

Ατομική Διπλωματική Εργασία

**TrafficWare: Χαρτογράφηση Οδικής Κυκλοφορίας
με Τηλεπικοινωνιακά Δεδομένα**

Ιωάννα Γεωργίου

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2019

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

TrafficWare: Χαρτογράφηση Οδικής Κυκλοφορίας με Τηλεπικοινωνιακά Δεδομένα

Ιωάννα Γεωργίου

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 2019

Ευχαριστίες

Η εκπόνηση της παρούσας διπλωματικής δεν θα ήταν εφικτή χωρίς τη βοήθεια και στήριξη ορισμένων ατόμων, που θα ήταν παράληψή μου να μην ευχαριστήσω.

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

Ακολούθως, θα ήθελα να ευχαριστήσω την οικογένειά μου, που μου παρείχε ψυχολογική αλλά και οικονομική υποστήριξη καθ' όλη τη διάρκεια της ζωής μου, ειδικότερα στα τέσσερα χρόνια φοίτησής μου. Χωρίς αυτούς, τίποτα απ' όλα αυτά δεν θα ήταν εφικτό. Ακόμη, θα ήθελα να ευχαριστήσω ιδιαίτερα την αδελφή μου Μύρια, που με ενθάρρυνε γενικότερα, αλλά και για να συνεχίσω χωρίς να πτοούμαι από τις δυσκολίες και τα εμπόδια που έβρισκα μπροστά μου.

Τέλος, θα ήθελα να ευχαριστήσω τους φίλους μου Μιχάλη και Ευγενία, που μου παρείχαν στιγμές χαλάρωσης, διασκέδασης και συντροφιάς κατά την περίοδο της διπλωματικής μου εργασίας.

Περίληψη

Ένα ευρέως γνωστό πρόβλημα της σύγχρονης κοινωνίας είναι η οδική κίνηση. Η δημιουργία ενός συστήματος που θα παρέχει τη δυνατότητα παρατήρησης του οδικού δικτύου σε πραγματικό χρόνο, αποτελεί τη λύση στο πρόβλημα αυτό. Η παρούσα διπλωματική, στοχεύει στη μελέτη δημιουργίας τέτοιου συστήματος, καθώς βασίζεται σε τρία βασικά κριτήρια, τα οποία είναι η επίλυση του προβλήματος διατηρώντας την ιδιωτικότητα της θέσης των χρηστών, χρησιμοποιώντας την υπάρχουσα υποδομή (π.χ., κεραίες) και υλοποιώντας το σύστημα με ελάχιστο κόστος. Για τη μελέτη του συστήματος, έπρεπε να μελετηθούν παρόμοια συστήματα και τεχνολογίες που μπορεί το σύστημα να εκμεταλλευτεί για την επίλυση του προβλήματος, όπως και σχετικές έρευνες. Αναλύονται ακόμη, διάφοροι τρόποι για τη βέλτιστη υλοποίηση του συστήματος, μαζί με τα πλεονεκτήματα και τα μειονεκτήματά τους.

Στην εργασία αυτή παρουσιάζεται και αναλύεται η τελική ιδέα ενός συστήματος που ασχολείται τόσο με την ανάλυση της κυκλοφορίας, όσο και με την παρουσίαση της κίνησης πάνω σ' ένα χάρτη. Με σκοπό την πραγματοποίηση του συστήματος αυτού, έχει υλοποιηθεί μία εφαρμογή συλλογής διάφορων δεδομένων που απαιτούνται για την εύρεση οδικής κίνησης. Με τη βοήθεια της εφαρμογής αυτής, έχουν συλλεχθεί δεδομένα για διάφορες διαδρομές στην Λευκωσία. Έχει υλοποιηθεί ακόμη, μία ιστοσελίδα που εμφανίζει διαδραστικά γεωγραφικά δεδομένα πάνω σ' ένα χάρτη, με σκοπό να χρησιμοποιηθεί στο τελικό σύστημα. Τέλος, στην πλατφόρμα έχει ενσωματωθεί ένα σύστημα που αποσκοπεί στη λύση του προβλήματος αντιστοίχισης τροχιών πάνω σε χάρτη.

Επιπλέον, κατά τη διάρκεια της διπλωματικής έχει γίνει μία πειραματική αποτίμηση που αποτελείται από δύο πειράματα. Το πρώτο πείραμα αφορά τον χρόνο απόκρισης της ιστοσελίδας που έχει υλοποιηθεί, και συγκεκριμένα τον χρόνο που χρειάζεται να φορτώσει τα δεδομένα στο χάρτη. Ενώ, το δεύτερο πείραμα αποσκοπεί στην σύγκριση των χρόνων εκτέλεσης ενός προγράμματος που έχει γραφτεί σε διάφορες γλώσσες - τεχνολογίες.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Υποκίνηση Εργασίας	1
	1.2 Περιγραφή Συστήματος	2
	1.3 Συνεισφορές	3
	1.4 Περίγραμμα Εργασίας	4
Κεφάλαιο 2	Γνωσιολογικό Υπόβαθρο.....	6
	2.1 Αρχιτεκτονική και Συστήματα Τηλεπικοινωνιακών Παρόχων	6
	2.1.1 Κεραίες Κυψέλης	9
	2.1.2 Macro Cells	12
	2.1.3 Sector Cells	13
	2.2 Δεδομένα Τηλεπικοινωνιακών Παρόχων	14
	2.2.1 Call Detail Records (CDR)	14
	2.2.2 Network Measurement Report (NMR)	14
Κεφάλαιο 3	Σχετική Βιβλιογραφία.....	16
	3.1 Έρευνα σε Μεγάλα Τηλεπικοινωνιακά Δεδομένα	16
	3.1.1 Ανάλυση και Ανίχνευση σε Πραγματικό χρόνο	17
	3.1.2 Εμπειρία, Συμπεριφορά και Διατήρηση Αναλύσεων	19
	3.1.3 Θέματα Ιδιωτικότητας (Privacy)	20
	3.1.4 Αποθήκευση (Storage)	21
	3.2 Έρευνα σε Δεδομένα Κίνησης	22
	3.2.1 Πρώτη Γενιά Χαρτογράφησης Κυκλοφορίας	23
	3.2.2 Δεύτερη Γενιά Χαρτογράφησης Κυκλοφορίας	24
	3.2.3 Γενιά Δύομιση (2.5) Χαρτογράφησης Κυκλοφορίας	26
	3.2.4 Τρίτη Γενιά Χαρτογράφησης Κυκλοφορίας	32
	3.2.5 Τεχνολογίες Βασισμένες σε Οχήματα	36
Κεφάλαιο 4	Αρχιτεκτονική TrafficWare.....	38
	4.1 Μοντέλο Συστήματος	38

4.1.1 Σύστημα με Χρήση Δεδομένων Δικτύου	39
4.1.2 Σύστημα με Χρήση Πληθοπορισμού	39
4.1.3 Κύρια Επεξεργασία του Συστήματος	41
4.2 Θέματα Ιδιωτικότητας (Privacy)	45
4.3 Προβλήματα Ακρίβειας	46
Κεφάλαιο 5 Υποσύστημα Προβολής και Οπτικοποίησης.....	48
5.1 Τεχνικό Υπόβαθρο	48
5.1.1 Apache Spark	49
5.1.2 GIS	50
5.1.3 PostgreSQL	52
5.1.4 PostGIS	52
5.2 Δεδομένα Εισόδου	53
5.3 Πλατφόρμα TrafficWare	68
Κεφάλαιο 6 Υποσύστημα Αντιστοίχισης.....	75
6.1 Εφαρμογή για Συλλογή Τροχιών του TrafficWare	74
6.2 Αντιστοίχιση Τροχιών σε Χάρτη	79
Κεφάλαιο 7 Πειραματική Αποτίμηση.....	82
7.1 Πειραματική Μεθοδολογία	82
7.2 Απόδοση Αναλύσεων	88
7.2.1 Ανάλυση Χρόνου Απόκρισης της TrafficWare	
Ιστοσελίδας	88
7.2.2 Ανάλυση Χρόνου Επεξεργασίας Δεδομένων	92
Κεφάλαιο 8 Συμπεράσματα και Μελλοντική Εργασία	94
8.1 Συμπεράσματα	94
8.2 Μελλοντική Εργασία	96
Βιβλιογραφία.....	98

Π α ρ ά ρ τ η μ α Α..... Α-1

Π α ρ ά ρ τ η μ α Β..... Β-1

Π α ρ ά ρ τ η μ α Γ..... Γ-1

Κεφάλαιο 1

Εισαγωγή

1.1 Υποκίνηση Εργασίας	1
1.2 Περιγραφή Συστήματος	2
1.3 Συνεισφορές	3
1.4 Περίγραμμα Εργασίας	4

1.1 Υποκίνηση Εργασίας

Η κίνηση στους δρόμους αποτελεί ένα καθημερινό πρόβλημα και κυρίως στις αστικές περιοχές. Με την αύξηση της χρήσης οχημάτων και γενικότερα των οδικών δικτύων, τόσο περισσότερο μεγεθύνεται το συγκεκριμένο πρόβλημα. Η οδική κίνηση μπορεί να αποτελέσει ένα μεγάλο πρόβλημα στη ζωή πολλών ανθρώπων, αφού οδηγεί στη σπατάλη χρόνου των οδηγών και των επιβατών, ενώ ο χρόνος αυτός θα μπορούσε να ήταν πιο παραγωγικός. Οδηγεί στην καθυστερημένη άφιξη για απασχόληση, συνεδριάσεις, και εκπαίδευση. Λόγω της κίνησης, δημιουργείται η αδυναμία πρόβλεψης του χρόνου ταξιδιού με ακρίβεια, αναγκάζοντας τους οδηγούς να διαθέσουν περισσότερο χρόνο για να ταξιδέψουν για να είναι σίγουροι ότι θα φτάσουν στον προορισμό τους έγκαιρα. Η κίνηση, ακόμη, προκαλεί σπατάλη καυσίμου και κατ' επέκταση αύξηση της ατμοσφαιρικής ρύπανσης. Στην περίπτωση έκτακτων αναγκών, τα ασθενοφόρα παρεμποδίζονται από το να φτάσουν στον προορισμό τους όσο πιο γρήγορα μπορούν. Τέλος, η οδική κίνηση, αυξάνει την πιθανότητα συγκρούσεων, λόγω της στενής απόστασης μεταξύ των οχημάτων, όπως και λόγω της αστάθειας του ορίου ταχύτητας που χρησιμοποιείται.

Κάθε χρόνο, όπως αναφέρονται σε διάφορες πηγές [32], περίπου 1,25 εκατομμύρια άνθρωποι πεθαίνουν σε τροχαία ατυχήματα, δηλαδή κατά μέσο όρο, συμβαίνουν 3,287 θάνατοι καθημερινά. Επιπρόσθετα, 20-50 εκατομμύρια άτομα τραυματίζονται ετησίως λόγω αυτοκινητικών ατυχημάτων.

Η κατανόηση, η ανάλυση και η διαχείριση οδικής κυκλοφορίας μπορούν να ελαχιστοποιήσουν τον αριθμό των τροχαίων ατυχημάτων, να συμβάλουν στη βελτιστοποίηση της κατανάλωσης καυσίμων και ενέργειας, όπως και στην αποφυγή απροσδόκητων καθυστερήσεων. Ακόμη, μπορούν να βοηθήσουν στην κατανόηση της αιτίας που δημιουργεί την οδική κίνηση.

Στο μυαλό όλων μας, μία τηλεπικοινωνιακή εταιρεία έχει ως σκοπό μονάχα την παροχή τηλεπικοινωνιακών υπηρεσιών, όπως η κινητή τηλεφωνία. Ωστόσο, η υποδομή των υπηρεσιών αυτών, είναι φτιαγμένη με τέτοιο τρόπο που θα μπορούσαν να τις εκμεταλλευτούν και για την παροχή άλλων διαφορετικών λειτουργιών. Συγκεκριμένα, οι υποδομές αυτές είναι τοποθετημένες σε πυκνοκατοικημένους χώρους και αγροτικές περιοχές, παρέχοντάς τους την ευκαιρία συλλογής τεράστιου αριθμού δεδομένων σχετικά με την κίνηση. Με τα δεδομένα αυτά, θα μπορούν να παρέχουν πολύτιμες γνώσεις για την καλύτερη διαχείριση της οδικής κυκλοφορίας.

1.2 Περιγραφή Συστήματος

Στα πλαίσια της παρούσας διπλωματικής εργασίας θα παρουσιαστεί ένα πρωτότυπο του συστήματος που αποσκοπεί στην εύρεση της οδικής κίνησης. Εμπερικλείει μία αρχιτεκτονική που ασχολείται τόσο με την ανάλυση της κυκλοφορίας, όσο και με την παρουσίαση της κίνησης για οποιαδήποτε χρονική στιγμή πάνω σ' ένα χάρτη. Το σύστημα γενικότερα βασίζεται σε τρία βασικά κριτήρια. Τα κριτήρια αυτά είναι η επίλυση του προβλήματος διατηρώντας την ιδιωτικότητα της θέσης των χρηστών, χρησιμοποιώντας την υπάρχουσα υποδομή (π.χ., κεραίες) και υλοποιώντας το σύστημα με ελάχιστο κόστος. Επιπρόσθετα, γίνεται αναφορά σχετικά με την παρουσίαση και διαχείριση μεγάλων δεδομένων.

Κατά τη διάρκεια της διπλωματικής, έχουν μελετηθεί διάφοροι τρόποι για την βέλτιστη υλοποίηση του συστήματος. Στο παρόν κείμενο, αναλύονται οι τρόποι αυτοί και επεξηγούνται ξεχωριστά τα πλεονεκτήματα και τα μειονεκτήματα για κάθε λύση.

1.3 Συνεισφορές

Στην παρούσα έρευνα, μελετήθηκαν οι τρόποι επίλυσης ενός καινοτόμου προβλήματος που αποσκοπεί στην αποτίμηση κίνησης. Συγκεκριμένα, μελετήθηκε ιδιαίτερα η ανάλυση και η εύρεση της οδικής κυκλοφορίας, μέσω της μελέτης παρόμοιων ερευνών και τρόπων υλοποίησης με ίδιους ή παρόμοιους σκοπούς. Ακόμη, κρίθηκε αναγκαίο να διερευνηθούν τα πλεονεκτήματα, όπως και τα μειονεκτήματα για κάθε πιθανή λύση. Επιπρόσθετα, μελετήθηκε ένα ιδιαίτερα σημαντικό πρόβλημα, το πρόβλημα αντιστοίχισης τροχιών σε χάρτη (map - matching), μαζί με διάφορες λύσεις - υλοποιήσεις του προβλήματος.

Επιπλέον, έχουν συλλεχθεί διάφορα σύνολα δεδομένων (datasets) που μπορούν να βοηθήσουν στην εύρεση της οδικής κίνησης. Με σκοπό την απεικόνιση των δεδομένων αυτών, έχει υλοποιηθεί ένα πρωτότυπο σε αρχικό στάδιο, το οποίο παρουσιάζει διαδραστικά τα δεδομένα αυτά. Το πρωτότυπο αυτό, παρουσιάζει διάφορα δεδομένα σχετικά με θέσεις κεραιών, αλλά και με μετρήσεις χρηστών. Όσον αφορά τις μετρήσεις χρηστών, η ιστοσελίδα παρουσιάζει τα σημεία αυτά ως πολύγωνα, περικυκλώνοντας τις τοποθεσίες των σημείων που αναφέρονται στην ίδια κεραία. Για τις μετρήσεις χρηστών, έχει υλοποιηθεί μία εφαρμογή που συλλέγει διάφορα δεδομένα καθώς οι χρήστες οδηγούν στους δρόμους, με σκοπό την καταγραφή των κεραιών που εξυπηρετούν συγκεκριμένα σημεία / τροχιές.

1.4 Περίγραμμα Εργασίας

Στο δεύτερο κεφάλαιο, επεξηγούνται διάφορες έννοιες και ορισμοί που αφορούν τα συστήματα των τηλεπικοινωνιακών παρόχων, αλλά και τα τηλεπικοινωνιακά δεδομένα

που παράγουν και επεξεργάζονται. Ακόμη, υπάρχει μία περιγραφή σχετικά με την βασική λειτουργία των κεραιών κυψέλης.

Έπειτα, στο τρίτο κεφάλαιο γίνεται μία περιγραφή της σχετικής βιβλιογραφίας. Συγκεκριμένα, περιγράφονται οι σχετικές έρευνες που μελετούν μεγάλα τηλεπικοινωνιακά δεδομένα, αλλά και οι έρευνες που βασίζονται σε δεδομένα κίνησης. Ακόμη, αναφέρονται τα αποτελέσματα τους και τα διάφορα σημαντικά ευρήματα των αποτελεσμάτων τους.

Στο τέταρτο κεφάλαιο, περιγράφεται αναλυτικά η αρχιτεκτονική του συστήματος. Συγκεκριμένα, γίνεται μία εκτενής περιγραφή των δύο εναλλακτικών ιδεών του συστήματος, μαζί με τα θετικά, αρνητικά αλλά και όσα χρειάζονται για την υλοποίηση τους. Επιπρόσθετα, υπάρχει μία βασική εξήγηση σχετικά με τα θέματα ιδιωτικότητας των χρηστών που λήφθηκαν υπόψη κατά τη διάρκεια σχεδίασης της αρχιτεκτονικής. Τέλος, αναφέρονται διάφορα προβλήματα ακρίβειας που μπορούν να προκύψουν.

Στο πέμπτο κεφάλαιο, γίνεται μία περιγραφή του υποσυστήματος προβολής και οπτικοποίησης. Αρχικά, γίνεται μία σύντομη επεξήγηση του βασικού τεχνικού υποβάθρου που απαιτείται για την κατανόηση διαφόρων τεχνολογιών που χρησιμοποιούνται ή θα μπορούσαν να χρησιμοποιηθούν από το σύστημα. Έπειτα, αναλύεται η πλατφόρμα του συστήματος - TrafficWare, που υλοποιήθηκε κατά τη διάρκεια της διπλωματικής με σκοπό την απεικόνιση γεωγραφικών δεδομένων. Στο τέλος του κεφαλαίου αυτού, ακολουθεί μία γενική περιγραφή όλων των δεδομένων που δέχεται η ιστοσελίδα σαν είσοδο. Συγκεκριμένα τα δεδομένα αυτά, έχουν μελετηθεί με σκοπό την εύρεση των θέσεων των κεραιών ή των μετρικών χρηστών.

Στο έκτο κεφάλαιο, αναλύεται η εφαρμογή που δημιουργήθηκε στην παρούσα διπλωματική με σκοπό την συλλογή τροχιών. Συγκεκριμένα, η εφαρμογή συλλέγει διάφορα τηλεπικοινωνιακά δεδομένα καθώς οι χρήστες οδηγούν. Ακόμη, στο κεφάλαιο αυτό, γίνεται μια εξήγηση του προβλήματος αντιστοίχισης τροχιών σε χάρτη (map - matching) και περιγράφεται το σύστημα που χρησιμοποιήθηκε για την επίλυση του προβλήματος αυτού.

Στο έβδομο κεφάλαιο, περιγράφονται τα πειράματα και η ανάλυση τους που έγιναν κατά τη διάρκεια της διπλωματικής. Η πειραματική αποτίμηση χωρίζεται σε δύο πειράματα, το πρώτο αφορά την απόδοση της πλατφόρμας που υλοποιήθηκε και τον χρόνο που χρειάζεται για να φορτώσει δεδομένα. Στην ανάλυση του πειράματος αυτού, η ιστοσελίδα συγκρίνεται με άλλες ιστοσελίδες που αποσκοπούν στην απεικόνιση γεωγραφικών δεδομένων. Το δεύτερο πείραμα, αφορά την σύγκριση των χρόνων εκτέλεσης ενός προγράμματος που γράφτηκε σε διάφορες τεχνολογίες - γλώσσες (Spark Java, Java, NodeJs και PHP).

Τέλος, στο όγδοο και τελευταίο κεφάλαιο, περιγράφονται τα συμπεράσματα της παρούσας διπλωματικής και προτείνονται μελλοντικές επεκτάσεις ή βελτιώσεις που μπορούν να γίνουν.

Κεφάλαιο 2

Γνωσιολογικό Υπόβαθρο

2.1 Αρχιτεκτονική και Συστήματα Τηλεπικοινωνιακών Παρόχων	6
2.1.1 Κεραίες Κυψέλης	9
2.1.2 Macro Cells	12
2.1.3 Sector Cells	13
2.2 Δεδομένα Τηλεπικοινωνιακών Παρόχων	14
2.2.1 Call Detail Records (CDR)	14
2.2.2 Network Measurement Report (NMR)	14

Στην ενότητα αυτή, θα περιγραφεί και θα εξηγηθεί το γνωσιολογικό υπόβαθρο που αποκτήθηκε κατά τη διάρκεια της παρούσας διπλωματικής. Το υπόβαθρο αυτό, σχετίζεται με την αρχιτεκτονική, τα συστήματα και τα δεδομένα που χρησιμοποιούν οι τηλεπικοινωνιακοί πάροχοι.

2.1 Αρχιτεκτονική και Συστήματα Τηλεπικοινωνιακών Παρόχων

Πίσω από την παροχή τηλεπικοινωνιακών δικτύων, όπως και σε κάθε άλλο σύστημα, κρύβεται μια αρχιτεκτονική συστήματος. Στην προκειμένη περίπτωση, η αρχιτεκτονική αυτή απεικονίζεται στο Σχήμα 2.1. Στο διάγραμμα αυτό, απεικονίζονται τα κύρια στοιχεία και το σχέδιο οργάνωσης μιας αρχιτεκτονικής δικτύου κινητής τηλεφωνίας. Καταρχάς, στο Σχήμα φαίνεται το κύριο στοιχείο που χρειάζεται για την αρχιτεκτονική αυτή, όπου είναι ο σταθμός βάσης (base station) πολλαπλών προδιαγραφών που υποστηρίζει 2G, 3G, 4G, LTE, και πολύ σύντομα τα 5G δίκτυα. Κάθε συσκευή, ανεξαρτήτως του δικτύου που χρησιμοποιεί και υποστηρίζει, δημιουργεί μία σύνδεση μέσω του ασύρματου πύργου (cell tower) με σημείο πρόσβασης (access point) τον

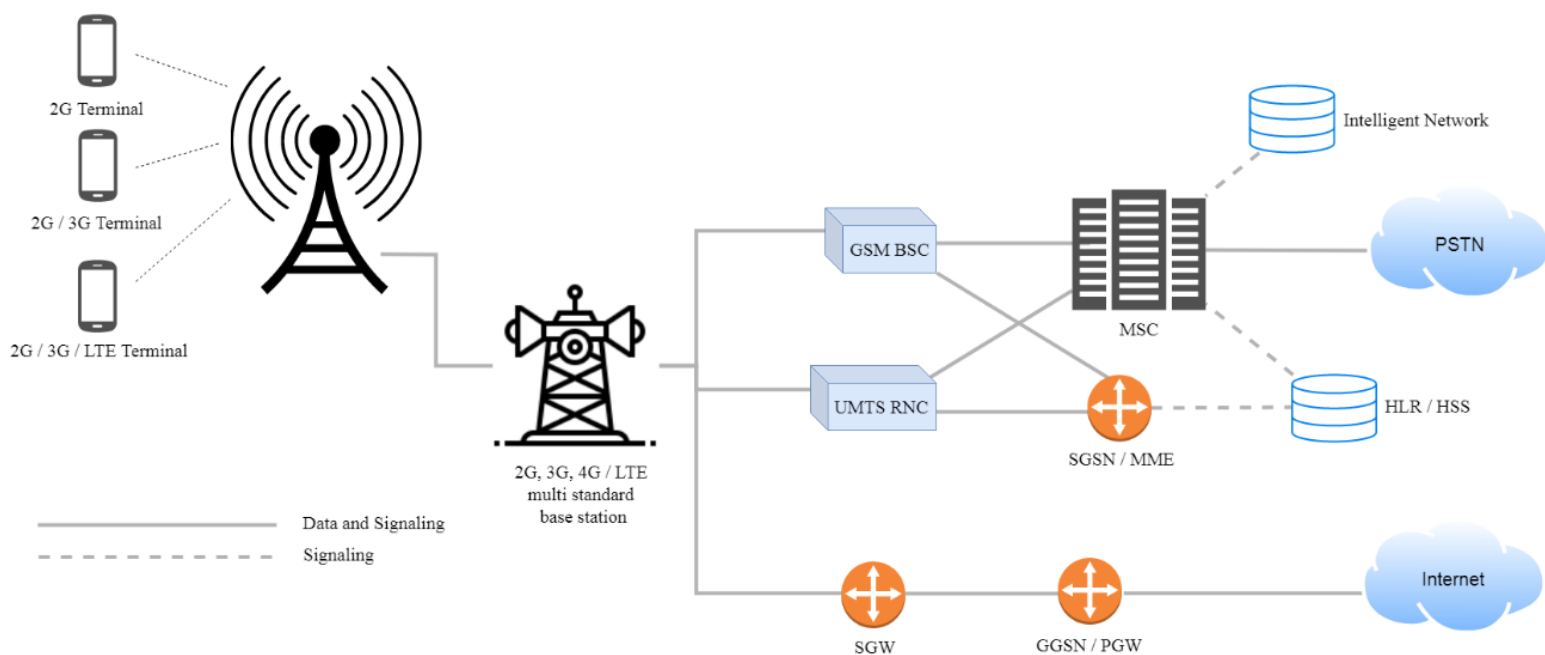
σταθμό βάσης. Στην περίπτωση που η σύνδεση είναι επιτυχής, τότε ο σταθμός βάσης ανταποκρίνεται στην κλήση της κινητής συσκευής.

Ακόμη, στην αρχιτεκτονική αυτή εμπλέκονται συστατικά τα οποία είναι υπεύθυνα για την καταγραφή διάφορων δεδομένων που αφορούν το δίκτυο. Τα δεδομένα αυτά εμπερικλείουν δεδομένα τιμολόγησης, λογιστικής, κτλ.

Το Σχήμα 2.1 μπορεί να χωριστεί σε δύο μέρη, στο ράδιο μέρος (radio) και στον πυρήνα (core). Το ράδιο μέρος, αποτελείται από τις συσκευές, τις κεραίες και τον σταθμό βάσης. Ο πυρήνας αποτελείται από το δεξιό μέρος του Σχήματος, δηλαδή τα εξής στοιχεία: GSM BSC (Base Station Controller), UMTS RNC (Radio Network Controller), MSC (Mobile Switching Station), SGW (Cisco Serving Gateway), GGSN (Gateway GPRS Support Node) / PGW (Cisco Packet Data Network Gateway), SGSN (Serving GPRS Support Node) / MME (Mobile Management Entity), Intelligent Network, HLR (Home Location Register) / HSS (Home Subscriber Server), PSTN (Public Switched Telephone Network) και το Internet.

Το GSM BSC [34] ελέγχει διάφορους σταθμούς βάσης που λειτουργούν ως δέκτες κλήσεων (Base Transceiver Stations). Συγκεκριμένα, είναι υπεύθυνο για την κατανομή ραδιοφωνικών πόρων σε μια κλήση που έγινε από κάποιο κινητό τηλέφωνο, αλλά και για τις μεταβιβάσεις συνδέσεων (handovers) που γίνονται μεταξύ των σταθμών βάσης. Παράλληλα, κάποιες μεταβιβάσεις συνδέσεων (handovers) ελέγχονται από το MSC (Mobile Switching Centre) [35], που λειτουργεί για 2G δίκτυα. Το MSC ακόμη, είναι υπεύθυνο για τον έλεγχο των στοιχείων του υποσυστήματος μεταγωγής δικτύου. Εναλλακτικά ή προσαρμοστικά, το MSC μπορεί επίσης να χρησιμοποιηθεί για δίκτυα GSM, εάν ο κατασκευαστής έχει υλοποιήσει υποστήριξη στο MSC για δίκτυα GSM. Το RNC [30] είναι ένα στοιχείο καθοδήγησης για τα UMTS (Universal Mobile Telecommunications System) δίκτυα και είναι υπεύθυνο για τον έλεγχο των nodeB (κεραίες για UMTS δίκτυα) που συνδέονται με αυτό. Το SGSN [36] διαχειρίζεται όλα τα δεδομένα μεταγωγής πακέτων (packet switched data) εντός του δικτύου, και το GGSN είναι υπεύθυνο για τη διασύνδεση του δικτύου GPRS με εξωτερικά δίκτυα μεταγωγής πακέτου (packet switched networks). Το GGSN και το SGSN αποτελούν τα κύρια συστατικά για το GPRS (General Packet Radio Service) δίκτυο. Απ' την άλλη, το

PGW [33] χρησιμοποιείται για τα 4G δίκτυα και λειτουργεί ως διασύνδεση μεταξύ του δικτύου LTE και άλλων δικτύων πακέτων δεδομένων (packet data networks). Το MME είναι ο βασικός κόμβος ελέγχου για τα LTE δίκτυα, είναι υπεύθυνο για τη διαδικασία τηλεειδοποίησης (paging) και ετικετοποίησης (tagging) σε περιπτώσεις που ο χρήστης βρίσκεται σε αναμονή, αλλά και σε περιπτώσεις αναμεταδόσεων (retransmissions) των κλήσεων. Το SGW χρησιμοποιείται για 4G δίκτυα, προωθεί και δρομολογεί τα πακέτα προς και από το eNodeB (κεραίες που υποστηρίζουν LTE δίκτυα) και την πύλη δικτύου πακέτων δεδομένων (PGW). Το HLR [31] είναι η κύρια βάση δεδομένων για την αποθήκευση πληροφοριών σχετικά με τους συνδρομητές των κινητών δικτύων και αποτελεί αναπόσπαστο τμήμα των CDMA (Code Division Multiple Access), TDMA (Time Division Multiple Access), και GSM (Global System for Mobile Communications) δικτύων. Το HSS είναι μία βελτιωμένη εκδοχή του HLR. Τέλος, το PSTN είναι η παγκόσμια συλλογή διασυνδεδεμένων φωνητικών δημόσιων τηλεφωνικών δικτύων, τόσο εμπορικών, όσο και κρατικών επιχειρήσεων.



Σχήμα 2.1 Αρχιτεκτονική Τηλεπικοινωνιακού Δικτύου [63]

2.1.1 Κεραίες Κυψέλης

Οι συνδέσεις των κινητών συσκευών επιτυγχάνονται με τη βοήθεια των κεραιών κυψέλης. Οι κεραίες αυτές, βρίσκονται σε διάφορα σημεία των πόλεων - περιοχών, με σκοπό την καλύτερη κάλυψη δικτύου σε οποιοδήποτε σημείο βρίσκεται ο κάθε χρήστης. Στο Σχήμα 2.2 φαίνονται τα διάφορα σημεία τοποθέτησης κεραιών, π.χ., στην οροφή κτηρίων, σε υπόγειους χώρους στάθμευσης (parking), σε λόφους, κτλ.

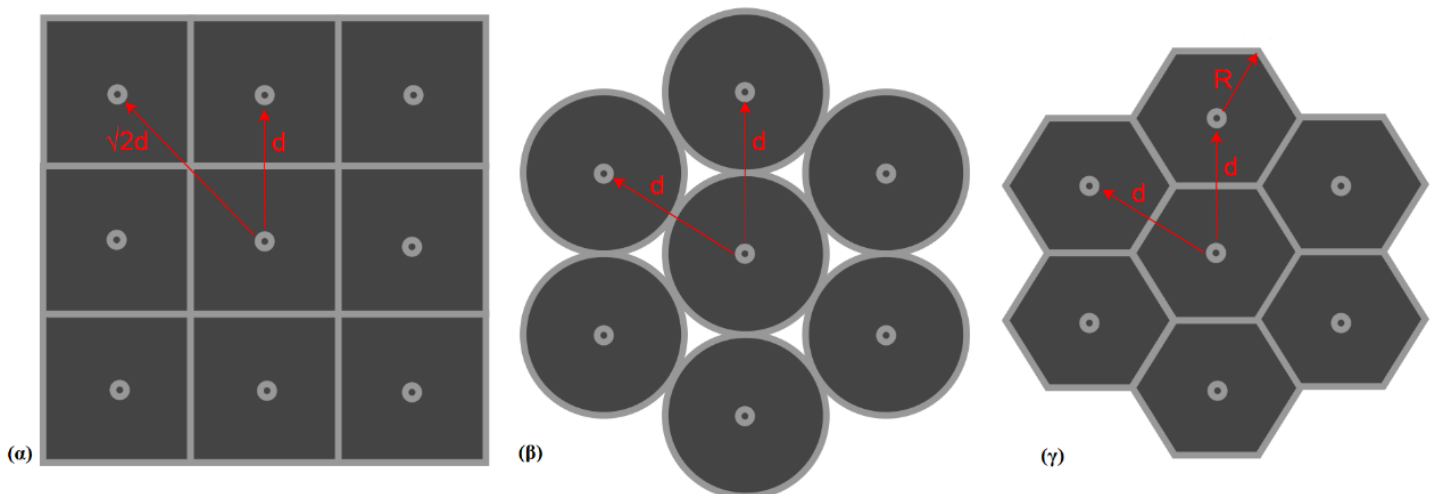


Σχήμα 2.2 Τηλεπικοινωνιακό Δίκτυο [8]

Στο σημείο αυτό, παράγονται διάφορα δεδομένα, όπως η ισχύς σήματος (signal strength), η αποσύνδεση κλήσεων (call drops), οι μετρήσεις εύρους ζώνης (bandwidth measurements), κτλ. Η συλλογή των δεδομένων αυτών, αποσκοπεί στη βελτίωση του δικτύου, όπως επίσης και στο να λειτουργήσει ως τρόπος χρέωσης των πελατών. Ταυτόχρονα, συλλέγονται δεδομένα που αφορούν τις μετρήσεις και τον φόρτο εργασίας

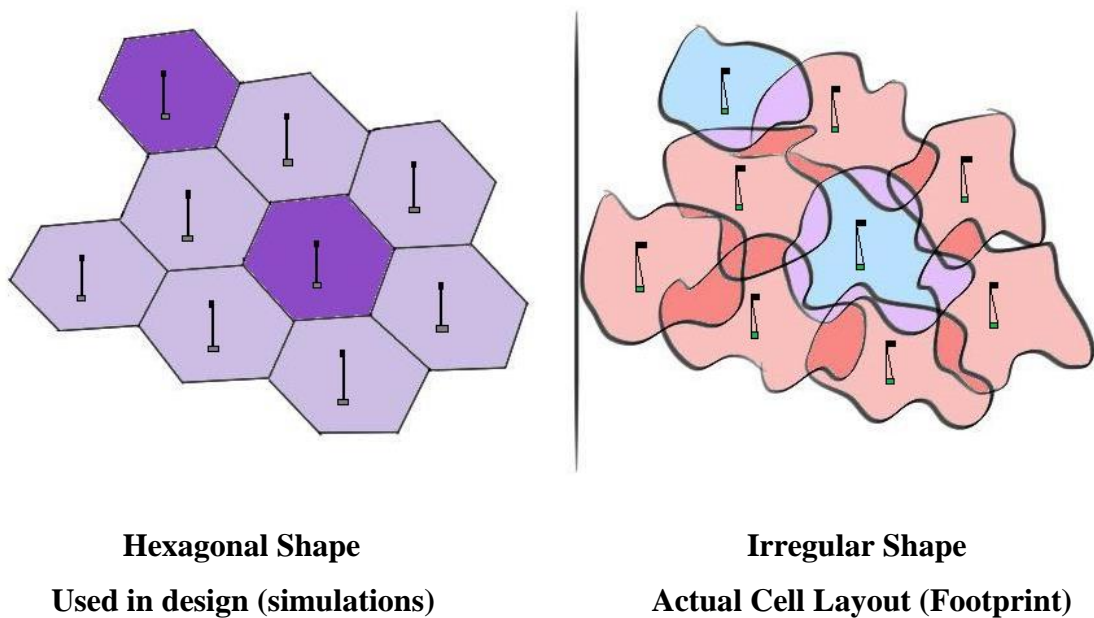
της κάθε κεραίας. Όλα τα δεδομένα συνολικά, μπορεί να φτάσουν σε μέγεθος έως και τα 5TB καθημερινά για 10 εκατομμύρια πελάτες.

Οι κεραίες μεταξύ τους απέχουν διαφορετική απόσταση, η οποία εξαρτάται από διάφορους παράγοντες, π.χ., ο μέσος αριθμός χρηστών που αναμένεται να είναι συνδεδεμένοι σε μια περιοχή. Οι αποστάσεις αυτές μπορούν να απεικονιστούν με πολλούς τρόπους, όπως φαίνεται και στο Σχήμα 2.3. Η απεικόνιση των αποστάσεων - περιμέτρων των κεραιών χρειάζονται για την καλύτερη ανάλυση και αξιολόγηση του δικτύου. Στο Σχήμα 2.3 (α) παρουσιάζεται η απεικόνιση των περιμέτρων που καλύπτουν οι κεραίες με τετράγωνα. Με τον τρόπο αυτό, όλος ο χώρος καλύπτεται ομοιόμορφα αλλά οι κεραίες δεν ισαπέχουν μεταξύ τους. Ο δεύτερος τρόπος απεικόνισης των αποστάσεων που εξυπηρετούν οι κεραίες, είναι με τη σχεδίαση κύκλων, όπου έτσι οι αντένες ισαπέχουν μεταξύ τους. Παρ' όλα αυτά, η απεικόνιση των αποστάσεων χρησιμοποιώντας κύκλους οδηγεί στην ύπαρξη κενών μεταξύ των κεραιών, δηλαδή να υπάρχουν σημεία που δεν εξυπηρετούνται από καμία κεραία. Έτσι, ο τρίτος τρόπος, όπου είναι ο ιδανικότερος, είναι η απεικόνιση των αποστάσεων με εξάγωνα έτσι ώστε, οι κεραίες να ισαπέχουν μεταξύ τους και ταυτόχρονα να μην υπάρχουν σημεία που δεν εξυπηρετούνται από καμία κεραία.



Σχήμα 2.3 Περίμετρος Κεραιών [62]

Τα συστήματα σχεδιάζονται έτσι ώστε ιδανικά η κάθε κεραία να καλύπτει μία συγκεκριμένη περιοχή. Παρ' όλα αυτά, λόγω των φυσικών συνθηκών, οι κεραίες δεν καλύπτουν τις περιοχές όπως έχουν σχεδιαστεί. Αυτό ακριβώς το γεγονός απεικονίζεται και στο Σχήμα 2.4. Κάποιες περιοχές μπορεί να μην καλύπτονται από καμία κεραία, ή μπορεί να καλύπτονται από μία έως και περισσότερες κεραίες ταυτόχρονα. Σκοπός των τηλεπικοινωνιακών παρόχων είναι η κάλυψη όλων των περιοχών, έστω από μία κεραία. Γι' αυτό τον λόγο, οι συσκευές δεν συνδέονται απαραίτητα με την κοντινότερη (physically) κεραία, αλλά με την κεραία που παρέχει το ισχυρότερο σήμα.

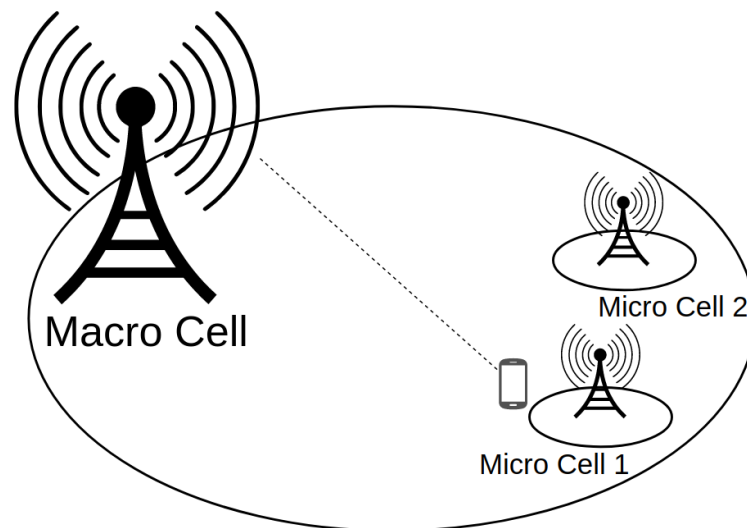


Σχήμα 2.4 Αρχιτεκτονική Τηλεπικοινωνιακού Δικτύου [62]

Καθώς οι χρήστες κινούνται, η συσκευή συνδέεται με την κεραία που παρέχει το ισχυρότερο σήμα. Γι' αυτό, συχνά χρειάζεται η συσκευή να συνδεθεί σε διαφορετική κεραία (handover). Η ενέργεια αυτή γίνεται απίστευτα γρήγορα, χωρίς να αντιληφθούν την αλλαγή αυτή οι χρήστες. Όταν η συσκευή συνδεθεί σε μια νέα κεραία, τότε όλα τα πακέτα που λαμβάνει ή στέλνει η συσκευή, μεταφέρονται στην νέα κεραία, έτσι ώστε να μην διακοπεί η υπηρεσία.

2.1.2 Macro Cells

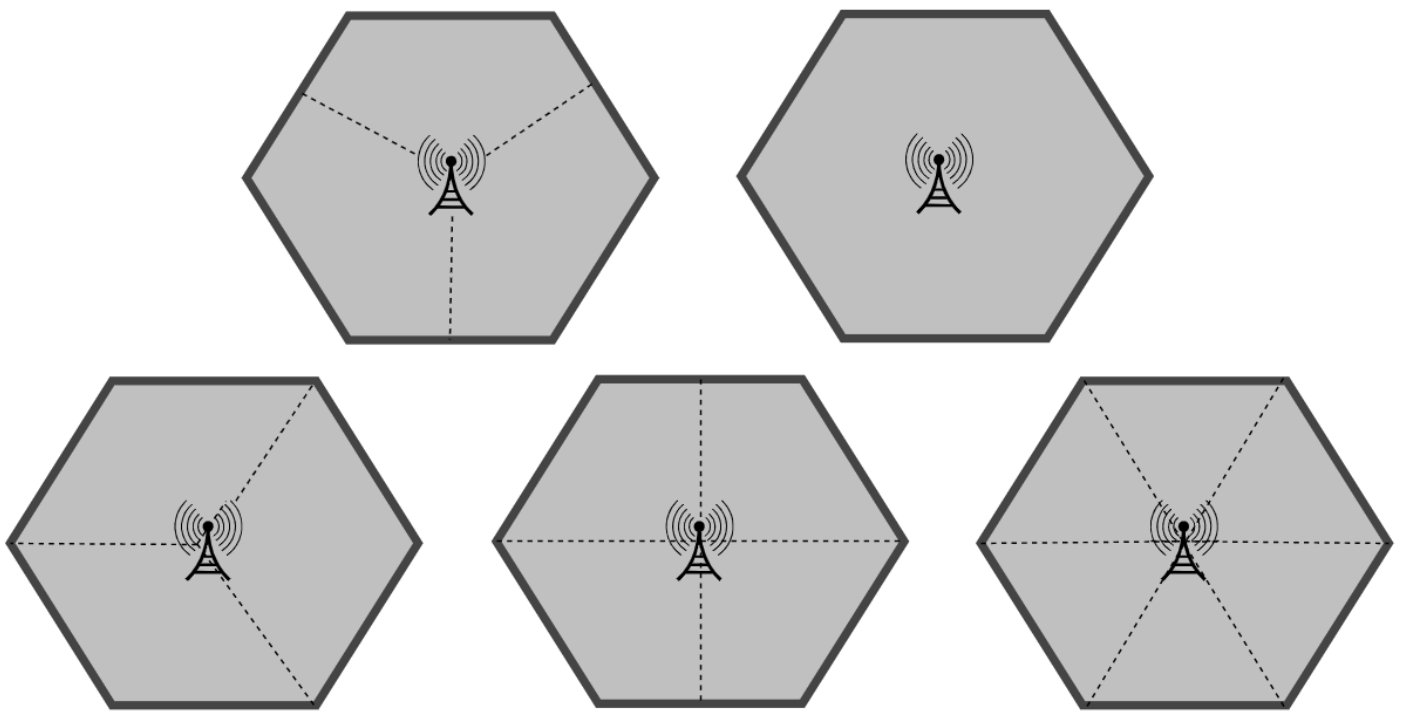
Σε καταστάσεις που οι χρήστες αναγκάζονται να συνδέονται συνεχώς με καινούργιες κεραίες (handover), π.χ., όταν οι χρήστες οδηγούν σε αυτοκινητόδρομους (highways), έχουν εγκατασταθεί τα Macro Cells με σκοπό την κάλυψη των χρηστών αυτών. Τα Macro Cells έχουν δημιουργηθεί έτσι ώστε να μη χρειάζονται οι συσκευές να συνδέονται συνέχεια σε καινούργια κεραία (handover). Οι χρήστες συνδέονται από Micro Cells σε Macro Cells όταν κινούνται με πολύ μεγάλη ταχύτητα, ενώ όταν οι χρήστες κινούνται με αργή ταχύτητα ή δεν κινούνται καθόλου, τότε μεταφέρονται πίσω στα Micro Cells. Οι Macro κεραίες μπορούν να καλύψουν πολύ μεγαλύτερες αποστάσεις από μια κανονική κεραία. Συγκεκριμένα, μια κανονική κεραία μπορεί να παρέχει σήμα σε περιμέτρους απόστασης 200 - 2000m, σε αντίθεση με τις macro κεραίες που μπορούν να παρέχουν σήμα σε περιμέτρους απόστασης 1 - 30km. Στο Σχήμα 2.5, απεικονίζεται και η διαφορά απόστασης που μπορεί η macro κεραία να παρέχει σήμα σε σχέση με τις κανονικές κεραίες (microcells) που υπάρχουν εντός πόλεων.



