή αρχή και βάση ενός ολοκληρωμένου συστήματος τοποθέτησης και πλοήγησης σε εσωτερικούς χώρους και αποτελεί το έναυσμα για περαιτέρω ανέλιξη και επέκταση.

8.2 Μελλοντικές επεκτάσεις

Αυτή τη στιγμή το AnyPlace είναι σε καλό σημείο από άποψη λειτουργιών και αρχιτεκτονικής, πρέπει όμως να γίνει κάποια επίβλεψη του και δοκιμή του σε επίπεδο χρήστη και εργονομίας του ώστε πλέον να φύγει από την ερευνητική περιοχή και να γίνει διαθέσιμο για πρόσβαση από τον οποιοδήποτε, αφού εκτός από τον AnyPlace Viewer, δεν μπορεί κάποιος να αποκτήσει πρόσβαση.

Επίσης, ο AnyPlace Navigator μπορεί να επεκταθεί περισσότερο ώστε να γίνει ακόμα πιο αυτόματη η διαδικασία εύρεσης και φόρτωσης του κατάλληλου κτιρίου και ορόφου τόσο στην έναρξη της εφαρμογής όσο και κατά την πλοήγηση και χρήση της εφαρμογής στη συνέχεια καθώς κινείται ο χρήστης (αλλαγή ορόφου και κτιρίου αυτόματα).

Επιπλέον, ο Architect μπορεί να γίνει πιο φιλικός προς τον χρήστη ή και να επεκταθεί με περισσότερες λειτουργίες που θα επιτρέπουν στον χρήστη να χαρτογραφήσει με

περισσότερη ακρίβεια τα κτίρια και να συνενώσει σημεία ενδιαφέροντος στο εσωτερικό τους.

Εκτός από το θέμα εμφάνισης και ευχρηστίας των εφαρμογών, σημαντικό συστατικό του συστήματος είναι ο LPRadioScale ο οποίος δίνει στο AnyPlace την δυνατότητα επεξεργασίας μεγάλων δεδομένων κατά τη δημιουργία του ραδιοχάρτη, αφού έχει αναπτυχθεί έτσι ώστε να μην χρησιμοποιεί μνήμη περισσότερη από όση χρειάζεται για να κρατά τα σημεία πρόσβασης του κτιρίου, ενώ αντίθετα αλγόριθμοι που χειρίζονται όλα τα αποτυπώματα μαζί αντιμετωπίζουν προβλήματα σε τεράστιους ραδιοχάρτες.

Είναι πολύ σημαντικό και καίριο να μελετηθεί περαιτέρω και πιο αναλυτικά η επιρροή των διάφορων φίλτρων και να γίνει εκτενής ανάλυση των διαφορετικών τιμών των κριτηρίων αλλά και συνδυασμός τους ώστε να επιτυγχάνεται η μείωση του μεγέθους ραδιοχάρτη, χωρίς ωστόσο να μειώνεται η ακρίβεια της τοποθέτησης θέσης του χρήστη. Κάτι το οποίο δεν μελετήθηκε σε αυτή την διπλωματική εργασία.

Βιβλιογραφία

- [1] “Strategy Analytics.” [Online]. Available: <http://www.strategyanalytics.com>.
- [2] P. Bahl and V. N. Padmanabhan, “RADAR: an in-building RF-based user location and tracking system,” 2000, pp. 775–784.
- [3] J. Dean and S. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters,” presented at the OSDI’04: Sixth Symposium on Operating System Design and Implementation, 2004.
- [4] “Hadoop Homepage.” [Online]. Available: <http://hadoop.apache.org/>.
- [5] “Android Homepage.” [Online]. Available: <http://www.android.com/>.
- [6] M. Ghaddar, L. Talbi, and T. A. Denidni, “Human body modelling for prediction of effect of people on indoor propagation channel,” *Electron. Lett.*, vol. 40, no. 25, p. 1592, 2004.
- [7] J. Ryckaert, P. De Doncker, R. Meys, A. de Le Hoye, and S. Donnay, “Channel model for wireless communication around human body,” *Electron. Lett.*, vol. 40, no. 9, pp. 543–544, Apr. 2004.
- [8] J. Park, D. Curtis, S. Teller, and J. Ledlie, “Implications of device diversity for organic localization,” in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 3182–3190.
- [9] Y. Kim, H. Shin, and H. Cha, “Smartphone-based Wi-Fi pedestrian-tracking system tolerating the RSS variance problem,” in *2012 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2012, pp. 11–19.
- [10] R. Want, A. Hopper, V. Falcao, and J. Gibbons, “The active badge location system,” *ACM Trans. Inf. Syst. TOIS*, vol. 10, no. 1, pp. 91–102, 1992.
- [11] R. Azuma, “Tracking Requirements for Augmented Reality,” *Commun ACM*, vol. 36, no. 7, pp. 50–51, Jul. 1993.
- [12] B. Gergely, S. Diez Martinez, T. Jelle, and J. Krogstie, “Navigating MazeMap: indoor human mobility, spatio-logical ties and future potential,” 2014.
- [13] “Download MazeMap.” [Online]. Available: : <https://mazemap.com/what-it-is>.
- [14] C. Laoudias, G. Constantinou, M. Constantinides, S. Nicolaou, D. Zeinalipour-Yazti, and C. G. Panayiotou, “The airplace indoor positioning platform for android smartphones,” in *Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, 2012, pp. 312–315.
- [15] Y. Kim, Y. Chon, H. Cha, M. Ji, and S. Park, “Scalable and Consistent Radio Map Management Scheme for Participatory Sensing-based Wi-Fi Fingerprinting.”

