

Ατομική Διπλωματική Εργασία

**ΤΕΧΝΙΚΕΣ ΠΡΟΦΟΡΤΩΣΗΣ ΡΑΔΙΟΧΑΡΤΩΝ ΓΙΑ
ΓΕΩΤΟΠΟΘΕΤΗΣΗ ΣΕ ΕΣΩΤΕΡΙΚΟΥΣ ΧΩΡΟΥΣ**

Γιάννης Ευαγόρου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2014

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Τεχνικές Προφόρτωσης Ραδιοχαρτών για Γεωτοποθέτηση σε Εσωτερικούς Χώρους

Γιάννης Ευαγόρου

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 θέλω να ευχαριστήσω τον Δρ. Δημήτρη Ζεϊναλιπούρ που μου έδωσε την ευκαιρία να συμμετέχω ως ενεργό μέλος και βοηθός του ερευνητικού εργαστηρίου DMSL. Τα πολύτιμα εφόδια που απέκτησα κατά την διάρκεια της παρουσίας μου στο DMSL, θα με βοηθήσουν να ανταπεξέλθω σε ερευνητικό και σε επαγγελματικό επίπεδο, κατά την διάρκεια της σταδιοδρομίας μου στην Πληροφορική.

Ακόμη θα ήθελα να ευχαριστήσω τους διδακτορικούς φοιτητές του εργαστηρίου για τις άριστες συμβουλές πάνω σε τεχνητά ζητήματα.

Επίσης, θέλω να ευχαριστήσω την οικογένεια μου για όλη την οικονομική στήριξη κατά την διάρκεια των σπουδών μου. Τέλος θέλω να ευχαριστήσω τους φίλους μου και την Σωτηρούλα για την συντροφιά τους κατά την διάρκεια της διπλωματικής μου εργασίας και των σπουδών μου γενικότερα.

Περίληψη

Η πλέον πιο διαδεδομένη τεχνολογία Γεωτοποθέτησης GPS αποδεικνύεται μη επαρκής για εσωτερικούς χώρους, καθώς το σήμα των δορυφόρων εξασθενεί εντός των κτιρίων. Αυτό καθιστά την χρήση των συστημάτων GPS σε εσωτερικούς χώρους, ακατάλληλη. Για αυτό τον λόγο η κοινότητα έχει προτείνει μια πλειάδα από νέες τεχνολογίες, εναλλακτικές των συστημάτων GPS που προσφέρουν Γεωτοποθέτηση υψηλής ακρίβειας σε εσωτερικούς χώρους.

Μία αξιοσημείωτη τεχνολογία γεωτοποθέτησης και αυτή που θα μελετηθεί σε αυτή την διπλωματική εργασία, είναι η τεχνική της γεωτοποθέτησης με την χρήση Αποτυπωμάτων Τιμών Ισχύος. Τα αποτυπώματα αυτά φυλάσσονται στον αυτό-καλούμενο Ραδιοχάρτη τον οποίο πρέπει να ανακτήσει ο χρήστης πριν την πλοήγηση του. Η τεχνολογία αυτή παρέχει υψηλής ακρίβειας γεωτοποθέτηση χρησιμοποιώντας την υφιστάμενη υποδομή και προ-συλλεγμένα χωρικά δεδομένα του κτιρίου. Συνήθως τα δεδομένα αυτά φυλάσσονται σε μια βάση δεδομένων που βρίσκεται σε κάποιο εξυπηρετητή. Ο εξυπηρετητής συγκρίνει το τρέχον αποτύπωμα του χρήστη με τα προ-συλλεγμένα χωρικά δεδομένα του κτιρίου, υπολογίζοντας με αυτό τον τρόπο την τοποθεσία του χρήστη. Πολλές φορές ο χρήστης γεωτοποθετείται σε περιβάλλον όπου η σύνδεση του με τον εξυπηρετητή είναι διακοπτόμενη. Αυτό αποτελεί σημαντικό εμπόδιο στην γεωτοποθέτηση του χρήστη αφού δεν είναι σε θέση να φορτώσει τους Ραδιοχάρτες (με τις τιμές ισχύος του κτιρίου) στην κινητή συσκευή του και να πλοηγηθεί. Για την επίλυση του πιο πάνω προβλήματος υποθέτουμε ότι ο Ραδιοχάρτης ενός κτιρίου διαμελίζεται σε μικρότερους ραδιοχάρτες (οι αυτόκαλούμενοι Μερικοί Ραδιοχάρτες). Η συνεισφορά αυτής της Διπλωματικής Εργασίας είναι ο ορισμός νέων αλγορίθμων Προφόρτωσης (Prefetching Algorithms) των Ραδιοχαρτών οι οποίοι θα επιτρέπουν σε ένα χρήστη να πλοηγείται με μεγαλύτερη επιτυχία σε ένα περιβάλλον διακοπτόμενης σύνδεσης. Στόχος λοιπόν είναι η γεωτοποθέτηση με την χρήση μερικών Ραδιοχαρτών αντί ολόκληρου του Ραδιοχάρτη του κτιρίου και παράλληλα ελαχιστοποίηση του σφάλματος γεωτοποθέτησης, σε ένα περιβάλλον διακοπτόμενης σύνδεσης στο Διαδίκτυο.

Περιεχόμενα

1	Εισαγωγή	1
1.1	Γενικά	1
1.1.1	Νέα εποχή με τις έξυπνες κινητές συσκευές	1
1.1.2	Πρόβλημα Γεωτοποθέτησης και Τεχνολογίες Γεωτοποθέτησης σε Εσωτερικά Περιβάλλοντα	3
1.2	Πρόταση Αλγορίθμου Προφόρτωσης των Ραδιοχαρτών	6
1.3	Πειραματική Σύνοψη	8
1.4	Συνεισφορά Ατομικής Διπλωματικής Εργασίας	9
1.5	Διάγραμμα Ατομικής Διπλωματικής Εργασίας	11
2	Σχετική Εργασία	12
2.1	Γεωτοποθέτηση Σε Εσωτερικούς Χώρους	12
2.2	Προφόρτωση στο Διαδίκτυο	13
2.2.1	Μηχανή Προβλέψεων στο Διαδίκτυο	14
2.2.2	Μηχανή Προφόρτωσης στο Διαδίκτυο	15
2.2.3	Αλγόριθμος Πρόβλεψης Dependency Graph	15
2.3	Αλγόριθμος PageRank	16
2.3.1	Τυπικός Ορισμός του PageRank	17
2.3.2	Αναπλήρωση Παγιδευμένης Σημαντικότητας (Rank Sinks)	17
2.3.3	Παράγοντας Απόσβεσης (Damping Factor)	18
2.4	Αλγόριθμος Ομαδοποίησης BFR	19

3	Διατύπωση του Προβλήματος και Μοντέλο Συστήματος	22
3.1	Μοντέλο Συστήματος	22
3.1.1	Γράφημα Εξαρτήσεων - Κατασκευή και Εκπαίδευση	24
3.2	Διατύπωση του Προβλήματος	27
4	Αλγόριθμοι Προφόρτωσης Ραδιοχαρτών	28
4.1	Community Based Algorithm - CBA	29
4.1.1	Επεξήγηση Αλγορίθμου CBA	29
4.1.2	Ανάλυση Πολυπλοκότητας Αλγορίθμου CBA	31
4.2	Building Based Algorithm - BBA	33
4.2.1	Επεξήγηση Αλγορίθμου BBA	33
4.2.2	Ανάλυση Πολυπλοκότητας Αλγορίθμου BBA	38
4.3	Αλγόριθμος Μηχανής Πρόβλεψης	38
4.3.1	Επεξήγηση Αλγορίθμου	38
4.3.2	Αλγόριθμος Μηχανής Πρόβλεψης-Ανάλυση Πολυπλοκότητας	42
4.4	Αλγόριθμος Μηχανής Προφόρτωσης	43
4.5	Παράδειγμα Εκτέλεσης Αλγορίθμου Πρόβλεψης και Προφόρτωσης	47
5	Πρότυπο Συστήματος	50
5.1	Τεχνολογικό Υπόβαθρο Εξυπηρετητή	51
5.1.1	Κατανεμημένο Πλαίσιο Hadoop	51
5.1.2	Κατανεμημένη Μη-Σχεσιακή Βάση Δεδομένων HBase	53
5.1.3	Εξυπηρετητής Ιστού Play Framework	55
5.2	Τεχνολογικό Υπόβαθρο Πελάτη	56
5.2.1	Ανάπτυξη λογισμικού στην πλατφόρμα Android	56
5.2.2	Διεπαφή Προγραμματισμού Εφαρμογών χαρτών Google	57
5.3	Επισκόπηση Υλοποίησης	57
5.4	Υλοποίηση Εξυπηρετητή	57
5.4.1	Μονάδα Ιστού Play Framework	58
5.4.2	Γραφική Διεπαφή Χρήστη	62
6	Πειραματική Αξιολόγηση	65
6.1	Μεθοδολογία Αξιολόγησης	66

6.1.1	Περιγραφή Σεναρίου	67
6.1.2	Δεδομένα Χαρτών Τιμών Ισχύος	68
6.1.3	Αλγόριθμοι και Μετρικές	69
6.2	Αποτελέσματα Πειραματικής Αποτίμησης	69
6.2.1	Αποτελέσματα Αξιολόγησης ως προς τον αριθμό των Bytes	69
6.2.2	Αποτελέσματα Αξιολόγησης ως προς τον χρόνο εξυπηρέτησης . .	70
6.2.3	Αποτελέσματα Αξιολόγησης ως προς την ακρίβεια της γεωτοπο- θέτησης	71
7	Συμπεράσματα και Μελλοντική Εργασία	73
7.1	Συμπεράσματα	73
7.2	Μελλοντική Εργασία	75
7.2.1	Βελτίωση Υποδομής	75
7.2.2	Αλγοριθμικές Βελτιώσεις	75
	Βιβλιογραφία	77
	Παράρτημα Α	A-1
	Παράρτημα Β	B-1

Κεφάλαιο 1

Εισαγωγή

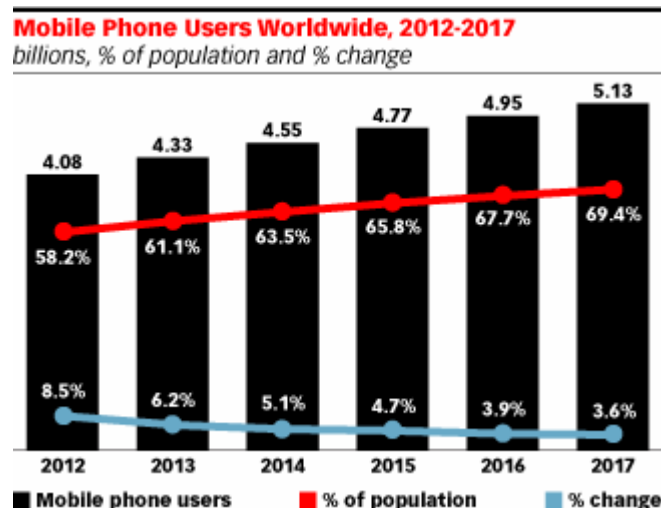
1.1	Γενικά	1
1.1.1	Νέα εποχή με τις έξυπνες κινητές συσκευές	1
1.1.2	Πρόβλημα Γεωτοποθέτησης και Τεχνολογίες Γεωτοποθέτησης σε Εσωτερικά Περιβάλλοντα	3
1.2	Πρόταση Αλγορίθμου Προφύρτωσης των Ραδιοχαρτών	6
1.3	Πειραματική Σύνοψη	8
1.4	Συνεισφορά Ατομικής Διπλωματικής Εργασίας	9
1.5	Διάγραμμα Ατομικής Διπλωματικής Εργασίας	11

1.1 Γενικά

1.1.1 Νέα εποχή με τις έξυπνες κινητές συσκευές

Αναμφισβήτητα, ο αριθμός των Κινητών Συσκευών τα τελευταία χρόνια αυξάνεται με γοργούς ρυθμούς. Σύμφωνα με μελέτη της εταιρίας *eMarketer*¹ 4.55 δισεκατομμύρια άνθρωποι παγκοσμίως θα χρησιμοποιούν κινητές συσκευές την χρονιά που διανύουμε (2014). Σύμφωνα με την ίδια έρευνα, χρήστες από αναπτυσσόμενες χώρες θα αυξήσουν περαιτέρω τα ποσοστά χρήσης των κινητών συσκευών [7]. Ο συνολικός αριθμός χρηστών κινητής συσκευής, ξεπέρασε το 2012 το 1 δισεκατομμύριο, και θα κυμαίνεται στα 1.75 δισεκατομμύρια το 2014. Οι προηγούμενες τάσεις περιγράφονται συνοπτικά στο Σχήμα 1.1.

¹ <http://goo.gl/ELBS9m>



Σχήμα 1.1: Ποσοστά χρήσης κινητών συσκευών ¹

Οι μαύρες μπάρες περιγράφουν με καθαρούς αριθμούς τον αριθμό των χρηστών που κάνουν χρήση κινητών συσκευών. Το 2012, 4.08 δισεκατομμύρια έκαναν χρήση κινητών συσκευών, ο αριθμός αυτός έχει σημειώσει σταθερή αύξηση μέχρι σήμερα και θα συνεχίσει να φανερώνει αυτή την τάση μέχρι το 2017 σύμφωνα με την έρευνα. Η κόκκινη γραμμή περιγράφει την τάση χρήσης κινητών συσκευών σε ποσοστό του παγκόσμιου πληθυσμού. Ένα εντυπωσιακό ποσοστό των 69.4% του παγκόσμιου πληθυσμού αναμένεται να κάνει χρήση κινητής συσκευής μέχρι το 2017. Παρόλα αυτά, το ενδιαφέρον μας επικεντρώνεται στις έξυπνες κινητές συσκευές των οποίων ο αριθμός ακολουθεί παρόμοιες τάσεις. Είναι γενικά αποδεκτό, ότι οι τιμές των έξυπνων κινητών συσκευών έχουν μειωθεί σημαντικά τις τελευταίες δύο δεκαετίες. Το γεγονός αυτό, αυξάνει περαιτέρω την χρήση τους.

Smartphone Users and Penetration Worldwide, 2012-2017						
	2012	2013	2014	2015	2016	2017
Smartphone Users (billions)	1.13	1.43	1.75	2.03	2.28	2.50
%change	68.4%	27.1%	22.5%	15.9%	12.3%	9.7%
%of mobile phone users	27.6%	33.0%	38.5%	42.6%	46.1%	48.8%
% of population	16.0%	20.2%	24.4%	28.0%	31.2%	33.8%

Πίνακας 1.1: Ποσοστά χρήσης Έξυπνων Κινητών Συσκευών ¹

Παρατηρώντας τον Πίνακα 1.1, είναι προφανές ότι το ποσοστό χρήσης των κινητών συσκευών θα συνεχίσει να έχει ανοδική τάση όπως και στην περίπτωση των κινητών

συσκευών γενικά. Οι ισχυρές υπολογιστικές ικανότητες σε αυτές τις συσκευές, σε συνδυασμό με τον αυξανόμενο αριθμό των αισθητήρων τους, εισήγαγαν μία άνευ προηγουμένου γκάμα από εφαρμογές (στο Google Play υπάρχουν περισσότερες από 1,200,334 εφαρμογές².)

1.1.2 Πρόβλημα Γεωτοποθέτησης και Τεχνολογίες Γεωτοποθέτησης σε Εσωτερικά Περιβάλλοντα

Είναι κοινώς αποδεκτό ότι οι άνθρωποι περνούν τον μεγαλύτερο τους χρόνο μέσα σε εσωτερικούς χώρους. Αν αναλογιστούμε την ποικιλία και τις διαστάσεις αυτών των χώρων είναι ολοένα και πιο επιθυμητή η Γεωτοποθέτηση (*Indoor Location Based Services*) μέσα σε εσωτερικούς χώρους. Για παράδειγμα οι επισκέπτες ενός μεγάλου μουσείου, έχουν την ανάγκη τεχνολογιών γεωτοποθέτησης σε εσωτερικούς χώρους, με σκοπό να εντοπίσουν τα εκθέματα τα οποία θέλουν να δουν. Επιπρόσθετα, ολοένα και περισσότερες εφαρμογές Έξυπνων Κινητών Συσκευών απαιτούν την γνώση της τοποθεσίας τους.

Δυστυχώς η πιο διαδεδομένη τεχνολογία γεωτοποθέτησης *GPS* δεν είναι κατάλληλη για εσωτερικά περιβάλλοντα. Πιο ειδικά, το δορυφορικό σήμα που χρησιμοποιεί η τεχνολογία *GPS* είναι σημαντικά εξασθενημένο μέσα σε εσωτερικούς χώρους και αρκετά ενεργοβόρο, κάτι το οποίο μειώνει την αυτονομία των συσκευών. Για αυτό τον λόγο, ένα ευρύ φάσμα τεχνολογιών (Τεχνολογίες Υπερύθρων, Bluetooth, RFID κ.α.) έχει αναπτυχθεί για την γεωτοποθέτηση μέσα σε εσωτερικά περιβάλλοντα [10]. Παρόλα αυτά οι προηγούμενες τεχνολογίες γεωτοποθέτησης απαιτούν την χρήση επιπλέον εξοπλισμού πέραν της έξυπνης κινητής συσκευής, κάτι το οποίο μειώνει την ευελιξία των χρηστών. Για την μεγαλύτερη άνεση των χρηστών η τεχνολογία γεωτοποθέτησης πρέπει να είναι πρακτική στην χρήση. Οι συσκευές γεωτοποθέτησης πρέπει να είναι μικρές, με αμελητέο βάρος και χαμηλή ενεργειακή κατανάλωση (και άρα μεγαλύτερη αυτονομία).

Σε αυτή την διπλωματική εργασία θα μελετηθούν τεχνικές που αφορούν την γεωτοποθέτηση με την χρήση αποτυπωμάτων (*Fingerprinting Positioning*). Η Γεωτοποθέτηση με την χρήση αποτυπωμάτων παρέχει υψηλής ακρίβειας γεωτοποθέτηση, χρησιμοποιώντας υφιστάμενα χωρικά δεδομένα. Η τεχνική αυτή αποτελείται από δύο φάσεις, την φάση της

²<https://www.appbrain.com/stats/number-of-android-apps>

εκτός σύνδεσης και εκπαίδευσης του συστήματος (*Offline Training Phase*) και την φάση της γεωτοποθέτησης:

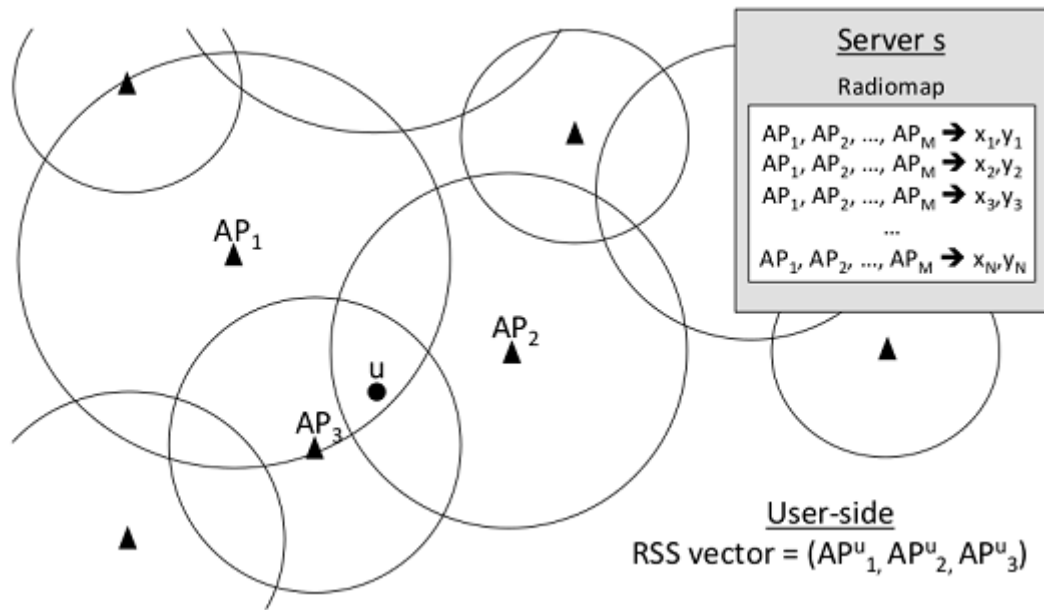
- Στην εκτός σύνδεσης φάση, χρήστες κινούνται στον χώρο και συλλέγουν πληροφορίες σημείων. Αυτές οι πληροφορίες περιλαμβάνουν ένα σύνολο Ασύρματων Σημείων Πρόσβασης στο Διαδίκτυο που η κινητή συσκευή ακούει στην συγκεκριμένη τοποθεσία. Κάθε τέτοιο σημείο πρόσβασης συνοδεύεται από την ισχύ με την οποία καταγράφηκε στην συγκεκριμένη τοποθεσία.
- Στην φάση της γεωτοποθέτησης, ο κάτοχος κινητής συσκευής σαρώνει παθητικά τα Σημεία Πρόσβασης που γειτνιάζουν μαζί του, μετρώντας επίσης την ισχύ με την οποία ακούγεται το συγκεκριμένο Σημείο Πρόσβασης. Αυτή η πληροφορία συγκρίνεται με την υφιστάμενη πληροφορία που συλλέχθηκε στην εκτός σύνδεσης Φάση. Αυτή η σύγκριση παρέχει την πιο πιθανή τοποθεσία που δύναται να είναι ο χρήστης.

	Access Points			
	AP_1	AP_2	...	AP_M
x_1, y_1	RSS_1	RSS_2	...	RSS_M
x_2, y_2	RSS_1	RSS_2	...	RSS_M
x_3, y_3	RSS_1	RSS_2	...	RSS_M
...	...	...	...	...
x_n, y_n	RSS_1	RSS_2	...	RSS_M

Πίνακας 1.2: Δείγμα Ραδιοχάρτη

Σε αυτό το σημείο αξίζει να σημειωθεί ότι η εκτός σύνδεση φάση δημιουργεί τον αυτό-καλούμενο Ραδιοχάρτη (*Radio-Map*). Μπορούμε να φανταστούμε τον Ραδιοχάρτη ως ένα δισδιάστατο πίνακα ο οποίος περιέχει αντιστοιχήσεις από μοναδικές τοποθεσίες σε Τιμές Ισχύος. Ένα δείγμα ραδιοχάρτη φαίνεται στο Σχήμα 1.2. Ειδικότερα, μια μοναδική τοποθεσία συσχετίζεται με δύο σύνολα πληροφοριών.

$$(x, y) = \begin{cases} AP = AP_1, AP_2, \dots, AP_n \\ RSS = RSS_1, RSS_2, \dots, RSS_n \end{cases}$$



Σχήμα 1.2: Κίνηση χρήστη με την τεχνική των Αποτυπωμάτων

Το σύνολο AP είναι το σύνολο των Σημείων Πρόσβασης που ακούγονται στην συγκεκριμένη τοποθεσία (x, y) . Το σύνολο RSS είναι οι αντίστοιχες τιμές ισχύος με τις οποίες ακούγονται τα Σημεία Πρόσβασης. Προφανώς, τα δύο πιο πάνω σύνολα πληροφοριών, AP και RSS , έχουν το ίδιο μέγεθος. Η προσέγγιση που ακολουθήσαμε σε αυτή την Διπλωματική Εργασία, είναι η γεωτοποθέτηση χρησιμοποιώντας την υφιστάμενη υποδομή του κτιρίου Πληροφορικής και την Τεχνική Γεωτοποθέτησης με την χρήση Αποτυπωμάτων. Πριν την διαδικασία της γεωτοποθέτησης κάποιο εργατικό προσωπικό ή καθημερινοί χρήστες του χώρου πρέπει να συλλέξουν τα Αποτυπώματα Τιμών Ισχύος Λαμβανόμενου Σήματος (*RSS Fingerprints*). Καθώς ένας χρήστης έξυπνης κινητής συσκευής κινείται σε ένα εσωτερικό χώρο, σαρώνει παθητικά τα ειδικά πακέτα (*beacon packets*) που αποστέλλονται από τα Ασύρματα Σημεία Πρόσβασης για να γνωστοποιήσουν την ύπαρξη υποδομής Ασύρματης Πρόσβασης στο Διαδίκτυο. Ακολούθως, υπολογίζεται η πιο πιθανή τοποθεσία του χρήστη χρησιμοποιώντας υφιστάμενους αλγόριθμους γεωτοποθέτησης *KNN*, *MMSE* [14]. Το πλεονέκτημα έναντι των άλλων τεχνικών γεωτοποθέτησης είναι ότι δεν χρειάζεται κάποιος επιπλέον εξοπλισμός για την γεωτοποθέτηση των χρηστών.

Το Σχήμα 1.2 απεικονίζει διαγραμματικά την κίνηση ενός χρήστη σε έναν εσωτερικό χώρο. Καθώς κινείται συλλέγει τις τιμές ισχύος των σημείων πρόσβασης που γειτνιάζουν μαζί του και κατασκευάζει το αυτό-καλούμενο αποτύπωμα τιμών ισχύος. Το αποτύπωμα αυτό χρησιμοποιείται για τον υπολογισμό της τοποθεσίας του χρήστη, βρίσκοντας το

καλύτερο ταίρι από τον ραδιοχάρτη του κτιρίου. Στο παράδειγμα του Σχήματος 1.2 ο χρήστης u ακούει μόνο τρία ασύρματα σημεία πρόσβασης. Σημειώνεται επίσης ότι πολλά από τα σημεία πρόσβασης στο ίδιο παράδειγμα, είναι εκτός εμβέλειας του χρήστη.

1.2 Πρόταση Αλγορίθμου Προφόρτωσης των Ραδιοχαρτών

Όπως αναφέραμε και στο προηγούμενο υπό-κεφάλαιο, για να εκτελέσει ένας χρήστης γεωτοποθέτηση χρησιμοποιώντας την τεχνική Γεωτοποθέτησης Αποτυπωμάτων πρέπει να ζητήσει εκ των προτέρων τον Ραδιοχάρτη του κτιρίου. Οι χάρτες αυτοί ενδέχεται να είναι πολύ μεγάλοι, για παράδειγμα ο ιστιότοπος WiGLE.net έχει αυτή την στιγμή, περίπου 2.14 δισεκατομμύρια μοναδικά καταγεγραμμένων αποτυπωμάτων από εθελοντές που εκτελούσαν διαδικασίες γεωγραφικού εντοπισμού (*wardriving*).

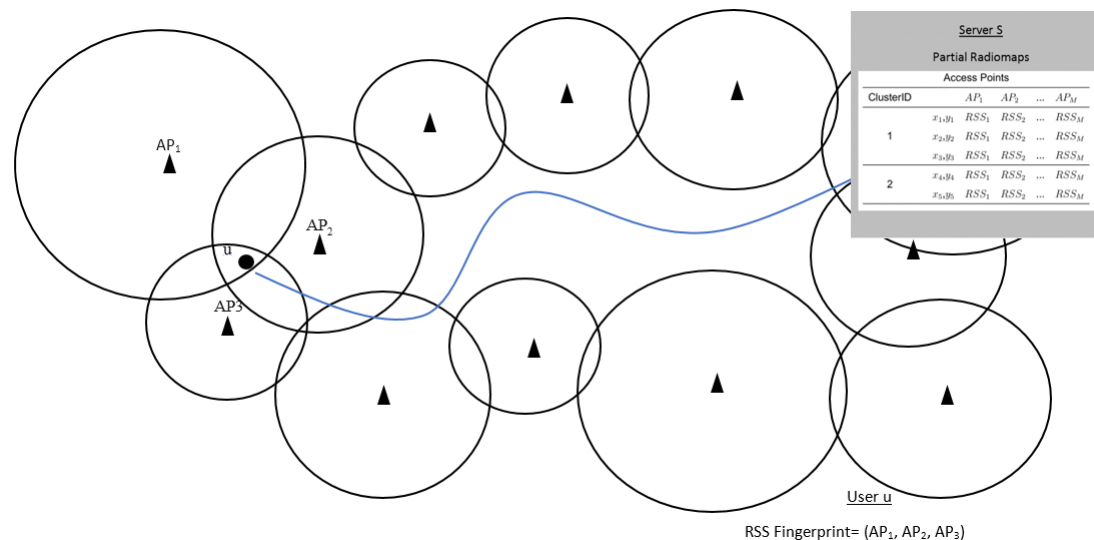
Καθώς ένας κάτοχος Έξυπνης Κινητής Συσκευής, κινείται στο εσωτερικό ενός κτιρίου η σύνδεση του στο διαδίκτυο είναι διακοπτόμενη (*Intermittent Connectivity*). Αυτό αποτελεί πρόβλημα, εφόσον ο χρήστης πρέπει να βρίσκεται σε επικοινωνία με τον εξυπηρετητή ο οποίος έχει στην κατοχή του τους ραδιοχάρτες του κτιρίου. Για την αντιμετώπιση του προβλήματος αυτού προτείνουμε ένα αλγόριθμο Προφόρτωσης των ραδιοχαρτών. Η προφόρτωση γίνεται στο επίπεδο της συστοιχίας. Μια συστοιχία είναι ένας μερικός ραδιοχάρτης ο οποίος περιέχει γειτονικά, στον Ευκλείδειο χώρο, σημεία ενός ορόφου. Για κάθε τέτοιο σημείο γνωρίζουμε ποια Σημεία Ασύρματης Πρόσβασης ακούγονται στην συγκεκριμένη τοποθεσία και με πόση ισχύ λαμβάνεται το κάθε σήμα.

Στο ένα άκρο, η προφόρτωση των τιμών ισχύος θα μπορούσε να γίνεται στο επίπεδο του ενός γεωγραφικού σημείου. Παρόλα αυτά, η προσέγγιση αυτή δεν είναι αποδοτική καθώς η κινητή συσκευή θα έπρεπε να βρίσκεται σε συνεχή επικοινωνία με τον εξυπηρετητή S . Δεδομένου ότι η σύνδεση είναι διακοπτόμενη, η προηγούμενη προσέγγιση δεν θα δούλευε. Στο άλλο αντίθετο άκρο, η προφόρτωση θα μπορούσε να γίνεται στο επίπεδο όλων των συστοιχιών ενός κτιρίου (και συνεπώς ολόκληρο τον ραδιοχάρτη). Και πάλι αυτή η προσέγγιση δεν είναι αποδοτική από πλευράς χρόνου και κατανάλωσης ενέργειας, καθώς το μέγεθος των ραδιοχαρτών για μεγάλα κτίρια είναι σημαντικά μεγάλο [12].