Σχήμα 2.5 Απεικόνιση Περιμέτρων Παροχής Σήματος Μεταξύ Macro και Micro Κεραίες [62]

2.1.3 Sector Cells

Κάθε κεραία μπορεί να παρέχει σήμα σε συγκεκριμένες περιοχές - περιμέτρους. Για την καλύτερη και πλήρη παροχή σήματος, κάποιες κεραίες χωρίζονται σε τομείς (sectors). Όπως φαίνεται και στο Σχήμα 2.6, κάποιες κεραίες μπορεί να χωρίζονται σε 3, 4, ή και 6 τομείς. Κάθε τομέας μπορεί να θεωρηθεί ως διαφορετική κεραία, αφού έχει το δικό του cellid και τα δικά του κανάλια συχνότητας. Ουσιαστικά, μειώνει την παρεμβολή μεταξύ κεραιών που εκπέμπουν παρόμοια συχνότητα. Αυτό, επιτρέπει την επαναχρησιμοποίηση πιο πυκνής συχνότητας. Η τοποθέτηση των κεραιών αυτών είναι λιγότερο δαπανηρή, την δημιουργία σταθμών βάσης σε νέες θέσης.



Σχήμα 2.6 Αρχιτεκτονική Τηλεπικοινωνιακού Δικτύου [62]

2.2 Δεδομένα Τηλεπικοινωνιακών Παρόχων

Όπως και κάθε σύστημα, τα συστήματα των τηλεπικοινωνιακών παρόχων εξάγουν και επεξεργάζονται διάφορα δεδομένα. Στη φάση αυτή, θα αναφερθούν κάποια από τα δεδομένα αυτά.

2.2.1 Call Detail Record (CDR)

Με κάθε δραστηριότητα που εκτελεί ο χρήστης, π.χ., κλήση, μηνύματα, σύνδεση στο διαδίκτυο, κτλ, δημιουργείται μια εγγραφή (record) στο σύστημα των τηλεπικοινωνιακών παρόχων. Η εγγραφή αυτή ονομάζεται CDR [29]. Τα CDR περιέχουν τον αριθμό τηλεφώνου του συνδρομητή που προκάλεσε την κλήση, τον αριθμό τηλεφώνου που λαμβάνει την κλήση, την ώρα έναρξης της κλήσης, την διάρκεια κλήσης, τον αριθμό τηλεφώνου που θα χρεωθεί για την κλήση, την αναγνώριση του τηλεφωνικού κέντρου ή του εξοπλισμού που συντάσσει το αρχείο και έναν μοναδικό αριθμό ακολουθίας που αναγνωρίζει την εγγραφή (record). Ακόμη περιέχει επιπρόσθετα ψηφία στον καλούμενο αριθμό που χρησιμοποιείται για τη δρομολόγηση ή τη χρέωση της κλήσης, τα αποτελέσματα της κλήσης (π.χ., αν έγινε επιτυχώς η κλήση), τη διαδρομή με την οποία εισήλθε η κλήση, τη διαδρομή με την οποία εξήλθε η κλήση, τον τύπο κλήσης (π.χ., φωνητική, sms, κτλ) και τέλος, οποιαδήποτε κατάσταση σφάλματος που προέκυψε.

Έπειτα, οι παροχείς συγκεντρώνουν όλες τις εγγραφές και τις χρησιμοποιούν για τον υπολογισμό χρέωσης του κάθε χρήστη. Στο σημείο αυτό, είναι αξιοσημείωτο να αναφερθεί ότι οι CDR εγγραφές αναφέρονται μόνο στα χαρακτηριστικά που αναφέρθηκαν πιο πάνω και όχι στο αυτούσιο περιεχόμενο της κλήσης ή του μηνύματος.

2.2.2 Network Measurement Report (NMR)

Οι κινητές συσκευές περιοδικά παίρνουν μετρήσεις σχετικά με την ισχύ του σήματος (RSSI) που λαμβάνουν από τις κεραίες που είναι συνδεδεμένες. Οι μετρήσεις αποστέλλονται πίσω στις κεραίες για να δημιουργηθούν αναφορές μετρήσεων

(Network Measurement Reports). Οι μετρήσεις συμπεριλαμβάνουν στατιστικά που έχουν να κάνουν με το δίκτυο και την απόδοσή του. Αναλύοντας τις αναφορές αυτές, οι πάροχοι είναι σε θέση να γνωρίζουν την κατάσταση των κεραιών, και γενικότερα του δικτύου. Ταυτόχρονα, οι πάροχοι είναι σε θέση να γνωρίζουν τη θέση του χρήστη σε επίπεδο κεραίας. Οι μετρήσεις, ακόμη, παρέχουν δεδομένα όπως τον αριθμό των handover που χρειάστηκαν για να γίνει μια συσκευή μεταξύ κεραιών, τον αριθμό των αποτυχημένων συνδέσεων (call drops), κτλ. Παρόμοια αρχεία είναι και τα CHR (Call Trace History) / PCHR (Performance Call Trace History) [Κεφάλαιο 5], κτλ.

Κεφάλαιο 3

Σχετική Βιβλιογραφία

3.1 Έρευνα σε Μεγάλα Τηλεπικοινωνιακά Δεδομένα	16
3.1.1 Ανάλυση και Ανίχνευση σε Πραγματικό χρόνο	17
3.1.2 Εμπειρία, Συμπεριφορά και Διατήρηση Αναλύσεων	19
3.1.3 Θέματα Ιδιωτικότητας (Privacy)	20
3.1.4 Αποθήκευση (Storage)	21
3.2 Έρευνα σε Δεδομένα Κίνησης	22
3.2.1 Πρώτη Γενιά Χαρτογράφησης Κυκλοφορίας	23
3.2.2 Δεύτερη Γενιά Χαρτογράφησης Κυκλοφορίας	24
3.2.3 Γενιά Δύομιση (2.5) Χαρτογράφησης Κυκλοφορίας	26
3.2.4 Τρίτη Γενιά Χαρτογράφησης Κυκλοφορίας	32
3.2.5 Τεχνολογίες Βασισμένες σε Οχήματα	36

Στο παρόν κεφάλαιο θα συνοψιστούν βασικές πηγές βιβλιογραφίας που ασχολήθηκαν με την έρευνα σε μεγάλα τηλεπικοινωνιακά δεδομένα και σε δεδομένα κίνησης. Συγκεκριμένα, θα περιγραφεί η διαχείριση της αποθήκευσης των μεγάλων δεδομένων, η ανάλυση τους σε πραγματικό χρόνο, αλλά και τα διάφορα θέματα ιδιωτικότητας που προκύπτουν κατά τη διάρκεια ανάλυσης των δεδομένων αυτών. Τέλος, θα περιγραφούν και αναλυθούν οι διάφορες γενιές τεχνολογιών που υπήρξαν με σκοπό την χαρτογράφηση κυκλοφορίας.

3.1 Έρευνα σε Μεγάλα Τηλεπικοινωνιακά Δεδομένα

Τα τελευταία 20 χρόνια η τεχνολογία υπέστη ραγδαία εξέλιξη. Με την εξέλιξη αυτή, εξελίχθηκε και το μέγεθος της πληροφορίας που χρειάζονται οι υπολογιστές να

αποθηκεύσουν και να επεξεργάζονται. Κάπως έτσι, δημιουργήθηκαν τα Μεγάλα Δεδομένα (big data). Μεγάλα Δεδομένα ονομάζονται τα δεδομένα που έχουν μέγεθος από μια ντουζίνα GB και άνω. Στην ενότητα αυτή θα αναφερθούν διάφορες έρευνες που βασίζονται σε μεγάλα τηλεπικοινωνιακά δεδομένα.

3.1.1 Ανάλυση και Ανίχνευση σε Πραγματικό χρόνο

Στο σημείο αυτό θα αναφερθούν κάποιες από τις πιο σημαντικές έρευνες που υπήρξαν σχετικά με την ανάλυση και την ανίχνευση σε πραγματικό χρόνο. Μία από αυτές ήταν η OceanRT [27], μία από τις πρώτες σημαντικές αναλύσεις δεδομένων σε πραγματικό χρόνο, η οποία θα επεξηγηθεί με περισσότερη λεπτομέρεια στη συνέχεια. Μετά από την έρευνα αυτή, δημοσιεύθηκε η έρευνα με όνομα OceanST [26]. Σημαντικό είναι να αναφερθεί ότι και οι δύο έρευνες που προαναφέρθηκαν έχουν διεξαχθεί από την εταιρεία Huawei. Η OceanST διαθέτει ένα αποτελεσματικό μηχανισμό φόρτωσης των συνεχώς αυξημένων δεδομένων MBB και νέες χωροχρονικές (spatiotemporal) δομές δεικτών για τη διεκπεραίωση ερωτημάτων, που είναι είτε ακριβείς, είτε κατά προσέγγιση. Ακόμη, υπάρχει και η έρευνα CellIQ [14], η οποία βελτιστοποιεί διάφορα επερωτήματα, όπως τα “*spatiotemporal traffic hotspots*”. Αντιπροσωπεύει τα στιγμιότυπα των δεδομένων ενός κυψελοειδούς δικτύου (cellular networks) ως γραφήματα και αξιοποιεί τη χωρική (spatial) και προσωρινή τοποθεσία των δεδομένων αυτών.

Το χ.ο 2015 δημοσιεύτηκε ένα άλλο άρθρο, το οποίο έχει ως θέμα την ανάλυση σε κίνηση. Συγκεκριμένα, η έρευνα αυτή ασχολείται με την υψηλή απόδοση επεξεργασίας γεγονότων και την ανάλυση σε πραγματικό χρόνο στην ίδια βάση δεδομένων [5]. Αναπτύσσει ένα κλιμακωτό καταναεμημένο σύστημα που επεξεργάζεται αποτελεσματικά μικτά workloads και αναλύει επερωτήματα σε σχέση με τα τηλεπικοινωνιακά δεδομένα. Ένα χρόνο μετά, παρουσιάζεται ένα άλλο project, το οποίο ασχολείται με τη χρήση Telco Measurement Records με σκοπό τον τοπικό εντοπισμό χρηστών εντός μιας πόλης [28]. Το 2012 δημοσιεύεται μια έρευνα με σκοπό την ανάπτυξη ενός συστήματος πάνω σε middleware InfoSphere Streams της IBM, που αναλύει 6 δισεκατομμύρια CDR (Call Detail Records) ανά ημέρα σε πραγματικό χρόνο

[4]. Τέλος, ένα χρόνο αργότερα, παρουσιάζεται ένα σύστημα για τη διατήρηση των προφίλ των κλήσεων των πελατών με τη χρήση κλιμακούμενης, κατακεντρωμένης και stream επεξεργασίας [1].

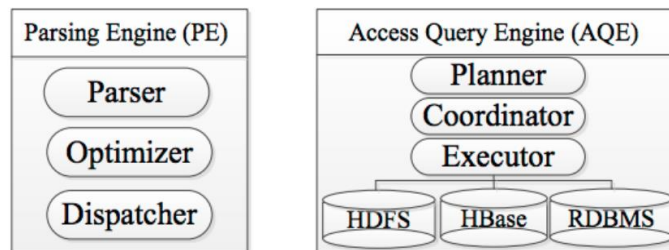
OceanRT: Ανάλυση Μεγάλων Προσωρινών Δεδομένων σε Πραγματικό Χρόνο

Η δημοσίευση αυτή είχε ως θέμα την ανάλυση μεγάλων προσωρινών δεδομένων. Υλοποιήθηκε μια υποδομή σε cloud, που εκτελεί online αναλύσεις σε πραγματικό χρόνο σε μεγάλης κλίμακας χρονικά δεδομένα. Ένα παράδειγμα δεδομένων που μπορεί το σύστημα να επεξεργαστεί είναι αρχεία καταγραφής κλήσεων. Αυτό που διαφοροποιεί την τακτική αυτή από άλλες, είναι το γεγονός ότι τα περισσότερα συστήματα ανάλυσης μεγάλων δεδομένων επεξεργάζονται τα δεδομένα offline, με αποτέλεσμα να περιορίζονται στην αποδοτικότητα. Αντίθετα, το σύστημα OceanRT χρησιμοποιεί μια νέα υπολογιστική αρχιτεκτονική με ενισχυμένο παραλληλισμό κι ένα νέο Σχήμα αποθήκευσης, το οποίο είναι βελτιστοποιημένο για τη νέα αρχιτεκτονική του συστήματος, καθώς τα δεδομένα και τα επρωτήματα (queries) είναι προσωρινά. Το σύστημα αποτελείται από διασυνδεδεμένους (interconnected) μηχανισμούς πρόσβασης - επρωτημάτων, καθώς και ένα σύστημα αποθήκευσης που εξασφαλίζει τοπικά δεδομένα και γρήγορη πρόσβαση σε προσωρινά δεδομένα. Επιπλέον, το σύστημα μπορεί να επεκταθεί με περισσότερους υπολογιστικούς πόρους.

Το demo του συστήματος διαχωρίζει ένα πραγματικό σύνολο δεδομένων καταγραφής κλήσεων, όπου έχουν μέγεθος περίπου 5TB την ημέρα. Οι χρήστες αλληλεπιδρούν με το demo μέσω επρωτημάτων HiveQL.

Όπως φαίνεται στο Σχήμα 3.1, το σύστημα χωρίζεται σε διασυνδεδεμένους κόμβους, όπου για κάθε κόμβο υπάρχει ανάλυση, βελτιστοποίηση και χρονοπρογραμματισμός επρωτημάτων. Ακόμη, χρησιμοποιεί ένα διαμορφωμένο HDFS σύστημα (Hadoop Distributed File System) για την αποθήκευση των εγγράφων με παρόμοια χρονικά χαρακτηριστικά. Λόγω αυτού, στο σύστημα υλοποιήθηκε ένα ευρετήριο με σκοπό την εκτέλεση επρωτημάτων.

Συγκεκριμένα, στο αριστερό μέρος του Σχήματος 3.1 φαίνεται η Μηχανή Ανάλυσης (Parsing Engine) που αποτελείται από τον parser, τον optimizer, και τον dispatcher. Ο parser αναλύει τα SQL επρωτήματα σε πλάνα εκτέλεσης (execution plans), τα οποία έπειτα αποστέλλονται στον optimizer. Ο optimizer βελτιστοποιεί τα πλάνα αυτά μέσω της βάσης δεδομένων Postgres. Ο dispatcher μεταφέρει τα επρωτήματα σε κόμβους που αποθηκεύουν τα σχετικά δεδομένα. Στο δεξιό μέρος του Σχήματος 3.1 απεικονίζεται η Μηχανή αναζήτησης Επερωτημάτων (Access Query Engine) που εκτελεί το πλάνο εκτέλεσης και δημιουργεί ενδιάμεσα αποτελέσματα. Τέλος, η AQE μηχανή μπορεί να διαβάζει και να γράφει δεδομένα σε HDFS, HBase ή RDBMS βάσεις δεδομένων.



Σχήμα 3.1 Γραφική Απεικόνιση της Ανάλυσης και Βελτιστοποίησης Επερωτημάτων [27]

Εν κατακλείδι, το σύστημα είναι περίπου δέκα φορές γρηγορότερο από άλλα συστήματα που επεξεργάζονται μεγάλα δεδομένα. Συγκεκριμένα, είναι δέκα φορές γρηγορότερο από το Apache Impala, δώδεκα φορές γρηγορότερο από το Shark (SQL Spark), και διακόσιες φορές γρηγορότερο από το Apache Hive.

3.1.2 Εμπειρία, Συμπεριφορά και Διατήρηση Αναλύσεων

Για την επεξεργασία και αποθήκευση μεγάλων δεδομένων απαιτήθηκε η διεξαγωγή έρευνας σχετικά με την εμπειρία, συμπεριφορά και διατήρηση αναλύσεων. Μία από τις έρευνες που ο Huang ασχολήθηκε μαζί με τους συνεργάτες [12] του πάνω στον τομέα

αυτό, απέδειξε εμπειρικά ότι η απόδοση πρόβλεψης του πελάτη μπορεί να βελτιωθεί σημαντικά μέσω των τηλεπικοινωνιακών δεδομένων.

Μία άλλη έρευνα περιλάμβανε την υλοποίηση ενός framework για την πρόβλεψη συμπεριφοράς των χρηστών. Στην έρευνα αυτή [18] αξιοποιήθηκαν πάνω από ένα εκατομμύριο χρήστες, οι οποίοι αντιπροσωπεύθηκαν ως έγγραφα που περιέχουν μια συλλογή από μεταβαλλόμενες “*spatiotemporal* λέξεις” που εκφράζουν την συμπεριφορά των χρηστών. Αυτό γίνεται μέσω της εξαγωγής των εγγράφων πρόσβασης (access records) από τα MBB δεδομένα, γνωστοποιώντας έτσι διάφορα θέματα για κάθε χρήστη. Αυτό, είναι χρήσιμο για την πρόβλεψη συμπεριφοράς των χρηστών.

Ο Ho με τους συνεργάτες του [10] δημοσίευσαν μία έρευνα που προτείνει έναν κατανεμημένο αλγόριθμο ανίχνευσης κοινότητας που έχει στόχο να ανακαλύψει ομάδες χρηστών που μοιράζονται παρόμοιες ιδιότητες, αντανακλώντας έτσι τη συμπεριφορά των πελατών.

3.1.3 Θέματα Ιδιωτικότητας (Privacy)

Στην πληροφορική, υπάρχει πάντοτε ένα πρόβλημα που πρέπει να λαμβάνεται υπόψη κατά τη σχεδίαση και υλοποίηση συστημάτων. Το πρόβλημα αυτό είναι τα προβλήματα ιδιωτικότητας που προκύπτουν, όπου στην περίπτωση των συστημάτων που επεξεργάζονται μεγάλα τηλεπικοινωνιακά δεδομένα προκύπτουν λόγω της αξιοποίησης προσωπικών δεδομένων. Γι’ αυτό τον λόγο πολλές έρευνες αφιερώθηκαν στην ασφάλεια σχετικά με τα μεγάλα δεδομένα.

Μία από τις έρευνες που ασχολήθηκαν με αυτό, είναι η έρευνα με θέμα τη διαφορετική ιδιωτικότητα στην τηλεπικοινωνιακή πλατφόρμα μεγάλων δεδομένων [11]. Στο άρθρο αυτό, μελετήθηκε η διαφορά προστασίας προσωπικών δεδομένων για εφαρμογές εξόρυξης δεδομένων μέσω των μεγάλων τηλεπικοινωνιακών δεδομένων. Η ιδιωτικότητα και η εμπιστευτικότητα είναι κρίσιμες για την αξιοπιστία των τηλεπικοινωνιακών παρόχων λόγω των εξαιρετικά ευαίσθητων δεδομένων του χρήστη που βρίσκονται στο CDR (Call Detail Record), όπως η διάρκεια μιας κλήσης. Η

συγκεκριμένη έρευνα έδειξε ότι για τα συστήματα εξόρυξης βιομηχανικών δεδομένων, οι ισχυρές εγγυήσεις προστασίας της ιδιωτικής ζωής που παρέχονται από τη διαφορά της ιδιωτικής ζωής διαπραγματεύονται απώλεια ακρίβειας 15% έως και 30%.

3.1.4 Αποθήκευση (Storage)

Ένα ακόμη σημαντικό θέμα που απασχολεί τα μεγάλα δεδομένα είναι η αποθήκευση και ειδικότερα το πώς μπορεί να γίνει εφικτή χωρίς να χαθεί τίποτα, ενώ ταυτοχρόνως να μην προκύψει σπατάλη χώρου στη μνήμη.

Για την επίτευξη αποθήκευσης μεγάλων δεδομένων αλλά και ‘ανακύκλωσης’ της μνήμης, πολλές έρευνες πρότειναν την συμπίεση των χωροχρονικών (spatiotemporal) δεδομένων. Κάποιες έρευνες το δοκίμασαν αυτό με δεδομένα καιρού (weather) [3], ενώ άλλες έρευνες δοκίμασαν την συμπίεση χωροχρονικών δεδομένων χρησιμοποιώντας συλλογικά έγγραφα κειμένου [24]. Υπήρξαν έρευνες που αξιοποιούν ροές (streams) δεδομένων κυμαινόμενου σημείου [6], και άλλες που χρησιμοποίησαν επιστημονικά δεδομένα κυμαινόμενου σημείου προσομοίωσης [15], [16], [21], [22]. Τέλος, άλλες έρευνες συγκεντρώθηκαν σε τεχνικές διαφορικής συμπίεσης και την βελτίωση μεταξύ του λόγου συμπίεσης και των χρόνων αποσυμπίεσης για αυξανόμενα αρχειακά δεδομένα (incremental archival data) [2], [9], [25].

Παρ’ όλα αυτά, καμία από τις πιο πάνω δημοσιεύσεις δεν είχαν προτείνει κατανεμημένα συστήματα και δεν μπορούσαν οι λύσεις αυτές να εφαρμοστούν απευθείας σε τηλεπικοινωνιακά δεδομένα. Για τον λόγο αυτό, προτάθηκε η έρευνα με όνομα Spate [7]. Το έργο αυτό, ασχολήθηκε με την ελαχιστοποίηση του αποθηκευτικού χώρου που απαιτείται για την διατήρηση δεδομένων με την πάροδο του χρόνου, και την ελαχιστοποίηση του χρόνου απόκρισης για επερωτήματα διερεύνησης χωροχρονικών (spatiotemporal) πρόσφατων δεδομένων. Το επίπεδο αποθήκευσης (storage layer) που χρησιμοποιήθηκε ήταν μία συμπίεση δεδομένων χωρίς απώλειες για να εισάγει πρόσφατα streams μεγάλων τηλεπικοινωνιακών δεδομένων με τον πιο συμπαγή τρόπο, διατηρώντας πλήρη ανάλυση για την εξερεύνηση δεδομένων. Το indexing επίπεδο του συστήματος φροντίζει για την προοδευτική απώλεια των λεπτομερειών στις

πληροφορίες, δημιουργώντας φθορά, καθώς τα δεδομένα γερνούν με το χρόνο. Το επίπεδο που ασχολείται με την εξερεύνηση δεδομένων παρέχει οπτικά μέσα για τη διερεύνηση του χωροχρονικού χώρου πληροφοριών που δημιουργείται μέσω του συστήματος. Συνοπτικά, το Spate σύστημα έδειξε ότι μπορεί να επιτύχει συγκρίσιμους χρόνους απόκρισης σε σχέση με άλλα συστήματα τελευταίας τεχνολογίας χρησιμοποιώντας μια λιγότερη τάξη μεγέθους σε χώρο αποθήκευσης (one magnitude of storage space less).

3.2 Έρευνα σε Δεδομένα Κίνησης

Όπως ήδη προαναφέρθηκε, η διπλωματική αυτή εστιάζει στην εύρεση της κίνησης στους δρόμους. Αυτό προϋποθέτει τη συλλογή και επεξεργασία δεδομένων που βρίσκονται σε κίνηση και γι' αυτόν ακριβώς τον λόγο χρειάστηκε επίσης να μελετηθούν διάφορες έρευνες που επεξεργάζονται δεδομένα κίνησης. Στην ενότητα αυτή θα αναφερθούν και θα επεξηγηθούν λεπτομερώς κάποιες από τις έρευνες αυτές.

Towards Real-Time Road Traffic Analytics using Telco Big Data

Το 2017, ο Κώστα μαζί με τους συνεργάτες του δημοσίευσαν την έρευνα [8] με θέμα την ανάλυση τηλεπικοινωνιακών δεδομένων σε πραγματικό χρόνο με σκοπό την εύρεση της κίνησης στους δρόμους. Ο σκοπός της παρούσας διπλωματικής είναι η υλοποίηση και συνέχιση της έρευνας του Κώστα και των συνεργατών του. Στη δημοσίευση περιγράφεται η αρχιτεκτονική του συστήματος που αποσκοπεί στην καινοτόμα ανάλυση και πρόβλεψη οδικής κυκλοφορίας. Συγκεκριμένα, παρέχει μοντελοποίηση και πρόβλεψη κυκλοφορίας σ' ένα μικρό επίπεδο, υπερβαίνοντας τις δυνατότητες που μπορούν να παρέχουν πληθοποριστικές (crowdsourcing) εφαρμογές πλοήγησης, οι οποίες είναι βασισμένες στη σύνδεσή τους με το διαδίκτυο. Ακόμη, έχει ως βασικό σκοπό τη διατήρηση των ορίων της ιδιωτικής ζωής των χρηστών, ώστε να αποφεύγονται οι κίνδυνοι έκθεσης δεδομένων ή της τοποθεσίας των χρηστών σε κινητές εφαρμογές τρίτων (third-party mobile applications). Τέλος, αποσκοπεί στη διαθεσιμότητα του συστήματος με το ελάχιστο δυνατόν κόστος, χρησιμοποιώντας

υπάρχουσα υποδομή, π.χ., τους πύργους των κεραιών και τα streams δεδομένων των τηλεπικοινωνιακών παρόχων που είναι διαθέσιμα.

3.2.1 Πρώτη Γενιά Χαρτογράφησης Κυκλοφορίας

Η κίνηση στους δρόμους είναι ένα μεγάλο πρόβλημα που όλοι αντιμετωπίζουμε καθημερινά, ειδικότερα στην Κύπρο που τα αυτοκίνητα είναι το κύριο μέσο μεταφοράς. Γι' αυτόν ακριβώς τον λόγο υπήρξαν προσπάθειες χαρτογράφησης της κυκλοφορίας στους δρόμους.

Στην πρώτη γενιά χαρτογράφησης χρησιμοποιήθηκε αποκλειστικά εξοπλισμός. Συγκεκριμένα, χρησιμοποιήθηκαν ανιχνευτές μαγνητικού βρόγχου [57] (magnetic loop detectors), ή αλλιώς αισθητήρες οχημάτων. Οι αισθητήρες αυτοί, ανιχνεύουν οχήματα που διέρχονται ή φτάνουν σε ένα συγκεκριμένο σημείο. Μπορούν να χρησιμοποιηθούν τόσο για ανίχνευση οχημάτων, δηλαδή τον ακριβή αριθμό οχημάτων που κινούνται στον δρόμο, όσο και για συστήματα καθοδήγησης και πληροφόρησης στάθμευσης. Ανιχνεύουν μόνο μεταλλικές μάζες πάνω από ένα ορισμένο μέγεθος, έτσι ώστε να ανιχνεύονται σωστά μόνο οχήματα, αγνοώντας πεζούς, ποδήλατα, μοτοσυκλέτες κτλ. Δίνουν ακόμη τη δυνατότητα ρύθμισης, έτσι ώστε να ανιχνεύονται και μοτοσυκλέτες / σκούτερ. Ο ανιχνευτής φαίνεται και στο Σχήμα 3.2.



Σχήμα 3.2 Ανιχνευτής Μαγνητικού Βρόγχου [58]

Ως δεύτερο μέσο για την χαρτογράφηση κυκλοφορίας χρησιμοποιήθηκαν κάμερες κυκλοφορίας που είναι τοποθετημένες σε φώτα τροχαίας ή στο πλάι μέρος των δρόμων. Τυπικά, στην Κύπρο οι κάμερες αυτές υπάρχουν για την παρακολούθηση των ταχυτήτων που κινούνται τα οχήματα έτσι ώστε να υποβάλλουν πρόστιμο σε όσους το ξεπερνούν. Παρ' όλα αυτά, μπορούν να χρησιμοποιηθούν και για την ανίχνευση της κυκλοφορίας. Οι κάμερες αυτές φαίνονται και στο Σχήμα 3.3.



Σχήμα 3.3 Κάμερες Κυκλοφορίας [59]

Το πρόβλημα με την πρώτη γενιά, ήταν ότι βασίστηκε στην εγκατάσταση και συντήρηση ακριβού εξοπλισμού, που θα έπρεπε να τοποθετηθεί είτε σε όλους τους δρόμους, είτε στους πιο κύριους. Αυτό είχε ως αποτέλεσμα να κάνει σχεδόν αδύνατη την κλιμάκωση του εξοπλισμού αυτού σε μεγάλες αστικές και αγροτικές περιοχές.

3.2.2 Δεύτερη Γενιά Χαρτογράφησης Κυκλοφορίας

Με την ύπαρξη της πρώτης γενιάς, αλλά και την εμφάνιση των έξυπνων κινητών (smartphones), γεννήθηκε η ανάγκη δημιουργίας πιο αυτοματοποιημένων τρόπων για την χαρτογράφηση κυκλοφορίας, γεγονός που οδήγησε στη δημιουργία της δεύτερης γενιάς.

Η δεύτερη γενιά βασίζεται σε φορητά οχήματα και οχήματα ανίχνευσης. Επιπρόσθετα, βασίζεται σε ταξί ή λεωφορεία που είναι εξοπλισμένα με μια ειδική εφαρμογή για την ενημέρωση της τοποθεσίας τους. Τέλος, υπάρχουν κι άλλες τεχνολογίες που βοηθούν στη χαρτογράφηση κυκλοφορίας, όπως το MIT Cartel, Here Maps, Google Maps, κτλ.

MIT Cartel

Η έρευνα αυτή [13] υλοποίησε ένα κινητό σύστημα υπολογιστικών αισθητήρων σχεδιασμένο να συλλέγει, να επεξεργάζεται, να παραδίδει και να απεικονίζει δεδομένα από αισθητήρες τοποθετημένους σε κινητές μονάδες, π.χ., αυτοκίνητα.

Ένας Cartel κόμβος είναι ένας ενσωματωμένος κινητός υπολογιστής συνδεδεμένος σε ένα σύνολο αισθητήρων. Κάθε κόμβος συγκεντρώνει και επεξεργάζεται τοπικά τις μετρήσεις αισθητήρων, οι οποίες συλλέγονται καθώς οι χρήστες οδηγούν, κι έπειτα τις παραδίδει στην κεντρική βάση δεδομένων, όπου εκεί τα δεδομένα αναλύονται και απεικονίζονται. Στο Σχήμα 3.4 απεικονίζεται ο κόμβος, ο οποίος βασίζεται κυρίως σε ασύρματες συνδέσεις με το διαδίκτυο, π.χ., Wi-Fi, Bluetooth, κτλ. Στη βάση δεδομένων, πραγματοποιείται το φιλτράρισμα, η συνόψιση αλλά και η εφαρμογή προτεραιότητας στα δεδομένα.



Σχήμα 3.4 Cartel Κόμβος [13]

Επιπλέον, υλοποιήθηκε μια απλή διεπαφή προγραμματισμένη να χειρίζεται μεγάλο αριθμό ετερογενών δεδομένων από αισθητήρες. Η διεπαφή ακόμη, μπορεί να απαντά σε επερωτήματα (queries) σχετικά με τα δεδομένα αυτά και είναι συνδεδεμένη με μία κεντρική βάση δεδομένων.

3.2.3 Γενιά Δυόμιση (2.5) Χαρτογράφησης Κυκλοφορίας

Μια κάπως βελτιωμένη προσέγγιση της δεύτερης γενιάς, αποτέλεσε η γενιά 2.5. Με την έγκριση χρηστών, χρησιμοποιήθηκαν οι κινητές συσκευές τους για τη συνεισφορά δεδομένων τόσο μέσω πληθοπορισμού (crowdsourcing), δηλαδή να παρέχει ο ίδιος ο χρήστης πληροφορίες στο σύστημα, όσο και σε περίπτωση που ο χρήστης επιτρέπει στο σύστημα να λαμβάνει την θέση του (location). Συστήματα που εφάρμοσαν την λογική αυτή είναι το Google Maps, Apple Maps, Waze, Nokia's Here Maps, ο Ctrack Αλγόριθμος κτλ.

Τα προβλήματα της δεύτερης και δυόμιση γενιάς είναι ότι βασίζονται στην πλήρη συνεργασία των χρηστών που πολλές φορές είναι δύσκολη ή μειωμένη. Ακόμη, τα συστήματα αυτά εκθέτουν τη θέση των χρηστών σε κινητές εφαρμογές τρίτων (third-party mobile application), με αποτέλεσμα της εισαγωγής ανησυχίας σχετικά με την προστασία της ιδιωτικότητας των χρηστών.

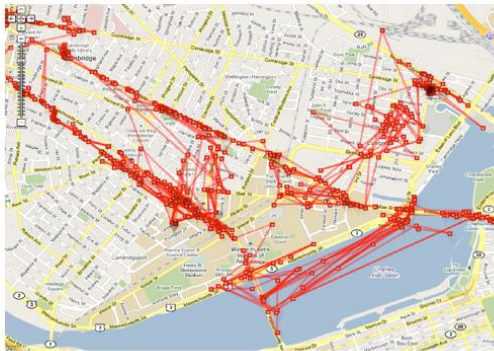
Ctrack Αλγόριθμος

Η αντιστοίχιση χαρτών και συγκεκριμένα η αντιστοίχιση τροχιάς, είναι ένα ευρέως γνωστό πρόβλημα στις μέρες μας στον τομέα της πληροφορικής. Πολλές εφαρμογές είναι σχεδιασμένες βάσει της τοποθεσίας του χρήστη, π.χ., εφαρμογές που παρακολουθούν την κίνηση, εφαρμογές πλοήγησης, κτλ. Η πιο εύκολη λύση για το πρόβλημα αυτό, είναι μέσω της χρήσης του GPS, μιας κι αυτό μπορεί να μας δώσει τα πιο ακριβή αποτελέσματα. Παρ' όλα αυτά, η χρησιμοποίηση GPS οδηγεί σε μεγάλη και γρήγορη σπατάλη της μπαταρίας των κινητών συσκευών. Γι' αυτόν τον λόγο, ο Ctrack Αλγόριθμος [23] προτάθηκε για την επίλυση του προβλήματος αυτού, μειώνοντας κατά πολύ την χρήση μπαταρίας.

Περιγραφή Αλγορίθμου

Ο αλγόριθμος αυτός, αρχικά παίρνει και αποθηκεύει GSM τροχιές από πραγματικούς χρήστες και τις αποθηκεύει σε μια βάση δεδομένων. Οι τροχιές αυτές, έχουν παράλληλα γύρω στα 100 - 200 μέτρα σφάλμα (error) από τις πραγματικές τροχιές που οι χρήστες έχουν ακολουθήσει. Κατά τη συλλογή των τροχιών, η android εφαρμογή που χρησιμοποιούν, συλλέγει δεδομένα σχετικά με το σήμα του χρήστη, τις κεραίες που εντοπίζει η συσκευή, αλλά και την τοποθεσία του χρήστη χρησιμοποιώντας GPS. Επιπρόσθετα, η εφαρμογή συλλέγει επιλεκτικά κάποια δεδομένα χρησιμοποιώντας τους αισθητήρες της συσκευής, (π.χ., πυξίδα της συσκευής), οι οποίοι θα χρησιμοποιηθούν τόσο για την καλύτερη ανίχνευση στροφών κατά τη διάρκεια της τροχιάς, όσο και τον εντοπισμό των χρονικών στιγμών που κινείται ο χρήστης.

Όπως φαίνεται και στο Σχήμα 3.5, το κύριο μέρος του αλγορίθμου χωρίζεται σε τέσσερα βήματα. Στο πρώτο βήμα, φορτώνονται στη μνήμη όλες οι μετρήσεις που υπάρχουν στη βάση δεδομένων. Οι μετρήσεις αυτές, συμπεριλαμβάνουν τις ταυτότητες (ids) των κεραιών, μαζί με όλες τις τιμές σήματος που υπάρχουν αποθηκευμένες στη βάση για κάθε μία από τις κεραίες. Συγκεκριμένα, τα ακατέργαστα αυτά σημεία, που φαίνονται και στο Σχήμα 3.5 (α), αποτελούν την κεντρική τοποθεσία για κάθε κεραία σε σχέση με όλες τις μετρήσεις που υπάρχουν για αυτή. Το δεύτερο βήμα χωρίζεται σε δύο περάσματα. Το πρώτο πέραςμα γίνεται μέσω του Hidden Markov Model (HMM) αλγόριθμου, ο οποίος χρησιμοποιείται ιδιαίτερα για την επίλυση του map - matching προβλήματος. Συγκεκριμένα, στο βήμα αυτό, διαχωρίζει τον χώρο - χάρτη σε κελιά (πλέγμα) και καθορίζει την πιο πιθανή ακολουθία συνδεδεμένων κελιών. Για την καλύτερη ακολουθία συνδεδεμένων κελιών, το αποτέλεσμα του HMM περνά από μια φάση εξομάλυνσης και παρεμβολής, όπως φαίνεται στο Σχήμα 3.5 (γ). Στο Σχήμα 3.5 (δ) απεικονίζεται το δεύτερο πέραςμα, το οποίο χρησιμοποιεί διαφορετικό HMM για να αντιστοιχίσει την ακολουθία συνδεδεμένων κελιών σε οδικά τμήματα - δρόμους, και παίρνει σαν είσοδο το αποτέλεσμα που φαίνεται στο Σχήμα 3.5 (γ). Σε όλες τις εικόνες του Σχήματος 3.5, η μαύρη γραμμή αντιπροσωπεύει την πραγματική διαδρομή που ο χρήστης ακολούθησε.



(α) Ακατέργαστα Σημεία από Τροχιές



(β) Ακολουθία Συνδεδεμένων Κελιών - Πλέγμα



(γ) Εφαρμογή Εξομάλυνσης και Παρεμβολής σε Πλέγμα



(δ) Ομαλό Πλέγμα – Τμήματα Οδικού Δρόμου

Σχήμα 3.5 Γραφική Απεικόνιση Κάθε Βήματος του Ctrack Αλγόριθμου σε Σχέση με την Πραγματική Τροχιά [23]

Εκπαίδευση Συστήματος

Το σύστημα, στο σημείο αυτό, χωρίζει την γεωγραφική περιοχή που πρέπει να επεξεργαστεί, σε ομοιόμορφα τετραγωνικά πλέγματα σταθερού μεγέθους. Η εκπαίδευση αυτή γίνεται πάνω σε κινητές συσκευές κατά τη διάρκεια των τροχιών, όπου η εφαρμογή αποθηκεύει ένα log αρχείο που περιέχει timestamp μαζί με τα GPS σημεία που η τροχιά έχει καλύψει και τις κεραίες που σχετίζονται με την διαδρομή αυτή. Για κάθε πλέγμα G που ανήκει στην τροχιά, η βάση δεδομένων αποθηκεύει F_G , όπου είναι το σύνολο των ξεχωριστών τροχιών που φαίνονται στο G . Μόλις κατασκευαστεί η βάση δεδομένων κατάρτισης, μπορεί να χρησιμοποιηθεί για την

αντιστοίχιση ή την παρακολούθηση οποιασδήποτε τροχιάς. Παράλληλα, η βάση δεδομένων θα χρειάζεται σπάνια αναβάθμιση.

Επαλήθευση Αλληλουχίας Δικτύου

Η επαλήθευση αλληλουχίας δικτύου, αλλιώς δημιουργία πλέγματος, όπως αναφέρθηκε πιο πάνω, χρησιμοποιεί τον αλγόριθμο Hidden Markov Model (HMM). Ο HMM αλγόριθμος είναι μια διαδικασία Markov διακριτού χρόνου με ένα σύνολο κρυμμένων καταστάσεων και παρατηρήσεων. Κάθε κατάσταση εκπέμπει μια παρατήρηση, της οποίας η πιθανότητα δίνεται από μια εκπομπή βαθμολογιών (emission score). Ακόμη, ο αλγόριθμος αυτός επιτρέπει μεταβάσεις ανάμεσα στις κρυμμένες καταστάσεις οποιαδήποτε στιγμή. Οι μεταβάσεις αυτές διέπονται από ένα διαφορετικό σύνολο πιθανοτήτων, που ονομάζονται αποτελέσματα μετάβασης (transition score).

Ο Hidden Markov Model αλγόριθμος, χρησιμοποιήθηκε για τον προσδιορισμό της ακολουθίας των αντίστοιχων κεραιών, που σχετίζονται με συγκεκριμένα κελιά του πλέγματος, σε σχέση με την ακολουθία της τροχιάς.

Στο πρώτο HMM πέρασμα του αλγορίθμου, οι κρυμμένες μεταβάσεις είναι τα κελιά του πλέγματος και οι παρατηρήσεις είναι οι τροχιές. Η βαθμολογία που εκπεμπέθηκε (emission score), καταγράφει την πιθανότητα παρατήρησης της τροχιάς F στο σημείο G. Η μεταβατική βαθμολογία καταγράφει την πιθανότητα μετάβασης από την πρώτη στη δεύτερη κεραία σε ένα μόνο χρονικό βήμα.

Ο Ctrack αλγόριθμος, αρχικά επεξεργάζεται τις GSM τροχιές, τις οποίες παίρνει ως είσοδο χρησιμοποιώντας την τεχνική ‘Windowing’. Στην συνέχεια, προσπαθεί να βρει την μέγιστη ακολουθία πιθανοτήτων των κεραιών, που αντιστοιχούν στο πλέγμα, που αντιστοιχεί στη ‘windowed’ εκδοχή της τροχιάς. Η πιθανότητα μέγιστης ακολουθίας ορίζεται ως η ακολουθία που δίνει τη μεγαλύτερη εκπομπή βαθμολογίας (emission score) και τα μεγαλύτερα αποτελέσματα μετάβασης (transition score). Τα αποτελέσματα μετάβασης υπολογίζονται βάσει τον αλγόριθμο Manhattan, σε σχέση με το αν ο χρήστης στην τροχιά έχει μεταβεί σε διαφορετική κεραία ή όχι. Αν έχει μεταβεί

σε διαφορετική κεραία, τότε ο αλγόριθμος προσπαθεί να βρει σε ποια κοντινή κεραία έχει μεταβεί.

Στο σημείο αυτό, είναι σημαντικό να αναφερθεί ότι ο HMM αλγόριθμος εμπερικλείει ένα περιορισμό ταχύτητας (speed constraint) που απαγορεύει μεταβάσεις από ένα τμήμα εάν δεν έχει περάσει επαρκής χρόνος στο τμήμα αυτό. Η μέγιστη αυτή ταχύτητα βαθμονομείται ανάλογα με το αν ένας χρήστης περπατά ή οδηγεί όταν χρησιμοποιεί την εφαρμογή.

Τεχνική Windowing

Η τεχνική ομαδοποιεί ακατέργαστες τροχιές και συμβάλλει στην καλύτερη ακρίβεια. Η ανάγκη της προήλθε από το γεγονός ότι οι τροχιές εισέρχονται στο σύστημα κάθε δευτερόλεπτο, όμως οι τροχιές του κάθε δευτερολέπτου μπορεί να μην περιλαμβάνει κεραίες οι οποίες σχετίζονται με την ίδια περιοχή της συγκεκριμένης τροχιάς. Γι' αυτό τον λόγο, οι τροχιές χωρίζονται σε 'παράθυρα' τοποθεσιών.

Εξομάλυνση και Παρεμβολή (Smoothing and Interpolation)

Η φάση αυτή παίρνει σαν είσοδο το πλέγμα και το μετατρέπει σε μία ακολουθία από γεωγραφικά μήκη και πλάτη (lat, lon), τα οποία θα επεξεργαστούν στην φάση της αντιστοίχισης οδικού τμήματος.

Συγκεκριμένα, στην εξομάλυνση παίρνει κάθε πλέγμα από μια ακολουθία τροχιάς και υπολογίζεται το κέντρο των σημείων του πλέγματος. Παρ' όλα αυτά, είναι συχνό φαινόμενο η ύπαρξη συχνότητας υψηλού θορύβου (high frequency noise), με αποτέλεσμα τη μειωμένη ακρίβεια. Το πρόβλημα αυτό επιλύθηκε με την εφαρμογή εξομάλυνσης, η οποία εφαρμόζει σε μικρό επίπεδο την τεχνική Windowing στα κέντρα των σημείων αυτών.

Απ' την άλλη, η παρεμβολή λύνει το πρόβλημα του λανθασμένου υπολογισμού τροχιάς που κατευθύνεται συχνά μπρος - πίσω. Το πρόβλημα αυτό συμβαίνει επειδή η

αντιστοίχιση οδικού τμήματος παράγει μια συνεχή τροχιά όπου κάθε τμήμα αντιστοιχεί σε τουλάχιστον μία τιμή εισόδου. Η ελάχιστη επιθυμητή συχνότητα τιμών εισόδου για την αντιστοίχιση τμήματος είναι μία. Με αυτόν τον τρόπο, ο αλγόριθμος εξασφαλίζει ότι και το μικρότερο τμήμα αντιστοιχεί σε τουλάχιστον μία τιμή.

