



## ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΤΠΟΛΟΓΙΣΤΩΝ

**Σύστημα προσομοίωσης εκτέλεσης  
αλγορίθμων συνάρθροισης δεδομένων με  
χρήση Google Maps**

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλέκα Κ. Σεληνιωτάκη

Εγκρίθηκε απ' την τριμελή επιτροπή:

Δεληγιαννάκης Αντώνιος      Γαροφαλάκης Μίνως      Μπλέτσας Άγγελος  
Επικούρος Καθηγητής                  Καθηγητής                  Επίκουρος Καθηγητής  
**Επιβλέπων**

7 Μαρτίου 2011



# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>11</b>
<b>2</b>	<b>Δίκτυα Αισθητήρων</b>	<b>14</b>
2.1	Κόμβος Αισθητήρας . . . . .	14
2.2	Δίκτυο Αισθητήρων . . . . .	16
2.3	Παράγοντες Σχεδιασμού Δικτύων	
	Αισθητήρων . . . . .	18
	2.3.1 Κατανάλωση Ισχύος . . . . .	19
	2.3.2 Ανοχή σε αποτυχίες κόμβων . . . . .	19
	2.3.3 Επεκτασιμότητα . . . . .	20
	2.3.4 Κόστος Παραγωγής . . . . .	20
	2.3.5 Περιοσρισμοί Υλικού . . . . .	20
	2.3.6 Τοπολογία Δικτύου Αισθητήρων . . . . .	21
	2.3.7 Περιβάλλον . . . . .	21
	2.3.8 Μέσα Μετάδοσης . . . . .	21
<b>3</b>	<b>Google Maps</b>	<b>22</b>
3.1	Google Maps και Google Maps API . . . . .	24
3.2	Ο Χάρτης . . . . .	25
	3.2.1 Συντεταγμένες (γεωγραφικό μήκος και πλάτος) . . . . .	25
	3.2.2 Δημιουργία του χάρτη . . . . .	26
3.3	Markers . . . . .	32
	3.3.1 MarkersOptions . . . . .	33
	3.3.2 Τοποθέτηση αισθητήρων . . . . .	34

3.3.3	Events	36
3.4	InfoWindows	37
3.5	Polylines	37
3.5.1	PolylineOptions	39
3.5.2	Events	41
3.5.3	Τοποθέτηση των Polylines	41
3.6	Polygons	43
3.6.1	PolygonOptions	44
3.6.2	Events	45
3.6.3	Τοποθέτηση πολυγώνου στο χάρτη	45
3.6.4	Εντοπισμός αισθητήρων εντός του πολυγώνου.	47
3.7	Κύκλος	49
3.7.1	CircleOptions	49
3.7.2	Τοποθέτηση σαρωτή πάνω στο χάρτη	50
3.8	Σύνοψη	52
<b>4</b>	<b>Παρουσίαση γραφικής διεπαφής</b>	<b>53</b>
4.1	Sensors' Field	54
4.2	Algorithms' Field	55
4.3	Simple/Real Operation	56
4.4	Data Selection Options	57
4.5	More Options	58
4.6	Drop Down Menus	58
<b>5</b>	<b>Αλγόριθμοι Συνάθροισης Δεδομένων</b>	<b>61</b>
5.1	Επερωτήσεις (Queries)	63
5.2	Συναθροιστικές Συναρτήσεις	64
5.3	TAG (Tiny AGgregation)	66
5.3.1	Κύρια ιδέα του TAG	66
5.3.2	Μετάδοση μηνυμάτων	67
5.3.3	Διαμέριση της κάθε εποχής	69
5.3.4	Πλεονεκτήματα του TAG	69

5.3.5	Σύνοψη . . . . .	70
5.4	TiNA (Temporal Coherency-Aware in-Network Aggregation) . . . . .	71
5.4.1	Κύρια Ιδέα . . . . .	71
5.4.2	Η συνάθροιση στο TiNA . . . . .	72
5.4.3	Η συμπεριφορά των αισθητήρων στο TiNA . . . . .	73
5.4.4	Επικείμενα προβλήματα . . . . .	74
5.4.5	Σύνοψη . . . . .	74
5.5	Uniform . . . . .	75
5.5.1	Ενεργοί/Παθητικοί κόμβοι . . . . .	75
5.5.2	Λειτουργία του αλγορίθμου . . . . .	76
5.5.3	Σύνοψη . . . . .	77
5.6	PGA (Potential Gains Adjustment) . . . . .	77
5.6.1	Κύρια Ιδέα . . . . .	77
5.6.2	Λειτουργία Αλγορίθμου . . . . .	78
5.6.3	Τπολογισμός στατιστικών παραμέτρων . . . . .	79
5.6.4	PGA_Root και PGA_Local . . . . .	81
5.6.5	Σύνοψη . . . . .	81
<b>6</b>	<b>Πειράματα</b>	<b>83</b>
6.1	Τλοποίηση SQL επερωτήσεων στη γραφική διεπαφή . . . . .	83
6.1.1	Τλοποίηση SQL επερωτήσεων . . . . .	83
6.2	Απόδοση Αλγορίθμων . . . . .	93
6.2.1	TAG - TiNA . . . . .	93
6.2.2	TiNA από 0% έως 25% . . . . .	96
6.2.3	PGA και Uniform . . . . .	97
6.2.4	PGA_Root και PGA_Local . . . . .	100
<b>Βιβλιογραφία</b>		<b>101</b>

# Κατάλογος Σχημάτων

2.1	Μονάδες που απαρτίζουν τον κόμβο-αισθητήρα.	15
2.2	Ενδεικτική συσκευή κόμβου-αισθητήρα.	16
2.3	Συνήθης Αρχιτεκτονική Δικτύου Αισθητήρων.	18
3.1	Το κέντρο του κόσμου με γεωγραφικό πλάτος και μήκος 0, κείται λίγο έξω απ'τις δυτικές ακτές της Αφρικής.	26
3.2	Οι διαθέσιμοι τύποι εμφάνισης του χάρτη.	29
3.3	Οι διαθέσιμοι τύποι εμφάνισης του navigation control.	30
3.4	Ο δορυφορικός χάρτης που δημιουργήθηκε, με επίκεντρο την Πάρνηθα και επίπεδο εστίασης ίσο με 10.	31
3.5	Το μενού που εμφανίζεται πάνω στο χάρτη κάνοντας με το ποντίκι διπλό click πάνω σε αυτόν.	32
3.6	Χρήση της ιδιότητας "title".	33
3.7	Εικόνες που απεικονίζουν τους αισθητήρες.	34
3.8	Η τοποθέτηση των αισθητήρων εντός των ορίων που καθορίστηκαν με την κλάση LatLangBounds.	35
3.9	Συνέπειες των γεγονότων click και rightclick αντίστοιχα.	36
3.10	Μία Polyline αποτελείται από σημεία που ενώνονται με γραμμές.	38
3.11	Οι polylines απεικονίζουν την τοπολογία του δικτύου αισθητήρων.	39
3.12	To InfoWindow που εμφανίζεται με μονό click πάνω στη γραμμή.	41
3.13	Σε ένα πολύγωνο, το σημείο έναρξης ενώνεται με το σημείο τερματισμού.	43
3.14	Ενδεικτικό πολύγωνο εφαρμογής, χρησιμοποιώντας τις ιδιότητες του αντικειμένου PolygonOptions.	44

3.15 Το InfoWindow ενός πολυγώνου. . . . .	45
3.16 Διαδοχικές φάσεις τοποθέτησης ενός πολυγώνου. . . . .	46
3.17 Ένα πολύγωνο με 14 πλευρές. Η κόκκινη τελεία αποτελεί το σημείο για το οποίο πρέπει να αποφασιστεί αν κείται εντός του πολυγώνου. . . . .	47
3.18 Διάφορες μορφές πολυγώνων με σημεία εντοπισμού. . . . .	48
3.19 Ενδεικτικός κύκλος που δημιουργείται χρησιμοποιώντας τις ιδιότητες του αντικειμένου CircleOptions. . . . .	50
3.20 Σταδιακή αναπαράσταση σάρωσης. . . . .	51
4.1 Σύστημα προσομοίωσης εκτέλεσης αλγορίθμων. . . . .	53
4.2 Το πεδίο Sensors' Field της φόρμας εισαγωγής δεδομένων. . . . .	54
4.3 Το πεδίο Algorithms' Field της φόρμας εισαγωγής δεδομένων. . . . .	56
4.4 Παράμετροι δεδομένων για την απλή λειτουργία και τη λειτουργία πραγ- ματικού χρόνου. . . . .	57
4.5 Πεδίο για την προέλευση των δεδομένων. . . . .	58
4.6 . . . . .	59
5.1 Οι μερικώς συναθροιστικές τιμές ρέουν προς την ρίζα κατά τη διάρκεια μίας εποχής. . . . .	70
5.2 Δέντρο Συνάθροισης δεδομένων και κατανομή φίλτρων. . . . .	76
5.3 Ενδεχόμενο κέρδος ενός κόμβου . . . . .	80
6.1 Τοποθέτηση δικτύου αισθητήρων στην Πάρνηθα. . . . .	85
6.2 SQL επερώτηση №1. . . . .	87
6.3 SQL επερώτηση №2. . . . .	89
6.4 Αποτελέσματα SQL επερώτησης №2. . . . .	90
6.5 SQL επερώτηση №3. . . . .	92
6.6 Σύγκριση αλγορίθμων TAG και TiNA(0%) . . . . .	95
6.7 TiNA με $tct \in [0\%, 25\%]$ . . . . .	96
6.8 Συμπεριφορά των αλγορίθμων Uniform και PGA_Root αλλάζοντας την παράμετρο Upd. . . . .	97
6.9 Απόδοση των αλγορίθμων PGA_Root και Uniform για τις δύο τοπολογίες του δικτύου. . . . .	99

6.10 Σύγκριση αλγορίθμων PGA_Root και PGA_Local. . . . .	100
--	-----

# Περίληψη

Τα ασύρματα δίκτυα αισθητήρων είναι μια συνεχώς αναπτυσσόμενη και εξελισσόμενη τεχνολογία, η οποία έχει αλλάξει τον τρόπο με τον οποίο ο άνθρωπος αντιλαμβάνεται το περιβάλλον. Δίκτυα αισθητήρων έχουν χρησιμοποιηθεί σε ποικίλες εφαρμογές όπως στην ιατρική, στην παρακολούθηση του περιβάλλοντος, σε «έξυπνα σπίτια», κ.α. Αντίθετα με τα συμβατικά ασύρματα δίκτυα, τα δίκτυα αισθητήρων έχουν σχεδιαστεί για να λειτουργούν χωρίς επιτήρηση για μεγάλα χρονικά διαστήματα στα οποία είναι συνήθως δύσκολο, έως αδύνατο να αντικατασταθεί ή να επαναφορτιστεί η πηγή ενέργειάς τους (δηλ. μπαταρία). Με την παρακολούθηση των ασύρματων δικτύων και την παρατήρηση της συμπεριφοράς αυτών υπό ορισμένες συνθήκες, επιτυγχάνεται η σωστότερη επιλογή κάποιων παραμέτρων που μειώνουν την ενέργεια του ασύρματου δικτύου και διατηρούν την ποιότητα των δεδομένων σε υψηλά επίπεδα. Ενδεικτικά, κάποια παραδείγματα παραμέτρων είναι το πλήθος των αισθητήρων, ο αλγόριθμος συνάθροισης δεδομένων ή η τοπολογία του δικτύου αισθητήρων. Η κατάλληλη επιλογή όλων των παραγόντων αυξάνει τη διάρκεια ζωής και την αξιοπιστία του ασύρματου δικτύου.

Στην παρούσα διπλωματική εργασία αναπτύχθηκε ένα σύστημα προσομοίωσης εκτέλεσης αλγορίθμων συλλογής δεδομένων σε δίκτυα αισθητήρων. Είναι ένα σύστημα επεξεργασίας ερωτημάτων (queries) που τίθενται απ'το χρήστη για την εξόρυξη πληροφοριών απ'το δίκτυο αισθητήρων. Δοθέντος μίας επερώτησης που έχει καθοριστεί απ'τα δεδομένα που μας ενδιαφέρουν (πλήθος αισθητήρων, ρυθμός συλλογής δεδομένων, είδος μέτρησης κ.α), συλλέγονται οι κατάλληλες μετρήσεις, συναθροίζονται βάσει του αλγορίθμου που έχει επιλεγεί και διαβιβάζονται στο χρήστη μέσω μίας διεπαφής που έχει αναπτυχθεί με χρήση των Google Maps. Τα

δεδομένα των αισθητήρων που χρησιμοποιούνται για την προσομοίωση των παραπάνω αλγορίθμων, είναι είτε πραγματικά δεδομένα από το Intel Berkeley Research lab τα οποία αποθηκεύτηκαν σε μία βάση δεδομένων, είτε προέρχονται από μία γεννήτρια τυχαίων τιμών, οι οποίες ακολουθούν το πρότυπο του "τυχαίου περίπατου" (random walk pattern).

## Ευχαριστίες

Θα ήθελα αρχικά να πω ένα μεγάλο ευχαριστώ στον επιβλέποντα καθηγητή της διπλωματικής μου εργασίας κ.Αντώνιο Δεληγιαννάκη για την εμπιστοσύνη που μου έδειξε στο να φέρω εις πέρας το συγκεκριμένο θέμα. Με την εκπόνηση αυτής της εργασίας γνώρισα τον τομέα των ασύρματων δικτύων αισθητήρων, για τα οποία γνώριζα ελάχιστα πράγματα. Το θέμα της διπλωματικής αποτέλεσε επίσης το βασικό μου χριτήριο για την επιλογή του μεταπτυχιακού προγράμματος, που θα ακολουθήσω τα επόμενα ακαδημαϊκά έτη στο Πανεπιστήμιο Κρήτης και ίσως και στην επαγγελματική μου πορεία. Οι υποδείξεις και οι παρατηρήσεις του κ.Δεληγιαννάκη ήταν πολύ χρήσιμες και βοήθησαν στο να γίνει ο τρόπος εργασίας μου αποδοτικότερος.

Θα ήθελα επίσης να ευχαριστήσω τους κυρίους Μίνωα Γαροφαλάκη και Άγγελο Μπλέτσα για το χρόνο που αφιέρωσαν για την ανάγνωση και την αξιολόγηση της διπλωματικής μου εργασίας και για τις εποικοδομητικές παρατηρήσεις και συμβουλές τους.

Δεν μπορώ να παραλείψω να ευχαριστήσω την κ.Πολυζένη Αράπη, η οποία αφιέρωσε πολύ χρόνο για την επίλυση ενός προβλήματος που αντιμετώπισα κατά την εκπόνηση της διπλωματικής μου εργασίας.

Τέλος, ευχαριστώ και αφιερώνω την εργασία αυτή στους γονείς μου (Κωνσταντίνο και Αγγελική) για την στήριξη που μου προσέφεραν καθ'όλη τη διάρκεια των σπουδών μου στο Πολυτεχνείο. Η αγάπη και η βοήθειά τους ήταν πραγματικά πολύτιμη για μένα.

# Κεφάλαιο 1

## Εισαγωγή

Σκοπός της παρούσας διπλωματικής εργασίας είναι η προσομοίωση λειτουργίας αλγορίθμων συλλογής δεδομένων, παρέχοντας ταυτόχρονα δυνατότητες απεικόνισης μέσω Google Maps. Προσφέρεται στο χρήστη μία γραφική διεπαφή, στην οποία ανατίθενται επερωτήσεις (queries) με SQL-μορφή. Μέσω των επερωτήσεων καθορίζονται τα δεδομένα προς απόσπαση από το δίκτυο αισθητήρων όπως επίσης και άλλοι παράμετροι. Αναφορικά, κάποιοι παράμετροι είναι ο ρυθμός με τον οποίο ανανεώνονται τα δεδομένα ή το είδος της τοπολογίας του δέντρου συνάθροισης δεδομένων. Η διεπαφή, πέραν της απλής επιλογής παραμέτρων, παρέχει την εύκολη συλλογή πληροφοριών είτε από το δίκτυο, είτε από τους κόμβους - αισθητήρες μεμονωμένα. Επιπλέον, παρέχονται πληροφορίες συνδεσιμότητας, ενώ ταυτόχρονα επιτρέπεται η επιλογή του συνόλου των κόμβων, για τους οποίους θέλουμε να εκτελέσουμε επερωτήματα.

Πρωτίστως, ο χρήστης καλείται να εγκαταστήσει το δίκτυο αισθητήρων, το οποίο αποτελείται από ένα σταθμό-βάση και από τους κόμβους - αισθητήρες. Ο χρήστης μετά την τοποθέτηση του δικτύου μπορεί να αλληλεπιδράσει με τη διεπαφή, αυξάνοντας και μειώνοντας την πυκνότητα των αισθητήρων σε επιθυμητές περιοχές ή ακόμα και αλλάζοντας την τοποθεσία τους. Αφού έχει καθοριστεί το δίκτυο, μπορεί να οριστεί η επιθυμητή επερώτηση μέσω της γραφικής διεπαφής. Ο χρήστης επιλέγει το είδος της μέτρησης προς παρατήρηση, τον τρόπο συλλογής των μετρήσεων (αν συλλέγονται μετρήσεις από όλους τους κόμβους ή μόνο από τους κόμβους -φύλλα) και την επιθυμητή συναθροιστική συνάρτηση. Αφού καθοριστούν οι προηγούμενες

παράμετροι, συλλέγονται τα δεδομένα, συναθροίζονται και δρομολογούνται προς το σταθμό-βάση. Αυτή η διαδικασία πραγματοποιείται με ενεργειακά αποδοτικούς αλγορίθμους επεξεργασίας, που εκτελούνται μέσα στο δίκτυο (in-network aggregation).

Επιπλέον επειδή γνωστά εργαλεία, όπως το TinyDB, παρέχουν τη δυνατότητα εξαγωγής των συλλεγόμενων (πραγματικών) δεδομένων από μία βάση, προσφέρεται απ'το σύστημα που αναπτύχθηκε, η δυνατότητα εκτέλεσης αλγορίθμων με αντίστοιχα δεδομένα. Ο χρήστης καθορίζει απ'τη φόρμα εισαγωγής δεδομένων, αν οι μετρήσεις που θα χρησιμοποιηθούν είναι τυχαίες ή πραγματικές.

Τέλος, υπάρχουν δύο δυνατές λειτουργίες του συστήματος προσομοίωσης: Η απλή λειτουργία και η λειτουργία πραγματικού χρόνου. Στην πρώτη, ο χρήστης καθορίζει το πλήθος των εποχών<sup>1</sup> για τις οποίες θα ”τρέξει” η εφαρμογή, όπως επίσης και τα σφάλματα τα οποία μπορεί να δεχτεί. Με αυτή τη λειτουργία, παρουσιάζονται συγκεντρωτικά αποτελέσματα για το κάθε σφάλμα ξεχωριστά με τις μορφές που αναφέρονται παρακάτω. Σύμφωνα με τη δεύτερη λειτουργία, ο χρήστης παρατηρεί τη συμπεριφορά του δικτύου για κάθε εποχή σε ένα διάστημα (εποχών) που ορίζει ο ίδιος. Πιο συγκεκριμένα, εισάγει τη χρονική διάρκεια της κάθε εποχής, το id της τελευταίας εποχής προς παρατήρηση και ένα συνολικό σφάλμα για την εφαρμογή. Για παράδειγμα, αν ανατεθεί στην πρώτη παράμετρο η τιμή 2s, στη δεύτερη η τιμή 8s και στην τρίτη 10, η εφαρμογή θα έχει όριο ανοχής σφάλματος ίσο με 10 και θα παρουσιάζονται αποτελέσματα στο χρήστη κάθε 2s μέχρι και το 8°s.

Τα αποτελέσματα της επερώτησης μεταφέρονται στο χρήστη με τη μορφή γραφικής παράστασης ή με πίνακα που περιλαμβάνει μετρήσεις. Ο χρήστης μπορεί να ενημερωθεί για το πλήθος των απεσταλμένων μηνυμάτων εντός του δικτύου ή απ'τον κάθε κόμβο αισθητήρα, για την εναπομείνουσα ή καταναλισκόμενη ενέργεια είτε όλου του δικτύου είτε των κόμβων-αισθητήρων μεμονωμένα, για το αποτέλεσμα της τελευταίας συναθροιστικής συνάρτησης και για τη συμπεριφορά της καταναλισκόμενης ενέργειας του δικτύου.

---

<sup>1</sup> Εποχή είναι το χρονικό διάστημα που μεσολαβεί μεταξύ δύο διαδοχικών μετρήσεων του αισθητήρα.

Τα Google Maps είναι ο πιο αποδοτικός τρόπος απεικόνισης χωρικής πληροφορίας, που σε συνδυασμό με το Google Maps API προσφέρει στους χρήστες υψηλά ανταποκρινόμενες υπηρεσίες. Τα Google Maps και οι υπηρεσίες τους αποτέλεσαν την ιδανική πλατφόρμα για την εικονική προσέγγιση του δικτύου αισθητήρων. Επιπροσθέτως, με χρήση του συγκεκριμένου εργαλείου επισημαίνονται ενδιαφέροντα γεγονότα που αφορούν στους αισθητήρες και στο δίκτυο αισθητήρων. Τα Google Maps αποτέλεσαν την κύρια συνιστώσα της παρούσας διπλωματική εργασίας, αφού είναι μία πρωτοποριακή, ευχάριστη και εύκολη προς τη χρήση υπηρεσία.

Η δομή της εργασίας έχει ως εξής: Στο δεύτερο κεφάλαιο πραγματοποιείται μια βιβλιογραφική έρευνα, η οποία αποσκοπεί στην εξοικείωση του αναγνώστη με όρους και ιδιαιτερότητες των ασυρμάτων δικτύων αισθητήρων. Στο κεφάλαιο 3 αναφέρεται η χρήση των Google Maps στην παρούσα διπλωματική εργασία. Παρουσιάζονται οι ακλάσεις και τα χαρακτηριστικά που χρησιμοποιήθηκαν για την ανάπτυξη του γραφικού interface. Ακολουθεί το τέταρτο κεφάλαιο με την περιγραφή της λειτουργικότητας της γραφικής διεπαφής που αναπτύχθηκε. Το πέμπτο κεφάλαιο αναφέρει τί είναι ένας αλγόριθμος συνάθροισης δεδομένων. Επιπλέον, περιγράφεται αναλυτικά η λειτουργία των πέντε αλγορίθμων συνάθροισης δεδομένων που αναπτύχθηκαν στην παρούσα εφαρμογή. Τέλος, στο έκτο κεφάλαιο παρουσιάζονται ενδεικτικά κάποια πειράματα της εφαρμογής. Στο πρώτο μέρος παρουσιάζεται η γραφική διεπαφή και οι δυνατότητες που δίνονται στο χρήστη μέσω αυτής. Στο δεύτερο μέρος του συγκεκριμένου κεφαλαίου φαίνεται και σχολιάζεται η απόδοση των αλγορίθμων συνάθροισης δεδομένων.

## Κεφάλαιο 2

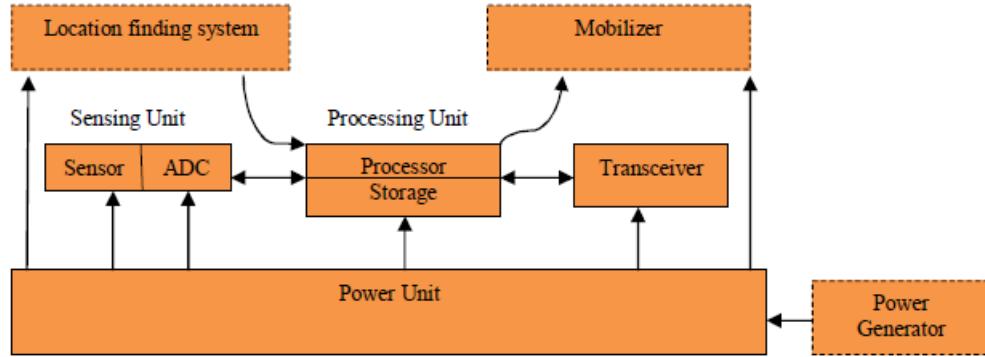
### Δίκτυα Αισθητήρων

Σε αυτό το κεφάλαιο γίνεται μία αναφορά σε όρους και σε ιδιαιτερότητες των ασύρματων δικτύων αισθητήρων. Πιο συγκεκριμένα, στην ενότητα 2.1 αναλύεται η δομή ενός κόμβου αισθητήρα (το κύριο μέρος ενός δικτύου αισθητήρων), στην ενότητα 2.2 αναφέρονται κάποιες βασικές έννοιες για τα δίκτυα αισθητήρων, και τέλος, στην ενότητα 2.3 περιγράφονται οι παράγοντες που επηρεάζουν το σχεδιασμό των δικτύων.

#### 2.1 Κόμβος Αισθητήρας

Ένας κόμβος - αισθητήρας αποτελείται από τέσσερεις βασικές μονάδες, όπως απεικονίζονται στο σχήμα 2.1: μία μονάδα αίσθησης (sensing unit), μία μονάδα επεξεργασίας (processing unit), μία μονάδα εκπομπής - λήψης (transceiver unit) και μία μονάδα ισχύος (power unit). Επίσης, οι κόμβοι μπορεί να έχουν πρόσθια συστατικά μέρη ανάλογα με την εφαρμογή, όπως σύστημα εντοπισμού (location finding system), γεννήτρια ισχύος (power generator) και μονάδα κίνησης (mobilizer) [4].

Οι μονάδες αίσθησης αποτελούνται συνήθως από δύο υπομονάδες : αισθητήρες και μετατροπείς αναλογικού σήματος σε ψηφιακό (ADCs). Τα αναλογικά σήματα που παράγονται από τους αισθητήρες, με βάση το παρατηρούμενο φαινόμενο, μετατρέπονται σε ψηφιακά μέσω του ADC και στη συνέχεια διοχετεύονται στη μονάδα επεξεργασίας. Η μονάδα επεξεργασίας, που στη γενική περίπτωση συνδέεται με μία μικρή μονάδα αποθήκευσης, διαχειρίζεται τις διαδικασίες που επιτρέπουν στον κόμβο

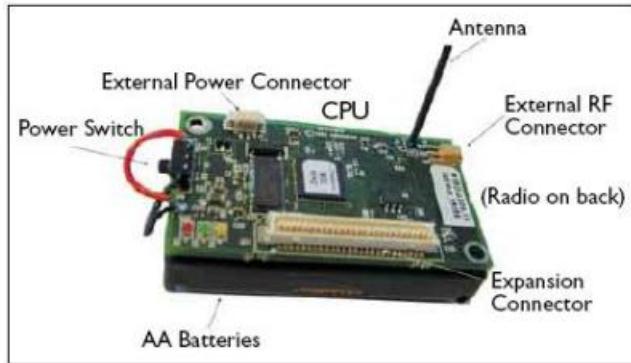


Σχήμα 2.1: Μονάδες που απαρτίζουν τον κόμβο-αισθητήρα.

να συνεργάζεται με τους άλλους κόμβους ώστε να διεκπεραιωθεί το έργο που τους έχει ανατεθεί. Η συνεργασία μεταξύ των κόμβων και, κατ'επέκταση, η σύνδεσή τους στο δίκτυο επιτυγχάνεται με τη μονάδα εκπομπής - λήψης. Από τα πιο σημαντικά συστατικά μέρη ενός κόμβου-αισθητήρα είναι η μονάδα ισχύος. Οι μονάδες ισχύος είναι δυνατόν να υποστηρίζονται από μονάδες συλλογής ισχύος, όπως είναι οι ηλιακές κυψέλες, ή από μπαταρίες επαναφορτιζόμενες ή μη.

Οι περισσότερες από τις τεχνικές δρομολόγησης και τα περισσότερα από τα έργα, που ανατίθενται στους κόμβους, απαιτούν γνώση της τοποθεσίας με υψηλή ακρίβεια. Έτσι, είναι σύνηθες ένας κόμβος να διαθέτει σύστημα εντοπισμού. Κάποιες φορές είναι χρήσιμη μία μονάδα κίνησης για τη μετακίνηση των αισθητήρων, όταν κάτι τέτοιο απαιτείται από το έργο που έχει ανατεθεί στους κόμβους. Όλες οι υπομονάδες είναι πιθανό να πρέπει να τοποθετηθούν σε μία μονάδα με μικρό μέγεθος. Το μέγεθος ενός κόμβου, μπορεί να κυμαίνεται από τις διαστάσεις μίας κούτας υποδημάτων μέχρι και το μέγεθος ενός κόκκου σκόνης, παρ'οτι πλήρως λειτουργικοί κόμβοι με αμιγείς μικροσκοπικές διαστάσεις δεν έχουν ακόμα κατασκευαστεί.

Το κόστος των αισθητήρων κόμβων παρουσιάζει και αυτό μεγάλη διακύμανση και εξαρτάται άμεσα από το μέγεθος του δικτύου και την πολυπλοκότητα της συσκευής. Οι περιορισμοί στο κόστος και στο μέγεθος των αισθητήρων κόμβων έχουν άμεση επίδραση στον περιορισμό των ενεργειακών τους πόρων, της μνήμης, της υπολογιστικής ισχύος τους και του εύρους ζώνης που χρησιμοποιούν.



Σχήμα 2.2: Ενδεικτική συσκευή κόμβου-αισθητήρα.