Επομένως η προφόρτωση γίνεται στο επίπεδο της συστοιχίας χρησιμοποιώντας μια Μηχανή Προβλέψεων (*Prediction Engine*) βασισμένη πάνω σε ένα γράφημα εξαρτήσεων.

Με αυτό τον τρόπο επιτυγχάνουμε όσο το δυνατόν λιγότερη επικοινωνία με τον εξυπηρετητή και μείωση του απαιτούμενου χρόνου εξυπηρέτησης ενός χρήστη. Προφανώς, σε ένα περιβάλλον διακοπτόμενης σύνδεσης η εξυπηρέτηση του πελάτη πρέπει να γίνεται σποραδικά και σε πολύ μικρά χρονικά διαστήματα. Παράλληλα, ο χρήστης μπορεί να κινείται επιτυχώς μέσα στον χώρο, χωρίς συνεχή επικοινωνία με τον εξυπηρετητή. Συνεπώς, ο αλγόριθμος Προφόρτωσης που προτείνουμε είναι αποτελεσματικός για περιβάλλοντα με διακοπτόμενη σύνδεση στο δίκτυο.

Όπως έχουμε ήδη αναφέρει η Μηχανή Προβλέψεων κάνει χρήση ενός γραφήματος εξαρτήσεων. Το γράφημα αυτό είναι κατευθυνόμενο και περιγράφει πιθανολογικά τις διαδρομές των χρηστών σε ένα εσωτερικό χώρο. Αν αναλογιστούμε τις διαδρομές που κάνουμε καθημερινά, θα παρατηρήσουμε ότι ακολουθούμε κάποια μοτίβα κίνησης μας επαναλαμβάνονται ξανά και ξανά. Για παράδειγμα, ένας υπάλληλος όταν καταφθάνει στην δουλειά του, πρώτα θα περάσει την κάρτα του, μετά θα περάσει να πάρει την αλληλογραφία του και μετά θα πάει στο γραφείο του. Σωστά μπορεί να υποθέσει κάποιος ότι κάποια πρότυπα κίνησης επαναλαμβάνονται συνέχεια.



Σχήμα 1.3: Κίνηση χρήστη σε περιβάλλον διακοπτόμενης σύνδεσης

Στο Σχήμα 1.3 φαίνεται η πρόθεση του χρήστη να διασχίσει και να πλοηγηθεί σε ένα περιβάλλον διακοπτόμενης σύνδεσης. Το μονοπάτι που πρόκειται να ακολουθήσει ο χρήστης απεικονίζεται με την μπλε γραμμή. Παρατηρούμε ότι κατά μήκος της διαδρομής του χρήστη δεν υπάρχει επικάλυψη από κανένα σημείο πρόσβασης, οπότε ο χρήστης δεν είναι σε θέση να πλοηγηθεί με επιτυχία διότι δεν είναι σε θέση να επικοινωνήσει με τον

εξυπηρετητή. Στο ένα άκρο ο ραδιοχάρτης θα μπορούσε να βρίσκεται εξ' ολοκλήρου στον εξυπηρετητή (*τεχνική Server Side Radiomap*), σε αυτή την περίπτωση ο χρήστης πρέπει να επικοινωνεί με τον εξυπηρετητή για κάθε νέα τοποθεσία του. Στο άλλο άκρο ο χρήστης μπορεί να προφορτώσει ολόκληρο τον ραδιοχάρτη του κτιρίου στην συσκευή του και να εκτελέσει πλοήγηση (*τεχνική Client Side Radiomap*). Όπως αναφέρθηκε ήδη αυτή η προσέγγιση δεν συνιστάται για μεγάλου μεγέθους ραδιοχάρτες [12]. Τονίζεται ότι στο Σχήμα 1.3 ο χρήστης δεν είναι σε θέση να ενωθεί με τα σημεία πρόσβασης μόνιμα. Αλλά τα λαμβάνει με κάποια τιμή ισχύος, και άρα οι αλγόριθμοι γεωτοποθέτησης είναι σε θέση να υπολογίσουν την τοποθεσία του.

1.3 Πειραματική Σύνοψη

Το πλαίσιο αξιολόγησης των αλγορίθμων αποτελείται από ένα Κατανεμημένο Προσομοιωτή γραμμένο στην προγραμματιστική γλώσσα *Java*. Θα γίνει αξιολόγηση των αλγορίθμων Πρόβλεψης και Προφόρτωσης χρησιμοποιώντας ως μετρικές αξιολόγησης τον αριθμό των *Bytes* που ανταλλάσσονται μεταξύ εξυπηρετητή και πελάτη και τον απαιτούμενο χρόνο εξυπηρέτησης. Οι αλγόριθμοι Πρόβλεψης και Προφόρτωσης θα συγκριθούν με την προσέγγιση όπου ολόκληρος ο Ραδιοχάρτης κατεβαίνει στην κινητή συσκευή εξαρχής (πριν την έναρξη της γεωτοποθέτησης από τον χρήστη). Η προσέγγιση αυτή ονομάζεται *Client Side Radiomap (CSR)*. Προφανώς με τον αλγόριθμο *CSR* ο πελάτης μπορεί να κινείται στον χώρο χωρίς να ανησυχεί για την σύνδεση του στο Διαδίκτυο. Παρόλα αυτά σε περιβάλλοντα με διακοπτόμενη σύνδεση στο Διαδίκτυο, η προφόρτωση ολόκληρου του Ραδιοχάρτη στην κινητή συσκευή ίσως να μην είναι εφικτή λόγω του μεγάλου μεγέθους του Ραδιοχάρτη και άρα του μεγαλύτερου χρόνου ολικής προ-ανάκτησης του. Ο αλγόριθμος *CSR* λοιπόν, έχει στην κατοχή του κατά την διάρκεια ολόκληρης της πλοήγησης του χρήστη σε ένα κτίριο, ολόκληρο τον Ραδιοχάρτη. Αυτό έχει ως συνέπεια ο αλγόριθμος να παρέχει γεωτοποθέτηση υψηλής ακρίβειας όταν δεν υπάρχει σύνδεση στο Διαδίκτυο.

Από την άλλη πλευρά, οι αλγόριθμοι Πρόβλεψης και Προφόρτωσης στο ίδιο περιβάλλον διακοπτόμενης σύνδεσης, δεν έχουν εξαρχής ολόκληρο τον Ραδιοχάρτη πριν την έναρξη της γεωτοποθέτησης του πελάτη. Αντίθετα με τον αλγόριθμο *CSR* ο οποίος έχει στην κατοχή του ολόκληρο τον Ραδιοχάρτη, οι αλγόριθμοι Προφόρτωσης έχουν στην διάθεση τους μερικούς Ραδιοχάρτες. Προφανώς αυτό μειώνει την ακρίβεια της γεωτοποθέ-

τησης σημαντικά, ιδίως όταν ο χρήστης κινείται σε χώρους για τους οποίους δεν έχει μερικό Ραδιοχάρτη. Επομένως στο Κεφάλαιο 6 θα γίνει αποτίμηση της ακρίβειας γεωτοποθέτησης των Αλγορίθμων Πρόβλεψης-Προφόρτωσης όταν συμβαίνουν αποτυχίες στο Δίκτυο του χρήστη. Η ακρίβεια θα υπολογιστεί χρησιμοποιώντας την εξίσωση του Μέσου Απόλυτου Σφάλματος (*Mean Absolute Error*).

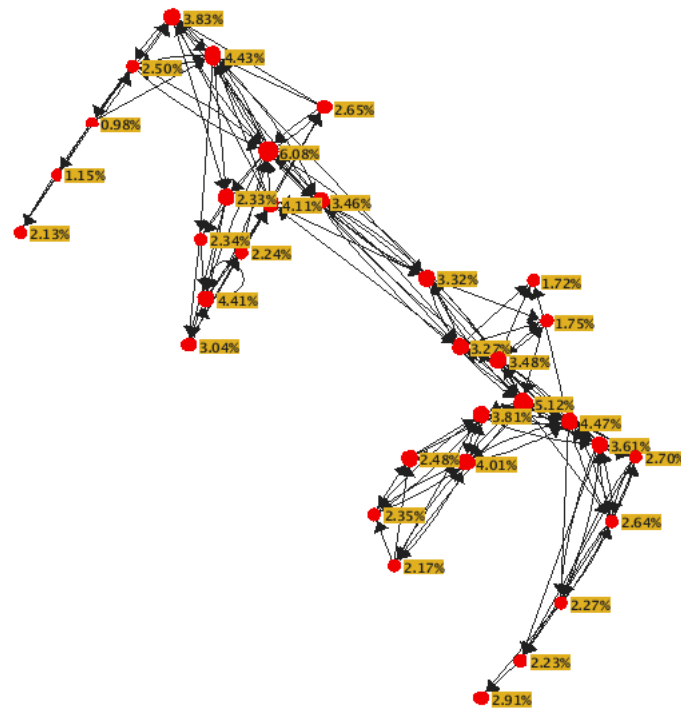


Σχήμα 1.4: Πλοήγηση μέχρι πριν την αποτυχία του δικτύου

1.4 Συνεισφορά Ατομικής Διπλωματικής Εργασίας

Σε αυτή την Διπλωματική Εργασία έχει οριστεί και μελετηθεί το πρόβλημα της διακοπτόμενης σύνδεσης στο διαδίκτυο, οποιουδήποτε πελάτη που θέλει να εκτελέσει γεωτοποθέτηση σε ένα εσωτερικό περιβάλλον. Το πρόβλημα αυτό είναι σημαντικό καθώς ολοένα και περισσότερες εφαρμογές χρειάζονται γνώση της τοποθεσίας του χρήστη. Οι χρήστες αυτοί πολύ συχνά βρίσκονται σε μεγάλα εσωτερικά περιβάλλοντα όπου η σύνδεση στο διαδίκτυο δεν επικαλύπτει ολόκληρο τον εσωτερικό χώρο. Με την πρόταση των Αλγορίθμων Προφόρτωσης που ορίζονται σε αυτή την διπλωματική εργασία, δίνουμε την δυνατότητα σε αυτούς του χρήστες να πλοηγούνται με ικανοποιητική ακρίβεια, σε περιβάλλοντα διακοπτόμενης σύνδεσης. Επίσης οι αλγόριθμοι Προφόρτωσης εκτελούν προ-ανάκτηση των μερικών Ραδιοχαρτών που είναι το πιθανότερο να προσπελαστούν από τον χρήστη. Πιο συγκεκριμένα, οι αλγόριθμοι Προφόρτωσης συμβουλεύονται το ιστορικό

κίνησης των χρηστών στο ίδιο περιβάλλον γεωτοποθέτησης. Συμπεραίνουμε ότι οι πιο δημοφιλείς τοποθεσίες ενός εσωτερικού περιβάλλοντος θα δίνονται στον χρήστη (σύμφωνα βέβαια με την τρέχουσα τοποθεσία του). Στο Σχήμα 1.4 φαίνεται μια υλοποίηση των πιο πάνω αλγορίθμων, όπου απεικονίζεται η προφόρτωση των μερικών ραδιοχαρτών σε ένα περιβάλλον διακοπτόμενης σύνδεσης.



Σχήμα 1.5: Γράφημα Εξαρτήσεων Περιβάλλοντος Αξιολόγησης

Ο πρώτος αλγόριθμος που προτείνεται βασίζεται πάνω σε ένα γράφημα εξαρτήσεων για την εκτέλεση των προβλέψεων. Το γράφημα αυτό, περιέχει τις πιθανότητες μετάβασης ενός χρήστη από κάποιο κόμβο σε ένα γειτονικό κόμβο. Η πιθανότητα αυτή υπολογίζεται λαμβάνοντας υπόψη την συχνότητα της κάθε μετάβασης από ένα αρχικό κόμβο σε κάποιο κόμβο προορισμού και την συχνότητα του κάθε εναρκτηρίου κόμβου. Ο λόγος των δύο προηγούμενων στοιχείων αποτελούν και την πιθανότητα μετάβασης ενός χρήστη από κάποιο αρχικό κόμβο σε κάποιο τελικό κόμβο.

Ο δεύτερος αλγόριθμος που προτείνεται βασίζεται πάνω σε ένα γράφημα στο οποίο έχει εφαρμοστεί ο αλγόριθμος *PageRank* ο οποίος αναθέτει μια σημαντικότητα σε κάθε κόμβο. Δεδομένου της τρέχουσας τοποθεσίας του χρήστη (κόμβος) προτείνονται οι γειτονικοί κόμβοι που έχουν την μεγαλύτερη σημαντικότητα. Υπενθυμίζεται ότι ο αλγόριθμος *PageR-*

ank υποθέτει το μοντέλο ενός τυχαίου περιηγητή (*Random Surfer*) ο οποίος είναι σε θέση να μεταπηδήσει σε ένα τυχαίο κόμβο του γραφήματος. Στην δική μας περίπτωση, υποθέτουμε το ίδιο μοντέλο για την εκπαίδευση του γραφήματος αλλά κατά την διάρκεια της γεωτοποθέτησης του χρήστη, ο χρήστης δεν είναι σε θέση να μεταπηδήσει σε κάποιο τυχαίο κόμβο (για αυτό και επιλέγεται ο σημαντικότερος *Γειτονικός* κόμβος, δεδομένου του τρέχον κόμβου του χρήστη).

Στο Σχήμα 1.5 φαίνεται το γράφημα εξαρτήσεων που κατασκευάστηκε για το πραγματικό περιβάλλον αξιολόγησης του αλγορίθμου (το κτίριο Πληροφορική του Πανεπιστημίου Κύπρου). Παρατηρούμε ότι οι ακμές είναι κατευθυνόμενες και στον κάθε κόμβο ανατίθεται ένα ποσοστό το οποίο είναι η σημαντικότητα του κάθε κόμβου (η σημαντικότητα αυτή δίνεται από τον αλγόριθμο BBA). Τέλος οι πιθανότητες μετάβασης (το βάρος της κάθε ακμής) που χρησιμοποιούνται από τον αλγόριθμο *CBA*³ παραλείπονται για λόγους σαφήνειας.

1.5 Διάγραμμα Ατομικής Διπλωματικής Εργασίας

Στο Κεφάλαιο 2 θα μελετηθεί η σχετική εργασία της ΑΔΕ. Ειδικότερα, θα μελετηθούν αλγόριθμοι Προφόρτωσης στο Διαδίκτυο και ο αλγόριθμος ανάθεσης σημαντικότητας *PageRank*. Στο επόμενο κεφάλαιο θα διατυπωθεί πιο τυπικά το πρόβλημα το οποίο προσπαθούμε να επιλύσουμε και θα παρουσιαστεί το μοντέλο συστήματος πάνω στο οποίο βασίζονται οι αλγόριθμοι Προφόρτωσης και Πρόβλεψης. Στο Κεφάλαιο 4 παρουσιάζονται οι αλγόριθμοι Πρόβλεψης και Προφόρτωσης και αναλύεται η χρονική πολυπλοκότητα τους. Στην συνέχεια, στο Κεφάλαιο 5 παρουσιάζεται ένα πρότυπο σύστημα υλοποίησης το οποίο περιλαμβάνει την Μηχανή Προφόρτωσης στην πλευρά του πελάτη και την Μηχανή Πρόβλεψης στην πλευρά του εξυπηρετητή. Η πειραματική αξιολόγηση των αλγορίθμων βρίσκεται στο Κεφάλαιο 6 όπου επεξηγείται η μεθοδολογία αξιολόγησης καθώς και το περιβάλλον αξιολόγησης. Τέλος, στο Κεφάλαιο 7 θα παρουσιαστούν συμπεράσματα και μελλοντική δουλεία που ίσως να χρειαστεί για την επέκταση αυτής της Διπλωματικής Εργασίας.

³Περισσότερα για τους αλγόριθμους BBA και CBA στα επόμενα κεφάλαια

Κεφάλαιο 2

Σχετική Εργασία

2.1	Γεωτοποθέτηση Σε Εσωτερικούς Χώρους	12
2.2	Προφόρτωση στο Διαδίκτυο	13
2.2.1	Μηχανή Προβλέψεων στο Διαδίκτυο	14
2.2.2	Μηχανή Προφόρτωσης στο Διαδίκτυο	15
2.2.3	Αλγόριθμος Πρόβλεψης Dependency Graph	15
2.3	Αλγόριθμος PageRank	16
2.3.1	Τυπικός Ορισμός του PageRank	17
2.3.2	Αναπλήρωση Παγιδευμένης Σημαντικότητας (Rank Sinks) . . .	17
2.3.3	Παράγοντας Απόσβεσης (Damping Factor)	18
2.4	Αλγόριθμος Ομαδοποίησης BFR	19

2.1 Γεωτοποθέτηση Σε Εσωτερικούς Χώρους

Μια από τις πιο διαδεδομένες τεχνικές γεωτοποθέτησης σε εσωτερικούς χώρους είναι αυτή που κάνει χρήση Αποτυπωμάτων Τιμών Ισχύος (*RSS Fingerprints*) για την γεωτοποθέτηση του χρήστη. Όπως έχει αναφερθεί και στην εισαγωγή, τα συστήματα αυτά χρησιμοποιούν την υφιστάμενη δομή ασύρματης πρόσβασης στο Διαδίκτυο με σκοπό την παροχή υπηρεσιών γεωτοποθέτησης. Επιπλέον, η τεχνική αυτή υποθέτει ότι κάθε μοναδική θέση στον χώρο συσχετίζεται με ένα σύνολο αποτυπωμάτων. Κάθε αποτύπωμα είναι ένα σύνολο από εντάσεις σήματος, όπου η κάθε ένταση προέρχεται από ένα μοναδικό σημείο πρόσβασης. Με την χρήση αλγορίθμων γεωτοποθέτησης υπολο-

γίζεται η πιο πιθανή τοποθεσία του χρήστη από ένα σύνολο αποτυπωμάτων το οποίο έχει συλλεχθεί στην εκτός σύνδεσης και εκπαίδευσης φάση. Ο Ραδιοχάρτης που συλλέχθηκε στην εκτός σύνδεσης φάση περιέχει σημεία αναφοράς σε ένα εσωτερικό περιβάλλον γεωτοποθέτησης. Τέλος, τα σημεία αναφοράς περιγράφονται πλήρως από ένα διάνυσμα τιμών ισχύος για κάθε ασύρματο σημείο πρόσβασης που βρίσκεται γειτονικά του σημείου αναφοράς.

Μια υποδειγματική δουλειά της πιο πάνω τεχνικής είναι το σύστημα *Airplace* [13] το οποίο εκμεταλλεύεται τις τιμές ισχύος με τις οποίες ακούγεται ένα σύνολο σημείων πρόσβασης. Οι τιμές ισχύος αυτές μαζεύονται από την κινητή συσκευή ως μέρος της καθημερινής λειτουργίας του δικτύου, δηλαδή με την παθητική σάρωση ειδικών πακέτων (*Beacon Packets*) που μεταδίδονται από γειτονικά σημεία πρόσβασης. Η φάση γεωτοποθέτησης του συστήματος *Airplace* προϋποθέτει την ύπαρξη των Ραδιοχαρτών οι οποίοι έχουν δημιουργηθεί εκ των προτέρων σε αρκετά σημεία αναφοράς ενός εσωτερικού περιβάλλοντος.

Μια άλλη εξαιρετικά διαδεδομένη τεχνολογία γεωτοποθέτησης είναι η χρήση έτοιμων Υπηρεσιών Γεωτοποθέτησης όπως είναι η δημοφιλής διεπαφή προγραμματισμού γεωτοποθέσις της *Google* (*Google Maps Geolocation API*). Η προηγούμενη διεπαφή επιστρέφει την τοποθεσία του χρήστη αντλώντας την πληροφορία αυτή από τους πύργους τηλεφωνίας (*Cell Towers*) και τα Ασύρματα Σημεία Πρόσβασης (*WiFi Access Points*) που ανιχνεύονται από την κινητή συσκευή του χρήστη. Η επικοινωνία με την Υπηρεσία Γεωτοποθέτησης της *Google* γίνεται με την χρήση ενός πρωτοκόλλου σε μορφή *JSON*. Στο Σχήμα 2.1 φαίνεται η αλληλεπίδραση μεταξύ ενός πελάτη *Android* και των εξυπηρετητών της *Google*.

Προφανώς αυτή η προσέγγιση χρειάζεται συνεχή σύνδεση στο διαδίκτυο κάτι το οποίο σε πολλές περιπτώσεις δεν είναι εφικτό. Σε αυτή την διπλωματική εργασία θα μελετήσουμε την γεωτοποθέτηση με την χρήση Αποτυπωμάτων Τιμών Ισχύος αλλά και αλγόριθμους Προφόρτωσης των Ραδιοχαρτών με απώτερο στόχο την επιτυχή γεωτοποθέτηση του χρήστη σε ένα εσωτερικό περιβάλλον που υποφέρει από διακοπτόμενη σύνδεση στο διαδίκτυο.

1	{ "homeMobileCountryCode": 310,	1	... continued
	"homeMobileNetworkCode": 260,		"signalStrength": 8,
3	"radioType": "gsm",	3	"age": 0,
	"carrier": "T-Mobile",		"signalToNoiseRatio": -65,
5	"cellTowers": [{	5	"channel": 8 }, {
	"cellId": 39627456,		"macAddress": "
7	"locationAreaCode": 40495,	7	"01:23:45:67:89:AC",
	"mobileCountryCode": 310,		"signalStrength": 4,
9	"mobileNetworkCode": 260,	9	"age": 0 }]}
	"age": 0,		{ "location": {
11	"signalStrength": -95}],	11	"lat": 51.0,
	"wifiAccessPoints": [{		"lng": -0.1
13	"macAddress": "01:23:45:67:89:AB",	13	},
			"accuracy": 1200.4}

Client Location Request

Server Location Response

Σχήμα 2.1: Αίτηση τοποθεσίας από την Διεπαφή Γεωτοποθέτησης της Google

2.2 Προφόρτωση στο Διαδίκτυο

Για την Προφόρτωση στο Διαδίκτυο (*Web Prefetching*) έχουν προταθεί πολλές τεχνικές και αλγόριθμοι. Σε αυτό το τμήμα θα αναφέρουμε τις πιο σημαντικές δουλειές που αφορούν την Προφόρτωση στο Διαδίκτυο. Πάνω σε παρόμοιες ιδέες βασίζεται και η δουλειά σε αυτή την διπλωματική εργασία. Η Προφόρτωση στο Διαδίκτυο είναι μια τεχνική η οποία χρησιμοποιείται για την μείωση του χρόνου αναμονής των χρηστών καθώς πλοηγούνται στο Διαδίκτυο. Για την εκτέλεση της Προφόρτωσης στο Διαδίκτυο είναι απαραίτητη η υλοποίηση δύο μηχανών, της Μηχανής Πρόβλεψης και της Μηχανής Προφόρτωσης.

2.2.1 Μηχανή Προβλέψεων στο Διαδίκτυο

Η Μηχανή Προβλέψεων (*Prediction Engine*) είναι μέρος του συστήματος Προφόρτωσης και ο ρόλος του είναι να εικάζει τις επόμενες προσβάσεις του χρήστη [5]. Η μηχανή προβλέψεων βρίσκεται είτε στον εξυπηρετητή, είτε σε ένα άλλο στρώμα της αρχιτεκτονικής Διαδικτύου (για παράδειγμα στον Πληρεξούσιο Εξυπηρετητή-Proxy Server). Κάθε προσέγγιση μπορεί να παρέχει διαφορετικές προβλέψεις, καθώς ένας πληρεξούσιος

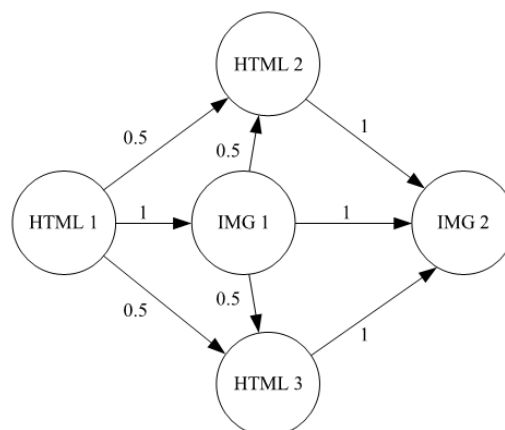
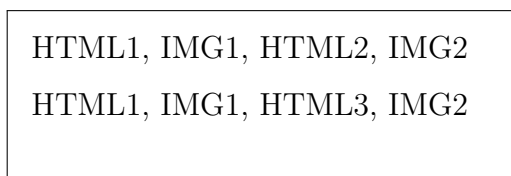
εξυπηρετητής είναι σε θέση να γνωρίζει μοτίβα πρόσβασης ενός χρήστη, κατά μήκος διαφορετικών εξυπηρετητών, ενώ ένας κανονικός εξυπηρετητής όχι. Η έξοδος της Μηχανής Πρόβλεψης είναι μια λίστα υποδείξεων (*Hint List*), όπου κάθε αντικείμενο αυτής της λίστας είναι ένα URI (*Uniform Resource Identifier*) το οποίο αναγνωρίζει μοναδικά έναν πόρο (οποιοδήποτε αντικείμενο περιέχει μια σελίδα, π.χ. μια εικόνα).

2.2.2 Μηχανή Προφόρτωσης στο Διαδίκτυο

Η Μηχανή Προφόρτωσης (*Prefetching Engine*) είναι το δεύτερο απαραίτητο μέρος του συστήματος Προφόρτωσης. Μπορεί να βρίσκεται σε διάφορα στρώματα της Αρχιτεκτονικής Διαδικτύου αλλά η τρέχουσα τάση τοποθετεί την Μηχανή Προφόρτωσης στον πελάτη (για παράδειγμα ο Περιηγητής Διαδικτύου της *Mozilla* περιέχει την Μηχανή Προφόρτωσης [5]). Η μηχανή Προφόρτωσης είναι αυτή η οποία δέχεται την λίστα υποδείξεων από την μηχανή προβλέψεων, επεξεργάζεται κάθε καταχώρηση στην λίστα και αποφασίζει πότε θα προφορτώσει κάποιο πόρο (για παράδειγμα όταν ο χρήστης είναι αδρανής). Όταν ο χρήστης ζητήσει κάποιον πόρο, τον οποίο έχει υποδείξει η λίστα υποδείξεων ο χρόνος αναμονής του χρήστη μειώνεται σημαντικά.

2.2.3 Αλγόριθμος Πρόβλεψης *Dependency Graph*

Ο αλγόριθμος *Dependency Graph* κατασκευάζει ένα γράφημα εξαρτήσεων το οποίο περιλαμβάνει κάθε Διαδικτυακό πόρο που έχει προσπελαστεί από κάποιον συγκεκριμένο χρήστη. Ειδικότερα, στο γράφημα αυτό υπάρχει ένας κόμβος για κάθε διαδικτυακό πόρο που έχει προσπελαστεί, από κάποιο χρήστη κατά την διάρκεια της πλοήγησης του στο διαδίκτυο παλαιότερα. Ας υποθέσουμε δύο Διαδικτυακούς πόρους R_a και R_b , τότε υπάρχει μια ακμή $R_a \rightarrow R_b$ εάν και μόνο εάν κάποια στιγμή ο χρήστης ζήτησε τον πόρο R_b μετά τον πόρο R_a . Σημειώνεται ότι το μονοπάτι του χρήστη από τον πόρο R_a στον πόρο R_b , μπορεί να περιέχει πολλούς ενδιάμεσους πόρους. Το μέγεθος του μονοπατιού αυτού ονομάζεται *μέγεθος παραθύρου* (*Lookahead Window*) και συμβολίζεται με w . Το βάρος της ακμής είναι ο λόγος του αριθμού των προσβάσεων στον πόρο R_b , διαμέσου ενός μονοπατιού (μεγέθους w) πόρων από τον πόρο R_a , προς τον αριθμό των αιτήσεων του πόρου R_a . Αν υποθέσουμε το ιστορικό αιτήσεων ενός χρήστη που φαίνεται στο Σχήμα 2.2, τότε προκύπτει το γράφημα εξαρτήσεων στο Σχήμα 2.3.



Σχήμα 2.2: Ιστορικό Αιτήσεων Χρηστών

Σχήμα 2.3: Γράφημα Εξαρτήσεων

2.3 Αλγόριθμος PageRank

Ο αλγόριθμος *PageRank* μοντελοποιεί την συμπεριφορά ενός τυχαίου περιηγητή ¹ στο Διαδίκτυο και χρησιμοποιείται από την μηχανή αναζήτησης *Google* για να μετρήσει την σημαντικότητα των ιστοσελίδων. Για τον υπολογισμό αυτής της σημαντικότητας ο αλγόριθμος *PageRank* λαμβάνει υπόψη του την δομή των συνδέσμων στο Διαδίκτυο. Πιο συγκεκριμένα το Διαδίκτυο περιγράφεται από ένα γράφημα (*Webgraph*), όπου οι κόμβοι του είναι οι σελίδες του διαδικτύου και οι ακμές χρησιμοποιούνται για να υποδείξουν ότι υπάρχει ένας υπέρ-σύνδεσμος που ενώνει δύο σελίδες του Διαδικτύου. Μια σελίδα στο διαδίκτυο περιέχει ένα αριθμό από αναφορές, δηλαδή υπέρ-συνδέσμους που οδηγούν σε άλλες σελίδες (στο διαδικτυακό γράφημα οι υπέρ-συνδέσμοι αυτοί απεικονίζονται από εξερχόμενες ακμές (*Forward link*) από ένα κόμβο, όπου ο κόμβος αντιπροσωπεύει την σελίδα που περιέχει τους υπέρ-συνδέσμους). Επίσης, μια σελίδα του διαδικτύου μπορεί να αναφέρεται από κάποιες άλλες. Στο διαδικτυακό γράφημα οι υπέρ-συνδέσμοι αυτοί απεικονίζονται ως εισερχόμενες ακμές (*back-links*) σε ένα κόμβο, όπου ο κόμβος αντιπροσωπεύει την σελίδα η οποία αναφέρεται. Σε γενικές γραμμές, μια σελίδα η οποία αναφέρεται σε μεγάλο βαθμό (δηλαδή έχει πολλές εισερχόμενες ακμές στο γράφημα) έχει μεγαλύτερη σημαντικότητα. Παρόλα αυτά, μια σελίδα *A* η οποία αναφέρεται από μια πολύ σημαντική σελίδα (π.χ. *www.cnn.com*), είναι περισσότερο σημαντική από μια σελίδα *B*, η οποία αναφέρεται από πολλές ασήμαντες σελίδες.