Αντιστοίχιση Οδικού Τμήματος (Segment Matching)

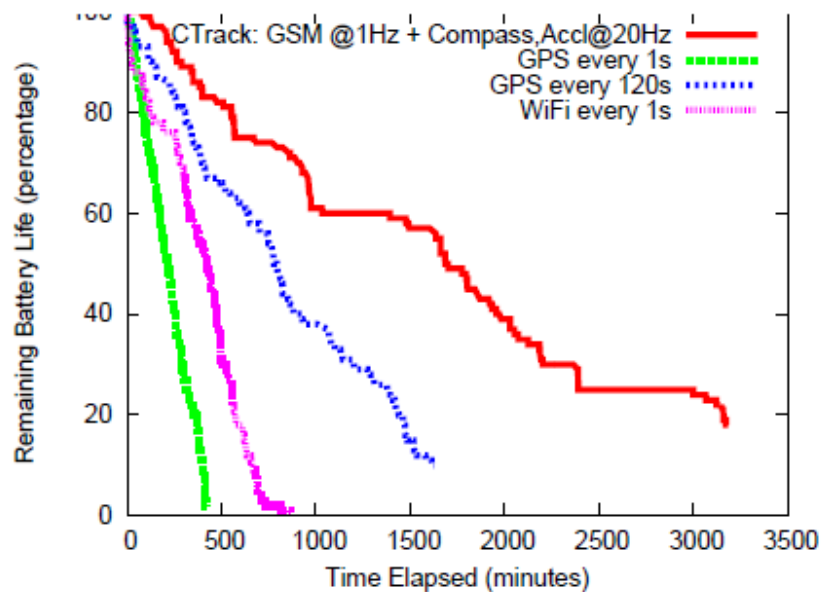
Στην φάση αυτή, γίνεται η αντιστοίχιση τροχιών με οδικά τμήματα πάνω σε χάρτη. Το σύστημα παίρνει σαν είσοδο τις τροχιές που πέρασαν από την φάση της εξομάλυνσης και της παρεμβολής μαζί με τα επιλεκτικά δεδομένα που μαζεύτηκαν μέσω των αισθητήρων της συσκευής για να καθορίσει την πιο πιθανή ακολουθία τμημάτων που διέσχισε ο χρήστης.

Το αποτέλεσμα της φάσης αυτής είναι ένα σύνολο από τμήματα, ένα ανά σημείο από την τροχιά, δείχνοντας την υπολογιζόμενη διαδρομή του χρήστη. Κατά μέσο όρο είναι η ίδια με τη διαδρομή που έχει δοθεί σαν είσοδο.

Αξιολόγηση Αλγορίθμου

Όπως φαίνεται και στο Σχήμα 3.6, χρησιμοποιώντας κινητά δίκτυα (GSM) αντί GPS, επιτυγχάνεται η λιγότερη κατανάλωση μπαταρίας, λόγω του ότι τα δίκτυα αυτά είναι πάντοτε ενεργοποιημένα σε όλες τις κινητές συσκευές, σε αντίθεση με το GPS.

Εν κατακλείδι, ο αλγόριθμος επιτυγχάνει 75% ακρίβεια και 80% ακρίβεια ανάκλησης. Τα αποτελέσματα αυτά, είναι καλύτερα από άλλες παρόμοιες εφαρμογές, και συγκεκριμένα μειώνει τα σφάλματα αντιστοίχισης τροχιών κατά 2,5 φορές. Παράλληλα, μειώνει τα σφάλματα αντιστοίχισης χαρτών κατά 3,5 φορές.



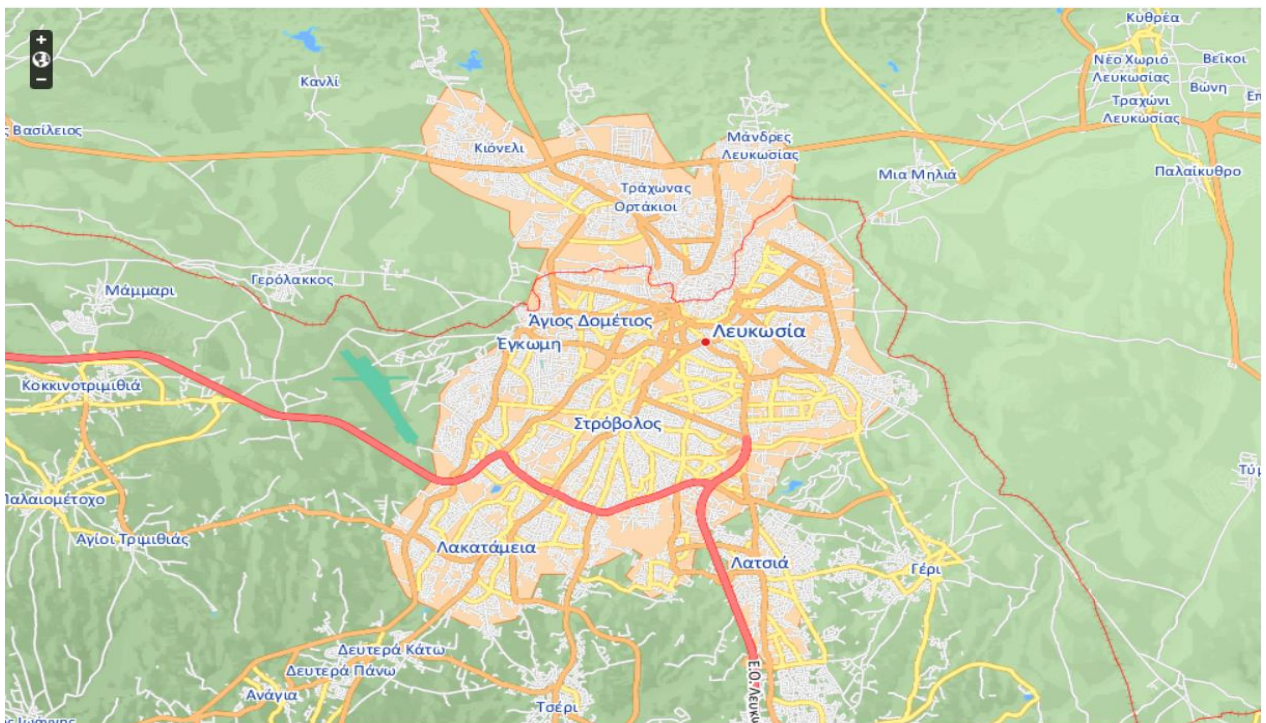
Σχήμα 3.6 Γραφική Απεικόνιση της Κατανάλωσης Μπαταρίας για WiFi, GPS, και Ctrack Αλγόριθμο σε Android Συσκευή [23]

3.2.4 Τρίτη Γενιά Χαρτογράφησης Κυκλοφορίας

Συστήματα τρίτης γενιάς, όπως το TOMTOM, αλλά και το σύστημα που θα προταθεί στην παρούσα διπλωματική, βασίζονται σε δεδομένα που έχουν ήδη παραχθεί από συστήματα που ανήκουν στην 2.5 γενιά χωρίς να χρειάζονται επιπλέον εξοπλισμό ή κόστος. Ακόμη, τα συστήματα αυτά, αποσκοπούν στην εφαρμογή και λύση θεμάτων που ενδιαφέρουν ιδιαίτερα την κοινωνία μας σήμερα, όπως τα θέματα ιδιωτικότητας των χρηστών ή τα θέματα ακρίβειας που μπορεί προηγούμενα συστήματα να είχαν.

Σύστημα TOMTOM

Ακόμη μία προσπάθεια επίλυσης του προβλήματος χαρτογράφησης κυκλοφορίας, ήταν αυτή του συστήματος TOMTOM. Σκοπός του συστήματος είναι η παροχή κυκλοφορίας (traffic), πλοήγησης και χαρτογράφησης. Συγκεκριμένα, χαρτογραφεί την κυκλοφορία με τη βοήθεια χρηστών που χρησιμοποιούν Vodafone δίκτυο. Καθώς οι χρήστες οδηγούν, ταυτόχρονα το σύστημα ενημερώνεται με πληροφορίες σχετικά με την κυκλοφορία. Επιπρόσθετα, το σύστημα χρησιμοποιεί χωρική (spatial) αναλυτική διεπαφή για την απεικόνιση της κυκλοφορίας και επικεντρώνεται στη λειτουργικότητα της δρομολόγησης (plan routing). Η διεπαφή αυτή διακρίνεται και στο Σχήμα 3.7, όπου φαίνεται η λειτουργία του συστήματος για τα οδικά δίκτυα της Κύπρου. Παρ' όλα αυτά, το σύστημα δεν είναι ικανό να παρέχει τη μελέτη της κυκλοφορίας για διάφορες χρονικές στιγμές, παρά μόνο για τη συγκεκριμένη δεδομένη στιγμή. Τέλος, λόγω του ότι το σύστημα TOMTOM είναι ένα κλειστό σύστημα, δεν είναι δυνατή η λεπτομερής εξακρίβωση του πώς λειτουργεί.



Σχήμα 3.7 TomTom Σύστημα Κύπρου [60]

Εντοπισμός Χρηστών σε Κλίμακα Πόλης με Τηλεπικοινωνιακά Δεδομένα

Ο εντοπισμός κινητών συσκευών σε κλίμακα πόλης χρησιμοποιώντας δεδομένα μετρήσεων (Measurement Reports) είναι κάτι αρκετά δύσκολο και ταυτόχρονα κάτι αρκετά χρήσιμο. Η επίτευξη του εντοπισμού ενός χρήστη μέσω αυτών των δεδομένων θα έλυνε πολλά προβλήματα, όπως η γρήγορη κατανάλωση μπαταρίας. Στην έρευνα αυτή [28], μελετήθηκε το πώς θα μπορούσε αυτό να πραγματοποιηθεί και πόσο ακριβής θα μπορούσε να είναι ο τρόπος αυτός. Το σύστημα βασίζεται σε μεθοδολογίες σχετικά με map - matching προβλήματα αλλά και με machine learning τεχνικές.

Τα δεδομένα μετρήσεων (MR) μετρούν τις παραμέτρους του ραδιοσήματος όταν οι κινητές συσκευές συνδέονται με base stations - κεραίες για την πραγματοποίηση / λήψη κλήσεων ή υπηρεσιών κινητής ευρυζωνικής σύνδεσης (MBB). Συγκεκριμένα, κρατούν δεδομένα όπως το σήμα που λαμβάνουν οι συσκευές από τις κεραίες (Radio Signal Strength Indicator).

Κατά τη διάρκεια της έρευνας αυτής, φάνηκε ότι πολλές ευρέως αναγνωρίσιμες εφαρμογές που βασίζονται στην τοποθεσία, έχουν συσσωρεύσει πολλά GPS δεδομένα σε Τηλεπικοινωνιακά δεδομένα (Telco Big Data) τα οποία μπορούν να χρησιμοποιηθούν αυτόματα με σκοπό την εκπαίδευση συστημάτων να εντοπίζουν με ακρίβεια την τοποθεσία των χρηστών μέσω των MR. Την λογική αυτή ακολούθησε και η συγκεκριμένη έρευνα. Επωφελούμενοι από αυτά τα μεγάλα τηλεπικοινωνιακά δεδομένα, μέσω αυτής της έρευνας, αναπτύχθηκε ένα μοντέλο υποκειμενικής παλινδρόμησης (Context - Aware Coarse - to - Fine Regression) ή αλλιώς CCR. Το μοντέλο αυτό αναπτύχθηκε πάνω σε Spark / Hadoop πλατφόρμα που βασίζεται σε τηλεπικοινωνιακά δεδομένα με σκοπό τον τοπικό εντοπισμό σε κινητές συσκευές.

Ως πρώτο βήμα, σχεδιάστηκε ένας αλγόριθμος αντιστοίχισης σε χάρτη (map - matching) και παρεμβολής για την κωδικοποίηση πληροφοριών σχετικά με τα οδικά δίκτυα. Στο βήμα αυτό γίνεται πρώτα η αντιστοίχιση των GPS συντεταγμένων με μια τροχιά στο οδικό δίκτυο. Μετά, για κάθε ζεύγος γειτονικών ταιριασμένων σημείων, τα

οποία απέχουν ένα σταθερό μικρό μήκος χρόνου μεταξύ τους, υπολογίζεται η πιο συχνή διαδρομή μεταξύ τους και γίνεται μια γραφική παράσταση που απεικονίζει μια ομοιόμορφη παρεμβολή βάσει αυτού. Τα βήματα αυτά γίνονται παράλληλα αφού είναι υλοποιημένα πάνω σε Spark / Hadoop σύστημα. Αυτό έχει ως αποτέλεσμα την αύξηση του ρυθμού δειγματοληψίας των GPS δεδομένων.

Έπειτα, υλοποιήθηκε ένα μοντέλο παλινδρόμησης δύο επιπέδων για να καταγραφούν τα coarse - to - fine contextual χαρακτηριστικά σε σύντομο χρονικό διάστημα. Αυτό γίνεται για την επίτευξη πιο βελτιωμένης απόδοσης εντοπισμού του χρήστη. Η φάση αυτή δίνει ως αποτέλεσμα την υπολογιζόμενη πρόβλεψη σχετικά με την τοποθεσία του χρήστη.

Ιδιαίτερα σημαντικό χαρακτηριστικό για την τακτική αυτή είναι η συχνή ανανέωση GPS δεδομένων. Αυτό βοηθά το σύστημα να πετυχαίνει καλύτερη επίδοση και καλύτερο αποτέλεσμα. Ο λόγος που χρειάζεται συχνή ανανέωση δεδομένων είναι επειδή για την ίδια τοποθεσία μια κινητή συσκευή μπορεί να παίρνει διαφορετικό σήμα από την κεραία.

Στο Σχήμα 3.8 φαίνονται κάποια από τα αποτελέσματα του συστήματος. Τα μπλε σημεία αντιπροσωπεύουν τις προβλέψεις του συστήματος σχετικά με την τοποθεσία της συσκευής, ενώ οι κόκκινες γραμμές αντιπροσωπεύουν την πραγματική τροχιά των χρηστών. Όπως φαίνεται και στο Σχήμα, το σύστημα έχει μικρά σφάλματα όταν η τροχιά είναι στον ίδιο δρόμο ή υπάρχουν σφάλματα όπως το να εμφανίζει την τροχιά να κατευθύνεται μπρος - πίσω. Αυτό συμβαίνει επειδή τα δεδομένα που δόθηκαν ως εκπαίδευση είναι ανεπαρκή για να καλύψουν την ακριβή τοποθεσία. Στις περιπτώσεις που οι τροχιές συμπεριλαμβάνουν παράλληλους δρόμους, τα σφάλματα είναι ακόμη μεγαλύτερα. Αυτό συμβαίνει συχνά, λόγω της χαμηλής χρονικής σταθερότητας και της ευαισθησίας που έχουν τα MR δεδομένα. Αυτό μπορεί να διορθωθεί με την συλλογή περισσότερων δεδομένων σε περιοχές με πυκνά οδικά δίκτυα.

Συμπερασματικά, το σύστημα λειτουργεί καλά υπό διαφορετικές ιδιότητες των MR και επιτυγχάνει ως μέσο όρο σφάλμα 110 μέτρα με 80 μέτρα, ξεπερνώντας συστήματα τελευταίας τεχνολογίας που βασίζονται σε καταγραφή τροχιών (fingerprints).



Σχήμα 3.8 Γραφική Απεικόνιση του Αποτελέσματος του Αλγορίθμου [28]

3.2.5 Τεχνολογίες Βασισμένες σε Οχήματα

Τα τελευταία χρόνια έχουν αναπτυχθεί ιδιαίτερα τεχνολογίες που είναι βασισμένες σε οχήματα (technologies based on vehicles) [61]. Οι τεχνολογίες αυτές μπορούν να αναφερθούν και ως V2X ή Vehicle to everything. Πρόκειται για συστήματα επικοινωνίας μεταξύ ενός οχήματος με οποιοδήποτε αντικείμενο – οντότητα. Η επικοινωνία αυτή, αποσκοπεί στην εξασφάλιση ασφάλειας στους δρόμους και στην εξοικονόμηση ενέργειας. Το V2X μπορεί να κατηγοριοποιηθεί σε V2I (vehicle to infrastructure), V2N (vehicle to network), V2V (vehicle to vehicle), V2P (vehicle to pedestrian), V2D (vehicle to device) και V2G (vehicle to grid). Η τεχνολογία V2G αποτελεί ένα σύστημα στο οποίο οχήματα επικοινωνούν με το ηλεκτρικό δίκτυο. Συγκεκριμένα, ηλεκτρικά / υβριδικά οχήματα ή οχήματα με ηλεκτρικές μπαταρίες, αποθηκεύουν και εκκενώνουν ηλεκτρική ενέργεια που παράγεται από ανανεώσιμες πηγές ενέργειας όπως η ηλιακή και η αιολική. Τα οχήματα αυτά μπορούν να συνδεθούν με ηλεκτρικά δίκτυα, όπως αυτά που υπάρχουν σε σπίτια και σε περίπτωση απώλειας ενέργειας τα οχήματα μπορούν να παρέχουν πίσω ρεύμα στο δίκτυο. Η τεχνολογία V2N αποσκοπεί στην ευρεία επικοινωνία οχημάτων με τα κυψελοειδούς δίκτυα, όπως LTE, έτσι ώστε τα οχήματα να λαμβάνουν ειδοποιήσεις μετάδοσης σχετικά με ατυχήματα ή συμφόρηση στον δρόμο. Απ' την άλλη, η τεχνολογία V2V θα ήταν ιδιαίτερα χρήσιμη για την χαρτογράφηση οδικής κίνησης, αφού αποσκοπεί στη μεταφορά δεδομένων σε

πραγματικό χρόνο και γενικότερα στην ασύρματη επικοινωνία οχημάτων. Η επικοινωνία αυτή θα μπορούσε να χρησιμοποιηθεί και για την ενημέρωση κίνησης ή ατυχημάτων. Παρ' όλα αυτά, η προσέγγιση αυτή είναι ιδιαίτερα δύσκολη να καθοριστεί ως ένα προκαθορισμένο πρωτόκολλο οχημάτων, αφού υπάρχει ένα τεράστιο φάσμα από αυτοκινητοβιομηχανίες. Η τεχνολογία V2I βασίζεται στον συντονισμό της υποδομής, συλλέγοντας πληροφορίες σχετικά με την κυκλοφορία και τις οδικές συνθήκες, με απώτερο σκοπό τον συντονισμό συγκεκριμένων συμπεριφορών σε μία ομάδα οχημάτων. Επιτρέπει στην υποδομή να καθορίσει τις ιδανικές ταχύτητες και επιταχύνσεις των οχημάτων, όπως και των αποστάσεων μεταξύ των οχημάτων, βάσει των συνθηκών κυκλοφορίας. Αυτό θα μπορούσε να μειώσει σημαντικά, τόσο τα αυτοκινητιστικά ατυχήματα, όσο και την κατανάλωση καυσίμου. Η επικοινωνία οχημάτων με συσκευές (V2D) είναι ένας ιδιαίτερος τρόπος επικοινωνίας οχημάτων που συνίσταται στην ανταλλαγή πληροφοριών μεταξύ ενός οχήματος και κάθε ηλεκτρονικής συσκευής που μπορεί να συνδεθεί με το συγκεκριμένο όχημα. Η συνδεσιμότητα αυτή προσφέρει μία καλύτερη οδηγική εμπειρία, παρέχοντας στον οδηγό πληροφορίες σχετικά με τα γύρω οχήματα και την υποδομή, καθιστώντας την αλληλεπίδραση οδηγού – αυτοκινήτου απλούστερη. Τέλος, η επικοινωνία μέσω οχήματος και πεζού (V2P) προσπαθεί να εξασφαλίσει περισσότερη ασφάλεια των πεζών, μέσω της επικοινωνίας οχημάτων – πεζών. Για την εξασφάλιση της επικοινωνίας υπάρχουν ειδικές εφαρμογές που αποσκοπούν στη συγκεκριμένη λειτουργία.

Σε γενικές γραμμές, οι τεχνολογίες αυτές θα μπορούσαν να αποτελέσουν σημαντικό παράγοντα για την εύκολη χαρτογράφηση οδικής κυκλοφορίας. Αναμένεται η περαιτέρω ανάπτυξή τους στο μέλλον, αν και θα επιφέρουν διάφορα προβλήματα συμβατικότητας, λόγω των διαφόρων πρωτοκόλλων που ακολουθεί κάθε βιομηχανική εταιρεία δημιουργίας τεχνολογιών ή οχημάτων.

Κεφάλαιο 4

Αρχιτεκτονική TrafficWare

4.1 Μοντέλο Συστήματος	38
4.1.1 Σύστημα με Χρήση Δεδομένων Δικτύου	39
4.1.2 Σύστημα με Χρήση Πληθοπορισμού	39
4.1.3 Κύρια Επεξεργασία του Συστήματος	41
4.2 Θέματα Ιδιωτικότητας (Privacy)	45
4.3 Προβλήματα Ακρίβειας	46

Στο κεφάλαιο αυτό θα περιγραφεί η αρχιτεκτονική και η γενικότερη ιδέα του τελικού συστήματος περιλαμβάνοντας όλες τις εναλλακτικές λύσεις του προβλήματος. Αρχικά θα αναφερθούν τα δύο μοντέλα του συστήματος, και έπειτα, τα προαπαιτούμενα της υλοποίησης του. Στο τέλος του κεφαλαίου αυτού, θα επεξηγηθούν τα θέματα ιδιωτικότητας που έπρεπε να ληφθούν υπόψη κατά τη σχεδίαση του συστήματος, αλλά και τα προβλήματα ακρίβειας που μπορούν να προκύψουν κατά την υλοποίηση του συστήματος.

4.1 Μοντέλο Συστήματος

Το σύστημα γενικότερα βασίζεται σε τρία βασικά κριτήρια. Τα κριτήρια αυτά είναι η επίλυση του προβλήματος διατηρώντας την ιδιωτικότητα της θέσης των χρηστών, χρησιμοποιώντας την υπάρχουσα υποδομή, π.χ., κεραίες, και υλοποιώντας το σύστημα με ελάχιστο κόστος. Στο κεφάλαιο αυτό, θα αναφερθούν και θα επεξηγηθούν αναλυτικά τα δύο μοντέλα που το σύστημα θα μπορούσε να ακολουθήσει. Ακόμη, θα αναλυθούν τόσο τα θετικά, όσο και τα αρνητικά που ενέχει κάθε επιλογή ξεχωριστά.

4.1.1 Σύστημα με Χρήση Δεδομένων Δικτύου

Όπως αναφέρεται παρακάτω [Κεφάλαιο 5], κατά τη διάρκεια της σύνδεσης μιας συσκευής με μια κεραία γίνεται ανταλλαγή κάποιων δεδομένων. Τα δεδομένα αυτά ανταλλάσσονται μέσω ενός αρχείου που ονομάζεται PCHR και περιέχουν κάποια βασικά δεδομένα σχετικά με τη σύνδεση του χρήστη με την κεραία. Τα δεδομένα αυτά θα μπορούσαν εξίσου να ήταν δεδομένα σχετικά με ένα μήνυμα (SMS) ή μία κλήση που λαμβάνει ή στέλνει ένας χρήστης. Παρ' όλα αυτά, δεν γνωρίζουμε με σιγουριά κατά πόσο το αρχείο αυτό συμπεριλαμβάνει και την τοποθεσία του χρήστη. Εάν υποθέσουμε ότι το αρχείο όντως συμπεριλαμβάνει και την τοποθεσία του χρήστη τότε θα μπορούσε να υλοποιηθεί η ιδέα που θα αναφερθεί στην ενότητα αυτή.

Όπως φαίνεται και στο Σχήμα 4.1, η πρώτη ιδέα είναι η αποθήκευση των δεδομένων που τα PCHR αρχεία περιλαμβάνουν σε μια βάση δεδομένων. Ακολούθως, τα δεδομένα που υπάρχουν στη βάση δεδομένων θα επεξεργάζονται από το σύστημα με απώτερο σκοπό να υπολογίζεται η οδική κίνηση στους δρόμους. Η κύρια επεξεργασία του συστήματος, δηλαδή το πώς θα υπολογίζεται η οδική κίνηση, αναφέρεται πιο κάτω.



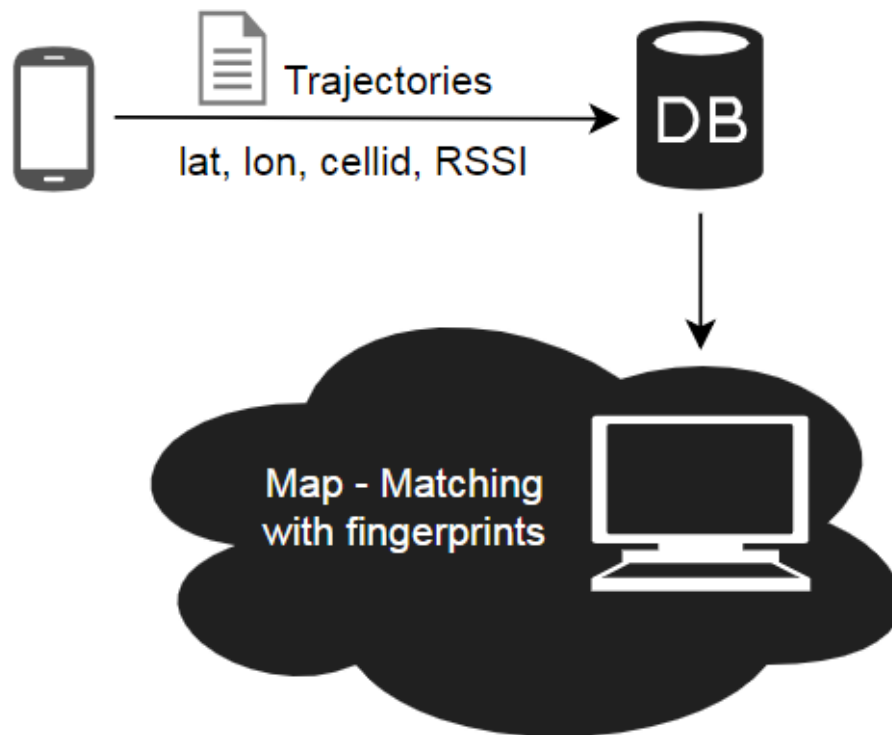
Σχήμα 4.1 Γραφική Απεικόνιση της Πρώτης Εναλλακτικής

4.1.2 Σύστημα με Χρήση Πληθοπορισμού

Η ιδανική λύση για την εύρεση οδικής κίνησης με λιγότερο κόστος, είναι με τη χρήση δεδομένων δικτύου, δηλαδή μέσω εύρεση της θέσης του χρήστη μέσω των

πληροφοριών που παρέχουν τα PCHR αρχεία. Παρ' όλα αυτά, λόγω της δυσκολίας υλοποίησης αυτής της ιδέας, εφόσον δεν είναι ευρέως γνωστή η δομή του PCHR αρχείου, προτάθηκε ένα δεύτερο μοντέλο που βασίζεται στη χρήση πληθοπορισμού. Στο δεύτερο μοντέλο, προτείνεται μια παρόμοια λύση με αυτή που υλοποιήθηκε στον Ctrack αλγόριθμο [22], όπου αρχικά γίνεται η συλλογή τροχιών (trajectories), έτσι ώστε να μπορεί να γίνει η αντιστοίχιση κάποιων γεωγραφικών μηκών και πλατών με κάποιες τιμές σήματος αλλά και με συγκεκριμένες κεραίες. Στην φάση αυτή, κατά τη διάρκεια των τροχιών συλλέγεται η GPS θέση του χρήστη, το σήμα που λαμβάνει στη συσκευή του, το id της κεραίας που η συσκευή είναι συνδεδεμένη αλλά και τα id των κεραίων που μπορεί η συσκευή να ανιχνεύσει. Έπειτα, δημιουργείται ένα πλέγμα με σκοπό τη σωστή αντιστοίχιση σε δρόμους βάσει της προηγούμενης θέσης του χρήστη και του ορίου ταχύτητας. Όλα όσα μόλις αναφέρθηκαν, μπορούν να ονομαστούν ως η φάση εκπαίδευσης του συστήματος για την επίτευξη του map - match προβλήματος.

Ιδανικά, ο αλγόριθμος αυτός, μετά την εκπαίδευση, όπως απεικονίζεται και στο Σχήμα 4.2, θα μπορεί να λαμβάνει σε πραγματικό χρόνο μετρήσεις από τις συσκευές και να γίνεται παράλληλα η αντιστοίχιση πάνω στον χάρτη. Οι μετρήσεις αυτές θα περιλαμβάνουν τις κεραίες που η συσκευή μπορεί να ανιχνεύσει αλλά και το σήμα που μπορεί να λάβει από αυτές.



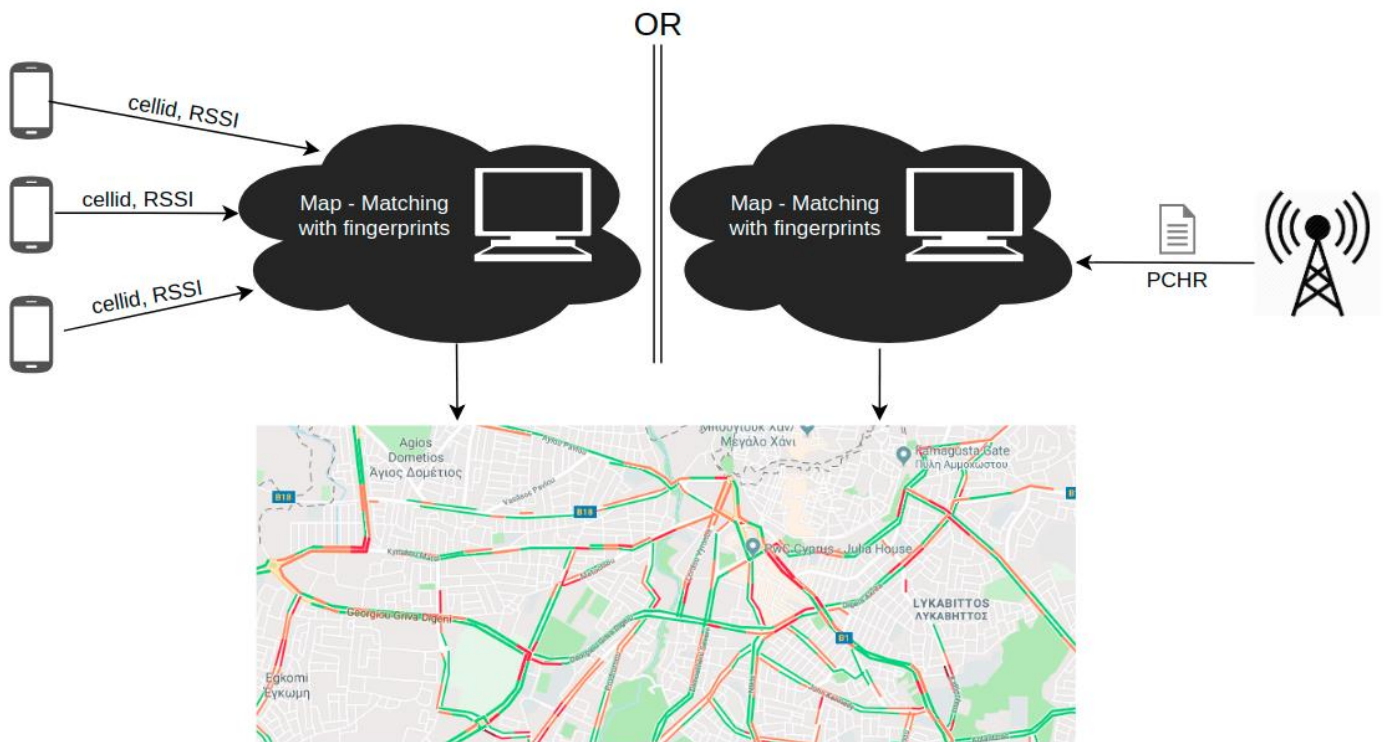
Σχήμα 4.2 Γραφική Απεικόνιση της Δεύτερης Εναλλακτικής

4.1.3 Κύρια Επεξεργασία του Συστήματος

Γενικότερα, το σύστημα έχει σκοπό να λαμβάνει ως είσοδο είτε PCHR αρχεία, είτε μετρήσεις που περιλαμβάνουν τις κεραίες που η συσκευή μπορεί να ανιχνεύσει μαζί με τις RSSI (Received Signal Strength Indication) τιμές που λαμβάνει από αυτές. Το σύστημα θα πρέπει έπειτα να αντιστοιχεί όλες τις εισόδους πάνω σε ένα χάρτη, και βάσει των δεδομένων που συνεχίζει να λαμβάνει σε πραγματικό χρόνο, να υπολογίζει τη διαδρομή που διανύουν οι χρήστες και την ταχύτητά τους. Το τελικό αποτέλεσμα του συστήματος θα πρέπει να είναι η απεικόνιση της κίνησης στους οδικούς δρόμους. Τη γενική αυτή ιδέα παρουσιάζει και το Σχήμα 4.3 που ακολουθεί.

Πιο συγκεκριμένα, το σύστημα εν τέλει θα μοιάζει όπως φαίνεται στο Σχήμα 4.4. Για να μπορέσει το σύστημα να επεξεργαστεί τα δεδομένα εισόδου παράλληλα και σε

πραγματικό χρόνο, θα χρησιμοποιεί το Apache Spark [Κεφάλαιο 5]. Στο σημείο αυτό, θα επιλέγονται (map - reduce) μόνο τα δεδομένα που αφορούν οδηγούς οχημάτων και όχι πεζούς. Για την επίτευξη αυτής της επιλογής δεδομένων, θα χρησιμοποιηθεί ένας κατάλληλος αλγόριθμος που μπορεί να ξεχωρίσει τις δύο ομάδες αυτές. Ως βάση δεδομένων θα χρησιμοποιηθεί ένα GIS [Κεφάλαιο 5] σύστημα, συγκεκριμένα PostgreSQL [Κεφάλαιο 5], για την καλύτερη αποθήκευση, διαχείριση και παρουσίαση γεωγραφικών δεδομένων.

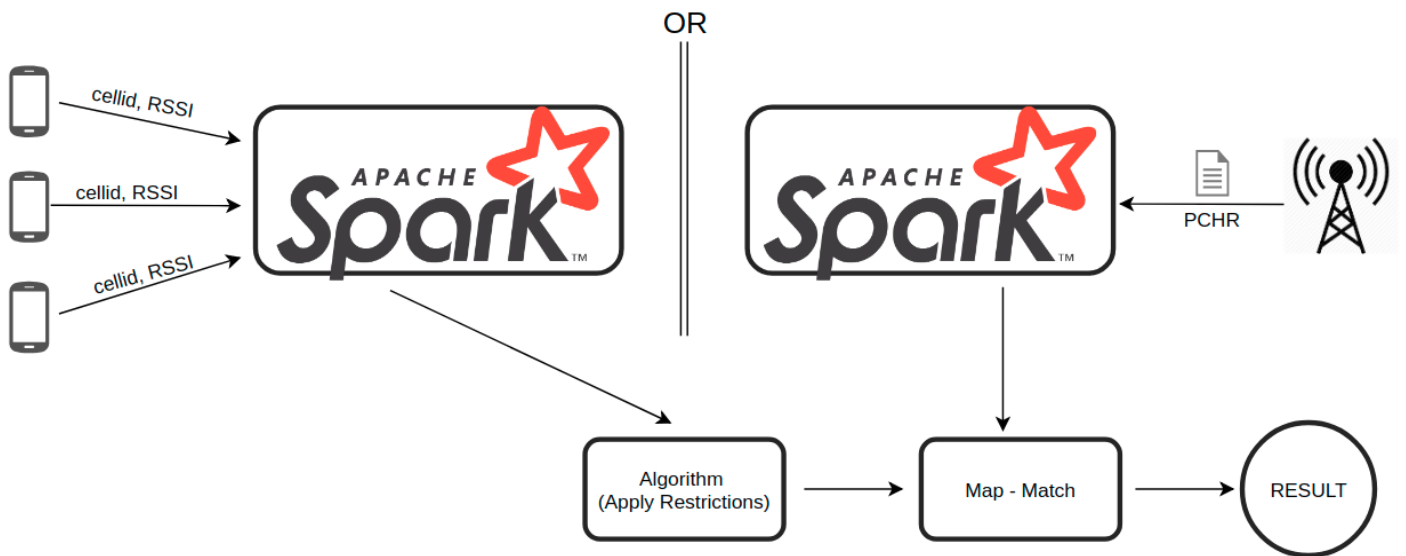


Σχήμα 4.3 Γραφική Απεικόνιση της Ολικής - Τελικής Αρχιτεκτονικής

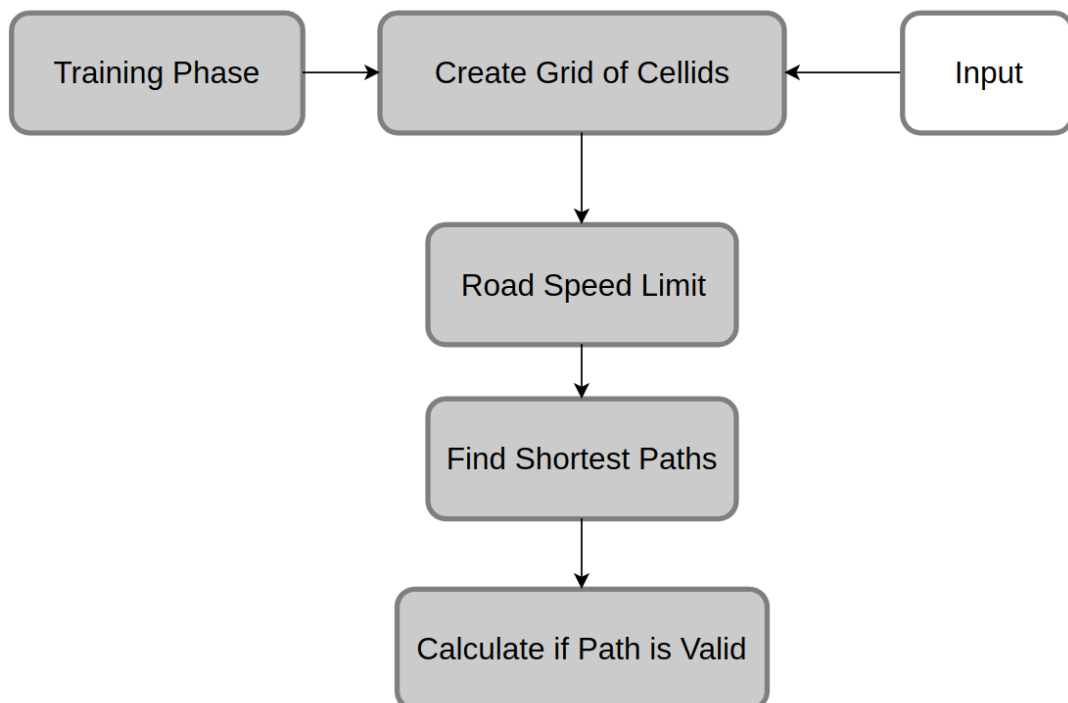
Για το πρώτο μοντέλο, που φαίνεται στα αριστερά του Σχήματος 4.4, το σύστημα απευθείας θα εκτελεί map - match αλγόριθμο, αφού θεωρούμε ότι τα PCHR αρχεία εμπερικλείουν την θέση του χρήστη. Έτσι, το σύστημα θα τοποθετεί απευθείας στον

χάρτη τη θέση του χρήστη και ανάλογα με την ταχύτητα που κινείται και τον αριθμό των χρηστών που υπάρχουν στον ίδιο δρόμο, θα υπολογίσει εάν υπάρχει κίνηση.

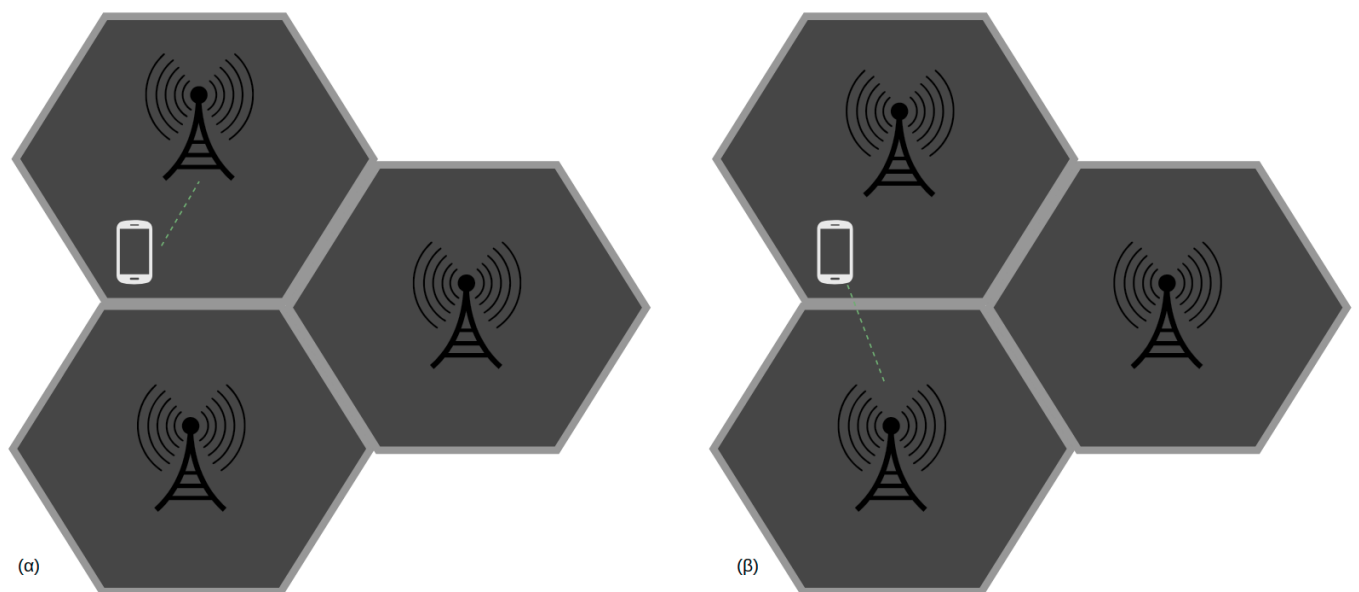
Απ' την άλλη, για το δεύτερο μοντέλο το σύστημα παράλληλα εφαρμόζει κάποιους περιορισμούς για να περιορίσει τον χώρο που θα μπορούσε ο χρήστης να βρίσκεται. Τους περιορισμούς αυτούς τους εφαρμόζει μέσω map - reduce μεθόδους, εκτελώντας τους περιορισμούς ως reduce μέθοδο. Ο πρώτος περιορισμός που θα εφαρμόζεται, είναι ο περιορισμός βάσει των cell id, που στην ουσία περιορίζει το πού μπορεί να βρίσκεται ο χρήστης μέσα σε μια περίμετρο. Η περίμετρος αυτή περιλαμβάνει την απόσταση της περιμέτρου, η οποία καλύπτεται από την κεραία που είναι συνδεδεμένη με την συσκευή του χρήστη, αλλά και τις περιμέτρους των κεραιών που η συσκευή ανιχνεύει. Αυτό απεικονίζεται και στο Σχήμα 4.6. Ο λόγος που ο περιορισμός γίνεται ανάμεσα όλων των κεραιών που ανιχνεύει αντί να υποθέτει ότι βρίσκεται στην περίμετρο που επικαλύπτει η κεραία που η συσκευή είναι συνδεδεμένη, είναι επειδή η συσκευή συνδέεται με την κεραία που έχει το πιο ισχυρό σήμα και όχι αυτή με την πιο κοντινή απόσταση. Για τον περιορισμό αυτό, το σύστημα θα έχει περάσει αρχικά από μία εκπαιδευτική φάση έτσι ώστε να φτιάξει ένα πλέγμα χωρίζοντας τα σημεία του χάρτη. Στα σημεία αυτά θα ανατίθενται ποιες κεραίες ανιχνεύονται από τα συγκεκριμένα σημεία. Ο δεύτερος περιορισμός που το σύστημα εφαρμόζει είναι βάσει του ορίου ταχύτητας. Συγκεκριμένα, το σύστημα προσπαθεί να περιορίσει τα σημεία που μπορεί ο χρήστης να κινήθηκε σε σχέση με το προηγούμενο σημείο που βρισκόταν. Η σύγκριση αυτή γίνεται μέσω του υπολογισμού της πιο σύντομης διαδρομής από το προηγούμενο γνωστό σημείο στο καινούργιο, βάσει του ορίου ταχύτητας. Στη συνέχεια, το σύστημα αντιστοιχεί τις διαδρομές στον χάρτη. Ο αλγόριθμος αυτός απεικονίζεται και στο Σχήμα 4.5. Τέλος, το σύστημα υπολογίζει ανάλογα με την ταχύτητα που κινείται ο κάθε χρήστης και τον αριθμό των χρηστών που υπάρχουν στον ίδιο δρόμο, εάν υπάρχει κίνηση. Το μοντέλο αυτό απεικονίζεται στο δεξιό μέρος του Σχήματος 4.4.



Σχήμα 4.4 Γραφική Απεικόνιση της Κόριας Επεξεργασίας



Σχήμα 4.5 Αλγόριθμος της Κόριας Επεξεργασίας



Σχήμα 4.6 Περίμετρος Μεταξύ Κεραιών

4.2 Θέματα Ιδιωτικότητας (Privacy)

Όπως αναφέρθηκε και στο κεφάλαιο 3, ένα από τα μεγαλύτερα θέματα που υπάρχουν στην πληροφορική, είναι τα θέματα ιδιωτικότητας, πόσο μάλλον σε ένα έργο που βασίζεται στην ανάκτηση δεδομένων από τους χρήστες. Τα δεδομένα αυτά είναι ιδιαίτερα ευαίσθητα, αφού το σύστημα χρειάζεται την τοποθεσία (location) του χρήστη. Ακόμη, στο πρώτο μοντέλο του συστήματος, τα PCHR αρχεία περιέχουν επιπλέον ευαίσθητα δεδομένα, αφού περιέχουν πληροφορίες όπως τι κλήσεις έχουν γίνει από τον χρήστη.