## 2.2 Δίκτυο Αισθητήρων

Ένα ασύρματο δίκτυο αισθητήρων είναι ένα ασύρματο δίκτυο αποτελούμενο από χωρικά κατανεμημένες αυτόνομες συσκευές - κόμβους που χρησιμοποιούν αισθητήρες. Οι κόμβοι, συνεργαζόμενοι, καταγράφουν φυσικά και κλιματολογικά φαινόμενα όπως θερμοκρασία, υγρασία, ήχο, διονήσεις, πίεση, κίνηση ή ρυπογόνες ουσίες σε διάφορες τοποθεσίες. Η μη προκαθορισμένη θέση των κόμβων και η απαίτηση για συλλογική προσπάθεια προς εκπλήρωση του σκοπού τους, απαιτούν πρωτόκολλα και αλγορίθμους τα οποία θα παρέχουν στο δίκτυο τη δυνατότητα της αυτοργάνωσης (self-organization). Ο τελικός προορισμός των μετρήσεων στα δίκτυα αισθητήρων είναι συνήθως ένα κέντρο συλλογής με μεγάλα ενεργειακά αποθέματα και με δυνατότητα περαιτέρω επεξεργασίας των δεδομένων, με απώτερο σκοπό την λήψη αποφάσεων.

Η ανάπτυξη των ασύρματων δικτύων αισθητήρων ξεκίνησε με τη χρήση τους σε στρατιωτικές επιχειρήσεις, όπως π.χ. την παρακολούθηση του πεδίου μάχης, αλλά σήμερα βρίσκουν ευρεία εφαρμογή τόσο στο βιομηχανικό, όσο και στον ιδιωτικό τομέα. Χαρακτηριστικά παραδείγματα αποτελούν η παρακολούθηση και ο έλεγχος της διαδικασίας παραγωγής μιας βιομηχανίας, η παρακολούθηση κλιματολογικών συνθηκών και οικοσυστημάτων, η χρήση σε ιατρικές εφαρμογές και ο έλεγχος της κυκλοφορίας στα οδικά δίκτυα.

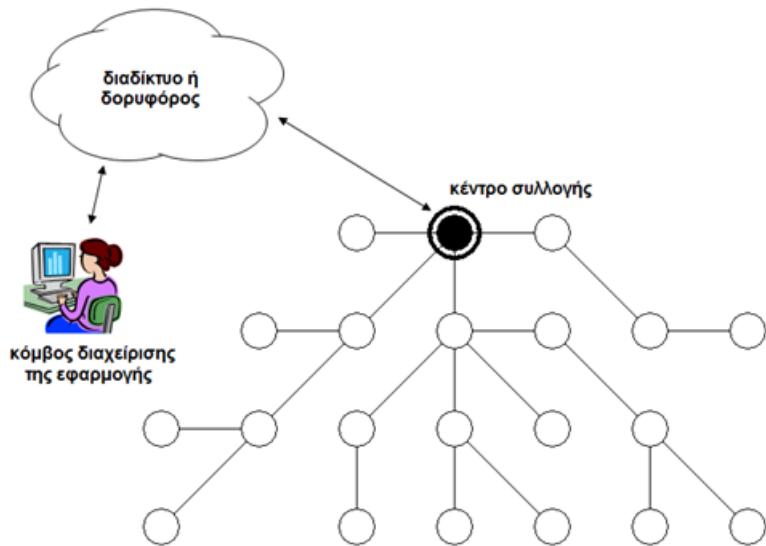
Οι αισθητήρες τοποθετούνται συνήθως με τυχαίο τρόπο στο πεδίο παρατήρησης

(sensor field). Η πυκνότητα των κόμβων στο πεδίο παρατήρησης εξαρτάται άμεσα τόσο από τις απαιτήσεις της εφαρμογής όσο και από αυτές του προς παρατήρηση φαινομένου. Κάθε αισθητήρας κόμβος συλλέγει δεδομένα, τα οποία και προωθεί μέσω των γειτόνων του σε ένα κέντρο συλλογής (sink). Η σταδιακή προώθηση προτιμάται σε σχέση με την απευθείας αποστολή, διότι είναι ενεργειακά πιο αποδοτική, όπως αναλύεται και στο κεφάλαιο 5. Τα μονοπάτια πολλαπλών βημάτων, που δημιουργούνται μέσα στο δίκτυο, καταλήγουν στον ίδιο προορισμό με αποτέλεσμα να διαμορφώνεται μια δενδρική αρχιτεκτονική στο δίκτυο όπως απεικονίζεται στο σχήμα 2.3.

Το κέντρο συλλογής με την σειρά του, επικοινωνεί με τον κόμβο διαχείρισης της εφαρμογής μέσω διαδικτύου ή δορυφόρου. Αξίζει επίσης να σημειωθεί πως τα μονοπάτια μέσα στο δίκτυο δεν είναι προκαθορισμένα και μπορεί να μεταβάλλονται ανάλογα με τις συνθήκες παρατήρησης ή τα ενεργειακά αποθέματα του κάθε κόμβου.

Τα θέματα που σχετίζονται με την διατήρηση και την αλλαγή της τοπολογίας του δικτύου αισθητήρων, χωρίζονται σε τρεις φάσεις:

- **Φάση πριν την τοποθέτηση και φάση τοποθέτησης των κόμβων:** Οι κόμβοι μπορούν να ριφθούν ομαδικά ή να τοποθετηθούν ένας -ένας μέσα στο πεδίο παρατήρησης. Μπορούν να τοποθετηθούν με ρίψη από αεροπλάνο, να διανεμηθούν μέσα σε στρατιωτικά βλήματα ή και να τοποθετηθούν ένας - ένας από ανθρώπους ή ρομπότ.
- **Φάση μετά την τοποθέτηση:** Μετά την τοποθέτηση των κόμβων-αισθητήρων, οι αλλαγές στην τοπολογία του δικτύου μπορούν να οφείλονται σε λόγους, όπως η αλλαγή της θέσης των αισθητήρων, οι μεταβολές στην ακτίνα μετάδοσης τους (λόγω μεγάλης κίνησης, ψηρύζου, κινούμενων εμποδίων κ.τ.λ.), η διαθέσιμη ενέργεια τους, και σε άλλες περιπτώσεις δυσλειτουργίας.
- **Φάση τοποθέτησης επιπρόσθετων κόμβων :** Πρόσθετοι κόμβοι μπορούν να τοποθετηθούν οποιαδήποτε χρονική στιγμή, για να αντικαταστήσουν τους κόμβους που δυσλειτουργούν ή εξαιτίας αλλαγών στη δυναμική του έργου.



Σχήμα 2.3: Συνήμης Αρχιτεκτονική Δικτύου Αισθητήρων.

## 2.3 Παράγοντες Σχεδιασμού Δικτύων Αισθητήρων

Κατά την σχεδίαση ενός ασύρματου δικτύου αισθητήρων πρέπει να ληφθεί υπ' όψιν μια πληθώρα παραγόντων για να ικανοποιηθούν οι απαιτήσεις της εφαρμογής με τον καλύτερο δυνατό τρόπο και με το μικρότερο δυνατό κόστος. Τέτοιοι παράγοντες είναι η κατανάλωση ισχύος (power consumption), η ανοχή σε αποτυχίες (fault tolerance), η επεκτασιμότητα (scalability), το κόστος παραγωγής των αισθητήρων (production costs), οι περιορισμοί υλικού (hardware constraints), η τοπολογία του δικτύου (topology), το περιβάλλον (environment) και τα μέσα μετάδοσης (transmission media).

Οι παραπάνω παράγοντες είναι και αυτοί που οδηγούν στον σχεδιασμό και στην επιλογή των κατάλληλων αλγορίθμων συνάθροισης δεδομένων, έτσι ώστε να αυξηθεί η αποδοτικότητα του δικτύου και να περιοριστεί η άσκοπη κατανάλωση ενέργειας.

### 2.3.1 Κατανάλωση Ισχύος

Ο ασύρματος κόμβος, καθώς είναι μία μικροηλεκτρονική συσκευή, μπορεί να εφοδιαστεί μόνο με περιορισμένη ισχύ . Σε μερικά σενάρια εφαρμογών, η ανανέωση των αποθεμάτων ισχύος μπορεί να είναι αδύνατη. Συνεπώς, η διάρκεια ζωής ενός κόμβου παρουσιάζει ισχυρή εξάρτηση από τη μονάδα συλλογής ισχύος. Σε ένα ad-hoc δίκτυο <sup>1</sup> αισθητήρων πολλαπλών βημάτων, κάθε κόμβος έχει διττό ρόλο: να παράγει ο ίδιος δεδομένα καθώς και να δρομολογεί τα δεδομένα που λαμβάνει από τους γείτονες του. Η δυσλειτουργία μερικών κόμβων μπορεί να προκαλέσει σημαντικές τοπολογικές αλλαγές και ίσως απαιτεί νέα δρομολόγηση των πακέτων και αναδιοργάνωση του δικτύου. Έτσι, η διαφύλαξη ισχύος και η διαχείριση αυτής έχουν αξιοσημείωτη σημασία. Κύριες πηγές κατανάλωσης ενέργειας είναι το άνοιγμα/κλείσιμο του ασυρμάτου, η μετάδοση και η λήψη των δεδομένων, η αναμονή για δεδομένα (χωρίς να λαμβάνεται κάποια μέτρηση εκείνη την ώρα), οι υπολογισμοί στον επεξεργαστή, η λήψη μετρήσεων από τα αισθητήρια όργανα και τέλος, πολύ μικρή κατανάλωση ενέργειας υπάρχει όταν ο κόμβος - αισθητήρας βρίσκεται σε κατάσταση ύπνου.

### 2.3.2 Ανοχή σε αποτυχίες κόμβων

Οι αισθητήρες τροφοδοτούνται από την μπαταρία τους και τοποθετούνται συνήθως σε περιβάλλοντα με αντίξοες συνθήκες (π.χ. πεδίο μάχης, σελήνη). Είναι λοιπόν φυσικό εξαιτίας της περιορισμένης ενέργειάς τους και των παρεμβολών από το περιβάλλον να υποστούν βλάβη ή να διακόψουν την λειτουργία τους λόγω έλλειψης ενέργειας. Ένα τέτοιο γεγονός δεν πρέπει να θέτει σε κίνδυνο την ακεραιότητα ολόκληρου του δικτύου. Πρέπει λοιπόν, το δίκτυο να είναι αξιόπιστο και ανθεκτικό στις αποτυχίες των κόμβων. Η αξιοπιστία του αυτή προέρχεται από την ικανότητα επαναπροσδιορισμού των μονοπατιών πολλαπλών βημάτων από τον κόμβο αισθητήρα στο κέντρο συλλογής.

---

<sup>1</sup> Ad-hoc δίκτυο, είναι μία συλλογή αυτόνομων κόμβων οι οποίοι δεν στηρίζονται σε μία προκαθορισμένη δομή για να κρατάει το δίκτυο σε συνοχή. Οι κόμβοι επικοινωνούν μεταξύ τους χρησιμοποιώντας ασύρματη επικοινωνία και λειτουργούν ακολουθώντας ένα μοντέλο όμοτυμων οντοτήτων (peer-to-peer) [6].

### 2.3.3 Επεκτασιμότητα

Ένα δίκτυο αισθητήρων πρέπει να είναι άμεσα επεκτάσιμο. Πολλές φορές οι κόμβοι αισθητήρες χαλάνε ή σταματούν την λειτουργία τους λόγω έλλειψης ενέργειας και πρέπει να αντικατασταθούν. Σε άλλες περιπτώσεις μπορεί να χριθεί αναγκαία η αύξηση της πυκνότητας των κόμβων ως προς την παρατήρηση της εφαρμογής ή ακόμα και η επέκταση του δικτύου για την κάλυψη μεγαλύτερου πεδίου παρατήρησης. Τα ασύρματα δίκτυα αισθητήρων λοιπόν θα πρέπει να είναι ικανά να λειτουργούν με μεγάλο πλήθος ή/και με μεγάλη πυκνότητα κόμβων. Για το λόγο αυτό, το δίκτυο πρέπει να έχει τη δυνατότητα για άμεση προσαρμογή στις νέες συνθήκες και με όσο το δυνατόν μικρότερο κόστος.

### 2.3.4 Κόστος Παραγωγής

Στις περιπτώσεις όπου η υπό μελέτη εφαρμογή, απαιτεί μεγάλο πλήθος κόμβων, το κόστος παραγωγής ενός κόμβου αποτελεί το σημαντικότερο παράγοντα για τον υπολογισμό του συνολικού κόστους του δικτύου. Συνεπώς το κόστος του κάθε αισθητήρα κόμβου πρέπει να διατηρηθεί χαμηλό. Το κόστος των κόμβων αισθητήρων είναι κυμαίνομενο, μεταξύ μερικών χιλιάδων δολαρίων έως μερικά cents, ανάλογα με τις απαιτήσεις της εφαρμογής και την πολυπλοκότητα των μεμονωμένων κόμβων. Οι περιορισμοί στο κόστος έχουν σαν αποτέλεσμα αντίστοιχους περιορισμούς στην ενέργεια, στη μνήμη, στην υπολογιστική ισχύ και στο εύρος μετάδοσης.

### 2.3.5 Περιορισμοί Υλικού

Όπως έχει αναφερθεί, το μέγεθος του αισθητήρα αποτελεί σημαντικό σχεδιαστικό παράγοντα. Επιπροσθέτως, οι κόμβοι θα πρέπει να καταναλώνουν πολύ χαμηλή ισχύ, να λειτουργούν σε υψηλές χωρικές πυκνότητες, να έχουν χαμηλό κόστος παραγωγής, να είναι αυτόνομοι ώστε να λειτουργούν χωρίς επίβλεψη και να προσαρμόζονται στο περιβάλλον.

### 2.3.6 Τοπολογία Δικτύου Αισθητήρων

Η μεγάλη πυκνότητα κόμβων σε ένα πεδίο παρατήρησης και η επιρρέπεια τους σε συχνές αποτυχίες καθιστούν επιτακτική την ανάγκη διατήρησης σωστής και λειτουργικής τοπολογίας στο δίκτυο. Επιπλέον, η τοπολογία του δικτύου επηρεάζει τις αποφάσεις, που λαμβάνονται όσον αφορά στον αλγόριθμο δρομολόγησης που θα χρησιμοποιηθεί ή στην εφαρμογή, την οποία επιθυμούμε να μελετήσουμε.

### 2.3.7 Περιβάλλον

Οι κόμβοι-αισθητήρες τοποθετούνται με υψηλή πυκνότητα είτε πολύ κοντά είτε ακριβώς μέσα στο προς παρατήρηση φαινόμενο. Έτσι, συνήθως λειτουργούν χωρίς επίβλεψη σε απομακρυσμένες γεωγραφικές περιοχές. Μπορούν να λειτουργούν στο εσωτερικό ενός μεγάλου μηχανήματος, στο πυθμένα ενός ωκεανού, σε ένα βιολογικά ή χημικά μολυσμένο πεδίο, σε ένα πεδίο μάχης πέρα από τις εχθρικές γραμμές, μέσα σε ένα μεγάλο σπίτι ή κτίριο.

### 2.3.8 Μέσα Μετάδοσης

Οι ζεύξεις μεταξύ των κόμβων στα ασύρματα δίκτυα αισθητήρων μπορούν να χρησιμοποιούν ραδιοκύματα, υπέρυθρη ακτινοβολία ή οπτικά μέσα. Σε κάθε περίπτωση, το μέσο μετάδοσης που θα επιλεγεί θα πρέπει να είναι διαθέσιμο παγκοσμίως για να επιτρέπεται η λειτουργία τους από οποιοδήποτε μέρος. Σε ό,τι αφορά την χρήση υπέρυθρης ακτινοβολίας ως μέσο μετάδοσης, αξίζει να σημειωθεί πως παρουσιάζει αρκετά πλεονεκτήματα. Δεν απαιτείται η παραχώρηση άδειας και η επικοινωνία είναι ανθεκτική σε παρεμβολές προερχόμενες από ηλεκτρικές συσκευές. Επίσης οι συσκευές εκπομπής και λήψης υπέρυθρης ακτινοβολίας είναι φυηγές και εύκολες στην κατασκευή τους. Από την άλλη μεριά, υπάρχει αποίτηση οπτικής επαφής μεταξύ αποστολέα και παραλήπτη.

## Κεφάλαιο 3

# Google Maps

Ένα μεγάλο πλήθος από εργαλεία παρακολούθησης του γεωγραφικού χώρου μέσω διαδικτύου, έχουν αναπτυχθεί απ'τις μεγαλύτερες εταιρείες όπως Google, Yahoo, Microsoft και Amazon. Ως "αρχηγός" της νέας αυτής τεχνολογίας έχει αναδυθεί η Google με το προϊόν Google Maps. Το συγκεκριμένο προϊόν χρησιμοποιείται ελεύθερα απ'τους χρήστες, και παρέχει σε αυτούς μία ακριβή και υψηλά ανταποκρινόμενη οπτική διεπαφή (Interface), όπως επίσης και ένα API<sup>1</sup> που δίνει τη δυνατότητα προσθήκης διαφόρων χαρακτηριστικών στους χάρτες. Η συγκεκριμένη διεπαφή, δημιουργήθηκε χρησιμοποιώντας AJAX τεχνολογίες μαζί με λεπτομερή οδικά και εναέρια δεδομένα.

Μέρος της παρούσας διπλωματικής εργασίας ήταν η ανάπτυξη ενός γραφικού περιβάλλοντος για την παρακολούθηση δικτύων αισθητήρων. Για την ανάπτυξη του συγκεκριμένου γραφικού περιβάλλοντος έγινε χρήση των Google Maps, και των βιβλιοθηκών Ext JS 3.2.1 και JFlot. Πιο συγκεκριμένα, δίνεται η δυνατότητα στο χρήστη, μέσω μίας φόρμας εισαγωγής και επιλογής δεδομένων, να τοποθετήσει ένα δίκτυο αισθητήρων στην Πάρνηθα και να παρατηρεί τη συμπεριφορά του, αλλάζοντας παραμέτρους, όπως ο αλγόριθμος συνάθροισης δεδομένων, η τοπολογία του δικτύου, το πλήθος των αισθητήρων κ.λ.π. Με την χρήση των Google Maps, και αφού έχει τοποθετηθεί το δίκτυο αισθητήρων στο χάρτη, ο χρήστης αλληλεπιδρά με αυτό, έχοντας τις παρακάτω επιλογές:

---

<sup>1</sup> Application Programming Interface.

- Έχει οπτική επαιφή του δικτύου και των επιπέδων στα οποία ανήκουν οι κόμβοι. Αυτό σημαίνει, ότι είναι γνώστης της τοπολογίας του δεντρου συνάθροισης δεδομένων <sup>2</sup>.
- Έχει την δυνατότητα αλλαγής της τοπολογίας του δικτύου με την προσθήκη, την αφαίρεση και την μετακίνηση των κόμβων-αισθητήρων.
- Εμφανίζονται ενημερωτικά μηνύματα από κάθε αισθητήρα (InfoWindows). Ο χρήστης μέσω των InfoWindows πληροφορείται από τον κάθε κόμβο-αισθητήρα σχετικά με την γεωγραφική του θέση πάνω στο χάρτη. Επιπλέον, ενημερώνεται για το πλήθος των μηνυμάτων που έχει αποστείλει σε όλη τη διάρκεια ζωής του, για την εναπομείνουσα/καταναλισκόμενη ενέργεια του και τέλος για τον κόμβο-πατέρα του.
- Ο χρήστης μπορεί να ορίσει ένα πολύγωνο στο χάρτη. Με το πολύγωνο, γίνεται διαλογή ενός συγκεκριμένου πεδίου αισθητήρων και μελετάται η συμπεριφορά του.
- Ο χρήστης έχει την επιλογή τοποθέτησης ενός σαρωτή στο χάρτη, ο οποίος μετράει την εναπομείνουσα και καταναλισκόμενη ενέργεια στο εδάφιο που καλύπτει.
- Τέλος, ενημερώνεται για τους αισθητήρες με περιορισμένη εναπομείνουσα ενέργεια, οι οποίοι σε αυτήν την περίπτωση αποκτούν κόκκινο χρώμα. Με αυτόν τον τρόπο, ο χρήστης έχει την δυνατότητα της άμεσης αντικατάστασής τους χωρίς να επηρεαστεί η ακεραιότητα του ασύρματου δικτύου.

H Ext JS 3.2.1 είναι μία cross-browser Javascript βιβλιοθήκη. Παρέχει δωρεάν μία πληθώρα συναρτήσεων για το σχηματισμό διαδικτυακών εφαρμογών. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκαν ορισμένες μέθοδοι προς σχηματισμό μίας φόρμας εισαγωγής δεδομένων. Με τη συγκεκριμένη φόρμα εισαγωγής δεδομένων ο χρήστης σχηματίζει με έμμεσο τρόπο τις επιθυμητές επερωτήσεις (queries) στο δίκτυο αισθητήρων. Αναφορικά με την JFlot, είναι μία βιβλιοθήκη που

---

<sup>2</sup> Δέντρο Συνάθροισης Δεδομένων: Η δενδρική αρχιτεκτονική του δικτύου αισθητήρων, μέσω της οποίας διανέμονται τα αιτήματα των κόμβων σε όλο το δίκτυο.

παρέχει συναρτήσεις για την απεικόνιση γραφικών παραστάσεων στην ιστοσελίδα. Στην συγκεκριμένη περίπτωση χρησιμοποιήθηκαν συναρτήσεις απ' αυτή τη βιβλιοθήκη για να παρουσιαστούν οι γραφικές που απεικονίζουν την απόδοση των αλγορίθμων συνάθροισης δεδομένων.

Στο παρόν κεφάλαιο, αναλύονται οι υπηρεσίες που χρησιμοποιήθηκαν απ' το Google Maps API. Πιο συγκεκριμένα, στο εδάφιο 3.1, περιγράφεται τι είναι Google Maps και Google Maps API, στην ενότητα 3.2 αναφέρεται η διαδικασία με την οποία δημιουργείται ο χάρτης, στο χωρίο 3.3 αναπτύσσεται ο τρόπος με τον οποίο σηματοδοτούνται συγκεκριμένες περιοχές στο χάρτη και στην ενότητα 3.4 θα δούμε τη λειτουργία των InfoWindows. Στα επόμενα εδάφια που ακολουθούν, περιγράφονται τα γεωμετρικά σχήματα που δημιουργήθηκαν για την παρούσα εφαρμογή. Στην ενότητα 3.5 εξηγείται η χρήση των polylines, όπως επίσης και ο τρόπος με τον οποίο δημιουργούνται. Στα εδάφια 3.6 και 3.7, περιγράφονται οι λειτουργίες του πολυγώνου και του κύκλου αντίστοιχα. Τέλος, στο χωρίο 3.8 υπάρχει μία σύνοψη του κεφαλαίου.

### 3.1 Google Maps και Google Maps API

Google Maps είναι μία υπηρεσία απεικόνισης χαρτών στο Web, που προσφέρεται δωρεάν απ' την εταιρεία Google. Όπως αναφέρθηκε και προηγουμένως είναι μία δικτυακή υπηρεσία, που προσφέρει δωρεάν χαρτογραφικά υπόβαθρα και υπόβαθρα με δορυφορικές απεικονίσεις και αεροφωτογραφίες. Η συγκεκριμένη υπηρεσία δεν είναι τίποτα άλλο, παρά η συνάρμοση των HTML, CSS και Javascript μαζί. Ο χάρτης αποτελείται από επιμέρους "πλακίδια", τα οποία είναι εικόνες που φορτώνονται στο προσκήνιο με AJAX κλήσεις. Ακολούθως, φορτώνονται σε ένα <div> tag μίας HTML σελίδας. Καθώς πλοηγούμαστε στο χάρτη, το API αποστέλλει πληροφορίες σχετικά με τα επίπεδα εστίασης (zoom levels) και τις νέες συντεταγμένες στις AJAX κλήσεις, και έτσι επιστρέφονται οι νέες εικόνες [7].

Με το Google Maps API, η Google επιτρέπει την ενσωμάτωση χαρτών Google Maps σε άλλες ιστοσελίδες καθώς και την απεικόνιση δεδομένων σε αυτούς. Πρόκειται για ένα Javascript API που αποτελείται από αρχεία, τα οποία περιλαμβάνουν

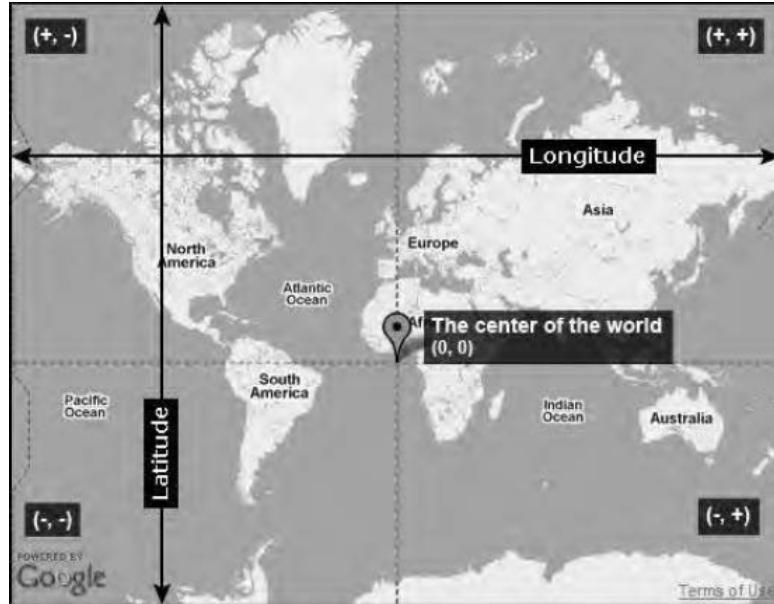
Κλάσεις (Classes) με ιδιότητες (Properties), Συμβάντα (Events) και Μεθόδους (Methods). Τα προηγούμενα χαρακτηριστικά, είναι δομημένα με τρόπο που κάνουν εφικτή την πρόσβαση στα δεδομένα και επιτρέπουν την ανάκτηση αυτών, όπως επίσης και τον εμπλουτισμό τους πριν απ'την απεικόνιση.

## 3.2 Ο Χάρτης

Στο συγκεκριμένο εδάφιο, όπου παρουσιαστούν οι τεχνικές που χρησιμοποιήθηκαν για τη δημιουργία του χάρτη. Πρωτίστως γίνεται μία σύντομη αναφορά για τις συντεταγμένες του συστήματος και ακολούθως περιγράφονται τα αρχεία, που χρησιμοποιήθηκαν από το API για την εμφάνιση του χάρτη [5].

### 3.2.1 Συντεταγμένες (γεωγραφικό μήκος και πλάτος)

Οι συντεταγμένες χρησιμοποιούνται για τον ακριβή προσδιορισμό θέσεων στο χάρτη. Υπάρχουν πολλά διαφορετικά συστήματα συντεταγμένων, όπως τα NAD83, ETRS89, EPSG:4326 και EGM96. Το σύστημα συντεταγμένων που χρησιμοποιείται από το Google Maps είναι το World Geodetic System 84 (WGS 84), το οποίο είναι το ίδιο με αυτό που χρησιμοποιείται από το Global Positioning System (GPS). Οι συντεταγμένες, εκφράζονται με τη χρήση του γεωγραφικού μήκους και πλάτους. Το γεωγραφικό πλάτος μετριέται από το Νότο προς το Βορρά, ενώ το γεωγραφικό μήκος μετριέται από τη Δύση προς την Ανατολή. Στον Ισημερινό το γεωγραφικό πλάτος είναι 0. Αυτό σημαίνει ότι οποιοδήποτε σημείο κάτω από τον Ισημερινό (νότιο ημισφαίριο) χαρακτηρίζεται από αρνητικό γεωγραφικό πλάτος, ενώ οποιοδήποτε σημείο άνωθεν του Ισημερινού (βόρειο ημισφαίριο), έχει θετικό γεωγραφικό πλάτος. Ομοίως, υπάρχει μία μηδενική νοητή γραμμή για το γεωγραφικό μήκος, ο μεσημβρινός, που διέρχεται από το Αστεροσκοπείο του Greenwich. Κάθε σημείο, που βρίσκεται ανατολικά του συγκεκριμένου μεσημβρινού, χαρακτηρίζεται από θετικό γεωγραφικό μήκος, ενώ τα σημεία που βρίσκονται δυτικά αυτού, χαρακτηρίζονται από αρνητικό γεωγραφικό μήκος. Στο σχήμα 3.1 απεικονίζονται οι προαναφερόμενες συμβάσεις.



Σχήμα 3.1: Το κέντρο του κόσμου με γεωγραφικό πλάτος και μήκος 0, κείται λίγο έξω από τις δυτικές ακτές της Αφρικής.

### 3.2.2 Δημιουργία του χάρτη

Όπως έχει ήδη αναφερθεί, οι χάρτες της υπηρεσίας Google Maps απεικονίζονται σε ιστοσελίδες. Ο πυρήνας για κάθε ιστοσελίδα είναι ένα HTML αρχείο. Έτσι λοιπόν, αρχικά πρέπει να δημιουργηθεί μία HTML σελίδα και έπειτα να εισαχθεί ο χάρτης. Η έκδοση που χρησιμοποιήθηκε για τη δημιουργία της παρούσας ιστοσελίδας ήταν η HTML5. Ακολούθως αναφέρονται κάποια σημαντικά στοιχεία σχετικά με την δημιουργία της HTML σελίδας και στη συνέχεια περιγράφεται λεπτομερώς η σύνθεση του χάρτη.