¹Δεδομένης μια αρχικής σελίδας ακολουθεί κάποιους συνδέσμους χωρίς να οπισθοχωρεί σε μια προηγούμενη σελίδα. Εν τέλει όμως, ο χρήστης "βαριέται" και ξεκινά από μια άλλη τυχαία σελίδα

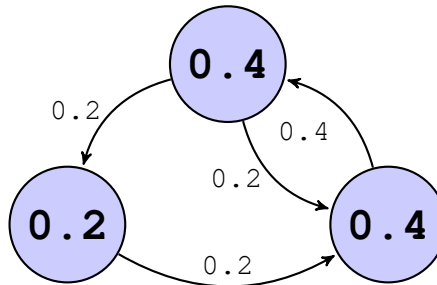
2.3.1 Τυπικός Ορισμός του PageRank

Ας ορίσουμε μια σελίδα A με $Forward_A$ να είναι το σύνολο των σελίδων πάνω στις οποίες δείχνει η σελίδα A και με $Back_A$ να είναι το σύνολο των σελίδων οι οποίες δείχνουν στην σελίδα A . Για τον υπολογισμό της σημαντικότητας του κόμβου A αθροίζεται η σημαντικότητα των κόμβων που δείχνουν σε αυτόν, όπου στην περίπτωση μας είναι το σύνολο $Back_A$. Έπειτα, διαιρείται το άθροισμα αυτό με τον πληθικό αριθμό του συνόλου $Forward_A$ (το οποίο είναι το σύνολο των σελίδων που αναφέρει η σελίδα A). Έστω ότι η σημαντικότητα συμβολίζεται με R , τότε ο τυπικός ορισμός της σημαντικότητας της σελίδας A είναι:

Ορισμός 1
$$R(A) = c \cdot \sum_{b \in Back_A} \frac{R(b)}{|Forward_A|}$$

όπου $0 < c < 1$ είναι μια σταθερά που χρησιμοποιείται για κανονικοποίηση.

Από τον προηγούμενο τυπικό ορισμό είναι ξεκάθαρο ότι η σημαντικότητα μιας σελίδας διαιρείται με τον αριθμό των εξερχόμενων συνδέσμων της σελίδας. Σημειώνεται ότι ο αλγόριθμος *PageRank* εκτελεί κάποιες επαναλήψεις πριν να συγκλίνει εντελώς. Ένα απλό παράδειγμα της εκτέλεσης του πιο πάνω αλγορίθμου φαίνεται στο Σχήμα 2.4.

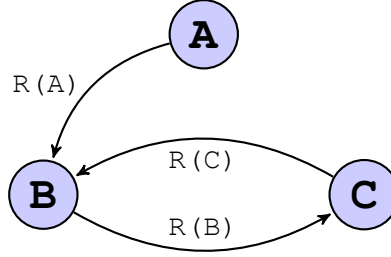


Σχήμα 2.4: Υπολογισμός Σημαντικότητας *PageRank*

2.3.2 Αναπλήρωση Παγιδευμένης Σημαντικότητας (Rank Sinks)

Η προηγούμενη επεξήγηση του αλγορίθμου *PageRank* αγνοούσε κάποιου είδους παγίδες οι οποίες ονομάζονται *Rank Sink*. Οι παγίδες αυτές σχηματίζονται στο Διαδικτυακό Γράφημα όταν δύο κόμβοι δείχνουν ο ένας στον άλλο χωρίς να δείχνουν σε κάποιο άλλο κόμβο. Ας υποθέσουμε λοιπόν το σενάριο που φαίνεται στο Σχήμα 2.5

Δεδομένου ότι ο αλγόριθμος *PageRank* χρειάζεται κάποιο αριθμό επαναλήψεων για να συγκλίνει τότε ο κόμβος B θα παίρνει σημαντικότητα από τον κόμβο A με αποτέλεσμα



Σχήμα 2.5: Περίπτωση Παγίδας

οι δύο κόμβοι C και B να συσσωρεύουν σημαντικότητα ο ένας από τον άλλο, χωρίς να δίνουν σε κάποιο τρίτο κόμβο του γραφήματος. Για να επανορθώσει ο αλγόριθμος την πιο πάνω κατάσταση κάνει χρήση ενός περαιτέρω διανύσματος E πάνω στον Ορισμό 1. Το διάνυσμα E είναι μια κατανομή από ιστοσελίδες στις οποίες μπορεί να μεταβεί κάποιος τυχαίος περιηγητής και χρησιμοποιείται για να δώσει μια εκτενή εικόνα του Διαδικτύου.

Ορισμός 2 $R'(A) = c \cdot \sum_{b \in \text{Back}_A} \frac{R'(b)}{|\text{Forward}_A|} + c \cdot E(A)$

έτσι ώστε το c να μεγιστοποιείται και το $\|R'\| = 1$ (όπου το $\|R'\|$ είναι το L_1 norm του R')

Αν κάποιος περιηγητής στο Διαδίκτυο μπει σε κάποιον επαναληπτικό βρόγχο ιστοσελίδων (όπως στην εικόνα 2.5), το πιθανότερο είναι ότι θα μεταβεί σε κάποια άλλη ιστοσελίδα, δεν θα συνεχίσει την επανάληψη πάνω στις ίδιες σελίδες. Συνεπώς, το διάνυσμα E μοντελοποιεί αυτή την συμπεριφορά του χρήστη.

2.3.3 Παράγοντας Απόσβεσης (Damping Factor)

Όπως αναφέραμε και στην εισαγωγή του αλγορίθμου *PageRank*, το PageRank μιας σελίδας είναι ουσιαστικά το μοντέλο συμπεριφοράς ενός χρήστη. Ο χρήστης αυτός διαλέγει μια αρχική σελίδα και ακολουθεί τους συνδέσμους, χωρίς να οπισθοχωρεί σε προηγούμενη σελίδα. Στο τέλος όμως, ο χρήστης αυτός βαριέται και ξεκινά σε άλλη τυχαία σελίδα. Ο παράγοντας απόσβεσης είναι ένας παράγοντας ο οποίος περιγράφει την σελίδα και είναι η πιθανότητα ότι ένας χρήστης που βρίσκεται στην εν λόγω σελίδα, θα βαρεθεί και θα επιλέξει άλλη τυχαία σελίδα. Ο Ορισμός 3 λαμβάνει υπόψη τον πιο πάνω παράγοντα απόσβεσης.

Ορισμός 3 $R(A) = (1 - d) + d \cdot \sum_{b \in \text{Back}_A} \frac{R(b)}{|\text{Forward}_A|}$

όπου d είναι ο παράγοντας απόσβεσης.

2.4 Αλγόριθμος Ομαδοποίησης BFR

Ο αλγόριθμος Bradley Fayyad Reina *BFR* είναι αλγόριθμος ομαδοποίησης (*Clustering Algorithm*) ο οποίος χρησιμοποιείται για την ομαδοποίηση μεγάλων δεδομένων.² Η φιλοσοφία του αλγορίθμου είναι ότι εντοπίζει περιοχές δεδομένων οι οποίες μπορούν να συμπεστούν (*Compressed Set*), περιοχές δεδομένων οι οποίες πρέπει να διατηρηθούν στην μνήμη (*Retained Set*) και περιοχές δεδομένων οι οποίες μπορούν να απορριφθούν (*Discard Set*) [1]. Σε περίπτωση που κάποια σημεία συμπιέζονται ή απορρίπτονται πρέπει να διατηρήσουμε κάποια πληροφορία για αυτά πριν την απόρριψη τους. Η πληροφορία (ή Περιλήψεις Σημείων) αυτή είναι η εξής τριάδα:

- Ο αριθμός των σημείων δεδομένων μέσα στο σύνολο, έστω N .
- Το άθροισμα της κάθε διάστασης όλων των σημείων του συνόλου. Μπορούμε να αναπαραστήσουμε αυτή την πληροφορία ως ένα μονοδιάστατο διάνυσμα μήκους d (αριθμός διαστάσεων), έστω SUM .
- Το άθροισμα των τετραγώνων της κάθε διάστασης όλων των σημείων του συνόλου. Και πάλι αυτή η πληροφορία μπορεί να αναπαρασταθεί από ένα διάνυσμα μήκους d , έστω $SUMSQ$.

Η πιο πάνω πληροφορία είναι ένας έμμεσος τρόπος για τον υπολογισμό του κέντρου της κάθε συστοιχίας. Για τον υπολογισμό κέντρου, διαιρούμε το κάθε σημείο του διανύσματος SUM με τον αριθμό των στοιχείων του συνόλου N : SUM_i/N . Επίσης, οι πιο πάνω περιλήψεις χρησιμοποιούνται για τον υπολογισμό της τυπικής απόκλισης σε κάθε διάσταση $\sqrt{SUMSQ_i/N - (SUM_i/N)^2}$ η οποία χρησιμοποιείται για τον υπολογισμό της απόστασης *Mahalanobis*.

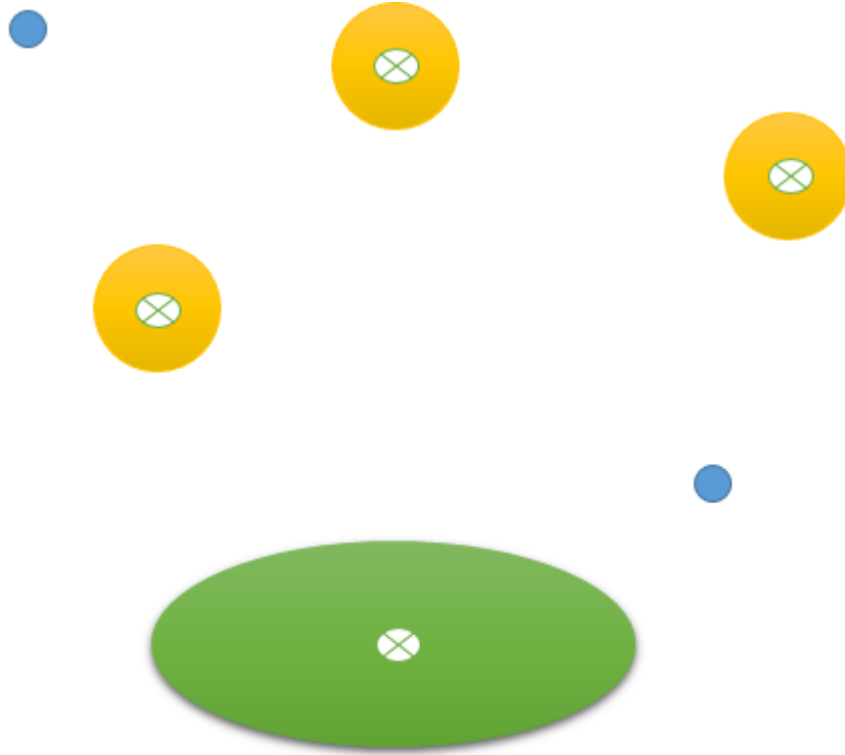
Το Συμπιεσμένο σύνολο (*Compressed Set*) περιέχει περιλήψεις σημείων και όχι τα ίδια τα σημεία δεδομένων. Οι περιλήψεις αυτές είναι για τα σημεία τα οποία βρέθηκαν κοντά μεταξύ τους αλλά όχι κοντά σε κάποια συστοιχία δεδομένων.

Το Διατηρητέο σύνολο (*Retained Set*) είναι σημεία δεδομένων τα οποία δεν μπορούν να ανατεθούν σε κάποια συστοιχία και ούτε είναι κοντά σε κάποιο άλλο σημείο για να

²Αν και δεν υπήρξε άμεση εφαρμογή του αλγορίθμου BFR σε αυτή την Διπλωματική Εργασία, κάποιες βασικές ιδέες του, χρησιμοποιούνται στον αλγόριθμο Προφόρτωσης

σχηματίσουν μαζί μια νέα συστοιχία (και άρα να μεταφερθούν στο συμπιεσμένο σύνολο). Τα σημεία δεδομένων δεν απορρίπτονται από την μνήμη στο Διατηρητέο Σύνολο.

Το Απορριφθέν σύνολο *Discard Set* ονομάζεται έτσι διότι τα πραγματικά σημεία δεδομένων απορρίπτονται από την μνήμη, προσαρμόζοντας αρχικά τις περιλήψεις σημείων, τις τρεις μετρικές N , SUM και $SUMSQ$ δηλαδή.



Σχήμα 2.6: Σύνολα Σημείων του Αλγορίθμου BFR

Όλες οι πιο πάνω ιδέες απεικονίζονται διαγραμματικά στο Σχήμα 2.6. Τα μπλε σημεία είναι το Διατηρητέο σύνολο, το πράσινο σημείο είναι μια από τις συστοιχίες του Απορριφθέν συνόλου, ενώ τα κίτρινα σημεία είναι σημεία του Συμπιεσμένου συνόλου. Όπως αναφέραμε ήδη τα πραγματικά σημεία δεν διατηρούνται για το Συμπιεσμένο και το Απορριφθέν σύνολο. Ο σταυρός απεικονίζει το κέντρο της συστοιχίας (*centroid*). Μια παραλλαγή των ιδεών αυτών χρησιμοποιείται και στον αλγόριθμο Προφόρτωσης (περισσότερα για αυτό στο Κεφάλαιο 4).

Ο αλγόριθμος διαβάζει τεμάχια δεδομένων (*Chunks*) και για κάθε σημείο p :

1. Αν το σημείο p είναι αρκετά κοντά στο κέντρο οποιασδήποτε συστοιχίας, πρόσθεσε το σημείο στην συστοιχία αυτή και προσαρμοσε τα στατιστικά στοιχεία της συστοιχίας N , SUM και $SUMSQ$. Το σημείο p ακολούθως απορρίπτεται.

2. Αν το σημείο p δεν βρίσκεται κοντά σε καμιά συστοιχία το ομαδοποιούμε με τα σημεία του Διατηρητέου συνόλου ,χρησιμοποιώντας ένα οποιοδήποτε αλγόριθμο Ομαδοποίησης.
3. Συγχώνευση των συστοιχιών του Συμπιεσμένου συνόλου και των συστοιχιών που προκύπτουν από το βήμα 2, αν είναι επαρκώς κοντά.

Τέλος, ορίζουμε το "επαρκώς κοντά" ως την απόσταση *Mahalanobis* η οποία είναι η απόσταση ενός σημείου από το κέντρο μιας συστοιχίας, διαιρούμενη με την τυπική απόκλιση της συστοιχίας σε κάθε διάσταση (βλέπε Ορισμό 4).

Ορισμός 4 *Ας υποθέσουμε ένα σημείο x στον d -διάστατο χώρο : $x = [x_1, x_2, \dots, x_n]$ και το κέντρο της συστοιχίας C ως $c = [c_1, c_2, \dots, c_n]$. Επίσης ορίζουμε την τυπική απόκλιση των σημείων της συστοιχίας σε κάθε διάσταση ως σ_i . Τότε, η απόσταση Mahalanobis μεταξύ του σημείου και της συστοιχίας υπολογίζεται με τον εξής τύπο:*

$$\sqrt{\sum_{i=1}^d \left(\frac{x_i - c_i}{\sigma_i} \right)^2}$$

Κεφάλαιο 3

Διατύπωση του Προβλήματος και Μοντέλο Συστήματος

3.1 Μοντέλο Συστήματος	22
3.1.1 Γράφημα Εξαρτήσεων - Κατασκευή και Εκπαίδευση	24
3.2 Διατύπωση του Προβλήματος	27

3.1 Μοντέλο Συστήματος

Σε αυτό το υπό-κεφάλαιο, θα οριστεί η βασική ορολογία και υποθέσεις πάνω στις οποίες βασίζεται η διατύπωση του προβλήματος αλλά και ο αλγόριθμοι. Τα κύρια σύμβολα αλλά και οι ορισμοί τους, ορίζονται περιληπτικά στον Πίνακα 3.1. Υποθέτουμε ότι μια περιοχή A καλύπτεται από ένα σύνολο από Σημεία Πρόσβασης Ασύρματων Δικτύων *Wifi APs*. Αυτή η περιοχή δεν είναι απαραίτητα συνεχόμενη, και δεν υποθέτουμε κάποια συγκεκριμένη κατανομή των Σημείων Πρόσβασης στο σύνολο $AP = \{ AP_1, AP_2, \dots, AP_n \}$. Κάθε Σημείο Πρόσβασης AP_i του συνόλου AP έχει ένα μοναδικό αναγνωριστικό (το BSSID ή MAC Address της συσκευής δικτύου) που μεταδίδεται και είναι δημόσια διαθέσιμο. Στον πραγματικό κόσμο, η περιοχή A μπορεί να περιλαμβάνει πολλές περιοχές και εσωτερικά κτίρια σε οποιαδήποτε γωνιά του πλανήτη. Τα προβλήματα επεκτασιμότητας θα αναλυθούν περαιτέρω σε επόμενο κεφάλαιο, αφού επεξηγηθεί η αρχιτεκτονική του εξυπηρετητή, ο οποίος κάνει χρήση του κατανεμημένου συστήματος αρχείων *HDFS* που είναι κατάλληλο για σύνολα δεδομένων μεγάλου όγκου.

Ένας χρήστης έξυπνης κινητής συσκευής μέσα σε κάποια περιοχή A μπορεί να χρησιμοποιήσει την κεραία Ασύρματου Δικτύου της συσκευής του για να μετρήσει την Ισχύ

Σύμβολο	Ορισμός
A	Περιοχή Πλοήγησης
AP	Σύνολο Σημείων Πρόσβασης Ασύρματων Δικτύων
$u \in U$	Σύνολο χρηστών Έξυπνων Κινητών Συσκευών
S	Εξυπηρετητής
R	Σύνολο των Τιμών Ισχύος για την περιοχή A
C	Σύνολο Συστοιχιών για την περιοχή A
T	Σύνολο Τροχιών όλων των Χρηστών
G	Γράφημα Εξαρτήσεων
V	Σύνολο κόμβων/συστοιχιών
E	Σύνολο ακμών/μεταβάσεων
W	Παράθυρο Εκπαίδευσης

Πίνακας 3.1: Βασική Ορολογία Μοντέλου Συστήματος

Λαμβανόμενου Σήματος (*RSS*) και να πάρει το αναγνωριστικό (*BSSID*) κάποιου σημείου πρόσβασης AP_i που ανήκει στο σύνολο AP . Κάθε χρήστης u μπορεί να επικοινωνήσει με τον εξυπηρετητή S μέσω Ασύρματου Δικτύου *WiFi* ή σύνδεσης Κινητών δεδομένων (*Mobile Data*). Ο εξυπηρετητής S περιλαμβάνει τον *Χάρτη Τιμών Ισχύος (Radiomap)* της περιοχής A , ο οποίος είναι ένας πίνακας δύο διαστάσεων που εκφράζει την σχέση μεταξύ μοναδικών συντεταγμένων της περιοχής A και των τιμών Ισχύος Λαμβανόμενου Σήματος (*RSS*), για κάθε $AP_i \in AP$.

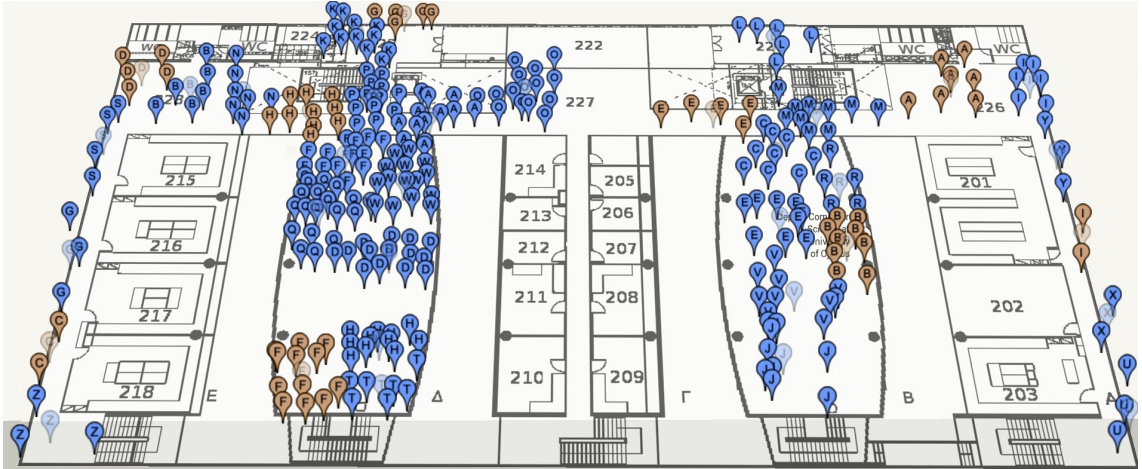
Η συλλογή των τιμών ισχύος (*RSS Fingerprints*) R για την περιοχή A έγινε εκ των προτέρων και για τον υπολογισμό της τοποθεσίας του χρήστη χρησιμοποιούνται υφιστάμενοι αλγόριθμοι γεωτοποθέτησης [14]. Επίσης υποθέτουμε τον διαμελισμό του συνόλου R σε συστοιχίες (*clusters*) τιμών ισχύος (*RSS Fingerprints*). Ο διαμελισμός του συνόλου R είναι γεωγραφικός, δηλαδή γίνεται ομαδοποίηση των τιμών ισχύος βάσει των συντεταγμένων. Επομένως, συντεταγμένες οι οποίες βρίσκονται κοντά στον Ευκλείδειο Χώρο (*Euclidean Space*) σχηματίζουν μαζί μια συστοιχία. Ορίζουμε ως C_i το σύνολο των τιμών ισχύος που απαρτίζουν την συστοιχία i . Όπως φαίνεται και στον Πίνακα 3.2 ο ραδιοχάρτης διαιρείται σε πιο μικρούς ραδιοχάρτες, ενώ στο Σχήμα 3.1 φαίνεται το πραγματικό περιβάλλον αξιολόγησης του αλγορίθμου¹.

¹Στο Σχήμα 1.1 κάθε διαφορετικό χρώμα και γράμμα αντιπροσωπεύει μια διαφορετική συστοιχία σημείων

Access Points					
ClusterID		AP_1	AP_2	...	AP_M
1	x_1, y_1	RSS_1	RSS_2	...	RSS_M
	x_2, y_2	RSS_1	RSS_2	...	RSS_M
	x_3, y_3	RSS_1	RSS_2	...	RSS_M
2	x_4, y_4	RSS_1	RSS_2	...	RSS_M
	x_5, y_5	RSS_1	RSS_2	...	RSS_M

Πίνακας 3.2: Μερικοί Ραδιοχάρτες

Επιπρόσθετα, ο αλγόριθμος που προτείνεται σε αυτή την διπλωματική εργασία, προϋποθέτει την εκπαίδευση του χρησιμοποιώντας το ιστορικό των τροχιών όλων των χρηστών που κινούνται σε ένα εσωτερικό χώρο της περιοχής A . Ορίζουμε το σύνολο των τροχιών των χρηστών ως T . Το σύνολο αυτό είναι ακολουθίες ακεραίων, όπου κάθε ακεραίος αντιπροσωπεύει την διέλευση του χρήστη μέσα από μια συστοιχία σημείων. Ένα παράδειγμα συνόλου τροχιών φαίνεται στο Σχήμα 3.2.



Σχήμα 3.1: Computer Science Building/Floor 2

3.1.1 Γράφημα Εξαρτήσεων - Κατασκευή και Εκπαίδευση

Οι ιστορικές τροχιές των χρηστών, χρησιμοποιούνται για την κατασκευή ενός γραφήματος εξαρτήσεων (*Dependency Graph*) G , ο οποίος περιγράφει πιθανολογικά τις πιο πιθανές διαδρομές, δεδομένου της τοποθεσίας του χρήστη. Πιο τυπικά, $G=(V,E)$ όπου $V=\{v_1, v_2, \dots, v_n\}$ είναι το σύνολο των κόμβων του γραφήματος. Οι κόμβοι αντιπροσωπεύουν τις συ-

1,2,3,4
4,3,2,1
4,3,2
3,2,1
4,3
2,1
1,2
4,1

Σχήμα 3.2: Ιστορικό Τροχιών/User Sessions

στοιχίες από τις οποίες διέρχονται οι χρήστες της περιοχής A . Επιπλέον, οι ακμές $E = \{e_1, e_2, \dots, e_n\}$ αποτελούν τις μεταβάσεις των χρηστών από μια συστοιχία σημείων/κόμβο σε μια άλλη/άλλο. Επομένως, μια ακμή περιγράφει την πιθανότητα μετάβασης από τον κόμβο προέλευσης προς τον κόμβο προορισμού. Για παράδειγμα, η ακμή $e_{i,j}$ (με κόμβο προορισμού j και κόμβο προέλευσης i) με $\Pr(e_{i,j})=0.6$, ισχυρίζεται ότι η πιθανότητα μετάβασης ενός χρήστη που βρίσκεται στην συστοιχία i , στην συστοιχία j είναι 60%. Η πιθανότητα που περιγράφει μιαν ακμή είναι η υπό συνθήκη πιθανότητα (*Conditional Probability*) $\Pr(i | j) = \frac{f(i \cap j)}{f(j)}$ και είναι το πηλίκο της συχνότητας των μεταβάσεων από την συστοιχία i προς την συστοιχία j , ως προς την συχνότητα διέλευσης των χρηστών από την συστοιχία i . Το πιο κάτω παράδειγμα αποσαφηνίζει όλους τους πιο πάνω ορισμούς. Δεδομένου του ιστορικού τροχιών στο Σχήμα 3.2 υπολογίζονται οι συχνότητες των κόμβων (βλέπε Πίνακα 3.3) αλλά και οι συχνότητες της κάθε μετάβασης (βλέπε Πίνακα 3.4) .

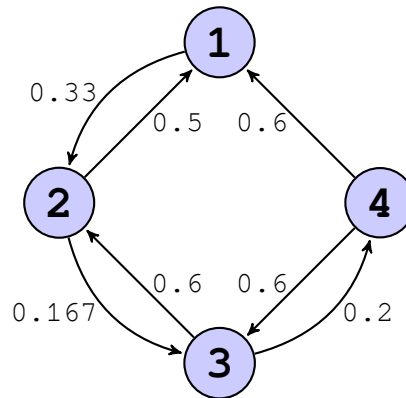
Συνεπώς, για τον υπολογισμό της υπό συνθήκης πιθανότητας $\Pr(1 | 2) = \frac{f(1 \cap 2)}{f(2)}$ διαιρούμε την συχνότητα των μεταβάσεων από τον κόμβο 1 στον κόμβο 2 με την συχνότητα του κόμβου 1. Επομένως $\Pr(1 | 2) = \frac{f(1 \cap 2)}{f(1)} = \frac{2}{6}$. Το γράφημα εξαρτήσεων μετά τον υπολογισμό όλων των πιθανοτήτων φαίνεται στο Σχήμα 3.3.

Κόμβος	Συχνότητα
1	6
2	6
3	5
4	5

Πίνακας 3.3: Συχνότητες Κόμβων

Μετάβαση	Συχνότητα
1->2	2
2->3	1
3->4	1
4->3	3
3->2	3
2->1	3
4->1	3

Πίνακας 3.4: Συχνότητες Μεταβάσεων



Σχήμα 3.3: Γράφημα Εξαρτήσεων/Dependency Graph

Το πιο πάνω γράφημα εξαρτήσεων χρησιμοποιείται από τον αλγόριθμο Προβλέψεων (*Prediction Algorithm*) για να εικάσει ποια είναι η πιο πιθανή επόμενη συστοιχία διέλευσης του χρήστη. Περισσότερα για τον αλγόριθμο Πρόβλεψης στο Κεφάλαιο 4. Τέλος, ορίζουμε ως W το παράθυρο (*Lookahead Window*) το οποίο είναι ένας φυσικός ακέραιος αριθμός $W \in \mathbb{N}$ που αντιπροσωπεύει το βάθος στο οποίο γίνεται η εκπαίδευση του πιο πάνω γραφήματος. Για παράδειγμα για $W=2$ και την τροχιά 1,2,3 συμπεριλαμβάνεται και η μετάβαση 1->3 εκτός από τις 1->2 και 2->3, στην εκπαίδευση του γραφήματος. Σημειώνεται ότι για την εκπαίδευση του γραφήματος στο Σχήμα 3.3 έχουμε υποθέσει $W=1$.

3.2 Διατύπωση του Προβλήματος

Καθώς ένας κάτοχος Έξυπνης Κινητής Συσκευής, κινείται στο εσωτερικό ενός κτιρίου της περιοχής A η σύνδεση του στο διαδίκτυο είναι διακοπτόμενη (Intermittent Connectivity). Αυτό αποτελεί πρόβλημα, εφόσον ο χρήστης πρέπει να βρίσκεται σε επικοινωνία με τον εξυπηρετητή S ο οποίος έχει στην κατοχή του τους ραδιοχάρτες του κτιρίου. Για την αντιμετώπιση του προβλήματος αυτού προτείνουμε ένα αλγόριθμο Προφόρτωσης των ραδιοχαρτών. Η προφόρτωση γίνεται στο επίπεδο της συστοιχίας. Στο ένα άκρο, η προφόρτωση των τιμών ισχύος θα μπορούσε να γίνεται στο επίπεδο του ενός γεωγραφικού σημείου. Παρόλα αυτά, η προσέγγιση αυτή δεν είναι αποδοτική καθώς η κινητή συσκευή θα έπρεπε να βρίσκεται σε συνεχή επικοινωνία με τον εξυπηρετητή S . Δεδομένου ότι η σύνδεση είναι διακοπτόμενη, η προηγούμενη προσέγγιση δεν θα δούλευε. Στο άλλο αντίθετο άκρο, η προφόρτωση θα μπορούσε να γίνεται στο επίπεδο όλων των συστοιχιών ενός κτιρίου (και συνεπώς ολόκληρο τον ραδιοχάρτη). Και πάλι αυτή η προσέγγιση δεν είναι αποδοτική από πλευράς χρόνου και κατανάλωσης ενέργειας, καθώς το μέγεθος των ραδιοχαρτών για μεγάλα κτίρια είναι σημαντικά μεγάλο [12]. Επομένως η προφόρτωση γίνεται στο επίπεδο της συστοιχίας χρησιμοποιώντας μια Μηχανή Προβλέψεων (*Prediction Engine*) βασισμένη πάνω σε ένα γράφημα εξαρτήσεων (Σχήμα 3.3). Με αυτό τον τρόπο επιτυγχάνουμε όσο το δυνατόν λιγότερη επικοινωνία με τον εξυπηρετητή και μείωση του απαιτούμενου χρόνου εξυπηρέτησης ενός χρήστη. Παράλληλα, ο χρήστης μπορεί να κινείται επιτυχώς μέσα στον χώρο, χωρίς συνεχή επικοινωνία με τον εξυπηρετητή. Συνεπώς, ο αλγόριθμος Προφόρτωσης που προτείνουμε είναι αποτελεσματικός για περιβάλλοντα με διακοπτόμενη σύνδεση στο δίκτυο.