Μία λύση στο πρόβλημα αυτό, θα μπορούσε να είναι η προσθήκη σφάλματος (error) στα δεδομένα - αλλοίωση δεδομένων, ως αποτέλεσμα την επικράτηση ιδιωτικότητας. Παρ' όλα αυτά, η λύση αυτή μπορεί να δημιουργήσει προβλήματα, αφού η προσθήκη σφαλμάτων μπορεί να οδηγήσει σε λανθασμένα αποτελέσματα από το σύστημα.

Μια δεύτερη επιλογή για την επίλυση του προβλήματος αυτού, θα μπορούσε να ήταν η εφαρμογή του αλγορίθμου k-anonymity στα δεδομένα κατά τη διάρκεια της αντιστοίχισης χαρτών (map - matching), έτσι ώστε να μην είναι δυνατή η σύνταξη των ευαίσθητων δεδομένων μέσα σε στατιστικά που μπορούν να εφαρμοστούν αργότερα στο σύστημα (στατιστικά σχετικά με την κυκλοφορία στα οδικά δίκτυα). Ο αλγόριθμος αυτός μπορεί να προστατεύσει την ταυτότητα των χρηστών κρατώντας την κρυφή. Παρ' όλα αυτά, ο αλγόριθμος αποτυγχάνει να προστατεύσει δεδομένα σε περιπτώσεις όπου οι τιμές για ένα ή περισσότερα χαρακτηριστικά (attributes) σε μία ομάδα από αρχεία (records) είναι ίδια. Για την επίλυση του ελαττώματος αυτού, θα μπορούσε να χρησιμοποιηθούν επεκτάσεις του αλγορίθμου k - anonymity, όπως οι αλγόριθμοι t-Closeness [17] και I-diversity [19], οι οποίοι προσφέρουν μεγαλύτερη ασφάλεια. Η επίλυση αυτή, μπορεί να εφαρμοστεί και στα δύο μοντέλα συστήματος που έχουν προταθεί πιο πάνω.

4.3 Προβλήματα Ακρίβειας

Στο σημείο αυτό, είναι αξιοσημείωτο να σημειωθούν τα προβλήματα ακρίβειας που θα υπάρξουν στην προσπάθεια καταγραφής της οδικής κυκλοφορίας. Οποιοδήποτε μοντέλο συστήματος υλοποιηθεί, θα υπάρξουν προβλήματα ακρίβειας, μιας και οι συσκευές δεν συνδέονται πάντα με την κοντινότερη κεραία σε φυσικό επίπεδο. Συνδέονται, όμως, με την κεραία που παρέχει το ισχυρότερο σήμα, χωρίς να σημαίνει ότι η κεραία αυτή είναι σε όλες τις περιπτώσεις η κοντινότερη. Άρα οι αλγόριθμοι που θα υλοποιούνταν κατά τη διάρκεια της έρευνας, θα έβγαζαν λανθασμένα αποτελέσματα στις περιπτώσεις που οι κεραίες δεν είναι οι κοντινότερες σε φυσικό επίπεδο.

Επιπλέον, προκύπτει το πρόβλημα ότι σχεδόν είναι αδύνατη η καταγραφή οδικής κίνησης στους αυτοκινητόδρομους, αφού στα σημεία εκείνα οι χρήστες εξυπηρετούνται από τα Macro Cells [Κεφάλαιο 2] που μπορούν να καλύψουν αποστάσεις των 1 - 30km.

Τέλος, το μεγαλύτερο πρόβλημα ακρίβειας, είναι ότι το σύστημα θα έχει σφάλμα περίπου 50 - 1000m, λόγω του ότι βασίζεται στον εντοπισμό με την βοήθεια των

κεραιών. Υπό κανονικές συνθήκες, τα συστήματα που χρησιμοποιούν GPS για να εντοπίσουν την τοποθεσία του χρήστη, έχουν σφάλμα περίπου 10m.

Κεφάλαιο 5

Υποσύστημα Προβολής και Οπτικοποίησης

5.1 Τεχνικό Υπόβαθρο	48
5.1.1 Apache Spark	49
5.1.2 GIS	50
5.1.3 PostgreSQL	52
5.1.4 PostGIS	52
5.2 Δεδομένα Εισόδου	53
5.3 Πλατφόρμα TrafficWare	68

Στην ενότητα αυτή, αρχικά, θα υπάρξει μία σύντομη περιγραφή κάποιων βασικών τεχνολογιών που χρησιμοποιούνται ή θα μπορούσαν να χρησιμοποιηθούν από την πλατφόρμα, έπειτα θα περιγραφεί η πλατφόρμα που υλοποιήθηκε κατά τη διάρκεια της διπλωματικής, και τέλος, θα υπάρξει μια αναφορά και εξήγηση των δεδομένων που η ιστοσελίδα εμφανίζει. Η ιστοσελίδα που υπάρχει μέχρι στιγμής, μπορεί να λειτουργήσει ως ένα μικρό demo για πως το τελικό σύστημα θα πρέπει να μοιάζει.

5.1 Τεχνικό Υπόβαθρο

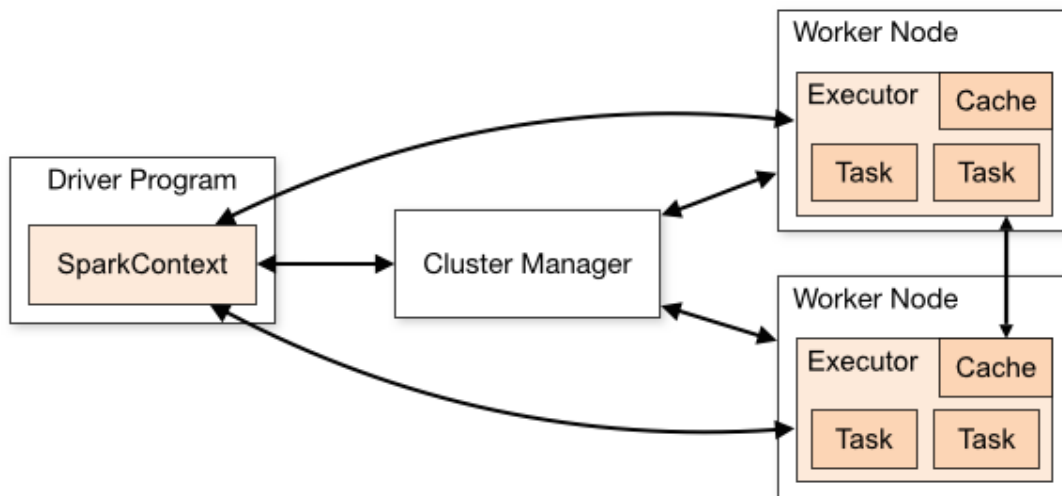
Σε αυτό το υποκεφάλαιο, θα περιγραφούν όλοι οι απαραίτητοι όροι, αλγόριθμοι και τεχνολογίες που χρειάζονται για την καλύτερη κατανόηση της λειτουργίας του συστήματος.

5.1.1 Apache Spark

Το Apache Spark [38] αποτελεί ένα ανοικτού κώδικα κατανεμημένο σύστημα για γενική και εύκολη χρήση. Συγκεκριμένα, είναι ένα cluster - computing framework. Χρησιμοποιείται για γρήγορη, ευέλικτη και παράλληλη επεξεργασία δεδομένων μεγάλης κλίμακας (big data). Το σύστημα είναι σχεδιασμένο έτσι ώστε να αξιοποιεί αποδοτικά την κύρια μνήμη (ram) για να μπορεί να ανταπεξέλθει και να ανταποκριθεί σε μεγάλο φόρτο εργασίας (workloads). Μπορεί να λειτουργήσει είτε τοπικά, είτε σε κατανεμημένες υπολογιστικές μηχανές. Αυτή η ενοποιημένη μηχανή ανάλυσης, μπορεί να επεξεργαστεί διάφορα αποθετηρία δεδομένων (data repositories), όπως το Hadoop Distributed File System (HDFS), NoSQL βάσεις δεδομένων αλλά και σχεσιακές βάσεις δεδομένων, όπως το Apache Hive.

Όπως αναφέρθηκε και πιο πάνω, το Spark υποστηρίζει την επεξεργασία εντός της μνήμης για να ενισχύσει την απόδοση μεγάλων εφαρμογών, ενώ παράλληλα μπορεί να εκτελεί συμβατική επεξεργασία δίσκων (conventional disk - based processing) όταν τα σύνολα δεδομένων είναι πολύ μεγάλα για να χωρέσουν στη διαθέσιμη μνήμη του συστήματος. Επιπρόσθετα, επιτρέπει την επεξεργασία σε πραγματικό χρόνο (real - time processing).

Ο πυρήνας του Spark χρησιμοποιεί ως βασικό τύπο δεδομένων το εύκαμπτο κατανεμημένο σύνολο δεδομένων (resilient distributed dataset - RDD). Συγκεντρώνει τα δεδομένα και τα χωρίζει σε ένα server cluster / master, το οποίο μπορεί στη συνέχεια να μεταφερθεί σε διαφορετικό αποθηκευτικό χώρο. Ο χρήστης δεν χρειάζεται να καθορίζει το πού αποστέλλονται συγκεκριμένα αρχεία ή ποιοι υπολογιστικού πόροι χρησιμοποιούνται για την αποθήκευση ή την ανάκτηση αρχείων. Η λογική αυτή απεικονίζεται και στο Σχήμα 5.1.



Σχήμα 5.1 Apache Spark Αρχιτεκτονική [38]

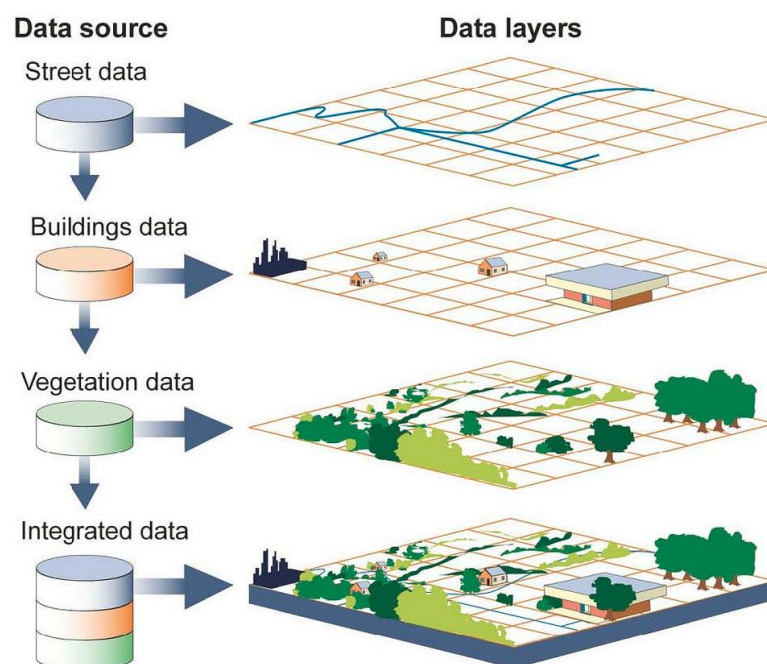
5.1.2 GIS

Το σύστημα GIS [39] (Geographic Information System) είναι ένα γεωγραφικό σύστημα πληροφοριών, σχεδιασμένο για να συλλαμβάνει, να αποθηκεύει, να διαχειρίζεται και να παρουσιάζει χωρικά ή γεωγραφικά δεδομένα (spatial data). Οι εφαρμογές GIS είναι εργαλεία που επιτρέπουν στους χρήστες να δημιουργήσουν επερωτήματα διαδραστικά, να αναλύσουν χωρικές ή αλλιώς γεωγραφικά συνδεδεμένες (geographically - referenced) πληροφορίες, να επεξεργαστούν δεδομένα σε χάρτες και να παρουσιάσουν τα αποτελέσματα όλων αυτών των λειτουργιών.

Τα δεδομένα αυτά, μπορούν να αναπαρασταθούν ως πραγματικά αντικείμενα, π.χ., δρόμοι, δέντρα, οδοί, υψόμετρο, κτλ. Τα αντικείμενα μπορούν να διαχωριστούν σε δύο ομάδες, στα διακριτά αντικείμενα, όπως ένα σπίτι, ή το ποσό της βροχόπτωσης, κτλ. Παραδοσιακά, υπάρχουν δύο μέθοδοι που χρησιμοποιούνται για την αποθήκευση δεδομένων, οι εικόνες ράστερ (raster images) και τα διανύσματα (vectors). Ακόμη, υπάρχουν τα σημεία (points), οι γραμμές, και τα πολύγωνα, που χρησιμοποιούνται ως

χαρακτηριστικά αντιστοίχισης θέσης. Αυτό ακριβώς απεικονίζεται και στο Σχήμα 5.2, δείχνοντας τα διάφορα επίπεδα δεδομένων που το GIS διαχειρίζεται.

Συγκεκριμένα, μπορούν να αποτυπώσουν χωρικά (spatial) δεδομένα σε γεωγραφικό, χαρτογραφικό ή και καρτεσιανό σύστημα συντεταγμένων. Βασικό χαρακτηριστικό του συστήματος είναι ότι τα δεδομένα συνδέονται με περιγραφικά δεδομένα, π.χ., πληθυσμό. Η τεχνολογία που χρησιμοποιείται για τη λειτουργία αυτή, βασίζεται είτε στο σχεσιακό μοντέλο δεδομένων, είτε στο αντικειμενοστραφές (object - oriented). Στο σχεσιακό μοντέλο τα περιγραφικά δεδομένα βρίσκονται αρχικά σε ένα πίνακα και αργότερα συσχετίζονται με τα χωρικά δεδομένα μέσω κάποιων μοναδικών τιμών που είναι κοινές και στα δύο είδη δεδομένων (foreign keys). Απ' την άλλη, στο αντικειμενοστραφές μοντέλο και τα δύο είδη δεδομένων μετατρέπονται σε ένα ομαδοποιημένο αντικείμενο (object), έτσι ώστε το κάθε αντικείμενο να έχει ως χαρακτηριστικά του μια περιγραφή και ένα γεωγραφικό μήκος και πλάτος.



Σχήμα 5.2 Επίπεδα Δεδομένων ενός GIS Συστήματος [40]

Τέλος, το GIS μπορεί να οριστεί ως διάφορες τεχνολογίες, διαδικασίες, τεχνικές και μεθόδους. Είναι συνδεδεμένο με πολλές λειτουργίες και έχει πολλές εφαρμογές σχετικά με τη μηχανική, τον προγραμματισμό, τις τηλεπικοινωνίες, κτλ. Για τον λόγο αυτό, οι GIS εφαρμογές και εντοπισμού θέσης μπορούν να αποτελέσουν θεμέλιο για πολλές υπηρεσίες που βασίζονται στην τοποθεσία, ανάλυση και απεικόνιση.

5.1.3 PostgreSQL

Η PostgreSQL [41] ή αλλιώς Postgres, είναι ένα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων (RDBMS ή αλλιώς relational database management system) ανοικτού κώδικα που δίνει έμφαση στην επεκτατικότητα και την τήρηση των προτύπων. Μπορεί να διαχειριστεί workloads που κυμαίνονται από εφαρμογές μιας μηχανής έως Web Services ή αποθήκευση δεδομένων από πολλούς ταυτόχρονους χρήστες (multi - version control). Έχει σχεδιαστεί για Unix συστήματα, και υπάρχει ως προκαθορισμένη (default) βάση στο Mac Os Server.

Προσφέρει υποστήριξη για λειτουργίες RDBMS, όπως ενημερώσεις (updates), materialized views, triggers, foreign keys, functions και stored procedures.

5.1.4 PostGIS

Το PostGIS [42] είναι ένα λογισμικό ανοικτού κώδικα που προσθέτει υποστήριξη για γεωγραφικά αντικείμενα (geographic objects) στη PostgreSQL. Το PostGIS ακολουθεί απλά χαρακτηριστικά SQL και έχει υλοποιηθεί ως επέκταση της PostgreSQL.

Το λογισμικό έχει ως χαρακτηριστικά τύπους γεωμετρίας για σημεία (points), γραμμές, πολύγωνα, πολλαπλά σημεία (multipoints) και γεωμετρίες. Ακόμη, παρέχει χωρικούς φορείς (spatial operators) για τον προσδιορισμό γεωχωρικών (geospatial) μετρήσεων όπως περιοχή (area), απόσταση, μήκος και περίμετρος. Υπάρχουν και spatial operators για τον προσδιορισμό γεωχωρικών λειτουργιών, όπως συνένωση (union), διαφορά, συμμετρική διαφορά και buffers. Γενικότερα, το σύστημα βασίζεται σε γεωμετρίες ελαφρού βάρους και ευρετήρια βελτιστοποιημένα για τη μείωση του αποτυπώματος

(footprint) δίσκου και μνήμης. Η χρήση γεωμετριών ελαφρού βάρους βοηθά τους servers να αυξήσουν την ποσότητα των δεδομένων που μεταφέρονται από την τους δίσκους στην μνήμη RAM, βελτιώνοντας σημαντικά την απόδοση επερωτημάτων (query performance).

Κάποια από τα συστήματα που χρησιμοποιούν PostGIS είναι το ArcGIS, CartoDB, GeoServer, OpenStreetMap, QGIS, κτλ. Όλα αυτά τα συστήματα το χρησιμοποιούν για την αποθήκευση και διαχείριση μεγάλων γεωγραφικών δεδομένων.

5.2 Δεδομένα Συστήματος

Στην παρούσα ενότητα, αναφέρονται όλα τα σύνολα και είδη δεδομένων που μελετήθηκαν κατά τη διάρκεια της διπλωματικής αυτής. Η μελέτη των δεδομένων αυτών έγινε με την ελπίδα αξιοποίησής τους ως προς την εύρεση οδικής κίνησης μέσω των κεραιών. Για τον λόγο αυτό, μελετήθηκε το αρχείο PCHR, το οποίο είναι το στοιχείο κλειδί για την υλοποίηση συστήματος χρησιμοποιώντας δεδομένα δικτύου [Κεφάλαιο 4], αλλά και άλλα είδη δεδομένων όπως Opencellid, MLS, Wigle και κάποια κυβερνητικά δεδομένα. Τα είδη δεδομένων που μόλις αναφέρθηκαν μελετήθηκαν είτε για την εύρεση των θέσεων των κεραιών, είτε για την εύρεση τροχιών (trajectories) χρηστών. Όλα τα δεδομένα που μελετήθηκαν, εκτός του PCHR, είναι ανοικτά προς όλους.

PCHR

Με τη δημιουργία κινητών δικτύων και τηλεφώνων δημιουργήθηκε η ανάγκη παροχής τους. Για τον σκοπό αυτό, δημιουργήθηκαν οι τηλεπικοινωνιακοί πάροχοι. Για την παροχή των υπηρεσιών αυτών, οι πάροχοι απέκτησαν την ανάγκη να γνωρίζουν την απόδοση αντενών - κεραιών, έτσι ώστε να γνωρίζουν εκ των προτέρων την ύπαρξη προβλημάτων που μπορούν να προκύψουν. Ακόμη, δημιουργήθηκε η ανάγκη να γνωρίζουν εάν υπάρχουν χρήστες που έχουν συχνές αποσυνδέσεις της συσκευής τους από το δίκτυο. Για την μελέτη της ανάγκης αυτής δημιουργήθηκαν τα CHR αρχεία. Τα PCHR (Performance Call Trace History) logfiles είναι CHR (Call Trace history) αρχεία

τα οποία είναι υπεύθυνα για την αποθήκευση της απόδοσης της κεραίας και της συσκευής του χρήστη κατά τη διάρκεια της σύνδεσής τους. Συγκεκριμένα, τα αρχεία αυτά αποθηκεύουν χαρακτηριστικά όπως blocked / dropped, (cell tower) και throughput αναλύσεις. Η ανάλυση block / dropped είναι συγκεκριμένα για την ανάλυση των κλήσεων που σφραγίζονται - μπλοκάρονται ή χάνεται η σύνδεση τους. Η ανάλυση των cell και throughput, αποτελεί μια πιο γενική ανάλυση σχετικά με την κεραία και το σήμα που παρέχει. Επιπλέον, το αρχείο αυτό παρέχει και λίγα δεδομένα σχετικά με τον χρήστη (π.χ., αριθμός τηλεφώνου).

Τα αρχεία αυτά έχουν δυαδική (binary) μορφή, έχοντας συνήθως μέγεθος γύρω στα 10.5MB και έχουν τις εξής ονομασίες:

- 'RNC0101_02Log20110404000002_20110404010109.log.zip', ισχύει για την πρώτη έκδοχή έως και την έκδοχή R11,
- 'BSC0121_[UHR]02Log20111010231746(DST)_20111010232743(DST).log.zip', όπου 'DST' αντιστοιχεί για τον προορισμό (Destination) του αρχείου και ισχύει για την έκδοση R12.
- 'BSC0121_[PCHR]02Log20111010231746(DST)_20111010232743(DST).log.zip' όπου 'DST' αντιστοιχεί για τον προορισμό (Destination) του αρχείου και ισχύει για την έκδοση R13 και R14.

Δυστυχώς η μορφή των PCHR δεν είναι ευρέως γνωστή, γι' αυτό η επεξεργασία και η ανάλυση των αρχείων αυτών μπορεί να γίνει εφικτή μόνο με τη βοήθεια συγκεκριμένων εργαλείων (π.χ., Huawei Omstar, Aexio Xeus Pro, κτλ). Ακόμη, όπως φαίνεται και στο Σχήμα 5.3 η προσπάθεια αποκωδικοποίησης του αρχείου μέσω της εντολής hexdump είναι ανεπιτυχής.

Όπως αναφέρθηκε πιο πάνω, γνωρίζουμε ότι το PCHR αρχείο περιέχει πληροφορίες σχετικά με την σύνδεση της συσκευής του χρήστη με την κεραία, αλλά και πληροφορίες σχετικά με την κεραία που η συσκευή είναι συνδεδεμένη. Έτσι, μπορούμε να υποθέσουμε ότι το PCHR αρχείο μπορεί να περιέχει κάποιες από τις πληροφορίες που φαίνονται στο Σχήμα 5.4. Συγκεκριμένα, στο Σχήμα 5.4 απεικονίζεται το Σχήμα (schema) του αρχείου που αντιστοιχεί για κάθε BSC αρχείο, όπου περιέχει πληροφορίες για κάθε σταθμό βάσης (base station).

00000000	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00
*			
00000100	54 54 5f 5f 54 4f 4f 4c	4d 42 53 43 00 00 00 00	TT__TOOLMBSC....
00000110	00 00 00 00 56 39 30 30	52 30 31 34 00 00 00 00V900R014....
00000120	43 30 30 53 50 48 35 35	32 00 00 00 12 34 00 00	C00SPH552....4..
00000130	00 00 00 00 00 00 00 00	07 df 06 0f 4d 42 53 43MBSC
00000140	20 43 48 52 00 00 00 00	00 00 00 00 00 00 00 00	CHR.....
00000150	00 00 00 00 00 00 00 00	00 00 00 00 00 00 01 5f
00000160	00 00 00 00 ab ab 00 06	00 00 02 27 50 a8 08 08 'P...
00000170	38 01 00 00 0d 0f 2f 0b	03 10 00 0d 0f 06 0f 0d	8...../.....
00000180	0d 0c 5e 3e 90 a9 ea 03	fc 11 00 53 fc 8c 01 b3	..^>.....S....
00000190	08 7a d3 7e 82 10 10 48	04 50 79 02 82 10 10 00	.z.~...H.Py....
000001a0	00 01 31 00 00 06 ff 04	00 03 fc 05 02 c9 00 00	..1.....
000001b0	00 e0 00 98 03 60 00 18	06 01 00 9b 28 01 00 10 '(...
000001c0	15 00 00 00 00 82 01 e0	03 e9 a0 34 22 fe 88 404"...@
000001d0	0a 00 40 01 03 e9 a1 91	00 fe 00 50 a8 0d 72 13	..@.....P...r.
000001e0	00 41 ff 30 60 05 05 00	00 37 ed 00 00 38 68 00	.A.0`....7...8h.
000001f0	00 37 ea 00 73 02 c4 03	02 02 06 03 04 00 f4 24	.7..s.....\$
00000200	00 00 1f 40 00 01 40 6f	40 00 00 3e 80 07 00 00	...@...@o@...>...
00000210	30 00 02 03 e9 a0 b5 03	e9 a0 34 03 e9 a0 b5 03	0.....4.....
00000220	e9 a0 b5 15 00 36 c0 00	00 38 8d 00 00 00 17 076...8.....
00000230	02 03 e9 a0 b5 23 22 44	80 03 03 e9 a0 34 0d 17#"D.....4..
00000240	06 91 fe 02 03 e9 c8 70	0d 16 03 e9 c9 eb 11 18p.....
00000250	01 80 03 e9 a0 34 06 91	fe 0f 00 15 15 00 33 c04.....3..
00000260	00 00 37 80 00 04 00 1e	0b 01 03 e9 a0 b5 1d 1d	..7.....
00000270	44 80 03 03 03 e9 a0 34	1e 1e 03 e9 c8 70 0d 16	D.....4.....p..
00000280	03 e9 c9 eb 11 18 01 80	03 e9 a0 34 06 91 fe 0f4.....
00000290	00 00 1a 00 9a 25 25 00	00 00 00 00 00 00 00 01%%.....
000002a0	01 00 00 00 00 01 00 00	00 01 00 00 00 01 00 00
000002b0	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00

Σχήμα 5.3 Hexdump πρώτες 20 γραμμές του PCHR αρχείου

```

<?xml version="1.0" encoding="iso-8859-1"?>
<MOTree>
  <MO className="BSC6000NE" fdn=".3221229568.3221233664.3221282818">
    <attr name="fdn">.3221229568.3221233664.3221282818</attr>
    <attr name="IP">10.4.137.21</attr>
    <attr name="MOIndex">2</attr>
    <attr name="className">BSC6000NE</attr>
    <attr name="latitude">255.0</attr>
    <attr name="locationName">Arena</attr>
    <attr name="longitude">194.0</attr>
    <attr name="matchVersion">iManagerM2000_BSC6000_MATCH_ENG_V200R006C01B624</attr>
    <attr name="name">BSCNTB31</attr>
    <attr name="neID">2</attr>
    <attr name="neType">BSC6000</attr>
    <attr name="neVersion">BSC6000V900R008C12</attr>
    <attr name="realLatitude">0DN</attr>
    <attr name="realLongitude">0DE</attr>
    <attr name="vendorName">Huawei</attr>
    <MO className="BSC6000BCGrp" fdn=".3221229568.3221233664.3221282818.3221417986">
      <attr name="fdn">.3221229568.3221233664.3221282818.3221417986</attr>
      <attr name="BSCIndex">0</attr>
      <attr name="GrpIndex">0</attr>
      <attr name="MOIndex">2_0_0</attr>
      <attr name="className">BSC6000BCGrp</attr>
      <attr name="name">CBCGrp</attr>
      <attr name="neID">2</attr>
    </MO>.....
  </MO>.....

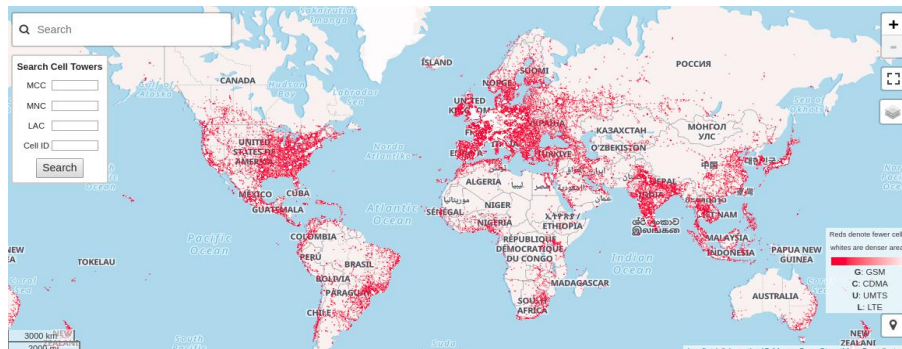
```

Σχήμα 5.4 Hexdump πρώτες 20 γραμμές του PCHR αρχείου [43]

Opencellid και MLS

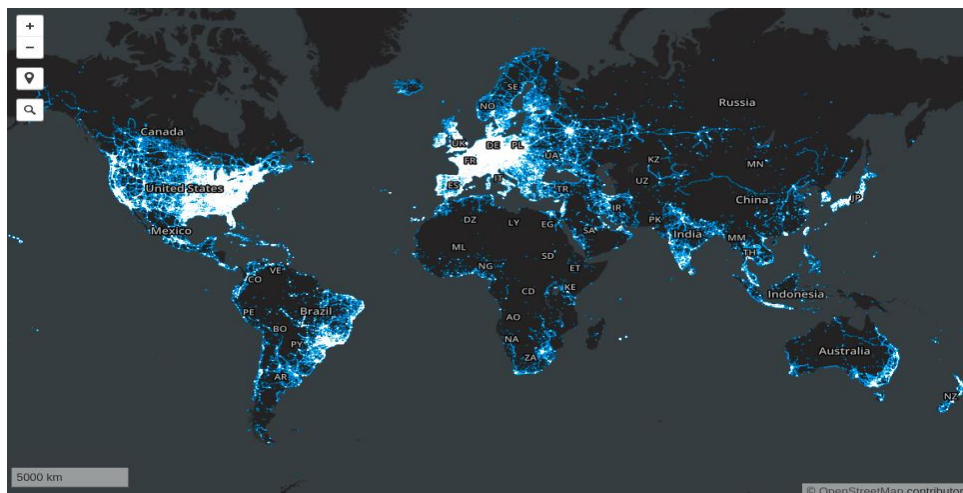
Κάποια από τα σύνολα δεδομένων που μελετήθηκαν κατά τη διάρκεια της διπλωματικής αυτής, ήταν τα Opencellid [44] και MLS [45]. Όπως φαίνεται και στα Σχήματα 5.5 και 5.6, τα δεδομένα έχουν συλλεχθεί σε παγκόσμιο επίπεδο.

Το OpenCellID project είναι ένα από τα μεγαλύτερα προγράμματα συνεργατικής κοινότητας παγκοσμίως που συλλέγει θέσεις GPS κυψελοειδών πύργων, που χρησιμοποιούνται δωρεάν, για πλήθος εμπορικών και ιδιωτικών σκοπών. Δημιουργήθηκε κυρίως για να χρησιμεύσει ως πηγή δεδομένων για τον εντοπισμό του GSM (localization). Η βάση δεδομένων τους είναι διαθέσιμη προς όλους, με σκοπό την προώθηση της δωρεάν χρήσης και αναδιανομής των δεδομένων.



Σχήμα 5.5 OpenCellId Δεδομένα [44]

Το MLS (Mozilla Location Service) project είναι μια ανοιχτή υπηρεσία εντοπισμού θέσης (localization) από την Mozilla, η οποία επιτρέπει στις συσκευές να καθορίζουν την τοποθεσία τους βάσει την υποδομή δικτύων όπως κεραίες (Cell Towers), Bluetooth και σημεία πρόσβασης WiFi. Η υπηρεσία αυτή, χρησιμοποιείται σε διάφορα δορυφορικά συστήματα πλοήγησης. Βασίζεται σε μια κοινότητα που περιλαμβάνει άτομα που συλλέγουν δεδομένα, προγραμματιστές που χρησιμοποιούν την υπηρεσία, και οργανισμούς που μοιράζονται πληροφορίες σχετικά με την υποδομή τους.



Σχήμα 5.6 MLS Δεδομένα [45]

Και οι δύο αυτές πηγές δεδομένων έχουν την ίδια μορφή αρχείων. Τα αρχεία αυτά περιέχουν τα εξής δεδομένα:

Radio, MCC, Net, Area, Cell, Unit, Lat, Lon, Range, Samples, Changeable, Created, Updated, AverageSignal. Το Radio περιέχει τον τύπο του network, π.χ., GSM, UMTS, LTE, κτλ. Το MCC περιέχει το Mobile Country Code, που στην ουσία είναι ένας αριθμός που ανατίθεται ανά χώρα. Το Net, ή αλλιώς MNC (Mobile Network Code), είναι ο αριθμός που ανατίθεται σε κάθε πάροχο τηλεπικοινωνιακών δικτύων. Το Area για GSM και UMTS δίκτυα είναι το LAC (Location Area Code) και για LTE δίκτυα είναι το TAC (Tracking Area Code). Το Cell για τα GSM και LTE δίκτυα είναι το Cell id (το id που αντιστοιχεί σε κάθε κεραία ή κάποιο τομέα μιας κεραίας), για τα UMTS δίκτυα είναι το UTRAN Cell id, το οποίο είναι η συνένωση των δύο bytes του cell id. Το Unit για GSM δίκτυα είναι κενό, για UMTS δίκτυα είναι το PSC (Primary Scrambling Code) και για τα LTE δίκτυα είναι το PCI (Physical Cell Id). Το lat και lon είναι τα γεωγραφικά μήκη και πλάτη. Το range είναι η εκτίμηση της περιοχής ραδιοσυχνοτήτων σε μέτρα, πρόκειται για ένα υπολογισμό σχετικά με το πόσο μεγάλη είναι η κάθε περιοχή κεραιών ως ακτίνα γύρω από την εκτιμώμενη θέση. Το range βασίζεται σε παρατηρήσεις ή μια έγκυρη πηγή γνώσης. Το Samples είναι ο συνολικός αριθμός εγγραφών - μετρήσεων που χρησιμοποιήθηκαν για να υπολογιστεί η εκτιμώμενη θέση (position) μιας κεραίας, το εύρος (range) που καλύπτει αλλά και τη μέση τιμή σήματος (averageSignal) που παρέχει. Το Changeable πρόκειται για το αν η συγκεκριμένη κεραία - record είναι μια εκτίμηση θέσης βασισμένη σε μετρήσεις και επομένως υπόκειται σε αλλαγές στο μέλλον, ή είναι μια ακριβής θέση που εισάγεται από μια έγκυρη πηγή γνώσης. Το Created και Updated είναι οι ημερομηνίες (timestamps) σχετικά με το πότε το συγκεκριμένο record πρωτοδημιουργήθηκε και πότε ανανεώθηκε τελευταία φορά αντίστοιχα. Τέλος, το AverageSignal είναι η μέση ισχύς σήματος από όλες τις παρατηρήσεις για την συγκεκριμένη κεραία και έχει χρησιμοποιηθεί για την εκτίμηση θέσης βάσει της ποιότητας του σήματος.

Κατά τη διάρκεια της διπλωματικής ανακτήθηκαν όλες οι κεραίες που αφορούν την Κύπρο. Τα δεδομένα αυτά μαζί είναι συνολικά 54992 εγγραφές των 9376KB. Συγκεκριμένα από την OpenCellId βάση δεδομένων ανακτήθηκαν 19427 εγγραφές των

3312KB, και από την MLS βάση δεδομένων ανακτήθηκαν 35565 εγγραφές των 6064KB που αναφέρονται σε κυπριακές κεραίες.

Wigle.net

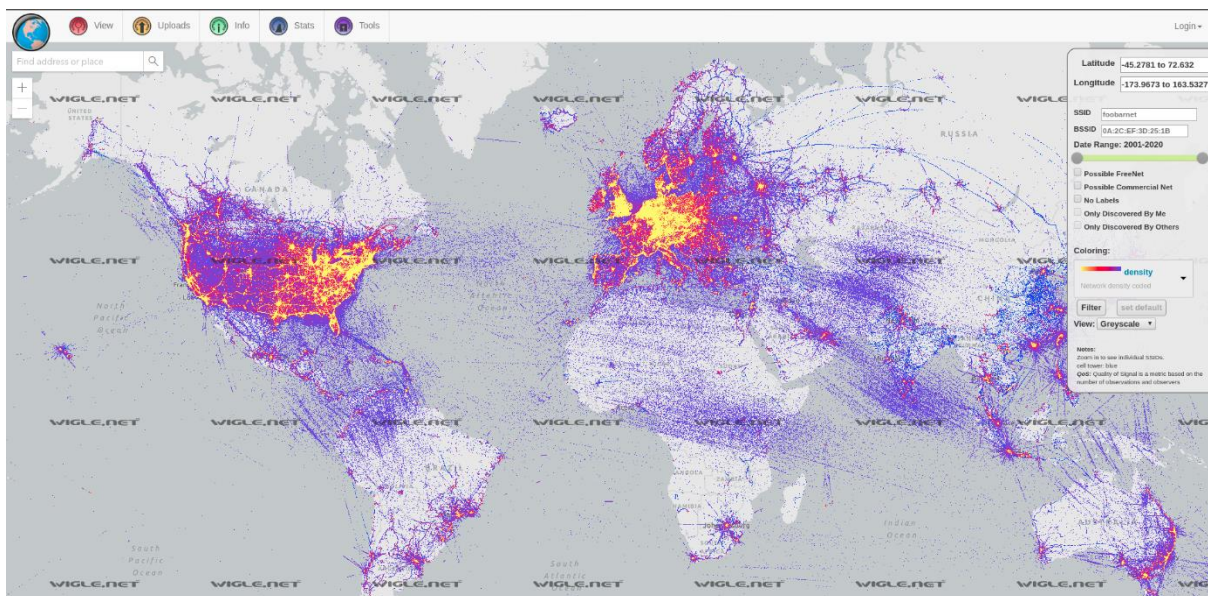
Μία ακόμη πηγή δεδομένων που έχει μελετηθεί κατά τη διάρκεια της διπλωματικής αυτής, είναι τα Wigle.net [46] (Wireless Geographic Logging Engine) δεδομένα. Το έργο αυτό συγκεντρώνει τις τοποθεσίες και τις πληροφορίες των ασύρματων δικτύων παγκοσμίως σε μια κεντρική βάση δεδομένων. Συγκεκριμένα, συλλέγουν δεδομένα σχετικά με WiFi Access Points και Cell Towers. Το έργο αυτό ακόμη, παρέχει API για την εκτέλεση επερωτημάτων (queries) και την ενημέρωση δεδομένων στη βάση δεδομένων. Στο Σχήμα 5.7 απεικονίζεται σε χάρτη η βάση δεδομένων του συστήματος.

Το API τους προσφέρει τις υπολογιζόμενες (approximate) θέσεις των σημείων πρόσβασης WiFi και τις υπολογιζόμενες θέσεις των κεραιών (Cell Towers). Επιπρόσθετα, διαθέτει διάφορες πληροφορίες σχετικά με τα ασύρματα αυτά δίκτυα. Για παράδειγμα, για τις υπολογιζόμενες θέσεις των κεραιών ή των σημείων πρόσβασης WiFi, το API επιστρέφει τα εξής δεδομένα:

- Trilat και Trilong, που είναι το υπολογιζόμενο γεωγραφικό μήκος και πλάτος αντίστοιχα της κεραίας / σημείου πρόσβασης WiFi.
- SSID (Service Set Identifier), που είναι το όνομα του WiFi ή το όνομα του τηλεπικοινωνιακού παρόχου.
- QOS (Quality of Service), που μπορεί να πάρει την τιμή μεταξύ 0 και 7.
- FirstTime, LastTime είναι οι ημερομηνίες (timestamps) της πρώτης και της τελευταίας φοράς αντίστοιχα που κάποιος χρήστης πήρε μέτρηση σχετικά με τη συγκεκριμένη κεραία.
- Lastupdt είναι η τελευταία φορά που ανανεώθηκε η υπολογιζόμενη θέση της κεραίας.
- House Number, είναι η διεύθυνση ή ο ταχυδρομικός κώδικας που βρίσκεται τοποθετημένη η κεραία ή το WiFi. Στην περίπτωση της κεραίας είναι σχεδόν πάντα κενό.

- Road, City, Region, Country, είναι η οδός, η πόλη, η περιοχή και η χώρα αντίστοιχα που είναι τοποθετημένη η κεραία ή το WiFi.
- Id, είναι ο μοναδικός αριθμός που αντιστοιχεί σε κάθε κεραία ή σε κάθε τμήμα (sector) της κεραίας ή στο σημείο πρόσβασης WiFi.
- Attributes, είναι ο τύπος δικτύου που παρέχει η κεραία / WiFi μαζί με τα αρχικά της χώρας που είναι τοποθετημένη. Π.χ., UMTS;cy
- GenType, είναι ο τύπος του δικτύου, π.χ., GSM / UMTS / LTE ή WiFi.

Κατά την ανάκτηση των Wigle δεδομένων, ανακτήθηκαν όλες οι κεραίες που αφορούν μόνο την Κύπρο. Τα δεδομένα αυτά είναι συνολικά 4463 εγγραφές των 1352KB.



Σχήμα 5.7 Wigle.net Δεδομένα [46]

Επιπλέον, στα πλαίσια της παρούσας διπλωματικής, χρειάστηκε να ανακτηθούν τα ακατέργαστα (raw) δεδομένα που η wigle βάση παρέχει. Συγκεκριμένα, τα δεδομένα αυτά είναι όλες οι μετρήσεις που συλλεχθήκανε από χρήστες σχετικά με μια κεραία. Για να πραγματοποιηθούν οι ανακτήσεις αυτές έπρεπε να γίνουν επερωτήματα βάσει του id των κεραιών. Η ανάκτηση των δεδομένων έχει γίνει μέσω του εξής script:

```
while read line; do
    wget -x --load-cookies cookies.txt $line;
done < "$1"
```

Το bash script αυτό, παίρνει ως παράμετρο ένα αρχείο που περιέχει urls με την εξής δομή:

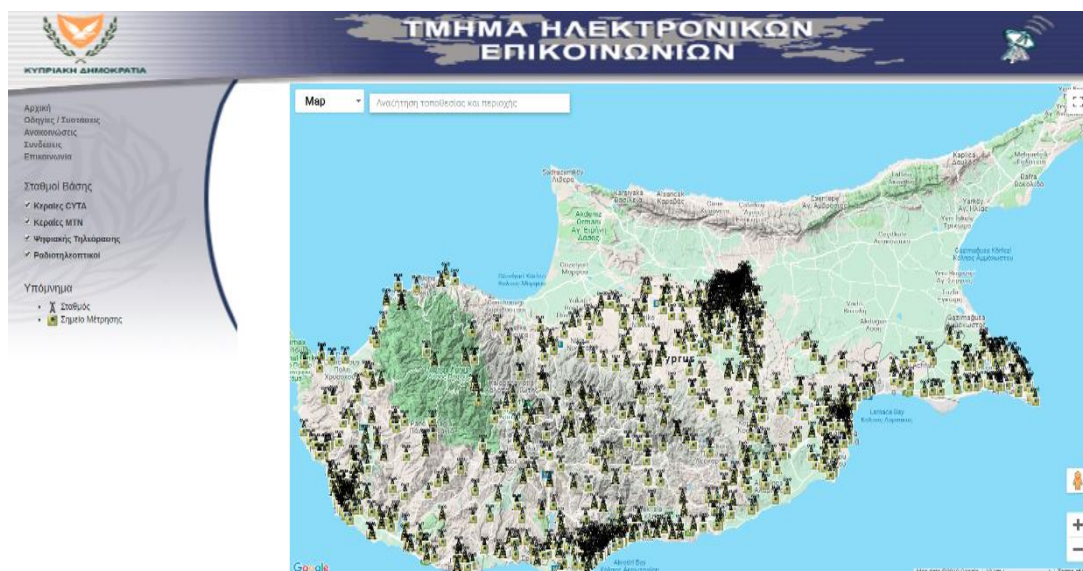
https://api.wigle.net/api/v2/network/detail?netid=&operator=* &lac=* &cid=* &system=&network=&basestation=&Query2=Query.

Όπου υπάρχουν αστεράκια μπαίνουν αντίστοιχα ο κωδικός του τηλεπικοινωνιακού παρόχου, το cellid και το lac. Συγκεκριμένα, για τον κωδικό του τηλεπικοινωνιακού παρόχου μπαίνει πρώτα ο αριθμός που αντιστοιχεί για την κάθε χώρα, π.χ., για την Κύπρο είναι ο αριθμός 280, και έπειτα ο κωδικός που αντιστοιχεί στον τηλεπικοινωνιακό πάροχο, π.χ., για την MTN είναι ο αριθμός 10, άρα στην περίπτωση αυτή ο αριθμός 28010 θα έμπαινε σαν παράμετρο του operator στο url. Τα id που χρησιμοποιήθηκαν για την εκτέλεση των επρωτημάτων συλλέχθηκαν από τα συστήματα opencellid, MLS, αλλά και από κεραίες που ήταν γνωστές λόγω των ακατέργαστων (raw) δεδομένων που μαζεύτηκαν μέσω του TrafficWare application που υλοποιήθηκε κατά τη διάρκεια της διπλωματικής. Τα δεδομένα που επέστρεφε το Wigle.net περιείχαν γεωγραφικά μήκη και πλάτη, είδος δικτύου, όνομα δικτύου και σήμα. Η Wigle.net ιστοσελίδα έχει ως περιορισμό όλα τα αιτήματα (requests) που μπορεί ο κάθε χρήστης να κάνει είναι 20 ανά ημέρα, όπου συνολικά χρειαζόταν γύρω στα 20 δευτερόλεπτα για την αίτηση και λήψη αποτελεσμάτων. Τα αποτελέσματα των request πολλές φορές μπορεί να ήταν κενά λόγω του ότι η ζητούμενη κεραία δεν υπάρχει στη Wigle βάση δεδομένων.