#### HTML σελίδα

Το πρώτο tag που χρησιμοποιήθηκε ήταν το `<html>` tag, το οποίο γνωστοποιεί στον browser την αρχή και το τέλος του HTML εγγράφου. Επιπλέον αποτελεί τη ρίζα του εγγράφου και όλα τα υπόλοιπα tags περικλείονται εντός αυτού. Ακολουθεί το `<head>` tag, στο οποίο περιλαμβάνονται τα ακόλουθα : ο τίτλος της ιστοσελίδας, που εισάγεται με το `<title>` tag και το `<meta>` element, όπου δηλώνεται το

είδος κωδικοποίησης χαρακτήρων (χρησιμοποιήθηκε το UTF-8 , αφού η συγκεκριμένη κωδικοποίηση περιλαμβάνει ειδικούς χαρακτήρες για όλες τις γλώσσες) . Το συγκεκριμένο στοιχείο είναι απαραίτητο, έτσι ώστε το έγγραφο να είναι έγκυρο. Τέλος, το <body> tag περιλαμβάνει τα στοιχεία, που είναι ορατά στην ιστοσελίδα (ο χάρτης, η φόρμα συμπλήρωσης δεδομένων, οι γραφικές παραστάσεις και τα vertical menus). Επιπλέον, εντός του προαναφερόμενου στοιχείου υπάρχει το <div> tag, όπου ορίζεται ένα ID για τον χάρτη (id="map\_canvas") και η εμφάνιση αυτού στην ιστοσελίδα. Δηλαδή, ο χάρτης να έχει πλάτος 70% της ιστοσελίδας και ύψος 100% αυτής.

Αφού δημιουργηθεί η HTML σελίδα, πρέπει να "φορτώσουμε" το Google Maps API. Όπως έχει αναφερθεί και παραπάνω, το API είναι ένα Javascript αρχείο που φιλοξενείται στους servers της Google. "Φορτώνεται" με ένα <script> tag εντός του <head> tag . Το <script> στοιχείο έχει δύο ορίσματα που πρέπει να προσδιοριστούν : Το πρώτο όρισμα είναι το type και το δεύτερο είναι το src. Στο πρώτο όρισμα ανατίθεται το "text/javascript" και στο δεύτερο ανατίθεται το URL που υποδεικνύει την τοποθεσία του API :

```
<script type="text/javascript"
src="http://maps.google.com/maps/api/js?libraries=geometry&sensor=false">
</script>
```

'Όπως μπορούμε να δούμε και παραπάνω, μετά το URL ακολουθεί η ανάθεση των μεταβλητών libraries και sensor. Με την ανάθεση της πρώτης μεταβλητής ως "geometry", γίνεται χρήση ορισμένων συναρτήσεων από τη συγκεκριμένη βιβλιοθήκη. Η δεύτερη μεταβλητή χρησιμοποιείται για να γνωστοποιήσει αν η συσκευή που μεταχειρίζεται τους χάρτες διαθέτει GPS . Η τελευταία επιλογή χρησιμοποιείται για στατιστικούς λόγους από την Google προς τους χρήστες της και δεν αφορά στις παροχές εντοπισμού θέσης που προσφέρει η υπηρεσία (geolocation).

### Σύνθεση του χάρτη

Για τη σύνθεση του χάρτη και την εμφάνιση αυτού στην ιστοσελίδα, χρησιμοποιήθηκε η κλάση Map του API. Ο constructor της συγκεκριμένης κλάσης δέχεται πρωτίστως ως όρισμα το ID που έχει ανατεθεί στο χάρτη εντός του <div> element , και

ακολούθως κάποιες ιδιότητες (MapOptions) που χαρακτηρίζουν την προαναφερόμενη κλάση. Οι ιδιότητες αυτές περιγράφονται ακολούθως:

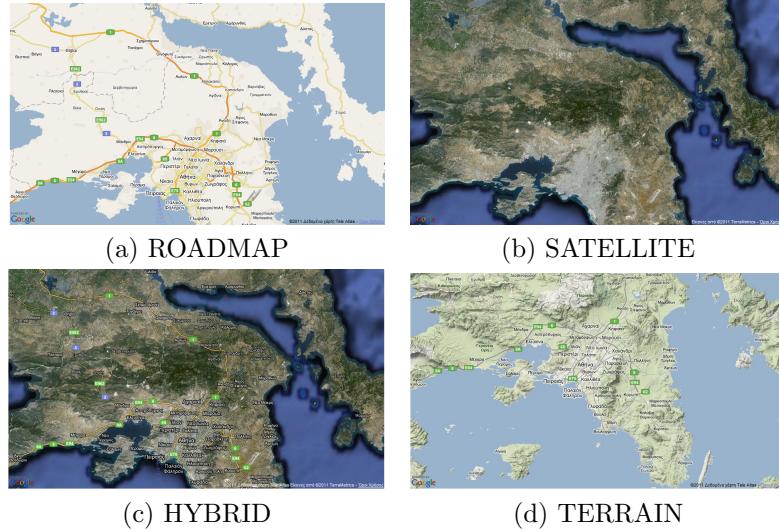
**center:** Εδώ ανατίθεται το σημείο που επιθυμούμε να εστιάσει ο χάρτης. Για την ανάθεση του συγκεκριμένου σημείου, χρησιμοποιείται η κλάση *LatLng*, η οποία επιστρέφει ένα αντικείμενο τύπου *LatLng*. Ο constructor της κλάσης αυτής, δεχεται ως ορίσματα το γεωγραφικό πλάτος και μήκος του σημείου εστίασης, όπως επίσης και μία προαιρετική boolean μεταβλητή (*noWrap*). Αν η προηγούμενη μεταβλητή είναι "false", τότε το γεωγραφικό πλάτος πρέπει να βρίσκεται εντός των  $\pm 90$  μοιρών και το γεωγραφικό μήκος εντός των  $\pm 180$  μοιρών. Διαφορετικά, τα νούμερα που χαρακτηρίζουν τα προηγούμενα μεγέθη χρησιμοποιούνται ως έχουν. Εδώ, το σημείο εστίασης ήταν η Παρνηθα, η οποία χαρακτηρίζεται από γεωγραφικό πλάτος και μήκος 38.16375412214895, 23.74975999148561 [8] αντίστοιχα.

**disableDoubleClickZoom:** Προκαθορισμένα, κάνοντας διπλό κλικ σε οποιοδήποτε σημείο του χάρτη αυξάνεται το επίπεδο εστίασής του. Για την αποφυγή της συγκεκριμένης συμπεριφοράς, η συγκεκριμένη μεταβλητή ορίστηκε ως true.

**zoom:** Η συγκεκριμένη ιδιότητα είναι υποχρεωτική, και καθορίζει το αρχικό επίπεδο εστίασης του χάρτη.

**mapTypeId:** Με την συγκεκριμένη ιδιότητα μπορούμε να επιλέξουμε το είδος του χάρτη με το οποίο θέλουμε να πλοηγηθούμε. Δίνονται οι εξής επιλογές:

- Οδικός (ROADMAP): Αυτός ο τύπος χάρτη, είναι ένας κανονικός, οδικός χάρτης (σχήμα 3.2(a))
- Δορυφορικός (SATELLITE): Στο συγκεκριμένο τύπο-χάρτη, παρουσιάζονται δορυφορικές εικόνες (σχήμα 3.2(b))
- Υβριδικός (HYBRID): Εδώ απεικονίζονται διαυγή στρώματα απ'την πλειοψηφία των δρόμων από δορυφορικές εικόνες (σχήμα 3.2(c))



Σχήμα 3.2: Οι διαθέσιμοι τύποι εμφάνισης του χάρτη.

- Έδαφικός (TERRAIN): Με τη συγκεκριμένη επιλογή, απεικονίζονται χάρτες με φυσικά χαρακτηριστικά όπως το έδαφος και η βλάστηση (σχήμα 3.2(d)).

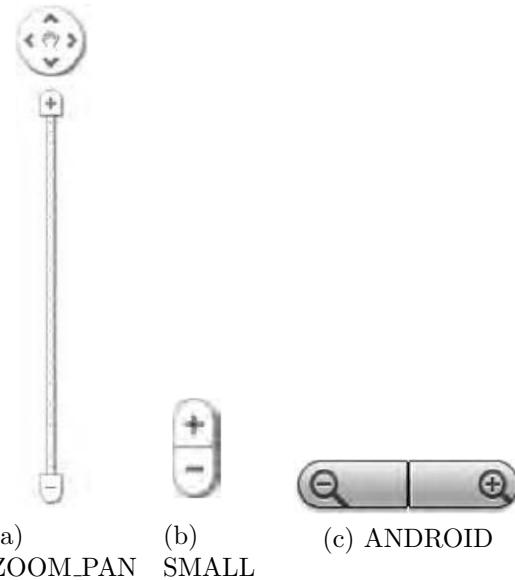
**navigationControl:** Αυτή η ιδιότητα, η οποία δέχεται boolean όρισμα, αφορά στην εμφάνιση (ή όχι) ενός εργαλείου πλοήγησης (navigation control) στο χάρτη (σχήμα 3.3). Διαθέτει δύο ιδιότητες:

- position
- style

**position:** Η συγκεκριμένη ιδιότητα, καθορίζει την τοποθεσία του εργαλείου πάνω στο χάρτη.

**style:** Εδώ δίνονται τέσσερεις επιλογές, οι τρεις εκ των οποίων παρουσιάζονται στο σχήμα 3.3. Οι επιλογές, είναι οι εξής:

- DEFAULT : Με τη συγκεκριμένη επιλογή, τοποθετείται στο χάρτη ο τύπος που αρμόζει στο μέγεθος αυτού (SMALL/ZOOM\_PAN).



Σχήμα 3.3: Οι διαθέσιμοι τύποι εμφάνισης του navigation control.

- SMALL: Αυτό είναι το μικρότερο navigation control και δίνει μόνο την επιλογή εστίασης στο χάρτη (σχήμα 3.3(b)).
- ANDROID: Αυτός ο τύπος, ειδικά, εξυπηρετεί τα κινητά τηλέφωνα με λογισμικό "Android". Διαφέρει απ'τις άλλες δύο επιλογές, όχι μόνο στην εμφάνιση, αλλά και στην προκαθορισμένη του θέση, η οποία είναι η bottom\_center (σχήμα 3.3(c)).
- ZOOM\_PAN: Ο συγκεκριμένος τύπος εμφάνισης είναι ο μεγαλύτερος και προσφέρει, εκτός απ'την δυνατότητα εστίασης, και τη δυνατότητα μετακίνησης στο χάρτη στις τέσσερις κατευθύνσεις (σχήμα 3.3(a)).

Συγκεντρώνοντας όλα τα παραπάνω χαρακτηριστικά δημιουργήθηκε ο χάρτης του σχήματος 3.4.

## Events

Όπως έχει ήδη αναφερθεί και σε προηγούμενο εδάφιο, κάθε κλάση περιλαμβάνει κάποια συμβάντα (Events). Τα events της κλάσης Map που χρησιμοποιήθηκαν στην παρούσα εφαρμογή, ήταν τα εξής:



Σχήμα 3.4: Ο δορυφορικός χάρτης που δημιουργήθηκε, με επίκεντρο την Πάρνηθα και επίπεδο εστίασης ίσο με 10.

**rightclick:** Το συγκεκριμένο γεγονός ενεργοποιείται με δεξί κλικ πάνω στο χάρτη. Τότε εμφανίζεται ένα dropdown menu όπως απεικονίζεται στο σχήμα 3.5. Αυτό το event είναι ένα "Mouse Event". Σύμφωνα με το API, τα γεγονότα που σχετίζονται με τον παραπάνω τύπο (click, rightclick, dblclick, mousemove, mouseout, mouseover) διαθέτουν την ιδιότητα "latLng". Με χρήση της ιδιότητας αυτής, επιστρέφεται το γεωγραφικό πλάτος και μήκος που έλαβε χώρα το γεγονός. Αφού πάρω τις γεωγραφικές συντεταγμένες του σημείου, τις περνάω ως όρισμα στη συνάρτηση *fromLatLngToContainerPixel*. Η συνάρτηση αυτή ανήκει στις ιδιότητες του αντικειμένου MapCanvasProject της κλάσης Map. Επιπλέον, μετατρέπει τις διοθείσες συντεταγμένες σε pixels (x=latitude, y=longitude). Έτσι λοιπόν, το μενού προβάλλεται στο σημείο που απέχει απ' την κορυφή της ιστοσελίδας κατά x, και αριστερά αυτής κατά y.

**click:** Σε περίπτωση που πραγματοποιηθεί ένα κλικ πάνω στο χάρτη, τότε εξαφανίζονται οποιαδήποτε παράθυρα είναι ανοιχτά πάνω σε αυτόν (το μενού του χάρτη ή του αισθητήρα, ο σαρωτής και κάποια ενημερωτικά μηνύματα).



Σχήμα 3.5: Το μενού που εμφανίζεται πάνω στο χάρτη κάνοντας με το ποντίκι διπλό click πάνω σε αυτόν.

**dbclick:** Όπως έχει αναφερθεί και παραπάνω, στο χρήστη δίνεται η δυνατότητα να τοποθετήσει έναν στρογγυλό σαρωτή για να ενημερωθεί για την εναπομείνουσα και καταναλισκόμενη ενέργεια ενός συγκεκριμένου πεδίου. Το συγκεκριμένο γεγονός ενεργοποιείται, μόνο, αν το επιλέξει ο χρήστης απ' το μενού του χάρτη. Με το πρώτο διπλό κλικ, που πραγματοποιείται στο χάρτη, καθορίζεται το κέντρο του σαρωτή, ενώ με το δεύτερο προσδιορίζεται ένα σημείο της περιμέτρου του κύκλου (ουσιαστικά έχει καθοριστεί η ακτίνα αυτού.)

### 3.3 Markers

Η πιο κοινή χρήση των χαρτών στο Διαδίκτυο είναι η ακριβής απεικόνιση γεωγραφικών θέσεων πάνω σε αυτούς. Το Google Maps API παρέχει την κλάση "Marker", με την οποία δημιουργούνται σημάδια (markers) πάνω στο χάρτη. Ένας marker είναι ουσιαστικά μία μικρή εικόνα, που τοποθετείται σε συγκεκριμένη θέση.

Για την απεικόνιση του δικτύου αισθητήρων στην Πάρνηθα, χρησιμοποιήθηκαν οι markers για να "ενσαρκώσουν" τους αισθητήρες του δικτύου. Σε αυτό το κεφάλαιο, θα παρουσιαστούν τα χαρακτηριστικά, που χρησιμοποιήθηκαν από την κλάση Marker για την επίτευξη του προηγούμενου σκοπού. Ο constructor της προηγούμενης κλάσης παίρνει ως όρισμα μόνο το αντικείμενο MarkerOptions, οι ιδιότητες του οποίου αναλύονται ακολούθως.

### 3.3.1 MarkersOptions

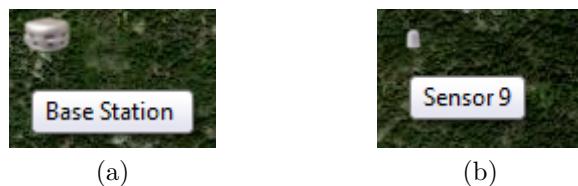
Στην παρούσα διπλωματική εργασία, χρησιμοποιήθηκαν ορισμένες απότις ιδιότητες του συγκεκριμένου αντικειμένου. Με βάση αυτές τις ιδιότητες, σχηματίστηκαν οι markers που αντιπροσωπεύουν τους αισθητήρες του δικτύου.

**position:** Στη συγκεκριμένη ιδιότητα ανατίθεται ένα όρισμα τύπου LatLng, το οποίο καθορίζει την ακριβή θέση, στην οποία θα τοποθετηθεί ο marker.

**map:** Με αυτήν την ιδιότητα προσδιορίζεται ο χάρτης στον οποίο θα τοποθετηθεί ο marker. Παίρνει ως όρισμα ένα αντικείμενο τύπου Map.

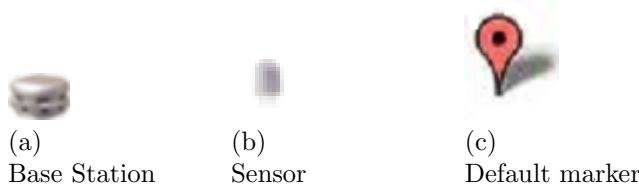
**draggable:** Όπως έχει αναφερθεί και σε άλλο εδάφιο, δίνεται η δυνατότητα στο χρήστη να μετακινήσει τους αισθητήρες του δικτύου. Αυτή η δυνατότητα επιτυγχάνεται με την ανάθεση της παραπάνω ιδιότητας ως "true". Διαφορετικά ο marker δεν μπορεί να μετακινηθεί σε άλλο σημείο του χάρτη.

**title:** Αν το ποντίκι βρίσκεται πάνω σε έναν marker, τότε μπορεί να εμφανιστεί ένα μικρό παραλληλόγραμμο, που να εμφανίζει κάποιες πληροφορίες. Το όρισμα αυτής της ιδιότητας είναι ένα string. Στην παρούσα περίπτωση, ο χρήστης μπορεί να ενημερωθεί για το id του αισθητήρα, όπως απεικονίζεται ακολούθως:



Σχήμα 3.6: Χρήση της ιδιότητας "title".

**icon:** Σε περίπτωση που δεν επιθυμούμε τη default εικόνα του marker (σχήμα 3.7 (c)), η ιδιότητα αυτή μας δίνει την επιλογή να την αλλάξουμε. Απλά, αναθέτουμε το path στο οποίο βρίσκεται αποθηκευμένη η επιθυμητή εικόνα. Για την απεικόνιση του σταθμού βάσης (base station) χρησιμοποιήθηκε η εικόνα του σχήματος 3.7 (a), ενώ η απεικόνιση του αισθητήρα γίνεται, όπως απεικονίζεται στο σχήμα 3.7 (b).



Σχήμα 3.7: Εικόνες που απεικονίζουν τους αισθητήρες.

### 3.3.2 Τοποθέτηση αισθητήρων

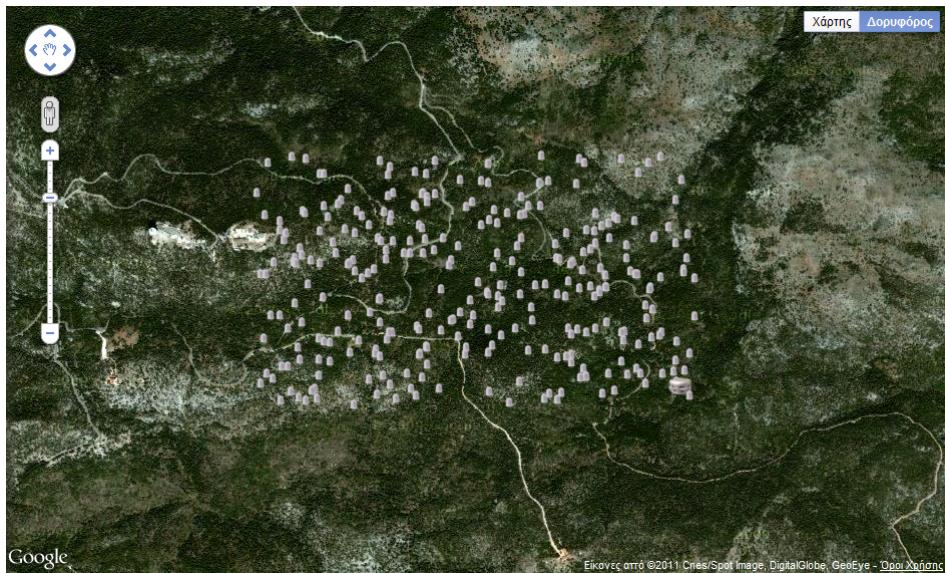
Η αρχική τοποθέτηση των αισθητήρων του δικτύου είναι τυχαία. Ο χρήστης μετά την τοποθέτησή τους, μπορεί να παρέμβει στην τοπολογία του δικτύου με τους τρόπους που έχουν αναφερθεί. Ακολούθως, περιγράφεται ο τρόπος με τον οποίο πραγματοποιείται ο προσδιορισμός των αρχικών θέσεων των αισθητήρων του δικτύου.

Αρχικά έπρεπε να προσδιοριστεί μία συγκεκριμένη περιοχή στην οποία θα τοποθετούνται οι αισθητήρες. Για τον προσδιορισμό αυτής της περιοχής, χρησιμοποιήθηκε η κλάση *"LatLangBounds"*. Ένα αντικείμενο της προαναφερόμενης κλάσης αντιπροσωπεύει ένα παραλληλόγραμμο καθορίζοντας, έτσι, μία συγκεκριμένη περιοχή. Ο constructor αυτής παίρνει δύο προαιρετικά ορίσματα, τα οποία είναι η νοτιοδυτική και η βορειοανατολική γωνία του παραλληλογράμμου. Αυτά τα ορίσματα είναι τύπου *LatLang*.

Στην παρούσα εφαρμογή, καθορίστηκαν πρωτίστως οι προαναφερόμενες γωνίες ως εξής:

```
var southWest=new google.maps.LatLng(38.166318767472,23.73321702560387);
var northEast=new google.maps.LatLng(38.174706117795,23.75199165664635);
```

Στην συνέχεια ανατίθενται οι γωνίες αυτές ως ορίσματα στον constructor της κλάση *LatLangBounds*. Έπειτα με χρήση της συνάρτησης *"fitBounds"* της κλάσης *Map*, ο χάρτης προσαρμόζεται στα δοθέντα όρια. Χρησιμοποιώντας τη συνάρτηση *random* και βάσει του γεωγραφικού πλάτους και μήκους των γωνιών, που προσδιορίστηκαν παραπάνω, παίρνω τις τυχαίες θέσεις των αισθητήρων, οι οποίες κείνται εντός του φανταστικού παραλληλόγραμμο που έχει δημιουργηθεί. Στο σχήμα 3.8 (a) απεικονίζεται ο χάρτης και οι αισθητήρες, που έχουν τοποθετηθεί εντός των ορίων, και στο σχήμα 3.8 (b) ο ίδιος χάρτης παρουσιάζεται με μεγαλύτερο επίπεδο εστίασης.



(a) Επίπεδο εστίασης: 10



(b) Επίπεδο εστίασης: 16

Σχήμα 3.8: Η τοποθέτηση των αισθητήρων εντός των ορίων που καθορίστηκαν με την αλάση LatLngBounds.

### 3.3.3 Events

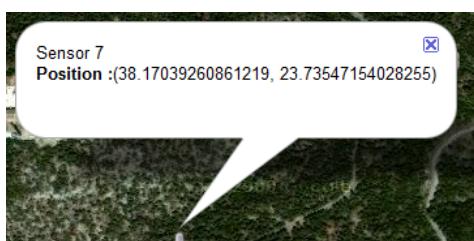
Τα events που χρησιμοποιήθηκαν από την κλάση Marker, για να προσφέρουν κάποιες υπηρεσίες στο χρήστη, όσον αφορά στους αισθητήρες, ήταν τα ακόλουθα:

**click:** Σε περίπτωση που ο χρήστης πραγματοποιήσει μονό click πάνω σε έναν αισθητήρα, τότε εμφανίζεται ένα ”σύννεφο” πάνω από αυτόν (σχήμα 3.9(a)). Το συγκεκριμένο παράθυρο μας ενημερώνει για την ακριβή τοποθεσία του αισθητήρα, όπως επίσης και για το id του. Στο εδάφιο 3.4 γίνεται αναφορά για την συγκεκριμένη υπηρεσία (InfoWindows).

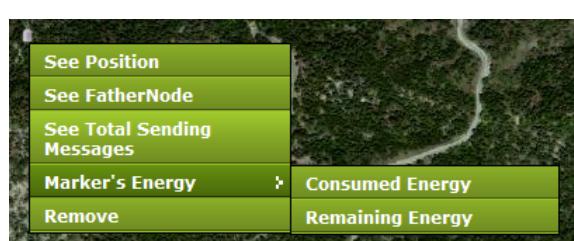
**rightclick:** Όταν πραγματοποιηθεί δεξί click πάνω σε έναν αισθητήρα εμφανίζεται το dropdown menu, όπως απεικονίζεται στο σχήμα 3.9(b).

**dragstart:** Το συγκεκριμένο γεγονός, ενεργοποιείται κατά την έναρξη της μετακίνησης του αισθητήρα. Σε αυτήν την περίπτωση αποκρύπτονται τα πάντα από το χάρτη (InfoWindows, Polylines, dropdown menus, πολύγωνα, σαρωτές).

**dragend:** Αντίστοιχα και εδώ, το συγκεκριμένο γεγονός ενεργοποιείται μόλις σταματήσει η μετακίνηση του αισθητήρα. Εδώ γίνεται ανάθεση της νέας θέσης του αισθητήρα όπως επίσης και ανανέωση στο περιεχόμενο των InfoWindows.



(a) InfoWindow



(b) Dropdown Menu

Σχήμα 3.9: Συνέπειες των γεγονότων click και rightclick αντίστοιχα.

## 3.4 InfoWindows

Αρκετά συχνά, όταν σηματοδοτούνται συγκεκριμένες θέσεις στο χάρτη, χρειάζεται να εμφανίζονται κάποιες επιπρόσθετες πληροφορίες σχετικά με αυτές. Η δυνατότητα αυτή προσφέρεται απ'την κλάση *"InfoWindow"* του Google Maps API. Χρησιμοποιώντας ένα αντικείμενο της προαναφερόμενης κλάσης, δημιουργείται ένα παράθυρο, όπως απεικονίζεται στο σχήμα 3.9(a). Το συγκεκριμένο παράθυρο τοποθετείται οπουδήποτε πάνω στο χάρτη σε μία συγκεκριμένη θέση, αλλά η πιο κοινή τοποθεσία είναι ακριβώς πάνω από ένα marker.

Ο constructor της συγκεκριμένης κλάσης δέχεται ως όρισμα μόνο τις ιδιότητες του αντικειμένου InfoWindowOptions. Οι ιδιότητες που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία, ήταν οι εξής δύο:

- **content:** Η συγκεκριμένη ιδιότητα δέχεται ως όρισμα ένα string, με το οποίο καθορίζεται το κείμενο του InfoWindow. Το κείμενο μπορεί να είναι ένα HTML στοιχείο, ένα απλό string, ή ένα string που περιλαμβάνει HTML. Το μέγεθος του παραθύρου προσαρμόζεται ανάλογα με το μέγεθος του κειμένου.
- **position:** Ως όρισμα εδώ, ανατίθεται ένα αντικείμενο τύπου LatLng. Με αυτή την ιδιότητα καθορίζεται το σημείο στο οποίο θα εμφανιστεί το παράθυρο.

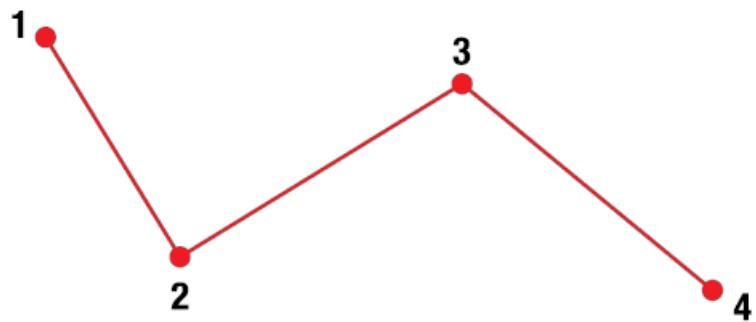
Για να εμφανιστεί το παράθυρο του κάθε αισθητήρα με μονό click πάνω σε αυτόν, χρησιμοποιείται η συνάρτηση *open* της κλάσης InfoWindow. Με αυτόν τον τρόπο ανοίγει το InfoWindow και είναι ορατό πάνω στο χάρτη. Η μέθοδος αυτή δέχεται ως πρώτο όρισμα το χάρτη, στον οποίο θα εμφανιστεί το παράθυρο (σε περίπτωση που υπάρχουν παραπάνω του ενός χάρτη στην ίδια ιστοσελίδα). Το δεύτερο όρισμα είναι προαιρετικό και δηλώνει το αντικείμενο (marker, polygon, polyline), στο οποίο αντιστοιχεί το InfoWindow. Με χρήση της συνάρτησης *close*, το InfoWindow κλείνει. Η συνάρτηση αυτή δεν δέχεται κανένα όρισμα.

## 3.5 Polylines

To Google Maps API δίνει τη δυνατότητα απεικόνισης γεωμετρικών σχημάτων πάνω στο χάρτη. Μία απ'τις κλάσεις που προσφέρει αυτήν την επιλογή είναι η *"Polyline"*.

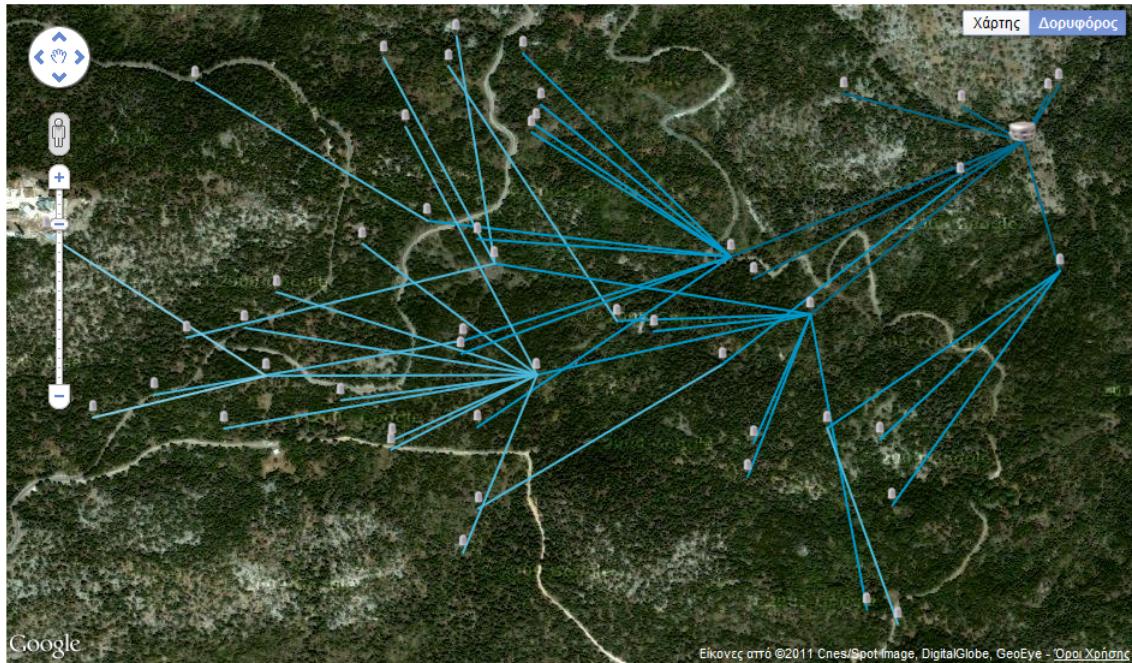
Αυτή η κλάση παρέχει τα απαραίτητα εργαλεία για επισήμανση δρόμων, συνόρων και άλλων περιοχών πάνω στο χάρτη. Επιπλέον, ιδιαίτερη χρησιμότητα παρουσιάζεται στη δεικτοδότηση οδικών κατευθύνσεων.