Κεφάλαιο 4

Αλγόριθμοι Προφόρτωσης Ραδιοχαρτών

4.1	Community Based Algorithm - CBA	29
4.1.1	Επεξήγηση Αλγορίθμου CBA	29
4.1.2	Ανάλυση Πολυπλοκότητας Αλγορίθμου CBA	31
4.2	Building Based Algorithm - BBA	33
4.2.1	Επεξήγηση Αλγορίθμου BBA	33
4.2.2	Ανάλυση Πολυπλοκότητας Αλγορίθμου BBA	38
4.3	Αλγόριθμος Μηχανής Πρόβλεψης	38
4.3.1	Επεξήγηση Αλγορίθμου	38
4.3.2	Αλγόριθμος Μηχανής Πρόβλεψης-Ανάλυση Πολυπλοκότητας	42
4.4	Αλγόριθμος Μηχανής Προφόρτωσης	43
4.5	Παράδειγμα Εκτέλεσης Αλγορίθμου Πρόβλεψης και Προφόρτωσης	47

Οι δύο αλγόριθμοι που προτείνονται σε αυτή την διπλωματική εργασία είναι ο *Community Based Algorithm (CBA)* και ο *Building Based Algorithm (BBA)*. Ο αλγόριθμος CBA προσπαθεί να προβλέψει τις μελλοντικές συστοιχίες των χρηστών βάσει του υφιστάμενου ιστορικού κίνησης των χρηστών. Για την εκπαίδευση του αλγορίθμου χρησιμοποιείται ένα σύνολο τροχιών και η εκπαίδευση γίνεται όπως περιγράφεται στο υπό-κεφάλαιο 3.1.1. Αντίθετα, ο δεύτερος αλγόριθμος BBA, χρησιμοποιεί τον αλγόριθμο PageRank για να αναθέσει μια σημαντικότητα σε κάθε συστοιχία (κόμβο). Στην πρώτη περίπτωση γίνεται προφόρτωση της συστοιχίας σημείων που έχει την μεγαλύτερη πιθανότητα μετάβασης από κάποια συστοιχία προέλευσης, ενώ στην δεύτερη περίπτωση επιλέγεται η συστοιχία με την μεγαλύτερη σημαντικότητα δεδομένου μιας συστοιχίας προέλευσης.

Σύμβολο	Επεξήγηση
t	Τροχιά Χρήστη $t_1, t_2, \dots, t_n$. Μια ακολουθία από ακέραιους αριθμούς
t_i	Μεμονωμένη Ταυτότητα Συστοιχίας
V	Σύνολο Κορυφών
FV	Πίνακας Συχνοτήτων Κορυφών
E	Σύνολο Κορυφών
FE	Πίνακας Συχνοτήτων Ακμών
$e_{i,j}$	Ακμή από τον κόμβο i στον κόμβο j
$w(e_{i,j})$	Πιθανότητα Μετάβασης από τον κόμβο i στον κόμβο j
G	Γράφημα Εξαρτήσεων

Πίνακας 4.1: Συμβολισμός Αλγορίθμου 1

4.1 Community Based Algorithm - CBA

Ο CBA αλγόριθμος κατασκευάζει ένα γράφημα εξαρτήσεων το οποίο απεικονίζει τις συνήθειες των χρηστών κατά την διάρκεια της πλοήγησης τους σε ένα εσωτερικό χώρο. Το γράφημα εξαρτήσεων περιέχει ένα κόμβο για κάθε συστοιχία που έχει προσπελαστεί από κάποιο χρήστη, οποιαδήποτε στιγμή στο παρελθόν. Υπάρχει μια ακμή στο γράφημα εάν και μόνο εάν κάποια στιγμή στο παρελθόν, κάποιος χρήστης μετέβη από μια συστοιχία προέλευσης A σε μια συστοιχία προορισμού B , διαμέσου W προσβάσεων σε άλλες συστοιχίες του χώρου. Ο ίδιος αλγόριθμος χρησιμοποιείται και για την προφόρτωση σελίδων στο Διαδίκτυο (*Web Prefetching*) [6]. Ο Αλγόριθμος 1 είναι ο αλγόριθμος κατασκευής του γραφήματος εξαρτήσεων για τον αλγόριθμο CBA.

4.1.1 Επεξήγηση Αλγορίθμου CBA

Ο Αλγόριθμος 1 αποτελείται από τρεις φάσεις. Και οι τρεις φάσεις υποθέτουν τον συμβολισμό που περιγράφεται στον Πίνακα 4.1. Στην πρώτη φάση, εκτελείται ένα πέρασμα πάνω σε όλες τις τροχιές των χρηστών (βλέπε γραμμή 6), όπου το σύνολο των κόμβων κατασκευάζεται και οι συχνότητες των κόμβων υπολογίζονται. Στο δεύτερο πέρασμα πάνω από τις τροχιές (βλέπε γραμμή 14), προστίθενται στο σύνολο των ακμών, ακμές βάθους 1 μέχρι W . Τέλος, στο τρίτο πέρασμα των τροχιών (βλέπε γραμμή 25),

υπολογίζονται τα βάρη των ακμών τα οποία είναι ουσιαστικά οι πιθανότητες μετάβασης από ένα κόμβο/συστοιχία σε άλλο/άλλη.

Αλγόριθμος 1 Κατασκευή Γραφήματος Εξαρτήσεων - CBA

Input: Το Σύνολο Τροχιών όλων των Χρηστών T , Το Παράθυρο Πρόβλεψης W

Output: Το γράφημα εξαρτήσεων G

1: **procedure** CBA(T, W)

2: $V = \{\}$ ▷ V : οι κόμβοι του γραφήματος

3: $FV = \{\}$ ▷ FV : οι συχνότητες των κόμβων του γραφήματος

4: $E = \{\}$ ▷ E : οι ακμές του γραφήματος

5: $FE = \{\}$ ▷ FE : οι συχνότητες των ακμών του γραφήματος

Πρώτο Πέρασμα

6: **for** t in T **do** ▷ t : μια τροχιά, μια ακολουθία από ταυτότητες συστοιχιών

7: **if** $t_i \notin V$ **then** ▷ t_i : μια ταυτότητα συστοιχίας

8: $V = V \cup t_i$

9: $FV[t_i] = 1$

10: **else**

11: $FV[t_i] = FV[t_i] + 1$

12: **end if**

13: **end for**

Πριν το οποιοδήποτε πέρασμα πάνω στις τροχιές των χρηστών αρχικοποιούνται τέσσερα σύνολα V , FV , E και FE . Το σύνολο V είναι το σύνολο των κόμβων, το σύνολο E είναι όλες οι ακμές του γραφήματος. Μπορούμε να φανταστούμε τα σύνολα FV και FE ως μονοδιάστατους πίνακες που κρατούν τις συχνότητες του κάθε κόμβου και της κάθε ακμής.

Αλγόριθμος 1 Κατασκευή Γραφήματος Εξαρτήσεων - CBA (Συνέχεια)

Δεύτερο Πέρασμα

```
14:   for  $j \leftarrow 1, W$  do
15:       for  $t$  in  $T$  do           ▷  $t$ :μια τροχιά, μια ακολουθία από ταυτότητες συστοιχιών
16:           if  $t_{i,i+j} \notin E$  then
17:                $E = E \cup t_{i,i+j}$ 
18:                $FE[t_{i,i+j}] = 1$ 
19:           else
20:                $FE[t_{i,i+j}] = FE[t_{i,i+j}] + 1$ 
21:           end if
22:       end for
23:   end for
```

Στην δεύτερη φάση του αλγορίθμου, αυξάνονται οι συχνότητες της κάθε μετάβασης/ακμής, για διάφορα μεγέθη παραθύρου. Υπενθυμίζεται ότι με ένα παράθυρο μεγέθους δύο, οι κόμβοι προορισμού και προέλευσης δύο περιέχουν ένα ενδιάμεσο κόμβου μεταξύ τους. Η ίδια λογική εφαρμόζεται για κάθε μέγεθος παραθύρου μέχρι το W .

Αλγόριθμος 1 Κατασκευή Γραφήματος Εξαρτήσεων - CBA (Συνέχεια)

Τρίτο Πέρασμα

```
24:   for  $e_{i,j}$  in  $E$  do           ▷  $e$ :ακμή/μετάβαση
25:       ▷ Διαίρεσε την συχνότητα μετάβασης με την συχνότητα του κόμβου προέλευσης
26:        $w(e_{i,j}) = FE(e_{i,j})/FV(e_i)$  ▷  $w(e_{i,j})$ :Βάρος ακμής/Πιθανότητα Μετάβασης
27:   end for
28:   return  $G$                        ▷  $G$ :το γράφημα εξαρτήσεων
29: end procedure
```

4.1.2 Ανάλυση Πολυπλοκότητας Αλγορίθμου CBA

Ανάλυση Χρονικής Πολυπλοκότητας Πρώτης Φάσης

Ας θέσουμε το μέγεθος της τροχιάς ενός χρήστη ως $|t|$ και το μέγεθος του συνόλου τροχιών ως $|T|$. Η φάση δημιουργίας των κόμβων του γραφήματος, διατρέχει όλες τις ταυτότητες των συστοιχιών t_i για κάθε τροχιά του συνόλου T . Επομένως η χρονική πολυπλοκότητα του αλγορίθμου είναι τετράγωνη. Πιο συγκεκριμένα είναι το γινόμενο του μεγέθους

της τροχιάς του χρήστη ($|t|$) με τον συνολικό αριθμό τροχιών ($|T|$). Σε συμβολισμό Ασυμπτωτικής Ανάλυσης η πολυπλοκότητα είναι $\mathcal{O}(|T| \cdot |t|)$.

Ανάλυση Χρονικής Πολυπλοκότητας Δεύτερης Φάσης

Η πολυπλοκότητα της δεύτερης φάσης είναι παρόμοια με αυτή της πρώτης φάσης με την διαφορά ότι προσθέτουμε ακμές στο σύνολο E για διάφορα μεγέθη παραθύρου. Το τυπικό μέγεθος είναι γύρω στο 3. Σημειώνεται ότι παράθυρα μεγέθους γύρω στο 5 μπορούν να προ-φορτώσουν τις τιμές ισχύος του μισού ορόφου, κάτι το οποίο δεν συνιστάται για λόγους επίδοσης. Επομένως, δεδομένου του μικρού μεγέθους του παραθύρου μπορούμε να το απαλείψουμε από την ανάλυση πολυπλοκότητας για την δεύτερη φάση. Συνεπώς η χρονική πολυπλοκότητα του αλγορίθμου, είναι το γινόμενο του μεγέθους της τροχιάς του χρήστη ($|t|$) με τον συνολικό αριθμό τροχιών ($|T|$). Σε συμβολισμό Ασυμπτωτικής Ανάλυσης η πολυπλοκότητα είναι $\mathcal{O}(|T| \cdot |t|)$.

Ανάλυση Χρονικής Πολυπλοκότητας Τρίτης Φάσης

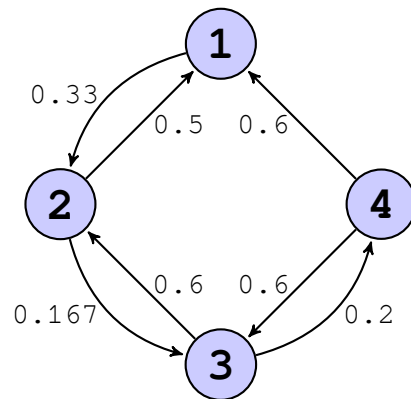
Στην τρίτη φάση διατρέχουμε όλες τις ακμές και υπολογίζουμε το βάρος της κάθε μιας ακμής. Η πολυπλοκότητα είναι γραμμική και σε συμβολισμό Ασυμπτωτικής Ανάλυσης η πολυπλοκότητα είναι $\mathcal{O}(|E|)$, όπου E είναι ο συνολικός αριθμός ακμών του γραφήματος.

Η συνολική πολυπλοκότητα του αλγορίθμου CBA είναι $\mathcal{O}(|T| \cdot |t|) + \mathcal{O}(|T| \cdot |t|) + \mathcal{O}(|E|) = 2 \cdot \mathcal{O}(|T| \cdot |t|) + \mathcal{O}(|E|) \approx \mathcal{O}(|T| \cdot |t|)$.

Παράδειγμα Πλοήγησης χρήστη με τον αλγόριθμο CBA

Όπως βλέπουμε στο Σχήμα 4.2 το γράφημα εξαρτήσεων περιέχει μια πιθανότητα για κάθε μετάβαση του χρήστη από ένα δεδομένο κόμβο σε κάποιο άλλο. Αν υποθέσουμε ότι ο χρήστης βρίσκεται στον κόμβο 3, τότε ο χρήστης είτε θα μεταβεί στην περιοχή (συστοιχία) 2 είτε θα μεταβεί στην περιοχή 4. Οι συνήθειες των χρηστών όμως μας υποδεικνύουν ότι ο χρήστης είναι πιθανότερο να μεταβεί στην συστοιχία 2 με πιθανότητα 60%. Ο αλγόριθμος Προβλέψεων θα προτείνει τον κόμβο 2 ως την επόμενη πιο πιθανή συστοιχία του χρήστη. Σε περίπτωση που το βάθος της Προφόρτωσης είναι δύο, ο αλγόριθμος Προβλέψεων θα επιστρέψει στην ίδια αίτηση και την συστοιχία 1 (η οποία είναι η πιο πιθανή επόμενη συστοιχία μετά τον κόμβο 2).

1,2,3,4
4,3,2,1
4,3,2
3,2,1
4,3
2,1
1,2
4,1



Σχήμα 4.1: Ιστορικό Τροχιών

Σχήμα 4.2: Γράφημα Εξαρτήσεων CBA

4.2 Building Based Algorithm - BBA

4.2.1 Επεξήγηση Αλγορίθμου BBA

Ο Building Based αλγόριθμος μιμείται τον αλγόριθμο *PageRank*, ο οποίος χρησιμοποιείται από την *Google* [2,15] για να βαθμολογήσει τις σελίδες που εμφανίζονται στα αποτελέσματα αναζήτησης της. Στην περίπτωση του BBA, δέχεται ένα γράφημα το οποίο αντιπροσωπεύει τις μεταβάσεις των χρηστών από συστοιχία σε συστοιχία μέσα σε ένα εσωτερικό χώρο. Δεδομένου του γραφήματος αυτού, αναθέτει σε κάθε συστοιχία/κόμβο μια σημαντικότητα. Η σημαντικότητα αυτή, αντιπροσωπεύει την πιθανότητα ένας τυχαίος χρήστης κινητής συσκευής να βρεθεί στην συγκεκριμένη συστοιχία/κόμβο. Η πιθανότητα αυτή ονομάζεται *παράγοντας απόσβεσης (Damping Factor)*. Ας θέσουμε τον Παράγοντα Απόσβεσης ως d . Ένας τυχαίος χρήστης κινητής συσκευής επιλέγει μια τυχαία ακμή με πιθανότητα d , ενώ με πιθανότητα $1-d$ μεταφέρεται σε έναν τυχαίο κόμβο (ο οποίος πιθανόν να μην συνδέεται με τον τρέχον κόμβο του).

Αλγόριθμος 2 Building Based Algorithm - BBA

Input: Το Γράφημα Εξαρτήσεων G

Output: Το Γράφημα σημαντικότητας G' , όπου κάθε κόμβος πλέον έχει μια σημαντικότητα

1: **procedure** BBA(G)

Φάση Αρχικοποίησης

2: $I[] = \{\}$ ▷ I : πίνακας με την σημαντικότητα του κάθε κόμβου

3: $newRanks[] = \{\}$

4: $damping_factor = 0.85$

5: $precision = 1 * 10^{-5}$

6: $iterationCount = 0$

7: $normDiff = 0$

8: $initial_rank = 1/|V|$ ▷ $|V|$: ο συνολικός αριθμός των κόμβων του γραφήματος

9: **for** v in V **do** ▷ v : τρέχων κόμβος, V : οι κόμβοι του γραφήματος G

10: $I[v] = initial_rank$

11: **end for**

Φάσης Αρχικοποίησης

Σε αυτή τη φάση συμβαίνουν οι απαραίτητες αρχικοποιήσεις. Ο παράγοντας απόσβεσης αρχικοποιείται ως 0.85. Η αρχική τιμή αυτή προτείνεται από τους Brin & Page στο [2]. Ακολούθως ορίζεται μια αρχική σημαντικότητα για κάθε κόμβο του γραφήματος ($1/|V|$), όπου $|V|$ είναι ο αριθμός των κόμβων του γραφήματος. Περισσότερες λεπτομέρειες για τον συμβολισμό βρίσκονται στον Πίνακα 4.2. Επίσης ορίζεται η ακρίβεια του αλγορίθμου ($precision$). Αυτή η μεταβλητή καθορίζει ουσιαστικά τον αριθμό των επαναλήψεων που θα τρέξει ο αλγόριθμος BBA μέχρι το γράφημα να συγκλίνει.

Αλγόριθμος 2 Building Based Algorithm - BBA (Συνέχεια)

Δεύτερη Φάση

```
12:   repeat
13:      $dampingTerm = (1 - dampingFactor)/|V|$ 
14:      $danglingRank = 0$ 
15:      $newRanks[] = \{\}$ 
16:     for  $v$  in  $V$  do           ▷  $v$ :τρέχων κόμβος,  $V$ :οι κόμβοι του γραφήματος  $G$ 
17:        $sum = 0$ 
18:       for  $e_{i,v}$  in  $E$  do       ▷  $e_{i,v}$ :κάθε εισερχόμενη ακμή του κόμβου  $v$ 
                                   ▷  $i$  είναι ο κόμβος προέλευσης της ακμής  $e_{i,v}$ 
19:          $sum+ = I[i]/Out(i)$      ▷  $e_{i,v}$ :υπολόγισε την σημαντικότητα
                                   ▷ κόμβου  $v$ 
                                   ▷  $Out(i)$  αριθμός εξερχόμενων ακμών του κόμβου  $i$ 
20:   end for
```

Δεύτερη Φάση

Ο υπολογισμός στην γραμμή 13 αποτελεί τον υπολογισμό του παράγοντα απόσβεσης (*Damping Factor*). Είναι η πιθανότητα μετάβασης του περιηγητή σε έναν τυχαίο κόμβο (τηλέμεταφορά), χωρίς να ακολουθήσει κάποια από τις εξερχόμενες ακμές του. Κάτι τέτοιο βέβαια δεν ισχύει στην περίπτωση της γεωτοποθέτησης, παρόλα αυτά ο υπολογισμός του παράγοντα απόσβεσης είναι απαραίτητος διότι λαμβάνει υπόψη τα αδιέξοδα. Αν ο αλγόριθμος δεν λάβει υπόψη τα αδιέξοδα στο γράφημα, τότε οι αδιέξοδοι κόμβοι θα συσσωρεύουν σημαντικότητα, χωρίς να δίνουν σημαντικότητα σε άλλους κόμβους (περιοχές). Αυτό έχει ως αποτέλεσμα η σημαντικότητα των κόμβων του γραφήματος να διοχετεύεται έξω από το γράφημα. Προφανώς αυτό σημαίνει και λανθασμένη ανάθεση σημαντικότητας σε κάθε κόμβο. Περαιτέρω, τα αδιέξοδα υπάρχουν και πρέπει να λαμβάνονται υπόψη σε σενάρια γεωτοποθέτησης, για παράδειγμα η έξοδος ενός κτιρίου.

Στην γραμμή 16 διατρέχονται όλοι οι κόμβοι του γραφήματος και για κάθε κόμβο v αθροίζουμε την σημαντικότητα του κάθε κόμβου (έστω i) που δείχνει σε αυτόν. Κάθε κόμβος i δεν δίνει όλη την σημαντικότητα του στον κόμβο v , διότι δείχνει και σε άλλους κόμβους στο γράφημα. Αντίθετα παραχωρεί μόνο το $1/Out(i)$ της σημαντικότητας του στον κόμβο v , όπου $Out(i)$ είναι ο αριθμός των εξερχόμενων ακμών του κόμβου i .

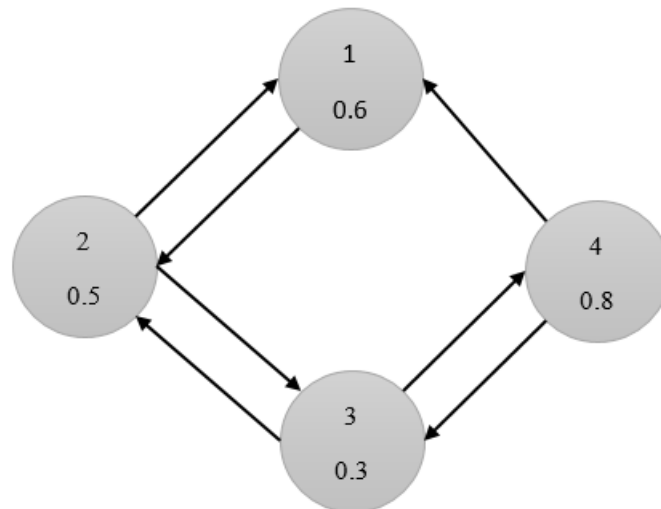
Αλγόριθμος 2 Building Based Algorithm - BBA (Συνέχεια)

```
21:       $rank = dampingTerm + dampingFactor * sum$ 
22:       $newRanks[v] = rank$ 
23:      if  $Out(v) == 0$  then  $\triangleright Out(v)$ :αριθμός εξερχόμενων ακμών του κόμβου
       $v$ 
24:           $danglingRank+ = I[v]$ 
25:      end if
26:  end for
27:   $danglingRank* = dampingFactor / |V|$ 
28:   $normDiff = 0$ 
29:  for  $v$  in  $V$  do
30:       $currentRank = I[v]$ 
31:       $newRank = newRanks[v] + danglingRank$ 
32:       $normDiff+ = |newRank - currentRank|$ 
33:       $I[v] = newRank$ 
34:  end for
35:   $iterationCount = iterationCount + 1$ 
36:  until  $normDiff > precision$ 
37:  return  $G'$   $\triangleright G'$ :το γράφημα σημαντικότητας
38: end procedure
```

Σύμβολο	Επεξήγηση
damping_factor	Παράγοντας Απόσβεσης
iterationCount	Μετρητής Επαναλήψεων
precision	Ακρίβεια Αλγορίθμου PageRank
normDiff	Απόκλιση από την καθορισμένη ακρίβεια
initial_rank	Αρχική Σημαντικότητα Κόμβου
V	Σύνολο Κορυφών
E	Σύνολο Ακμών
V	Συνολικός Αριθμός Κορυφών
E	Συνολικός Αριθμός Ακμών

Πίνακας 4.2: Συμβολισμός Αλγορίθμου 1

Παράδειγμα Πλοήγησης χρήστη με τον αλγόριθμο BBA



Σχήμα 4.3: Παράδειγμα Πλοήγησης σε γράφημα σημαντικότητα BBA

Όπως βλέπουμε στο Σχήμα 4.2, κάθε κόμβος χαρακτηρίζεται από μια σημαντικότητα. Ας υποθέσουμε ότι ο χρήστης βρίσκεται στον κόμβο τέσσερα. Ο αλγόριθμος BBA προτείνει ως τον πιο πιθανό επόμενο κόμβο, εκείνο με σημαντικότητα 0.6. Και πάλι μπορεί να γίνει πρόβλεψη πάνω στην πρόβλεψη σε περίπτωση που το βάθος προφόρτωσης είναι 2. Σε αυτή την περίπτωση στην ίδια αίτηση του χρήστη, ο BBA θα πρότεινε τις συστοιχίες με σημαντικότητα 0.6 και 0.5.

4.2.2 Ανάλυση Πολυπλοκότητας Αλγορίθμου BBA

Ανάλυση Χρονικής Πολυπλοκότητας Πρώτης Φάσης

Η χρονική πολυπλοκότητα της πρώτης φάσης είναι γραμμική. Πιο συγκεκριμένα, στην φάση αρχικοποίησης διατρέχουμε όλους τους κόμβους του γραφήματος και αναθέτουμε μια αρχική σημαντικότητα. Επομένως, η χρονική πολυπλοκότητα της πρώτης φάσης είναι $\mathcal{O}(|V|)$.

Ανάλυση Χρονικής Πολυπλοκότητας Δεύτερης Φάσης

Κάθε επανάληψη παίρνει χρόνο $\mathcal{O}(|V| \cdot |\mathcal{E}|)$, καθώς η επανάληψη στην γραμμή 16 εκτελείται για όλους τους κόμβους του γραφήματος και για κάθε τέτοιο κόμβο λαμβάνονται υπόψη οι εισερχόμενες ακμές του (βλέπε 18). Επομένως, στην χειρίστη περίπτωση ο αριθμός των ακμών που ελέγχονται είναι $|\mathcal{E}|$. Τέλος, ο αριθμός των επαναλήψεων του αλγορίθμου δεν είναι γνωστός. Ορίζοντας ως R τον συνολικό αριθμό επαναλήψεων του αλγορίθμου η χρονική πολυπλοκότητα του αλγορίθμου είναι $\mathcal{R} \cdot \mathcal{O}(|V| \cdot |\mathcal{E}|)$.

4.3 Αλγόριθμος Μηχανής Πρόβλεψης

Όπως έχει αναφερθεί και στην σχετική εργασία αυτής της Διπλωματικής Εργασίας, για την υλοποίηση οποιουδήποτε Συστήματος Προφόρτωσης στο Διαδίκτυο είναι απαραίτητη και η υλοποίηση μιας Μηχανής Πρόβλεψης (*Prediction Engine*). Αυτή η μηχανή συνήθως υλοποιείται στον εξυπηρετητή και είναι μέρος του Συστήματος Προφόρτωσης (*Prefetching System*). Στόχος της μηχανής αυτής είναι η πρόβλεψη των επόμενων προσβάσεων του χρήστη [5]. Σημειώνεται επίσης ότι η μηχανή προβλέψεων μπορεί να τοποθετηθεί σε οποιοδήποτε μέρος της Αρχιτεκτονικής Διαδικτύου (*Web Architecture*). Η τρέχουσα υλοποίηση υποθέτει την ύπαρξη της μηχανής πρόβλεψης στην πλευρά του εξυπηρετητή.

4.3.1 Επεξήγηση Αλγορίθμου

Ο χρήστης κινητής συσκευής επιθυμώντας να διαπιστώσει την τοποθεσία του, σαρώνει παθητικά τα Σημεία Πρόσβασης Ασύρματου Δικτύου που βρίσκονται γύρω του. Ακολούθως στέλλει αυτά τα Σημεία Πρόσβασης στον εξυπηρετητή, όπου η Μηχανή Πρόβλεψης αναλαμβάνει να δώσει την τρέχουσα τοποθεσία στον χρήστη μαζί με κάποιες προβλέψεις. Οι προβλέψεις αλλά και η τρέχουσα τοποθεσία που δίνεται πίσω στον χρήστη είναι συστοι-

χίες τιμών ισχύος¹, ή αλλιώς μερικοί ραδιοχάρτες. Εκτός από τους μερικούς ραδιοχάρτες, αποστέλλονται στον πελάτη και συμπιεσμένες συστοιχίες οι οποίες αποτελούνται από την ίδια πλειάδα πληροφοριών που χρησιμοποιεί ο αλγόριθμος ομαδοποίησης *BFR*. Η πλειάδα αυτή περιέχει τον αριθμό των σημείων (N) της συστοιχίας, το άθροισμα των σημείων σε κάθε διάσταση (διάνυσμα SUM) και το άθροισμα των τετραγώνων των σημείων σε κάθε διάσταση (διάνυσμα $SUMSQ$). Με αυτό τον τρόπο, και με τον υπολογισμό της απόστασης *Mahalanobis* η μηχανή Προφόρτωσης είναι σε θέση να επιβεβαιώνει τις προβλέψεις που λαμβάνει, αλλά και να ζητά άλλες συστοιχίες σε περίπτωση που οι προβλέψεις ήταν λανθασμένες.

Αλγόριθμος 3 Αλγόριθμος Μηχανής Πρόβλεψης

Input: Τα Σημεία Πρόσβασης Ασύρματου Δικτύου AP που ακούει ο χρήστης

Τις Τιμές Ισχύος RSS για τα αντίστοιχα Σημεία Πρόσβασης του συνόλου AP

Το Παράθυρο Πρόβλεψης W , Τον Αλγόριθμο Προφόρτωσης AL , Το γράφημα εξαρτήσεων G , Οι συστοιχίες που έχουν ήδη προσπελαστεί από τον χρήστη VIS , Αριθμός των προβλέψεων που θα επιστραφούν στον πελάτη n

Output:

PR ▷ Το σύνολο των προβλέψεων PR

CM ▷ Το Συμπιεσμένο Σύνολο CM

1: **procedure** PREDENGINE(AP, RSS, W, AL, G, VI)

Φάση Αρχικοποίησης

2: $RBT=initialize()$ ▷ Αρχικοποίηση ενός Red Black Tree

3: $(lat,lon)=CalculateLocation(AP,RSS)$ ▷ Υπολογισμός Τοποθεσίας

4: $n=ClosestCentroid(lat,lon)$ ▷ Εύρεση της πλησιέστερης συστοιχίας/κόμβου

Φάση Αρχικοποίησης

Στην φάση αρχικοποίησης του Αλγορίθμου αρχικοποιούνται οι απαραίτητες Δομές Δεδομένων και μεταβλητές. Αρχικά γίνεται η αρχικοποίηση του *Red Black Tree* το οποίο χρησιμοποιείται για την αποδοτική εύρεση των μεγαλύτερων πιθανοτήτων μετάβασης του χρήστη (σε λογαριθμικό χρόνο) . Στην γραμμή 3 υπολογίζεται η τοποθεσία του χρήστη χρησιμοποιώντας του υφιστάμενους αλγόριθμους γεωτοποθέτησης (π.χ. *KNN*, *WKNN*,

¹Όπως αναφέραμε και στο Μοντέλο Συστήματος ο Αλγόριθμος Προφόρτωσης δουλεύει σε επίπεδο συστοιχίας

MMSE, *WMMSE*). Σημειώνεται ότι οι προαναφερθείς αλγόριθμοι υπολογίζουν κατά προσέγγιση την τοποθεσία του χρήστη. Οπότε είναι απαραίτητο να βρεθεί το πλησιέστερο κέντρο συστοιχίας του χρήστη (βλέπε γραμμή 4). Το πλησιέστερο κέντρο αυτό υπολογίζεται χρησιμοποιώντας την *Ευκλείδεια Απόσταση*.