- [16] M. Ester, H. Kriegel, J. S, and X. Xu, “A density-based algorithm for discovering clusters in large spatial databases with noise,” 1996, pp. 226–231.
- [17] S. Eisa, J. Peixoto, F. Meneses, and A. Moreira, “Removing useless APs and fingerprints from WiFi indoor positioning radio maps,” 2013.
- [18] “Amazon Web Services.” [Online]. Available: <https://aws.amazon.com/>.
- [19] “Google Cloud Platform.” [Online]. Available: <https://cloud.google.com/>.
- [20] “Message Passing Interface - MPI: http://en.wikipedia.org/wiki/Message_Passing_Interface.”.
- [21] “Map Reduce Slides.” [Online]. Available: <http://research.google.com/archive/mapreduce-osdi04-slides/index-auto-0004.html>.
- [22] “Play framework.” [Online]. Available: <http://www.playframework.com/>.
- [23] “JSON Homepage.” [Online]. Available: <http://www.json.org/>.
- [24] “Couchbase.” [Online]. Available: <http://www.couchbase.com/>.
- [25] “GlusterFS.” [Online]. Available: <http://www.gluster.org/>.
- [26] “Dart Homepage.” [Online]. Available: <https://www.dartlang.org/>.
- [27] “AngularDart Homepage.” [Online]. Available: <https://angulardart.org/>.
- [28] “AngularJS Homepage.” [Online]. Available: <https://angularjs.org/>.
- [29] S. Ghemawat, H. Gobioff, and S.-T. Leung, “The Google file system,” in *ACM SIGOPS Operating Systems Review*, 2003, vol. 37, pp. 29–43.
- [30] “Hadoop in IBM.” [Online]. Available: <http://www-01.ibm.com/software/data/infosphere/hadoop/>.
- [31] “Hadoop explanation from Information Week.” [Online]. Available: <http://www.informationweek.com/big-data/software-platforms/how-to-explain-hadoop-to-non-geeks/d/d-id/899721>.
- [32] “Hadoop slides - Worcester Polytechnic Institute.” [Online]. Available: <http://web.cs.wpi.edu/~cs561/s12/Lectures/6/Hadoop.pdf>.
- [33] C. Laoudias, P. Kemppi, and C. G. Panayiotou, “Localization Using Radial Basis Function Networks and Signal Trench Fingerprints in WLAN,” in *Proceedings of the 28th IEEE Conference on Global Telecommunications*, Piscataway, NJ, USA, 2009, pp. 3805–3810.
- [34] “Google Places API.” [Online]. Available: <https://developers.google.com/places/documentation/>.

- [35] “MVC: Model-View-Controller.” [Online]. Available:
<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>.
- [36] “What is Java.” [Online]. Available:
http://java.com/en/download/whatis_java.jsp.
- [37] “Scala Language.” [Online]. Available: <http://www.scala-lang.org/>.
- [38] “LinkedIn Engineering - Play Framework,” *The Play Framework at LinkedIn*.
[Online]. Available: <http://engineering.linkedin.com/play/play-framework-linkedin>.
- [39] “Akka.” [Online]. Available: [http://en.wikipedia.org/wiki/Akka_\(toolkit\)](http://en.wikipedia.org/wiki/Akka_(toolkit)).
- [40] “Python Homepage.” [Online]. Available: <https://www.python.org/>.
- [41] “Bash Scripting.” [Online]. Available: <http://www.gnu.org/software/bash/>.
- [42] “ImageMagick Homepage.” [Online]. Available: <http://www.imagemagick.org/>.
- [43] “AdvPNG Homepage.” [Online]. Available:
<http://advancemame.sourceforge.net/doc-advpng.html>.
- [44] “Dijkstra’s Algorithm.” [Online]. Available:
http://en.wikipedia.org/wiki/Dijkstra's_algorithm.
- [45] “Memcached Homepage.” [Online]. Available: <http://memcached.org/>.
- [46] “MongoDB Homepage.” [Online]. Available: <http://www.mongodb.org/>.
- [47] “HyperTable Database Homepage.” [Online]. Available: <http://hypertable.org/>.
- [48] “Open Street Maps.” [Online]. Available: <http://www.openstreetmap.org/>.
- [49] “Google Directions API.” [Online]. Available:
<https://developers.google.com/maps/documentation/directions/>.