Οι polylines αποτελούνται από ενωμένες γραμμές. Κάθε γραμμή αποτελείται από δύο σημεία: Το σημείο εκκίνησης και το σημείο τερματισμού. Τα προαναφερόμενα σημεία είναι γεωγραφικές συντεταγμένες. Όυσιαστικά, οι polylines είναι μία συλλογή από σημεία που ενώνονται μεταξύ τους με γραμμές:



Σχήμα 3.10: Μία Polyline αποτελείται από σημεία που ενώνονται με γραμμές.

Στην παρούσα διπλωματική εργασία τα polylines χρησιμοποιήθηκαν για να έχει ο χρήστης μία οπτική επαφή του δικτύου που αναπτύσσεται (σχήμα 3.11). Με αυτόν τον τρόπο, μπορεί να διακρίνει το επίπεδο στο οποίο ανήκει ο κάθε αισθητήρας και τον πατέρα αυτού (δηλαδή σε ποιόν κόμβο στέλνει τις μετρήσεις του).



Σχήμα 3.11: Οι polylines απεικονίζουν την τοπολογία του δικτύου αισθητήρων.

Ο constructor της κλάσης που περιγράφεται σε αυτό το χωρίο, δέχεται ένα όρισμα τύπου ”PolylineOptions”. Παρακάτω, αναλύονται οι ιδιότητες που εφαρμόστηκαν, για την εμφάνιση των Polylines .

### 3.5.1 PolylineOptions

Το αντικείμενο αυτό διαθέτει διάφορες ιδιότητες, αλλά μόνο μία είναι υποχρεωτική: η ιδιότητα *path*. Βάσει των χαρακτηριστικών του προαναφερόμενου αντικειμένου καθορίζεται το μονοπάτι που θα σηματοδοτεί μία polyline, όπως επίσης και ο τρόπος απεικόνισής του (χρώμα, αμυδρότητα και πάχος γραμμής). Πιο συγκεκριμένα έγινε χρήση των ακόλουθων ιδιοτήτων:

**path:** Η ιδιότητα αυτή καθορίζει το μονοπάτι της polyline. Δέχεται ως όρισμα έναν πίνακα με αντικείμενα τύπου LatLang, που καθορίζουν τη διαδοχική σειρά των συντεταγμένων του μονοπατιού.

**map:** Εδώ, ως όρισμα περνάμε το id του χάρτη στον οποίο επιθυμούμε να απεικονιστούν οι polylines. Την ίδια λειτουργία έχει η συνάρτηση "setMap" της ίδιας κλάσης.

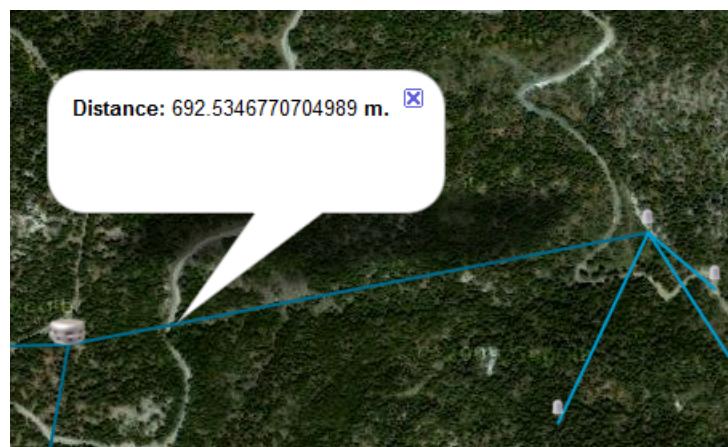
**strokeColor:** Με αυτήν την ιδιότητα καθορίζεται το χρώμα της γραμμής. Η τιμή που χρησιμοποιείται είναι ένας δεκαεξαδικός αριθμός. Το προκαθορισμένο χρώμα είναι το μαύρο (#000000). Όπως μπορεί να παρατηρηθεί στο σχήμα 3.11, οι κόμβοι διαφορετικών επιπέδων συνδέονται με τους προγόνους τους με διαφορετικό χρώμα. Δηλαδή οι κόμβοι του πρώτου επιπέδου συνδέονται με τον Base Station με γραμμές σκούρου μπλε χρώματος (#006B8F), οι κόμβοι του επιπέδου δύο συνδέονται με τους προγόνους τους με πιο ανοιχτό μπλε (#0099CC) κ.ο.κ.

**strokeOpacity:** Ο αριθμός, που τίθεται ως όρισμα σε αυτήν την ιδιότητα καθορίζει τη θολότητα της γραμμής. Παίρνει τιμές εντός του διαστήματος [0.0,1.0]. Με την τιμή "1.0" η γραμμή είναι 100% ορατή, ενώ αντιθέτως με την τιμή "0.0" η γραμμή είναι 0% ορατή. Η προκαθορισμένη τιμή για αυτήν την ιδιότητα είναι το 1.0 και αυτή η τιμή χρησιμοποιήθηκε στην απεικόνιση των γραμμών του σχήματος 3.11.

**strokeWeight:** Αυτή η ιδιότητα καθορίζει το πάχος της γραμμής σε pixels. Παίρνει ως όρισμα έναν αριθμό, που δηλώνει τα επιθυμητά pixels του πλάτους της γραμμής. Το προκαθορισμένο μέγεθος είναι 3 pixels, ενώ στην παρούσα περίπτωση οι γραμμές έχουν φάρδος 2 pixels.

### 3.5.2 Events

Όπως και οι προηγούμενες κλάσεις, έτσι και αυτή διαθέτει κάποια events. Το μόνο event που χρησιμοποιήθηκε εδώ, ήταν το "click" event. Μόλις πραγματοποιηθεί μονό click πάνω σε μία polyline εμφανίζεται ένα InfoWindow, το οποίο μας πληροφορεί για την απόσταση των αισθητήρων που ενώνει, όπως απεικονίζεται στο ακόλουθο σχήμα:



Σχήμα 3.12: To InfoWindow που εμφανίζεται με μονό click πάνω στη γραμμή.

### 3.5.3 Τοποθέτηση των Polylineς

Ο χρήστης μέσω της φόρμας συμπλήρωσης δεδομένων προσδιορίζει την απόσταση που θα έχουν οι κόμβοι πατέρας - παιδί (broadcast). Ο υπολογισμός της απόστασης μεταξύ δύο σημείων πάνω στο χάρτη, πραγματοποιείται με χρήση της συνάρτησης *computeDistanceBetween*. Αυτή η μέθοδος δέχεται ως ορίσματα δύο αντικείμενα τύπου LatLang, που προσδιορίζουν τα σημεία μεταξύ των οποίων επιθυμούμε να υπολογιστεί η απόσταση. Η συνάρτηση αυτή ανήκει στην βιβλιοθήκη "Geometry Library" του Google Maps API και χρησιμοποιεί την *Haversine Formula*, για τον υπολογισμό της απόστασης δύο αισθητήρων. Πιο συγκεκριμένα, σύμφωνα με την Haversine Formula [10] υπολογίζεται η απόσταση δύο σημείων, τα οποία βρίσκονται σε μεγάλη σφαιρική επιφάνεια. Ισχύει ότι η απόσταση μεταξύ των δύο σημείων,

είναι το μήκος του συντομότερου μονοπατιού που τα ενώνει. Η εξίσωση που μας δίνει την απόσταση, είναι η εξής:

$$\text{haversin}\left(\frac{d}{R}\right) = \text{haversin}(\varphi_2 - \varphi_1) + \cos(\varphi_1) \cos(\varphi_2) \text{haversin}(\Delta\lambda) \quad (3.1)$$

όπου

- *haversin*: είναι η haversine συνάρτηση :

$$\text{haversin}(\theta) = \sin^2\left(\frac{\theta}{2}\right) = \frac{(1 - \cos(\theta))}{2} \quad (3.2)$$

- $d$ : είναι η απόσταση μεταξύ των δύο σημείων,
- $R$ : είναι η ακτίνα της σφαίρας. Στην περίπτωσή μας η σφαίρα στην οποία βρίσκονται τα σημεια είναι η γη, που έχει ακτίνα 6378137 μέτρα,
- $\varphi_1$ : το γεωγραφικό πλάτος του ενός σημείου,
- $\varphi_2$ : το γεωγραφικό πλάτος του δεύτερου σημείου,
- $\Delta\lambda$  : η διαφορά του γεωγραφικού μήκους των δύο σημείων.

Το όρισμα της συνάρτησης *haversin* στο αριστερό μέρος της εξίσωσης 3.1 είναι σε ακτίνια. Σε μοίρες έχουμε:  $\text{haversin}\left(\frac{180^\circ d}{\pi R}\right)$ . Λύνοντας τώρα την εξίσωση 3.1 ως προς  $d$ , έχουμε:

$$d = R \text{haversin}^{-1}(h) = 2R \arcsin(\sqrt{h}) \quad (3.3)$$

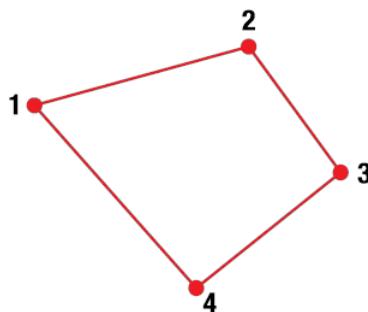
όπου  $h = \text{haversin}\left(\frac{d}{R}\right)$ .

Για την τοποθέτηση των polylines με τέτοιο τρόπο ώστε να απεικονίζουν το δίκτυο αισθητήρων, πραγματοποιήθηκε η εξής διαδικασία: Σε έναν πίνακα αποθηκεύτηκαν 20 διαφορετικοί δεκαεξαδικοί αριθμοί χρωμάτων. Στη συνέχεια, για κάθε αισθητήρα θέτω σε έναν πίνακα τις συντεταγμένες αυτού, όπως επίσης και τις συντεταγμένες του κόμβου-πατέρα του (ο κόμβος - πατέρας του κάθε αισθητήρα έχει καθοριστεί με βάση το broadcast). Για να πάρω τις συντεταγμένες των αισθητήρων, χρησιμοποιώ τη συνάρτηση "getPosition()" της κλάσης Marker. Η συνάρτηση αυτή επιστρέφει την ακριβή τοποθεσία του αισθητήρα ως ένα αντικείμενο τύπου LatLang.

Ακολούθως, θέτω στην ιδιότητα `path`, τον πίνακα με τις συντεταγμένες των δύο κόμβων. Στην ιδιότητα `color` γίνεται αναθέση χρώματος αντίστοιχο με το επίπεδο του αισθητήρα (επιλέγεται το χρώμα που βρίσκεται στην ίδια θέση του πίνακα με το επίπεδο του αισθητήρα). Τα ορίσματα των υπόλοιπων ιδιοτήτων παραμένουν ίδια για όλες τις γραμμές. Στα εδάφια 3.6.3 και 3.7.2 περιγράφονται δύο ακόμα εφαρμογές των polylines.

## 3.6 Polygons

Πολλές φορές επιθυμούμε να τονίσουμε συγκεκριμένες περιοχές πάνω στο χάρτη, όπως μία χώρα ή μία πόλη. Για την έμφαση αυτών των περιοχών χρησιμοποιούνται πολύγωνα, τα οποία σχηματίζονται με χρήση της κλάσης `Polygon`, που προσφέρει το Google Maps API. Τα πολύγωνα είναι όμοια με τις polylines. Η κύρια διαφορά έγκειται στο ότι στα πολύγωνα τα σημεία έναρξης και τερματισμού είναι πάντα ενωμένα, όπως απεικονίζεται στο σχήμα 3.13.



Σχήμα 3.13: Σε ένα πολύγωνο, το σημείο έναρξης ενώνεται με το σημείο τερματισμού.

Όπως έχει ήδη αναφερθεί, ο χρήστης έχει τη δυνατότητα να επιλέξει μία συγκεκριμένη περιοχή του δικτύου και να μελετήσει τη συμπεριφορά της. Η επιλογή της επιθυμητής περιοχής επιτυγχάνεται με τον καθορισμό πολυγώνων πάνω στον χάρτη, επισημαίνοντας έτσι το κατάλληλο πεδίο. Ο constructor της κλάσης αυτής δέχεται ένα μοναδικό όρισμα, τύπου `"PolygonOptions"`. Οι ιδιότητες του αντικειμένου αυτού αναλύονται στο επόμενο εδάφιο.

### 3.6.1 PolygonOptions

Βάσει των ιδιοτήτων, που διαθέτει το αντικείμενο PolygonOptions, καθορίζονται τα όρια του πολυγώνου, η μορφή που θα έχουν οι πλευρές του, όπως επίσης και η μορφή του εσωτερικού του.

Κάποιες από τις ιδιότητες που χρησιμοποιήθηκαν από αυτό το αντικείμενο είναι ίδιες με τις ιδιότητες του αντικειμένου PolylineOptions, που αναπτύχθηκαν στο εδάφιο 3.5.1. Έτσι λοιπόν, οι ιδιότητες *path*, *map*, *strokeColor*, *strokeOpacity*, και *strokeWeight*, χρησιμοποιήθηκαν όπως περιγράφηκε παραπάνω (η μόνη διαφορά είναι ότι το χρώμα των πλευρών του πολυγώνου είναι το πράσινο (#99FF99)). Οι διαφορετικές ιδιότητες που διαθέτει το αντικείμενο PolylineOptions, είναι οι εξής δύο:



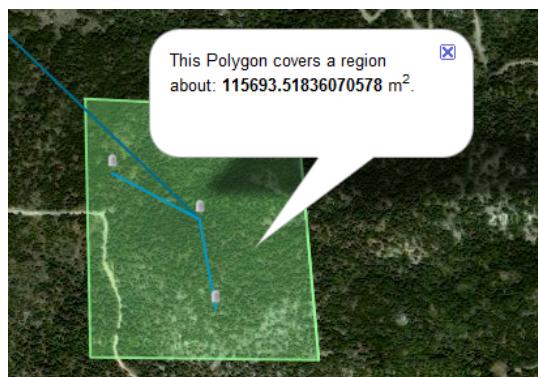
Σχήμα 3.14: Ενδεικτικό πολύγωνο εφαρμογής, χρησιμοποιώντας τις ιδιότητες του αντικειμένου PolygonOptions.

**fillColor:** Καθορίζει το χρώμα της περιοχής εντός του πολυγώνου. Το χρώμα καθορίζεται και εδώ με δεκαεξαδικό αριθμό και το προκαθορισμένο χρώμα είναι το μαύρο. Στην περίπτωσή μας, το χρώμα που επιλέχθηκε ήταν το πράσινο.

**fillOpacity:** Αυτή η ιδιότητα καθορίζει την ευκρίνεια του χρώματος στο εσωτερικό του πολυγώνου. Παίρνει τιμές εντός του διαστήματος [0.0...1.0] και η τιμή που χρησιμοποιήθηκε για τα πολύγωνα της εφαρμογής είναι η 0.35. Βάσει των παραπάνω ιδιοτήτων, τα πολύγωνα της εφαρμογής έχουν τη μορφή του σχήματος 3.14.

### 3.6.2 Events

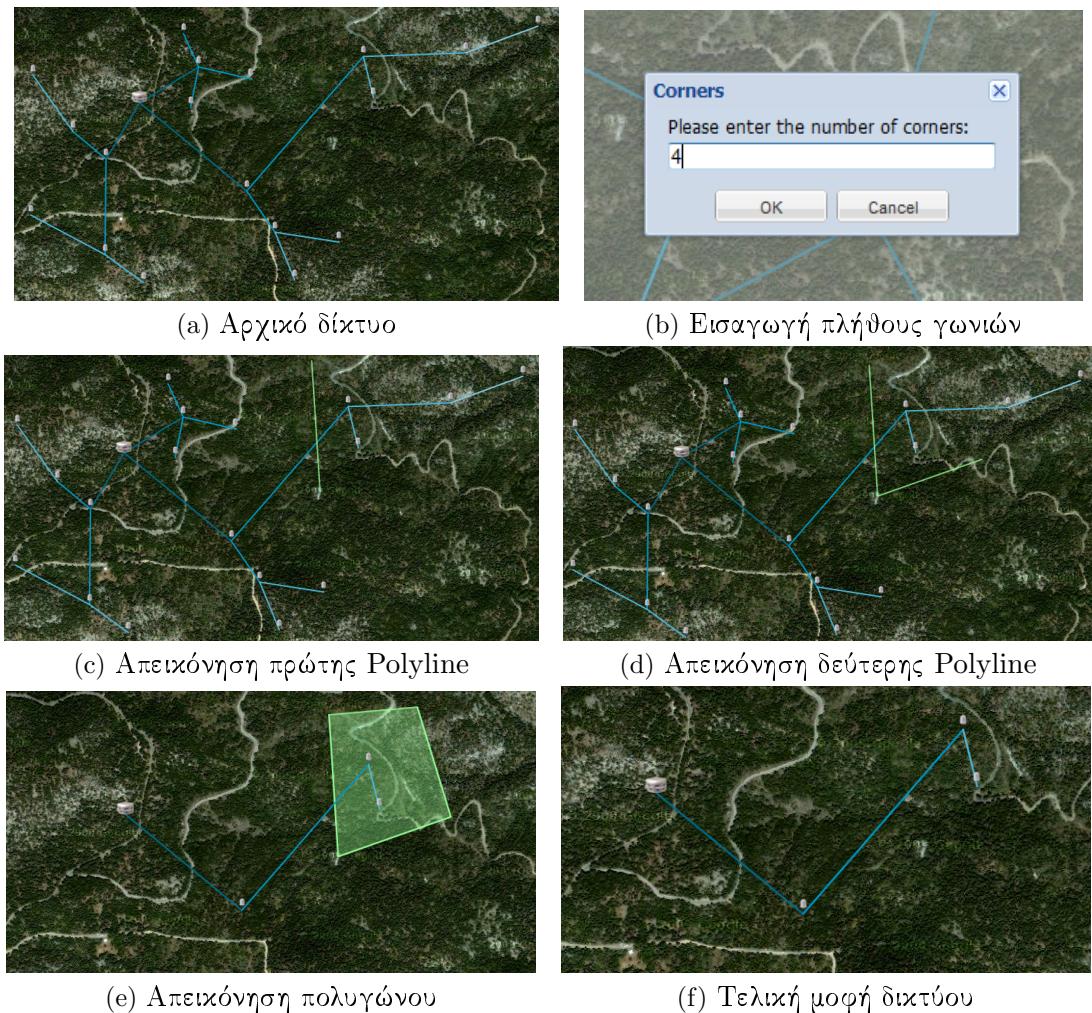
Ομοίως και εδώ με την προηγούμενη αλάση, χρησιμοποιήθηκε μόνο το event "click". Πραγματοποιώντας μονό click πάνω στο πολύγωνο, εμφανίζεται ένα InfoWindow που πληροφορεί το χρήστη για την έκταση της περιοχής που καλύπτει (σχήμα 3.15). Για τον υπολογισμό της έκτασης χρησιμοποιήθηκε η συνάρτηση "computeArea()". Η συνάρτηση αυτή ανήκει στη βιβλιοθήκη "Geometry Library". Δέχεται ως όρισμα, τον πίνακα με τις συντεταγμένες που δηλώνουν το περίγραμμα του πολυγώνου (δηλαδή τον ίδιο πίνακα που τέθηκε ως όρισμα στην ιδιότητα path).



Σχήμα 3.15: Το InfoWindow ενός πολυγώνου.

### 3.6.3 Τοποθέτηση πολυγώνου στο χάρτη

Για την τοποθέτηση ενός πολυγώνου στο χάρτη, πρωτίστως, ο χρήστης πρέπει να εισάγει το επιθυμητό πλήθος γωνιών του πολυγώνου. Το πολύγωνο πρέπει να έχει παραπάνω από δύο γωνίες. Σε αντίθετη περίπτωση, εμφανίζεται ενημερωτικό μήνυμα, που προειδοποιεί το χρήστη για τη σωστή επιλογή του πλήθους των γωνιών. Μετά τη συμπλήρωση της φόρμας, που καθορίζει το πλήθος των γωνιών, ο χρήστης πραγματοποιεί μονά clicks πάνω στο χάρτη. Με αυτόν τον τρόπο καθορίζει τα σημεία, που επιθυμεί να βρίσκονται οι γωνίες του πολυγώνου. Για τη διευκόλυνση του χρήστη, με polylines, που εμφανίζονται τη στιγμή που πραγματοποιούνται τα μονά clicks, ο χρήστης έχει μία οπτική προσωρινή εποπτεία του πολυγώνου. Στο σχήμα 3.16 περιγράφεται η διαδικασία για τον καθορισμό ενός τετραπλεύρου:



Σχήμα 3.16: Διαδοχικές φάσεις τοποθέτησης ενός πολυγώνου.

### 3.6.4 Εντοπισμός αισθητήρων εντός του πολυγώνου.

Στο συγκεκριμένο εδάφιο, όταν αναπτυχθεί η διαδικασία, με την οποία αποφασίζεται αν ένας αισθητήρας βρίσκεται εντός ή εκτός ενός πολυγώνου.

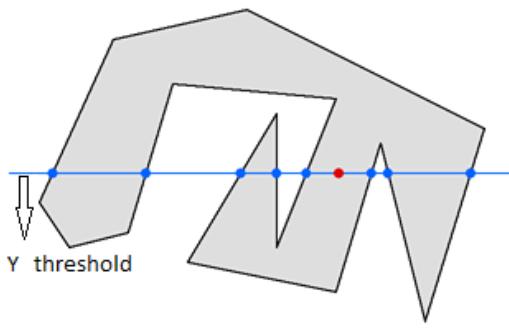
Έστω ότι έχουμε ένα πολύγωνο και ένα σημείο (όταν χαρακτηρίζεται ως test point), για το οποίο θέλουμε να προσδιορίσουμε, αν βρίσκεται εντός του πολυγώνου. Η λύση είναι να συγχρηθεί κάθε πλευρά του πολυγώνου με την ευθεία που διέρχεται κάθετα από μία ευθεία της γραμμής του test point και να συμπληρωθεί μία λίστα με κόμβους. Κάθε κόμβος δηλώνει ένα σημείο το οποίο μίας πλευράς του πολυγώνου με το Y threshold του test point (σχήμα 3.17) [9]. Στο παράδειγμα του διπλανού σχήματος οκτώ πλευρές του πολυγώνου διασταυρώνονται με το Y threshold,

ενώ οι υπόλοιπες έχουν όχι. Αν υπάρχει περιττό πλήθος κόμβων δεξιά και αριστερά του test point, τότε αυτό κείται εντός του πολυγώνου.

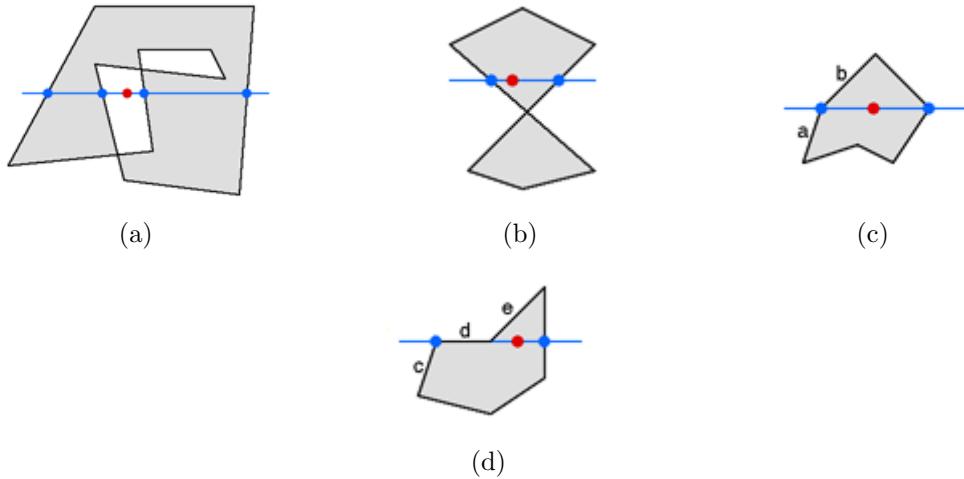
Σε αντίθετη περίπτωση, βρίσκεται εκτός του πολυγώνου. Όπως είναι φανερό στο παράδειγμα του σχήματος 3.17, δεξιά της κόκκινης τελείας (test point) υπάρχουν τρεις κόμβοι και αριστερά υπάρχουν πέντε κόμβοι. Εφόσον και απ'τις δύο πλευρές του σημείου ενδιαιφέροντος έχουμε περιττό πλήθος κόμβων, το σημείο ανήκει εντός του πολυγώνου.

Στο σχήμα 3.18(a) παρατηρούμε ένα πολύγωνο το οποίο διασταυρώνεται με τον ευαυτό του. Τα τμήματα του πολυγώνου, που επικαλύπτονται, ανατρέπουν το ένα το άλλο. Έτσι, το test point είναι εκτός του πολυγώνου, όπως ιποδηλώνει το άριθμο πλήθος κόμβων δεξιά και αριστερά αυτού.

Μία διαφορετική περίπτωση παρουσιάζεται στο σχήμα 3.18(b). Ναι μεν στο εξάπλευρο, που απεικονίζεται, τέμνονται δύο πλευρές του, αλλά αυτό δεν αποτελεί



Σχήμα 3.17: Ένα πολύγωνο με 14 πλευρές. Η κόκκινη τελεία αποτελεί το σημείο για το οποίο πρέπει να αποφασιστεί αν κείται εντός του πολυγώνου.



Σχήμα 3.18: Διάφορες μορφές πολυγώνων με σημεία εντοπισμού.

πρόβλημα στον εντοπισμό της θέσης του σημείου. Δεξιά και αριστερά υπάρχει περιττό πλήθος κόμβων και έτσι το test point βρίσκεται εντός του πολυγώνου.

Στο σχήμα 3.18(c) απεικονίζεται η περίπτωση όπου το σημείο τομής δύο πλευρών του πολυγώνου πέφτει ακριβώς πάνω στο Y threshold του test point. Εδώ, γίνεται η εξής υπόθεση: Σημείο τομής μεταξύ του Y threshold και μίας πλευράς πολυγώνου έχουμε όταν η μία άκρη της πλευράς βρίσκεται κάτω (αυθαίρετη επιλογή) απ'το Y threshold, και η άλλη άκρη βρίσκεται ακριβώς πάνω στο Y threshold ή πάνω απ'αυτό. Σύμφωνα με την προηγούμενη περίπτωση, η πλευρά **a** δημιουργεί έναν κόμβο, αφού η μία άκρη βρίσκεται κάτω απ'το όριο και η άλλη ακριβώς πάνω σε αυτό. Η πλευρά **b** δεν δημιουργεί κάποιον κόμβο, διότι η μία άκρη αυτής βρίσκεται ακριβώς πάνω στο όριο και η άλλη πάνω απ'αυτό.

Τέλος, στην εικόνα 3.18(d) απεικονίζεται η περίπτωση όπου η πλευρά ενός πολυγώνου κείται ακριβώς πάνω στο Y threshold. Ακολουθώντας τον προηγούμενο κανόνα, η πλευρά **c** δημιουργεί έναν κόμβο, διότι η μία άκρη της βρίσκεται κάτω απ'το Y threshold και η άλλη πάνω σε αυτό. Η πλευρά **d** δεν δημιουργεί κάποιον κόμβο, αφού και οι δύο άκρες τις βρίσκονται ακριβώς πάνω στο Y threshold. Επίσης, η πλευρά **e** δεν δημιουργεί κάποιον κόμβο, διότι η μία άκρη βρίσκεται πάνω στο Y threshold και η άλλη πάνω απ'αυτό.

## 3.7 Κύκλος

Άλλη μία κλάση, με την οποία δημιουργούνται γεωμετρικά σχήματα, είναι η κλάση ”Circle”. Με την προαναφερόμενη κλάση δημιουργείται και τοποθετείται ένας κύκλος πάνω στο χάρτη.

Το γεωμετρικό αυτό σχήμα, χρησιμοποιείται στην παρούσα εφαρμογή για να καθοριστεί ένας σαρωτής πάνω στο χάρτη, ο οποίος υπολογίζει την εναπομείνουσα/καταναλισκόμενη ενέργεια του πεδίου που σαρώνει. Ο constructor της κλάσης αυτής δέχεται ένα όρισμα τύπου CircleOptions. Το αντικείμενο αυτό, περιγράφεται στο επόμενο εδάφιο.

### 3.7.1 CircleOptions

Οι ιδιότητες, που διαθέτει το αντικείμενο CircleOptions, δίνουν στο χρήστη την ευχέρεια να σχηματίσει έναν κύκλο καθορίζοντας το κέντρο του, την ακτίνα του και την εμφάνισή του.