Αλγόριθμος 3 Αλγόριθμος Μηχανής Πρόβλεψης (Συνέχεια)

Φάση Πρόβλεψης

```

5:   for  $e_{n,i}$  in  $E$  do                                ▷ Για κάθε εξερχόμενη ακμή του κόμβου  $n$ 
6:       if  $i \notin VI$  then                                ▷ Εάν ο κόμβος προορισμού δεν έχει προσπελαστεί ήδη
7:           if  $AL = 'CBA'$  then                            ▷ Εάν είναι ο αλγόριθμος CBA
8:                $RBT = RBT \cup w(e_{n,i})$                 ▷ Πρόσθεση του βάρους της ακμής στο Red
               Black Tree
               ▷ Πολυπλοκότητα πρόσθεσης:  $\log(n)$ 
9:           else                                          ▷ Εάν είναι ο αλγόριθμος BBA
10:               $RBT = RBT \cup I[i]$                     ▷ Πρόσθεση της σημαντικότητας του
               ▷ κόμβου προορισμού  $i$ 
               ▷ στο Red Black Tree. Πολυπλοκότητα πρόσθεσης:  $\log(n)$ 
11:           end if
12:       end if
13:   end for

```

Φάση Πρόβλεψης

Σε αυτό το σημείο διατρέχονται όλες οι εξερχόμενες ακμές για την τρέχουσα συστοιχία του χρήστη n (βλέπε γραμμή 5). Ο αλγόριθμος της Μηχανής Πρόβλεψης διαφοροποιείται ανάλογα με τον Αλγόριθμο Προφόρτωσης που έχει επιλέξει ο χρήστης u . Εάν είναι ο αλγόριθμος CBA, βάζουμε τα βάρη των εξερχόμενων ακμών στο δέντρο (Red Black Tree). Υπενθυμίζεται ότι αυτά τα βάρη είναι οι πιθανότητες μετάβασης του χρήστη από τον τρέχον κόμβο του σε κάποιο γειτονικό κόμβο. Από την άλλη πλευρά, εάν είναι ο αλγόριθμος BBA, στο δέντρο βάζουμε την σημαντικότητα των κόμβων V_n, i . Το σύνολο V_n, i απαρτίζεται από τους κόμβους στους οποίους οδηγούν οι εξερχόμενες ακμές του κόμβου n . Παρατηρούμε λοιπόν, ότι ο αλγόριθμος BBA επιλέγει τους πιο σημαντικούς κόμβους οι οποίοι είναι συνδεδεμένοι μέσω κάποιας ακμής με τον κόμβο n .

Αλγόριθμος 3 Αλγόριθμος Μηχανής Πρόβλεψης (Συνέχεια)

```
14:    $PR = \{\}$       ▷ Αρχικοποίηση του συνόλου προβλέψεων που θα επιστραφεί στον  
      πελάτη  $PR$   
15:   for  $w \leftarrow 1, W$  do      ▷ Για κάθε μέγεθος παραθύρου  $1 \rightarrow W$   
16:        $PR = PR \cup RBT.max(n, w)$       ▷  $RBT.max(n, w)$ :επιστροφή των  $n$   
      προβλέψεων με την μεγαλύτερη  
      ▷ πιθανότητα για το παράθυρο  $w$   
17:   end for  
18:   for each neighbor of  $n$  do      ▷ Για κάθε γειτονική συστοιχία της τρέχουσας  
      συστοιχίας  
19:        $CM[neighbor] = CM[neighbor] \cup SUM[neighbor]$   
20:        $CM[neighbor] = CM[neighbor] \cup [neighbor]$   
21:        $CM[neighbor] = CM[neighbor] \cup SUMSQ[neighbor]$   
22:   end for  
23:   return  $PR, CM$       ▷ Επέστρεψε το σύνολο των προβλέψεων  
      ▷ και το συμπιεσμένο σύνολο  
24: end procedure
```

Τέλος, στην γραμμή 15 φτιάχνουμε το σύνολο προβλέψεων. Για κάθε μέγεθος παραθύρου $w \in (1 \rightarrow W)$ προσθέτουμε στο σύνολο PR τις πιο πιθανές προβλέψεις για το παράθυρο w . Στην γραμμή 18 του αλγορίθμου Πρόβλεψης, προστίθεται στο συμπιεσμένο σύνολο η τυπική απόκλιση της κάθε γειτονικής συστοιχίας από τον τρέχον κόμβο του χρήστη n . Το συμπιεσμένο σύνολο είναι σημαντικό και πρέπει να επιστραφεί στον πελάτη διότι με αυτή την πληροφορία μπορούμε να υπολογίσουμε την απόσταση *Mahalanobis* προς κάθε γείτονα. Αν οι προβλέψεις που έχει κάνει η Μηχανή Πρόβλεψης είναι λανθασμένες, και ο χρήστης προσεγγίζει συστοιχία για την οποία δεν έχει μερικό ραδιοχάρτη, μπορεί να ζητήσει από τον εξυπηρετητή την σωστή συστοιχία. Η μηχανή Προφόρτωσης θα ζητήσει την συστοιχία όταν η απόσταση *Mahalanobis* πέσει κάτω από ένα κατώφλι.

Ορισμός 5 *Ασυμπίεστη Συστοιχία*

Είναι μερικός ραδιοχάρτης τον οποίο επιστρέφει η μηχανή πρόβλεψης είτε ως υπολογισμό τοποθεσίας είτε ως πρόβλεψη. Με την Ασυμπίεστη Συστοιχία ο χρήστης μπορεί να εκτελέσει γεωτοποθέτηση εντός της συστοιχίας αυτής.

Ορισμός 6 Συμπιεσμένη Συστοιχία Είναι μια πλειάδα πληροφοριών οι οποίες χρησιμοποιούνται για τον υπολογισμό της απόστασης Mahalanobis από την συμπιεσμένη συστοιχία. Η πλειάδα αυτή αποτελείται από τον συνολικό αριθμό των σημείων μέσα στην συστοιχία N , το άθροισμα του κάθε σημείου σε κάθε διάσταση SUM και το τετράγωνο του αθροίσματος του κάθε σημείου σε κάθε διάσταση $SUMSQ$. Με την Συμπιεσμένη Συστοιχία ο χρήστης **δεν μπορεί** να εκτελέσει γεωτοποθέτηση. Αντίθετα, οι Συμπιεσμένες Συστοιχίες είναι περιοχές στις οποίες πιθανόν να πλοηγηθεί ο χρήστης αλλά δεν έχει τον αντίστοιχο μερικό ραδιοχάρτη. Όταν ο πελάτης προσεγγίσει ικανοποιητικά μια Συμπιεσμένη Συστοιχία φορτώνεται ο αντίστοιχος μερικός ραδιοχάρτης για διόρθωση της λανθασμένης πρόβλεψης.

4.3.2 Αλγόριθμος Μηχανής Πρόβλεψης-Ανάλυση Πολυπλοκότητας

Όπως παρατηρούμε στην γραμμή 5 του Αλγορίθμου 3, για κάθε εξερχόμενη ακμή του τρέχον κόμβου του χρήστη (n), βάζουμε στο δέντρο όλα τα βάρη ακμών (ή την σημαντικότητα των εξερχόμενων κόμβων). Επίσης, είναι γνωστό ότι η χρονική πολυπλοκότητα της εισαγωγής σε ένα *Red Black Tree* είναι λογαριθμική². Αν θέσουμε ως $|E_{n,i}|$ το σύνολο των εξερχόμενων ακμών του κόμβου n , τότε η χρονική πολυπλοκότητα του αλγορίθμου 3 είναι $\mathcal{O}(|E_{n,i}| \cdot \log(|E_{n,i}|))$.

²<http://goo.gl/fH51RQ>

4.4 Αλγόριθμος Μηχανής Προφόρτωσης

Αλγόριθμος 4 Αλγόριθμος Μηχανής Προφόρτωσης

Input: Ένα σύνολο προβλέψεων PR , Το συμπιεσμένο σύνολο CM

1: **procedure** PREFENGINE(PR, CM)

Φάση Αρχικοποίησης

- | | | |
|----|---------------------|---|
| 2: | $VI=\{\}$ | ▷ Σύνολο Συστοιχιών που έχουν προσπελαστεί ήδη |
| 3: | $AP=\{\}$ | ▷ Το Σύνολο των Ασύρματων Σημείων Πρόσβασης που ακούει ο χρήστης |
| 4: | $RSS=\{\}$ | ▷ Οι τιμές ισχύος των Ασύρματων Σημείων Πρόσβασης
▷ που ακούει ο χρήστης |
| 5: | $MD_THRESHOLD=1.5$ | ▷ Αρχικοποίηση του κατωφλίου |
| 6: | $RM=\{\}$ | ▷ Μερικός Ραδιοχάρτης Κινητής Συσκευής |
| 7: | $OPEN=\{\}$ | ▷ Συστοιχίες που δεν έχουν επιβεβαιωθεί ως σωστές |
| 8: | $CLOSE=\{\}$ | ▷ Συστοιχίες που έχουν επιβεβαιωθεί ως σωστές |
-

Σε Συστήματα Προφόρτωσης, εκτός από την Μηχανή Προβλέψεων υπάρχει επίσης και η *Μηχανή Προφόρτωσης (Prefetching Engine)*. Η Μηχανή Προφόρτωσης είναι αυτή η οποία επεξεργάζεται τις συστοιχίες που έχουν προβλεφτεί [5]. Αναφέρεται επίσης, ότι η Μηχανή Προφόρτωσης μπορεί να βρίσκεται είτε στον πελάτη είτε στον εξυπηρετητή. Στην υλοποίηση αυτής της Διπλωματικής Εργασίας η μηχανή βρίσκεται στον πελάτη. Κατά την κίνηση του ο Χρήστης Κινητής Συσκευής u , λαμβάνει συστοιχίες τις οποίες ίσως να χρειαστεί. Ο Αλγόριθμος της Μηχανής Προφόρτωσης εξετάζει κατά πόσο οι προβλέψεις αυτές είναι ορθές ή λανθασμένες. Πιο συγκεκριμένα διατηρούνται δύο σύνολα συστοιχιών, το $OPEN$ και το $CLOSE$. Το σύνολο $OPEN$ περιέχει όλες τις προβλεπόμενες συστοιχίες που δεν έχουν επιβεβαιωθεί ως σωστές ακόμα. Το σύνολο $CLOSE$ περιέχει όλες τις προβλεπόμενες συστοιχίες που έχουν επιβεβαιωθεί ως σωστές. Ακολουθώς υπολογίζεται η απόσταση *Mahalanobis* από την τρέχον συστοιχία του χρήστη προς την κάθε πρόβλεψη. Εάν η απόσταση αυτή, πέσει κάτω από κάποιο κατώφλι, τότε προστίθεται στο σύνολο $CLOSE$.

Αλγόριθμος 4 Αλγόριθμος Μηχανής Προφόρτωσης (Συνέχεια)

```
9:   for  $cid$  in  $PR$  do                                ▷ Για κάθε ταυτότητα προβλεπόμενης συστοιχίας:  $cid$ 
10:       $VI = VI \cup cid$                                 ▷ Πρόσθεσε στο σύνολο των κόμβων
                                                    ▷ που έχουν προσπελαστεί
11:       $RM = RM \cup PR[cid].radiomap$                 ▷ Πρόσθεσε τον μερικό ραδιοχάρτη
                                                    ▷ της πρόβλεψης στους μερικούς Ραδιοχάρτες της Κινητής Συσκευής
12:      if  $cid \notin OPEN$  and  $cid \notin CLOSE$  then
13:          $OPEN = OPEN \cup cid$                     ▷ Πρόσθεσε στο σύνολο  $OPEN$  τις
                                                    ▷ προβλέψεις-κόμβους που δεν έχουν επιβεβαιωθεί
14:      end if
15:   end for
```

Η μηχανή Προφόρτωσης λαμβάνοντας το σύνολο των Προβλέψεων PR και το Συμπίε-
σμένο σύνολο CM , προσθέτει στο σύνολο των μερικών Ραδιοχαρτών του, τους μερικούς ρα-
διοχάρτες για τις συστοιχίες που έχουν προβλεφθεί (βλέπε γραμμή 11). Στην γραμμή 12 του
αλγορίθμου, προστίθενται στο σύνολο $OPEN$ οι συστοιχίες που δεν έχουν προσπελαστεί
από τον χρήστη, και άρα δεν έχουν επιβεβαιωθεί ως σωστές.

Στην συνέχεια καθώς ο χρήστης κινείται, για κάθε νέα τοποθεσία του χρήστη, υπολογί-
ζεται η απόσταση *Mahalanobis* από την νέα του τοποθεσία προς κάθε γειτονική συστοιχία
(είτε αυτή είναι συμπίεσμένη συστοιχία, είτε συστοιχία πρόβλεψης). Σε περίπτωση που η
απόσταση *Mahalanobis* με κάποια συστοιχία K πέσει κάτω από ένα κατώφλι η ενέργεια
που λαμβάνεται από την Μηχανή Προφόρτωσης είναι διαφορετική, ανάλογα με το αν η
συστοιχία K είναι συστοιχία πρόβλεψης ή συμπίεσμένη συστοιχία. Εάν είναι πρόβλεψη
τότε αφαιρείται από το σύνολο $OPEN$ και προστίθεται στο σύνολο $CLOSE$ και $RIGHT$.

Το σύνολο $RIGHT$ χρησιμοποιείται μετά για την προφόρτωση των επόμενων πιθανών
συστοιχιών (βλέπε γραμμή 34). Εάν η συστοιχία K είναι συμπίεσμένη, ο αλγόριθμος
ζητά από τον εξυπηρετητή την ασυμπίεστη συστοιχία K και τις επόμενες πιο πιθανές της
συστοιχίες.

Αλγόριθμος 4 Αλγόριθμος Μηχανής Προφύρτωσης (Συνέχεια)

```
16:   while user is moving do
17:      $RIGHT = \{\}$ 
18:      $(lat, lon) = calculateLocation(RM, AP, RSS)$ 
19:     for each neighbor of the user's current cluster do
20:        $MD[neighbor] = calculateMahalanobis(lat, lon, neighbor)$ 
                                     ▷ Υπολόγισε την απόσταση Mahalanobis
                                     ▷ από την τρέχον τοποθεσία προς όλους τους γείτονες
21:     end for
22:     for pid in  $CM$  do                                     ▷ Για κάθε γειτονική συστοιχία του χρήστη
23:       if  $MD[pid] \leq MD\_THRESHOLD$  and  $pid \in PR$  then
                                     ▷ Εάν η γειτονική συστοιχία είναι πρόβλεψη και
                                     ▷ η απόσταση της πέσει κάτω από κάποιο κατώφλι
24:          $OPEN = OPEN \setminus pid$                                      ▷ Αφαίρεσε από το ανοιχτό σύνολο
25:          $CLOSE = CLOSE \cup pid$                                      ▷ Πρόσθεσε στο κλειστό σύνολο
26:          $RIGHT = RIGHT \cup pid$                                      ▷ Πρόσθεσε την πρόβλεψη
                                     ▷ στις σωστές προβλέψεις
```

Αλγόριθμος 4 Αλγόριθμος Μηχανής Προφόρτωσης (Συνέχεια)

```
27:         else
28:             if  $MD[pid] \leq MD\_THRESHOLD$  and  $pid \notin PR$  then
                ▷ Εάν η γειτονική συστοιχία δεν είναι πρόβλεψη
                ▷ και η απόσταση της πέσει κάτω από κάποιο κατώφλι
29:                  $CLOSE = CLOSE \cup pid$           ▷ Πρόσθεσε στο κλειστό σύνολο
30:                  $fetchCluster(pid)$               ▷ Λάθος Πρόβλεψη: Φέρε από
                ▷ τον εξυπηρετητή τον γείτονα pid
31:             end if
32:         end if
33:     end for
34:     for right in  $RIGHT$  do                      ▷ Για κάθε ορθή πρόβλεψη
35:          $FecthLikelyNext(right)$                 ▷ Προφόρτωσε την πιο πιθανή
                ▷ επόμενη γειτονική συστοιχία
36:     end for
37: end while
38: end procedure
```

4.5 Παράδειγμα Εκτέλεσης Αλγόριθμου Πρόβλεψης και Προφόρτωσης

Στο παράδειγμα που θα παρουσιαστεί χρησιμοποιήθηκε για τις προβλέψεις ο αλγόριθμος CBA. Ο αλγόριθμος BBA τυγχάνει στο περιβάλλον αξιολόγησης των αλγορίθμων να προτείνει τις ίδιες συστοιχίες. Οπότε παραλείπουμε το παράδειγμα εκτέλεσης για τον αλγόριθμο BBA.

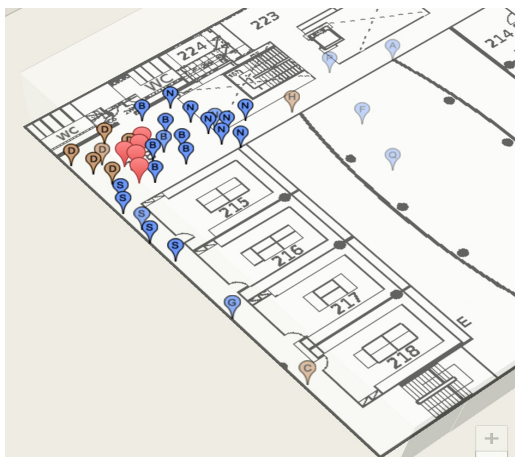


Σχήμα 4.4: Πριν την γεωτοποθέτηση



Σχήμα 4.5: 3η Γεωτοποθέτηση

Στο Σχήμα 4.4 ο πελάτης έστειλε στον εξυπηρετητή ένα σύνολο από σημεία πρόσβασης με την ισχύ που τα άκουσε. Ο εξυπηρετητής αντιλήφθηκε ότι, ο πελάτης βρίσκεται στην συστοιχία D , οπότε απαντάει στον πελάτη με τον μερικό ραδιοχάρτη της συστοιχίας D και της επόμενης πιο πιθανής συστοιχίας B . Στις επόμενες τρεις γεωτοποθετήσεις του χρήστη, ο χρήστης προσεγγίζει μια συστοιχία για την οποία δεν έχει μερικό ραδιοχάρτη. Αντίθετα έχει μια συμπιεσμένη συστοιχία, την S . Στο Σχήμα 4.5 λοιπόν, προσεγγίζει (δηλαδή η απόσταση *Mahalanobis* πέφτει κάτω από ένα κατώφλι) την συμπιεσμένη συστοιχία S , ο αλγόριθμος Προφόρτωσης αντιλαμβάνεται το λάθος του και ζητά από τον εξυπηρετητή την μη συμπιεσμένη συστοιχία S , ούτως ώστε ο πελάτης να εκτελέσει γεωτοποθέτηση.



Σχήμα 4.6: 5η Γεωτοποθέτηση

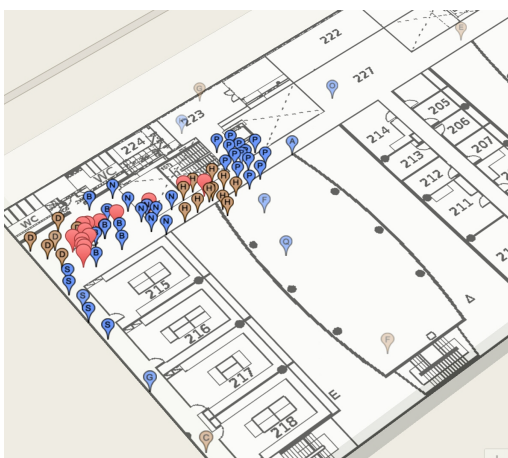


Σχήμα 4.7: 12η Γεωτοποθέτηση

Στο Σχήμα 4.6 ο πελάτης ξανά-προσεγγίζει ικανοποιητικά την συστοιχία *B*, οπότε η πρόβλεψη για αυτή την συστοιχία επιβεβαιώνεται και προφορτώνεται η επόμενη πιο πιθανή *N*. Στο Σχήμα 4.7 και μετά από 12 γεωτοποθετήσεις ο πελάτης προσεγγίζει ικανοποιητικά την συστοιχία *N*, η πρόβλεψη επιβεβαιώνεται ξανά και προφορτώνεται η επόμενη πιο πιθανή συστοιχία *H*.



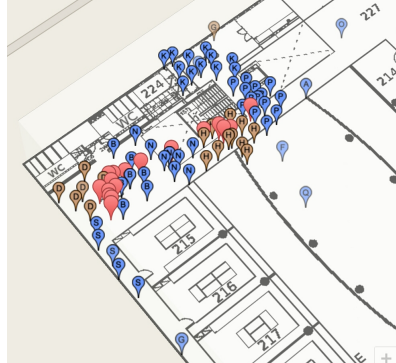
Σχήμα 4.8: 13η Γεωτοποθέτηση



Σχήμα 4.9: 14η Γεωτοποθέτηση

Το ίδιο μοτίβο κίνησης και επιβεβαίωσης επαναλαμβάνεται μέχρι την 17η γεωτοποθέτηση. Είναι σημαντικό να προσέξει κάποιος ότι ανά πάσα χρονική στιγμή ο πελάτης έχει στην διάθεση του και ασυμπίεστες και συμπιεσμένες συστοιχίες. Οι ασυμπίεστες συστοιχίες είναι μερικοί ραδιοχάρτες τους οποίους χρησιμοποιεί ο πελάτης για να πλοηγηθεί. Αντίθετα οι ασυμπίεστες συστοιχίες έχουν ως σκοπό να διορθώσουν μια λανθασμένη πρόβλεψη που έχει κάνει η μηχανή Προβλέψεων. Στο Σχήμα 4.10, αν ο χρήστης τελικά

αποφασίζει να πλοηγηθεί προς την συστοιχία A για την οποία δεν έχει αποτυπώματα τιμών ισχύος, η μηχανή πρόφόρτωσης θα διορθώσει την κατάσταση, χρησιμοποιώντας την ασυμπίεστη συστοιχία A .



Σχήμα 4.10: 17η Γεωτοποθέτηση

Κεφάλαιο 5

Πρότυπο Συστήματος

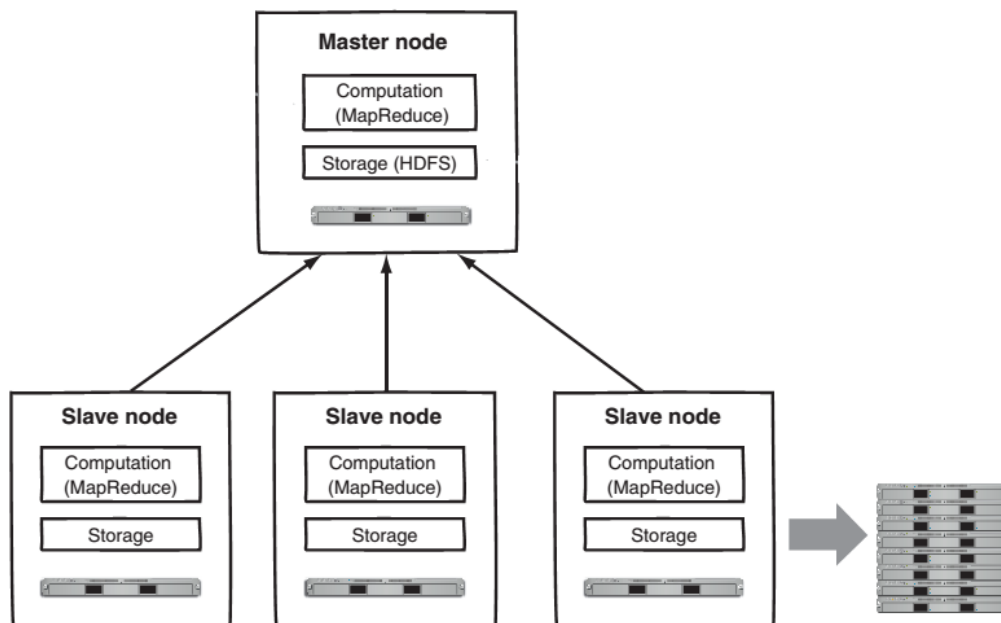
5.1	Τεχνολογικό Υπόβαθρο Εξυπηρετητή	51
5.1.1	Κατανεμημένο Πλαίσιο Hadoop	51
5.1.2	Κατανεμημένη Μη-Σχεσιακή Βάση Δεδομένων HBase	53
5.1.3	Εξυπηρετητής Ιστού Play Framework	55
5.2	Τεχνολογικό Υπόβαθρο Πελάτη	56
5.2.1	Ανάπτυξη λογισμικού στην πλατφόρμα Android	56
5.2.2	Διεπαφή Προγραμματισμού Εφαρμογών χαρτών Google	57
5.3	Επισκόπηση Υλοποίησης	57
5.4	Υλοποίηση Εξυπηρετητή	57
5.4.1	Μονάδα Ιστού Play Framework	58
5.4.2	Γραφική Διεπαφή Χρήστη	62

Σε αυτό το τμήμα περιγράφεται ένα πρότυπο συστήματος το οποίο έχει αναπτυχθεί στα πλαίσια αυτής της Διπλωματικής Εργασίας. Η ανάπτυξη του συστήματος έγινε χρησιμοποιώντας το Λειτουργικό Σύστημα *Android*. Επίσης, παρέχουμε μια επισκόπηση των λειτουργιών του Γραφικού Περιβάλλοντος Διεπαφής (*Graphical User Interface*). Περαιτέρω, περιγράφεται η αρχιτεκτονική του εξυπηρετητή.

5.1 Τεχνολογικό Υπόβαθρο Εξυπηρετητή

5.1.1 Κατανεμημένο Πλαίσιο Hadoop

Ο εξυπηρετητής τρέχει πάνω στο *Hadoop*. Το *Hadoop* είναι ένα Πλαίσιο Ανοικτού Κώδικα (*Open Source Framework*) το οποίο παρέχει κατανεμημένο αποθηκευτικό χώρο καθώς και άλλα πλαίσια για τον προγραμματισμό πάνω σε Μεγάλα Δεδομένα (*Big Data*). Σημειώνεται ότι το ευρύτερο πλαίσιο το οποίο ονομάζεται *Hadoop* αποτελείται από δύο επιμέρους πλαίσια: το κατανεμημένο Σύστημα Αρχείων (*Hadoop Distributed File System-HDFS*) και το προγραμματιστικό πλαίσιο *Map-Reduce* το οποίο είναι ένα ευέλικτο και παράλληλο πλαίσιο για τον προγραμματισμό Μεγάλων Δεδομένων. Μια αφαιρετική αναπαράσταση του πλαισίου *Hadoop* φαίνεται στο Σχήμα 5.1. Για την αύξηση της υπολογιστικής δύναμης αλλά και του αποθηκευτικού χώρου είναι απαραίτητη η πρόσθεση περισσότερων κόμβων (*Slave Nodes*).