Κυβερνητικά Δεδομένα

Κατά τη διάρκεια της αναζήτησης ανοικτών δεδομένων, εντοπίστηκαν κυβερνητικά ανοικτά δεδομένα και πιο συγκεκριμένα, η εθνική διαδικτυακή πύλη ανοικτών δεδομένων (open data portal [47]). Αυτά τα κυβερνητικά Κυπριακά δεδομένα ανήκουν στο Υπουργείο Οικονομικών, Τμήμα Δημόσιας Διοίκησης και Προσωπικού και διαθέτουν δεδομένα των πλείστων φορέων του κυπριακού δημοσίου για περαιτέρω χρήση. Κάποια από τα δεδομένα αυτά περιέχουν και τον κατάλογο αδειοδοτημένων σταθμών ραδιοεπικοινωνίας και συγκεκριμένα όλες τις τοποθεσίες των κυψελών GSM (Global System For Mobile) και DCS (Digital Cellular System). Συγκεκριμένα, για κάθε κεραία παρέχουν τα εξής δεδομένα: Coding, name, address, lat, lon, gsm cells, dsc cells, ant height, services, site installation και area. Το coding και name είναι ο κωδικός και το όνομα της κεραίας αντίστοιχα, παρ' όλα αυτά σε όλες τις κεραίες που ανακτήθηκαν, αυτά τα δύο χαρακτηριστικά ήταν παντού τα ίδια. Το address είναι η διεύθυνση που βρίσκεται τοποθετημένη η κεραία. Το lat, lon είναι το γεωγραφικό μήκος και πλάτος της τοποθεσίας της κεραίας αντίστοιχα. Το gsm cells είναι ο αριθμός των κυψελών που υπάρχει σε κάθε σταθμό (base station), ενώ το dcs (digital cellular system) cells είναι ο αριθμός των κυψελών που χρησιμοποιούνται για συχνότητες ραδίου (radio frequencies). Το ant height είναι το ύψος της κάθε κεραίας - σταθμού (base station). Το services είναι ο τύπος δικτύου που παρέχει η κάθε κεραία, π.χ., GSM, UMTS, LTE, κτλ. Το site installation είναι το φυσικό σημείο που βρίσκεται η κεραία, π.χ., η οροφή ενός κτιρίου. Τέλος, το area είναι ο τύπος περιοχής που είναι τοποθετημένη η κεραία, π.χ., αστική.

Παρ' όλα αυτά, τα δεδομένα δεν μπορούν να αξιοποιηθούν για την εύρεση οδικής κίνησης λόγω του ότι δεν παρέχουν τα Cell Ids. Τα δεδομένα είναι συνολικά 1837 εγγραφές των 520KB.



Σχήμα 5.8 Κυβερνητικά Δεδομένα [48]

Ακόμη, εντοπίστηκε και ο χάρτης του τμήματος ηλεκτρονικών επικοινωνιών που απεικονίζεται στο Σχήμα 5.8. Ο χάρτης [48] αυτός, παρουσιάζει σχεδόν τα ίδια δεδομένα με την εθνική διαδικτυακή πύλη ανοικτών δεδομένων, παρ' όλα αυτά υπάρχουν και κάποια επιπλέον δεδομένα. Τα επιπλέον αυτά δεδομένα είναι ο συντελεστής έκθεσης EQ (Σχήμα 5.9) και οι φορές κάτω από τα Εθνικά Όρια Έκθεσης, που αποτελούν τους 'μέσους όρους' σε μια φάση ανίχνευσης 0Hz - 300GHz. Μέσω αυτών των δεδομένων, μπορεί να αντιστραφεί η φόρμουλα για την εύρεση της ισχύος σ ημάτων των κεραιών, π.χ., για 2G δίκτυα η ισχύς θα μπορούσε να ήταν 900MHz, 800MHz, 1800MHz και 1900MHz, για 3G δίκτυα: 850MHz, 900MHz και 2.1GHz, και για LTE δίκτυα: 1800MHz, 2.1GH και 2.6GHz. Τα δεδομένα αυτά, λόγω πολυπλοκότητας και έλλειψης χρόνου δεν μελετήθηκαν εις βάθος, παρ' όλα αυτά θα μπορούσαν να χρησιμοποιηθούν για την εύρεση κίνησης. Ακόμη, παρόμοια δεδομένα θα μπορούσαν να εντοπιστούν παγκοσμίως, αφού τουλάχιστον για την Ευρωπαϊκή

Ένωση υπάρχει η κατευθυντήρια γραμμή [49] σχετικά με την αποθήκευση και διαχείριση των δεδομένων αυτών.

$$EQ = \sum_{f_i=1\text{Hz}}^{1\text{MHz}} \frac{E_{mi}}{E_{l,i}} + \sum_{f_j>1\text{MHz}}^{10\text{MHz}} \frac{E_{mi}}{\alpha} + \frac{E_e}{E_l}$$

όπου,

E_{mi} : η μετρούμενη ένταση του ηλεκτρικού πεδίου από υφιστάμενη πηγή με συχνότητα λειτουργίας f_i ,

E_{li} : το επίπεδο αναφοράς (Παράρτημα) στη συχνότητα λειτουργίας f_i ,

E_e : η υπολογιζόμενη ένταση του ηλεκτρικού πεδίου από τη λειτουργία του νέου σταθμού ραδιοεπικοινωνίας,

E_l : το επίπεδο αναφοράς στη συχνότητα λειτουργίας του νέου σταθμού ραδιοεπικοινωνίας, $\alpha = 87$ (V/m).

Σχήμα 5.9 Φόρμουλα συντελεστή EQ [50]

OpenStreetMap

Το OpenStreetMap [51] (OSM) είναι ένα project που αποσκοπεί στην δημιουργία ενός δωρεάν επεξεργάσιμου χάρτη του κόσμου. Τα δεδομένα που παράγονται από το έργο θεωρούνται το πρωταρχικό προϊόν και όχι ο χάρτης. Η ανάγκη της δημιουργίας και ανάπτυξης του συστήματος αυτού προκλήθηκε από τους περιορισμούς που υπάρχουν στη χρήση ή διαθεσιμότητα πληροφοριών σε μεγάλο μέρος του κόσμου. Ακόμη, το σύστημα αναπτύχθηκε με σκοπό την δημιουργία φθηνών φορητών δορυφορικών συσκευών πλοήγησης (navigation devices). Το σύστημα αυτό θεωρείται σημαντικό παράδειγμα εθελοντικής γεωγραφικής πληροφορίας. Οι πληροφορίες που διαθέτει έχουν συλλεχθεί από διάφορους χρήστες (crowdsourcing) και είναι ανοικτές προς όλους.

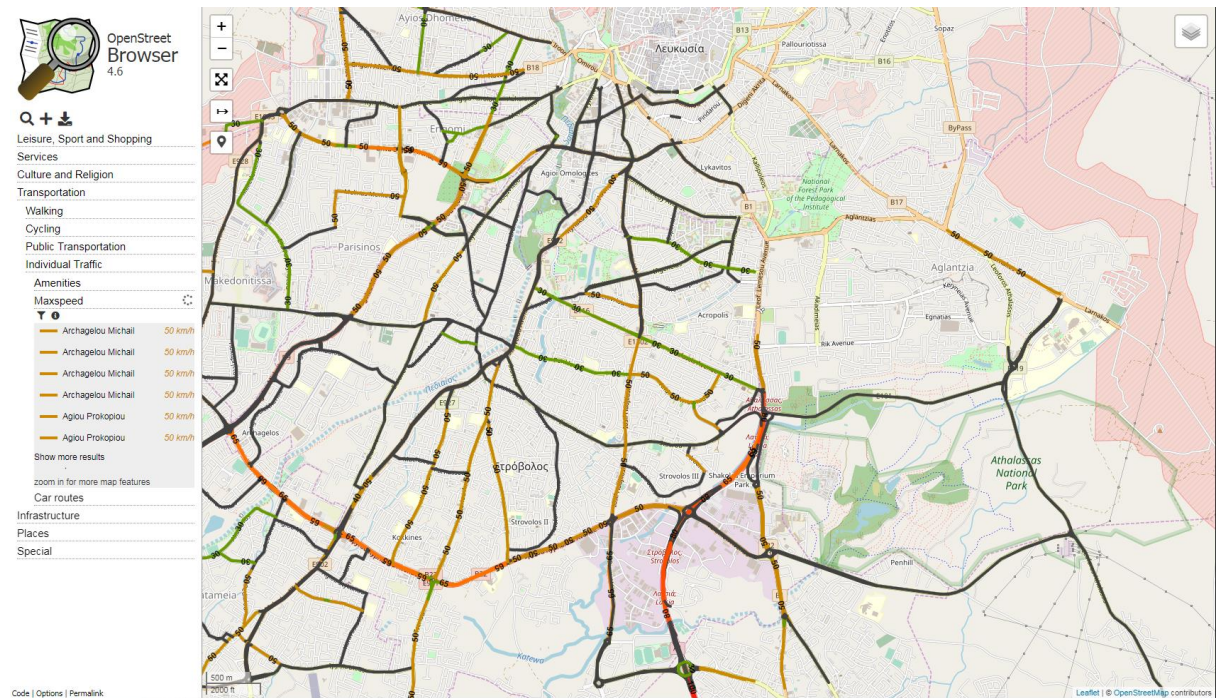
Τα δεδομένα που προσφέρει έχουν χρησιμοποιηθεί από συστήματα όπως Facebook, Geocaching, Foursquare κτλ, με σκοπό την αντικατάσταση των Google Maps.

Τα αρχέτυπα δεδομένων του OSM αποθηκεύονται και επεξεργάζονται σε διαφορετικές μορφές. Το κύριο αντίγραφο των δεδομένων αποθηκεύεται στην κύρια βάση δεδομένων του συστήματος, όπου είναι μία Postgres βάση δεδομένων με PostGIS extension. Η βάση αυτή, έχει έναν πίνακα για κάθε πρωτόκολλο δεδομένων, με μεμονωμένα αντικείμενα αποθηκευμένα ως σειρές (rows). Όλες οι επεξεργασίες συμβαίνουν στη βάση δεδομένων και δημιουργούνται όλες οι άλλες μορφές δεδομένων από αυτήν.

Για τη μεταφορά δεδομένων, δημιουργούνται πολλές βάσεις δεδομένων, οι οποίες είναι διαθέσιμες για λήψη. Το πλήρες αυτό σύστημα που επιτρέπει τη λήψη (download) των δεδομένων αυτών, λέγεται planet.osm [52]. Το planet.osm παρέχει αρχεία δύο μορφών, Extensible Markup Language (XML) και Protocol Buffer Binary Format (PBF).

Το OpenStreetMap χρησιμοποιεί μια δομή γεωχωρικών δεδομένων με τέσσερα βασικά στοιχεία. Τα τέσσερα αυτά στοιχεία είναι οι κόμβοι (nodes), οι δρόμοι (ways), οι σχέσεις (relations) και οι ετικέτες (tags). Οι κόμβοι είναι τα σημεία με γεωγραφική θέση που αποθηκεύονται ως συντεταγμένες (ζεύγη γεωγραφικού μήκους και πλάτους). Χρησιμοποιούνται και για τη συσχέτιση δρόμων (ways), αλλά και για την αντιπροσώπευση χαρακτηριστικών - σημείων χωρίς μέγεθος, π.χ., κορυφές βουνών. Οι δρόμοι (ways) είναι ταξινομημένοι κατάλογοι κόμβων, που αντιπροσωπεύουν μια πολύγραμμη, ή ένα πολύγωνο αν σχηματίζουν ένα κλειστό βρόγχο (closed loop). Χρησιμοποιούνται τόσο για την απεικόνιση γραμμικών χαρακτηριστικών, όπως δρόμους και ποτάμια, όσο και για περιοχές όπως δάση, πάρκα, χώρους στάθμευσης και λίμνες. Απ' την άλλη, οι σχέσεις (relations) είναι ταξινομημένες λίστες κόμβων (nodes), δρόμων (ways) και σχέσεων (relations) και μαζί ονομάζονται 'μέλη' (members). Το κάθε μέλος μπορεί προαιρετικά να έχει μια συμβολοσειρά (string). Οι σχέσεις χρησιμοποιούνται για την αναπαράσταση της σχέσης υφιστάμενων κόμβων και δρόμων (ways), π.χ., στροφές στους οδικούς δρόμους. Τέλος, οι ετικέτες (tags) αποτελούν ζεύγη κλειδιών - τιμών και χρησιμοποιούνται για την αποθήκευση μεταδεδομένων (metadata) σχετικά με αντικείμενα του χάρτη, π.χ., ο τύπος, το όνομα και οι φυσικές του ιδιότητες.

Οι ετικέτες είναι πάντα συνδεδεμένες - εξαρτημένες, σε ένα αντικείμενο, σε ένα κόμβο, way ή μία σχέση.



Σχήμα 5.10 Road Speed Limits στη Λευκωσία [53]

Κατά τη διάρκεια της διπλωματικής αυτής, χρειάστηκε να μελετηθούν δεδομένα που παρέχουν τα όρια ταχυτήτων των οδικών δρόμων για την Κύπρο. Έτσι μελετήθηκαν τα OSM δεδομένα για την εύρεση των ορίων ταχυτήτων. Κατά τη διάρκεια της μελέτης των δεδομένων εντοπίστηκε ότι για κάθε way υπάρχουν καταγεγραμμένα τα όρια ταχυτήτων των οδικών δρόμων (road speed limits). Παρ' όλα αυτά, δεν χρησιμοποιήθηκαν λόγω του ότι τα δεδομένα αυτά για την Κύπρο ήταν ελάχιστα και περιλάμβαναν κυρίως αυτοκινητόδρομους (highways), μαζί με πολύ λίγους άλλους δρόμους. Αυτό απεικονίζεται και στο Σχήμα 5.10. Γενικότερα, η OpenStreetMap βάση δεδομένων για την Κύπρο, περιέχει 1614348 nodes, 165357 ways, και 1724 relations.

Επιπρόσθετα, στα πλαίσια της διπλωματικής, οι OpenStreetMap χάρτες έχουν χρησιμοποιηθεί στην TrafficWare πλατφόρμα.

Όρια Ταχυτήτων των Οδικών Δρόμων (Road Speed Limits)

Για την υλοποίηση του συστήματος, ένα βασικό στοιχείο είναι η γνωστοποίηση των ορίων ταχύτητας των οδικών δρόμων. Ο αλγόριθμος που θα αντιστοιχεί τους χρήστες πάνω στον χάρτη χρειάζεται να έχει ως περιορισμό τα όρια ταχυτήτων, έτσι ώστε να μπορεί να υπολογίσει μέσα στο χρονικό διάστημα που κινήθηκε ο χρήστης από το ένα σημείο στο άλλο - δηλαδή από το τελευταίο γνωστό σημείο που βρισκόταν, στην καινούργια τοποθεσία του χρήστη -, ποια ήταν η διαδρομή που ακολούθησε. Για να υπολογίσει την διαδρομή, θα πρέπει το σύστημα να λάβει υπόψη τα πιο σύντομα μονοπάτια από το ένα σημείο στο άλλο και το όριο ταχύτητας, έτσι ώστε να υπολογιστεί αν ο χρήστης είχε τον χρόνο να φτάσει στο καινούργιο σημείο μέσω των συγκεκριμένων διαδρομών που το σύστημα εξετάζει τη δεδομένη στιγμή.

Για τον σκοπό αυτό, μελετήθηκαν τα δεδομένα OpenStreetMap, αλλά δυστυχώς τα δεδομένα αυτά για την Κύπρο είναι ελάχιστα. Ακόμη, μελετήθηκε το HERE [54] API. Δυστυχώς, και σε αυτή την περίπτωση τα δεδομένα που αφορούν την Κύπρο ήταν ελάχιστα.

Ακόμη, η Google που έχει το δικό της σύστημα εύρεσης οδικής κίνησης (Google Maps [55]), δεν παρέχει τα όρια ταχυτήτων για όλους τους δρόμους της Κύπρου, ούτε βρίσκει την οδική κίνηση για όλους τους δρόμους. Παρ' όλα αυτά, η χρησιμοποίηση υπηρεσιών της Google δεν είναι δωρεάν, ούτε ανοικτή, γι' αυτό τον λόγο τα δεδομένα αυτά δεν μελετήθηκαν εις βάθος.

Επιπρόσθετα, μελετήθηκε η εφαρμογή Waze [56], που έχει γνωστοποιήσει αρκετά όρια ταχυτήτων. Τα δεδομένα της εφαρμογής αυτής είναι πληθοποριστικά (crowdsourcing), έτσι οι χρήστες ενημερώνουν το σύστημα με πληροφορίες. Το σύστημα αυτό είναι πλούσιο σε όρια ταχυτήτων στην Κύπρο, αλλά δυστυχώς δεν είναι ανοιχτά τα δεδομένα αυτά, αφού ακολουθεί την πολιτική της Google.

Τα μόνα Κυβερνητικά ανοιχτά δεδομένα που αναφέρονται στα όρια ταχυτήτων που υπάρχουν είναι η διαδρομή του νέου αυτοκινητόδρομου Λευκωσίας - Λεμεσού.

Για τον λόγο αυτό, το παρόν σύστημα δεν περιλαμβάνει όρια ταχυτήτων.

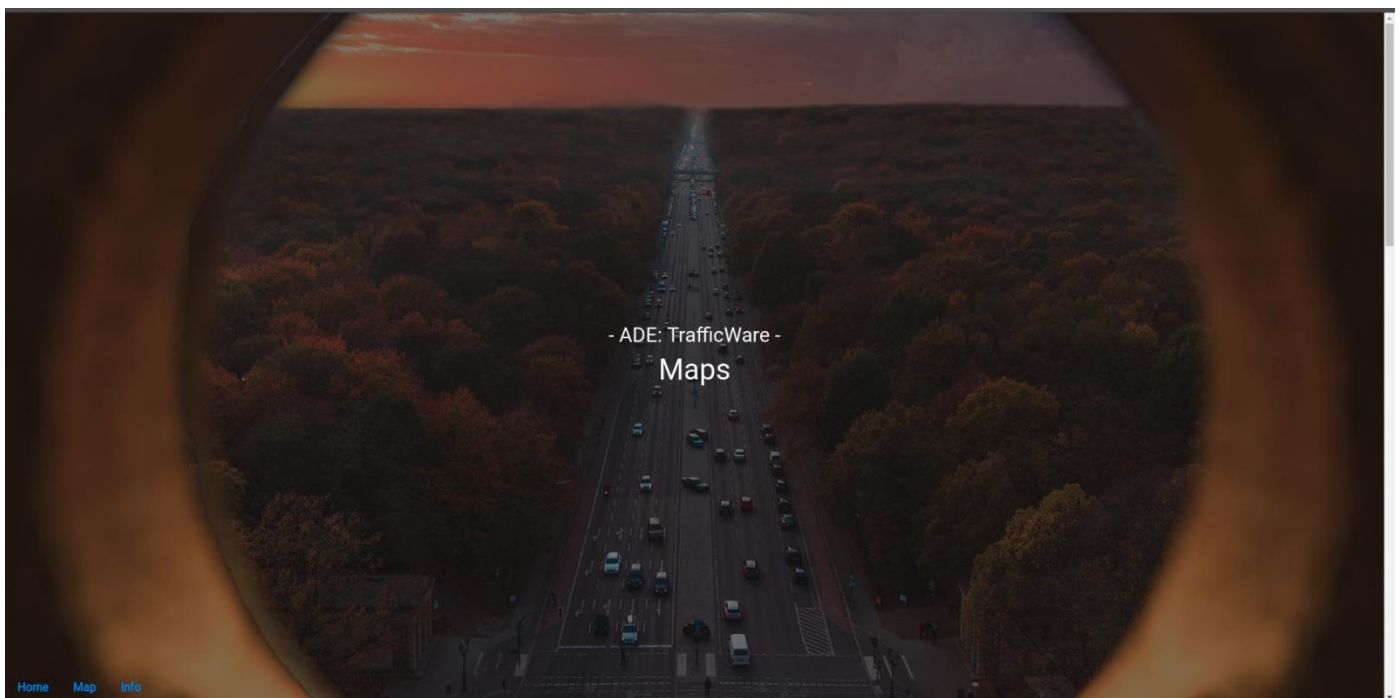
5.3 Πλατφόρμα TrafficWare

Κατά τη διάρκεια του πρώτου εξαμήνου, υλοποιήθηκε μία πλατφόρμα συστήματος με σκοπό την παρουσίαση μεγάλων δεδομένων σε ένα χάρτη. Επιπλέον, η ιστοσελίδα είναι διαδραστική και γρήγορη. Πίσω από αυτήν, κρύβονται αλγόριθμοι για την επίτευξη γρήγορης λήψης δεδομένων και γρήγορης διαδραστικότητας μεταξύ της ιστοσελίδας και του χρήστη. Στην ιστοσελίδα εμφανίζονται όσο πιο λίγα δεδομένα είναι δυνατόν ή όσα χρειάζονται, έτσι ώστε η ιστοσελίδα να είναι ελαφριά και γρήγορη στη συσκευή του χρήστη. Συγκεκριμένα, απεικονίζει μόνο τα δεδομένα που αφορούν τα πλακάκια (tiles) του χάρτη που παρουσιάζονται τη δεδομένη στιγμή στον χρήστη. Ακόμη, έχουν εφαρμοστεί διάφοροι τρόποι caching με σκοπό τη λιγότερη δυνατή ανάγκη να γίνονται HTTP αιτήματα στον διακομιστή (server). Η γενική ιδέα είναι ότι στο τελικό στάδιο της έρευνας, η ιστοσελίδα θα εμφανίζει πάνω στον χάρτη σε ποια σημεία υπάρχει οδική κίνηση για μια δεδομένη στιγμή.

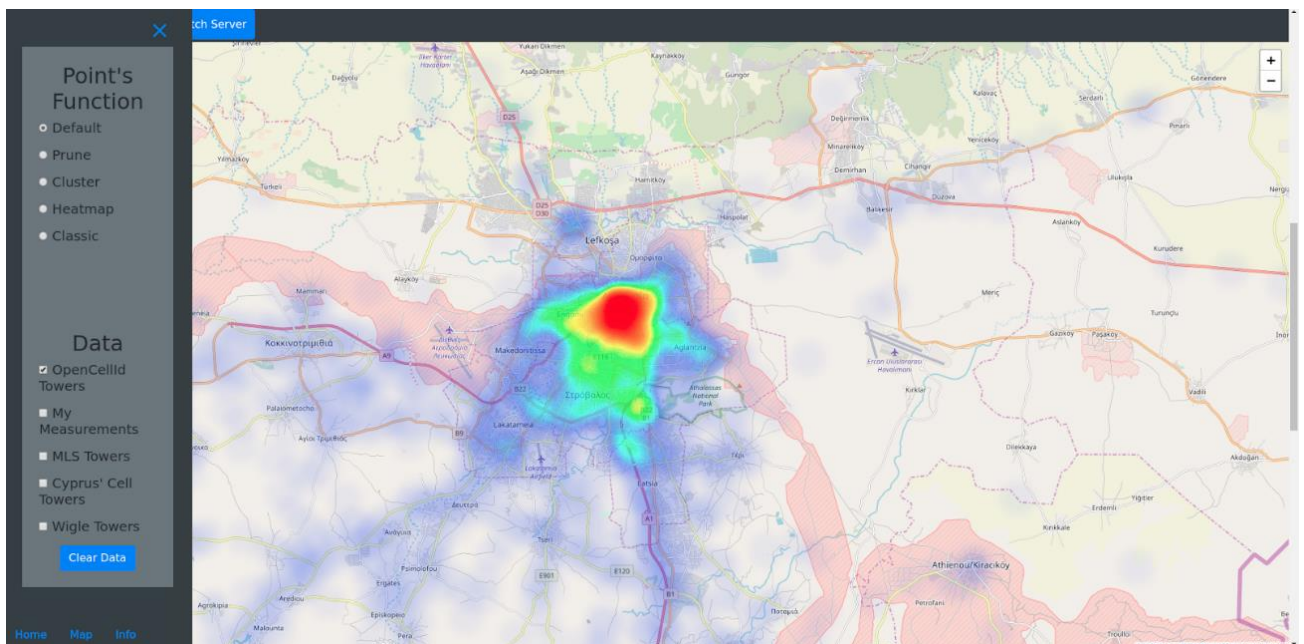
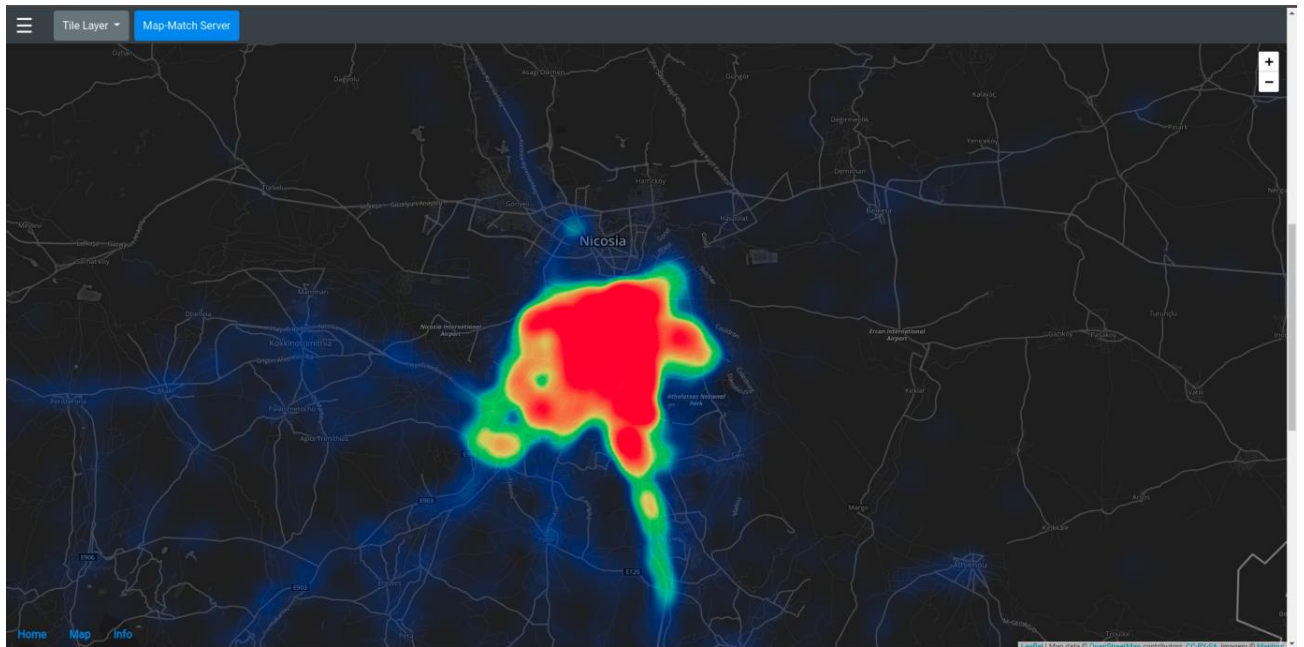
Ο κάθε χρήστης για να έχει πρόσβαση στην ιστοσελίδα, πρέπει να έχει πρόσβαση στο δίκτυο της πληροφορικής του Πανεπιστημίου Κύπρου, αφού το ip βρίσκεται κάτω από το δίκτυο αυτό. Η ιστοσελίδα βρίσκεται στη διεύθυνση <http://10.16.20.19/>. Η ιστοσελίδα έχει στηθεί σε ένα VM (Virtual Machine) που ανήκει στο DMSL (Data Management Systems Laboratory). Το VM λειτουργεί ως Server με τη βοήθεια του Apache Server και έχει τα εξής χαρακτηριστικά:

- Λειτουργικό Σύστημα: Ubuntu 16.04
- CPU: Intel(R) Xeon(R) 2.40GHz
- Ram: 20GB

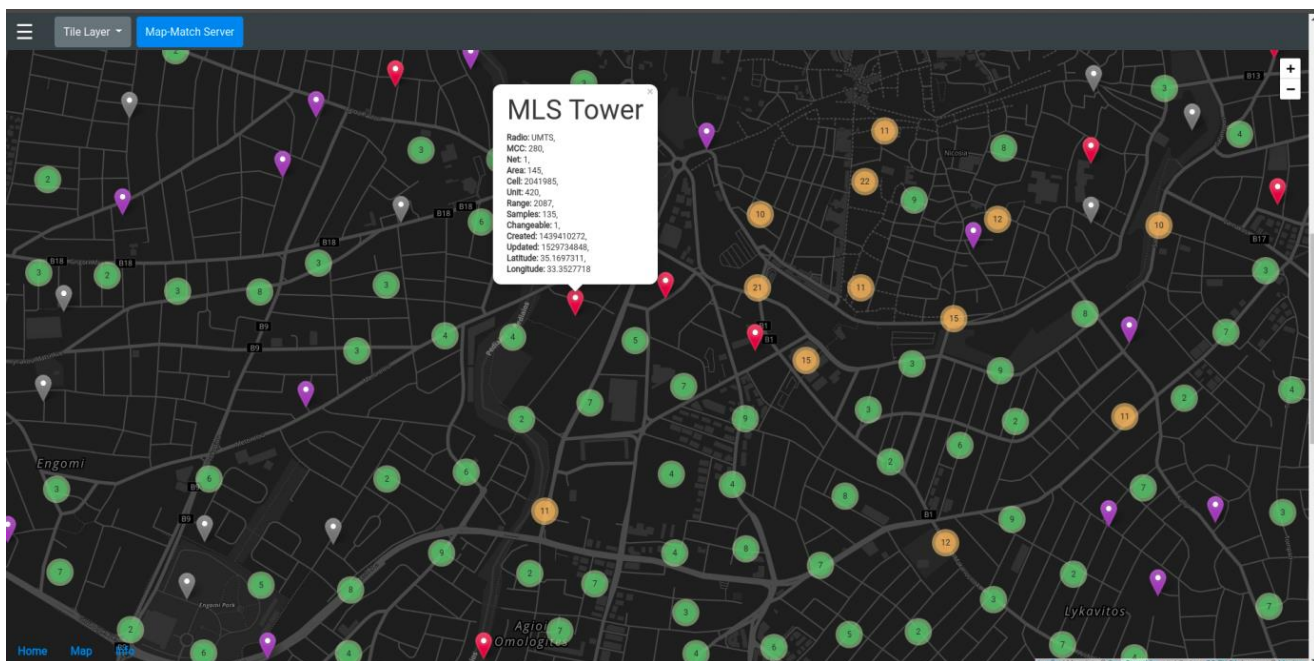
Η πρώτη σελίδα που διακρίνεται μόλις κάποιος μπει στην ιστοσελίδα είναι η εικόνα που φαίνεται στο Σχήμα 5.11. Στο κάτω μέρος της σελίδας, υπάρχει ένας κατάλογος με τρεις επιλογές, “Home”, “Map”, και “Info”. Στην επιλογή “Home”, ο χρήστης μεταφέρεται στην πρώτη σελίδα που βρίσκεται η εικόνα που φαίνεται στο Σχήμα 5.11. Πατώντας την επιλογή “Info”, ο χρήστης μεταφέρεται στο τέλος της ιστοσελίδας, όπου υπάρχουν κάποιες γενικές πληροφορίες, π.χ., από πού είναι τα tiles του χάρτη, κτλ. Στην επιλογή “Map”, υπάρχει το κύριο μέρος της ιστοσελίδας, που είναι ο διαδραστικός χάρτης. Το menu αυτό υπάρχει στο κάτω μέρος όλων των σελίδων. Στο σημείο αυτό είναι αξιοσημείωτο να αναφερθεί ότι όλες οι σελίδες στην πραγματικότητα είναι μία σελίδα χωρισμένη σε ενότητες (sections).



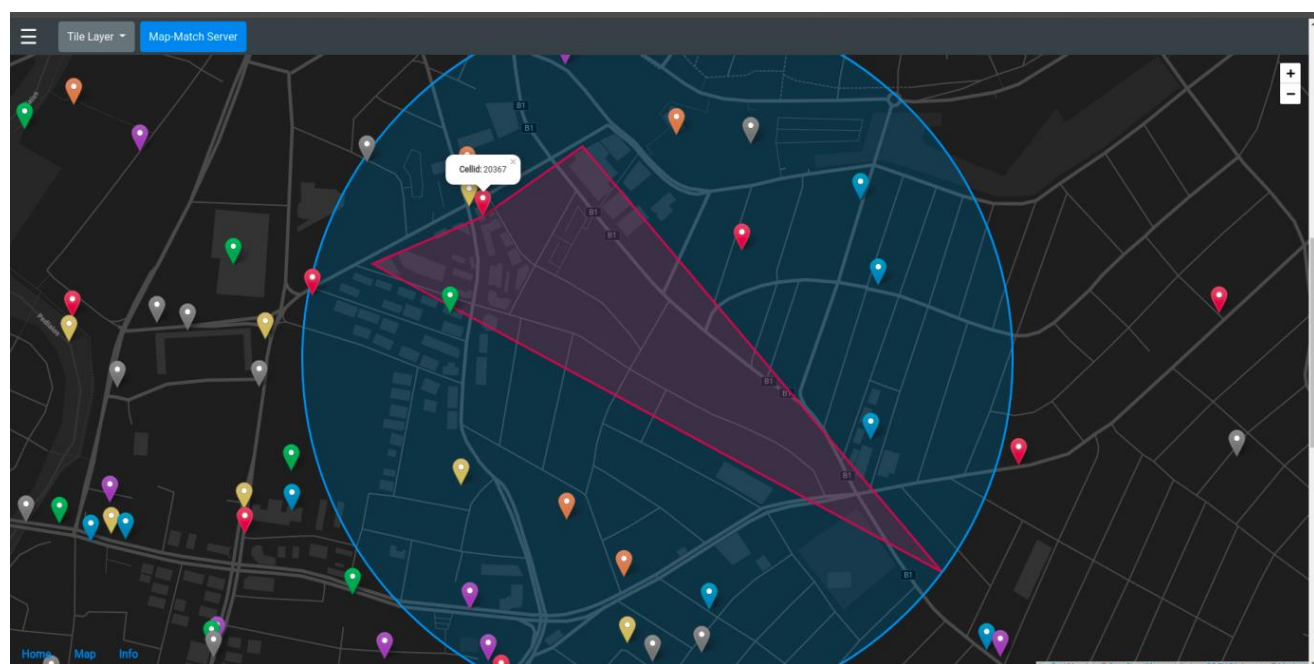
Σχήμα 5.11 Αρχική Σελίδα της Ιστοσελίδας



(γ)



(δ)



Σχήμα 5.12 Διεπαφή της Ιστοσελίδας

Στο Σχήμα 5.12 (α) παρουσιάζονται τα δεδομένα που εμφανίζονται προκαθορισμένα (by default) όταν επιλεγεί η επιλογή “Map”. Τα προκαθορισμένα δεδομένα που φαίνονται είναι τα OpenCellId δεδομένα, σε heatmap μορφή. Τα OpenCellId δεδομένα είναι οι υπολογιζόμενες (approximate) θέσεις των κεραιών. Τα δεδομένα αυτά θα εξηγηθούν στην επόμενη ενότητα. Καθώς ο χρήστης μεγεθύνει τον χάρτη, τα δεδομένα αλλάζουν μορφή και ο χρήστης μπορεί να τα δει με περισσότερη λεπτομέρεια. Όταν ο χάρτης φτάσει στο μέγιστο σημείο μεγέθυνσης, ο χρήστης μπορεί να πατήσει πάνω σε ένα pin και τότε εμφανίζονται περισσότερες πληροφορίες για την κεραία που έχει επιλέξει, όπως φαίνεται στο Σχήμα 5.12 (γ). Η ίδια διαδικασία συμβαίνει για όλα τα είδη δεδομένων.

Στο πάνω μέρος του χάρτη, υπάρχει ένας άλλος κατάλογος που δίνει στον χρήστη τη δυνατότητα αλλαγής τύπου (map tile layer) του χάρτη, την επιλογή να μεταφερθεί στον διακομιστή αντιστοίχισης χάρτη – τροχιών (Map – Matching Server), αλλά και την επιλογή ενός νέου καταλόγου (menu). Η επιλογή της μεταφοράς στον διακομιστή (server) αντιστοίχισης χάρτη θα αναφερθεί στο επόμενο κεφάλαιο. Η επιλογή αλλαγής τύπου χάρτη, δίνει τη δυνατότητα στον χρήστη να επιλέξει μεταξύ του προκαθορισμένου επιπέδου που φαίνεται στο Σχήμα 5.12 (β), του κλασικού επιπέδου, του σκοτεινού (dark) επιπέδου που φαίνεται και στο Σχήμα 5.12 (δ) και του δορυφορικού επιπέδου.

Όταν ο χρήστης επιλέξει τον νέο κατάλογο, τότε ο κατάλογος, που φαίνεται στο Σχήμα 5.12 (β), εμφανίζεται. Στον κατάλογο αυτό, ο χρήστης μπορεί να επιλέξει τόσο το είδος δεδομένων που θέλει να εμφανιστούν πάνω στον χάρτη, όσο και τη δυνατότητα επιλογής του τύπου εμφάνισης δεδομένων.

Για την επιλογή του είδους δεδομένων, υπάρχουν οι εξής επιλογές: OpenCellId, My Measurements, MLS, Cyprus’ Cell Towers, και Wigle, ενώ για την επιλογή τύπου απεικόνισης δεδομένων υπάρχουν οι εξής επιλογές: Default, Prune, Cluster, Heatmap, και Classic. Το σύστημα παρέχει στον χρήστη τη δυνατότητα επιλογής πολλαπλών ειδών δεδομένων. Όσον αφορά την επιλογή τύπου δεδομένων, ο χρήστης κάθε φορά πρέπει να επιλέξει μόνο ένα. Μόλις ο χρήστης κάνει την επιλογή του, την ίδια δεδομένη

στιγμή τα δεδομένα εμφανίζονται πάνω στον χάρτη. Για παράδειγμα, στο Σχήμα 5.12 (α) απεικονίζονται πολλά είδη δεδομένων μαζί σε μορφή heatmap.

Ο προκαθορισμένος τρόπος απεικόνισης δεδομένων της ιστοσελίδας είναι να εμφανίζει τα δεδομένα ως heatmap. Την τακτική αυτή την ακολουθεί, έτσι ώστε ο χάρτης να είναι πιο διαδραστικός, αλλά και για να παρουσιάζει συνολικά τα σημεία στα οποία υπάρχουν τα δεδομένα. Ακόμη, με τον τρόπο αυτό, το σύστημα είναι πιο γρήγορο όταν ο χρήστης κάνει γρήγορα zoom in και zoom out. Όταν ο χρήστης μεγεθύνει αρκετά τον χάρτη, έτσι ώστε να μπορούν τα δεδομένα να διακρίνονται σωστά, τότε το σύστημα εμφανίζει τα δεδομένα ως Clustering. Ο τρόπος αυτός, ομαδοποιεί κάποια pins, έτσι ώστε να μην υπερφορτώνει τον χάρτη. Αν ο χρήστης θέλει να δει με περισσότερη λεπτομέρεια τα δεδομένα, τότε θα πρέπει να μεγεθύνει ακόμη περισσότερο τον χάρτη, ούτως ώστε να εμφανίζονται τα pins ξεχωριστά. Στο σημείο αυτό, ο χρήστης θα μπορεί να πατήσει σε όποιο pin επιθυμεί για να μπορέσει να δει περισσότερες πληροφορίες σχετικά με το pin που έχει επιλέξει. Ο Clustering τρόπος φαίνεται και στο Σχήμα 5.12 (δ).

Ο Prune τρόπος απεικόνισης δεδομένων είναι σχεδόν ο ίδιος με τον Clustering τρόπο, αλλά έχει διαφορετική τακτική υλοποίησης. Ο λόγος που υπάρχουν και οι δύο αυτοί τρόποι, ήταν για τη διερεύνηση του πιο αποδοτικού τρόπου. Για την πιο γρήγορη απόδοση, ο Clustering τρόπος είναι καλύτερος, ενώ ο Prune τρόπος έχει μια καλύτερη τακτική υλοποίησης και καλύτερη τακτική ‘ομαδοποίησης’ pin. Τέλος, ο classic τρόπος είναι η απλή φόρτωση όλων των pin, όπως φαίνεται στο Σχήμα 5.12 (δ).

Ο τύπος δεδομένων ‘My Measurements’ είναι όλες οι μετρήσεις τροχιών που έχουν συλλεχθεί με τη βοήθεια της εφαρμογής [Κεφάλαιο 6] που έχει υλοποιηθεί κατά τη διάρκεια της διπλωματικής αυτής. Συγκεκριμένα, η συλλογή των δεδομένων αυτών έχει γίνει καθώς οδηγούσα σε διάφορα σημεία της Λευκωσίας. Ακόμη, τα δεδομένα αυτά εμπερικλείουν τα ακατέργαστα (raw) δεδομένα που συλλέχτηκαν από το Wigle.net. Εάν ο χρήστης επιλέξει να δει τα ‘My Measurements’, τότε εμφανίζεται μια νέα επιλογή απεικόνισης δεδομένων, η ‘Clustering By Ids’. Ο συγκεκριμένος τύπος δεδομένων, εμφανίζει ως ένα pin όλες τις μετρήσεις που υπάρχουν για μία κεραία (ένα cellid). Όταν ο χρήστης επιλέξει ένα pin, τότε εμφανίζεται ένας κύκλος που αντιπροσωπεύει την

περίμετρο που καλύπτει η κεραία, ένα μήνυμα που περιέχει το cellid της κεραίας, αλλά και ένα πολύγωνο που ‘περικυκλώνει’ όλα τα σημεία που είναι καταγεγραμμένα με το συγκεκριμένο cellid. Το πολύγωνο έχει υλοποιηθεί βασισμένο στον γνωστό gift wrapping αλγόριθμο. Στο Σχήμα 5.12 (δ) απεικονίζεται το αποτέλεσμα όταν επιλεγθεί η επιλογή ‘Clustering By Ids’.

Κεφάλαιο 6

Υποσύστημα Αντιστοίχισης

6.1 Εφαρμογή για Συλλογή Τροχιών του TrafficWare	75
6.2 Αντιστοίχιση Τροχιών σε Χάρτη	79

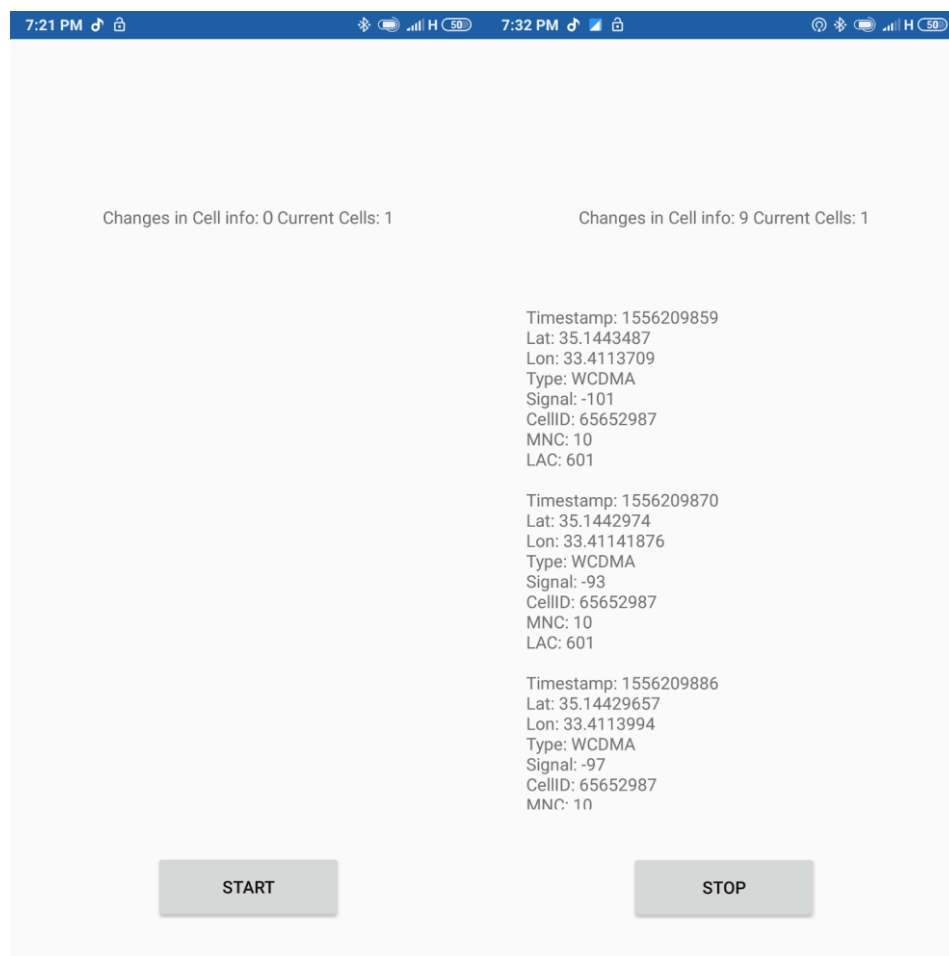
Στην ενότητα αυτή θα περιγραφεί η εφαρμογή που υλοποιήθηκε για την συλλογή διαφόρων δεδομένων τροχιών με σκοπό την χρησιμοποίησή τους στο τελικό σύστημα. Τα δεδομένα αυτά, περιλαμβάνουν διάφορες πληροφορίες σχετικά με τις κεραίες που η συσκευή ανιχνεύει ή συνδέεται. Ακόμη, θα περιγραφεί το σύστημα που ενσωματώθηκε στην TrafficWare πλατφόρμα, που λύνει το πρόβλημα της αντιστοίχισης τροχιών σε χάρτη.