Κάποιες ιδιότητες, που παρουσιάζουν την ίδια λειτουργικότητα με τις ιδιότητες των αντικειμένων PolylineOptions και PolygonOptions, είναι οι map, strokeColor, strokeOpacity και strokeWeight. Οι διαφορετικές ιδιότητες που χρησιμοποιήθηκαν ήταν οι εξής:

**center:** Η ιδιότητα αυτή δέχεται ένα όρισμα τύπου LatLang και καθορίζει το κέντρο του κύκλου.

**radius:** Ως όρισμα, εδώ, τίθεται ένας αριθμός, ο οποίος δηλώνει την ακτίνα του κύκλου σε μέτρα.

Ένας ενδεικτικός κύκλος που σχηματίζεται με χρήση των παραπάνω ιδιοτήτων, είναι αυτός που απεικονίζεται στο σχήμα 3.19.

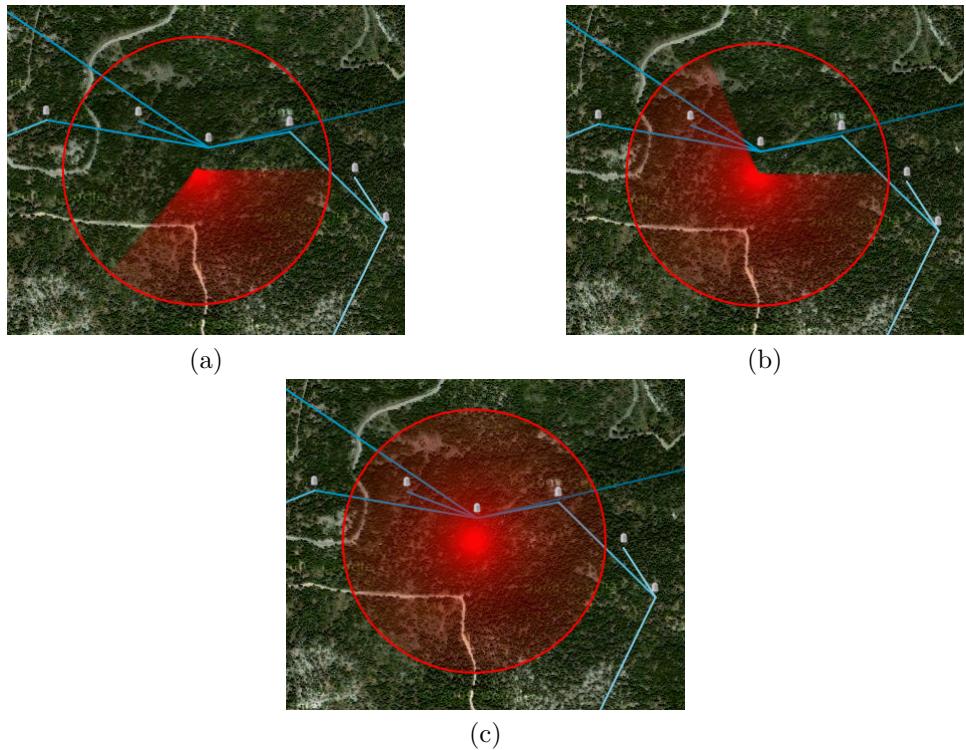


Σχήμα 3.19: Ενδεικτικός κύκλος που δημιουργείται χρησιμοποιώντας τις ιδιότητες του αντικειμένου CircleOptions.

### 3.7.2 Τοποθέτηση σαρωτή πάνω στο χάρτη

Ο χρήστης έχει την δυνατότητα να τοποθετήσει ένα σαρωτή πάνω στο χάρτη, διαλέγοντας την επιλογή ”Field’s Energy” από το dropdown menu του χάρτη. Μόλις γίνει αυτή η επιλογή, ενεργοποιείται το event dbclick του χάρτη. Ο χρήστης, τότε, καθορίζει με το πρώτο διπλό κλικ πάνω στο χάρτη, το σημείο που θα αποτελέσει το κέντρο του κύκλου και με το δεύτερο διπλό κλικ, ένα τυχαίο σημείο πάνω στην περιμέτρο του κύκλου. Ουσιαστικά, με αυτόν τον τρόπο έχει καθοριστεί η ακτίνα του κύκλου, αφού με χρήση της συνάρτησης ComputeDistanceBetween(), υπολογίζεται η απόσταση μεταξύ του κέντρου και του τυχαίου σημείου της περιμέτρου. Το αποτέλεσμα που επιστρέφει η προηγούμενη συνάρτηση τίθεται ως όρισμα στην ιδιότητα radius.

Μόλις καθοριστεί το κέντρο του κύκλου και η ακτίνα του, σχηματίζεται ο κύκλος. Η εικονική σάρωση που πραγματοποιείται, δημιουργήθηκε με τη χρήση των Polyline (σταδιακή αναπαράσταση στο σχήμα 3.20). Όσον αφορά στη σάρωση του πεδίου που καλύπτει, δημιουργούνται και τοποθετούνται polylines από το κέντρο του



Σχήμα 3.20: Σταδιακή αναπαράσταση σάρωσης.

κύκλου έως την περίμετρό του. Εδώ οι polylines έχουν χρώμα κόκκινο και ευχρίνεια 10%. Τοποθέτούνται σταδιακά πάνω στο χάρτη, και αυτό επιτυγχάνεται με χρήση της συνάρτησης "setTimeout" της JavaScript, η οποία παρεμβάλλει το χρονικό διάστημα των πέντε milliseconds στην εμφάνιση δύο διαδοχικών polylines. Οι συντεταγμένες του σημείου εκκίνησης της polyline είναι ίδιες με τις συντεταγμένες του κέντρου του κύκλου. Το σημείο τερματισμού της polyline βρίσκεται πάνω στην περίμετρο του κύκλου, και οι συντεταγμένες του κάθε σημείου υπολογίζονται βάσει των εξισώσεων 3.4 και 3.5 [11]:

$$x = x_o + \rho \cos(aRad) \quad (3.4)$$

$$y = y_o + \rho \sin(aRad) \quad (3.5)$$

όπου:

- $x_o, y_o$  είναι οι συντεταγμένες του κέντρου του κύκλου,
- $\rho$  είναι η ακτίνα του κύκλου,
- $aRad$  δίνεται απ' την εξής σχέση:  $aRad = \alpha \frac{\pi}{180}$ , όπου  $0 \leq \alpha \leq 360$ .

Τέλος, εντοπίζονται οι αισθητήρες που βρίσκονται εντός του κύκλου και υπολογίζεται η συνολική εναπομείνουσα/καταναλισκόμενη ενέργειά τους. Για τον εντοπισμό των αισθητήρων εντός του κύκλου, υπολογίζεται η απόσταση του κάθε αισθητήρα από το κέντρο του κύκλου. Αν αυτή η απόσταση είναι μικρότερη της ακτίνας του κύκλου, τότε ο αισθητήρας κείται εντός αυτού.

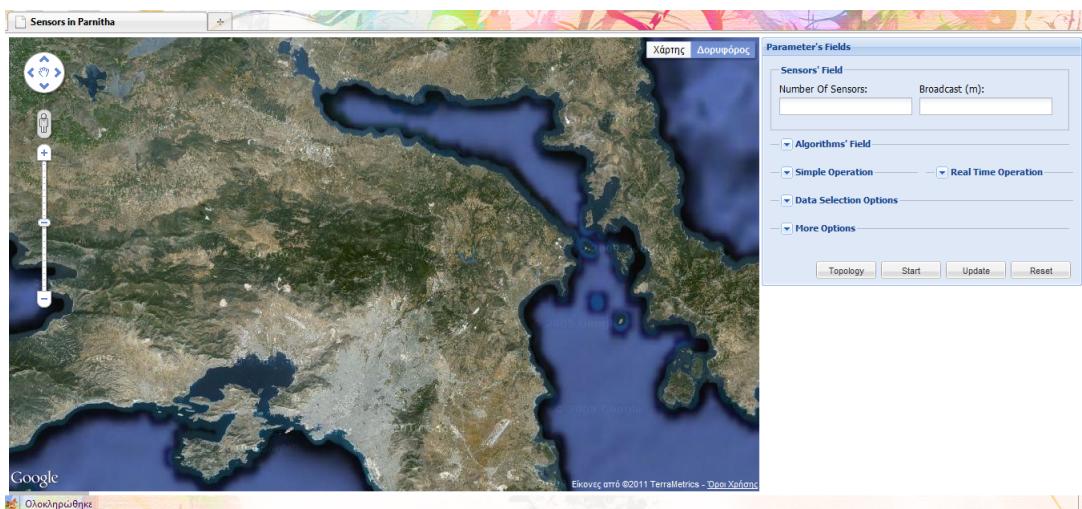
### 3.8 Σύνοψη

Στο παρόν κεφάλαιο περιγράφησαν οι υπηρεσίες του Google Maps API που χρησιμοποιήθηκαν στη γραφική διεπαφή, που αναπτύχθηκε στην παρούσα διπλωματική εργασία. Επιπλέον, αναφέρθηκαν οι κύριες ιδιότητες και συναρτήσεις που εφαρμόστηκαν από τις κλάσεις που χρησιμοποιήθηκαν. Βέβαια, το Google Maps API δεν περιορίζεται μόνο στα χαρακτηριστικά, που αναπτύχθηκαν παραπάνω. Διαθέτει μία μεγάλη συλλογή από επιπρόσθετα χαρακτηριστικά, τα οποία όμως δεν ήταν αναγκαίο να εφαρμοστούν στην παρούσα περίπτωση. Ακόμη, αναλύθηκαν και κάποιες τεχνικές, που χρησιμοποιήθηκαν όπως εντοπισμός σημείου εντός πολυγώνου ή εντός του κύκλου.

## Κεφάλαιο 4

### Παρουσίαση γραφικής διεπαφής

Στο προηγούμενο κεφάλαιο, παρουσιάστηκαν οι κλάσεις και οι ιδιότητες που χρησιμοποιήθηκαν από το Google Maps API προς τη δημιουργία του συστήματος προσομοίωσης εκτέλεσης αλγορίθμων συνάθροισης δεδομένων. Στο παρόν κεφάλαιο, θα παρουσιαστεί εξ'ολοκλήρου η λειτουργικότητα της γραφικής διεπαφής (interface) που αναπτύχθηκε.

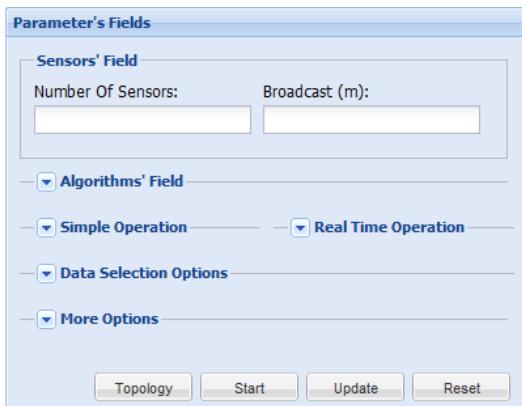


Σχήμα 4.1: Σύστημα προσομοίωσης εκτέλεσης αλγορίθμων.

Όπως απεικονίζεται στο σχήμα 4.1 το interface, που παρουσιάζεται στο χρήστη αποτελείται από το χάρτη και τη φόρμα εισαγωγής δεδομένων προς το σχηματισμό

των queries. Επιπλέον, υπάρχει ένα ελεύθερο πεδίο, στο οποίο τοποθετείται το αποτέλεσμα του query με τη μορφή πίνακα ή γραφικής παράστασης. Στις ενότητες που ακολουθούν περιγράφονται οι τομείς της φόρμας εισαγωγής δεδομένων και οι δυνατότητες του χρήστη που παρέχονται μέσω της διεπαφής.

## 4.1 Sensors' Field



Σχήμα 4.2: Το πεδίο Sensors' Field της φόρμας εισαγωγής δεδομένων.

πάρχουν οι εξής επιλογές:

Αρχικά συμπληρώνεται το πλήθος των αισθητήρων και η απόσταση (broadcast) που θα έχει ένας κόμβος απότον πατέρα του. Με βάση την προαναφερόμενη απόσταση, καθορίζεται η διασύνδεση μεταξύ των κόμβων του δικτύου, όπως επίσης και το επίπεδο στο οποίο ανήκει ο κάθε κόμβος.

Στην παρούσα φάση, ο χρήστης υπάρχει περίπτωση να κρίνει αναγκαία την επέκταση/συμίκρυνση του δικτύου ή την αύξηση/ελάττωση της πυκνότητας των αισθητήρων σε ορισμένες περιοχές. Για τις προηγούμενες περιπτώσεις, υ-

1. Μπορεί να αυξήσει/ελαττώσει το πλήθος των αισθητήρων από το αντίστοιχο πεδίο (Number Of Sensors). Στην συγκεκριμένη περίπτωση οι αισθητήρες τοποθετούνται σε νέα τυχαία τοπολογία.
2. Μπορεί να προσθέσει αισθητήρες σε συγκεκριμένες θέσεις. Όπως έχει προαναφερθεί, η συγκεκριμένη επιλογή προσφέρεται στο χρήστη με δεξιά κλικ πάνω στο χάρτη και πατώντας την επιλογή "AddSensor". Έτσι, ο αισθητήρας τοποθετείται στο συγκεκριμένο σημείο.
3. Διαγραφή αισθητήρων. Σε περίπτωση που ο χρήστης κρίνει μη αναγκαία

την ύπαρξη κάποιων αισθητήρων, μπορεί να τους αφαιρέσει από το δίκτυο. Η αφαίρεση πραγματοποιείται με δεξιά κλικ πάνω στον αισθητήρα προς διαγραφή και επιλέγεται το "RemoveSensor".

#### 4. Μετακίνηση των αισθητήρων στις επιθυμητές τοποθεσίες.

Και στις τέσσερεις παραπάνω περιπτώσεις, το δίκτυο προσαρμόζεται άμεσα στις νέες συνθήκες, μία πολύ σημαντική προϋπόθεση για την επίδοση του δικτύου, όπως αναφέρθηκε στην ενότητα 2.3.

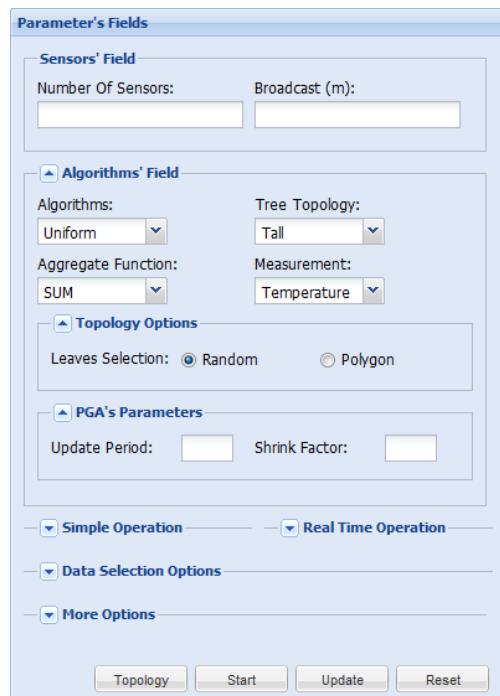
Τέλος, πρέπει να σημειωθεί ότι δεν επιτρέπεται στο χρήστη καμία άλλη ενέργεια, αν δεν έχει ορίσει το δίκτυο. Σ' αυτήν την περίπτωση, εμφανίζεται μήνυμα, το οποίο παροτρύνει το χρήστη προς τον ορισμό του δικτύου.

## 4.2 Algorithms' Field

Απ' το συγκεκριμένο πεδίο (σχήμα 4.2) ορίζονται :

- Η μέτρηση προς παρατήρηση, που μπορεί να είναι η θερμοκρασία, η υγρασία, η τάση ή ο φωτισμός.
- Αν τα δεδομένα από το περιβάλλον θα λαμβάνονται από όλους τους κόμβους ή μόνο από τους κόμβους - φύλλα. Για την πρώτη δυνατότητα από την επιλογή Tree Topology επιλέγεται το Tall, ενώ για τη δεύτερη το Leaves. Στην περίπτωση Leaves οι εσωτερικοί κόμβοι δεν δέχονται μετρήσεις, προωθούν όμως στους πατέρες τους τις συναθροιστικές τιμές που δέχονται από τα παιδιά τους.
- Ο αλγόριθμος συνάθροισης δεδομένων που θα χρησιμοποιηθεί. Διατίθενται οι ακόλουθοι αλγόριθμοι: Tag, TiNA, Uniform,  $PGA_{Root}$  και  $PGA_{Local}$ .
- Η συναθροιστική συνάρτηση της επερώτησης. Αυτή μπορεί να είναι η SUM (που ειστρέφει το άθροισμα των τιμών), η AVG (η οποία δίνει το μέσο όρο των τιμών), οι MIN και MAX (που επιστρέφουν την ελάχιστη και τη μέγιστη τιμή αντίστοιχα) και οι select \*, σύμφωνα με την οποία επιστρέφονται οι τιμές των αισθητήρων για όλες τις επιθυμητές εποχές.

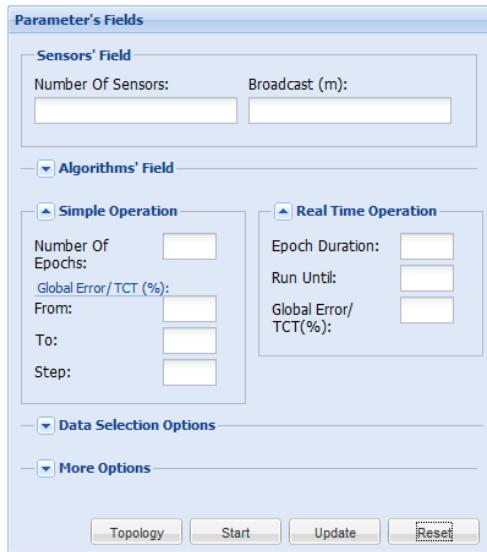
Σε περίπτωση που ο χρήστης επιλέξει ως αλγόριθμο συνάρθροισης δεδομένων είτε τον  $PGA_{Root}$  είτε τον  $PGA_{Local}$ , υποχρεούται να συμπληρώσει το πεδίο "PGA's Parameters." Εκεί ορίζεται η περίοδος ανανέωσης των φίλτρων και ο παραγοντας συμίκρυνσης αυτών. Η εφαρμογή δεν επιτρέπει στο χρήστη να "στείλει" την επερώτηση στο δίκτυο, αν δεν έχουν συμπληρωθεί οι προαναφερόμενες παράμετροι. Σε αυτήν την περίπτωση εμφανίζεται ενημερωτικό μήνυμα για τη συμπλήρωση των δεδομένων. Τέλος, στο συγκεκριμένο τμήμα, δίνεται η δυνατότητα επιλογής συγκεκριμένου πεδίου προς παρατήρηση. Πιο συγκεκριμένα, από το Topology Options επιλέγεται το Polygon. Στη συνέχεια, εισάγεται το πολύγωνο με τη διαδικασία που αναπτύχθηκε στο χωρίο 3.6.3. Αμέσως μετά την εισαγωγή του πολυγώνου, από το δίκτυο αφαιρούνται όλοι οι αισθητήρες που δεν περιλαμβάνονται εντός του πολυγώνου ή δεν είναι πρόγονοι των αισθητήρων που κείνται εντός αυτού.



Σχήμα 4.3: Το πεδίο Algorithms' Field της φόρμας εισαγωγής δεδομένων.

### 4.3 Simple/Real Operation

Το σύστημα προσομοίωσης εκτέλεσης αλγορίθμων, διαθέτει δύο λειτουργίες: την απλή και τη λειτουργία του πραγματικού χρόνου. Με βάση την απλή λειτουργία, ο χρήστης εισάγει το πλήθος των εποχών και τα δυνατά σφάλματα της εφαρμογής. Σ'αυτήν την περίπτωση, επιστρέφονται με τη μορφή γραφικής παράστασης τα συνολικά απεσταλμένα μηνύματα για τα αντίστοιχα σφάλματα. Σε περίπτωση που η συναρθροιστική συνάρτηση είναι η select \* επιστρέφεται ένας πίνακας, ο οποίος στην



**Σχήμα 4.4:** Παράμετροι δεδομένων για την απλή λειτουργία και τη λειτουργία πραγματικού χρόνου.

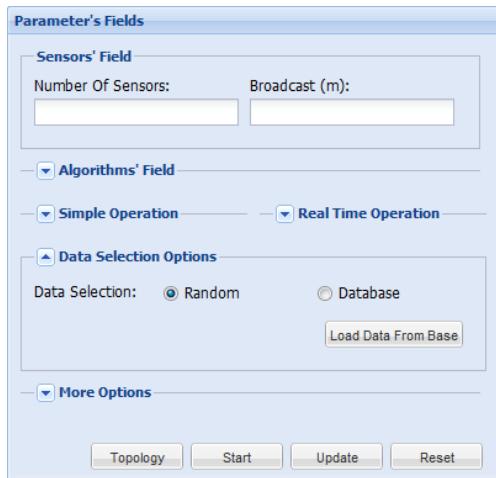
πρώτη στήλη περιλαμβάνει το σφάλμα της εφαρμογής, η δεύτερη στήλη αναφέρεται στο id του αισθητήρα, η τρίτη στήλη περιέχει το id της εποχής και στην τελευταία στήλη αναφέρεται η αντίστοιχη μέτρηση του αισθητήρα.

Σύμφωνα με τη λειτουργία του πραγματικού χρόνου, κάθε "epoch duration" δευτερόλεπτα δίνεται αποτέλεσμα συνάθροισης, μέχρι το "Run Until" δευτερόλεπτο. Επιπλέον εισάγεται μόνο ένα σφάλμα εφαρμογής σε αυτήν την λειτουργία.

## 4.4 Data Selection Options

Απ'το πεδίο αυτό, καθορίζεται η προέλευση των δεδομένων που θα χρησιμοποιηθούν για την προσομοίωση. Τα δεδομένα μπορεί να είναι τυχαία ή πραγματικά. Στην πρώτη περίπτωση δημιουργούνται τυχαίες τιμές για κάθε αισθητήρα και για κάθε εποχή εντός του διαστήματος [-2,2], ακολουθώντας τη μέθοδο του τυχαίου περιπάτου.

Στη δεύτερη περίπτωση, συλλέγονται πραγματικά δεδομένα από μία βάση δεδομένων. Οι πραγματικές αυτές μετρήσεις καταγράφηκαν απ'το Intel Berkeley Research lab.



Σχήμα 4.5: Πεδίο για την προέλευση των δεδομένων.

Σε περίπτωση που επιλεγεί η χρήση πραγματικών δεδομένων, ο χρήστης πρέπει να πατήσει το κουμπί "Load Data From Base". Μετά το πάτημα του κουμπιού, η εφαρμογή συνδέεται με τη βάση δεδομένων μέσω PHP. Ανακτούνται τα κατάλληλα δεδομένα και προωθούνται προς επεξεργασία. Εδώ, πρέπει να σημειωθεί ότι υπάρχουν δεδομένα για 50 αισθητήρες για 65535 εποχές

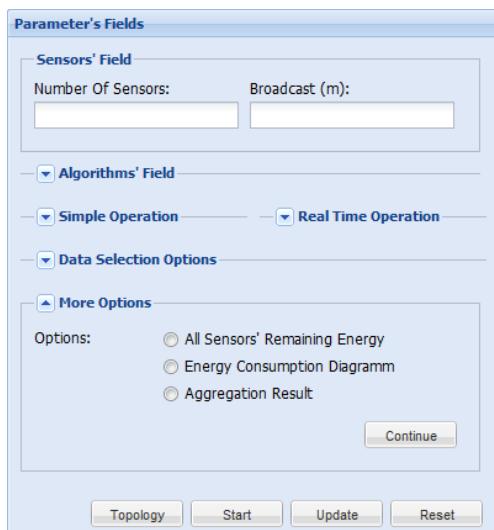
## 4.5 More Options

Απ'το συγκεκριμένο πεδίο (σχήμα 4.6), δίνονται οι εξής τρείς επιλογές:

- Εμφάνιση της εναπομείνουσας ενέργειας όλων των αισθητήρων με τη μορφή πίνακα.
- Απεικόνιση του διαγράμματος της καταναλισκόμενης ενέργειας του δικτύου.
- Εμφάνιση του αποτελέσματος της συναθροιστικής συνάρτησης.

## 4.6 Drop Down Menus

Όπως έχει αναφερθεί και στο προηγούμενο κεφάλαιο, με δεξί κλικ πάνω στο χάρτη και στον αισθητήρα εμφανίζονται τα αντίστοιχα drop down menus (σχήματα 3.7 και



Σχήμα 4.6

3.11(b) αντίστοιχα).

Όσον αφορά στο μενού του χάρτη, δίνονται οι εξής δυνατότητες:

- Τοποθέτηση ενός καινούριου αισθητήρα στο χάρτη στο σημείο που έχει πραγματοποιηθεί το event.
- Απόκρυψη/εμφάνιση των γραμμών που δείχνουν τη σύνδεση του δικτύου (Polylines).
- Τοποθέτηση ενός στρογγυλού σαρωτή (scanner), με το οποίο υπολογίζεται η εναπομείνουσα και καταναλισκόμενη ενέργεια του πεδίου, που έχει καθοριστεί.
- Ενημέρωση για την καταναλισκόμενη και εναπομείνουσα ενέργεια όλου του ασύρματου δικτύου.

Το μενού των αισθητήρων δίνει τη δυνατότητα στο χρήστη να ενημερωθεί μέσω των InfoWindows για :

- τον κόμβο-πατέρα
- τα συνολικά απεσταλμένα μηνύματα

- την εναπομείνουσα/καταναλισκόμενη ενέργεια του κάθε αισθητήρα.

Επιπλέον, με την επιλογή *Remove* ο συγκεκριμένος αισθητήρας μπορεί να αφαιρεθεί απ'το δίκτυο αισθητήρων.

## Κεφάλαιο 5

# Αλγόριθμοι Συνάρθροισης Δεδομένων

Τα ασύρματα δίκτυα αισθητήρων συντίθενται από εκατοντάδες αισθητήρες, οι οποίοι είναι ικανοί να καταγράψουν διάφορες φυσικές μετρήσεις. Επιπλέον, πρέπει να έχουν τη δυνατότητα εκτέλεσης υπολογισμών, να επικοινωνούν με τους υπόλοιπους κόμβους του δικτύου και να έχουν και την ικανότητα αυτο-οργάνωσης. Όλα τα προηγούμενα συντελούν στην συνεργατική επίτευξη έργων υψηλού επιπέδου.

Όπως έχει αναφερθεί και στο κεφάλαιο 2, η πηγή ενέργειας των αισθητήρων είναι η μπαταρία ή μονάδες συλλογής ισχύος (χυψέλες). Σε πολλές εφαρμογές οι αισθητήρες λειτουργούν ασυνόδευτοι και πρέπει να προσαρμόζονται σε προκλήσεις, που επιβάλλονται από ανεξέλεγκτες περιβαλλοντικές καταστάσεις. Συμπεραίνουμε ότι οι κόμβοι αισθητήρες είναι αυστηρώς συνδεδεμένοι με την διαθέσιμη δύναμη της πηγής ενέργειας τους, η οποία περιορίζει τη διάρκεια ζωής και την αξιοπιστία του ασύρματου δικτύου. Από τότε, λοιπόν, που οι ασύρματες επικοινωνίες καταναλώνουν σημαντικά ποσοστά ενέργειας, οι αισθητήρες πρέπει να ξοδεύουν όσο το δυνατόν λιγότερη ενέργεια. Άρα, ο έξυπνος χειρισμός της δυναμικής των δικτύων αισθητήρων είναι θεμελιώδης.

Η μακροβιότητα ενός δικτύου αισθητήρων βασίζεται σε μεγάλο βαθμό στην ύπαρξη ενεργειακά αποδοτικών αλγορίθμων για την απόκτηση, συνάρθροιση και αποθήκευση των δεδομένων που λαμβάνουν. Η επικοινωνία μέσω της μονάδας

εκπομπής - λήψης αποτελεί το βασικότερο παράγοντα κατανάλωσης ενέργειας μεταξύ των υπολοίπων, όπως είναι η αποθήκευση και η επεξεργασία.

Στις παραδοσιακές τεχνικές, τα δεδομένα όλων των αισθητήρων συγκεντρώνονται στο σταθμό βάσης και αμέσως μετά αποκτούνται από το φυσικό περιβάλλον. Από τότε, όμως, που η πηγή της ενέργειας αποτελεί σημαντικό περιορισμό, επιθυμούμε τα δεδομένα να ρέουν από τις πηγές προς τους υπόλοιπους κόμβους με όσο το δυνατόν αποδοτικότερη ενέργεια. Ο προηγούμενος στόχος επιτυγχάνεται με τη χρήση αποδοτικών αλγορίθμων συνάθροισης δεδομένων.

Η εντός του δικτύου συνάθροιση δεδομένων θεωρείται ως ο πιο αποδοτικός τρόπος προς μείωση του πλήθους των μηνυμάτων που ανταλλάσσονται εντός του δικτύου. Οι κόμβοι του δικτύου σχηματίζουν ένα δέντρο συνάθροισης δεδομένων, στο οποίο οι κόμβοι πατέρες συναθροίζουν τις τιμές που λαμβάνουν από τα παιδιά τους και προωθούν το αποτέλεσμα της συνάθροισης στους δικούς τους πατέρες. Με αυτόν τον τρόπο, ο κάθε αισθητήρας μεταφέρει μέρος των δεδομένων του και δεν προωθεί όλη την πληροφορία χωρίς επεξεργασία. Έτσι, λόγω του χαμηλού κόστους μεταφοράς μειώνεται η κατανάλωση ενέργειας των αισθητήρων και κατά συνέπεια και του δικτύου.