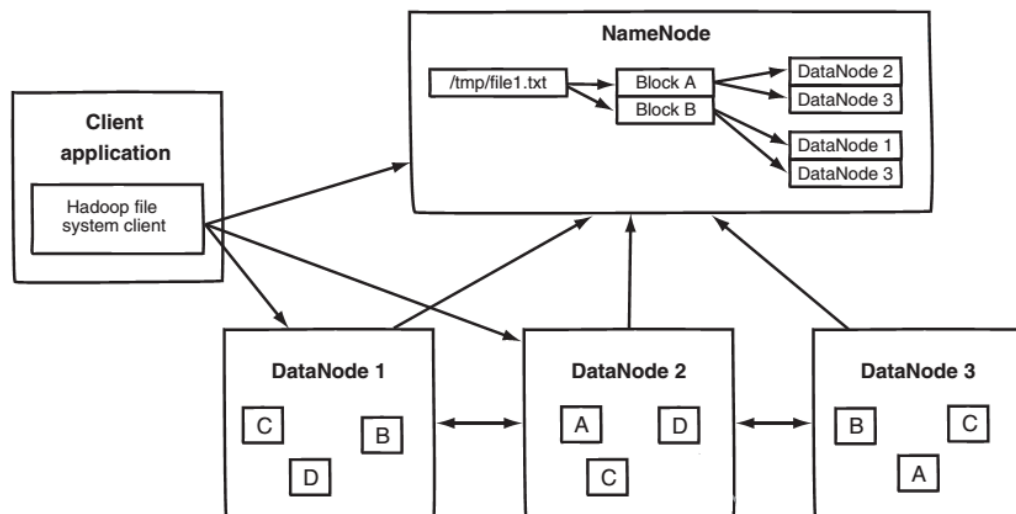
Σχήμα 5.1: Αφαιρετική Αρχιτεκτονική του Hadoop [11]

Κατανεμημένο Σύστημα Αρχείων *HDFS*

Είναι η ανοικτού κώδικα υλοποίηση του Συστήματος Αρχείων της Google (*Google File System*) [9]. Το κατανεμημένο Σύστημα Αρχείων *HDFS* παρέχει ψηλή Επεκτασιμότητα (*Scalability*) και Διαθεσιμότητα (*Availability*). Το Σύστημα Αρχείων

HDFS αντιγράφει, για ένα καθορισμένο αριθμό αντιγράφων (*Replication Factor*), τα δεδομένα σε πολλούς διαφορετικούς κόμβους. Με αυτό τον τρόπο αυξάνεται η ανοχή του συστήματος σε σφάλματα (*Fault Tolerance*) και η διαθεσιμότητα του Συστήματος Αρχείων (προφανώς, ένας πελάτης μπορεί να εξυπηρετηθεί από ένα αριθμό διαφορετικών κόμβων). Τέλος, αναφέρεται ότι η αντιγραφή των αρχείων γίνεται στο επίπεδο του μπλοκ στο *HDFS*. Το Σχήμα 5.2 απεικονίζει την αντιγραφή που γίνεται στο *HDFS* αλλά και τον τρόπο εξυπηρέτησης ενός πελάτη. Παρατηρώντας το Σχήμα 5.2 βλέπουμε δύο νέα στοιχεία, τον κόμβο *NameNode* και τον κόμβο *DataNode* (*Κόμβος Δεδομένων*). Ο κόμβος *NameNode*, ονομάζεται και *Master Node* και είναι αυτός ο οποίος έχει στην κατοχή του τις μετά-πληροφορίες του κάθε αρχείου του συστήματος. Αυτός ο κόμβος λοιπόν γνωρίζει που βρίσκονται ανά πάσα χρονική στιγμή το κάθε μπλοκ του κάθε αρχείου. Από την άλλη πλευρά, οι *DataNodes* κόμβοι του συστήματος αποθηκεύουν την πληροφορία του συστήματος αρχείων. Πιο ειδικά, ένας Κόμβος Δεδομένων έχει στην διάθεση του ένα αριθμό από μπλοκ τα οποία προέρχονται από διάφορα αρχεία. Από το παράδειγμα του Σχήματος 5.2 μπορούμε να συμπεράνουμε δύο πράγματα:

- Η αντιγραφή που συμβαίνει είναι δεύτερου βαθμού (*Replication Factor*).
- Το αρχείο */tmp/file1.txt* αποτελείται από δύο μπλοκ τα οποία είναι σκορπισμένα σε τρεις Κόμβους Δεδομένων



Σχήμα 5.2: Το Κατανεμημένο Σύστημα αρχείων *HDFS* [11]

Κατανεμημένο Προγραμματιστικό Μοντέλο *MapReduce*

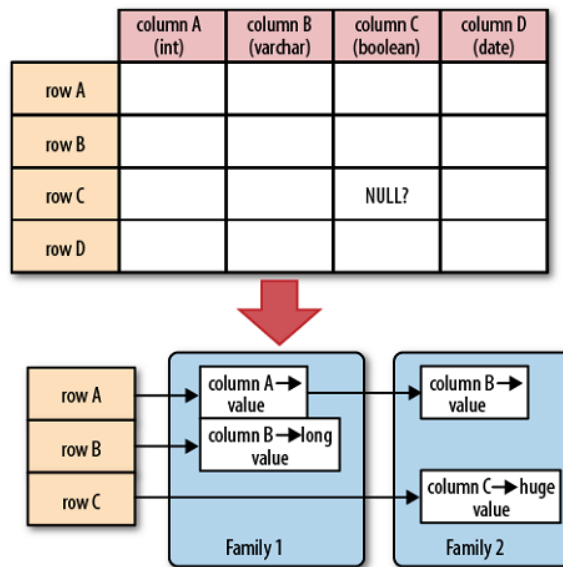
Είναι η ανοικτού κώδικα υλοποίηση του πλαισίου *MapReduce* που χρησιμοποιεί η *Google* για την ανάλυση των δεδομένων και παροχή υπηρεσιών της [4]. Το μοντέλο αυτό, απλοποιεί τον προγραμματισμό των Κατανεμημένων Συστημάτων. Στα Κατανεμημένα Συστήματα, που δεν κάνουν χρήση του πιο πάνω πλαισίου τα σφάλματα, ο παράλληλος υπολογισμός και η κατανομή της δουλειάς είναι ευθύνη του προγραμματιστή. Αντίθετα, σε ένα Κατανεμημένο Σύστημα που χρησιμοποιεί το πιο πάνω μοντέλο υπολογισμού, τα προβλήματα που μόλις αναφέραμε τα αντιμετωπίζει το ίδιο το πλαίσιο. Ευθύνη του προγραμματιστή είναι ο ορισμός δύο συναρτήσεων, της *Map* και της *Reduce*. Μια πιο λεπτομερής περιγραφή και η ροή εκτέλεσης μιας *MapReduce* εργασίας εξηγούνται στο Παράρτημα Α.

5.1.2 Κατανεμημένη Μη-Σχεσιακή Βάση Δεδομένων *HBase*

Η Βάση Δεδομένων *HBase* είναι η ανοικτού κώδικα υλοποίηση του *BigTable* της *Google* [3]. Στην *HBase* δεν ακολουθείται το παραδοσιακό μοντέλο των Σχεσιακών Βάσεων Δεδομένων. Αντίθετα, είναι μια κατανεμημένη και επεκτάσιμη Βάση Δεδομένων η οποία τρέχει πάνω από το Σύστημα Αρχείων *HDFS*. Σε αντίθεση με τις Σχεσιακές Βάσεις Δεδομένων δεν υπάρχει η έννοια του πρωτεύοντος κλειδιού (*Primary Key*) το οποίο αναγνωρίζει μοναδικά μια εγγραφή. Για την μοναδική αναγνώριση μια εγγραφής γίνεται χρήση μιας πλειάδας πληροφοριών. Η πλειάδα αυτή περιέχει τα εξής πεδία $\langle rowkey, columnkey, timestamp \rangle$. Περαιτέρω, είναι σε θέση να φυλάξει Πλήρως Δομημένα Δεδομένα (*Structured Data*), Ήμι-Δομημένα Δεδομένα (*Semi Structured Data*) και Μη-Δομημένα Δεδομένα (*Unstructured Data*), οπότε είναι κατάλληλη για μια πληθώρα εφαρμογών. Η ιδιόκτητη αντίστοιχη Βάση Δεδομένων *BigTable* της *Google* χρησιμοποιείται για ένα αριθμό εφαρμογών (Για παράδειγμα *Google Earth*, *Google Maps* και *Google Analytics*) [3]. Σημειώνεται επίσης ότι κάθε πλεονέκτημα που παρέχει το *HDFS* ισχύει και για την *HBase* (π.χ. Ανοχή Σφαλμάτων, Επεκτασιμότητα και Διαθεσιμότητα).

Μοντέλο Δεδομένων Μη-Σχεσιακής Βάσης Δεδομένων *HBase*

Αρχικά, ένας πίνακας ορίζεται ως ένα σύνολο γραμμών. Σε αντίθεση με τις Σχεσιακές Βάσεις Δεδομένων, το σχήμα ενός πίνακα στην *HBase* δεν ορίζεται αυστηρά την ώρα

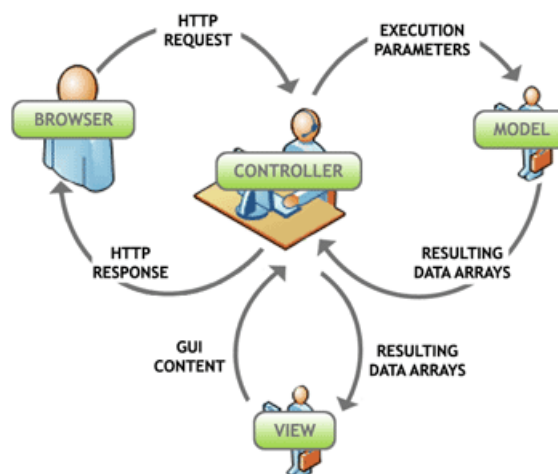


Σχήμα 5.3: Σύγκριση *HBase* και *RDBMS* [8]

της δημιουργίας του. Κατά την φάση δημιουργίας ενός πίνακα απαιτείται μόνο το όνομα του πίνακα καθώς και το όνομα μιας Οικογένειας Στηλών (*Column Family*). Συνεπώς, κάθε γραμμή ενός πίνακα μπορεί να έχει διαφορετικό αριθμό στηλών. Οπότε ένα σύνολο στηλών συνθέτει μια γραμμή, η οποία γραμμή αναγνωρίζεται μοναδικά από το κλειδί της γραμμής (*rowkey*). Η *HBase* παρέχει επίσης διάφορες εκδόσεις ανά κελί δεδομένων. Στην τελευταία λειτουργία οφείλεται και η ύπαρξη της Χρόνο-σφραγίδας στην πλειάδα πρόσβασης ενός κελιού. Η πλειάδα πρόσβασης $\langle rowkey, columnkey, timestamp \rangle$ που έχει αναφερθεί λίγο απλοϊκά στο προηγούμενο τμήμα ορίζεται ορθότερα αν ορίσουμε την Οικογένειας Στηλών (*Column Family*). Οι στήλες ομαδοποιούνται σε Οικογένειες Στηλών. Όλες οι στήλες μιας Οικογένειας Στηλών φυλάσσονται γειτονικά στον δίσκο σε αρχεία που ονομάζονται *HFiles*. Επομένως, η πλειάδα πρόσβασης ορίζεται καλύτερα ως: $\langle Table, RowKey, Family, Column, Timestamp \rangle$. Η εικόνα 5.3 παρέχει μια άριστη σύγκριση μεταξύ Σχεσιακών Βάσεων Δεδομένων και της *HBase*. Παρατηρώντας με μεγαλύτερη προσοχή την εικόνα 5.3 βλέπουμε ότι στην *HBase* δεν υπάρχει ανάγκη ορισμού τύπων όπως σε μια Σχεσιακή Βάση Δεδομένων. Επίσης βλέπουμε ότι κάθε γραμμή του πίνακα της *HBase* έχει τιμές για μερικές στήλες και όχι για όλες. Με αυτό τον τρόπο αποφεύγεται η χρήση ειδικών τιμών σε περίπτωση που κάποιο κελί είναι άδειο (*null*). Μια πιο λεπτομερής περιγραφής λειτουργία της κατανεμημένης Μη-Σχεσιακής Βάσης Δεδομένων *HBase* βρίσκεται στο Παράρτημα Β.

5.1.3 Εξυπηρετητής Ιστού Play Framework

Το πλαίσιο *Play* είναι ένα ανοιχτού κώδικα πλαίσιο Διαδικτύου, το οποίο έχει γραφτεί στις προγραμματιστικές γλώσσες *Java* και *Scala*. Το πλαίσιο αυτό ακολουθεί το πρότυπο αρχιτεκτονικής *MVC* (*Model View Controller*). Το ίδιο πρότυπο ακολουθούν και τα Διαδικτυακά πλαίσια προγραμματισμού *Django*¹ και *Ruby on Rails*². Επομένως, το λογισμικό του εξυπηρετητή διαιρείται σε τρία διασυνδεδεμένα κομμάτια: *Model*, *View* και *Controller*. Στο κομμάτι του μοντέλου (*Model*) περιέχονται τα δεδομένα της εφαρμογής, η λογική και οι λειτουργίες της εφαρμογής. Το κομμάτι της όψης (*View*) περιλαμβάνει οποιαδήποτε γραφική ή διαγραμματική αναπαράσταση. Τέλος, το κομμάτι του Ελεγκτή (*Controller*) δέχεται είσοδο από τον χρήστη την οποία ερμηνεύουν τα άλλα δύο κομμάτια. Ο τρόπος λειτουργίας του προτύπου παρουσιάζεται διαγραμματικά στην εικόνα 5.4.



Σχήμα 5.4: Ροή Λειτουργίας του προτύπου MVC

Κύρια Χαρακτηριστικά του Play Framework

Το *Play Framework* υποστηρίζει ένα πλήρως ασύγχρονο προγραμματιστικό μοντέλο. Αυτό είναι σημαντικό, αφού σε εφαρμογές μεγάλων δεδομένων, όπου υπάρχει σημαντικός αριθμός χρηστών για εξυπηρέτηση δεν υπάρχει χρόνος για συγχρονισμό. Η αρχιτεκτονική του πλαισίου έχει κατασκευαστεί εξαρχής με την υπόθεση ότι οι αιτήσεις είναι μακράς

¹<https://docs.djangoproject.com/en/dev/faq/general/>

²http://guides.rubyonrails.org/getting_started.html

διαρκείας. Επίσης, το πλαίσιο αυτό δεν κάνει χρήση οποιουδήποτε *Web Container* ³, πράγμα το οποίο κάνει το *Play Framework* πιο αποδοτικό σε σχέση με τις άλλες τεχνολογίες εξυπηρετητών.

Για μεγαλύτερο βαθμό ανταπόκρισης στους πελάτες το *Play Framework* δεν δημιουργεί ένα νήμα (*thread*) για κάθε αίτηση που έρχεται στον εξυπηρετητή. Αντίθετα, χρησιμοποιεί αρχιτεκτονική καθοδηγούμενη από συμβάντα (*Event Driver Architecture*), όπου ένα συμβάν δημιουργείται όταν μια αίτηση πελάτη ζητά κάτι από τον εξυπηρετητή. Καλό θα ήταν να αναφέρουμε ότι το πλαίσιο δεν κρατά κατάσταση για κάθε χρήστη. Επομένως μια ακολουθία από αιτήσεις ενός πελάτη δεν μπορεί να φυλάξει κάποια κατάσταση στον εξυπηρετητή.

Πλέον, υπάρχει μια μεγάλη ποικιλία από Αποθήκες Δεδομένων που κανένα πλαίσιο Διαδικτύου δεν είναι σε θέση να γνωρίζει ποια Αποθήκη Δεδομένων χρησιμοποιεί ένας προγραμματιστής. Παρόλα αυτά το πλαίσιο *Play* παρέχει κάποιες απλές βοηθητικές λειτουργίες όπως είναι η διαχείριση της σύνδεσης με μια βάση Δεδομένων. Με την βοήθεια των λειτουργιών αυτών η χρήση οποιασδήποτε Αποθήκης Δεδομένων είναι εύκολη υπόθεση στο Διαδικτυακό πλαίσιο *Play*.

Τα πιο πάνω χαρακτηριστικά κάνουν το *Play Framework* κατάλληλο για εξυπηρετητή που έχει στο προσκήνιο μια αποθήκη μεγάλων δεδομένων όπως είναι η *HBase*.

5.2 Τεχνολογικό Υπόβαθρο Πελάτη

Για την εφαρμογή του πελάτη χρησιμοποιήθηκαν: το Εργαλείο Ανάπτυξης Λογισμικού (SDK) του Λειτουργικού Συστήματος Android ⁴, η γλώσσα προγραμματισμού JAVA, η Διεπαφή Προγραμματισμού Εφαρμογών χαρτών Google έκδοση 2η (*APIv2*), και βιβλιοθήκες συμβατότητας με προηγούμενες εκδόσεις Android.

5.2.1 Ανάπτυξη λογισμικού στην πλατφόρμα Android

Το Λειτουργικό Σύστημα *Android* είναι μια ανοιχτού κώδικα πλατφόρμα για κινητές συσκευές κάθε τύπου, που χρησιμοποιεί πυρήνα Linux. Το Εργαλείο Ανάπτυξης Λογισμικού (SDK), που χρησιμοποιείται για την πλατφόρμα Android, είναι σε γλώσσα προγραμματισμού Java, και παρέχει την δυνατότητα στους προγραμματιστές να αλληλεπιδράσουν με

³Χρησιμοποιείται από άλλες τεχνολογίες Διαδικτύου π.χ. Java Servlets και είναι υπεύθυνο για την διαχείριση του κύκλου ζωής των Java Servlets

⁴Android Open Source Project (AOSP): <https://source.android.com/>

τον πυρήνα, ούτως ώστε να εξασφαλίσουν πρόσβαση στους πόρους που χρειάζονται, με απώτερο σκοπό την δημιουργία μιας ολοκληρωμένης εφαρμογής. Μερικοί από τους πόρους στις έξυπνες κινητές συσκευές είναι: η οθόνη αφής, η εσωτερική και εξωτερική μνήμη, τα ηχεία, ο δέκτης Ασύρματων Δικτύων, ο δέκτης Παγκόσμιου Συστήματος Συντεταγμένων, και μια σειρά από αισθητήρες όπως γυροσκόπιο, πυξίδα, βαρόμετρο.

5.2.2 Διεπαφή Προγραμματισμού Εφαρμογών χαρτών Google

Χρησιμοποιώντας την Διεπαφή Προγραμματισμού Εφαρμογών (API) χαρτών της Google, μπορούν να εισαχθούν χάρτες σε οποιαδήποτε εφαρμογή Android σε τρισδιάστατη μορφή βασισμένοι στα δεδομένα των χαρτών Google σε κάποια εφαρμογή στην πλατφόρμα Android. Η διεπαφή χειρίζεται αυτόματα την πρόσβαση στους εξυπηρετητές για την διάθεση των χαρτών, την εμφάνιση τους, αλλά και την αλληλεπίδραση που τυγχάνουν από τον χρήστη. Χρησιμοποιώντας την διεπαφή μπορούν να εισαχθούν στοιχεία στον χάρτη όπως σημεία, σχήματα, εικόνες, ή και επίπεδα. Περαιτέρω, μπορεί να εισαχθεί μια ποικιλία από αντικείμενα πάνω από τους χάρτες, όπως για παράδειγμα σημεία, σχήματα και εικόνες (π.χ. το αρχιτεκτονικό σχέδιο ενός ορόφου). Στην τρέχουσα υλοποίηση χρησιμοποιούνται τα προηγούμενα εργαλεία για την πρόσθεση αρχιτεκτονικού σχεδίου πάνω στους χάρτες και επίδειξη των γεωγραφικών σημείων ενός ορόφου πάνω στο σχέδιο αυτό.

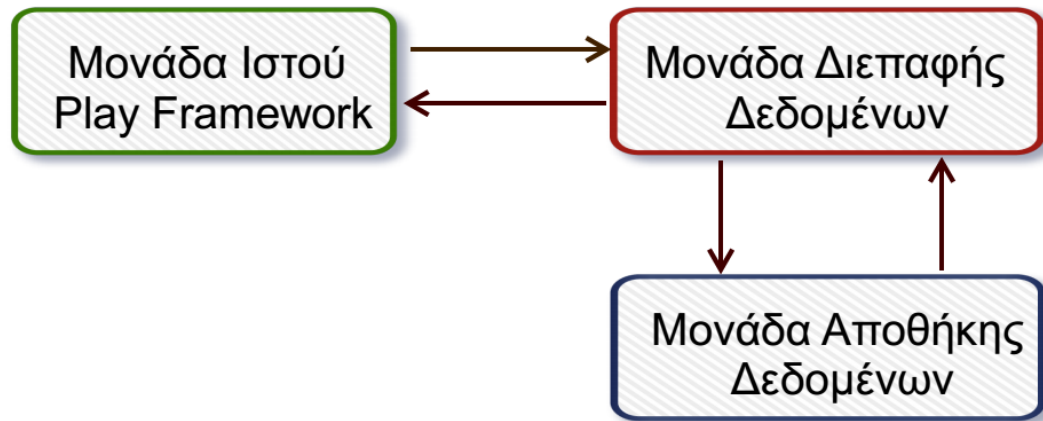
5.3 Επισκόπηση Υλοποίησης

Το λογισμικό του πελάτη έχει αναπτυχθεί χρησιμοποιώντας το Λειτουργικό Σύστημα *Android* και το πακέτο εγκατάστασης της εφαρμογής έχει μέγεθος 1,023 KB. Ο κώδικας του συστήματος είναι γραμμένος στην Γλώσσα Προγραμματισμού *JAVA* και αποτελείται από 8,456 γραμμές κώδικα. Πιο συγκεκριμένα, ο κώδικας του εξυπηρετητή χρησιμοποιεί ≈ 3967 και τρέχει πάνω σε *JDK 7* και στο Λειτουργικό Σύστημα *CentOS 6.4*, καθώς η εφαρμογή του πελάτη περιέχει ≈ 4138 γραμμές κώδικα συν ≈ 351 γραμμές *XML* για τις ρυθμίσεις και στοιχεία του Περιβάλλοντος Διεπαφής.

5.4 Υλοποίηση Εξυπηρετητή

Ο εξυπηρετητής αποτελείται από τρία στρώματα, η μονάδα Ιστού, η μονάδα διεπαφής με τα δεδομένα και η μονάδα Αποθήκης Δεδομένων. Η μονάδα Ιστού είναι το Διαδικτυακό

Πλαίσιο *Play Framework*. Η επιλογή αυτή ήταν πολύ καλή διότι το *Play Framework* υποστηρίζει την γλώσσα προγραμματισμού *JAVA*. Δεδομένου ότι οι διεπαφές προγραμματισμού της *HBase* και του *Hadoop* είναι επίσης σε γλώσσα προγραμματισμού *Java*⁵ το *Play Framework* αποτέλεσε μια άριστη επιλογή. Η αλληλεπίδραση μεταξύ αυτών των μονάδων φαίνεται στο Σχήμα 5.5.



Σχήμα 5.5: Αλληλεπίδραση Μονάδων Εξυπηρετητή

Η μονάδα διεπαφής με τα δεδομένα αποτελείται από τις Προγραμματιστικές Διεπαφές του *Hadoop* και της *HBase* οι οποίες είναι επίσης σε γλώσσα προγραμματισμού *JAVA*. Υπάρχουν διεπαφές προγραμματισμού σε επιπλέον γλώσσες όπως είναι η *Python*, αλλά η *JAVA* είναι μια πιο φυσική επιλογή αφού ολόκληρο το πλαίσιο *Hadoop* και η Αποθήκη Δεδομένων *HBase* είναι υλοποιημένες σε *JAVA*.

Η μονάδα Αποθήκης Δεδομένων αποτελείται από την Βάση Δεδομένων *HBase* η οποία είναι μια κατανεμημένη, επεκτάσιμη και με μεγάλη ανοχή στα σφάλματα αποθήκη δεδομένων. Η *HBase* κληρονομεί την ανοχή σφαλμάτων από το κατανεμημένο πλαίσιο *Hadoop* και το κατανεμημένο σύστημα *HDFS* το οποίο παρέχει έναν παράγοντα αντιγραφής της τάξης του τρία (συνήθως) στο επίπεδο του μπλοκ.

5.4.1 Μονάδα Ιστού Play Framework

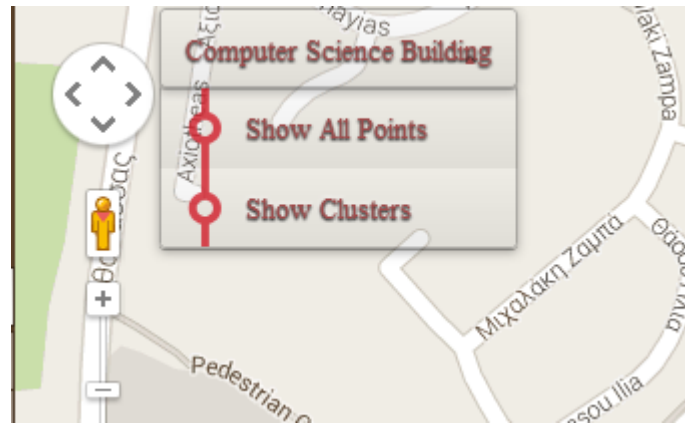
Ιστοσελίδα Επικύρωσης Ομαδοποίησης

Η μονάδα Ιστού *Play Framework* περιλαμβάνει την ανάπτυξη μιας απλής ιστοσελίδας⁶ η οποία χρησιμοποιήθηκε για επικύρωση της γεωγραφικής ομαδοποίησης αποτυπωμάτων.

⁵ Διεπαφές σε άλλες γλώσσες υπάρχουν αλλά δεν παρέχουν το ίδιο ποσοστό λειτουργικότητας

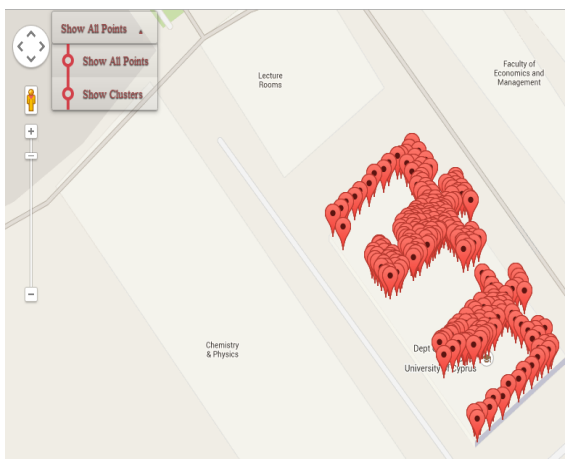
⁶ Η ιστοσελίδα μπορεί να βρεθεί στον σύνδεσμο <http://vectormap1.in.cs.ucy.ac.cy:8090/>

Η ιστοσελίδα αυτή ήταν χρήσιμη για στο στάδιο της προ-επεξεργασίας των ραδιοχαρτών και κατά τον σχηματισμό των συστοιχιών. Η ιστοσελίδα περιλαμβάνει ένα απλό μενού από το οποίο μπορεί κάποιος να δει τα αποτυπώματα τιμών ισχύος, αλλά και τις συστοιχίες τιμών ισχύος του περιβάλλοντος αξιολόγησης των αλγορίθμων.

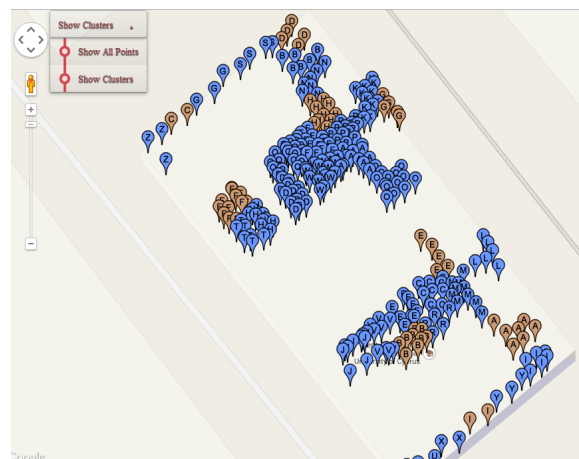


Σχήμα 5.6: Μενού Ιστοσελίδας

Όπως βλέπουμε στο Σχήμα 5.6 υποστηρίζονται δύο λειτουργίες. Παρουσίαση των αποτυπωμάτων στον χρήστη χωρίς ομαδοποίηση (βλέπε 5.7) και με ομαδοποίηση (βλέπε Σχήμα 5.8).



Σχήμα 5.7: Αποτυπώματα κτιρίου Πληροφορικής χωρίς Ομαδοποίησης



Σχήμα 5.8: Αποτυπώματα κτιρίου Πληροφορικής με Ομαδοποίησης

Εξυπηρετητής Μηχανής Προβλέψεων

Ο εξυπηρετητής της Μηχανής Προβλέψεων υποστηρίζει δύο βασικές αιτήσεις πελάτη. Και οι δύο αιτήσεις είναι τύπου *GET* όπως αυτό περιγράφεται από το πρωτόκολλο *HTTP*⁷ και είναι οι ακόλουθες:

- Η πρώτη αίτηση που υποστηρίζει είναι ο υπολογισμός της τοποθεσίας του πελάτη και επιστροφή ασυμπίεστων συστοιχιών σε αυτόν, η τρέχουσα συστοιχία του μαζί με τις προβλέψεις.
- Η δεύτερη αίτηση χρησιμοποιείται όταν υπάρξει επιβεβαίωση οποιασδήποτε συστοιχίας από την μηχανή Προφόρτωσης. Όταν ο πελάτης προσεγγίσει ικανοποιητικά μια συμπίεσμένη συστοιχία ζητείται ο αντίστοιχος μερικός ραδιοχάρτης της. Αντίθετα όταν ο πελάτης προσεγγίσει ικανοποιητικά μια μη συμπίεσμένη συστοιχία ζητείται από τον εξυπηρετητή η επόμενη πιο πιθανή μη συμπίεσμένη συστοιχία.

Η επικοινωνία μεταξύ πελάτη και εξυπηρετητή γίνεται με την μορφή *JSON*, το οποίο είναι πρότυπο αναπαράστασης δεδομένων σε μορφή κλειδιών/τιμών (key-value). Επιπρόσθετα χρησιμοποιείται ευρέως στην ανταλλαγή δεδομένων μεταξύ εξυπηρετητών και Διαδικτυακών εφαρμογών.

⁷[ftp://www.ietf.org/rfc/rfc2616.txt](http://www.ietf.org/rfc/rfc2616.txt)

```

1 http://vectormap1.in.cs.ucy.ac.cy
  :9000/location?rss_values=
  [{ "bssid": "00:1f:6c:a8:e9:12", "
    heading":0.0, "rss":4}
3 ,{ "bssid": "00:1f:6c:a8:e9:11", "
    heading":0.0, "rss":1}
  ,{ "bssid": "00:1f:6c:a8:e9:10", "
    heading":0.0, "rss":2}
5 ,{ "bssid": "d4:d7:48:b0:8f:40", "
    heading":0.0, "rss":6}
  ,{ "bssid": "d4:d7:48:d8:2c:70", "
    heading":0.0, "rss":-9}
7 ,{ "bssid": "00:0b:fd:4a:71:a2", "
    heading":0.0, "rss":-3}
  ,{ "bssid": "00:25:9c:99:90:d4", "
    heading":0.0, "rss":5}]
9 &cached_clusters={"lookahead_win": "
    1", "numof_pred": "1",
    "pred_algo": "cb_algorithm", "
    visited_clusters": []}

```

Predictions Request

```

http://vectormap1.in.cs.ucy.ac.
cy:9000/get_cluster_ex_conf?
cid=28
&cached_clusters={"lookahead_win
": "1", "numof_pred": "1",
"pred_algo": "cb_algorithm", "
visited_clusters": [1,28]}

```

Cluster Request

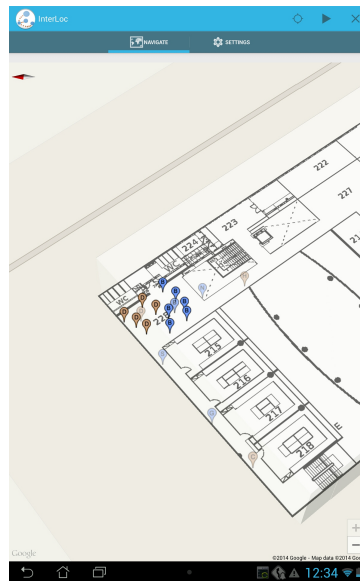
Σχήμα 5.9: Βασικές Αιτήσεις Πελάτη

Στην αίτηση υπολογισμού τοποθεσίας και προβλέψεων του Σχήματος 5.9 αποστέλλονται τα σημεία πρόσβασης που ακούει ο χρήστης κινητής συσκευής μαζί με την ισχύ που τα ακούει. Επίσης, στέλλει κάποιες επιπλέον παραμέτρους όπως είναι ο αριθμός των προβλέψεων, ο αλγόριθμος πρόβλεψης που θα χρησιμοποιηθεί, οι συστοιχίες που έχει προσπελάσει ήδη και το παράθυρο εκπαίδευσης του γραφήματος.