6.1 Εφαρμογή για Συλλογή Τροχιών

Κατά την αναζήτηση ανοικτών δεδομένων και τροχιών (trajectories) αποφασίστηκε ότι η δημιουργία μιας εφαρμογής που συλλέγει διάφορα δεδομένα τροχιών θα ήταν η καλύτερη επιλογή, αφού τα περισσότερα ανοικτά δεδομένα τροχιών δεν περιλάμβαναν cell id. Τα δεδομένα τροχιών περιλαμβάνουν πληροφορίες όπως γεωγραφικά μήκη και πλάτη, cell id και τύπο δικτύου π.χ., GSM, UMTS, LTE. Επιπλέον, για την υλοποίηση ενός συστήματος χρησιμοποιώντας πληθοπορισμό (crowdsourcing) [Κεφάλαιο 4], χρειάζεται να γνωρίζουμε όλες τις κεραίες που μία συσκευή ανιχνεύει και όχι μόνο την κεραία που είναι συνδεδεμένη. Όλες οι δωρεάν εφαρμογές που υπάρχουν μέχρι στιγμής, δεν επιτρέπουν την ανίχνευση ή αποθήκευση γειτονικών κεραιών.

Η εφαρμογή που υλοποιήθηκε είναι συμβατή για android συσκευές. Στο Σχήμα 6.1 απεικονίζεται η διεπαφή του συστήματος. Στο Σχήμα 6.1 (α) φαίνεται η κατάσταση της

εφαρμογής μόλις ο χρήστης ανοίξει την εφαρμογή. Στην φάση αυτή, η εφαρμογή δεν εμφανίζει τίποτα παρά μόνο ένα κουμπί που λέει “*START*”, για να ξεκινήσει ο χρήστης να παίρνει μετρήσεις, αλλά και ένα μήνυμα που λέει “*Changes in Cell info: 0, Current Cells: 1*”, το οποίο υποδηλώνει ότι δεν υπήρξαν αλλαγές σε σχέση με τη σύνδεση της συσκευής, αλλά και ότι η συσκευή ανιχνεύει μία κεραία. Ακόμη, στο Σχήμα 6.1 (β) απεικονίζεται η διεπαφή του συστήματος αφότου ο χρήστης έχει πατήσει το κουμπί “*START*”. Η εφαρμογή αντικαθιστά το κουμπί “*START*” με ένα κουμπί “*STOP*” και εμφανίζει στον χρήστη τις μετρήσεις που καταγράφονται.



(α)

(β)

Σχήμα 6.1 Διεπαφή της Εφαρμογής

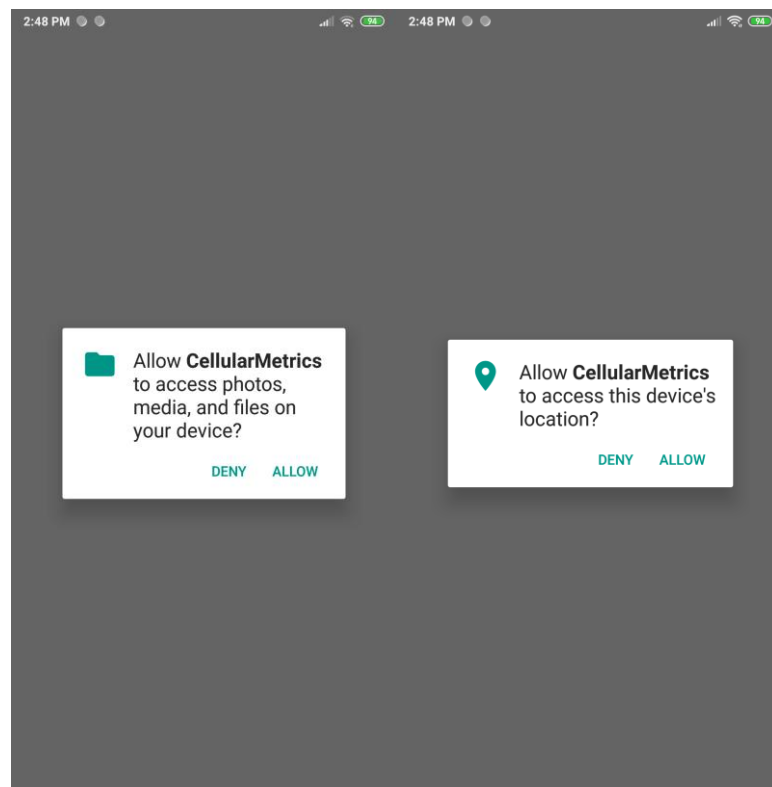
Κατά τη διάρκεια συλλογής μετρήσεων, η εφαρμογή δίνει τη δυνατότητα στον χρήστη να δει όλες τις μετρήσεις κάνοντας “*scroll down*” στην οθόνη. Καθώς ο χρήστης κάνει “*scroll down*” η οθόνη παραμένει η ίδια αλλάζοντας μόνο το κείμενο που βρίσκεται στο κέντρο της οθόνης, εμφανίζοντας έτσι στον χρήστη τις καινούργιες ή πιο παλιές μετρήσεις. Οι μετρήσεις που φαίνονται κάθε φορά στην οθόνη είναι μόνο οι μετρήσεις που έχουν καταγραφεί από την ώρα που ο χρήστης πάτησε το κουμπί “*Start*”. Παράλληλα, όταν ο χρήστης πατά το κουμπί “*Start*”, οι καινούργιες μετρήσεις προσθέτονται στο τέλος (append) ενός CSV αρχείου, με αποτέλεσμα το αρχείο να περιέχει τόσο παλαιότερες, όσο και πιο καινούργιες μετρήσεις. Το CSV αρχείο έχει την εξής μορφή: Timestamp, Lat, Lon, TypeofCell, Signal, CellId, MNC, LAC. Το Timestamp είναι η ημερομηνία και ώρα που η εφαρμογή εντόπισε αλλαγή σχετικά με το δίκτυο της συσκευής, η οποία θα μπορούσε να είναι η αλλαγή σύνδεσης της συσκευής με διαφορετική κεραία, η ανίχνευση νέας κεραίας, ή η αλλαγή σήματος που λαμβάνει η συσκευή από τις κεραίες. Το Lat και Lon αντιστοιχούν στο γεωγραφικό μήκος και πλάτος που ο χρήστης βρίσκεται όταν η εφαρμογή ανιχνεύει μία καινούργια αλλαγή σχετικά με το δίκτυο της συσκευής. Το TypeofCell είναι ο τύπος της κεραίας που ανιχνεύτηκε, π.χ., αν είναι GSM, UMTS, LTE, κτλ. Το Signal είναι το σήμα που παίρνει η συσκευή από την κεραία που καταγράφεται τη δεδομένη στιγμή. Το CellId για τα GSM και LTE δίκτυα είναι το Cell id (το id που αντιστοιχεί σε κάθε κεραία ή κάποιο τομέα μιας κεραίας), για τα UMTS δίκτυα είναι το UTRAN Cell id, το οποίο είναι η συνένωση των δύο bytes του cell id. Το Net, ή αλλιώς MNC (Mobile Network Code), είναι ο αριθμός που ανατίθεται σε κάθε πάροχο τηλεπικοινωνιακών δικτύων. Τέλος, το LAC (Location Area Code) είναι ένας μοναδικός αριθμός που αντιστοιχεί για κάθε περιοχή.

Είναι αξιοσημείωτο να αναφερθεί ότι η συσκευή μπορεί να ανιχνεύσει έως και 6 γειτονικές κεραίες που παρέχουν 2G / 3G δίκτυα, ενώ, για κεραίες που παρέχουν 4G δίκτυα, μπορεί να ανιχνευθεί μόνο η κεραία που η συσκευή είναι συνδεδεμένη. Αυτό συμβαίνει λόγω της υλοποίησης που έχουν οι android βιβλιοθήκες που είναι υπεύθυνες για την λήψη δικτύων.

Επίσης, η εφαρμογή είναι προγραμματισμένη έτσι ώστε να παραμένει η οθόνη ανοικτή κατά τη διάρκεια συλλογής πληροφοριών, έτσι ώστε ο χρήστης να μπορεί να

ενημερώνεται ότι η εφαρμογή συλλέγει σωστά τις μετρήσεις. Η εφαρμογή, παρ' όλα αυτά, δεν μπορεί να λειτουργήσει σωστά όταν βρίσκεται στο background, αφού για να μπορεί να ανιχνεύσει και να “ακούσει” κάθε αλλαγή σχετικά με την σύνδεση της συσκευής, πρέπει να έχει προτεραιότητα στην κεντρική μονάδα επεξεργασίας της συσκευής.

Τέλος, για να μπορέσει η εφαρμογή να λειτουργήσει, την πρώτη φορά χρήσης της, ο χρήστης καλείται να αποδεχτεί τα δικαιώματα (permissions) που χρειάζονται, τα οποία παρουσιάζονται και στο Σχήμα 6.2. Η αποδοχή των δικαιωμάτων από τον χρήστη, δίνει άδεια στην εφαρμογή να έχει πρόσβαση στα αρχεία της συσκευής, ώστε να μπορέσει να δημιουργήσει, να διαβάσει και να γράψει στο CSV αρχείο, όπως και να λαμβάνει τη θέση του χρήστη. Για τη σωστή λειτουργία της εφαρμογής, χρειάζεται ακόμη ο χρήστης να έχει ενεργοποιημένο το GPS της εφαρμογής και να έχει σύνδεση με κάποιο τηλεπικοινωνιακό πάροχο.



(α)

(β)

Σχήμα 6.2 Δικαιώματα της Εφαρμογής

6.2 Αντιστοίχιση Τροχιών σε Χάρτη

Για την υλοποίηση του συστήματος που θα μπορεί να απεικονίζει την κίνηση στους δρόμους, χρειάστηκε η μελέτη αλγορίθμων `map - matching`. Το πρόβλημα αντιστοίχισης (`map - matching`) είναι ένα αρκετά σημαντικό πρόβλημα για εφαρμογές που βασίζονται στην απεικόνιση διαδρομής πάνω σε χάρτη. Η απεικόνιση ενός σημείου σε ένα χάρτη είναι απλή, αφού το μόνο που χρειάζεται είναι ένα γεωγραφικό μήκος και πλάτος, παρ' όλα αυτά, η απεικόνιση μιας διαδρομής είναι αρκετά πιο περίπλοκη, αφού η διαδρομή αυτή πρέπει να συνενώνει διάφορα σημεία καθώς ταυτόχρονα αντιστοιχείται σε κάποιο οδικό δίκτυο. Τα σημεία αυτά θα μπορούσαν να είναι είτε η αρχή και το τέλος μιας διαδρομής, είτε μία ακολουθία από σημεία που αντιστοιχούν σε μία διαδρομή. Η πιο συχνή τακτική επίλυσης του προβλήματος είναι η αντιστοίχιση GPS σημείων με κόμβους που ανήκουν σε ένα γράφο / δίκτυο. Μία από τις καλύτερες υλοποιήσεις που λύνουν το `map - matching` πρόβλημα και είναι ανοιχτή προς όλους τους χρήστες, είναι αυτή του GraphHopper [37].

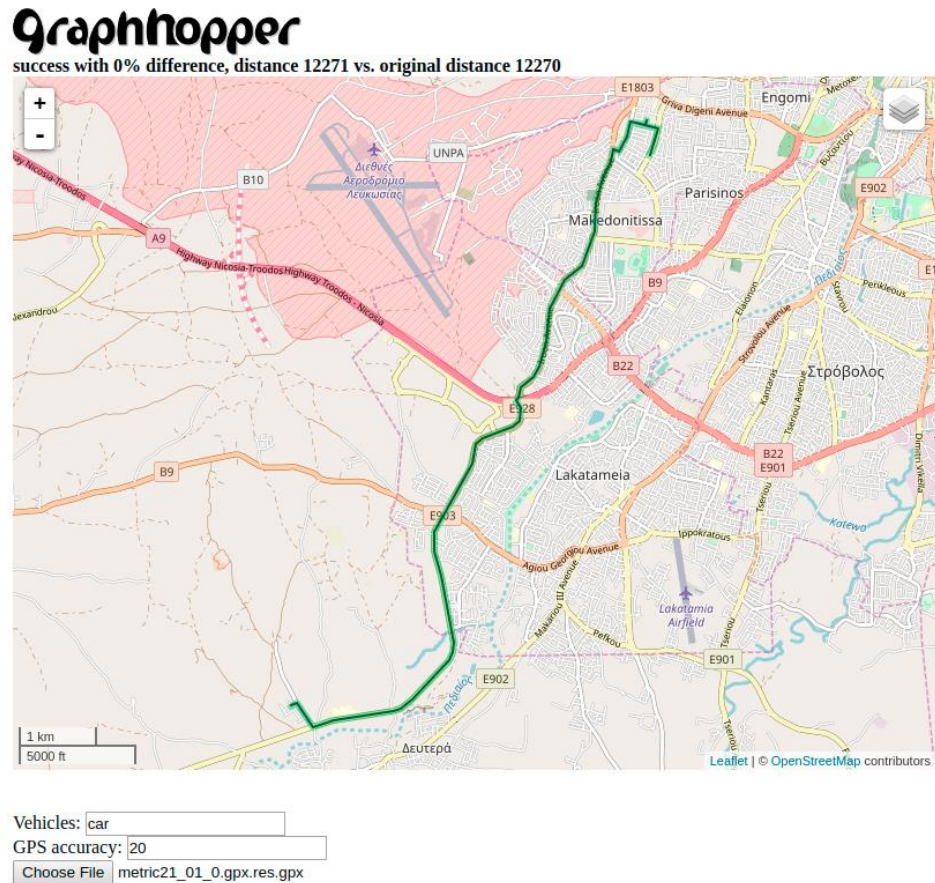
Το GraphHopper είναι μια βιβλιοθήκη δρομολόγησης ανοιχτού κώδικα (Apache License). Ταυτόχρονα, είναι και ένας διακομιστής (server) γραμμένος σε Java, που παρέχει μια διεπαφή ιστού που ονομάζεται GraphHopper Maps. Το σύστημα αυτό χρησιμοποιεί προκαθορισμένα (default) τα OpenStreetMap δεδομένα. Τα δεδομένα αυτά, χρησιμοποιούνται για την εξαγωγή του οδικού δικτύου, αλλά και των δεδομένων ανύψωσης (elevation data). Το σύστημα αυτό έχει ως απώτερο σκοπό την παρακολούθηση οχημάτων μέσω αντιστοίχισης χαρτών, την υποβοήθηση του πολεοδομικού σχεδιασμού, την προσομοίωση κυκλοφορίας και τον προσδιορισμό του δικτύου που είναι προσβάσιμο για αυτοκίνητα, πεζούς, ή και ποδήλατα. Τα προβλήματα αυτά είναι και τα προβλήματα που μελετήθηκαν κατά τη διάρκεια της παρούσας διπλωματικής. Το GraphHopper σύστημα μπορεί είτε να εγκατασταθεί και να χρησιμοποιηθεί ως διακομιστής (server) πάνω σε κάποια μηχανή, ή μπορεί να χρησιμοποιηθεί μέσω του API που προσφέρει.

Για την επίλυση του map - match προβλήματος, το GraphHopper σύστημα χρησιμοποιεί διάφορους αλγόριθμους, όπως Dijkstra, A*, και άλλα είδη αλγορίθμων διπλής κατεύθυνσης αναζήτησης (bidirectional search). Η διπλή κατεύθυνση αναζήτηση είναι ένα είδος αλγόριθμου που προσπαθεί με την χρήση κατευθυνόμενων γράφων να βρει το συντομότερο μονοπάτι μεταξύ δύο κόμβων. Ο ένας κόμβος είναι η “αρχή” της κατεύθυνσης και ο δεύτερος κόμβος αντιστοιχεί στο “τέλος” της κατεύθυνσης. Η επιλογή του αλγορίθμου αυτού έγινε λόγω της γρήγορης εκτέλεσης που ο αλγόριθμος έχει σε πολλές περιπτώσεις, όπως η αντιστοίχιση τροχιών σε χάρτη.

Γενικότερα, το σύστημα αυτό έχει βασιστεί στην έρευνα του Newson [20], όπου σχετίζεται με την εύρεση ενός αλγορίθμου που αποσκοπεί στην επίλυση του map - matching προβλήματος χρησιμοποιώντας τον Hidden Markov Model (HMM) αλγόριθμο. Συγκεκριμένα, η έρευνα προσπαθεί να αντιστοιχίσει στον χάρτη μία σειρά από σημεία που περιέχουν timestamp, γεωγραφικό μήκος και πλάτος. Ακόμη, ο Newson με τους συνεργάτες του έχουν λάβει υπόψη τους το θόρυβο (noise measurement) αλλά και τη διάταξη του οδικού δικτύου. Για κάθε είσοδο, ο αλγόριθμος προσπαθεί να βρει όλα τα πιθανά μονοπάτια που μπορούν να υπάρξουν για την σύνδεση του κάθε σημείου, όπου για κάθε σημείο υπολογίζεται ταυτόχρονα μία περίμετρος απόκλισης που μπορεί να υπήρξε κατά την συλλογή του γεωγραφικού μήκους και πλάτους μέσω του GPS.

Το GraphHopper σύστημα χρησιμοποιεί τον Viterbi αλγόριθμο και μία βιβλιοθήκη που παρέχει την υλοποίηση της έρευνας που αναφέρθηκε παραπάνω [20], για την εύρεση της καλύτερης αντιστοίχισης σε χάρτη. Ο Viterbi αλγόριθμος είναι ένας δυναμικός αλγόριθμος όπου αποσκοπεί στην εύρεση της πιο πιθανής ακολουθίας κρυφών καταστάσεων (Hidden States). Η ακολουθία αυτή χρησιμοποιείται για την καλύτερη λειτουργία του HMM αλγόριθμου.

Στο Σχήμα 6.3 φαίνεται η εφαρμογή του συστήματος όσο αφορά την σχεδίαση πάνω στον χάρτη. Στον χάρτη φαίνεται μια από τις τροχιές που εκτελέστηκαν με σκοπό την συλλογή δεδομένων. Η συλλογή δεδομένων έχει γίνει μέσω της εφαρμογής που αναφέρθηκε στο κεφάλαιο 6.1.



Σχήμα 6.3 Διεπαφή Συστήματος [37]

Κεφάλαιο 7

Πειραματική Αποτίμηση

7.1 Πειραματική Μεθοδολογία	82
7.2 Ανάλυση Αποτελεσμάτων	88
7.2.1 Ανάλυση Χρόνου Απόκρισης της TrafficWare Ιστοσελίδας	88
7.2.2 Ανάλυση Χρόνου Επεξεργασίας Δεδομένων	92

Στο κεφάλαιο αυτό θα περιγραφεί η πειραματική αποτίμηση που έγινε κατά τη διάρκεια της παρούσας διπλωματικής. Συγκεκριμένα, θα περιγραφούν τα δύο πειράματα που έγιναν, όπου το πρώτο αφορά την αξιολόγηση της TrafficWare πλατφόρμας σχετικά με την απόδοση και τους χρόνους απόκρισής της. Στο σημείο αυτό, θα συγκριθούν οι διάφοροι τρόποι απεικόνισης δεδομένων σε ένα χάρτη αλλά και οι χρόνοι απόδοσης άλλων ιστοσελίδων, όπως αυτή του OpenCellId. Το δεύτερο πείραμα αφορά την αξιολόγηση του χρόνου εκτέλεσης προγραμμάτων που γράφτηκαν σε διάφορες γλώσσες - τεχνολογίες. Σκοπός των προγραμμάτων αυτών, είναι η επεξεργασία και το φιλτράρισμα των δεδομένων. Τέλος, θα αναλυθούν τα αποτελέσματα των πειραμάτων αυτών.

7.1 Πειραματική Μεθοδολογία

Πρώτο Πείραμα

Κατά τη διάρκεια της παρούσας διπλωματικής εργασίας διεξήχθησαν δύο πειράματα. Το πρώτο πείραμα αφορά την αποτίμηση και αξιολόγηση της TrafficWare ιστοσελίδας στην αποδοτικότητά της. Συγκεκριμένα, θα αξιολογηθεί στον χρόνο απόκρισης που χρειάζεται για την απεικόνιση διάφορων δεδομένων που η ιστοσελίδα εμφανίζει βάσει

του τύπου απεικόνισης σημείων, π.χ., Heatmap, Cluster και Prune. Στο πείραμα, η ιστοσελίδα αρχικά εμφανίζει κάθε είδος δεδομένων με τον κάθε τύπο απεικόνισης ξεχωριστά, και έπειτα εμφανίζει πολλαπλά είδη δεδομένων με τη βοήθεια των συναρτήσεων Heatmap, Cluster και Prune. Με τον όρο εμφάνισης πολλαπλών ειδών δεδομένων, εννοείται ότι η ιστοσελίδα εμφανίζει αρχικά τα OpenCellId δεδομένα, μετά OpenCellId μαζί με τα MyMeasurements, έπειτα τα OpenCellId μαζί με τα MyMeasurements και τα MLS δεδομένα. Στη συνέχεια εμφανίζονται τα δεδομένα OpenCellId, MyMeasurements, MLS και CyprusCellTowers ταυτόχρονα, και τέλος, εμφανίζει όλα τα είδη δεδομένων ταυτόχρονα. Έπειτα, οι χρόνοι αυτοί θα συγκριθούν με χρόνους απόκρισης άλλων ιστοσελίδων που εμφανίζουν γεωγραφικά δεδομένα σε χάρτη. Σκοπός του πειράματος αυτού, ήταν η αξιολόγηση της ιστοσελίδας, μίας και κατά τη διάρκεια της διπλωματικής μελετήθηκαν οι διάφοροι τρόποι απεικόνισης μεγάλων γεωγραφικών δεδομένων.

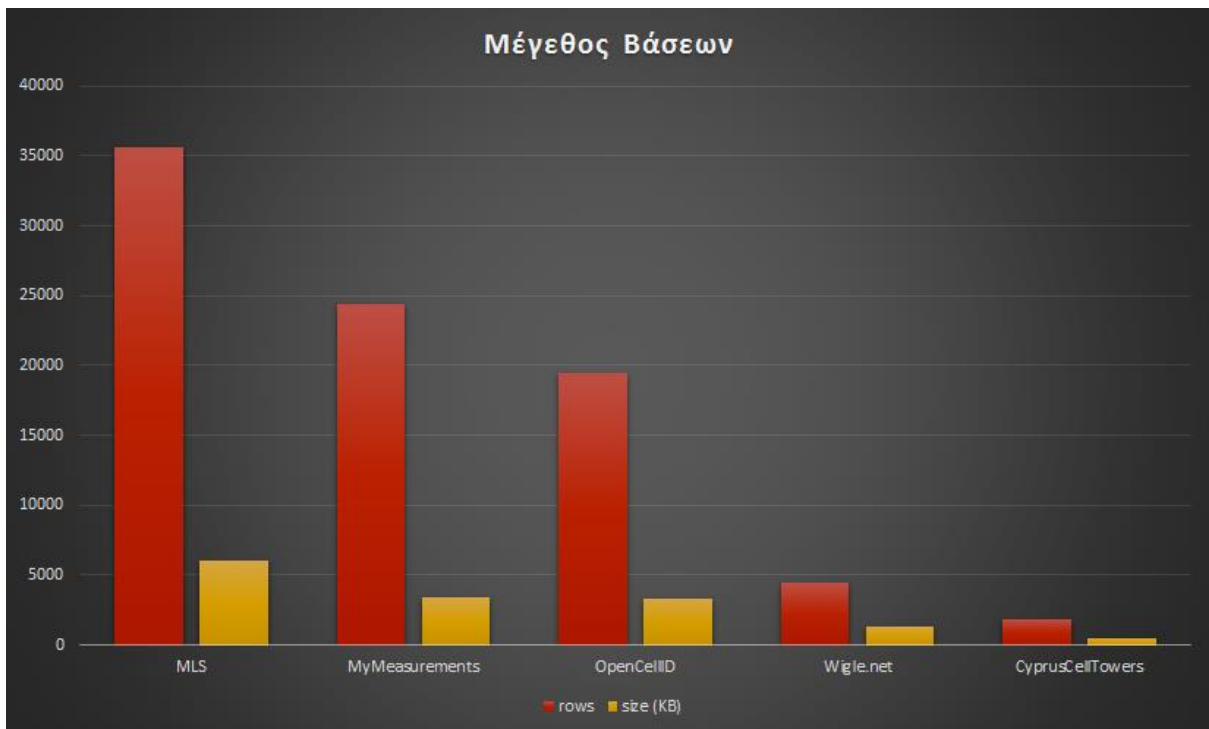
Το πρώτο πείραμα διεξήχθη πάνω σε ένα υπολογιστή με τα εξής χαρακτηριστικά:

- Ram: 8GB
- CPU: Intel Core i5-4590, 3.30GHz
- Λειτουργικό Σύστημα: Windows 7

Στο σημείο αυτό είναι αξιοσημείωτο να αναφερθεί ότι η ιστοσελίδα βρίσκεται διαθέσιμη σε συγκεκριμένο ip. Για την επίτευξη αυτού του σκοπού η ιστοσελίδα είναι στημένη πάνω σε ένα VM που λειτουργεί ως διακομιστής (server) μέσω της τεχνολογίας Apache Server. Στο VM αυτό, υπάρχει και η Postgresql βάση δεδομένων που χρειάζεται για τη λειτουργία της ιστοσελίδας. Η μηχανή αυτή διαθέτει λειτουργικό σύστημα Ubuntu 16.04, CPU Intel(R) Xeon(R) 2.40GHz, και 20GB ram.

Όπως αναφέρθηκε και στο κεφάλαιο 5, στην TrafficWare ιστοσελίδα εμφανίζονται τα εξής δεδομένα: OpenCellId, MLS, Cyprus Cell Towers, Wigle.net και οι μετρήσεις που έχουν ληφθεί μέσω της εφαρμογής συλλογής τροχιών. Όπως φαίνεται και στο Σχήμα 7.1, τα MLS δεδομένα έχουν το μεγαλύτερο ποσοστό σε εγγραφές και χώρο στη βάση δεδομένων, συγκεκριμένα υπάρχουν 35565 εγγραφές των 6064 KB. Έπειτα, το αμέσως μεγαλύτερο ποσοστό στη βάση δεδομένων το έχουν οι μετρήσεις που έχουν γίνει είτε

από την εφαρμογή που υλοποιήθηκε κατά τη διάρκεια της διπλωματικής, είτε από τα ακατέργαστα (raw) δεδομένα που ανακτήθηκαν από το Wigle.net. Οι μετρήσεις αυτές έχουν 24385 εγγραφές στην βάση δεδομένων, δεσμεύοντας έτσι 3416KB. Ακόμη, υπάρχουν τα OpenCellId που είναι οι υπολογιζόμενες θέσεις των κεραιών, έχοντας έτσι στη βάση 19427 εγγραφές των 3312 KB. Τα Wigle.net δεδομένα είναι επίσης οι υπολογιζόμενες θέσεις των κεραιών, με 4463 εγγραφές των 1352KB. Τέλος, τα CyprusCellTowers είναι οι καταγεγραμμένες θέσεις των κεραιών από την κυβέρνηση, έχοντας στη βάση 1837 εγγραφές των 520KB. Είναι αξιοσημείωτο να αναφερθεί ότι τα δεδομένα αυτά αναφέρονται αποκλειστικά για την Κύπρο, και γι' αυτόν ακριβώς τον λόγο, το μέγεθος των δεδομένων συνολικά δεν ξεπερνά τα 15MB.



Σχήμα 7.1 Μέγεθος Βάσης

Το μέγεθος που αναφέρεται πιο πάνω, εμπερικλείει και το μέγεθος των indexes που έχει ο κάθε πίνακας. Για κάθε πίνακα εφαρμόστηκαν indexes πάνω στα γεωγραφικά μήκη

και πλάτη. Τα indexes εφαρμόστηκαν έτσι ώστε να γίνει γρηγορότερη η ανάκτηση δεδομένων από τον κάθε πίνακα.

Στον OpenCellId και MLS πίνακα το κάθε στοιχείο έχει τον εξής τύπο:

- radio: character varying(6) NOT NULL
- mcc: integer
- net: integer
- area: integer
- cell: integer
- unit: integer
- lon: numeric NOT NULL
- lat: numeric NOT NULL
- range: integer
- samples: integer
- changeable: integer
- created: integer
- updated: integer
- averagesignal: integer

Για τον CyprusCellTowers πίνακα το κάθε στοιχείο έχει τον ακόλουθο τύπο:

- coding: text
- name: text
- address: text
- latitude: numeric NOT NULL
- longitude: numeric NOT NULL
- gsm: integer
- dcs: integer
- antenna_height: text
- services: text
- site_installation: text
- area: text
- colocation: text

Ο MyMeasurements πίνακας περιέχει τα εξής στοιχεία με τους εξής τύπους:

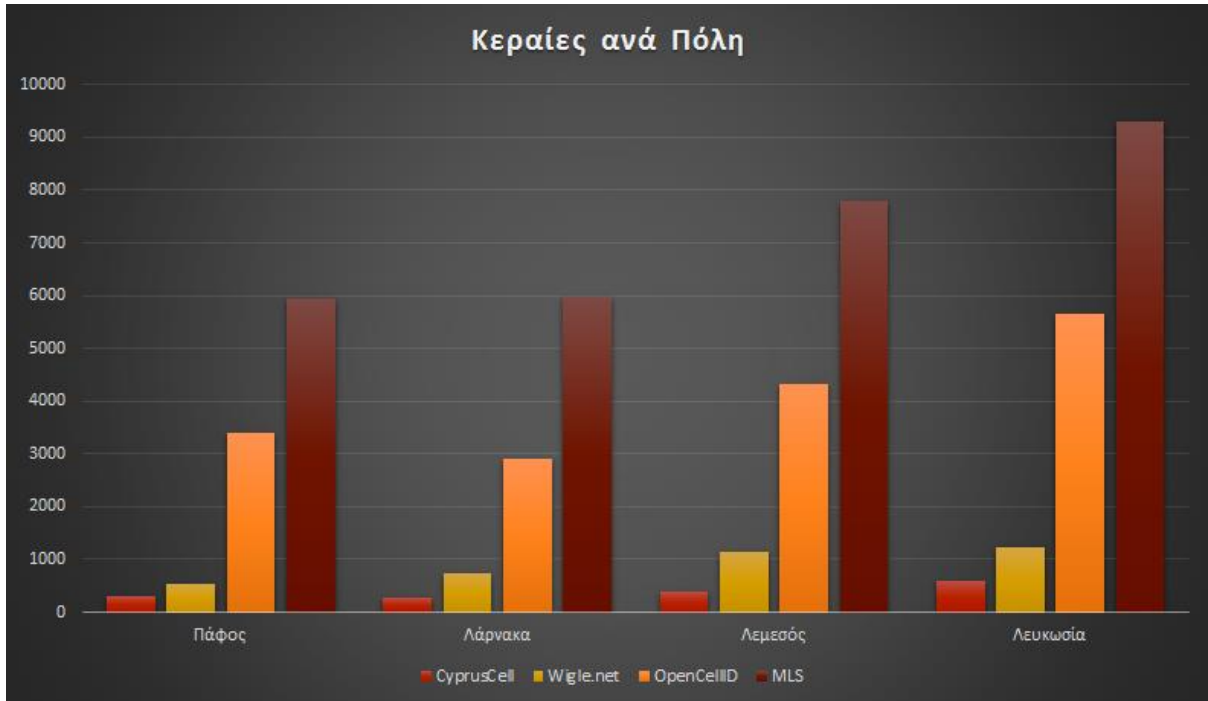
- timestamp: bigint
- lon: numeric NOT NULL
- lat: numeric NOT NULL
- type: character varying(6)
- signal: integer
- cellid: bigint
- mnc: integer
- lac: integer

Τέλος, ο Wigle.net πίνακας έχει τα εξής χαρακτηριστικά:

- lat: numeric NOT NULL
- lon: numeric NOT NULL
- ssid: text
- qos: integer
- transit: text
- firsttime: text
- lasttime: text
- lastupdt: text
- housenumber: text
- road: text
- city: text
- region: text
- country: text
- id: text
- attributes: character varying(10)
- gentype: character varying(6)

Υπήρξαν κάποια στατιστικά σχετικά με το ποσοστό κεραιών που υπάρχουν σε κάθε πόλη. Όπως φαίνεται και στο πιο κάτω Σχήμα (Σχήμα 7.2), το μεγαλύτερο ποσοστό κεραιών από όλα τα είδη δεδομένων που συλλέχθηκαν είναι στη Λευκωσία. Έπειτα, το αμέσως επόμενο μεγάλο ποσοστό είναι η Λεμεσός, μετά η Λάρνακα, και τέλος η Πάφος. Αυτό ήταν αναμενόμενο, μιας και η Λευκωσία είναι η πρωτεύουσα της Κύπρου και ταυτόχρονα η πιο πυκνοκατοικημένη πόλη. Όσο πιο πολλές συσκευές πιθανόν να

υπάρχουν σε μια πόλη, τόσο πιο μεγάλη είναι η ανάγκη περισσότερων κεραιών για κάλυψη των συσκευών αυτών.



Σχήμα 7.2 Κεραίες ανά Πόλη

Δεύτερο Πείραμα

Όπως προαναφέρθηκε, διεξήχθησαν δύο πειράματα. Στο δεύτερο πείραμα, γράφτηκαν τέσσερα προγράμματα που αποσκοπούν στην ίδια λειτουργία. Τα προγράμματα αυτά, αρχικά διαβάζουν από ένα αρχείο που περιέχει όλες τις εγγραφές που υπάρχουν στη βάση δεδομένων σχετικά με τα MyMeasurements. Έπειτα, τα προγράμματα επεξεργάζονται την κάθε εγγραφή μετατρέποντας τις, είτε σε μία συμβολοσειρά που περιέχει CellId και Lac, είτε σε μια πλειάδα που περιέχει τα δύο αυτά χαρακτηριστικά. Στη συνέχεια, τα δεδομένα αυτά περνούν από ένα φιλτράρισμα. Συγκεκριμένα, το φιλτράρισμα αυτό, αγνοεί όσες εγγραφές δεν περιέχουν Lac ίσο με 231. Τέλος, τα προγράμματα διαγράφουν πιθανόν διπλότυπα που μπορεί να υπάρχουν και τυπώνουν το αποτέλεσμα στην κονσόλα.

Τα προγράμματα έχουν γραφτεί σε Spark Java, απλή Java, NodeJs και PHP. Στην πρώτη περίπτωση, το πρόγραμμα έχει γραφτεί σε Java χρησιμοποιώντας τις βιβλιοθήκες της Spark, έτσι ώστε κατά τη διάρκεια του πειράματος να εκτελεστεί μέσω του Spark. Οι υπόλοιπες περιπτώσεις έχουν γραφτεί με τις τεχνολογίες αυτές έτσι ώστε να μπορούν να θεωρηθούν ως τρόποι φιλτραρίσματος και επεξεργασίας δεδομένων από την πλευρά του διακομιστή (server).

Οι εκτελέσεις των προγραμμάτων έχουν διεξαχθεί στο VM που χρησιμοποιήθηκε και για το πρώτο πείραμα. Το VM αυτό, διαθέτει λειτουργικό σύστημα Ubuntu 16.04, CPU Intel(R) Xeon(R) 2.40GHz, και 20GB ram και λειτουργεί ως server μέσω της τεχνολογίας Apache Server. Στο σημείο αυτό, είναι σημαντικό να αναφερθεί ότι στην περίπτωση του Spark προγράμματος, το VM λειτούργησε με Standalone mode, δηλαδή λειτούργησε και ως master και ως slave. Το πρόγραμμα αυτό εκτελέστηκε ως δουλειά (job) που ανατέθηκε στον slave.

Κατά τη διάρκεια εκτέλεσης των προγραμμάτων αυτών, έχει μετρηθεί ο χρόνος που χρειάζονται για την επίτευξη της λειτουργίας αυτής. Σκοπός του πειράματος, ήταν η σύγκριση των χρόνων εκτέλεσης.

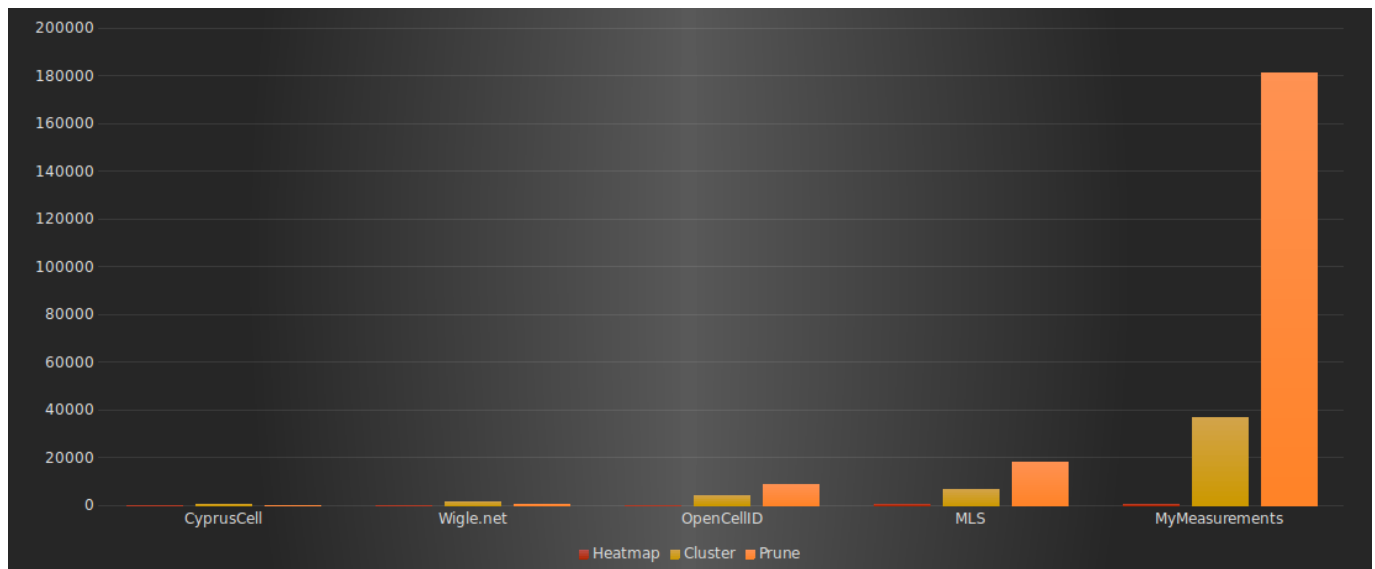
7.2 Ανάλυση Αποτελεσμάτων

Στην ενότητα αυτή θα αναφερθούν και αναλυθούν τα αποτελέσματα των δύο πειραμάτων. Ακόμη, θα επεξηγηθούν οι λόγοι - αιτίες που τα αποτελέσματα παίρνουν τις συγκεκριμένες μορφές / τιμές.

7.2.1 Ανάλυση Χρόνου Απόκρισης της TrafficWare Ιστοσελίδας

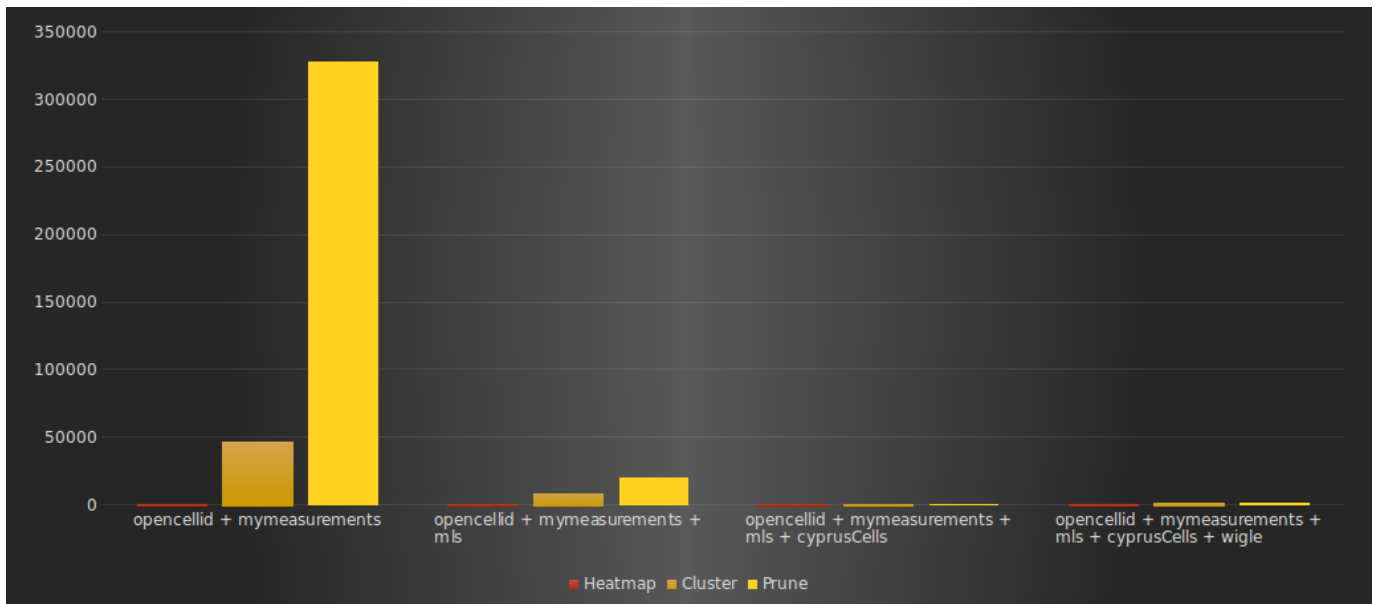
Στο Σχήμα 7.3 φαίνονται οι χρόνοι που χρειάστηκαν τα δεδομένα να φορτωθούν στον χάρτη (loading time) για κάθε είδος δεδομένων με κάθε είδος τύπου απεικόνισης. Όπως

φαίνεται, το Prune σε όλες τις περιπτώσεις χρειάστηκε τον περισσότερο χρόνο για την εμφάνιση των δεδομένων. Απ' την άλλη, το heatmap χρειάστηκε τον λιγότερο χρόνο, αφού σε εξαιρετικό χρόνο εμφανίζει γρήγορα και διαδραστικά τα δεδομένα. Ακόμη, παρατηρήθηκε ότι γενικότερα τα MyMeasurements χρειάστηκαν τον περισσότερο χρόνο απεικόνισης σε σχέση με τα άλλα είδη δεδομένων. Έπειτα, τον μεγαλύτερο χρόνο απόκρισης είχαν τα MLS δεδομένα, μετά τα OpenCellId, στη συνέχεια τα Wigle.net, και τέλος τα CyprusCellTowers. Σε όλες τις περιπτώσεις, εκτός από τα MyMeasurements, παρατηρείται ότι η σειρά για μεγαλύτερο χρόνο απόκρισης σχετίζεται με το μέγεθος των δεδομένων του κάθε είδους. Στο σημείο αυτό, είναι σημαντικό να αναφερθεί ότι η ιστοσελίδα φορτώνει μόνο τα δεδομένα που είναι διαθέσιμα για τα πλακάκια (tiles) που είναι επιλεγμένα στον χάρτη, έτσι είναι λογικό τα MyMeasurements να θέλουν τον περισσότερο χρόνο, μιας και κατά το πείραμα στον χάρτη ήταν επιλεγμένη η Λευκωσία. Τα MyMeasurements περιέχουν δεδομένα που αφορούν μόνο την Λευκωσία, ενώ τα άλλα είδη δεδομένων περιέχουν δεδομένα για ολόκληρη την Κύπρο.



Σχήμα 7.3 Ms Φόρτωσης Δεδομένων

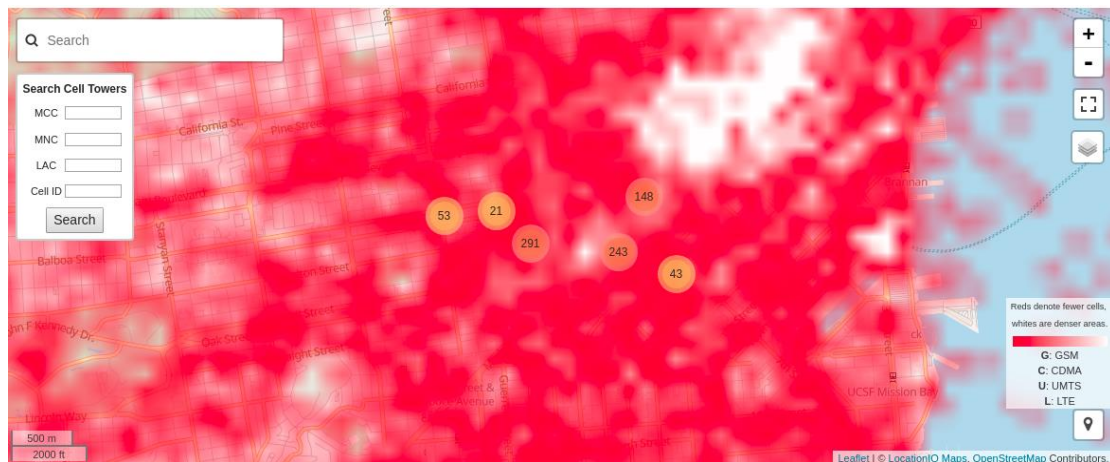
Έπειτα, μελετήθηκε ο χρόνος που χρειάστηκε για την απεικόνιση συνδυασμένων ειδών δεδομένων. Όπως φαίνεται και στο Σχήμα 7.4, η απεικόνιση των MyMeasurements χρειάστηκε τον περισσότερο χρόνο για την εμφάνιση των δεδομένων. Οι παρατηρήσεις είναι οι ίδιες με το προηγούμενο πείραμα, βλέποντας ότι το Prune χρειάστηκε τον περισσότερο χρόνο, μετά το Cluster και τέλος το Heatmap.