Στο συγκεκριμένο κεφάλαιο, θα αναλυθούν οι πέντε αλγόριθμοι συνάθροισης δεδομένων, που αναπτύχθηκαν στην παρούσα διπλωματική εργασία:

- Πρωτίστως, αναλύεται ο αλγόριθμος *TAG* (**T**iny **A**GGregation) στην ενότητα 5.3, που αποτελεί το θεμελιώδη λίθο για την ανάπτυξη των επόμενων αλγορίθμων.
- Στη συνέχεια, στο εδάφιο 5.4 αναπτύσσεται ο αλγόριθμος *TiNA* (**T**emporal coherency-aware **i**n-**N**etwork **A**ggregation).
- Ακολουθεί η ανάλυση του αλγορίθμου *Uniform* στο χωρίο 5.5.
- Τέλος, στην ενότητα 5.6 αναφέρεται ο αλγόριθμος *PGA* (**P**otential **G**ains **A**djustment), για τον οποίο υπάρχουν δύο διαφορετικές προσεγγίσεις: Οι προσεγγίσεις *AdjRoot* και *AdjLocal* που περιγράφονται στην υποενότητα 5.6.4.

Πριν την ανάλυση των παραπάνω αλγορίθμων, γίνεται μία σύντομη αναφορά στον όρο *επερωτήσεις* (*queries*) και στις κατηγορίες αυτών. Επιπλέον στο εδάφιο 5.2 περιγράφεται ποιες είναι οι συναρμολογημένες συναρτήσεις.

## 5.1 Επερωτήσεις (Queries)

Επερωτήση (query) είναι ένας τρόπος έκφρασης ερωτημάτων σε μία γραμμή για αναζήτηση πληροφορίας [12]. Τα ερωτήματα αυτά είναι γλωσσικές εκφράσεις που χρησιμοποιούνται για να γίνει ανάκτηση πληροφορίας. Η γλώσσα των επερωτήσεων που τίθενται σε δίκτυα αισθητήρων είναι τύπου SQL:

```
SELECT {agg(expr), attrs} FROM sensors
WHERE {selPreds}
GROUP BY {attrs}
HAVING {havingPreds}
EPOCH DURATION i
```

Το παραπάνω συντακτικό είναι παρόμοιο με SQL ερωτήματα. Μία σημαντική εντολή που υπάρχει στην παραπάνω μορφή είναι η "EPOCH DURATION", με την οποία καθορίζεται το χρονικό διάστημα μεταξύ διαδοχικών υπολογισμών επερωτήσης. Υπάρχουν διάφορα είδη επερωτήσεων, τα οποία αναλύονται ακολούθως:

**Ιστορικές επερωτήσεις:** Με αυτό το είδος των επερωτήσεων ζητείται μία ή συνήθως περισσότερες πρόσφατα συλλεγόμενες μετρήσεις. Κάποια παραδείγματα είναι: "Μετέδωσε όλες τις μετρήσεις της τελευταίας ώρας" ή "Μετέδωσε όλες τις μετρήσεις από την τελευταία μετάδοση". Οι ιστορικές επερωτήσεις είναι χρήσιμες για τη μείωση μεταδόσεων, τη συμπίεση των δεδομένων και σε περιπτώσεις διακοπώμενης επικοινωνίας μεταξύ δύο κόμβων.

**Τρέχουσες επερωτήσεις:** Με τις τρέχουσες επερωτήσεις ζητείται η τρέχουσα μέτρηση ενός υποσυνόλου των αισθητήρων ή ο υπολογισμός συναρμολογημένων τιμών σε αυτές, όπως είναι η μέση τιμή ή μέγιστη/ελάχιστη τιμή ανά περιοχή. Χρησιμοποιούνται για τον εντοπισμό ενδιαφέροντων γεγονότων μέσω συνάρμοισης και για γρήγορο εντοπισμό σημαντικών γεγονότων. Δύο παραδείγματα απ' αυτό το

είδος επερώτησης είναι ”Μετέδωσε τις τρέχουσες μετρήσεις όλων των κόμβων της περιοχής A” και ”Τη μέγιστη τιμή θιρύβου ανά δωμάτιο”.

**Select \* επερωτήσεις:** Εδώ, ζητείται η τρέχουσα μέτρηση ενός υποσυνόλου των αισθητήρων. Ο κόμβος, που συλλέγει τα αποτελέσματα της επερώτησης, λαμβάνει μία σειρά από ζεύγη < Αναγνωριστικό\_Αισθητήρα, Συλλεγόμενη\_Τιμή >.

**Συναθροιστικές επερωτήσεις:** Με αυτόν τον τύπο ερωτήσεων ζητούνται συναθροιστικές τιμές είτε συνολικά είτε ανά περιοχή/ομάδα αισθητήρων (εδώ αντιστοιχεί GROUP BY επερώτηση). Ανάλογα με το είδος της συναθροιστικής συνάρτησης, διαφέρει ο όγκος της μεταδιδόμενης πληροφορίας.

**Συνεχείς επερωτήσεις:** Αυτές οι ερωτήσεις τίθενται και ισχύουν είτε μέχρι ένα όριο, που καθορίζουν (για παράδειγμα μία ώρα) είτε μέχρι να αναρεθούν από άλλη επερώτηση του χρήστη.

Οι πιο συνηθισμένες επερωτήσεις σε δίκτυα αισθητήρων καθορίζουν κάθε πότε θα συλλέγονται και θα μεταδίδονται οι μετρήσεις.

Ένας σημαντικός όρος που θα αναφέρεται ακολούθως, είναι ο όρος **εποχή**. Ο όρος αυτός αναφέρεται στο χρόνο μεταξύ διαδοχικών συλλογών δεδομένων.

## 5.2 Συναθροιστικές Συναρτήσεις

Στην παρούσα διπλωματική εργασία εφαρμόστηκαν πέντε διαφορετικές συναρτήσεις συνάθροισης δεδομένων, και ήταν οι εξής: SUM, AVERAGE, MIN, MAX και SELECT\*. Οι συναθροιστικές συναρτήσεις ταξινομούνται βάσει τεσσάρων ιδιοτήτων που είναι ιδιαιτέρως σημαντικές για τα δίκτυα των αισθητήρων. Η ανάλυση των συγκεκριμένων ιδιοτήτων ακολουθεί.

Η πρώτη συνιστώσα είναι η **ευαίσθησία σε διπλοεγγραφές**. Οι συναρτήσεις που δεν είναι ευαίσθητες σε διπλοεγγραφές, όπως η MIN και MAX, δεν επηρεάζονται από διπλά δεδομένα που μπορεί να δώσει ένας αισθητήρας. Αντιθέτως, οι συναθροίσεις που είναι ευαίσθητες σε διπλοεγγραφές, όπως οι AVERAGE και SUM,

θα δώσουν άλλο αποτέλεσμα σε περίπτωση που δούμει διπλή εγγραφή από έναν αισθητήρα.

Οι πρότυπες (**exemplary**) συναρτήσεις επιστρέφουν μία ή περισσότερες αντι-προσωπευτικές τιμές από το σύνολο όλων των τιμών. Οι περιληπτικές (**summary**) συναρθροίσεις πραγματοποιούν κάποιους υπολογισμούς σε ένα σύνολο τιμών. Αναφορικά, στις exemplary συναρτήσεις εντάσσονται οι MIN, MAX και SELECT\*, ενώ στις summary εντάσσονται οι SUM και AVERAGE.

Η τρίτη ιδιότητα είναι η **μονοτονία**. Οι μονοτονικές συναρτήσεις (MIN, MAX, SUM) έχουν την εξής ιδιότητα: όταν δύο μερικώς αναφερόμενες εγγραφές  $s_1$  και  $s_2$  συνδυάζονται υπό μία συνάρτηση  $f$ , η εγγραφή που παίρνουμε ως αποτέλεσμα  $s'$  έχει μία από τις δύο ιδιότητες:

$$\forall s_1, s_2 e(s') \geq \text{MAX}(e(s_1), e(s_2)) \text{ ή}$$

$$\forall s_1, s_2 e(s') \leq \text{MIN}(e(s_1), e(s_2))$$

Αυτή η ιδιότητα είναι σημαντική όταν θέλουμε να αποφασίσουμε αν κάποια κατηγορήματα, όπως το HEAVING, μπορούν να εφαρμοστούν εντός του δικτύου, πριν η τελική τιμή προς συνάθροιση γίνει γνωστή. Με αυτόν τον τρόπο σώζονται μηνύματα, αφού μειώνεται η απόσταση μεταφοράς της μερικής εγγραφής εντός του δέντρου συνάθροισης. Οι συναρτήσεις MIN, MAX και SUM είναι μονοτονικές, σε αντίθεση με τις AVERAGE και SELECT\* που δεν είναι.

Η τελευταία ιδιότητα σχετίζεται με το **σύνολο των ορισμάτων (partial state)** που απαιτείται για κάθε εγγραφή. Για παράδειγμα, η συνάρτηση AVERAGE αποτελείται από ένα ζευγάρι τιμών ενώ η συνάρτηση SUM αποτελείται μόνο από μία τιμή.

Ο πίνακας που ακολουθεί παρουσιάζει συνοπτικά την υλοποίηση των συναρθροιστικών συναρτήσεων που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία. Υποθέτοντας ότι έχουμε δύο εγγραφές προς συνάθροιση  $R_1$  και  $R_2$  και τους μετρητές  $C_1$  και  $C_2$  έχουμε:

Συναθροιστική Συνάρτηση	Υλοποίηση
MIN	$< R > = \min\{R_1, R_2\}$
MAX	$< R > = \max\{R_1, R_2\}$
SUM	$< R > = \{R_1 + R_2\}$
AVG	$< R, C > = \{R_1 + R_2, C_1 + C_2\} (finalRecord : R \setminus C)$

Αυτό που αξίζει να σημειωθεί εδώ, είναι ότι η συνάρτηση AVG απαιτεί παραπάνω μεταδιδόμενη πληροφορία από κόμβο σε κόμβο και αυτό έχει ως συνέπεια να αυξάνεται η κατανάλωση ενέργειας του δικτύου σε σύγκριση με τις υπόλοιπες συναρτήσεις.

### 5.3 TAG (Tiny AGgregation)

Πριν το TAG [1], οι υλοποιήσεις συναθροίσεων δεδομένων πραγματοποιούνταν συγκεντρωτικά στο σταθμό βάσης. Δηλαδή οι αισθητήρες του δικτύου έστελναν όλες τις μετρήσεις τους στο σταθμό-βάση και αυτός στη συνέχεια πραγματοποιούσε τη συναθροιση των δεδομένων. Όμως, έγινε γρήγορα αντιληπτό ότι η επεξεργασία δεδομένων μέσα στο δίκτυο (in-network processing) είναι απαραίτητη για την εξοικονόμηση ενέργειας. Αυτό ήταν και το κίνητρο για τη δημιουργία του TAG.

#### 5.3.1 Κύρια ιδέα του TAG

Η κύρια ιδέα του TAG έγκειται στο ότι οι συναθροίσεις επεξεργάζονται και υπολογίζονται εντός του δικτύου, καθώς τα δεδομένα ρέουν διαμέσου των αισθητήρων. Με αυτόν τον τρόπο αποβάλλονται μη σχετικά δεδομένα και συνδυάζονται σχετικές μετρήσεις, ώστε να συμπυκνωθούν οι εγγραφές όπου είναι δυνατόν. Ο χρόνος διαιρείται σε εποχές διάρκειας i. Εντός του χρονικού αυτού διαστήματος πρέπει να παραχθεί μία συναθροιστική τιμή που συνδυάζει τις αναγνώσεις όλων των αισθητήρων του δικτύου μέχρι εκείνη την εποχή.

To TAG χωρίζεται σε δύο φάσεις:

- Στη φάση κατανομής, όπου οι συναθροιστικές επερωτήσεις διαδίδονται από το σταθμό-βάση στους αισθητήρες όλου του δέντρου (top-down διάσχιση).
- Στη φάση συλλογής, όπου οι τιμές που αφορούν στη συνάθροιση μεταδίδονται συνεχώς από τα παιδιά στους γονείς (bottom-up διάσχιση).

Πιο συγκεκριμένα, στη φάση συλλογής πρέπει να έχει εξασφαλιστεί ότι οι κόμβοι-πατέρες έχουν λάβει τις μετρήσεις των κόμβων-παιδιών τους πριν προωθήσουν μία συναθροιστική τιμή την τρέχουσα εποχή. Αυτό επιτυγχάνεται με την διαιρεση της κάθε εποχής σε επιμέρους χρονικά διαστήματα από τους κόμβους - πατέρες. Οι κόμβοι-παιδιά υποχρεούνται να μεταδώσουν τα δεδομένα τους εντός του συγκεκριμένου χρονικού διαστήματος. Επιπλέον, το χρονικό αυτό διάστημα είναι καθορισμένο έτσι ώστε να προλαβαίνει και ο κόμβος - πατέρας να προωθήσει τη δική του μέτρηση στο δίκτυο.

### 5.3.2 Μετάδοση μηνυμάτων

Στο συγκεκριμένο εδάφιο αναλύεται ο τρόπος με τον οποίο μεταδίδονται τα μηνύματα στο δίκτυο. Όταν ένας αισθητήρας ( $\rho$ ), λάβει ένα αίτημα προς συνάθροιση ( $r$ ) είτε από κάποιον άλλον αισθητήρα του δικτύου είτε από τον ίδιο το χρήστη, τότε ο  $\rho$  "ξυπνάει", συγχρονίζει το ρολόι του και ετοιμάζεται για τη συμμετοχή του στη συνάθροιση. Στο συναθροιστικό αίτημα ( $r$ ), συμπεριλαμβάνεται το διάστημα εντός του οποίου ο αποστολέας περιμένει μέχρι να λάβει τις εγγραφές από τον αισθητήρα  $\rho$ . Ο αισθητήρας  $\rho$  διαβιβάζει το αίτημα  $r$  προς τα κάτω στο δίκτυο, αναθέτοντας στα παιδιά του να του αποστείλουν τις πληροφορίες πριν το χρόνο λήξης που του έχει οριστεί. Οι κόμβοι συνεχίζουν να προωθούν το αίτημα με αυτόν τον τρόπο, έως ότου αυτό διαδοθεί σε όλο το δίκτυο.

Κατά τη διάρκεια μιας εποχής (μετά τη διάδοση του αιτήματος), κάθε αισθητήρας ακούει για μηνύματα από τα παιδιά του μέχρι τη λήξη του διαστήματος που έχει καθοριστεί από το αίτημα  $r$ . Τότε, ο αισθητήρας υπολογίζει μία μονομερή εγγραφή που αποτελείται από το συνδυασμό των τιμών που λαμβάνει από τα παιδιά του. Τέλος, κατά

τη διάρκεια του χρονικού διαστήματος που έχει καθοριστεί από τον κόμβο-πατέρα μεταφέρονται σταδιακά οι εγγραφές προς το σταθμό-βάσης του δικτύου.

Αξίζει να σημειωθεί, ότι ο χρόνος που ακούν οι κόμβοι-πατέρες είναι μεγαλύτερος από τον χρόνο που έχουν στη διάθεσή τους για τη μεταφορά των δεδομένων. Αυτό συμβαίνει για να καλύψουν περιορισμούς οι οποίοι ενδεχομένως προκύψουν από το συγχρονισμό του ρολογιού. Με αυτόν τον τρόπο, οι συναθροιστικές τιμές ρέουν προς το σταθμό βάσης διάστημα- προς -διάστημα. Τελικά, μετά από τις μονομερείς συναθροίσεις που προκύπτουν στα επιμέρους χρονικά διαστήματα μίας εποχής, μία πλήρης εγγραφή φτάνει στην ρίζα του δέντρου συνάθροισης (δηλαδή στο σταθμό-βάσης).

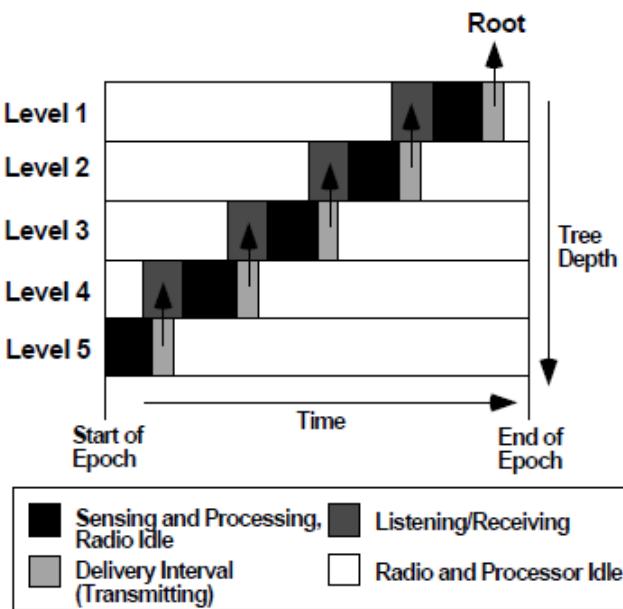
### 5.3.3 Διαμέριση της κάθε εποχής

Κατά το διαχωρισμό της κάθε εποχής σε επιμέρους διαστήματα, πρέπει να ληφθούν υπόψιν τα εξής: Κάθε χρονικό διάστημα πρέπει να είναι αρκετά μεγάλο έτσι ώστε να προλάβουν όλοι οι κόμβοι-παιδιά να αποστείλουν τις τιμές τους. Ιδιαίτερη προσοχή χρειάζεται, στο να μην ξεπεραστεί η χρονική διάρκεια της εποχής και κόμβοι που βρίσκονται σε υψηλό επίπεδο δεν έχουν προλάβει να αποστείλουν τις συναθροιστικές μετρήσεις τους. Επιπλέον, μεγάλα χρονικά διαστήματα απαιτούν τη συσκευή εκπομπής - λήψης να είναι ανοιχτή περισσότερη ώρα και έτσι καταναλώνεται πολύτιμη ενέργεια. Ακολουθήκε επομένως η εξής θεώρηση: Έστω ότι το δέντρο συνάθροισης δεδομένων του δικτύου αισθητήρων έχει μέγιστο βάθος  $d$  και ότι κάθε εποχή έχει διάρκεια epoch duration. Τότε, η κάθε εποχή θα χωριστεί σε  $d$  διαστήματα και κάθε διάστημα θα έχει χρονική διάρκεια ίση με  $(\text{epoch duration}) \setminus d$ . Οι κόμβοι που ανήκουν στο επίπεδο  $i$ , θα μεταδώσουν την μέτρησή τους στο  $i^{th}$  διάστημα.

Η παραπάνω διαδικασία, παρουσιάζεται στο σχήμα 5.1. Στο συγκεκριμένο παράδειγμα, το δέντρο συνάθροισης δεδομένων αποτελείται από πέντε επίπεδα. Η εποχή έχει χωριστεί σε πέντε επιμέρους διαστήματα και κάθε διάστημα διαιρείται εκ νέου στους εξής επιμέρους χρόνους: Στο χρόνο ακοής/λήψης, επεξεργασίας και μεταφοράς δεδομένων. Στο τμήμα της εποχής όπου οι αισθητήρες δεν πραγματοποιούν κάποιες απ'τις προηγούμενες λειτουργίες, παραμένουν αδρανείς. Αυτό είναι πολύ σημαντικό, καθ'οτι εξοικονομούνται μεγάλα ποσοστά ενέργειας.

### 5.3.4 Πλεονεκτήματα του TAG

Το βασικό πλεονέκτημα του TAG διαφαίνεται στη δραστική μείωση της επικοινωνίας που απαιτείται μεταξύ των κόμβων του δικτύου, σε σύγκριση με την τεχνική της κεντρικής συλλογής δεδομένων απ'το σταθμό-βάση. Μία σημαντική ικανότητα του TAG είναι η ανέχεια του στις απώλειες μηνυμάτων και στις ενδεχόμενες εμπλοκές του δικτύου αισθητήρων. Είναι πολύ πιθανόν στις συναθροιστικές τιμές ή στις απλές μετρήσεις των αισθητήρων να παρεμβληθούν λάθη. Επιπλέον, κάποιες συσκευές μπορεί να τεθούν εκτός δικτύου λόγω έλλειψης ενέργειας.



Σχήμα 5.1: Οι μερικώς συναθροιστικές τιμές ρέουν προς την ρίζα κατά τη διάρκεια μίας εποχής.

Η επεξεργασία των δεδομένων εντός του δικτύου στο TAG, επιτρέπει τη μετάδοση μόνο μίας τιμής από κάθε αισθητήρα ανά εποχή. Αυτό έχει ως συνέπεια να μην εξαντλείται η ενέργεια των αισθητήρων πολύ γρήγορα με την άσκοπη αποστολή μηνυμάτων.

Τέλος, με τον καταμερισμό του χρόνου σε εποχές και με το διαχωρισμό της κάθε εποχής σε επιμέρους διαστήματα, όπως παρατηρείται και στο σχήμα 5.1, το μεγαλύτερο διάστημα ο αισθητήρας παραμένει αδρανής. Αυτό καταλήγει σε περιορισμένη χρήση του επεξεργαστή και του ασύρματου και με αυτόν τον τρόπο, εξοικονομείται ενέργεια.

### 5.3.5 Σύνοψη

Ανακεφαλαιώνοντας, το TAG προσφέρει σημαντικές υπηρεσίες στα δίκτυα αισθητήρων. Όπως είδαμε και παραπάνω, πρωτίστως διαδίδεται η επερώτηση που τέθηκε στο σταθμό-βάση με top-down διάσχιση του δέντρου συναθροισης δεδομένων. Στην συνέχεια και αφού έχουν καθοριστεί τα επιμέρους χρονικά διαστήματα της κάθε

εποχής, κάθε κόμβος εντός συγκεκριμένου χρόνου που του έχει ανατεθεί από τον κόμβο-πατέρα του:

- λαμβάνει τις μετρήσεις από τα παιδιά του,
- τις επεξεργάζεται και τις συναθροίζει ανάλογα με τη συνάρτηση συνάθροισης που έχει καθοριστεί από το χρήστη
- και τέλος προωθεί το αποτέλεσμα της συνάθροισης στον κόμβο-πατέρα (bottom-up διάσχιση).

Έτσι, σε κάθε εποχή ο σταθμός - βάση, λαμβάνει μία μέτρηση που αντιστοιχεί στην επερώτηση που αρχικά είχε τεθεί στο δίκτυο.

## 5.4 TiNA (Temporal Coherency-Aware in-Network Aggregation)

Το TiNA [2] επεκτείνει τη συνάθροιση δεδομένων εντός του δικτύου χρησιμοποιώντας ένα ποσοστιαίο όριο ανοχής που εισάγει ο χρήστης. Μέσω αυτού του ορίου, όπως θα δούμε και παρακάτω, ελαχιστοποιείται το μέγεθος και το πλήθος των απεσταλμένων μηνυμάτων διατηρώντας την ποιότητα των δεδομένων σε υψηλό επίπεδο. Επιπλέον, πρέπει να αναφερθεί ότι το TiNA είναι ανεξάρτητο από βασικό πρωτόκολλο συγχρονισμού που χρησιμοποιείται από το δίκτυο για την ανταλλαγή των δεδομένων μεταξύ των κόμβων του δικτύου.

### 5.4.1 Κύρια Ιδέα

Δύο μετρήσεις που προέρχονται από δύο διαδοχικές εποχές από τον ίδιο αισθητήρα, υπό φυσιολογικές συνθήκες, διαφέρουν ελάχιστα μεταξύ τους. Τη συσχέτιση αυτή, εκμεταλλεύεται το TiNA εισάγοντας το ποσοστιαίο όριο ανοχής (temporal coherency tolerance - tct). Σύμφωνα με αυτό το όριο, μία μέτρηση από έναν αισθητήρα αποστέλλεται αν και μόνο αν η μέτρηση της τρέχουσας εποχής διαφέρει από τη μέτρηση της τελευταίας εποχής περισσότερο από tct. Το συγκεκριμένο όριο ανοχής το εισάγει ο χρήστης ή καθορίζεται από το δίκτυο σε περιπτώσεις που αυτό δεν μπορεί να στηρίξει το εισαγόμενο επίπεδο ανοχής.

To TiNA βασιζόμενο στην παραπάνω ιδεολογία εξοικονομεί ενέργεια για τους ακόλουθους δύο λόγους: Αρχικά, οι κόμβοι-φύλλα του δέντρου συνάθροισης δεδομένων, δεν αποστέλλουν νέα μέτρηση αν δεν διαφέρει απ' την παλιά περισσότερο από tct. Επίσης, ο περιορισμένος αριθμός των μηνυμάτων από τους κόμβους - φύλλα, έχει ως συνέπεια τη μείωση του μεγέθους των μηνυμάτων που αποστέλλονται οι εσωτερικοί κόμβοι του δικτύου.

Έτσι λοιπόν, με την περιορισμένη αποστολή μηνυμάτων και με το μειωμένο μέγεθος αυτών, το δίκτυο αισθητήρων μπορεί να αυξήσει τα αποθέματα ενέργειας του και να κρατήσει την ποιότητα των δεδομένων σε υψηλά επίπεδα.

#### 5.4.2 Η συνάθροιση στο TiNA

Η μορφή των επερωτήσεων που δέχεται ο σταθμός - βάση σε ένα δίκτυο αισθητηρίων όταν ακολουθείται το σχήμα TiNA, είναι η ακόλουθη:

```
SELECT {agg(expr), attrs}
FROM sensors
WHERE {selPreds}
GROUP BY {attrs}
EPOCH DURATION i
VALUES WITHIN tct
```

Εδώ απλά έχει προστεθεί η νέα εντολή "VALUES WITHIN tct" με την οποία προσδιορίζεται από το χρήστη το ποσοστό ανοχής των αλλαγών στις μετρήσεις. Το tct παίρνει τιμές από το μηδέν, που σημαίνει ότι θα προωθηθεί οποιαδήποτε μέτρηση αν συμβεί κάποια αλλαγή, εως ότου οποιοδήποτε θετικό αριθμό.

Το TiNA δημιουργήθηκε για να μπορεί να χρησιμοποιηθεί από οποιοδήποτε σύστημα πραγματοποιεί συνάθροιση δεδομένων εντός του δικτύου αισθητήρων. Όπως έχουμε ήδη προαναφέρει, με τη συνάθροιση δεδομένων εντός του δικτύου, πραγματοποιείται συγχώνευση δεδομένων από τους εσωτερικούς κόμβους του δικτύου, καθώς η πληροφορία μεταδίδεται από τους κόμβους - φύλλα στο σταθμό-βάση (δηλαδή στη ρίζα του δέντρου συνάθροισης).

Σύμφωνα με το TiNA αποφασίζεται ποιά δεδομένα θα προωθηθούν στο σταθμό-βάση και πώς θα συγχωνευθούν ή θα συναθροιστούν οι μετρήσεις εντός του δικτύου.

Με την παράμετρο  $tct$ , όπως έχει ήδη αναφερθεί και παραπάνω, καθορίζεται η ανοχή στις αλλαγές των μετρήσεων που λαμβάνουν οι αισθητήρες. Για παράδειγμα, αν το  $tct = 10\%$ , το δίκτυο αισθητήρων θα εκθέσει τις τρέχουσες μετρήσεις μόνο αν διαφέρουν απ'τις τελευταίες απεσταλμένες τιμές περισσότερο από 10%.

### 5.4.3 Η συμπεριφορά των αισθητήρων στο TiNA

Ένα δέντρο συνάθροισης δεδομένων αποτελείται απ'τους κόμβους - φύλλα και απ'τους εσωτερικούς κόμβους. Οι κόμβοι-φύλλα βρίσκονται στο τελευταίο επίπεδο του δέντρου και δεν έχουν απογόνους. Τους εσωτερικούς κόμβους αποτελούν όλοι οι υπόλοιποι κόμβοι του δέντρου συνάθροισης. Κάθε κόμβος ο οποίος εφαρμόζει τη λογική του TiNA, πρέπει να κρατάει πρόσθετη πληροφορία που θα αξιοποιήσει όταν έρθει η στιγμή της απόφασης για προώθηση (ή όχι) μίας μέτρησης.

Οι κόμβοι-φύλλα αποθηκεύουν μόνο την τελευταία απεσταλμένη μέτρηση προς τον κόμβο-πατέρα τους ( $V_{old}$ ). Όταν έρθουν νέα δεδομένα ( $V_{new}$ ), τότε οι δύο μετρήσεις ( $V_{old}$  και  $V_{new}$ ) θα συγχριθούν μεταξύ τους και η  $V_{new}$  θα προωθηθεί αν επαληθευτεί η ακόλουθη σχέση:

$$\frac{|V_{new} - V_{old}|}{V_{old}} \geq tct \quad (5.1)$$

Σε περίπτωση που δεν επαληθευτεί η σχέση 5.1 τότε η μέτρηση  $V_{new}$  απορρίπτεται και δεν μεταδίδεται.

Οι εσωτερικοί κόμβοι, κρατάνε ως πληροφορία εκτός απ'την τελευταία απεσταλμένη μέτρηση και τα τελευταία δεδομένα που έλαβαν από κάθε παιδί τους. Τα δεδομένα που έλαβαν απ'τα παιδιά τους μπορεί να είναι είτε απλές μετρήσεις που προέρχονται απ'τους κόμβους - φύλλα, είτε συναθροίσεις μετρήσεων από εσωτερικούς κόμβους. Τα δεδομένα που φτάνουν σε έναν εσωτερικό κόμβο απ'τους κόμβους παιδιά του συνδυάζονται, και υπολογίζεται ένα μονομερές αποτέλεσμα το οποίο προωθείται στους κόμβους που βρίσκονται πιο κοντά στο σταθμό-βάση. Αν κάποιος κόμβος, δεν λάβει πλήρη δεδομένα από κάποιο παιδί του, τότε χρησιμοποιεί τα τελευταία δεδομένα που του είχαν αποσταλεί απ'το συγκεκριμένο κόμβο-παιδί.