Στην αίτηση συστοιχίας του Σχήματος 5.9 αποστέλλεται ένας ακέραιος αριθμός, ο οποίος αντιπροσωπεύει την συστοιχία που επιβεβαιώθηκε από την μηχανή Προφόρτωσης. Προφανώς η αίτηση αυτή είναι πολύ πιο συμπαγής από την αίτηση υπολογισμού τοποθεσίας και προβλέψεων και συμβάλει στην μείωση της επικοινωνίας με τον εξυπηρετητή. Οι επιπρόσθετοι παράμετροι είναι ίδιες με την αίτηση *Predictions Request*.

5.4.2 Γραφική Διεπαφή Χρήστη

Η Γραφική Διεπαφή Χρήστη *GUI* περιέχει όλες τις απαραίτητες λειτουργίες εκτέλεσης των Αλγορίθμων Προφόρτωσης που προτάθηκαν σε αυτή την Διπλωματική Εργασία. Το *GUI* χωρίζεται σε δύο Διεπαφές. Η πρώτη Διεπαφή είναι αυτή η οποία περιέχει τις απαραίτητες ρυθμίσεις των αλγορίθμων καθώς και κάποιες ρυθμίσεις προγραμματιστή (Ενεργοποίηση Αποσφαλμάτωσης). Η δεύτερη διεπαφή, περιέχει ένα απλό περιβάλλον εκτέλεσης των αλγορίθμων Πρόβλεψης και Προφόρτωσης (βλέπε 5.10).



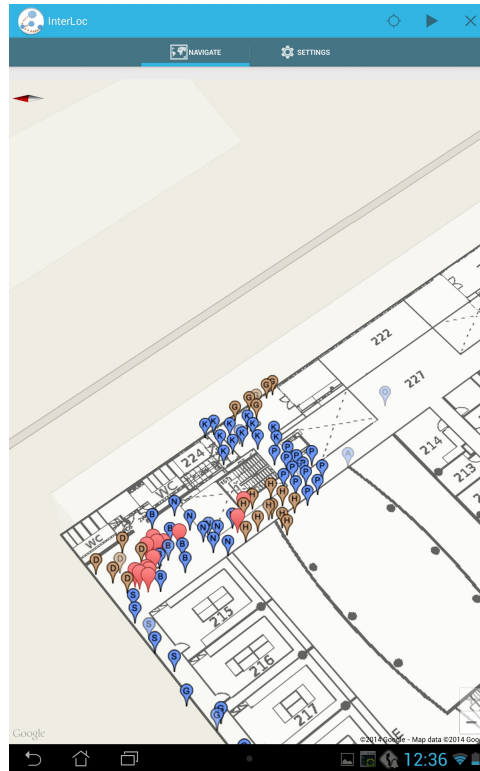
Σχήμα 5.10: Διεπαφή Πλοήγησης Χρήστη

Το περιβάλλον πλοήγησης περιέχει τρία κουμπιά (βλέπε Σχήμα 5.11). Το κουμπί με το κίτρινο περίγραμμα αρχικοποιεί το πείραμα, το κουμπί που περιβάλλεται με πράσινο χρώμα καθαρίζει το τρέχον πείραμα από την οθόνη της εφαρμογής και το κουμπί με το κόκκινο περίγραμμα εκτελεί μια γεωτοποθέτηση. Επίσης φαίνονται οι δύο καρτέλες με τις ονομασίες *Navigate* και *Settings* με τις οποίες ο χρήστης μπορεί να αλλάξει από τις ρυθμίσεις στο περιβάλλον πλοήγησης.



Σχήμα 5.11: Κουμπιά Αρχικοποίησης-Πλοήγησης

Στην διεπαφή που φαίνεται στο Σχήμα 5.10 υπάρχουν προφορτωμένες δύο συστοιχίες αποτυπωμάτων. Μετά από μερική πλοήγηση του χρήστη βλέπουμε ότι έχουν προφορτωθεί ακόμα επτά συστοιχίες (Σχήμα 5.12).

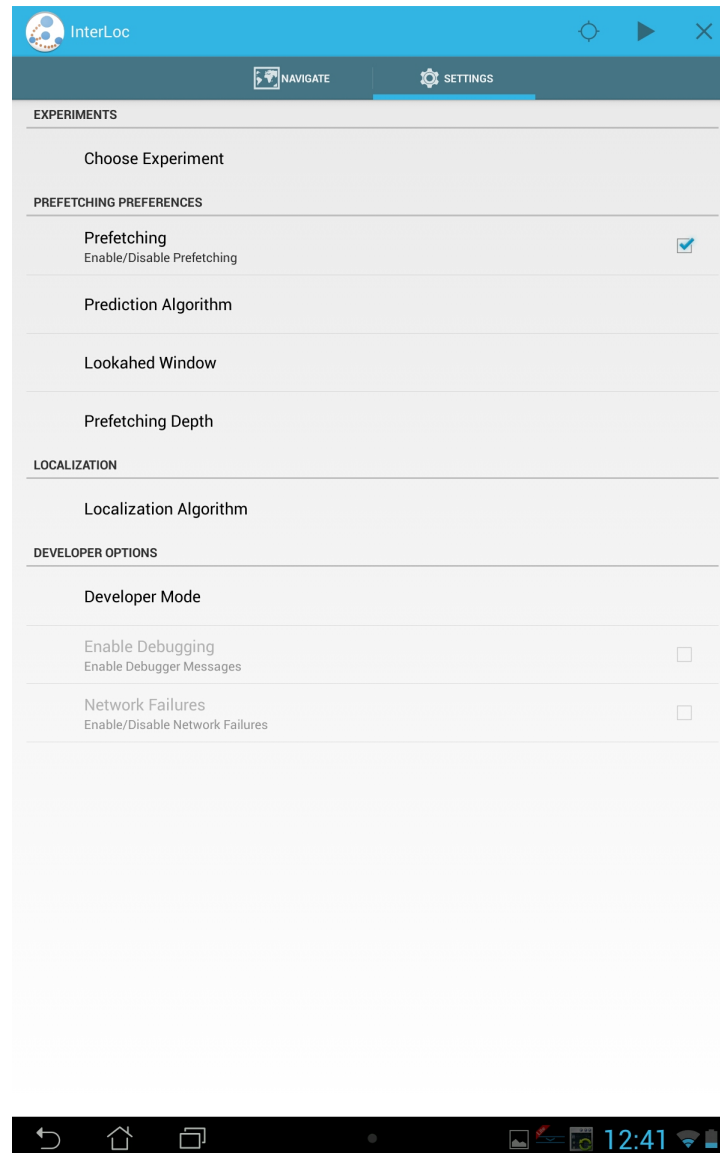


Σχήμα 5.12: Πλοήγηση Χρήστη στον χώρο

Τέλος, η διεπαφή των ρυθμίσεων φαίνεται στο Σχήμα 5.13 και περιλαμβάνει τις εξής ρυθμίσεις:

- Επιλογή Πειράματος (Διαδρομής). *Η τρέχουσα υλοποίηση περιέχει δύο διαδρομές.*
- Ενεργοποίηση/Απενεργοποίηση της Προφόρτωσης Ραδιοχαρτών. Σε περίπτωση που απενεργοποιηθεί η Προφόρτωση, ολόκληρος ο ραδιοχάρτης φορτώνεται στην συσκευή πριν την πλοήγηση (Client Side Radiomap).
- Αλγόριθμος Πρόβλεψης (CBA ή BBA). *Επιλογή αλγορίθμου Προφόρτωσης.*
- Βάθος Προφόρτωσης. *Αριθμός συστοιχιών προς προφόρτωση.*
- Αλγόριθμος Γεωτοποθέτησης (KNN, WKNN, MMSE, WMMSE)
- Ρυθμίσεις Προγραμματιστή

- Ενεργοποίηση/Απενεργοποίηση Αποσφαλμάτωσης. Δηλαδή, εμφάνιση αποστάσεων στον πελάτη από την τρέχουσα συστοιχία του προς τις γειτονικές.
- Ενεργοποίηση/Απενεργοποίηση Αποτυχιών Δικτύου. *Διακοπή της λειτουργίας του WiFi κατά την διάρκεια της γεωτοποθέτησης.*



Σχήμα 5.13: Ρυθμίσεις Εφαρμογής

Κεφάλαιο 6

Πειραματική Αξιολόγηση

6.1	Μεθοδολογία Αξιολόγησης	66
6.1.1	Περιγραφή Σεναρίου	67
6.1.2	Δεδομένα Χαρτών Τιμών Ισχύος	68
6.1.3	Αλγόριθμοι και Μετρικές	69
6.2	Αποτελέσματα Πειραματικής Αποτίμησης	69
6.2.1	Αποτελέσματα Αξιολόγησης ως προς τον αριθμό των Bytes	69
6.2.2	Αποτελέσματα Αξιολόγησης ως προς τον χρόνο εξυπηρέτησης	70
6.2.3	Αποτελέσματα Αξιολόγησης ως προς την ακρίβεια της γεωτοπο- θέτησης	71

Σε αυτό το κεφάλαιο περιγράφεται το πλαίσιο αξιολόγησης των αλγορίθμων, το οποίο αποτελείται από ένα Κατανεμημένο Προσομοιωτή γραμμένο στην προγραμματιστική γλώσσα *Java*. Ο αλγόριθμος Πρόβλεψης βρίσκεται στον εξυπηρετητή και από πίσω του βρίσκεται η υπολογιστική δύναμη μιας συστοιχίας Hadoop. Ο αλγόριθμος Προφόρτωσης υλοποιήθηκε στον πελάτη, πάνω στην πλατφόρμα *Android* και με την χρήση της προγραμματιστικής γλώσσας *Java*. Θα γίνει αξιολόγηση των αλγορίθμων Πρόβλεψης και Προφόρτωσης χρησιμοποιώντας ως μετρικές αξιολόγησης τον αριθμό των *Bytes* που ανταλλάσσονται μεταξύ εξυπηρετητή και πελάτη και τον απαιτούμενο χρόνο εξυπηρέτησης. Οι αλγόριθμοι Πρόβλεψης και Προφόρτωσης θα συγκριθούν με την προσέγγιση όπου ολόκληρος ο Ραδιοχάρτης κατεβαίνει στην κινητή συσκευή εξαρχής (πριν την έναρξη της γεωτοποθέτησης από τον χρήστη). Ας ονομάσουμε την προσέγγιση αυτή *Client Side*

Radiomap (CSR). Προφανώς με τον αλγόριθμο *CSR* ο πελάτης μπορεί να κινείται στον χώρο χωρίς να ανησυχεί για την σύνδεση του στο Διαδίκτυο. Ο αλγόριθμος *CSR* έχει στην κατοχή του κατά την διάρκεια ολόκληρης της πλοήγησης του χρήστη σε ένα κτίριο, ολόκληρο το σύνολο των αποτυπωμάτων (*RSS Fingerprints*). Αυτό έχει ως συνέπεια ο αλγόριθμος να παρέχει γεωτοποθέτηση υψηλής ακρίβειας όταν δεν υπάρχει σύνδεση στο Διαδίκτυο.

Από την άλλη πλευρά, οι αλγόριθμοι Πρόβλεψης και Προφόρτωσης δεν έχουν στην κατοχή τους όλα τα αποτυπώματα Ισχύος Λαμβανόμενου Σήματος (*RSS Fingerprints*), πριν την έναρξη της γεωτοποθέτησης του πελάτη. Υπενθυμίζεται ότι εξερευνούμε περιβάλλοντα με Διακοπτόμενη Σύνδεση στο Διαδίκτυο (*Intermittent Connectivity βλέπε υπό-κεφάλαιο 3.2*). Αντίθετα με τον αλγόριθμο *CSR* η σύνδεση του πελάτη στο Διαδίκτυο διακόπτεται ανά τακτά χρονικά διαστήματα. Αυτό έχει ως αποτέλεσμα ο πελάτης να μην έχει ολόκληρο τον Ραδιοχάρτη του κτιρίου στην συσκευή του αλλά να έχει μερικούς Ραδιοχάρτες. Προφανώς αυτό μειώνει την ακρίβεια της γεωτοποθέτησης σημαντικά, ιδίως όταν ο χρήστης κινείται σε χώρους για τους οποίους δεν έχει μερικό Ραδιοχάρτη. Σε αυτό το κεφάλαιο θα γίνει αποτίμηση της ακρίβειας γεωτοποθέτησης των Αλγορίθμων Πρόβλεψης-Προφόρτωσης όταν συμβαίνουν αποτυχίες στο Δίκτυο του χρήστη. Η ακρίβεια αυτή *AC*, θα υπολογιστεί χρησιμοποιώντας την εξίσωση του Μέσου Απόλυτου Σφάλματος (*Mean Absolute Error*).

Ορισμός 7 Η ακρίβεια *AC* υπολογίζεται βρίσκοντας την απόλυτη διαφορά μεταξύ της τοποθεσίας που δίνει ο *CSR* (έστω (x_m, y_m)) και της τοποθεσίας που δίνει ο οποιοσδήποτε αλγόριθμος Πρόβλεψης-Προφόρτωσης (έστω (x_j, y_j)). Τέλος, ας ορίσουμε τον αριθμό των προηγούμενων τοποθεσιών ως *n*. Τότε το σφάλμα υπολογίζεται χρησιμοποιώντας την πιο κάτω εξίσωση:

$$AC = \frac{1}{n} \sum_{i=1}^n (|x_m - x_j|_i + |y_m - y_j|_i)$$

Σκοπός μας είναι η ελαχιστοποίηση του πιο πάνω σφάλματος και η παροχή όσο το δυνατόν μεγαλύτερη ακρίβεια γεωτοποθέτησης.

6.1 Μεθοδολογία Αξιολόγησης

Για την αξιολόγηση των αλγορίθμων θα χρησιμοποιηθεί μια τυπική διαδρομή χρήστη των 17 σημείων. Για κάθε σημείο της διαδρομής ο πελάτης ακούει κάποια Ασύρματα

Σημεία Πρόσβασης με κάποια ισχύ. Στην πρώτη γεωτοποθέτηση ο πελάτης ζητά από τον εξυπηρετητή την συστοιχία στην οποία βρίσκεται. Μαζί με την τρέχουσα συστοιχία του χρήστη αποστέλλονται από τον εξυπηρετητή και κάποιος αριθμός από προβλέψεις. Σημειώνεται επίσης ότι ο πελάτης βρίσκεται σε περιβάλλον διακοπτόμενης σύνδεσης με το Διαδίκτυο. Πιο ειδικά, στο σενάριο αξιολόγησης ο πελάτης έχει σύνδεση με το Διαδίκτυο για τις πρώτες 4 γεωτοποθετήσεις. Για τις υπόλοιπες 13 γεωτοποθετήσεις ο πελάτης κινείται στον χώρο χωρίς καμία επικοινωνία με τον εξυπηρετητή. Το σενάριο που μόλις περιγράφηκε λοιπόν, έχει ποσοστό αποτυχίας σύνδεσης στον εξυπηρετητή 76%. Ο πελάτης γεωτοποθετείται με κάποιο σφάλμα στο προηγούμενο σενάριο. Για την ελαχιστοποίηση του σφάλματος δοκιμάζουμε διάφορα μεγέθη προφόρτωσης. Με την αύξηση του μεγέθους προφόρτωσης αναμένουμε το σφάλμα γεωτοποθέτησης να ελαχιστοποιείται.

6.1.1 Περιγραφή Σεναρίου

Όπως έχει αναφερθεί ήδη χρησιμοποιείται ένα σενάριο με 17 γεωτοποθετήσεις όπου ένας πελάτης εκτελεί κανονική επικοινωνία με τον εξυπηρετητή. Στην 4η γεωτοποθέτηση όμως, η επικοινωνία μεταξύ πελάτη και εξυπηρετητή διακόπτεται από μια αποτυχία δικτύου (π.χ. ο χρήστης βγαίνει εκτός εμβέλειας των Σημείων Πρόσβασης του Δικτύου). Το σενάριο μέχρι πριν την αποτυχία του δικτύου φαίνεται στο Σχήμα 6.1.



Σχήμα 6.1: Πλοήγηση μέχρι πριν την αποτυχία του δικτύου

Ο πελάτης κατόρθωσε να φέρει εφτά μερικούς ραδιοχάρτες ¹ πριν να αποτύχει το δίκτυο. Τονίζεται ότι τα κόκκινα σημεία είναι το μονοπάτι του χρήστη ενώ τα υπόλοιπα σημεία στον χάρτη είναι Αποτυπώματα Τιμών Ισχύος. Όπως βλέπουμε ο χρήστης είναι σε θέση να πλοηγηθεί χωρίς πρόβλημα μέχρι την συστοιχία Η. Η συνέχεια της πλοήγησης φαίνεται στο Σχήμα 6.2.



Σχήμα 6.2: Πλοήγηση μετά την αποτυχία του δικτύου (Βάθος Προφόρτωσης: 3)

6.1.2 Δεδομένα Χαρτών Τιμών Ισχύος

Για την αξιολόγηση των αλγορίθμων χρησιμοποιήθηκαν πραγματικά Δεδομένα Χαρτών Τιμών Ισχύος. Πιο συγκεκριμένα, χρησιμοποιήθηκε ο Ραδιοχάρτης του κτιρίου Πληροφορικής Πανεπιστημίου Κύπρου. Ο χάρτης αυτός παράχθηκε από την περισυλλογή δεδομένων στο κτήριο Πληροφορικής του Πανεπιστημίου Κύπρου. Για την χαρτογράφηση του κτιρίου χρησιμοποιήθηκε μια κινητή συσκευή με λειτουργικό σύστημα Android (HTC Desire) για την περισυλλογή Αποτυπωμάτων Ισχύος Λαμβανόμενου Σήματος (RSS Fingerprints) σε 1125 ξεχωριστές τοποθεσίες στους τέσσερις ορόφους του κτιρίου (συνολικά 33,225 μετρήσεις τιμών Ισχύος Λαμβανόμενου Σήματος). Υπάρχουν 120 Ασύρματα Σημεία Πρόσβασης Διαδικτύου εγκατεστημένα στους τέσσερις ορόφους του κτιρίου, συμπεριλαμβανομένων και των Ασύρματων Σημείων Πρόσβασης Διαδικτύου γειτονικών κτιρίων, τα οποία ακούγονται μερικώς σε κάποια σημεία του κτιρίου Πληροφορικής.

¹ Υπενθυμίζεται ότι κάθε διαφορετικό χρώμα και γράμμα είναι και μια διαφορετική συστοιχία

6.1.3 Αλγόριθμοι και Μετρικές

Η πειραματική αποτίμηση έχει ως στόχο την αξιολόγηση του Αλγόριθμου Πρόβλεψης και Προφόρτωσης, από την μια πλευρά χρησιμοποιώντας τον αλγόριθμο *CBA* για τις προβλέψεις και από την άλλη πλευρά χρησιμοποιώντας τον αλγόριθμο *BBA*. Η αξιολόγηση του Αλγόριθμου Πρόβλεψης και Προφόρτωσης θα περιλαμβάνει υπολογισμό της ακρίβειας γεωτοποθέτησης του αλγόριθμου (τον λόγο δηλαδή των ορθών προβλέψεων). Περαιτέρω, θα γίνουν οι ακόλουθες συγκρίσεις:

- Σύγκριση του Αλγόριθμου Πρόβλεψης και Προφόρτωσης που κάνει χρήση του αλγορίθμου προβλέψεων *CBA*, με την προσέγγιση *CSR*. Εδώ θα γίνει σύγκριση ως προς τον αριθμό των *Bytes* που ανταλλάσσονται μεταξύ εξυπηρετητή και πελάτη και ως προς τον απαιτούμενο χρόνο εξυπηρέτησης. Υπενθυμίζεται ότι ο πελάτης λαμβάνει μερικούς Ραδιοχάρτες όταν γίνεται χρήση του Αλγόριθμου Πρόβλεψης και Προφόρτωσης και όχι ολόκληρο τον Ραδιοχάρτη. Αναμένεται να έχουμε σημαντικά οφέλη και στις δύο πιο πάνω μετρικές.
- Σύγκριση του Αλγόριθμου Πρόβλεψης και Προφόρτωσης που κάνει χρήση του αλγορίθμου προβλέψεων *BBA*, με την προσέγγιση *CSR*. Όπως και πριν θα γίνει σύγκριση ως προς τον αριθμό των *Bytes* που ανταλλάσσονται μεταξύ εξυπηρετητή και πελάτη και ως προς τον απαιτούμενο χρόνο εξυπηρέτησης. Δεδομένου ότι ο πελάτης λαμβάνει μερικούς Ραδιοχάρτες όταν γίνεται χρήση του Αλγόριθμου Πρόβλεψης και Προφόρτωσης, αναμένεται να έχουμε οφέλη και στις δύο πιο πάνω μετρικές.

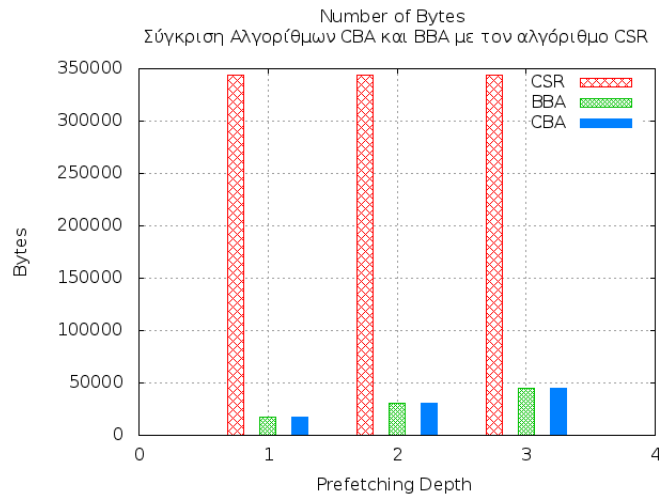
Αναμένουμε οι δυο πιο πάνω συγκρίσεις να είναι υπέρ του Αλγόριθμου Πρόβλεψης και Προφόρτωσης. Πιο αναλυτικές περιγραφές θα δοθούν πιο μετά μαζί με τις γραφικές παραστάσεις.

6.2 Αποτελέσματα Πειραματικής Αποτίμησης

6.2.1 Αποτελέσματα Αξιολόγησης ως προς τον αριθμό των Bytes

Εδώ περιγράφεται το ποσοστό της επικοινωνίας μεταξύ πελάτη και εξυπηρετητή κατά την διάρκεια που υπάρχει σύνδεση μαζί του. Όπως έχει τονιστεί μέχρι τώρα οι αλγόριθμοι

Πρόβλεψης και Προφόρτωσης δουλεύουν πάνω σε επίπεδο συστοιχίας (μερικοί ραδιοχάρτες). Επίσης ο αλγόριθμος Προφόρτωσης χρησιμοποιεί τις πιο συμπαγείς αιτήσεις που περιγράφηκαν στο υπό-κεφάλαιο 5.9. Είναι αναμενόμενο λοιπόν, το πλεονέκτημα των αλγορίθμων *CBA* και *BBA* έναντι του αλγορίθμου *CSR*. Τα αποτελέσματα των μετρήσεων φαίνονται στο Σχήμα 6.3.

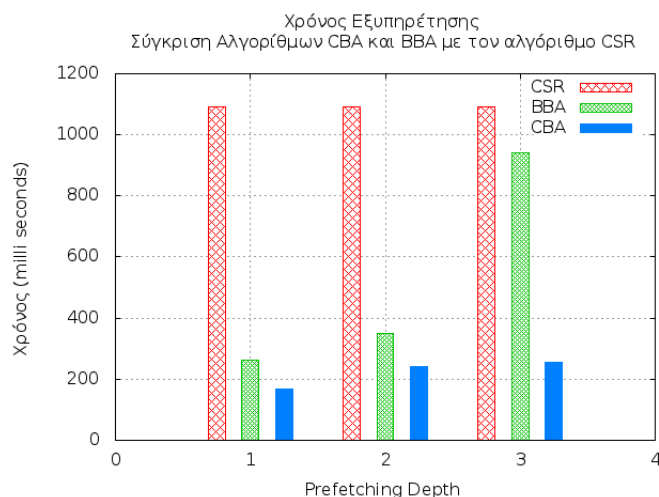


Σχήμα 6.3: Κόστος Επικοινωνίας Αλγορίθμων σε Bytes

Ο αλγόριθμος *CSR* φορτώνει τον Ραδιοχάρτη πριν την έναρξη της γεωτοποθέτησης. Στην περίπτωση μας ο Ραδιοχάρτης είναι γύρω στα 340 KB, ενώ για τους αλγορίθμους Πρόβλεψης και Προφόρτωσης το συνολικό μέγεθος μερικών Ραδιοχαρτών εξαρτάται από το ποσό της προφόρτωσης που γίνεται. Παρατηρούμε ότι όσο αυξάνεται το ποσό της προφόρτωσης αυξάνεται και ο αριθμός των Bytes που ανταλλάσσονται μεταξύ εξυπηρετητή και πελάτη. Παρόλα αυτά ο αριθμός των Bytes για τους αλγόριθμους Προφόρτωσης δεν προσεγγίζει ούτε στο ελάχιστο τον αριθμό των Bytes για την προσέγγιση *CSR*. Έχουμε λοιπόν σημαντική εξοικονόμηση Πόρων του Δικτύου.

6.2.2 Αποτελέσματα Αξιολόγησης ως προς τον χρόνο εξυπηρέτησης

Σε αυτό το σημείο δίνεται ο χρόνος εξυπηρέτησης ενός πελάτη από τον εξυπηρετητή. Ο χρόνος αυτός διαφέρει ανάλογα με τον αλγόριθμο που χρησιμοποιεί. Η προσέγγιση *CSR* φορτώνει ολόκληρο τον Ραδιοχάρτη στην συσκευή πριν την έναρξη της γεωτοποθέτησης. Για αυτό τον λόγο αναμένουμε ο χρόνος φόρτωσης του Ραδιοχάρτη για την τεχνική *CSR* να είναι σταθερός και ο μεγαλύτερος σε σχέση με τους αλγορίθμους Πρόβλεψης *CBA* και *BBA*. Τα αποτελέσματα των μετρήσεων φαίνονται στον Ορισμό 7



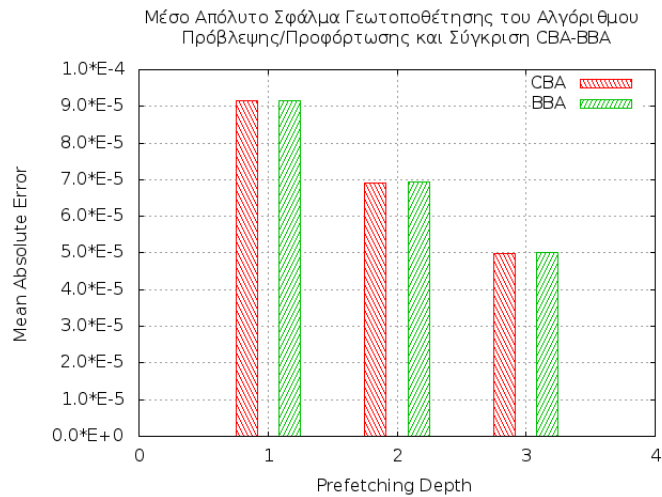
Σχήμα 6.4: Σύγκριση Αλγορίθμων CBA, BBA και CSR ως προς τον Χρόνο Εξυπηρέτησης

Οι αλγόριθμοι *CBA* και *BBA* έχουν μικρότερο χρόνο εξυπηρέτησης σε σχέση με την προσέγγιση *CSR* καθώς δουλεύουν σε επίπεδο συστοιχίας. Επίσης παρατηρούμε ότι ο αλγόριθμος Προβλέψεων *BBA* έχει μεγαλύτερο χρόνο εξυπηρέτησης σε σχέση με τον *CBA*.

6.2.3 Αποτελέσματα Αξιολόγησης ως προς την ακρίβεια της γεωτοποθέτησης

Για την ακρίβεια της γεωτοποθέτησης έχουμε δοκιμάσει διάφορα μεγέθη Πρόβλεψης και Προφόρτωσης με στόχο την ελαχιστοποίηση του σφάλματος γεωτοποθέτησης. Περαιτέρω, γίνεται σύγκριση των αλγορίθμων Πρόβλεψης *CBA* και *BBA*. Τα αποτελέσματα αυτά δίνονται στο Σχήμα 6.5. Η περιγραφή υπολογισμού του σφάλματος έχει περιγραφεί στον Ορισμό 7 και είναι η απόκλιση της γεωτοποθέτησης οποιουδήποτε αλγόριθμου Πρόβλεψης-Προφόρτωσης ως προς την ιδανική περίπτωση κατοχής ολόκληρου του Ραδιοχάρτη. Αναμένουμε ότι η αύξηση του μεγέθους Προφόρτωσης να αυξάνει και την ακρίβεια της γεωτοποθέτησης. Με την προφόρτωση περισσότερων συστοιχιών ο πελάτης έχει περισσότερα αποτυπώματα τιμών ισχύος. Με την αύξηση των αποτυπωμάτων τιμών ισχύος αυξάνεται και η ακρίβεια της γεωτοποθέτησης του χρήστη.

Σημειώνεται ότι στο Σχήμα 6.5 παραλείπεται ο αλγόριθμος *CSR* διότι η απόκλιση του από τον εαυτό του είναι σαφώς μηδέν. Παρατηρούμε ότι όσο αυξάνεται το μέγεθος της Προφόρτωσης μειώνεται και το σφάλμα γεωτοποθέτησης. Βέβαια, αυτό δεν συμβαίνει με μηδενικό κόστος, όσο αυξάνεται το μέγεθος Προφόρτωσης αυξάνεται και το ποσό των



Σχήμα 6.5: Σύγκριση Αλγορίθμων CBA, BBA και CSR ως προς το σφάλμα γεωτοποθέτησης

δεδομένων που ανταλλάσσονται μεταξύ εξυπηρετητή και πελάτη. Παρόλα αυτά η αύξηση αυτή δεν είναι ιδιαίτερα μεγάλη (βλέπε Σχήμα 6.3).