Σχήμα 7.4 Ms Φόρτωσης Δεδομένων

Στη συνέχεια, μελετήθηκε ο χρόνος που χρειάστηκε για την φόρτωση των δεδομένων στην OpenCellId ιστοσελίδα, MLS, και τέλος Wigle.net. Στην OpenCellID ιστοσελίδα όταν ο χάρτης βρίσκεται σε ένα αρκετά κοντινό zoom level χρησιμοποιεί την μέθοδο Cluster και για πιο μακρινά zoom levels χρησιμοποιεί heatmap. Την τακτική αυτή χρησιμοποιούν οι περισσότερες ιστοσελίδες όταν θέλουν να εμφανίσουν μεγάλα δεδομένα στον χάρτη. Παρατηρήθηκε ότι για το Cluster, χρειάζονται γύρω στα 2030ms για να φορτωθούν τα δεδομένα, ενώ για το heatmap η φόρτωση γίνεται απευθείας μαζί με την φόρτωση των tiles του χάρτη. Αυτό συμβαίνει επειδή φορτώνουν τον χάρτη ως εικόνες ανά πλακάκι (tile) και πάνω στην κάθε εικόνα υπάρχει ο χάρτης μαζί με το heatmap ζωγραφισμένο. Με τον τρόπο αυτό κερδίζεται χρόνος απόκρισης. Παρ' όλα

αυτά, το OpenCellId δεν φρόντισε να έχει τις κατάλληλες φωτογραφίες για διάφορα zoom levels, με αποτέλεσμα σε κάποια σημεία το heatmap να εμφανίζεται κάπως θολό, κάτι που φαίνεται και στο Σχήμα 7.5.



Σχήμα 7.5 OpenCellId Heatmap [44]

Οι ιστοσελίδες MLS και Wigle.net χρησιμοποιούν μόνο τη μέθοδο heatmap για την εμφάνιση των δεδομένων. Η MLS ιστοσελίδα χρειάζεται γύρω στα 3360ms για την φόρτωση του heatmap, και η Wigle.net ιστοσελίδα χρειάζεται μεταξύ των 6880-9740ms. Στην περίπτωση της Wigle.net ιστοσελίδας χρησιμοποιείται και εκεί η τακτική φόρτωσης του heatmap ως εικόνα μαζί με τον χάρτη. Ο λόγος που χρειάζεται περισσότερο χρόνο από το OpenCellId είναι επειδή για κάθε επίπεδο του χάρτη υπάρχουν και οι αντίστοιχες εικόνες για καλύτερη ποιότητα του χάρτη, άρα σε κάθε επίπεδο του χάρτη η ιστοσελίδα πρέπει να ανακτήσει και τις κατάλληλες εικόνες. Απ' την άλλη, η MLS ιστοσελίδα χρειάζεται λιγότερο χρόνο μιάς και ο χάρτης που χρησιμοποιεί είναι μικρότερος σε μέγεθος - διαστάσεις, από αυτόν που υπάρχει στις ιστοσελίδες Wigle.net, OpenCellId και TrafficWare. Στην MLS πλατφόρμα, φαίνεται να εφαρμόστηκε η ίδια τακτική εμφάνισης του heatmap ως εικόνα, ενώ ταυτόχρονα υπάρχει μια επιπλέον μικρή καθυστέρηση λόγω του ότι εφαρμόζονται κάποιες μικροαλλαγές αφότου φορτωθούν τα πλακάκια του χάρτη. Συγκεκριμένα, εμφανίζονται τα ονόματα των πόλεων, ενώ έπειτα φορτώνεται σε καλύτερη ποιότητα το heatmap. Εν κατακλείδι, οι ιστοσελίδες συγκριτικά, όσον αφορά τον χρόνο απόκρισης για την ίδια

μέθοδο απεικόνισης δεδομένων (Cluster / Heatmap), φαίνονται να έχουν χρόνους απόκρισης παρόμοιους ή και μεγαλύτερους από αυτούς που έχει η TrafficWare ιστοσελίδα.

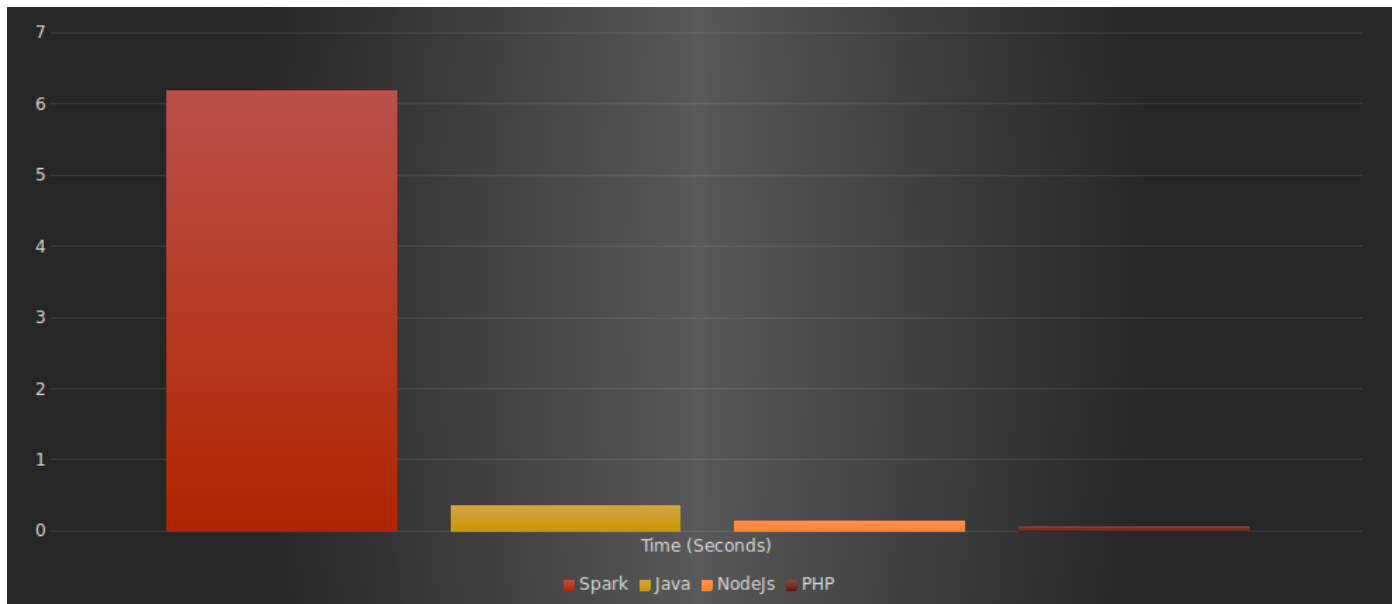
7.2.2 Ανάλυση Χρόνου Επεξεργασίας Δεδομένων

Στο πείραμα αυτό εξετάστηκε ο χρόνος που χρειάζεται να εκτελεστεί ένα πρόγραμμα χρησιμοποιώντας διάφορες τεχνολογίες. Συγκεκριμένα, το πρόγραμμα διαβάζει από ένα αρχείο που περιέχει όλα τα δεδομένα MyMeasurements και τα φιλτράρει. Όπως αναφέρθηκε και σε διάφορα άλλα σημεία, τα MyMeasurements δεδομένα περιέχουν τις μετρήσεις που συλλέχθηκαν κατά την εκτέλεση διαφόρων διαδρομών στην Λευκωσία, αλλά και μετρήσεις που συλλέχθηκαν από χρήστες που χρησιμοποιούν την Wigle εφαρμογή. Τα δεδομένα αυτά περιέχουν τα εξής δεδομένα: Timestamp, Lat, Lon, Type, Signal, CellId, Mnc, και Lac. Στο πείραμα αυτό, το πρόγραμμα αρχικά διαβάζει 24385 εγγραφές, και έπειτα δημιουργεί είτε συμβολοσειρές που περιέχουν το CellId και το Lac, είτε δημιουργεί πλειάδες (Tuples) που περιέχουν τα δύο αυτά χαρακτηριστικά. Στη συνέχεια, τα δεδομένα περνούν από ένα φιλτράρισμα, κρατώντας έτσι μόνο τις συμβολοσειρές / πλειάδες που περιέχουν LAC ίσο με 231. Πριν το τέλος, τα δεδομένα αυτά γίνονται διακριτά (distinct), δηλαδή γίνεται η διαγραφή διπλότυπων. Στο τέλος, το αποτέλεσμα τυπώνεται στην κονσόλα.

Όπως φαίνεται και στο Σχήμα 7.6, ο περισσότερος χρόνος εκτέλεσης γίνεται από το Spark, αφού για την έναρξη της δουλειάς (job) και την εκτέλεση του προγράμματος χρειάστηκε έως και 6,186 δευτερόλεπτα. Έπειτα, ο κώδικας που είναι γραμμένος σε Java έχει τον δεύτερο μεγαλύτερο χρόνο εκτέλεσης. Στη συνέχεια, παρατηρείται το NodeJs με χρόνο εκτέλεσης 0,145 δευτερολέπτων, και τέλος μέσω PHP με χρόνο απόκρισης 0.066".

Είναι σημαντικό να σημειωθεί ότι εκτέλεση των προγραμμάτων έγινε από έναν μόνο υπολογιστικό πόρο - VM. Στην περίπτωση του Spark, το VM λειτούργησε με Standalone mode, δηλαδή το VM ήταν ο master αλλά ο slave ταυτοχρόνως. Γι' αυτόν

ακριβώς τον λόγο είναι πολύ λογικό να χρειάζεται το μεγαλύτερο χρόνο εκτέλεσης, αφού πρέπει να ανατεθεί η δουλειά στον slave και έπειτα να εκτελεστεί το πρόγραμμα χρησιμοποιώντας RDDs. Απ' την άλλη, οι υπόλοιποι τρόποι εκτέλεσης επεξεργάζονται τα δεδομένα ως συμβολοσειρές, με αποτέλεσμα την ανάγκη λιγότερου χρόνου εκτέλεσης. Εάν το πείραμα επρόκειτο να εκτελεστεί σε κατανεμημένους (distributed) υπολογιστές, πιο πιθανό το spark να χρειαζόταν λιγότερο χρόνο σε σχέση με τις υπόλοιπες τεχνολογίες. Επίσης, είναι αξιοσημείωτο να αναφερθεί ότι το Spark είναι φτιαγμένο έτσι ώστε να λειτουργεί καλύτερα σε περιπτώσεις που η επεξεργασία των δεδομένων μπορεί να γίνει παράλληλα σε πολλούς πόρους παρά σε έναν σειριακά.



Σχήμα 7.6 Φιλτράρισμα Δεδομένων Χρησιμοποιώντας Διάφορες Τεχνολογίες

Κεφάλαιο 8

Συμπεράσματα και Μελλοντική Εργασία

8.1 Συμπεράσματα	94
8.2 Μελλοντική Εργασία	96

8.1 Συμπεράσματα

Στα πλαίσια της παρούσας διπλωματικής εργασίας, παρουσιάστηκε η ιδέα ενός συστήματος που αποσκοπεί στην εύρεση και ανάλυση της οδικής κυκλοφορίας έχοντας παράλληλα τρία βασικά κριτήρια. Τα κριτήρια αυτά είναι η επίλυση του προβλήματος διατηρώντας την ιδιωτικότητα της θέσης των χρηστών, χρησιμοποιώντας την υπάρχουσα υποδομή (π.χ., κεραίες) και υλοποιώντας το σύστημα με ελάχιστο κόστος. Για την επίτευξη των κριτηρίων αυτών, οι πηγές εισόδου που το σύστημα θα παίρνει είναι είτε ανοικτά δεδομένα, είτε από προϋπάρχων υποδομή, π.χ., κεραίες κυψέλης. Όσον αφορά την επικράτηση της ιδιωτικότητας των χρηστών, στα δεδομένα θα εφαρμόζονται επεκτάσεις του αλγορίθμου k - anonymity, έτσι ώστε να παραμένει η ταυτότητα των χρηστών κρυφή ή θα προστίθεται σφάλμα στα δεδομένα. Βέβαια, με την προσθήκη σφάλματος υπάρχει ο κίνδυνος μείωσης ακρίβειας του τελικού αποτελέσματος.

Για την υλοποίηση του συστήματος προτάθηκαν δύο βασικές ιδέες. Το πρώτο μοντέλο αποτελεί ένα σύστημα που χρησιμοποιεί δεδομένα δικτύου και βασίζεται στη χρησιμοποίηση PCHR αρχείων, υποθέτοντας ότι τα αρχεία περιέχουν την τοποθεσία χρηστών, να εξάγονται οι θέσεις (location) αυτές και να αντιστοιχίζονται πάνω στον χάρτη. Έπειτα, βάσει της ταχύτητας που οι χρήστες κινούνται, να υπολογίζεται η κυκλοφορία στους δρόμους. Συγκεκριμένα, το σύστημα θα υπολογίζει το πιο σύντομο

μονοπάτι που θα μπορούσε ο χρήστης να έχει ακολουθήσει από την τελευταία γνωστή τοποθεσία του στην καινούργια τοποθεσία. Ακόμη, για την εύρεση των μονοπατιών αυτών, θα ληφθεί υπόψη η ταχύτητα του χρήστη έτσι ώστε να μην θεωρηθεί πιθανό κάποιο μονοπάτι το οποίο δεν είχε αρκετό χρόνο ο χρήστης να ακολουθήσει βάσει της ταχύτητας που δικαιούται να οδηγεί.

Το σύστημα που χρησιμοποιεί πληθοπορισμό βασίζεται στην ίδια λογική, με τη διαφορά ότι οι χρήστες θα στέλνουν στο σύστημα τις αντένες που η συσκευή τους ανιχνεύει, και έπειτα το σύστημα θα υπολογίζει την τοποθεσία του χρήστη βάσει της πληροφορίας αυτής. Το σύστημα αρχικά, θα έχει περάσει από μία εκπαιδευτική φάση, έτσι ώστε να μπορεί να μπορεί να ταξινομήσει ποια σημεία εξυπηρετεί η κάθε κεραία. Έτσι, βάσει των θέσεων των χρηστών και της ταχύτητας που κινούνται, το σύστημα θα είναι σε θέση να υπολογίζει τις τροχιές των χρηστών, με απώτερο σκοπό την εύρεση οδικής κίνησης.

Κατά την προσπάθεια επίλυσης του προβλήματος εύρεσης οδικής κίνησης μπορεί εύκολα να υπάρξουν θέματα ακρίβειας, και γι' αυτόν ακριβώς τον λόγο, για την πλήρη υλοποίηση του συστήματος θα προκύψουν ιδιαίτερες δυσκολίες σε θέματα ακρίβειας. Για το πρώτο μοντέλο του συστήματος – το σύστημα που χρησιμοποιεί δεδομένα δικτύου -, αυτό συμβαίνει λόγω της άγνωστης δομής του PCHR αρχείου, ενώ για το δεύτερο μοντέλο του συστήματος – το σύστημα που χρησιμοποιεί πληθοπορισμό -, μπορούν να υπάρξουν θέματα ακρίβειας λόγω της μη σταθερής δομής περιμέτρου των κεραιών. Κάθε κεραία μπορεί να περιέχει από ένα μέχρι και έξι τομείς, όπου ο κάθε τομέας λειτουργεί σαν μία ξεχωριστή κεραία, αφού έχουν διαφορετικό cellid, κάτι που φαίνεται και στο Σχήμα 8.1.1. Έτσι, η αποτίμηση της ακριβούς περιμέτρου που μπορεί να καλύπτει η κεραία, δεν είναι πάντοτε σταθερή. Αυτό, έχει ως αποτέλεσμα να αδυνατεί το σύστημα να βρει την ακριβή θέση του χρήστη.

Όποιο μοντέλο συστήματος υλοποιηθεί, θα υπάρξουν επιπλέον προβλήματα ακρίβειας, εφόσον οι συσκευές δεν συνδέονται πάντα με την κοντινότερη κεραία σε φυσικό επίπεδο. Συνδέονται, όμως, με την κεραία που έχει το ισχυρότερο σήμα, χωρίς να σημαίνει ότι η κεραία αυτή είναι σε όλες τις περιπτώσεις η κοντινότερη. Άρα, οι

αλγόριθμοι που θα αποσκοπούν στην εύρεση της θέσης του χρήστη, θα έβγαζαν λανθασμένα αποτελέσματα.

Επιπρόσθετα, το σύστημα θα αδυνατεί να μελετήσει την οδική κυκλοφορία στους αυτοκινητόδρομους, αφού οι δρόμοι αυτοί εξυπηρετούνται από Macro Cells. Τα Macro Cells έχουν την δυνατότητα κάλυψης αποστάσεων των 1 - 30km, κάνοντας το εξαιρετικά δύσκολο του συστήματος να υπολογίζει την θέση των χρηστών στην περίπτωση που το σύστημα υλοποιηθεί με την χρήση πληθοπορισμού.

Όσον αφορά τα πειράματα που έγιναν κατά τη διάρκεια της παρούσας εργασίας, είναι σημαντικό να ξανά αναφερθεί ότι φάνηκε πως η TrafficWare πλατφόρμα μπορεί να συναγωνιστεί σε χρόνο άλλες ιστοσελίδες που αποσκοπούν στην απεικόνιση γεωγραφικών δεδομένων σε χάρτη, αφού έχουν τον ίδιο ή μεγαλύτερο χρόνο απόκρισης σε σχέση με την ιστοσελίδα TrafficWare. Ακόμη, στα πειράματα αποδείχτηκε ότι η πιο γρήγορη τεχνολογία επεξεργασίας και φιλτραρίσματος δεδομένων είναι η PHP. Παρ' όλα αυτά είναι αξιοσημείωτο να αναφερθεί ότι παράλληλη επεξεργασία δεδομένων σε πολλαπλές μηχανές είναι η ιδανικότερη, και η ιδανικότερη τεχνολογία για τη λειτουργία αυτή είναι η Apache Spark.

8.2 Μελλοντική Εργασία

Καταρχάς, η παρούσα διπλωματική θα μπορούσε να επεκταθεί στο μέλλον υλοποιώντας το σύστημα πλήρως. Η πιο εύκολη επιλογή, θα ήταν η υλοποίηση του συστήματος που χρησιμοποιεί πληθοπορισμό, το οποίο βασίζεται στην εύρεση και ανάλυση της οδικής κυκλοφορίας καθώς οι χρήστες στέλνουν στο σύστημα τις κεραίες που η συσκευή τους ανιχνεύει. Ακολούθως, το σύστημα θα υπολογίζει την τοποθεσία του χρήστη βάσει των κεραιών που η συσκευή ανιχνεύει. Για να υπολογίζει την τοποθεσία του χρήστη, το σύστημα θα περάσει στο αρχικό του στάδιο από μία εκπαίδευση, έτσι ώστε να μπορεί να ταξινομήσει ποια σημεία εξυπηρετεί η κάθε κεραία. Έτσι, το σύστημα θα είναι σε θέση να υπολογίζει τις θέσεις και τροχιές των χρηστών βάσει των πληροφοριών που ήδη γνωρίζει και βάσει της ταχύτητας που

δικαιούνται οι χρήστες να οδηγούν. Έπειτα, το σύστημα θα μπορεί να βρίσκει σε ποια σημεία υπάρχει οδική κίνηση.

Ακόμη, η ιδέα του συστήματος θα μπορούσε να επεκταθεί, ούτως ώστε να προβλέπει την κίνηση στους δρόμους για συγκεκριμένα χρονικά ωράρια. Για να πραγματοποιηθεί αυτό, το σύστημα θα πρέπει να αναλύει για κάθε ώρα την οδική κυκλοφορία. Έτσι, ταυτόχρονα, το σύστημα θα παρέχει τη δυνατότητα παρατήρησης της οδικής κυκλοφορίας για διάφορες χρονικές στιγμές.

Τέλος, όταν το σύστημα φτάσει στην τελική του φάση, θα μπορούσαν να γίνουν κάποια πειράματα, κατά τα οποία θα συγκρίνεται η απόδοση του συστήματος με άλλα προϋπάρχων συστήματα που έχουν τον ίδιο σκοπό και λειτουργία.

Βιβλιογραφία

- [1] M. A. Abbasoğlu, B. Gedik, and H. Ferhatosmanoğlu, “Aggregate profile clustering for telco analytics”, Proc. VLDB Endow., Aug. 2013.
- [2] S. Bhattacharjee, A. Chavan, S. Huang, A. Deshpande, and A. Parameswaran, “Principles of dataset versioning: Exploring the recreation/storage tradeoff”, Proceedings of the VLDB Endowment, 2015.
- [3] T. Bicer, J. Yin, D. Chiu, G. Agrawal, and K. Schuchardt, “Integrating online compression to accelerate large-scale data analytics applications”, in Parallel & Distributed Processing (IPDPS), 2013 IEEE 27th International Symposium on. IEEE, 2013.
- [4] E. Bouillet, R. Kothari, V. Kumar, L. Mignet, S. Nathan, A. Ranganathan, D. S. Turaga, O. Udrea, and O. Verscheure, “Processing 6 billion cdns/day: From research to production (experience report),” in Proceedings of the 6th ACM Intl. Conference on Distributed Event-Based Systems, ser. DEBS, 2012.
- [5] L. Braun, T. Etter, G. Gasparis, M. Kaufmann, D. Kossmann, D. Widmer, A. Avituz, A. Iliopoulos, E. Levy, and N. Liang, “Analytics in motion: High performance event-processing and real-time analytics in the same database”, in Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD. New York, NY, USA: ACM, 2015.
- [6] M. Burtscher and P. Ratanaworabhan, “Fpc: A high-speed compressor for double-precision floating-point data”, IEEE Transactions on Computers, 2009.
- [7] C. Costa, G. Chatzimilioudis, D. Zeinalipour-Yazti, M. F. Mokbel, “Efficient Exploration of Telco Big Data with Compression and Decaying”, in IEEE 33rd Intl. Conference on Data Engineering, 2017.

- [8] C. Costa, G. Chatzimilioudis, D. Zeinalipour-Yazti, M. F. Mokbel, “Towards Real-Time Road Traffic Analytics using Telco Big Data”, in Proceedings of the 11th BIRTE Conference on Real-Time Business Intelligence and Analytics, 2017.
- [9] F. Douglass and A. Iyengar, “Application-specific delta-encoding via resemblance detection”, in USENIX Annual Technical Conference, General Track, 2003.
- [10] Q. Ho, W. Lin, E. Shaham, S. Krishnaswamy, T. A. Dang, J. Wang, I. C. Zhongyan, and A. She-Nash, “A distributed graph algorithm for discovering unique behavioral groups from large-scale telco data”, in Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, ser. CIKM. New York, NY, USA: ACM, 2016.
- [11] X. Hu, M. Yuan, J. Yao, Y. Deng, L. Chen, Q. Yang, H. Guan, and J. Zeng, “Differential privacy in telco big data platform”, Proc. VLDB Endow., Aug. 2015.
- [12] Y. Huang, F. Zhu, M. Yuan, K. Deng, Y. Li, B. Ni, W. Dai, Q. Yang, and J. Zeng, “Telco churn prediction with big data”, in Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD. New York, NY, USA: ACM, 2015.
- [13] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. K. Miu, E. Shih, H. Balakrishnan, S. Madden, “CarTel: A Distributed Mobile Sensor Computing System”, In Proceedings of the 4th ACM SenSys Conference on Embedded Networked Sensor Systems, 2006.
- [14] A. P. Iyer, L. E. Li, and I. Stoica, “Celliq: Real-time cellular network analytics at scale”, in Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation, ser. NSDI’15. Berkeley, CA, USA: USENIX Association, 2015.

- [15] J. Jenkins, I. Arkatkar, S. Lakshminarasimhan, D. A. Boyuka II, E. R. Schendel, N. Shah, S. Ethier, C.-S. Chang, J. Chen, H. Kolla et al., “Alacrity: Analytics driven lossless data compression for rapid in-situ indexing, storing, and querying”, in Transactions on Large-Scale Data and Knowledge-Centered Systems X. Springer, 2013.
- [16] S. Lakshminarasimhan, N. Shah, S. Ethier, S. Klasky, R. Latham, R. Ross, and N. F. Samatova, “Compressing the incompressible with isabela: In-situ reduction of spatio-temporal data”, in European Conference on Parallel Processing. Springer, 2011.
- [17] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. 2007. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In Proceedings of the 23rd International Conference on Data Engineering (ICDE’07). IEEE, Istanbul, Turkey.
- [18] C. Luo, J. Zeng, M. Yuan, W. Dai, and Q. Yang, “Telco user activity level prediction with massive mobile broadband data”, ACM Trans. Intell. Syst. Technol., May 2016.
- [19] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. 2007. L-diversity: Privacy Beyond K-anonymity. ACM Trans. Knowl.
- [20] Newson, Paul, and John Krumm. "Hidden Markov map matching through noise and sparseness." Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, 2009.
- [21] E. R. Schendel, Y. Jin, N. Shah, J. Chen, C.-S. Chang, S.-H. Ku, S. Ethier, S. Klasky, R. Latham, R. Ross et al., “Isobar preconditioner for effective and high throughput lossless data compression”, in IEEE 28th Intl. Conference on Data Engineering, 2012.

- [22] E. Soroush and M. Balazinska, “Time travel in a scientific array database”, in Data Engineering (ICDE), 2013 IEEE 29th International Conference on. IEEE, 2013.
- [23] Arvind Thiagarajan, Lenin Ravindranath, Hari Balakrishnan, Samuel Madden, Lewis Girod, “Accurate, Low-Energy Trajectory Mapping for Mobile Devices”, 2011. In Proceedings of the 8th USENIX Symp on Networked Systems Design and Implementation (NSDI), Boston, MA.
- [24] H. Yan, S. Ding, and T. Suel, “Inverted index compression and query processing with optimized document ordering,” in Proceedings of the 18th intl. conference on World Wide Web. ACM, 2009.
- [25] L. L. You, K. T. Pollack, D. D. Long, and K. Gopinath, “Presidio: a framework for efficient archival data storage”, ACM Transactions on Storage (TOS), 2011.
- [26] M. Yuan, K. Deng, J. Zeng, Y. Li, B. Ni, X. He, F. Wang, W. Dai, and Q. Yang, “Oceanst: A distributed analytic system for large-scale spatiotemporal mobile broadband data”, Proc. VLDB Endow., vol. 7, no. 13, Aug. 2014.
- [27] S. Zhang, Y. Yang, W. Fan, L. Lan, and M. Yuan, “Oceanrt: Realtime analytics over large temporal data”, in Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, ser. SIGMOD ’14. New York, NY, USA: ACM, 2014.
- [28] F. Zhu, C. Luo, M. Yuan, Y. Zhu, Z. Zhang, T. Gu, K. Deng, W. Rao, and J. Zeng, “City-scale localization with telco big data”, in Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, ser. CIKM. New York, NY, USA: ACM, 2016.
- [29] https://en.wikipedia.org/wiki/Call_detail_record

- [30] https://en.wikipedia.org/wiki/Radio_Network_Controller
- [31] <https://searchnetworking.techtarget.com/definition/Home-Location-Register>
- [32] <https://www.asirt.org/safe-travel/road-safety-facts/>
- [33] <https://www.cisco.com/c/en/us/products/wireless/sgw-serving-gateway/index.html>
- [34] <http://www.telecomabc.com/b/bsc.html>
- [35] <http://www.telecomabc.com/m/msc.html>
- [36] <http://www.telecomabc.com/s/sgsn.html>
- [37] <https://github.com/graphhopper/map-matching>
- [38] <https://spark.apache.org/docs/latest/cluster-overview.html>
- [39] [https://en.wikipedia.org/wiki/Geographic information system](https://en.wikipedia.org/wiki/Geographic_information_system)
- [40] <https://www.nationalgeographic.org/encyclopedia/geographic-information-system-gis/>

- [41] <https://www.postgresql.org/>
- [42] <https://postgis.net/>
- [43] <https://docplayer.net/57724753-Call-trace-network-parameter-collection-procedures.html>
- [44] <https://opencellid.org/>
- [45] <https://location.services.mozilla.com/>
- [46] <https://wigle.net/>
- [47] <https://www.data.gov.cy/>
- [48] <http://www.emf.mcw.gov.cy/decemfmap/dec-emfmap.html>
- [49] <https://publications.europa.eu/en/publication-detail/-/publication/9509b04f-1df0-4221-bfa2-c7af77975556/language-en>
- [50] [http://www.emf.mcw.gov.cy/pdf/EMF assesment guideline.pdf](http://www.emf.mcw.gov.cy/pdf/EMF_assesment_guideline.pdf)
- [51] <https://www.openstreetmap.org/>

- [52] <https://planet.openstreetmap.org/>
- [53] <https://www.openstreetbrowser.org/#map=10/35.1667/33.3667>
- [54] <https://www.here.com/>
- [55] <https://www.google.com/maps/>
- [56] <https://www.waze.com/>
- [57] [https://en.wikipedia.org/wiki/Induction loop](https://en.wikipedia.org/wiki/Induction_loop)
- [58] <https://www.instructables.com/id/Trigger-GREEN-Traffic-Lights/>
- [59] <https://www.mprnews.org/story/2013/02/20/politics/traffic-camera-bill>
- [60] <http://www.traffic4cyprus.org.cy/trafficapp/?wp=map-gr>
- [61] <https://en.wikipedia.org/wiki/Vehicle-to-everything>

[62] <http://www.chettinadtech.ac.in/storage/11-12-27/11-12-27-09-41-04-1252-swaminathan.pdf>

[63] <https://www.conceptdraw.com/solution-park/computer-networks-telecommunication>

Παράρτημα Α

Στο παράρτημα αυτό, ακολουθεί όλος ο κώδικας που έχει γραφτεί για την υλοποίηση της TrafficWare πλατφόρμας που χρησιμοποιείται για την απεικόνιση γεωγραφικών δεδομένων.

```
<doctype html>
<html lang="en">
<head>
  <title>TrafficWare</title>
  <!-- Required meta tags -->
  <meta charset="utf-8">
  <meta names="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <link rel="shortcut icon" type="image/png" href="favicon.png" />

  <!-- Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwGgFAW/dAiS6JXm" crossorigin="anonymous">
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.7.0/css/font-awesome.min.css">

  <!-- Additional Css -->
  <link rel="stylesheet" href="style.css">
  <link rel="stylesheet" href="prune.css">

  <!-- jQuery first, then Popper.js, then Bootstrap JS -->
  <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-
KJ3o2DKtIkVYIK3UENZmM7KCKr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN" crossorigin="anonymous"></script>
  <script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js" integrity="sha384-
ApNbh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX397fakPskvXusvfa0b4Q" crossorigin="anonymous"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js" integrity="sha384-
JZr6Spej4U02d8J0t6vLEHfe/UQGiRRSQQxSfFWp1MquVdAyiUar5+76PVCmYI" crossorigin="anonymous"></script>

  <!-- Leaflet and marker cluster: -->

  <link rel="stylesheet" href="https://unpkg.com/leaflet@1.3.4/dist/leaflet.css" integrity="sha512-
puBpdR0798QZvTTbP4A8lx/I+A4dHDD0DGqYW6RQ+9jxkRfclaxxQb/SJAWZfWAKuyeQUyO7+7N4QKrDh+dRA=="
crossorigin="" />
  <script src="https://unpkg.com/leaflet@1.3.4/dist/leaflet.js" integrity="sha512-
nMMmRyT0vLqjP9hrbed9S+FzjZHW5gY1TWCHA5ckwXZBadntCNs8kEqAWdrb907rxCaA4IKTIWjDXZxfIOcA=="
crossorigin=""></script>
  <script src="http://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script type="text/javascript" src="http://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.4.1/leaflet.markercluster.js">
</script>
  <link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.4.1/MarkerCluster.css">
  <link rel="stylesheet" href="http://cdnjs.cloudflare.com/ajax/libs/leaflet.markercluster/1.4.1/MarkerCluster.Default.css">

  <!-- Prune cluster (replacing marker cluster): -->
  <script src="pruneCluster.js"></script>

  <!-- Heatmap: -->
  <script src="leaflet-heat.js"></script>

  <!-- for tiles: -->
  <script src="https://unpkg.com/geojson-vt@3.2.0/geojson-vt.js"></script>

</head>

<body>

  <!-- Navigation -->
  <ul class="nav fixed-bottom nav-tabs">
    <li class="nav-item">
      <a class="nav-link" href="#home"><b>Home</b></a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#map"><b>Map</b></a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="#info"><b>Info</b></a>
    </li>
  </ul>

</div>

  <br />
  <div class="text-center">
    <button type="button" class="btn btn-primary" id="clear">
      Clear Data
    </button>
  </div>
  <br />
</div>
</div>
</div>

<div id="mapid" class="main"></div>

<div id="info" class="bg-dark" style="height: 100%;">
  <div class="container padding-top-bottom" style="color: #FFF;">
    <h5><b>For OpenCellId nodes:</b> Grey for GSM, Red for UMTS, Purple for LTE and Blue for CDMA</h5>
    <br />
    <h5><b>For Cell Tower nodes:</b> Grey for 2G, Red for 3G and Purple for LTE/4G</h5>
    <br />
    <h5><b>Tiles:</b> OpenStreetMap, MapBox and ThunderForest,</h5>
    <br />
    <h5><b>Data:</b> OpenCellId, MLS, Wigle, Open Government Data of Cyprus, ADE Measurements</h5>
  </div>
</div>
<script src="map.js"></script>
<script src="nav.js"></script>
</body>

</html>
```

```
<!-- Navigation -->
<div id="home" class="banner">
  <div class="banner-text">
    <h3>- ADE: TrafficWare -</h3>
    <h1>Maps</h1>
  </div>
</div>
<!-- /Banner -->

<div id="map" class="main">
  <div class="pos-f">
    <div class="bg-dark p-1">

      <div class="btn-toolbar" role="toolbar" aria-label="Toolbar with button groups">

        <span class="navButton" onclick="openNav()">&#9776;</span>
        <div class="btn-group mr-2" role="group" aria-label="First group">
          <button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-
haspopup="true" aria-expanded="false">
            Tile Layer
          </button>
          <div class="dropdown-menu">
            <a class="dropdown-item" id="default">Default</a>
            <a class="dropdown-item" id="classic_tile">Classic</a>
            <a class="dropdown-item" id="satellite">Satellite</a>
            <a class="dropdown-item" id="dark">Dark</a>
          </div>
        </div>

        <button type="button" class="btn btn-primary" onclick="window.open('http://10.16.20.19:8989/app/')">
          Map-Match Server
        </button>

      </div>
    </div>
  </div>
</div>
<div id="mySidenav" class="sidenav">
  <a href="javascript:void(0)" class="closebtn" onclick="closeNav()">&times;</a>

  <div class="container bg-light-grey">
    <div class="bg-light-grey">
      <br />
      <h2>Point's Function</h2>
    </div>
    <div style="padding-right: 10px; padding-left: 10px;">
      <h5><label><input type="radio" name="funPoint" value="defaults" id="defaults"> Default</label></h5>
      <h5><label><input type="radio" name="funPoint" value="prune" id="prune"> Prune</label></h5>
      <h5><label><input type="radio" name="funPoint" value="cluster" id="cluster"> Cluster</label></h5>
      <h5><label><input type="radio" name="funPoint" value="heatmap" id="heatmap"> Heatmap</label></h5>
      <h5><label><input type="radio" name="funPoint" value="classic" id="classic"> Classic</label></h5>
      <h5><label id="clustering" style="visibility: hidden;"><input type="radio" name="funPoint" value="cluster_id"
id="cluster_id"> Clustering By Ids</label></h5>
    </div>
    <br />
    <br />
    <h2>Data</h2>
    <div style="padding-right: 10px; padding-left: 10px;">
      <h5><label><input type="checkbox" id="opencellid"> OpenCellId Towers</label></h5>
      <h5><label><input type="checkbox" id="measurements"> My Measurements</label></h5>
      <h5><label><input type="checkbox" id="mls"> MLS Towers</label></h5>
      <h5><label><input type="checkbox" id="cyprus_cell"> Cyprus' Cell Towers</label></h5>
      <h5><label><input type="checkbox" id="wigle"> Wigle Towers</label></h5>
    </div>
  </div>
</div>
```

```
</div>

  <br />
  <div class="text-center">
    <button type="button" class="btn btn-primary" id="clear">
      Clear Data
    </button>
  </div>
  <br />
</div>
</div>
</div>

<div id="mapid" class="main"></div>

<div id="info" class="bg-dark" style="height: 100%;">
  <div class="container padding-top-bottom" style="color: #FFF;">
    <h5><b>For OpenCellId nodes:</b> Grey for GSM, Red for UMTS, Purple for LTE and Blue for CDMA</h5>
    <br />
    <h5><b>For Cell Tower nodes:</b> Grey for 2G, Red for 3G and Purple for LTE/4G</h5>
    <br />
    <h5><b>Tiles:</b> OpenStreetMap, MapBox and ThunderForest,</h5>
    <br />
    <h5><b>Data:</b> OpenCellId, MLS, Wigle, Open Government Data of Cyprus, ADE Measurements</h5>
  </div>
</div>
<script src="map.js"></script>
<script src="nav.js"></script>
</body>

</html>
```

```
// data structures
var file = { phpfile: "tilegisdata.php", typeOfFile: "php" };

// variables
var zoom = 12;
var lat = 35.1522437;
var lon = 33.3751678;
var mymap = new L.map('mapid', { zoomControl: false });
var minlat;
var minlon;
var maxlat;
var maxlon;
var cached_points = [];
var defaults_flag = 1;
var option = 1;
var data_array = [];
var points = [];
var last_lat = 0, last_lon = 0, cell_count = 0, last_cellid = 0, polygon, circle, spiderfy = false;

// markers
var markers = new L.FeatureGroup();
var pruneCluster = new PruneClusterForLeaflet();
var clustering_points = [];
var cluster = L.markerClusterGroup();

var serverIP = "10.16.30.133";
var chooseFun = 'cluster';

var popup = L.popup();

var blueIcon = L.icon({
  iconUrl: 'https://cdn.rawgit.com/pointhi/leaflet-color-markers/master/img/marker-icon-blue.png',
  shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
  iconSize: [25, 41],
  iconAnchor: [12, 41],
  popupAnchor: [1, -34],
  shadowSize: [41, 41]

});

var redIcon = L.icon({
  iconUrl: 'https://cdn.rawgit.com/pointhi/leaflet-color-markers/master/img/marker-icon-red.png',
  shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
  iconSize: [25, 41],
  iconAnchor: [12, 41],
  popupAnchor: [1, -34],
  shadowSize: [41, 41]

});

var purpleIcon = L.icon({
  iconUrl: 'https://cdn.rawgit.com/pointhi/leaflet-color-markers/master/img/marker-icon-violet.png',
  shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
  iconSize: [25, 41],
  iconAnchor: [12, 41],
  popupAnchor: [1, -34],
  shadowSize: [41, 41]

});

var greyIcon = L.icon({
```

```

iconUrl: 'https://cdn.rawgit.com/pointhi/leaflet-color-markers/master/img/marker-icon-grey.png',
shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
iconSize: [25, 41],
iconAnchor: [12, 41],
popupAnchor: [1, -34],
shadowSize: [41, 41]
});

var greenIcon = L.icon({
  iconUrl: 'https://cdn.rawgit.com/pointhi/leaflet-color-markers/master/img/marker-icon-green.png',
  shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
  iconSize: [25, 41],
  iconAnchor: [12, 41],
  popupAnchor: [1, -34],
  shadowSize: [41, 41]
});

var yellowIcon = L.icon({
  iconUrl: 'https://cdn.rawgit.com/pointhi/leaflet-color-markers/master/img/marker-icon-yellow.png',
  shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
  iconSize: [25, 41],
  iconAnchor: [12, 41],
  popupAnchor: [1, -34],
  shadowSize: [41, 41]
});

var orangeIcon = L.icon({
  iconUrl: 'https://cdn.rawgit.com/pointhi/leaflet-color-markers/master/img/marker-icon-orange.png',
  shadowUrl: 'https://cdnjs.cloudflare.com/ajax/libs/leaflet/0.7.7/images/marker-shadow.png',
  iconSize: [25, 41],
  iconAnchor: [12, 41],
  popupAnchor: [1, -34],
  shadowSize: [41, 41]
});

var wigle_color = "default";

var layer = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  maxZoom: 18,
  minZoom: 2,
  maxBounds: L.latLngBounds(L.latLng(-90, -180), L.latLng(90, 180)),
  attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
}).addTo(mymap);

new L.Control.Zoom({ position: 'topright' }).addTo(mymap);
mymap.setView([lat, lon], zoom);

function defaultLayer() {
  mymap.removeLayer(layer);

  layer = L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
    maxZoom: 18,
    minZoom: 2,
    maxBounds: L.latLngBounds(L.latLng(-90, -180), L.latLng(90, 180)),
    attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
  }).addTo(mymap);
}

```

```

cluster.clearLayers();
pruneCluster.RemoveMarkers();
pruneCluster.ProcessView();
// remove previous polylines and circles
if (polygon != undefined)
  mymap.removeLayer(polygon);
if (circle != undefined)
  mymap.removeLayer(circle);

if (!(defaults_flag && (mymap.getZoom() == 14))) cached_points = [];

}

/** Calls the function to make an ajax call and fetch
 * tiled data from the database.
 */
function fetchPointFromDatabase(data_type) {

  file.typeOfFile = "php";
  if (data_type == "opencellid") file.phpfile = "tilegisdata.php";
  else if (data_type == "wigle") file.phpfile = "wigle.php";
  else if (data_type == "cyprus_cell") file.phpfile = "cyprusCells.php";
  else if (data_type == "measurements") file.phpfile = "metrics.php";
  else if (data_type == "misl") file.phpfile = "misl.php";
  else {

    data_array.forEach(function (element) {
      if (element == "opencellid") {
        file.phpfile = "tilegisdata.php";
        option = 1;
      }
      else if (element == "wigle") {
        file.phpfile = "wigle.php";
        option = 3;
      }
      else if (element == "cyprus_cell") {
        file.phpfile = "cyprusCells.php";
        option = 5;
      }
      else if (element == "measurements") {
        file.phpfile = "metrics.php";
        option = 4;
      }
      else if (element == "misl") {
        file.phpfile = "misl.php";
        option = 2;
      }
    });

    ajaxCall(executeQuery, option);
  }

  if (data_type != "unchecked")
    ajaxCall(executeQuery, option);
}

function changeTileData() {

  var north = mymap.getBounds().getNorth().toString();
  var south = mymap.getBounds().getSouth().toString();
  var east = mymap.getBounds().getEast().toString();
  var west = mymap.getBounds().getWest().toString();
}

```

```

function classicLayer() {
  mymap.removeLayer(layer);

  layer = L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access_token={accessToken}', {
    maxZoom: 18,
    minZoom: 2,
    maxBounds: L.latLngBounds(L.latLng(-90, -180), L.latLng(90, 180)),
    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, ' +
      '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
      'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
    id: 'mapbox.streets',
    accessToken:
      'pk.eyJ1Ijoiamdlb3JnMDIiLCJhIjoY2ptM292NjBnMW1zazNrbzV4dGhqNXVqYjU9.kkorbmktIF9J2JRPDOu3w'
  }).addTo(mymap);
}

function satelliteLayer() {
  mymap.removeLayer(layer);

  layer = L.tileLayer('https://api.tiles.mapbox.com/v4/{id}/{z}/{x}/{y}.png?access_token={accessToken}', {
    maxZoom: 18,
    minZoom: 2,
    maxBounds: L.latLngBounds(L.latLng(-90, -180), L.latLng(90, 180)),
    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, ' +
      '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
      'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
    id: 'mapbox.satellite',
    accessToken:
      'pk.eyJ1Ijoiamdlb3JnMDIiLCJhIjoY2ptM292NjBnMW1zazNrbzV4dGhqNXVqYjU9.kkorbmktIF9J2JRPDOu3w'
  }).addTo(mymap);
}

function darkLayer() {
  mymap.removeLayer(layer);

  layer = L.tileLayer('https://api.tiles.mapbox.com/v4/mapbox.dark/{z}/{x}/{y}.png?access_token={accessToken}', {
    maxZoom: 18,
    minZoom: 2,
    maxBounds: L.latLngBounds(L.latLng(-90, -180), L.latLng(90, 180)),
    attribution: 'Map data &copy; <a href="https://www.openstreetmap.org/">OpenStreetMap</a> contributors, ' +
      '<a href="https://creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, ' +
      'Imagery © <a href="https://www.mapbox.com/">Mapbox</a>',
    id: 'mapbox.streets',
    accessToken:
      'pk.eyJ1Ijoiamdlb3JnMDIiLCJhIjoY2ptM292NjBnMW1zazNrbzV4dGhqNXVqYjU9.kkorbmktIF9J2JRPDOu3w'
  }).addTo(mymap);
}

function onMapClick(e) {
  var latlon = e.latlng;
  popup
    .setLatLng(latlon)
    .setContent("You clicked the map at " + latlon.toString())
    .openOn(mymap);
}

function clearPoints(fun) {
  markers.clearLayers();

  if (maxlat != north || minlat != south || maxlon != east || minlon != west) {

    maxlat = north;
    minlat = south;
    maxlon = east;
    minlon = west;

    if (defaults_flag && (mymap.getZoom() == 15)) clearPoints();

    if (data_array.length > 1)
      points = [];
    else if (data_array.length == 1)
      if (data_array[1] != "measurements")
        points = [];

    fetchPointFromDatabase("unchecked");
  }
}

function executeQuery(data, option) {
  var fields = ["radio", "mcc", "net", "area", "cell", "unit", "lon", "lat", "range", "samples", "changeable", "created", "updated", "averageSignal"];
  var dataArray;
  if (option == 5) dataArray = data.split("", ",");
  else dataArray = data.split(", ,");
  dataArray.pop();

  wigle_color = 'default';
  var cellid = 0;
  var counter = 0;
  var cell_flag = false;
  dataArray.forEach(function (dataElement) {

    cell_flag = false;

    if (option == 5) dataElement = dataElement.split("", ",");
    else dataElement = dataElement.split(", ,");
    var popup = "";
    var type = dataElement[0];
    lat = dataElement[6];
    lon = dataElement[7];

    // for opencellid data
    if (option == 1) {

      lat = dataElement[7];
      lon = dataElement[6];

      popup = "<h1>OpenCellid Tower</h1><b>" + fields[0] + "</b>" + dataElement[0] + "<br /><b>" + fields[1] + "</b>" +
        dataElement[1] + "<br /><b>" +
        fields[2] + "</b>" + dataElement[2] + "<br /><b>" + fields[3] + "</b>" + dataElement[3] + "<br /><b>" +
        fields[4] + "</b>" + dataElement[4] + "<br /><b>" + fields[5] + "</b>" + dataElement[5] + "<br /><b>" +
        fields[6] + "</b>" + dataElement[6] + "<br /><b>" + fields[7] + "</b>" + dataElement[7] + "<br /><b>" +
        fields[8] + "</b>" + dataElement[8] + "<br /><b>" + fields[9] + "</b>" + dataElement[9] + "<br /><b>" +
        fields[13] + "</b>" + dataElement[13] + "<br />";

      // for wiggle data
    } else if (option == 3) {

      type = dataElement[15];
      lat = dataElement[0];
      lon = dataElement[1];

      cell_flag = false;

      if (cellid != dataElement[13]) {