Επίσης, οι εσωτερικοί κόμβοι θα πρέπει να προωθήσουν και τη δική τους μέτρηση. Αν η μέτρηση αυτή μπορεί να συναθροιστεί σε ένα σύνολο με μονομερείς εγγραφές,

πραγματοποιείται αυτή η συνάθροιση ανεξάρτητα από το όριο tct. Αυτό συμβαίνει διότι η μονομερής εγγραφή που θα προστεθεί στο σύνολο θα αλλάξει μόνο την τιμή του μερικού αποτελέσματος και όχι το μέγεθος αυτού. Όμως, σε περίπτωση που η νέα μέτρηση του εσωτερικού αισθητήρα καταλήγει στη δημιουργία νέου συνόλου δεδομένων, τότε πρέπει να ελεγχθεί αν αυτή είναι μεγαλύτερη από tct. Σε περίπτωση που ισχύει κάτι τέτοιο, το νέο σύνολο δημιουργείται, διαφορετικά όχι.

#### 5.4.4 Επικείμενα προβλήματα

Δύο θέματα που προκύπτουν εφαρμόζοντας το TiNA, είναι τα ακόλουθα:

- Πώς ο κόμβος - γονέας θα διαχρίνει, αν μία μέτρηση δεν ικανοποιεί το WHERE clause ή δεν ανήκει εντός του εύρους του tct. Το συγκεκριμένο θέμα επιλύεται ως εξής: Ο κόμβος - παιδί στέλνει ένα σύντομο μήνυμα ακύρωσης στον κόμβο-πατέρα του σε περίπτωση που η τιμή που ανήγνευσε δεν ικανοποιεί το WHERE clause. Με αυτόν τον τρόπο, ο γονέας δεν θα χρησιμοποιήσει την αποθηκευμένη ανάγνωση που αντιστοιχεί σε αυτό το παιδί.
- Πώς ο κόμβος - γονέας θα ξεχωρίσει αν ένας κόμβος - παιδί του, δεν έστειλε μέτρηση λόγω αποτυχίας του κόμβου (εξαιτίας κάποιας βλάβης) ή δεν πραγματοποιήθηκε αποστολή λόγω tct. Η επίλυση του προβλήματος επέρχεται με την αποστολή ενός μηνύματος heartbeat από τον κόμβο-παιδί στον κόμβο-πατέρα. Το συγκεκριμένο μήνυμα αποστέλλεται περιοδικά, για να γίνεται αντιληπτή η ύπαρξή του κόμβου στο δίκτυο.

#### 5.4.5 Σύνοψη

Συνοψίζοντας, όταν σε ένα δίκτυο αισθητήρων εφαρμοστεί ο αλγόριθμος συνάθροισης δεδομένων, TiNA, τότε τα επερωτήματα ανατίθενται στο σταθμό-βάση με τη μορφή που ειδαμε στην ενότητα 5.4.2. Μετά τη διάδοση της επερώτησης στο δέντρο συνάθροισης δεδομένων, συλλέγονται τα δεδομένα ξεκινώντας από τους κόμβους - φύλλα. Οι κόμβοι-φύλλα θα μεταδώσουν τη μέτρησή τους, αν επαληθεύεται η

**σχέση 5.1.** Στη συνέχεια, οι εσωτερικοί κόμβοι, αφού έχουν λάβει όλες τις μετρήσεις απ'τους κόμβους - παιδιά τους συναθροίζουν και τη δική τους μέτρηση και αποφασίζουν πάλι με τη σχέση 5.1 για την προώθηση (ή όχι) της συναθροιστικής τιμής.

Με αυτόν τον τρόπο μειώνονται τα μηνύματα προς αποστολή. Αυτό έχει ως αποτέλεσμα την εξοικονόμηση ενέργειας απ'τους κόμβους, και κατ'επέκταση και του δικτύου, και τη διατήρηση της καλής ποιότητας των δεδομένων.

## 5.5 Uniform

Ο συγκεκριμένος αλγόριθμος συνάθροισης δεδομένων έχει την εξής ιδιαιτερότητα: Εφαρμόζει φίλτρα στους κόμβους του δέντρου συνάθροισης δεδομένων και με αυτόν τον τρόπο περιορίζεται η μεταφορά των ασήμαντων μετρήσεων στο δίκτυο. Εδώ, χρησιμοποιείται ο αλγόριθμος του TAG για το συγχρονισμό της αποστολής και της λήψης δεδομένων. Ακολουθεί η περιγραφή κάποιων όρων (ενεργοί και παθητικοί), που θα χρησιμοποιηθούν αρκετά στη συνέχεια. Μετέπειτα, περιγράφεται η λειτουργία του αλγορίθμου [3].

### 5.5.1 Ενεργοί/Παθητικοί κόμβοι

Στο συγκεκριμένο εδάφιο περιγράφεται η λειτουργία των ενεργών και παθητικών κόμβων του δέντρου συνάθροισης δεδομένων. Οι κόμβοι διαχρίνονται σε **ενεργούς** και **παθητικούς** και ακολούθως αναλύεται η λειτουργικότητά τους:

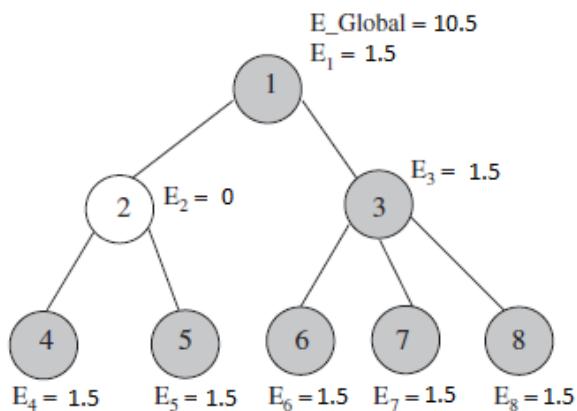
- **Ενεργός κόμβος - φύλλο:** Οι κόμβοι αυτού του είδους λαμβάνουν μετρήσεις απ'το περιβάλλον και τις προωθούν στον κόμβο - πατέρα τους.
- **Παθητικός εσωτερικός - κόμβος:** Εδώ, οι κόμβοι περιμένουν για μηνύματα απ'τα παιδιά τους. Αν ληφθούν ένα ή περισσότερα μηνύματα, οι κόμβοι συνδυάζουν όλες τις τιμές και αποστέλλουν μία τελική τιμή στον κόμβο-πατέρα τους.
- **Ενεργός εσωτερικός - κόμβος:** Οι εσωτερικοί και ενεργοί κόμβοι αποκτούν μέτρηση απ'το περιβάλλον και λαμβάνουν τις μετρήσεις των παιδιών τους. Στη

συνέχεια, συνδυάζονται όλες οι μετρήσεις και διαβιβάζονται στους κόμβους - γονείς.

### 5.5.2 Λειτουργία του αλγορίθμου

Όπως προαναφέρθηκε, ο αλγόριθμος εφαρμόζει φίλτρα σφάλματος στους κόμβους του δέντρου συνάθροισης. Αρχικά ο χρήστης δηλώνει το επιθυμητό σφάλμα όλης της εφαρμογής ( $E_{\text{Global}}$ ). Ακολούθως, το σφάλμα αυτό μοιράζεται ομοιόμορφα σε όλους τους ενεργούς κόμβους του δέντρου.

Στο σχήμα 5.2 απεικονίζεται ένα δέντρο συνάθροισης δεδομένων με ενεργούς και παθητικούς κόμβους, που απεικονίζονται με γκρι και άσπρο χρώμα αντίστοιχα. Το  $E_{\text{Global}}$  της εφαρμογής είναι 10.5 και κάθε ενεργός κόμβος αποκτά φίλτρο σφάλματος ( $E_i$ ) ίσο με  $E_{\text{Global}} \setminus$  Πλήθος Ενεργών Κόμβων. Στην προκειμένη περίπτωση οι ενεργοί κόμβοι του δέντρου συνάθροισης είναι 7, και το φίλτρο του κάθε ενεργού κόμβου είναι 1.5. Ο παθητικός κόμβος δεν διαθέτει φίλτρο.



Σχήμα 5.2: Δέντρο Συνάθροισης δεδομένων και κατανομή φίλτρων.

Μετά την κατανομή των φίλτρων ξεκινάει η διαδικασία συνάθροισης δεδομένων. Ξεκινώντας από το τελευταίο επίπεδο του δέντρου οι κόμβοι φύλλα αποστέλλουν τις μετρήσεις τους στους γονείς τους, αν και μόνο αν η διαφορά απ' την τελευταία απεσταλμένη μέτρηση τους είναι μεγαλύτερη από  $E_i$ . Ομοίως για τους εσωτερικούς - ενεργούς κόμβους, θα αποστείλουν το συναθροιστικό αποτέλεσμα μόνο στην

περίπτωση που διαφέρει από την προηγούμενη απεσταλμένη τιμή παραπάνω από  $E_i$ . Το ίδιο ισχύει και για τους παθητικούς - εσωτερικούς κόμβους.

Όπως γίνεται φανερό, οι κόμβοι του δέντρου συνάθροισης πρέπει να αποθηκεύουν ως πληροφορία την τελευταία μέτρηση που προώθησαν στο δίκτυο.

### 5.5.3 Σύνοψη

Η εφαρμογή των φίλτρων στους κόμβους ουσιαστικά δεν επιτρέπει να μεταδοθούν τιμές, οι οποίες δεν έχουν σημαντική απόκλιση από τις προηγούμενες τους. Η απόκλιση ορίζεται με το σφάλμα που ανατίθεται στους κόμβους. Με αυτόν τον τρόπο επιτυγχάνεται σημαντική μείωση στο πλήθος των μηνυμάτων που ανταλλάσσονται εντός του δικτύου.

## 5.6 PGA (Potential Gains Adjustment)

Ο αλγόριθμος που θα αναπτυχθεί στο συγκεκριμένο χωρίο βασίζεται στην εφαρμογή ανομοιόμορφων φίλτρων στους κόμβους του δέντρου συνάθροισης δεδομένων [3]. Κίνητρο για τον συγκεκριμένο αλγόριθμο ήταν η ύπαρξη ασταθών κόμβων στα δίκτυα αισθητήρων.

Η διαφορά μεταξύ αυτού του αλγορίθμου και του Uniform έγκειται στο ότι, εδώ, δεν χρησιμοποιούνται ομοιόμορφα φίλτρα. Το ποσοστό λάθους στην παρούσα περίπτωση περιοδικά μεταβάλλεται, για να προσαρμόζεται σε αλλαγές στα χαρακτηριστικά των δεδομένων.

### 5.6.1 Κύρια Ιδέα

Ο συγκεκριμένος αλγόριθμος, στηρίχθηκε στις εξής δύο παραδοχές:

- Ο κόμβος - γονέας μπορεί να αποκλείσει τα μηνύματα των παιδιών του στο δέντρο συνάθροισης δεδομένων, όταν η αθροιστική μεταβολή στις μετρήσεις είναι μικρή.
- Εδώ, γίνεται χρήση απλών και τοπικών στατιστικών δεδομένων, ώστε να εκτιμηθεί το ενδεχόμενο κέρδος από την κατανομή επιπρόσθετου σφάλματος σε

αισθητήρες του δικτύου.

Ο αλγόριθμος αυτός, λειτουργεί χρησιμοποιώντας μόνο τοπικές γνώσεις. Αυτό έχει ως συνέπεια, να μην μεταφέρονται πολλές πληροφορίες προς τη ρίζα του δέντρου και να εξοικονομείται ενέργεια απ'τους κόμβους του δικτύου. Αξίζει να σημειωθεί, ότι για την επικοινωνία μεταξύ των κόμβων του δικτύου χρησιμοποιήθηκε, και εδώ, ο αλγόριθμος TAG.

### 5.6.2 Λειτουργία Αλγορίθμου

Ο συγκεκριμένος αλγόριθμος διαιρείται σε τέσσερεις φάσεις: Στη φάση αρχικοποίησης, προσαρμογής των φίλτρων, συνάθροισης δεδομένων και μεταφοράς μηνύματος. Οι φάσεις αυτές αναλύονται ακολούθως:

#### Φάση Αρχικοποίησης:

Αρχικά, ανατίθενται σε όλους τους κόμβους ομοιόμορφα φίλτρα. Δηλαδή το συνολικό σφάλμα της εφαρμογής ( $E_{\text{Global}}$ ), που ορίζεται απ'το χρήστη, μοιράζεται σε όλους τους κόμβους ως εξής:  $E_i = E_{\text{Global}} \setminus$  Πλήθος των κόμβων του δικτύου. Το αρχικό πλάτος των φίλτρων είναι σημαντικό μόνο για τα αρχικά στάδια του αλγορίθμου, αφού στη συνέχεια, ο αλγόριθμος προσαρμόζει τα φίλτρα σύμφωνα με την δυναμική του κάθε κόμβου.

Πρέπει να σημειωθεί ότι ο όρος  $E_i$  αναφέρεται στο επιτρεπτό σφάλμα του κάθε κόμβου  $N_i$  και ο όρος  $E_{\text{Subi}}$  προσδιορίζει το μέγιστο επιτρεπτό σφάλμα του υποδέντρου του κόμβου  $N_i$ . Για τον υπολογισμό του  $E_{\text{Subi}}$ , απλά προστίθενται τα σφάλματα  $E_i$  όλων των απογόνων του κόμβου  $N_i$  και το σφάλμα  $E_i$  αυτού.

#### Φάση Προσαρμογής φίλτρων:

Η συγκεκριμένη φάση εκτελείται κάθε  $\text{Upd}$  εποχές. Το πρώτο βήμα για όλους τους κόμβους είναι να ελαττώσουν το πλάτος των φίλτρων τους κατά έναν παράγοντα σμίκρυνσης ( $\text{shrinkFactor}$ ). Οι παράμετροι  $\text{Upd}$  και  $\text{shrinkFactor}$  καθορίζονται απ'το χρήστη και η τελευταία παράμετρος παίρνει τιμές εντός του διαστήματος  $[0,1]$ .

Μετά τη συγχρυνση των φίλτρων η ρίζα του δέντρου θα έχει ένα πλεόνασμα λάθους, το οποίο είναι ίσο με  $E_{\text{Global}} \times (1-\text{shrinkFactor})$ . Η κατανομή αυτού του πλεονάσματος πραγματοποιείται χρησιμοποιώντας τις στατιστικές παραμέτρους του κάθε κόμβου. Λεπτομέρειες για αυτές τις παραμέτρους δίνονται στο εδάφιο 5.6.4.

#### **Φάση Συνάθροισης Δεδομένων:**

Σε κάθε εποχή, ο ενεργός - κόμβος λαμβάνει μία μέτρηση σχετικά με τα δεδομένα που παρατηρεί. Αν είναι εσωτερικός κόμβος περιμένει να λάβει εκτός απ'τη δική του μέτρηση και τις μετρήσεις των παιδιών του. Στη συνέχεια, υπολογίζεται μία συναθροιστική τιμή που εξαρτάται απ'τη συνάρτηση συνάθροισης δεδομένων.

#### **Φάση Μεταφοράς Μηνύματος:**

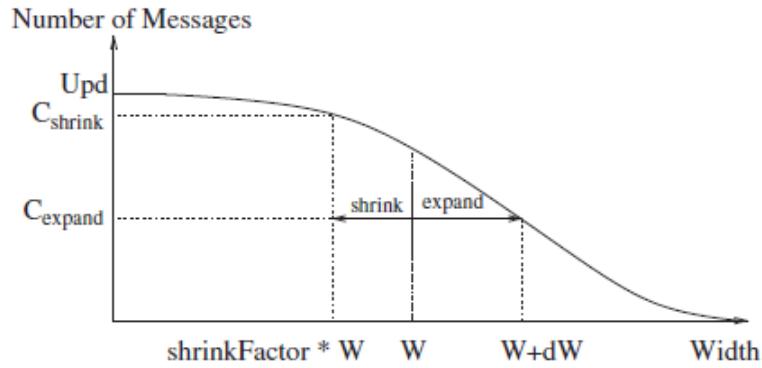
Μετά τον υπολογισμό της συναθροιστικής τιμής, πρέπει να αποφασιστεί αν αυτή η τιμή θα προωθηθεί στο υπόλοιπο δίκτυο. Μεταφορά μηνύματος απ'τον κόμβο-παιδί στον κόμβο-πατέρα πραγματοποιείται, μόνο αν η τιμή προς μετάδοση κείται εκτός του φίλτρου του κόμβου που την υπολόγισε.

### **5.6.3 Υπολογισμός στατιστικών παραμέτρων**

Όπως έχει ήδη αναφερθεί η κατανομή του πλεονάσματος λάθους στους κόμβους, γίνεται με τη χρήση κάποιων παραμέτρων που θα αναλυθούν ακολούθως. Μία απ'αυτές τις παραμέτρους είναι το ενδεχόμενο κέρδος ενός κόμβου (potential gain), αν αυξηθεί το πλάτος του φίλτρου του. Αυτό το κέρδος σχετίζεται με το πλήθος των μηνυμάτων που αναμένεται να σωθούν σε περίπτωση που αυξηθεί το σφάλμα του κόμβου.

Στο σχήμα 5.3 απεικονίζεται η αναμενόμενη συμπεριφορά ενός αισθητήρα  $N_i$  καθώς μεταβάλλεται το πλάτος του φίλτρου του  $W_i$ . Ο γ-άξονας αντιπροσωπεύει το πλήθος των μηνυμάτων που αποστέλλονται απ'τον κόμβο στον πατέρα του σε μία περίοδο από Upd εποχές. Θεωρώντας ότι οι μετρήσεις των κόμβων δεν είναι συνεχείς, αν το πλάτος του φίλτρου είναι μηδέν ( $W_i = E_i = 0$ ) αντιστοιχεί ένα μήνυμα για κάθε Upd εποχές. Αυξάνοντας το πλάτος του φίλτρου ο αριθμός των

μηνυμάτων, όπως ήταν και αναμενόμενο, μειώνεται.



Σχήμα 5.3: Ενδεχόμενο κέρδος ενός κόμβου

Στην συνέχεια, θα αναλυθούν κάποιες τοπικές στατιστικές παράμετροι ( $C_{shrink}$ ,  $C_{expand}$ ) που συλλέγονται κατά τη διάρκεια της εκτίμησης της επερώτησης. Θεωρούμε ότι φίλτρο με πλάτος  $W_i$  έχει ανατεθεί στον κόμβο  $N_i$  στην τελευταία φάση ανανέωσης των φίλτρων. Τη στιγμή που ο κόμβος χρησιμοποιεί το φίλτρο του  $W_i$ , για να αποφασίσει σχετικά με την αποστολή (ή όχι) της μέτρησής του, χρατάει κάποιες πληροφορίες υποθέτοντας τη συμίχυνση και την επέκταση του πλάτους του φίλτρου. Πιο συγκεκριμένα, ο μετρητής  $C_{shrink}$  αυξάνεται σε περίπτωση που η τιμή προς αποστολή μπορεί να μεταφερθεί με τη χρήση του φίλτρου  $W_i \times shrinkFactor$ . Αντίστοιχα, ο μετρητής  $C_{expand}$  αυξάνεται αν η τιμή προς αποστολή μεταφερόταν και με τη χρήση του φίλτρου  $W_i + dW$ . Το  $dW$  για τους κόμβους - φύλλα ορίζεται ως εξής:  $dW = E_Global \setminus N_{active}$ .<sup>1</sup> Όσον αφορά στους εσωτερικούς κόμβους έχουμε  $dW = num\_children_i^2 \times (E_Global \setminus N_{active})$ . Ακολουθεί ο υπολογισμός του potential gain, ως εξής:  $Gain_i = \delta G = C_{shrink} - C_{expand}$ .

Η τελευταία στατιστική παράμετρος, που χρησιμοποιείται στην κατανομή του πλεονάσματος λάθους που έχει η ρίζα του δέντρου μετά την τελευταία συμίχυνση των φίλτρων του κόμβου, είναι το  $CumGain_i$ . Αυτή η παράμετρος αντιπροσωπεύει το ενδεχόμενο κέρδος του υποδέντρου του κόμβου  $N_i$  και ορίζεται ως εξής:

<sup>1</sup>Το πλήθος των ενεργών κόμβων του δέντρου συνάθροισης δεδομένων.

<sup>2</sup>Το πλήθος των παιδιών του κόμβου  $N_i$ .

$$CumGain_i = \begin{cases} Gain_i & N_i: \text{κόμβος - φύλλο} \\ Gain_i + \sum_{j \in children_{N_i}} CumGain_j & \text{διαφορετικά} \end{cases}$$

#### 5.6.4 PGA\_Root και PGA\_Local

Όπως έχει αναφερθεί και παραπάνω, κάθε Upd εποχές τα φίλτρα όλων των κόμβων ελαττώνονται. Αυτό έχει ως αποτέλεσμα να υπάρχει ένα πλεόνασμα λάθους ( $E_{Additional} = E_{Global} \times (1 - shrinkFactor)$ ). Στο συγκεκριμένο εδάφιο θα αναπτυχθούν οι δύο προσεγγίσεις με τις οποίες μπορεί να κατανεμηθεί το πλεόνασμα αυτό στους κόμβους του δέντρου συνάθροισης.

Ακολουθώντας με την **PGA\_Root** προσέγγιση κάθε υποδέντρο  $j$  του κόμβου  $N_i$ , αυξάνει το  $E_{Subj}$  σύμφωνα με τη σχέση 5.2. Η κατανομή του πλεονάσματος πραγματοποιείται μόνο αν το  $E_{Additionalj}$  που υπολογίστηκε είναι μεγαλύτερο απ'την ποσότητα  $E_{Global} \setminus N_{active}$ .

$$E_{Additionalj} = \frac{E_{Additional} \times CumGain_j}{CumGain_i} \quad (5.2)$$

Σύμφωνα με την **PGA\_Local** προσέγγιση, οι κόμβοι μεταβιβάζουν το πλεόνασμα λάθους στα παιδιά τους με τρόπο level-by-level, αντί να ξεκινάει ολόκληρη η διαδικασία απ'τη ρίζα όπως στην προηγούμενη περίπτωση. Διασχίζοντας το δέντρο bottom-up κάθε εσωτερικός κόμβος  $i$  ισχυρίζεται ότι έχει ένα διαθέσιμο πλεόνασμα λάθους ίσο με  $E_{Additionali} = \sum_{j \in children_{N_i}} E_j \times (1 - shrinkFactor)$ . Αυτό το πλεόνασμα προέρχεται απ'τη συμίκρυνση των φίλτρων των απογόνων του κόμβου  $i$ . Η κατανομή αυτού του πλεονάσματος πραγματοποιείται με βάση τη σχέση 5.3

$$E_{Additionalj} = \frac{E_{Additionali} \times Gain_j}{Gain_i + \sum_{j \in children_{N_i}} Gain_j} \quad (5.3)$$

#### 5.6.5 Σύνοψη

Συνοψίζοντας, ο συγκεκριμένος αλγόριθμος προσαρμόζει σταδιακά τα φίλτρα των κόμβων του δέντρου συνάθροισης δεδομένων με βάση την λειτουργικότητά τους στο δίκτυο (αν αποστέλλουν πολλά ή λίγα μηνύματα). Αρχικά, το σφάλμα που εισάγει ο χρήστης μοιράζεται εξίσου το ίδιο σε όλους τους κόμβους του δικτύου. Κάθε

Upd εποχές τα φίλτρα όλων των κόμβων ελαττώνονται. Στην συνέχεια μοιράζεται το πλεόνασμα λάθους ακολουθώντας μία εκ των δύο μετρικών που παρουσιάστηκαν στην ενότητα 5.6.4

# Κεφάλαιο 6

## Πειράματα

Στο παρόν κεφάλαιο παρουσιάζονται τα πειράματα της τρέχουσας διπλωματικής εργασίας. Στην πρώτη ενότητα του κεφαλαίου επιδεικνύεται η υλοποίηση SQL επερωτήσεων στο γραφικό interface που υλοποιήθηκε, ενώ στο δεύτερο εδάφιο παρουσιάζεται η απόδοση των αλγορίθμων,.

### 6.1 Υλοποίηση SQL επερωτήσεων στη γραφική διεπαφή

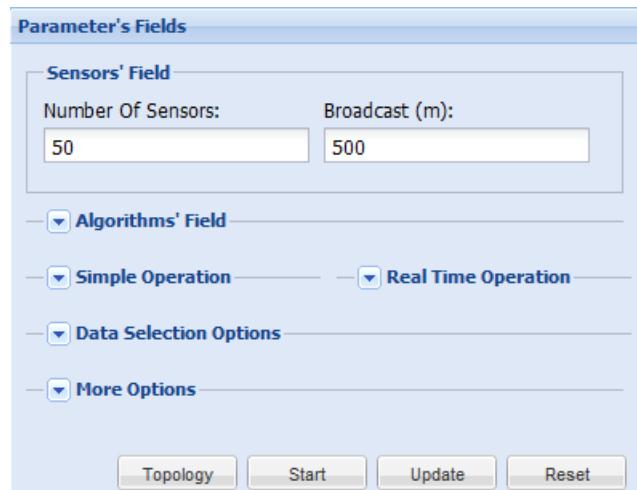
Σ' αυτήν την ενότητα όμως παρουσιαστεί ο τρόπος με τον οποίο υλοποιείται ένας ενδεικτικός αριθμός SQL επερωτήσεων στη γραφική διεπαφή που δημιουργήθηκε. Πριν την υλοποίηση των SQL ερωτημάτων, ακολουθεί μία σύντομη αναφορά για τη φόρμα εισαγωγής δεδομένων.

#### 6.1.1 Υλοποίηση SQL επερωτήσεων

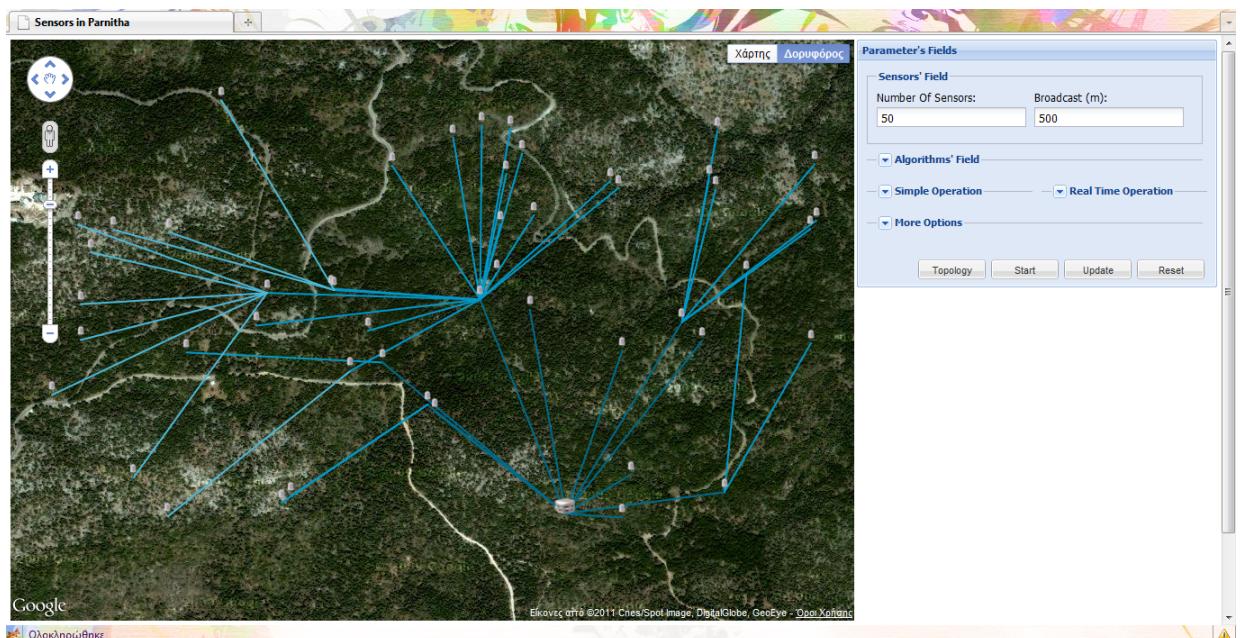
Πριν ανατεθεί η SQL επερώτηση στην εφαρμογή, πρέπει να τοποθετηθεί το δίκτυο αισθητήρων στην Πάρνηθα. Ο χρήστης πρέπει να συμπληρώσει το πεδίο "Sensors' Field" της φόρμας εισαγωγής δεδομένων. Δύο παράμετροι πρέπει να καθοριστούν εδώ: το πλήθος αισθητήρων και η μέγιστη απόσταση μεταξύ των κόμβων-πατέρων και των κόμβων-παιδιών. Αφού συμπληρωθούν οι δύο παράμετροι και ο χρήστης πατήσει το κουμπί "Topology", εμφανίζεται το δίκτυο αισθητήρων στην Πάρνηθα,

όπως φαίνεται στο σχήμα 6.1. Σε περίπτωση που ο χρήστης επιχειρήσει κάποια άλλη ενέργεια πριν την τοποθέτηση του δικτύου αισθητήρων εμφανίζεται ενημερωτικό μήνυμα, που τον προτρέπει να καθορίσει το δίκτυο αισθητήρων. Επιπλέον, ενημερωτικό μήνυμα εμφανίζεται και στην περίπτωση που ο χρήστης δεν έχει συμπληρώσει κάποια απότις προαναφερόμενες παραμέτρους.

Μετά τον ορισμό του δικτύου αισθητήρων, μπορούμε να υποβάλλουμε τα SQL ερωτήματα στην εφαρμογή μας. Ακολουθούν κάποια παραδείγματα, όπου πρωτίστως αναφέρεται το SQL ερώτημα, στη συνέχεια απεικονίζεται η υλοποίησή του στη φόρμα εισαγωγής δεδομένων και τέλος ακολουθεί η ολοκληρωμένη εικόνα των αποτελεσμάτων του query.