Κεφάλαιο 7

Συμπεράσματα και Μελλοντική Εργασία

7.1	Συμπεράσματα	73
7.2	Μελλοντική Εργασία	75
7.2.1	Βελτίωση Υποδομής	75
7.2.2	Αλγοριθμικές Βελτιώσεις	75

7.1 Συμπεράσματα

Σε αυτή την Διπλωματική Εργασία υλοποιήθηκαν και αξιολογήθηκαν δύο νέοι αλγόριθμοι Πρόβλεψης αλλά και ένας αλγόριθμος Προφόρτωσης Ραδιοχαρτών. Οι δύο αλγόριθμοι Προβλέψεων *CBA* και *BBA* χρησιμοποιούν το υφιστάμενο ιστορικό των τροχιών της κοινότητας σε ένα εσωτερικό χώρο, για την κατασκευή ενός γραφήματος το οποίο περιγράφει τις πιθανότητες μετάβασης από μια περιοχή σε άλλη, αλλά και την σημαντικότητα της κάθε περιοχής του κτιρίου. Δεδομένου μιας περιοχής οι προηγούμενοι αλγόριθμοι εισηγούνται τις επόμενες πιθανότερες περιοχές. Οι αλγόριθμοι είναι ιδανικοί για περιβάλλοντα με διακοπτόμενη σύνδεση στο διαδίκτυο, καθώς η χρήση τους επιτρέπει την πλοήγηση του χρήστη χρησιμοποιώντας μερικούς Ραδιοχάρτες, ελαχιστοποιώντας παράλληλα το σφάλμα πλοήγησης. Με την πρόταση των Αλγορίθμων Προφόρτωσης που ορίζονται σε αυτή την διπλωματική εργασία, δίνουμε την δυνατότητα σε χρήστες να πλοηγούνται με ικανοποιητική ακρίβεια, σε περιβάλλοντα διακοπτόμενης σύνδεσης.

Οι αλγόριθμοι Προφόρτωσης εκτελούν προ-ανάκτηση των μερικών Ραδιοχαρτών που είναι το πιθανότερο να προσπελαστούν από τον χρήστη. Αυτό όπως αποδεικνύεται πειραματικά ελαχιστοποιεί σε μεγάλο βαθμό το σφάλμα πλοήγησης .

Η υλοποίηση του συστήματος βασίστηκε πάνω στο μοντέλο αρχιτεκτονικής Εξυπηρετητή-Πελάτη *Server-Client*. Ο εξυπηρετητής είναι ταυτόχρονα και κύριος κόμβος μιας συστοιχίας *Hadoop*, όπου το κατανεμημένο σύστημα που παρέχει, χρησιμοποιείται από την βάση δεδομένων *HBase* για αύξηση της επεκτασιμότητας και της ανοχής σφαλμάτων. Οι αλγόριθμοι αξιολογήθηκαν χρησιμοποιώντας πραγματικά χωρικά δεδομένα και προσομοιώνοντας αποτυχίες στο δίκτυο. Ο συνδυασμός των Αλγορίθμων Πρόβλεψης και Προφόρτωσης έχει αποδειχθεί πειραματικά ότι ελαχιστοποιεί όχι μόνο τον χρόνο επικοινωνίας αλλά και το κόστος επικοινωνίας με τον εξυπηρετητή. Παρόλα αυτά, το βάθος Προφόρτωσης έχει αποδειχθεί πειραματικά ότι αυξάνει τον χρόνο και το κόστος επικοινωνίας με τον εξυπηρετητή. Παράλληλα, λόγω της φύσης των αλγορίθμων, γίνεται προφόρτωση των πιο δημοφιλών περιοχών του κτιρίου με αποτέλεσμα να ελαχιστοποιείται το σφάλμα πλοήγησης σε μεγαλύτερο βαθμό. Οι αλγόριθμοι συγκρίθηκαν με τον αλγόριθμο *CSR (Client Side Radiomap)* ο οποίος παρέχει πολύ υψηλή ακρίβεια γεωτοποθέτησης (διότι έχει στην διάθεση του ολόκληρο του Ραδιοχάρτη) και υπολογίστηκε το μέσο απόλυτο για τους δύο αλγόριθμους Προβλέψεων *CBA* και *BBA*. Η απόκλιση των δύο αλγορίθμων από τον αλγόριθμο *CSR* κυμαίνεται στα ίδια επίπεδα. Επιπλέον, το μέσο απόλυτο σφάλμα μειώνεται όσο αυξάνεται το βάθος της Προφόρτωσης.

Συμπεραίνουμε λοιπόν ότι το βάθος Προφόρτωσης πρέπει να γίνει προσεκτικά καθώς επηρεάζει τον χρόνο και το κόστος επικοινωνίας με τον εξυπηρετητή αλλά και την ακρίβεια της γεωτοποθέτησης. Το ιδανικό βάθος Προφόρτωσης διαφέρει από χώρο σε χώρο. Προφανώς σε ένα μικρό περιβάλλον, ένα μικρό βάθος προφόρτωσης παρέχει υψηλή ακρίβεια γεωτοποθέτησης διατηρώντας σε χαμηλά επίπεδα την επικοινωνία με τον εξυπηρετητή. Αντίθετα σε ένα μεγάλο περιβάλλον, ένα μεγαλύτερο μέγεθος παραθύρου χρειάζεται για την παροχή των προηγούμενων πλεονεκτημάτων.

7.2 Μελλοντική Εργασία

7.2.1 Βελτίωση Υποδομής

Λόγω της φύσης του προβλήματος αλλά και του τρόπου υλοποίησης των Αλγορίθμων, μια Βάση Δεδομένων Γραφημάτων (*Graph Database*) είναι ιδανική λύση για το πρόβλημα μας. Με μια τέτοια βάση μπορούμε να εκφράσουμε πιο φυσικά τους αλγόριθμους *CBA* και *BBA*. Οι Βάσεις Γραφημάτων παρέχουν γειτνίαση χωρίς την χρήση επιπρόσθετων ευρετηρίων στην βάση δεδομένων (*Index-Free Adjacency*). Αυτό σημαίνει ότι κάθε στοιχείο της βάσης έχει άμεσους δείκτες στα γειτονικά στοιχεία, καθιστώντας την χρήση ευρετηρίων για την εύρεση γειτόνων μη απαραίτητη.

7.2.2 Αλγοριθμικές Βελτιώσεις

Οι αλγόριθμοι *CBA* και *BBA* κατασκευάζουν τα γραφήματα εξαρτήσεων χρησιμοποιώντας το καθολικό ιστορικό κίνησης των χρηστών. Δεν λαμβάνουν υπόψη τους λοιπόν την χρονική στιγμή της κάθε τροχιάς. Έχοντας αυτή την πληροφορία, μπορούμε να κατασκευάσουμε διαφορετικά γραφήματα για διαφορετικά σημεία της μέρας. Για παράδειγμα, ένα γράφημα το οποίο να περιγράφει τις κινήσεις των χρηστών σε ένα εσωτερικό χώρο το πρωί, ένα δεύτερο γράφημα να περιγράφει τις κινήσεις των χρηστών το μεσημέρι κ.ο.κ.. Ας αναλογιστούμε λίγο την ορθότητα της υλοποίησης αυτής. Ένας υπάλληλος ο οποίος παραδίδει την αλληλογραφία στα γραφεία μιας εταιρίας, όπου τα γραφεία βρίσκονται σε δύο διαφορετικές περιοχές (π.χ. Α και Β) του ίδιου κτιρίου. Ο υπάλληλος παραδίδει την αλληλογραφία στα γραφεία της περιοχής Α το πρωί, ενώ την αλληλογραφία για την περιοχή Β την παραδίδει το μεσημέρι. Και αυτό δεν είναι ένα ειδικό σενάριο, όλοι μας ακολουθούμε διαφορετικά πρότυπα κίνησης σε διαφορετικές χρονικές στιγμές της μέρας (χωρίς όμως να το καταλαβαίνουμε πολλές φορές). Αυτή θα ήταν μια σημαντική μελλοντική βελτίωση των αλγορίθμων.

Σε μια πραγματική εφαρμογή αυτού του αλγορίθμου τα ιστορικά των χρηστών θα έρχονται ως ροές στο σύστημα. Προσαρμογή των αλγορίθμων έτσι ώστε να προσαρμόζουν γρήγορα και εύελικτα το γράφημα εξαρτήσεων. Επίσης κάποιες από τις ροές μπορεί να είναι κακόβουλες από χρήστες οι οποίοι έχουν ως σκοπό να μολύνουν το γράφημα με ψευδείς διαδρομές. Από πλευράς ασφάλειας λοιπόν, θα ήταν απαραίτητη η ανάπτυξη κάποιου αλγορίθμου για την ανάλυση και απόρριψη τέτοιων ροών από το σύστημα.

Όπως αναφέραμε ήδη το μέγεθος της Προφόρτωσης παίζει πολύ σημαντικό ρόλο στο ποσό της επικοινωνίας με τον εξυπηρετητή και στην ακρίβεια της γεωτοποθέτησης. Μια επέκταση του αλγορίθμου θα ήταν η δημιουργία κάποιου αλγορίθμου ο οποίος να αποφασίζει το μέγεθος της προφόρτωσης βάσει των διαστάσεων του κτιρίου.

Υλοποίηση κάποιου αλγορίθμου στην Μηχανή Προφόρτωσης που να υπολογίζει πότε να γίνεται προφόρτωση ολόκληρου του ραδιοχάρτη και πότε να γίνεται προφόρτωση μερικών ραδιοχαρτών (χρήση των αλγορίθμων Πρόβλεψης και Προφόρτωσης).

Τέλος, η μηχανή Προφόρτωσης η οποία εγκρίνει τις προβλέψεις ανάλογα με την απόσταση *Mahalanobis* που έχει ο χρήστης από μια πρόβλεψη, κάνει χρήση ενός κατωφλίου. Για να εγκριθεί μια συστοιχία η απόσταση *Mahalanobis* πρέπει να πέσει κάτω από το κατώφλι. Για την εκτέλεση ορθότερων αποφάσεων θα ήταν σημαντική η υλοποίηση ενός αλγορίθμου υπολογισμού του κατωφλίου.

Βιβλιογραφία

- [1] P.S. Bradley, Usama Fayyad, and Cory Reina. Scaling clustering algorithms to large databases. pages 9--15. AAAI Press, 1998.
- [2] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Comput. Netw. ISDN Syst.*, 30(1-7):107--117, April 1998.
- [3] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):4:1--4:26, June 2008.
- [4] Jeffrey Dean and et al. Mapreduce: Simplified data processing on large clusters, 2004.
- [5] Josep Domenech, Julio Sahuquillo, Jose A. Gil, and Ana Pont. The impact of the web prefetching architecture on the limits of reducing user's perceived latency. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, pages 740--744, Washington, DC, USA, 2006. IEEE Computer Society.
- [6] Sahuquillo J. Pont A. Domenech J., Gil J.A. Ddg: An efficient prefetching algorithm for current web generation. In *Proceedings of the 1st IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*. IEEE, 2006.
- [7] eMarketer Research Company. <http://www.emarketer.com/article/smartphone-users-worldwide-will-total-175-billion-2014/1010536>, 2014.
- [8] Lars George. Hbase: The definitive guide, 2011.

- [9] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung. The google file system. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, SOSP '03, pages 29--43, New York, NY, USA, 2003. ACM.
- [10] Yanying Gu, Anthony Lo, and Ignas G. Niemegeers. A survey of indoor positioning systems for wireless personal networks. *IEEE Communications Surveys and Tutorials*, 11(1):13--32, 2009.
- [11] Alex Holmes. Hadoop in practice, 2012.
- [12] Andreas Konstantinidis, Georgios Chatzimiloudis, Christos Laoudias, Silouanos Nicolaou, and Demetrios Zeinalipour-Yazti. Towards planet-scale localization on smartphones with a partial radiomap. In *Proceedings of the 4th ACM International Workshop on Hot Topics in Planet-scale Measurement*, HotPlanet '12, pages 9--14, New York, NY, USA, 2012. ACM.
- [13] Christos Laoudias, George Constantinou, Marios Constantinides, Silouanos Nicolaou, Demetrios Zeinalipour-Yazti, and Christos G. Panayiotou. The airplace indoor positioning platform for android smartphones. In *Proceedings of the 2012 IEEE 13th International Conference on Mobile Data Management (Mdm 2012)*, MDM '12, pages 312--315, Washington, DC, USA, 2012. IEEE Computer Society.
- [14] Binghao Li, James Salter, Andrew G. Dempster, and Chris Rizos. Indoor positioning techniques based on wireless lan. In *LAN, FIRST IEEE INTERNATIONAL CONFERENCE ON WIRELESS BROADBAND AND ULTRA WIDEBAND COMMUNICATIONS*, pages 13--16.
- [15] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120.

Παράρτημα Α

Κατανεμημένο Προγραμματιστικό Μοντέλο *MapReduce*

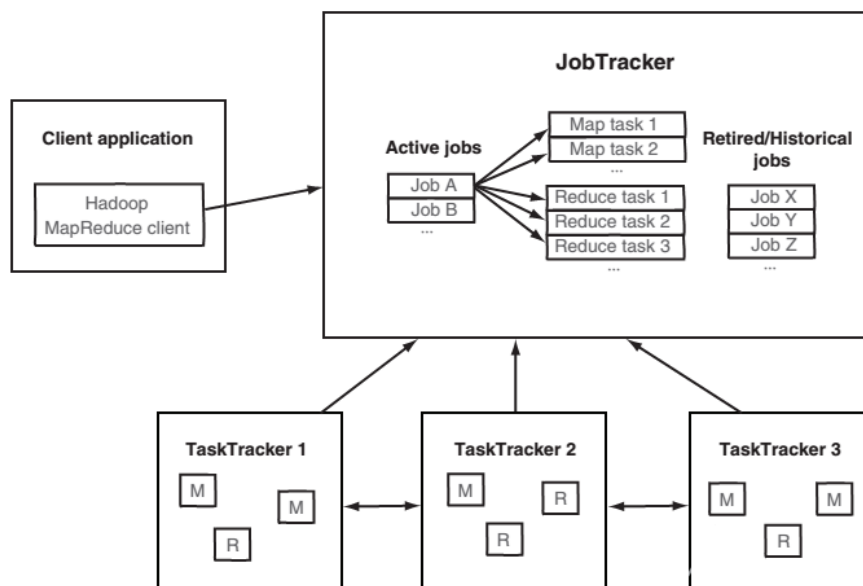
Είναι η ανοικτού κώδικα υλοποίηση του πλαισίου *MapReduce* που χρησιμοποιεί η *Google* για την ανάλυση των δεδομένων και παροχή υπηρεσιών της [4]. Το μοντέλο αυτό, απλοποιεί τον προγραμματισμό των Κατανεμημένων Συστημάτων. Στα Κατανεμημένα Συστήματα, που δεν κάνουν χρήση του πιο πάνω πλαισίου τα σφάλματα, ο παράλληλος υπολογισμός και η κατανομή της δουλειάς είναι ευθύνη του προγραμματιστή. Αντίθετα, σε ένα Κατανεμημένο Σύστημα που χρησιμοποιεί το πιο πάνω μοντέλο υπολογισμού, τα προβλήματα που μόλις αναφέραμε τα αντιμετωπίζει το ίδιο το πλαίσιο. Ευθύνη του προγραμματιστή είναι ο ορισμός δύο συναρτήσεων, της *Map* και της *Reduce*. Η ροή εκτέλεσης μιας *MapReduce* εργασίας έχει ως εξής:

1. Ο πελάτης καταχωρεί την δουλειά (*Job*) στην Συστοιχία του *Hadoop*. Η εργασία αυτή αποτελείται από τις δύο συναρτήσεις *Map* και *Reduce*.
2. Ο υπεύθυνος για την παραλαβή της εργασίας αυτής είναι ο κόμβος με την ονομασία *JobTracker*
3. Ο *JobTracker* χωρίζει δουλειά σε ένα αριθμό από εργασίες (*Tasks*). Κάθε τέτοιο *Task* εκτελεί τον κώδικα που όρισε ο προγραμματιστής στις συναρτήσεις *Map* και *Reduce*. Επιπλέον κάθε *Task* εκτελείται από μια διαφορετική διεργασία που ονομάζεται *TaskTracker*.
4. Το *MapReduce* πλαίσιο χωρίζει τα δεδομένα εισόδου σε κομμάτια (*Chunks* ή *Splits* σε ορολογία του *Hadoop*)
5. Για κάθε κομμάτι δημιουργείται μια εργασία (*Mapper*) η οποία εκτελεί τον κώδικα

της συνάρτησης *Map* πάνω στο συγκεκριμένο κομμάτι δεδομένων. Κάθε τέτοια εργασία παίρνει ως είσοδο ζευγάρια (key1-value1) από το συγκεκριμένο κομμάτι και μετά από κάποια επεξεργασία παράγει άλλα ενδιαμέσως ζευγάρια (key2-value2) .

6. Στην φάση του *Reduce* δημιουργείται μια εργασία (*Reducer*) για κάθε διαφορετικό ενδιαμέσως κλειδί (key2). Κάθε *Reducer* είναι υπεύθυνος για την παραλαβή όλων των τιμών (value2) που αντιστοιχούν στο key2. Συμπεραίνουμε λοιπόν ότι τις τιμές ενός κλειδιού, τις επεξεργάζεται όλες ένας και μόνο *Reducer*.

Μια γραφική αναπαράσταση των πιο πάνω σημείων φαίνεται στην εικόνα 7.1. Όπως είναι προφανές, ένας πελάτης που θέλει να καταχωρήσει μια δουλειά (*Job*) την καταχωρεί στον *JobTracker*. Η δουλειά του *JobTracker* είναι η αποδόμηση της δουλειάς σε ένα αριθμό από εργασίες (*Tasks*) και ο συντονισμός των κόμβων που τις εκτελούν. Τέλος, οι διεργασίες *TaskTracker* είναι αυτές που εκτελούν την πραγματική δουλειά (την συνάρτηση *Map* και *Reduce*).



Σχήμα 7.1: Το Κατανεμημένο Σύστημα αρχείων *HDFS* [11]

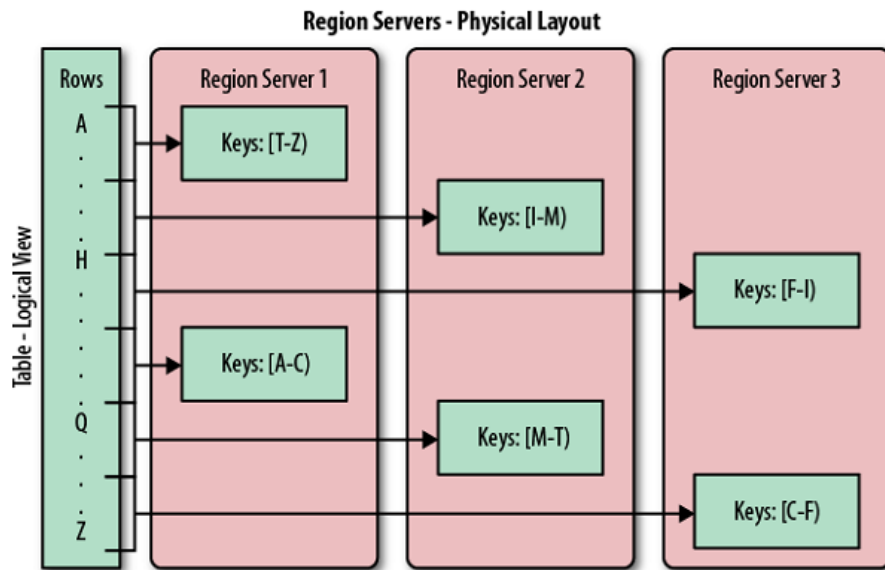
Παράρτημα Β

Περιοχές Δεδομένων και Επεκτασιμότητα

Στην *HBase*, Περιοχές Δεδομένων (*Regions*) είναι συνεχόμενες γραμμές, με λεξικογραφικά παρόμοια κλειδιά γραμμών (*rowkey*), ενός πίνακα που αποθηκεύονται μαζί στον δίσκο. Αυτές οι Περιοχές μεγαλώνουν και συρρικνώνονται ανάλογα με το πόσα δεδομένα έχει μια περιοχή. Όταν λοιπόν μια περιοχή μεγαλώσει αρκετά χωρίζεται σε δύο Περιοχές Δεδομένων. Από την άλλη πλευρά, όταν δύο Περιοχές Δεδομένων γίνουν σημαντικά μικρές συγχωνεύονται σε μια μεγαλύτερη Περιοχή. Ορθά μπορεί να υποθέσει κάποιος ότι ο αριθμός των περιοχών ενός πίνακα κυμαίνεται από μια σε εκατοντάδες. Αυτός ο αυτόματος τεμαχισμός και συγχώνευση των δεδομένων επιτρέπει στην *HBase* να επεκτείνεται σε χιλιάδες κόμβους. Την κάθε περιοχή ενός πίνακα, εξυπηρετεί ένας Εξυπηρετητής Περιοχής (*Region Server*). Παρόλα αυτά, ένας Εξυπηρετητής Περιοχής περιέχει περιοχές που ανήκουν σε διαφορετικούς πίνακες η κάθε μια.

Η πιο πάνω τεχνική ονομάζεται *Auto Sharding* και σε αυτή οφείλει η *HBase* την επεκτασιμότητα της. Όσο προστίθενται δεδομένα σε έναν πίνακα, τα δεδομένα αυτά κατανέμονται αυτόματα σε διαφορετικούς εξυπηρετητές. Στην ίδια τεχνική οφείλεται και η υψηλή διαθεσιμότητα της *HBase*. Στην εικόνα 7.2 φαίνεται ότι η κατανομή των δεδομένων γίνεται ανάλογα με το κλειδί της γραμμής. Για παράδειγμα γραμμές με κλειδιά που ξεκινούν από τα γράμματα Α, Β και Γ κατανέμονται στον Εξυπηρετητή Περιοχής 1.

Τα δεδομένα φυλάσσονται σε αρχεία τα οποία ονομάζονται *HFiles*. Κάθε *HFile* είναι μια ακολουθία από μπλοκ και διατηρεί ένα ευρετήριο στο τέλος του αρχείου, το οποίο φορτώνεται στην μνήμη όταν το *HFile* ανοίξει. Δεδομένου αυτού του ευρετηρίου οι επρωτήσεις ακρίβειας αλλά και οι επρωτήσεις ισότητας εκτελούνται πολύ αποδοτικά. Για την απάντηση επρωτήσεων δεδομένου ενός κλειδιού, γίνεται μια δυαδική αναζήτηση



Σχήμα 7.2: Τα δεδομένα κατανέμονται ανάλογα με το κλειδί της γραμμής τους σε ένα Εξυπηρετητή Περιοχής [8]

πάνω στο ευρετήριο του αρχείου. Έπειτα, το μπλοκ διαβάζεται από τον δίσκο. Συνεπώς, για την αναζήτηση σε ένα HFile υπάρχει κόστος 1 I/O, με την προϋπόθεση ότι το ευρετήριο είναι ήδη στην μνήμη.

Αρχιτεκτονική Κατανεμημένης Μη-Σχεσιακής Βάσης Δεδομένων HBase

Η αρχιτεκτονική της Κατανεμημένης Βάσης Δεδομένων HBase αποτελείται από τρία κύρια συστατικά:

- Κύριος Εξυπηρετητής (*Master Server*), όπου τρέχει η διεργασία *HMaster*
- Η βιβλιοθήκη Πελάτη (*Client Library*) η οποία χρησιμοποιείται για την αλληλεπίδραση με την βάση
- Εξυπηρετητές Περιοχής (*Region Servers*) οι οποίοι μπορούν να αυξηθούν ή να μειωθούν δυναμικά ανάλογα με τον φόρτο εξυπηρέτησης που έχει η βάση.

Η κύρια διεργασία, (*HMaster*) είναι υπεύθυνη για την ανάθεση περιοχών στους Εξυπηρετητές Περιοχής. Για την ανάθεση αυτή χρησιμοποιεί την υπηρεσία Apache ZooKeeper η οποία είναι μια αξιόπιστη, υψηλής διαθεσιμότητας κατανεμημένη υπηρεσία συντονισμού (*Coordination Service*). Επιπρόσθετα, η διεργασία *HMaster* είναι υπεύθυνη για την εξισορρόπηση του φόρτου περιοχών των Εξυπηρετητών Περιοχής. Για παράδειγμα, εάν

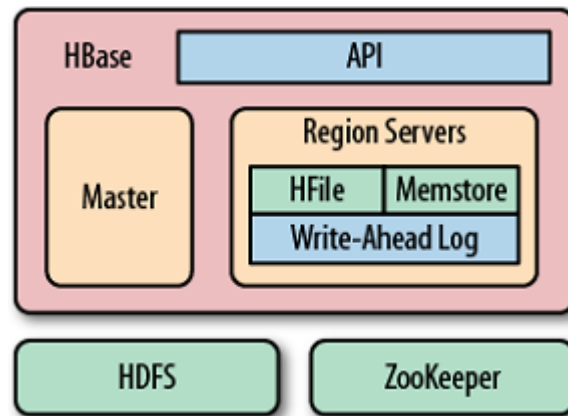
ένας εξυπηρετητής περιοχής z περιέχει πολλές περιοχές, θα έχει και υψηλό φόρτο εργασίας (διότι πολλοί πελάτες θα απευθύνονται σε αυτόν). Ένας από τους ρόλους της διεργασίας HMaster είναι η μετακίνηση κάποιων από τις περιοχές από τον κόμβο z σε άλλους εξυπηρετητές περιοχών με μικρότερο φόρτο. Ο Κύριος Εξυπηρετητής δεν λαμβάνει μέρος στην αποθήκευση των δεδομένων, καθώς ο φόρτος συντονισμού και εξισορρόπησης περιοχών είναι ήδη σημαντικός. Αντίθετα, διατηρεί και παρακολουθεί την κατάσταση του συμπλέγματος, είναι υπεύθυνος για τις αλλαγές στο Σχήμα της βάσης (π.χ. την δημιουργία πίνακα ή την πρόσθεση στήλης) . Ως φυσικό επακόλουθο της υλοποίησης της HBase, στην κορυφή του συστήματος Hadoop/HDFS, συναντάμε και πάλι την αρχιτεκτονική του κυρίου/υπηρετών (master/slaves). Στον κύριο κόμβο της Κατανεμημένης Βάσης HBase, ο οποίος αντιστοιχεί με τον κύριο κόμβο του συστήματος Hadoop/HDFS (δηλαδή είναι η ίδια φυσική μηχανή), τρέχει η διεργασία με το όνομα HMaster. Η HMaster είναι υπεύθυνη για την παρακολούθηση όλων των αντίστοιχων διεργασιών στο επίπεδο των υπηρετών που αφορούν την HBase, και ονομάζονται RegionServers. Χειρίζεται όλες τις αλλαγές στα μετά-δεδομένα της βάσης δεδομένων, εντοπίζει και επιλύει προβλήματα που προκύπτουν στους υπηρέτες, και αναθέτει περιοχές του πίνακα της βάσης δεδομένων σε αυτούς.

Οι εξυπηρετητές περιοχών είναι υπεύθυνοι για όλες τις αιτήσεις ανάγνωσης/εγγραφής για όλες τις περιοχές που περιέχουν, όπως επίσης και τον διαχωρισμό μιας περιοχής (σε μικρότερες περιοχές) όταν αυτή έχει ξεπεράσει κάποιο ορισμένο κατώφλι. Οι πελάτες προωθούν τα οποιαδήποτε αιτήματα σε αυτούς τους κόμβους απευθείας.

Επίσης το σύστημα της HBase μπορεί να εκκινήσει σε περιβάλλον πολλών κυρίων κόμβων, όπου ο πρώτος που θα του ανατεθεί από τον ZooKeeper (η διεργασία που είναι υπεύθυνη για τις διασυνδέσεις μεταξύ κατανεμημένων κόμβων στο HBase/HDFS) θα είναι ο κύριος κόμβος, και οι υπόλοιποι θα παραμείνουν απλά αδρανείς. Έτσι σε περιπτώσεις αποτυχίας του κύριου κόμβου, κάποιος άλλος κύριος κόμβος μπορεί να μεταβεί σε ενεργό κατάσταση και έτσι η βάση δεδομένων να συνεχίσει να λειτουργεί κανονικά. Επιπλέον, κατά την αποτυχία του κυρίου κόμβου και στην περίπτωση ύπαρξης μόνο ενός κυρίου κόμβου, για κάποιο χρονικό διάστημα οι πελάτες της βάσης δεδομένων μπορούν να συνεχίσουν να εξυπηρετούνται κανονικά, μιας και με την βοήθεια της διεργασίας ZooKeeper επικοινωνούν απευθείας με τους υπηρέτες, όπως γίνεται και στο σύστημα HDFS. Οι υπηρέτες στο σύστημα HBase, είναι οι ίδιοι φυσικοί κόμβοι με τους υπηρέτες του συστήματος Hadoop/HDFS, και τρέχουν την διεργασία RegionServer, η

οποία είναι υπεύθυνη για την εξυπηρέτηση και την διαχείριση των περιοχών του πίνακα που τους έχει ανατεθεί. Εξυπηρετούν για ανάγνωση και γραφή δεδομένων, μέσω των βασικών διεργασιών που παρέχει η HBase και θα επεξηγηθούν λεπτομερώς στην συνέχεια.

Το τελευταίο συστατικό το οποίο είναι απαραίτητο για την λειτουργία του συστήματος HBase, είναι ο προαναφερθείς ZooKeeper, ο οποίος είναι ενσωματωμένος στην HBase και συγκεκριμένα στον κύριο κόμβο. Η βασική λειτουργία του είναι η παροχή υψηλής απόδοσης συντονιστικής υπηρεσίας σε διαδεδομένες εφαρμογές. Εκτελεί κοινές λειτουργίες όπως ονοματοδοσία, διαχείριση ρυθμίσεων, συγχρονισμός μεταξύ των κόμβων. Η αφαιρετική αρχιτεκτονική της *HBase* φαίνεται στην εικόνα 7.3, όπου είναι ξεκάθαρο ότι η βάση κάνει εκτενή χρήση υπάρχοντων πλαισίων και υπηρεσιών (*ZooKeeper* και *HDFS*).



Σχήμα 7.3: Αρχιτεκτονική Κατανεμημένης Βάσης Δεδομένων HBase