```

```
switch (counter) {
    case 0:
        wigle_color = bluelcon;
        break;
    case 1:
        wigle_color = redlcon;
        break;
    case 2:
        wigle_color = purplelcon;
        break;
    case 3:
        wigle_color = greylcon;
        break;
    case 4:
        wigle_color = greenlcon;
        break;
    case 5:
        wigle_color = yellowlcon;
        break;
    case 6:
        wigle_color = orangelcon;
        break;
}
counter++;
cellid = dataElement[13];
cell_flag = true;
}

if (counter == 7) counter = 0;

popup = "<h1>Wigle Tower</h1><b>Lat: </b>" + dataElement[0] + ",<br /><b>Lon: </b>" + dataElement[1] + ",<br />
<b>SSID: </b>" + dataElement[2]
    + ",<br /><b>QOS: </b>" + dataElement[3] + ",<br /><br /><b>Transit: </b>" + dataElement[4]
    + ",<br /><b>First Time: </b>" + dataElement[5] + ",<br /><b>Last Time: </b>" + dataElement[6]
    + ",<br /><b>Last Updated: </b>" + dataElement[7] + ",<br /><b>House Number: </b>" + dataElement[8]
    + ",<br /><b>Road: </b>" + dataElement[9] + ",<br /><b>City: </b>" + dataElement[10]
    + ",<br /><b>Region: </b>" + dataElement[11] + ",<br /><b>Country: </b>" + dataElement[12]
    + ",<br /><b>ID: </b>" + dataElement[13] + ",<br /><b>Attributes: </b>" + dataElement[14]
    + ",<br /><b>Gentype: </b>" + dataElement[15];

type = dataElement[15];

// for cyprus' cell data
} else if (option == 5) {

    lat = dataElement[3];
    lon = dataElement[4];
    popup = "<h1>Cyprus' Cell Tower</h1><b>Code: </b>" + dataElement[0] + ",<br /><b>Name: </b>" + dataElement[1] +
",<br /><b>Address: </b>" + dataElement[2] +
    ",<br /><b>GSM: </b>" + dataElement[5] + ",<br /><b>DCS: </b>" + dataElement[6] + ",<br /><b>Antenna Height:
</b>" + dataElement[7] +
    ",<br /><b>Services: </b>" + dataElement[8] + ",<br /><b>Site Installation: </b>" + dataElement[9] + ",<br /><b>Area:
</b>" + dataElement[10] +
    ",<br /><b>Colocation: </b>" + dataElement[11] + ",<br /><b>Latitude: </b>" + dataElement[3] + ",<br />
<b>Longitude: </b>" + dataElement[4];

// for mls data
} else if (option == 2) {

    lat = dataElement[7];
    lon = dataElement[6];
    popup = "<h1>MLS Tower</h1><b>Radio: </b>" + dataElement[0] + ",<br /><b>MCC: </b>" + dataElement[1] + ",<br />
<b>Net: </b>" + dataElement[2] +
    ",<br /><b>Area: </b>" + dataElement[3] + ",<br /><b>Cell: </b>" + dataElement[4] + ",<br /><b>Unit: </b>" +
dataElement[5] +

    chooseFun = 'cluster_id';
}
}
} else {
    switch (chooseFun) {
        case "classic":
            clustering_points = [];
            points = [];
            classicPoints(lon, lat, type, popup);
            break;
        case "heatmap":
            clustering_points = [];
            points.push({ lat, lon });
            break;
        case "cluster":
            clustering_points = [];
            points = [];
            clusterPoints(lon, lat, type, popup);
            break;
        case "prune":
            clustering_points = [];
            points = [];
            prunePoints(lon, lat, type, popup);
            break;
        case "cluster_id":
            points = [];
            clusterIdPoints(lon, lat, cell_flag, popup, type, cellid);
            break;
    }
}
}
);

if (chooseFun == 'prune') pruneCluster.ProcessView();
else if (chooseFun == 'heatmap') heatmap(points);
}

function clusterIdPoints(lon, lat, cellid_flag, popup, cell, cellid) {

    var marker;

    if ((cellid_flag) && (last_lat != 0 && last_lon != 0)) {

        marker = L.marker([last_lat, last_lon], { icon: wigle_color });
        marker.bindPopup("<b>Cellid: </b>" + last_cellid);

        clustering_points.push({ cellid: last_cellid, map_point: marker });

        marker.on('click', function (e) {

            var points = [];
            var cell_id;
            var latlon;
            var i;

            for (i = 0; i < clustering_points.length; i++) {
                if (e.latlng == clustering_points[i].map_point.getLatLng()) {
                    cell_id = clustering_points[i].cellid;
                    break;
                }
            }
        })
    }
}
```

```
"<br /><b>Range: </b>" + dataElement[8] + ",<br /><b>Samples: </b>" + dataElement[9] + ",<br /><b>Changeable:
</b>" + dataElement[10] +
",<br /><b>Created: </b>" + dataElement[11] + ",<br /><b>Updated: </b>" + dataElement[12] + ",<br /><b>Latitude:
</b>" + dataElement[7] +
    ",<br /><b>Longitude: </b>" + dataElement[6];

// for my measurements
} else {

    lat = dataElement[1];
    lon = dataElement[2];

    popup = "<h1>My Measurements</h1>" +
        "<b>Timestamp: </b>" + dataElement[0] + ",<br /><b>Latitude: </b>" + dataElement[1] + ",<br /><b>Longitude:
</b>" + dataElement[2] + ",<br /><b>Type: </b>" + dataElement[3] +
        ",<br /><b>Signal: </b>" + dataElement[4] + ",<br /><b>Cellid: </b>" + dataElement[5] + ",<br /><b>Mnc: </b>" +
dataElement[6] + ",<br /><b>Lac: </b>" + dataElement[7];

    if (cellid != dataElement[5]) {
        switch (counter) {
            case 0:
                wigle_color = bluelcon;
                break;
            case 1:
                wigle_color = redlcon;
                break;
            case 2:
                wigle_color = purplelcon;
                break;
            case 3:
                wigle_color = greylcon;
                break;
            case 4:
                wigle_color = greenlcon;
                break;
            case 5:
                wigle_color = yellowlcon;
                break;
            case 6:
                wigle_color = orangelcon;
                break;
        }
        counter++;
        cellid = dataElement[5];
        cell_flag = true;
    }

    if (counter == 7) counter = 0;
}

if (defaults_flag) {
    if (mymap.getZoom() <= 14 && defaults_flag != 2) {
        points.push({ lat, lon });
        chooseFun = 'heatmap';
    }
} else {
    points = [];
    if (option != 4) {
        clusterPoints(lon, lat, type, popup);
        chooseFun = 'cluster';
    }
} else {
    clusterIdPoints(lon, lat, cell_flag, popup, type, cellid);

    clustering_points.forEach(function (point) {
        if (cell_id == point.cellid) {
            latlon = point.map_point.getLatLng();
            points.push(latlon);
        }
    });

    // find outer-points (gift-wrapping algorithm)
    var lefterPoint;
    i = 0;
    points.forEach(function (point) {
        if (i == 0) lefterPoint = point;
        else if (lefterPoint > point) lefterPoint = point;
        i++;
    });

    var pointOnHull = lefterPoint;
    var endpoint;
    var wrappedPoints = [];
    i = 0;
    do {
        wrappedPoints[i] = pointOnHull;
        endpoint = points[0];
        for (var j = 1; j < points.length; j++)
            if (endpoint == pointOnHull || points[j] < wrappedPoints[i])
                endpoint = points[j];
        i++;
        pointOnHull = endpoint;
    } while (endpoint != wrappedPoints[0]);

    // remove previous polylines and circles
    if (polygon != undefined)
        mymap.removeLayer(polygon);
    if (circle != undefined)
        mymap.removeLayer(circle);

    // add the new ones
    polygon = L.polygon(wrappedPoints, { color: 'red' }).addTo(mymap);
    mymap.fitBounds(polygon.getBounds());

    // find distance of polygon
    var totalDistance = polygon.getBounds()._northEast.distanceTo(polygon.getBounds()._southWest);

    // radius = totalDistance / 2
    circle = L.circle(polygon.getBounds().getCenter(), totalDistance / 2).addTo(mymap);

});

last_lat = 0;
last_lon = 0;
last_cellid = cellid;
markers.addLayer(marker);
mymap.addLayer(markers);

}

if (!checkMarker(lat, lon)) {

    if (cell != null) {

        if (wigle_color == 'default') {

            if ((cell == "GSM") || (cell == "2G"))
                marker = L.marker([lat, lon], { icon: greylcon });
        }
    }
}
```

```

    else if ((cell == "UMTS") || (cell == '3G'))
        marker = L.marker([lat, lon], { icon: redIcon });

    else if ((cell == "LTE") || (cell == 'LTE/4G'))
        marker = L.marker([lat, lon], { icon: purpleIcon });

    else marker = L.marker([lat, lon]);
} else marker = L.marker([lat, lon], { icon: wgle_color });

} else marker = L.marker([lat, lon]);

if (popup != null)
    marker.bindPopup(popup);

clustering_points.push({ cellid: cellid, map_point: marker });
cached_points.push({ on_map: 1, map_point: marker });
if (last_lat == 0 && last_lon == 0) {
    last_lat = lat;
    last_lon = lon;
}
}
}
}


```

```

/** This function takes as paramters the function that will call the ajax cal
 * and the type, if it is an ajax call for a php file or for a regular file.
 */

```

```
function ajaxCall(fun, option) {
```

```

    var url;
    var data;
    var datatype;
```

```

    url = file.phpfile;
```

```

    data = {
        minlon: minlon,
        minlat: minlat,
        maxlon: maxlon,
        maxlat: maxlat
    };
    datatype = "text";

```

```

$(document).ready(function () {
    $.ajax({
        type: "GET",
        url: url,
        data: data,
        dataType: datatype,
        success: function (response) {
            fun(response, option);
        }
    });
});

```

```
function checkMarker(lat, lon) {
    var exists = false;

```

```

    mymap.addLayer(markers);
}

```

```
function heatmap(points) {
```

```

    clearPoints("heatmap");

```

```

    points = points.map(function (p) {
        return [p.lat, p.lon, 0.5];
    });

```

```

    markers.addLayer(L.heatLayer(points, { radius: 25 }));
    mymap.addLayer(markers);

```

```

}

```

```
function clusterPoints(lon, lat, cell, popup) {
```

```

    if (!checkMarker(lat, lon)) {
        var cluster_point;
```

```

        if (cell != null) {
```

```

            if (wgle_color == 'default') {
```

```

                if ((cell == "GSM") || (cell == '2G'))
                    cluster_point = L.marker([lat, lon], { icon: greyIcon });

```

```

                else if ((cell == "UMTS") || (cell == '3G'))
                    cluster_point = L.marker([lat, lon], { icon: redIcon });

```

```

                else if ((cell == "LTE") || (cell == 'LTE/4G'))
                    cluster_point = L.marker([lat, lon], { icon: purpleIcon });

```

```

                else cluster_point = L.marker([lat, lon]);

```

```

            } else cluster_point = L.marker([lat, lon], { icon: wgle_color });

```

```

        } else cluster_point = L.marker([lat, lon]);

```

```

        if (popup != null)
            cluster_point.bindPopup(popup);

```

```

        cached_points.push({ lat, lon });
        cluster.addLayer(cluster_point);
        mymap.addLayer(cluster);
    }
}

```

```
function prunePoints(lon, lat, cell, popup) {
```

```

    if (!checkMarker(lat, lon)) {
        var prune_marker = new PruneCluster.Marker(lat, lon);

```

```

        if (cell != null) {
```

```

            if ((cell == "GSM") || (cell == '2G')) prune_marker.data.icon = greyIcon;
```

```

            else if ((cell == "UMTS") || (cell == '3G')) prune_marker.data.icon = redIcon;
```

```

            else if ((cell == "LTE") || (cell == 'LTE/4G')) prune_marker.data.icon = purpleIcon;
```

```

        }
    }
}

```

```

cached_points.forEach(function (point) {
    switch (chooseFun) {
        case "classic":
            if ((point.map_point.getLatLng().lat == lat) && (point.map_point.getLatLng().lng == lon)) {
                exists = true;
                return;
            }

            break;
        case "cluster":
            if ((point.lat == lat) && (point.lon == lon)) {
                exists = true;
                return;
            }

            break;
        case "prune":
            if ((point.position.lat == lat) && (point.position.lng == lon)) {
                exists = true;
                return;
            }
            break;
        case "cluster_id":
            if ((point.map_point.getLatLng().lat == lat) && (point.map_point.getLatLng().lng == lon)) {
                exists = true;
                return;
            }
            break;
    }
});
return exists;
}

```

```
function classicPoints(lon, lat, cell, popup) {
```

```

    if (!checkMarker(lat, lon)) {
        var marker;
```

```

        if (cell != null) {
```

```

            if (wgle_color == 'default') {
```

```

                if ((cell == "GSM") || (cell == '2G'))
                    marker = L.marker([lat, lon], { icon: greyIcon });

```

```

                else if ((cell == "UMTS") || (cell == '3G'))
                    marker = L.marker([lat, lon], { icon: redIcon });

```

```

                else if ((cell == "LTE") || (cell == 'LTE/4G'))
                    marker = L.marker([lat, lon], { icon: purpleIcon });

```

```

                else marker = L.marker([lat, lon]);

```

```

            } else marker = L.marker([lat, lon], { icon: wgle_color });

```

```

        } else marker = L.marker([lat, lon]);

```

```

        if (popup != null)
            marker.bindPopup(popup);

```

```

        cached_points.push({ on_map: 1, map_point: marker });
        markers.addLayer(marker);

```

```

        if (popup != null) prune_marker.data.popup = popup;
        cached_points.push(prune_marker);
        pruneCluster.RegisterMarker(prune_marker);
        mymap.addLayer(pruneCluster);
    }
}

```

```
function checkClusterIdVisible() {
```

```

    let clustering = document.getElementById("clustering");
    let wgle = document.getElementById("wgle");
    let mls = document.getElementById("mls");
    let opencellid = document.getElementById("opencellid");
    let cyprus_cell = document.getElementById("cyprus_cell");
    let measurements = document.getElementById("measurements");

```

```

    if (measurements.checked && !wgle.checked && !mls.checked && !opencellid.checked && !cyprus_cell.checked)
        clustering.style.visibility = "visible";
    else clustering.style.visibility = "hidden";

```

```

}

```

```

window.onload = pageLoad;
```

```
function pageLoad() {
```

```

    // assign onclick for the layer buttons
    document.getElementById("default").onclick = defaultLayer;
    document.getElementById("classic_tile").onclick = classicLayer;
    document.getElementById("satellite").onclick = satelliteLayer;
    document.getElementById("dark").onclick = darkLayer;

```

```

    // assign onclick for the type of points on radio buttons
    document.getElementById("defaults").onclick = function () {
        clearPoints();
        defaults_flag = 1;
        fetchPointFromDatabase("unchecked");
    };

```

```

    document.getElementById("prune").onclick = function () {
        chooseFun = 'prune';
        clearPoints();
        defaults_flag = 0;
        fetchPointFromDatabase("unchecked");
    };
    document.getElementById("cluster").onclick = function () {
        chooseFun = 'cluster';
        clearPoints();
        defaults_flag = 0;
        fetchPointFromDatabase("unchecked");
    };
    document.getElementById("heatmap").onclick = function () {
        chooseFun = 'heatmap';
        clearPoints();
        defaults_flag = 0;
        fetchPointFromDatabase("unchecked");
    };

```

```

    document.getElementById("classic").onclick = function () {
        chooseFun = 'classic';
        clearPoints();
        defaults_flag = 0;
        fetchPointFromDatabase("unchecked");
    };
}

```

```

document.getElementById("cluster_id").onclick = function () {
  chooseFun = "cluster_id";
  clearPoints();
  defaults_flag = 2;
  fetchPointFromDatabase("unchecked");
};

// assign onclick for the type of data on checkboxes
document.getElementById("opencellid").onclick = function () {

  checkClusterIdVisible();

  if (this.checked) {

    option = 1;
    data_array.push("opencellid");
    fetchPointFromDatabase("opencellid");
  }
  else {
    clearPoints();
    cached_points = [];
    points = [];
    data_array.splice(data_array.indexOf("opencellid"), 1);
    fetchPointFromDatabase("unchecked");
  }
};

document.getElementById("mls").onclick = function () {

  checkClusterIdVisible();

  if (this.checked) {
    option = 2;
    data_array.push("mls");
    fetchPointFromDatabase("mls");
  }
  else {
    clearPoints();
    cached_points = [];
    points = [];
    data_array.splice(data_array.indexOf("mls"), 1);
    fetchPointFromDatabase("unchecked");
  }
};

document.getElementById("wigle").onclick = function () {

  checkClusterIdVisible();

  if (this.checked) {
    option = 3;
    data_array.push("wigle");
    fetchPointFromDatabase("wigle");
  }
  else {
    clearPoints();
    cached_points = [];
    points = [];
    data_array.splice(data_array.indexOf("wigle"), 1);
    fetchPointFromDatabase("unchecked");
  }
};

document.getElementById("measurements").onclick = function () {

```

```

maxlat = mymap.getBounds().getNorth().toString();
minlat = mymap.getBounds().getSouth().toString();
maxlon = mymap.getBounds().getEast().toString();
minlon = mymap.getBounds().getWest().toString();

// load some data for the very first load
document.getElementById("opencellid").checked = true;
document.getElementById("defaults").checked = true;
option = 1;
data_array.push("opencellid");
fetchPointFromDatabase("opencellid");
}

```

```

checkClusterIdVisible();

if (this.checked) {

  option = 4;
  data_array.push("measurements");
  fetchPointFromDatabase("measurements");
}
else {
  clearPoints();
  cached_points = [];
  points = [];
  data_array.splice(data_array.indexOf("measurements"), 1);
  fetchPointFromDatabase("unchecked");
  clustering.style.visibility = "hidden";
}
};

document.getElementById("cyprus_cell").onclick = function () {

  checkClusterIdVisible();

  if (this.checked) {
    option = 5;
    data_array.push("cyprus_cell");
    fetchPointFromDatabase("cyprus_cell");
  }
  else {
    clearPoints();
    cached_points = [];
    points = [];
    data_array.splice(data_array.indexOf("cyprus_cell"), 1);
    fetchPointFromDatabase("unchecked");
  }
};

document.getElementById("clear").onclick = function () {

  $("#opencellid").prop("checked", false);
  $("#measurements").prop("checked", false);
  $("#mls").prop("checked", false);
  $("#cyprus_cell").prop("checked", false);
  $("#wigle").prop("checked", false);

  option = 0;
  clearPoints();
  cached_points = [];
  points = [];
  data_array = [];

  let clustering = document.getElementById("clustering");
  clustering.style.visibility = "hidden";

};

mymap.on('click', onMapClick);

mymap.on('moveend', changeTileData);

mymap.on('zoomend', changeTileData);

// take bounds

```

```

/* Set the width of the side navigation to 250px */
function openNav() {
  document.getElementById("mySidenav").style.width = "250px";
}

/* Set the width of the side navigation to 0 */
function closeNav() {
  document.getElementById("mySidenav").style.width = "0";
}

```

```
<?php
ini_set('display_errors', 1);

$host = '127.0.0.1';
$port = "";
$dbname = "";
$user = "";
$password = "";

if ($_GET) {
    $minlon = $_GET['minlon'];
    $minlat = $_GET['minlat'];
    $maxlon = $_GET['maxlon'];
    $maxlat = $_GET['maxlat'];
} else {
    $minlon = $argv[1];
    $minlat = $argv[2];
    $maxlon = $argv[3];
    $maxlat = $argv[4];
}

$conn = pg_connect("host=$host port=$port dbname=$dbname user=$user password=$password");
if (!$conn) {
    echo "Connection Error:" . pg_error();
    exit;
}

$sql = 'SELECT * FROM cellinfo WHERE "lon" <= ' . $maxlon . ' AND "lon" >= ' . $minlon . ' AND "lat" <= ' . $maxlat . ' AND "lat" >= ' . $minlat . ' ORDER BY cellid DESC;';

// sending the query
if (!$response = pg_query($conn, $sql)) {
    echo "A query error occurred.\n";
    exit;
}

// echo the data back
while ($row = pg_fetch_row($response)) {

    foreach($row as $i => $attr)
        echo $attr." ";
    echo "\n";
}

?>
```

```
<?php
ini_set('display_errors', 1);

$host = '127.0.0.1';
$port = "";
$dbname = "";
$user = "";
$password = "";

if ($_GET) {
    $minlon = $_GET['minlon'];
    $minlat = $_GET['minlat'];
    $maxlon = $_GET['maxlon'];
    $maxlat = $_GET['maxlat'];
} else {
    $minlon = $argv[1];
    $minlat = $argv[2];
    $maxlon = $argv[3];
    $maxlat = $argv[4];
}

$conn = pg_connect("host=$host port=$port dbname=$dbname user=$user password=$password");
if (!$conn) {
    echo "Connection Error:" . pg_error();
    exit;
}

$sql = 'SELECT * FROM public.mls_cell_towers WHERE "lon" <= ' . $maxlon . ' AND "lon" >= ' . $minlon . ' AND "lat" <= ' . $maxlat . ' AND "lat" >= ' . $minlat . " ";';

// sending the query
if (!$response = pg_query($conn, $sql)) {
    echo "A query error occurred.\n";
    exit;
}

// echo the data back
while ($row = pg_fetch_row($response)) {

    foreach($row as $i => $attr)
        echo $attr." ";
    echo "\n";
}

?>
```

```
<?php
ini_set('display_errors', 1);

$host = '127.0.0.1';
$port = "";
$dbname = "";
$user = "";
$password = "";

if ($_GET) {
    $minlon = $_GET['minlon'];
    $minlat = $_GET['minlat'];
    $maxlon = $_GET['maxlon'];
    $maxlat = $_GET['maxlat'];
} else {
    $minlon = $argv[1];
    $minlat = $argv[2];
    $maxlon = $argv[3];
    $maxlat = $argv[4];
}

$conn = pg_connect("host=$host port=$port dbname=$dbname user=$user password=$password");
if (!$conn) {
    echo "Connection Error:" . pg_error();
    exit;
}

$sql = 'SELECT * FROM public."cyprusCells" WHERE "longitude" <= ' . $maxlon . ' AND "longitude" >= ' . $minlon . ' AND "latitude" <= ' . $maxlat . ' AND "latitude" >= ' . $minlat . " ";';

// sending the query
if (!$response = pg_query($conn, $sql)) {
    echo "A query error occurred.\n";
    exit;
}

// echo the data back
while ($row = pg_fetch_row($response)) {

    foreach($row as $i => $attr)
        echo $attr."";
    echo "\n";
}

?>
```

```
<?php
ini_set('display_errors', 1);

$host = '127.0.0.1';
$port = "";
$dbname = "";
$user = "";
$password = "";

if ($_GET) {
    $minlon = $_GET['minlon'];
    $minlat = $_GET['minlat'];
    $maxlon = $_GET['maxlon'];
    $maxlat = $_GET['maxlat'];
} else {
    $minlon = $argv[1];
    $minlat = $argv[2];
    $maxlon = $argv[3];
    $maxlat = $argv[4];
}

$conn = pg_connect("host=$host port=$port dbname=$dbname user=$user password=$password");
if (!$conn) {
    echo "Connection Error:" . pg_error();
    exit;
}

$sql = "SELECT * FROM opencellid WHERE lon <= " . $maxlon . " AND lon >= " . $minlon . " AND lat <= " . $maxlat . " AND lat >= " . $minlat . " ";';
// for limiting data add to the query ex. limit 50

// sending the query
if (!$response = pg_query($conn, $sql)) {
    echo "A query error occurred.\n";
    exit;
}

// echo the data back
while ($row = pg_fetch_row($response)) {

    foreach($row as $i => $attr)
        echo $attr." ";
    echo "\n";
}

?>
```



```
<?php
ini_set('display_errors', 1);

$host = '127.0.0.1';
$port = "";
$dbname = "";
$user = "";
$password = "";

if ($_GET) {
    $minlon = $_GET['minlon'];
    $minlat = $_GET['minlat'];
    $maxlon = $_GET['maxlon'];
    $maxlat = $_GET['maxlat'];
} else {
    $minlon = $_argv[1];
    $minlat = $_argv[2];
    $maxlon = $_argv[3];
    $maxlat = $_argv[4];
}

$conn = pg_connect("host=$host port=$port dbname=$dbname user=$user password=$password");
if (!$conn) {
    echo "Connection Error: " . pg_error();
    exit;
}

$sql = "SELECT * FROM wgle WHERE 'lon' <= ' . $maxlon . ' AND 'lon' >= ' . $minlon . ' AND 'lat' <= ' . $maxlat . ' AND 'lat' >= ' . $minlat . ' ORDER BY id DESC;";

// sending the query
if ($response = pg_query($conn, $sql)) {
    echo "A query error occured.\n";
    exit;
}

// echo the data back
while ($row = pg_fetch_row($response)) {

    foreach($row as $i => $attr)
        echo $attr . ", ";
    echo " ";
}

?>
```

```
.margin-0 {
    margin-bottom: 0px;
    margin-top: 0px;
}

.padding-top-bottom {
    padding-top: 50px !important;
    padding-bottom: 50px !important;
}

/**** For navbar *****/

.navButton {
    font-size: 30px;
    cursor: pointer;
    color: white;
    padding-right: 30px;
    padding-left: 10px;
}

.sidenav {
    height: 100%; /* 100% Full-height */
    width: 0; /* 0 width - change this with JavaScript */
    position: fixed; /* Stay in place */
    z-index: 1; /* Stay on top */
    top: 0; /* Stay at the top */
    left: 0;
    background-color: #343b40;
    overflow-x: hidden; /* Disable horizontal scroll */
    padding-top: 60px; /* Place content 60px from the top */
    transition: 0.5s; /* 0.5 second transition effect to slide in the sidenav */
}

/* The navigation menu links */
.sidenav a {
    padding: 8px 8px 8px 32px;
    text-decoration: none;
    font-size: 25px;
    color: #007bff;
    display: block;
    transition: 0.3s;
}

/* When you mouse over the navigation links, change their color */
.sidenav a:hover {
    color: #ffffff;
}

/* Position and style the close button (top right corner) */
.sidenav .closebtn {
    position: absolute;
    top: 0;
    right: 25px;
    font-size: 36px;
    margin-left: 50px;
}

/* On smaller screens, where height is less than 450px, change the style of the sidenav (less padding and a smaller font size) */
@media screen and (max-height: 450px) {
    .sidenav {
        padding-top: 15px;
    }
    .sidenav a {

```

```
/* Importing external font family: */
@import url('https://fonts.googleapis.com/css?family=Roboto');

{
    font-family: 'Roboto', 'Verdana', sans-serif;
}

html, body {
    position: relative;
    height: 100%;
    width: 100%;
    margin: 0;
}

h1, h2, h3 {
    text-align: center;
}

.bg-light-grey {
    background-color: #6c757d;
}

.bg-light-grey label:hover {
    background-color: #545b62;
}

#mapid {
    height: 95%;
    width: 100%;
    margin: 0 auto;
    z-index: -1;
}

/**** Banner ****/
.banner {
    background-image: linear-gradient(rgb(0, 0, 0, 0.5), rgb(0, 0, 0, 0.5)), url("6.jpeg");
    height: 100%;
    background-position: center;
    background-repeat: no-repeat;
    background-size: cover;
    position: relative;
}

.banner-text {
    text-align: center;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    color: white;
}

/**** General Css ****/
.clear::after {
    content: " ";
    display: block;
    height: 0;
    clear: both;
    visibility: hidden;
}

.container {
    width: 80%;
    margin: auto;
}

font-size: 18px;
}
```

Παράρτημα Β

Στο παράρτημα αυτό, ακολουθεί όλος ο κώδικας που έχει γραφτεί για την υλοποίηση της TrafficWare εφαρμογής που χρησιμοποιήθηκε για τη συλλογή τροχιών.

```

package ucy.cs.cellularmetrics;

import android.Manifest;
import android.annotation.SuppressLint;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.app.Activity;
import android.os.Environment;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.telephony.CellInfo;
import android.telephony.CellInfoCdma;
import android.telephony.CellInfoGsm;
import android.telephony.CellInfoLte;
import android.telephony.CellInfoWcdma;
import android.telephony.CellLocation;
import android.telephony.PhoneStateListener;
import android.telephony.SignalStrength;
import android.telephony.TelephonyManager;
import android.text.method.ScrollingMovementMethod;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.List;

public class MainActivity extends Activity {
    int counter = 0;
    int changes = 0;
    TelephonyManager mTelephonyManager;
    TextView tvNetworkInfo;
    TextView txtList;
    Location location; // location
    double latitude = 0.0; // latitude
    double longitude = 0.0; // longitude
    LocationManager locationManager;
    // getting GPS status
    boolean isGPSEnabled;
    boolean permissions = false;

    private final LocationListener locationListener = new LocationListener() {
        public void onLocationChanged(Location location) {
            longitude = location.getLongitude();
            latitude = location.getLatitude();
        }
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
    }

    @Override
    public void onProviderEnabled(String provider) {
    }
}

```

```

    @Override
    public void onSignalStrengthsChanged(SignalStrength signalStrength) {
        onCellInfoChanged(null);
    }

    @Override
    public void onCellLocationChanged(CellLocation location) {
        onCellInfoChanged(null);
    }

    @SuppressLint("NewApi", "MissingPermission")
    @Override
    /**
     * Gets the cell info
     * @param cellInfoList, the list that contains all cells with their info
     */
    public void onCellInfoChanged(List<CellInfo> cellInfoList) {

        int signal = 0, cid = 0, mnc = 0, mcc = 0, lac = 0, cells = 0;
        String type = "";
        StringBuilder data = new StringBuilder();
        //StringBuilder error = new StringBuilder();

        Long tsLong = System.currentTimeMillis() / 1000;
        String ts = tsLong.toString();

        cellInfoList = mTelephonyManager.getAllCellInfo();
        if (mTelephonyManager.getAllCellInfo() != null) {
            int cellids[] = new int[cellInfoList.size()];
            for (int i = 0; i < cellInfoList.size(); i++) {
                CellInfo info = cellInfoList.get(i);
                if (info instanceof CellInfoGsm) {
                    signal = ((CellInfoGsm) info).getCellSignalStrength().getDbm();
                    type = "GSM";
                    cellids[i] = cid;
                    cid = ((CellInfoGsm) info).getCellIdentity().getCid();
                    mnc = ((CellInfoGsm) info).getCellIdentity().getMnc();
                    lac = ((CellInfoGsm) info).getCellIdentity().getLac();
                    mcc = ((CellInfoGsm) info).getCellIdentity().getMcc();
                } else if (info instanceof CellInfoWcdma) {
                    cellids[i] = cid;
                    signal = ((CellInfoWcdma) info).getCellSignalStrength().getDbm();
                    type = "WCDMA";
                    cid = ((CellInfoWcdma) info).getCellIdentity().getCid();
                    mnc = ((CellInfoWcdma) info).getCellIdentity().getMnc();
                    lac = ((CellInfoWcdma) info).getCellIdentity().getLac();
                    mcc = ((CellInfoWcdma) info).getCellIdentity().getMcc();
                } else if (info instanceof CellInfoCdma) {
                    cellids[i] = cid;
                    signal = ((CellInfoCdma) info).getCellSignalStrength().getDbm();
                    type = "CDMA";
                    cid = ((CellInfoCdma) info).getCellBaseStationId();
                    mnc = ((CellInfoCdma) info).getCellIdentity().getSystemId();
                    lac = ((CellInfoCdma) info).getCellIdentity().getNetworkId();

                    if (mnc != 0) mcc = 280;
                } else {
                    cellids[i] = cid;
                    signal = ((CellInfoLte) info).getCellSignalStrength().getDbm();
                    type = "LTE";
                    cid = ((CellInfoLte) info).getCellIdentity().getCi();
                    mnc = ((CellInfoLte) info).getCellIdentity().getMnc();
                    lac = ((CellInfoLte) info).getCellIdentity().getTac();
                    mcc = ((CellInfoLte) info).getCellIdentity().getMcc();
                }
            }
        }
    }

```

```

    @Override
    public void onProviderDisabled(String provider) {
    }
};

    @SuppressLint("MissingPermission")
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        tvNetworkInfo = (TextView) findViewById(R.id.tvNetworkInfo);
        txtList = (TextView) findViewById(R.id.tvNetworkInfo2);
        txtList.setMovementMethod(new ScrollingMovementMethod());

        if (getPermissions()) {
            permissions = true;
            locationManager = (LocationManager) MainActivity.this.getSystemService(LOCATION_SERVICE);
            isGPSEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

            if (locationManager != null) {
                location = locationManager
                        .getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                if (location != null) {
                    latitude = location.getLatitude();
                    longitude = location.getLongitude();
                }
            }

            locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 2000, 10, locationListener);
        } else {
            Toast toast = Toast.makeText(getApplicationContext(), "You have to give the right permissions",
                    LENGTH_LONG);
            toast.show();
        }
    }

    /**
     * Listener that waits the button to be clicked
     */
    @param view, the view of the app
    */
    public void clicked(View view) {
        Button bt = findViewById(R.id.button_start);
        if (permissions) {
            if (counter % 2 == 0) {
                bt.setText("Stop");
                onStart();
            } else {
                bt.setText("Start");
                onStop();
            }
            counter++;
        }
    }

    /**
     * This is the listener that waits the signal to change, or the cells to change
     * so that it can save the cell info
     */
    private PhoneStateListener mPhoneStateListener = new PhoneStateListener() {
    }
}

```

```

        }
        //if ((cid == 2147483647) || (cid == -1))
        // error.append(ts + " " + latitude + " " + longitude + " " + type + " ");
        //else {
        boolean flag = false;
        if (cid != 2147483647 && mcc == 280) {
            for (int j = 0; j < i; j++)
                if (cellids[j] == cid) {
                    flag = true;
                    break;
                }
            if (!flag)
                data.append(ts + " " + latitude + " " + longitude + " " + type + " " + signal + " " + cid + " " + mnc + " " + lac +
                        " " + "\n");
            cells++;
        }
        txtList.append("\n Timestamp: " + ts + "\n Lat: " + latitude + "\n Lon: " + longitude +
                "\n Type: " + type + "\n Signal: " + signal + "\n CellId: " + cid + "\n MNC: " + mnc + "\n LAC: " + lac + "\n");
    } else Log.i("info", "null cellinfo");
    if (data.length() > 0) writeFile(data, 0);
    //if (error.length() > 0) writeFile(error, 1);
    tvNetworkInfo.setText("Changes in Cell info: " + (changes++) + " Current Cells: " + cells);
}
}

```

```

    @Override
    /**
     * When the 'Start' button is pressed, starts getting cell info
     */
    public void onStart() {
        super.onStart();

        // Code below register your mPhoneStateListener will start to be called everytime cell info changed.
        // If you update any UI element, be sure that they were created already (since they are created during "onCreate".. and not
        at onStart)
        // I added LISTEN_CELL_LOCATION.. But I think that PhoneStateListener.LISTEN_CELL_INFO is enough
        mTelephonyManager = (TelephonyManager) getSystemService(TELEPHONY_SERVICE);
        mTelephonyManager.listen(mPhoneStateListener, PhoneStateListener.LISTEN_CELL_LOCATION |
                PhoneStateListener.LISTEN_CELL_INFO | PhoneStateListener.LISTEN_SIGNAL_STRENGTHS);
    }

    @Override
    /**
     * When the 'Stop' button is pressed, stops getting cell info
     */
    public void onStop() {
        // Code below unregister your listener... You will not receive any cell info change notification anymore
        TelephonyManager mTelephonyManager = (TelephonyManager) getSystemService(TELEPHONY_SERVICE);
        mTelephonyManager.listen(mPhoneStateListener, PhoneStateListener.LISTEN_NONE);
        super.onStop();
    }

    /**
     * This method writes into a file the cell measurements and errors if there are any kind of errors
     *
     * @param data, the data the will be written into the file.
     * @param type, 0 for measurements, 1 for errors
     */
    private void writeFile(StringBuilder data, int type) {

        File file = new File(Environment.getExternalStorageDirectory(), "/mymetrics");
        if (file.exists()) {
            file.mkdir();
        }
    }
}

```

```

try {
    File gpxfile;
    if (type == 0)
        gpxfile = new File(file, "measurements.csv");
    else gpxfile = new File(file, "errors.csv");

    //Read text from file
    StringBuilder text = new StringBuilder();
    try {
        BufferedReader br = new BufferedReader(new FileReader(gpxfile));
        String line;

        while ((line = br.readLine()) != null) {
            text.append(line);
            text.append("\n");
        }
        br.close();
    } catch (IOException e) {
        Log.e("Error", "Writing in file");
    }

    // write text into the file
    FileWriter writer = new FileWriter(gpxfile);
    writer.append(String.valueOf(text) + String.valueOf(data));

    writer.flush();
    writer.close();

} catch (Exception e) {
    Log.e("Exception", "File write failed: " + e.toString());
}

}

/**
 * This method gets the permissions for everything that the application uses
 */
public boolean getPermissions() {

    int result = 0, coarseLocation, write, read, fineLocation;

    // Permission for getting the device's location
    if (ContextCompat.checkSelfPermission(MainActivity.this,
        Manifest.permission.ACCESS_COARSE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {

        // No explanation needed; request the permission
        ActivityCompat.requestPermissions(MainActivity.this,
            new String[]{Manifest.permission.ACCESS_COARSE_LOCATION},
            result);

        coarseLocation = result;
    } else {
        // Permission has already been granted
        coarseLocation = 1;
    }

    // Writing Permission
    if (ContextCompat.checkSelfPermission(MainActivity.this,
        Manifest.permission.WRITE_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED) {

        // No explanation needed; request the permission

```

```

        ActivityCompat.requestPermissions(MainActivity.this,
            new String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE},
            result);

        write = result;

    } else {
        // Permission has already been granted
        write = 1;
    }

    // Reading Permission
    if (ContextCompat.checkSelfPermission(MainActivity.this,
        Manifest.permission.READ_EXTERNAL_STORAGE)
        != PackageManager.PERMISSION_GRANTED) {

        // No explanation needed; request the permission
        ActivityCompat.requestPermissions(MainActivity.this,
            new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
            result);

        read = result;

    } else {
        // Permission has already been granted
        read = 1;
    }

    // Permission for getting the device's location
    if (ContextCompat.checkSelfPermission(MainActivity.this,
        Manifest.permission.ACCESS_FINE_LOCATION)
        != PackageManager.PERMISSION_GRANTED) {

        // No explanation needed; request the permission
        ActivityCompat.requestPermissions(MainActivity.this,
            new String[]{Manifest.permission.ACCESS_FINE_LOCATION},
            result);

        fineLocation = result;
    } else {
        // Permission has already been granted
        fineLocation = 1;
    }

    return (coarseLocation == 1 && fineLocation == 1 && read == 1 && write == 1);
    //return true;
}
}

```

Παράρτημα Γ

Στο παράρτημα αυτό, ακολουθεί όλος ο κώδικας που έχει γραφτεί για την πραγματοποίηση των πειραμάτων.

```
package trafficware_java;

import java.io.BufferedReader;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.LinkedHashSet;
import java.util.List;
import java.util.Set;
import java.util.regex.Pattern;

/**
 * This class is reading from a file, and then prints only the cellid and lac
 * of the measurement that has lac = 231
 */
public class PlainJavaApp {

    // pattern to split later on commas
    private static final Pattern COMMA = Pattern.compile(",");

    /**
     * This method reads the file, stores it into a list, and then
     * returns it
     * @param filename, the name of the file
     * @return the list that contains the file's context
     */
    private List<String> readFile(String filename) {

        List<String> records = new ArrayList<String>();

        try {

            BufferedReader reader = new BufferedReader(new FileReader(filename));
            String line;

            while ((line = reader.readLine()) != null) {
                records.add(line);
            }

            reader.close();
            return records;

        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }

    }

    /**
     * This method filters the list, prints only the distinct cellid and lac
     * with lac = 231
     * @param records, the file's records
     */
    private void filterLacCellids(List<String> records) {

        Set<String> results = new LinkedHashSet<String>();

        records.forEach(x -> {
            List<String> elements = Arrays.asList(COMMA.split(x));
            if (elements.get(7).compareTo("231") == 0)
                results.add(elements.get(5) + "," + elements.get(7));
        });

        results.forEach(x -> {
            System.out.println(x.toString());
        });

    }

}
```

```
package cs.uci.ac.cy.trafficware;

import org.apache.spark.api.java.JavaSparkContext;

import scala.Tuple2;

import org.apache.spark.api.java.JavaRDD;

import java.lang.reflect.Array;
import java.util.Arrays;
import java.util.List;
import java.util.regex.Pattern;

import org.apache.spark.SparkConf;
import org.apache.spark.api.java.function.Function;

/**
 * This class is reading from a file, and then prints only the cellid and lac
 * of the measurement that has lac = 231 using spark
 */
public class App {

    // pattern to split later on commas
    private static final Pattern COMMA = Pattern.compile(",");

    public static void main(String[] args) {

        // create spark app
        SparkConf conf = new SparkConf().setAppName("TrafficWare");
        JavaSparkContext sc = new JavaSparkContext(conf);

        // read file
        JavaRDD<String> distFile = sc.textFile("filepath");

        // for each line do the following
        distFile.mapToPair(x -> {
            Integer lac = 0;
            Integer cellid = 0;

            // separates elements with comma
            List<String> elements = Arrays.asList(COMMA.split(x));

            // stores lac and cellid
            try {

                lac = Integer.parseInt(elements.get(7));
                cellid = Integer.parseInt(elements.get(5));

            } catch (NumberFormatException e) {

                return new Tuple2<Integer, Integer>(0, 0);

            }

            // converts cellid and lac into a tuple
            return new Tuple2<Integer, Integer>(lac, cellid);

        // now for each tuple, check if lac = 231, distinct elements, and print them
        }).filter(x -> x._1() == 231).distinct().foreach(word -> {

            System.out.println(word.toString());

        });

        sc.close();
    }

}
```

```
});

}

public static void main(String args[]) {

    PlainJavaApp consumer = new PlainJavaApp();
    List<String> myMeasurements = consumer.readFile("filepath");
    consumer.filterLacCellids(myMeasurements);
}

}
```

```

// requires to read from file
var fs = require('fs');

// function to apply distinct on array elements
const distinct = (value, index, self) => {

  return self.indexOf(value) === index;
}

// read file
fs.readFile('filepath', 'utf8', function(err, contents) {
  // split lines with new line
  var lines = contents.split(/\r?\n/);
  var array = [];

  // for each line store only the ones with LAC = 231
  lines.forEach(function(element) {

    var line = element.split(",");

    if (line[7] === "231")
      array.push(line[5] + "," + line[7]);

  });

  // filter data
  var filtered = array.filter(distinct);

  //print
  filtered.forEach(function(element) {
    console.log(element);
  });
});

```

```

<?php
ini_set('display_errors', 1);

// read from file
$files = file_get_contents("filepath");

// split new lines
$file = explode(PHP_EOL, $file);
$first = 0;
$array = array();

// foreach line do the following
foreach($file as $line) {

  // ignore first line
  if ($first == 0) $first = 1;
  else {
    // separate line with comma
    $str = explode(",", $line);

    // check if lac = 231
    if ($str[7] == 231)
      array_push($array, $str[5] . "," . $str[7]);
  }

}

// print only distinct lines
$result = array_unique($array);
print_r($result);
?>

```