(a) Συμπλήρωση του πεδίου "Sensors' Field"



(b) Απεικόνιση του δικτύου αισθητήρων

Σχήμα 6.1: Τοποθέτηση δικτύου αισθητήρων στην Πάρνηθα.

**Query N<sub>o</sub>1 :**

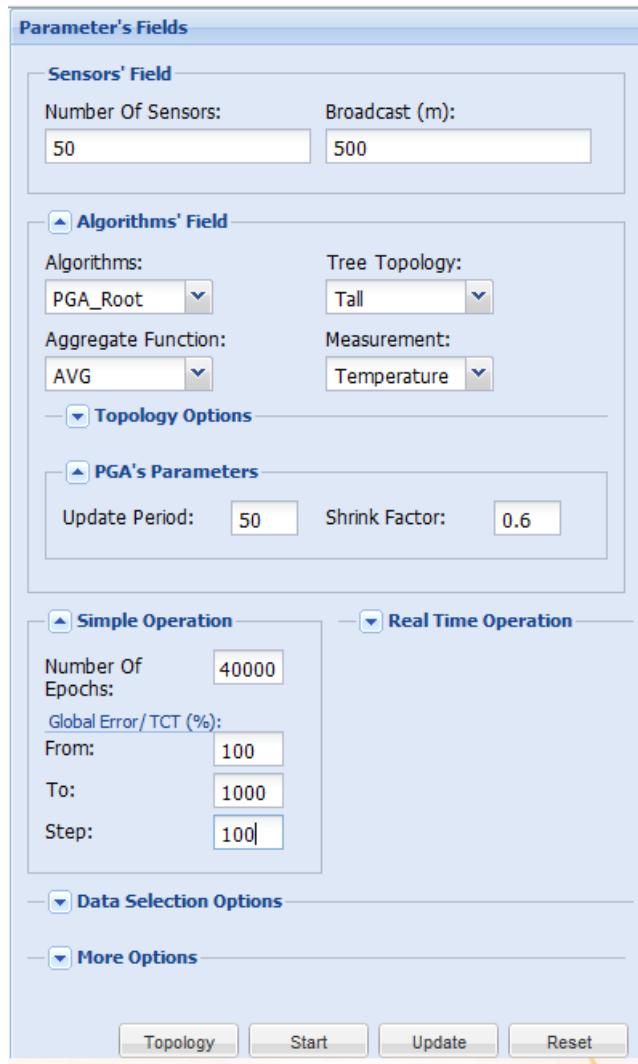
```

SELECT {AVG( Temperature )}
FROM sensors
WHERE {
    Aggregation_Function = PGA_Root AND
    Tree_Topoogy = Tall AND
    Data_Selection = random AND
    Leaves_Selection = random AND
    Upd = 50 AND
    Shrink_Factor = 0.6 AND
    Operation = Simple AND
    E_Global = 100:100:1000
}
NUMBER OF EPOCHS 40000

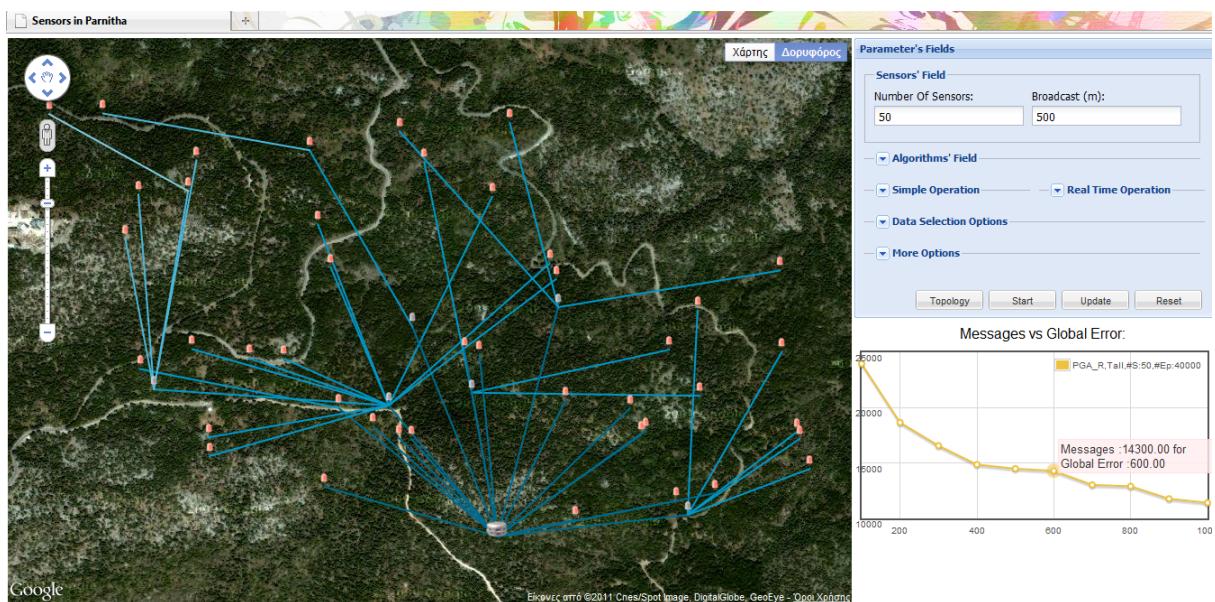
```

Με το προαναφερόμενο query, ζητάμε το μέσο όρο της θερμοκρασίας των αισθητήρων. Ως αλγόριθμος συνάρθροισης δεδομένων ορίστηκε ο PGA\_Root, η τοπολογία του δέντρου συνάρθροισης δεδομένων είναι η Tall (όλοι οι κόμβοι του δέντρου είναι ενεργοί) και τα δεδομένα τα αποκτούν οι κόμβοι από γεννήτρια τυχαίων αριθμών, οι οποίοι ακολουθούν το "random walk pattern". Επιπλέον, η επιλογή των κόμβων-φύλλων γίνεται τυχαία (και όχι με τη χρήση πολυγώνου), ανανέωση στα φίλτρα των αισθητήρων πραγματοποιείται κάθε 50 εποχές και ο παράγοντας συμχρυνσης είναι 0.6. Η λειτουργία είναι απλή και το σφάλμα που δέχεται η εφαρμογή κυμαίνεται από το 100 έως το 1000 με βήμα 100. Τέλος, για κάθε συνολικό σφάλμα, η εφαρμογή θα "τρέξει" για 40000 εποχές.

Στο σχήμα 6.2(b) έχουμε τα δεδομένα που επιστρέφει το query με τη μορφή γραφικής συνάρτησης. Απ' τη συγκεκριμένη γραφική μπορούμε να πληροφορηθούμε για το συνολικό αριθμό απεσταλμένων μηνυμάτων για το αντίστοιχο σφάλμα της εφαρμογής. Στο χάρτη μπορούμε να παρατηρήσουμε τους κόκκινους αισθητήρες, οι οποίοι έχουν περιορισμένη διαθέσιμη ενέργεια και πρέπει να αντικατασταθούν.



(a) Ορισμός των κατάλληλων παραμέτρων



(b) Αποτελέσματα της επερώτησης με τη μορφή γραφικής απεικόνισης

Figure 6.2: SQL επερώτηση N<sup>o</sup>1.

**Query N<sub>o</sub>2 :**

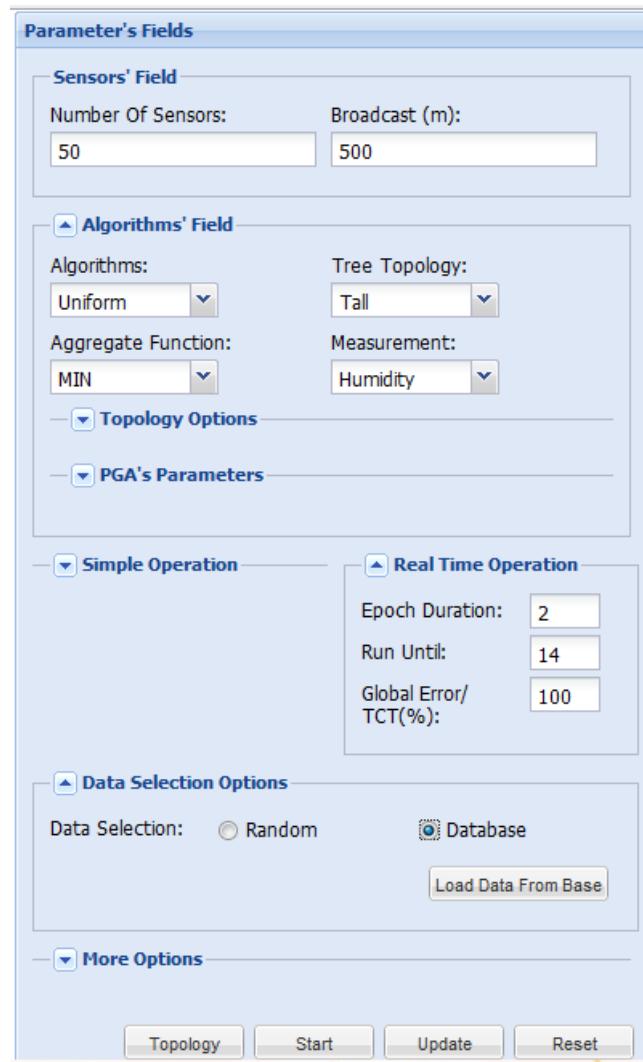
```

SELECT {MIN(Humidity)}
FROM sensors
WHERE {
    Aggregation_Function = Uniform AND
    Tree_Topoogy = Tleaves AND
    Data_Selection = Database AND
    Leaves_Selection = random AND
    Operation = Real_Time AND
    Global_Error = 100 AND
    Run Until= 14 sec
}
EPOCH DURATION 2 sec

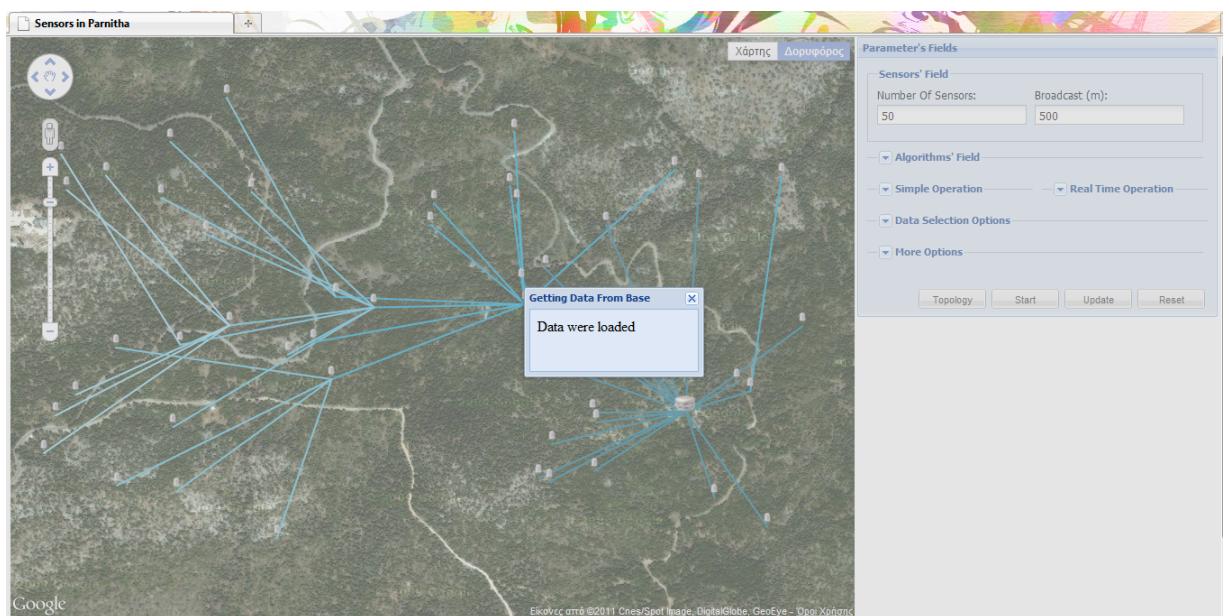
```

Με τη συγκεκριμένη επερώτηση, ζητάμε την ελάχιστη τιμή της υγρασίας των αισθητήρων κάθε 2 δευτερόλεπτα από το 2° μέχρι το 14° δευτερόλεπτο. Ο αλγόριθμος συνάρτησης δεδομένων είναι ο Uniform, η τοπολογία του δέντρου συναθροισης δεδομένων είναι η Tleaves (οι κόμβοι του δέντρου που είναι ενεργοί είναι οι κόμβοι φύλλα) και τα δεδομένα τα αποκτούν οι κόμβοι από τη βάση δεδομένων. Η επιλογή των κόμβων-φύλλων γίνεται τυχαία, η λειτουργία είναι η Real Time και το σφάλμα που δέχεται η εφαρμογή, σε αυτό το είδος λειτουργίας, είναι μόνο ένα, το Global Error = 100. Στην εικόνα 6.3 (b) το ενημερωτικό μήνυμα μας πληροφορεί ότι τα δεδομένα φορτώθηκαν από τη βάση.

Κατά τη διάρκεια εκτέλεσης του συγκεκριμένου query εμφανίζεται στην οθόνη κάθε 2 δευτερόλεπτα η γραφική που απεικονίζει το συνολικό πλήθος απεσταλμένων μηνυμάτων 6.4(a). Στη συνέχεια, για να ενημερωθούμε για το αποτέλεσμα της συναθροιστικής συνάρτησης, από το πεδίο "More Options" επιλέγουμε το Aggregation Result και εμφανίζεται ένας πίνακας στη θέση της γραφικής (σχήμα 6.4 (b)), από τον οποίο πληροφορούμαστε για το συναθροιστικό αποτέλεσμα της αντίστοιχης εποχής.

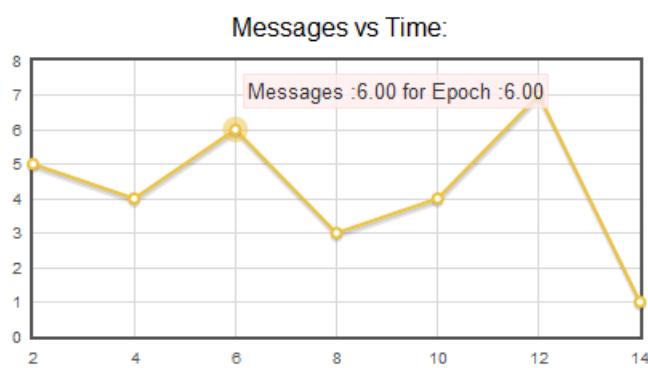


(a) Ορισμός των κατάλληλων παραμέτρων



(b) Ενημερωτικό μήνυμα μετά τη φόρτωση των δεδομένων από τη βάση.

Figure 6.3: SQL επερώτηση N<sup>o</sup>2.



(a) Αποτελέσματα της επερώτησης με τη μορφή γραφικής παράστασης

Aggregation Result	
Aggregation Result	Global Error/EPOCH
-3.7612626552581787	2
-2.3298637866973877	4
-2.576198101043701	6
-2.1313042640686035	8
-2.5682013034820557	10
-2.90846586227417	12
0	14

(b) Αποτελέσματα της επερώτησης με τη μορφή πίνακα

Figure 6.4: Αποτελέσματα SQL επερώτησης N<sup>o</sup>2.

**Query N<sub>o</sub>3 :**

```

SELECT { Select*(Light)}
FROM sensors
WHERE {
    Aggregation_Function = TiNA AND
    Tree_Topoogy = Tall AND
    Data_Selection = Database AND
    Leaves_Selection = random AND
    Operation = Simple AND
    TCT = 5:5:10
}

```

NUMBER OF EPOCHS 5

Σύμφωνα με την παραπάνω επερώτηση, ζητάμε όλες τις τιμές της μέτρησης του φωτός για πέντε εποχές. Ο αλγόριθμος που εφαρμόζεται για τη συνάρθροιση των δεδομένων είναι ο TiNA, η τοπολογία του δέντρου συνάρθροισης δεδομένων είναι η Tall και τα δεδομένα των αισθητήρων προέρχονται απότη βάση δεδομένων. Επίσης, η επιλογή των φύλλων γίνεται τυχαία, έχουμε απλή λειτουργία και το TCT παίρνει τις τιμές 5% και 10% (σχήμα 6.5(a)).

Στο σχήμα 6.5(b) βρίσκεται ο πίνακας που επιστρέφει τα αποτελέσματα της παραπάνω επερώτησης (τοποθετείται στη δεξιά κάτω γωνία της οθόνης, στη θέση της γραφικής συνάρτησης που είδαμε στα παραπάνω σχήματα).

**Parameter's Fields**

**Sensors' Field**

Number Of Sensors:  Broadcast (m):

**Algorithms' Field**

Algorithms:  Tree Topology:

Aggregate Function:  Measurement:

**Topology Options**

**PGA's Parameters**

**Simple Operation**

Number Of Epochs:   
 Global Error/TCT (%):  
 From:   
 To:   
 Step:

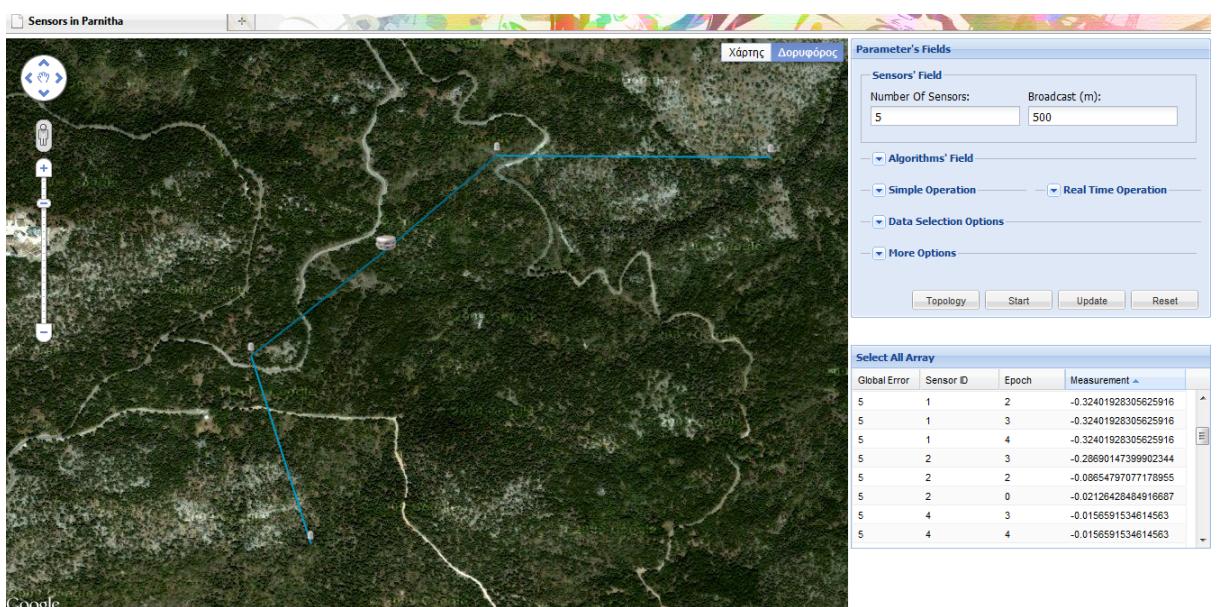
**Real Time Operation**

**Data Selection Options**

**More Options**

Topology Start Update Reset

(a) Ορισμός των κατάλληλων παραμέτρων



(b) Αποτελέσματα της επερώτησης με τη μορφή πίνακα.

Figure 6.5: SQL επερώτηση N°3.

## 6.2 Απόδοση Αλγορίθμων

Στο συγκεκριμένο εδάφιο παρουσιάζεται η απόδοση των αλγορίθμων συνάθροισης δεδομένων, που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία, μεταβάλλοντας κάποιες παραμέτρους. Ο συγχρονισμός των κόμβων-αισθητήρων στους αλγόριθμους TiNA, Uniform, PGA\_Root και PGA\_Local πραγματοποιήθηκε όπως αναφέρθηκε στο TAG (ενότητα 5.3).

### 6.2.1 TAG - TiNA

Στο συγκεκριμένο πείραμα χρησιμοποιήθηκαν 250 αισθητήρες σε τοπολογία Tall (όλοι οι κόμβοι είναι ενεργοί). Το πείραμα έτρεξε για 40000 εποχές με τυχαία δεδομένα, τα οποία ακολούθησαν το πρότυπο "random walk pattern"<sup>1</sup>. Οι αλγόριθμοι συνάθροισης δεδομένων που αναφέρονται εδώ είναι το TAG και το TiNA (0%), με το οποίο αποστέλλονται μόνο οι μετρήσεις που άλλαξαν. Εξετάστηκαν τρεις διαφορετικές συναθροιστικές συναρτήσεις, οι SUM, MAX και AVERAGE.

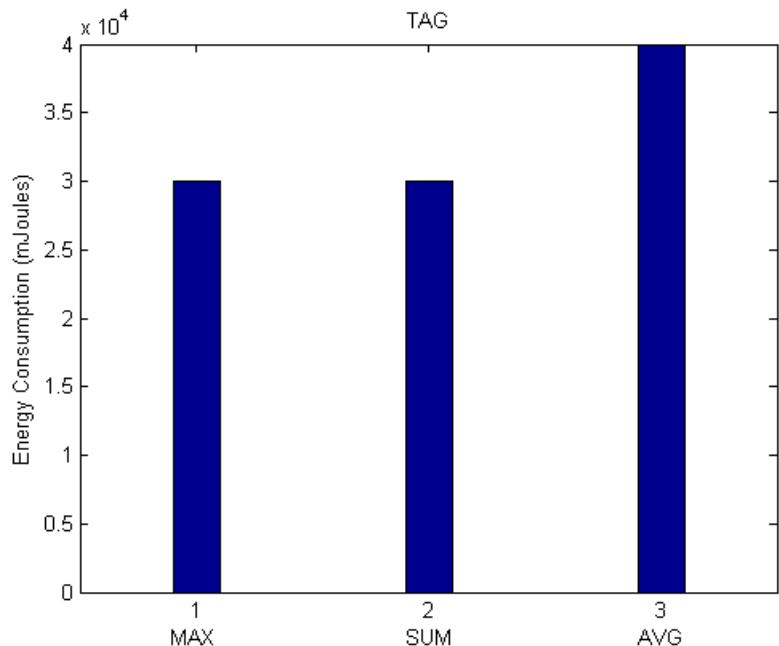
Αρχικά, συγκρίνοντας τα σχήματα 6.6(a) και 6.6(b) μπορεί να διαπιστωθεί ότι η ενέργεια που καταναλώνεται από το TiNA είναι λιγότερη απότην ενέργεια που καταναλώνεται από το TAG. Αυτή η συμπεριφορά ήταν αναμενόμενη, καθώτι με το TiNA δεν αποστέλλονται οι τιμές, που δεν έχουν αλλάξει απότην προηγούμενη εποχή (αφού υπάρχει το φίλτρο  $tct=0\%$ ). Αντιθέτως με το TAG μεταφέρονται όλες οι μετρήσεις των αισθητήρων σε κάθε εποχή.

Παρατηρώντας το σχήμα 6.6, διαπιστώνεται ότι για να υλοποιηθεί η συνάρτηση AVERAGE απαιτείται παραπάνω ενέργεια σε σύγκριση με τις άλλες δύο συναρτήσεις. Για να εφαρμοστεί η συνάρτηση AVERAGE, όπως είδαμε στην ενότητα 5.2 απαιτεί τη μεταφορά δύο εγγραφών (της μέτρησης και του counter) σε αντίθεση με τις συναρτήσεις SUM και MAX κατά τις οποίες μεταφέρεται μόνο μία εγγραφή απότον κάθε κόμβο-αισθητήρα.

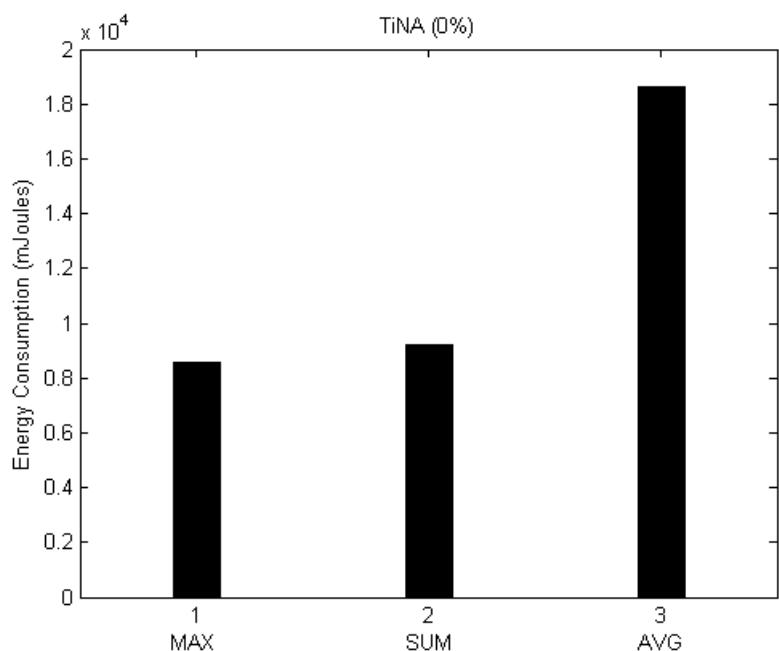
Επιπλέον, αξίζει να σημειωθεί ότι στο TAG οι συναρτήσεις SUM και MAX

<sup>1</sup> Δημιουργούνται τυχαίες τιμές για τους αισθητήρες μία μόνο φορά εντός του διαστήματος [-2,2]. Στη συνέχεια σε κάθε εποχή αναλόγως με το αν ο αισθητήρας είναι "Workaholic", προστίθεται στην προηγούμενη μέτρησή του τυχαία τιμή από το διάστημα [-2,2]

χρησιμοποιούν την ίδια ενέργεια, ενώ στο TiNA η συνάρτηση MAX έχει διαφορετικές απαιτήσεις. Αυτό μπορεί να ερμηνευθεί με ένα παράδειγμα ως εξής: Θεωρούμε ότι έχουμε ένα σύνολο μετρήσεων την εποχή e και και το ίδιο σύνολο από μετρήσεις, εκτός από μία, την εποχή e+1. Επίσης θεωρούμε ότι η MAX τιμή δεν έχει αλλάξει στις δύο διαδοχικές εποχές. Σύμφωνα λοιπόν με το TiNA και για τη συνάρτηση MAX δεν θα αποσταλλεί η μέτρηση της εποχής e+1. Σε περίπτωση όμως, που η συναθροιστική συνάρτηση ήταν η SUM, η μέτρηση της εποχής e+1 θα ήταν διαφορετική από τη μέτρηση της εποχής e και θα μεταφερόταν από τους κόμβους - αισθητήρες.



(a)



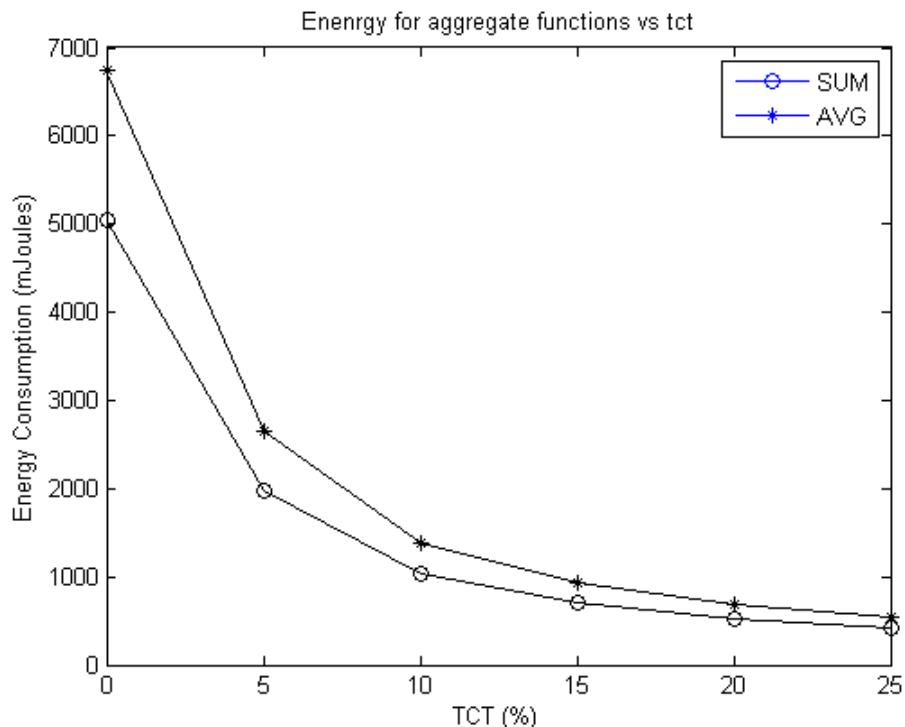
(b)

Σχήμα 6.6: Σύγκριση αλγορίθμων TAG και TiNA(0%) .

### 6.2.2 TiNA από 0% έως 25%

Για το πείραμα που θα αναφερθεί στο παρόν εδάφιο χρησιμοποιήθηκαν οι ίδιοι παράμετροι με το προηγούμενο πείραμα. Στη γραφική του σχήματος 6.7 απεικονίζεται η καταναλισκόμενη ενέργεια του δικτύου αισθητήρων για τις συναθροιστικές συναρτήσεις AVG και SUM, αλλάζοντας μόνο την παράμετρο tct. Ο οριζόντιος άξονας αντιπροσωπεύει το tct, το οποίο κυμαίνεται στο διάστημα [0%,25%] και ο κατακόρυφος άξονας αναφέρεται στην κατανάλωση ενέργειας του δικτύου.

Όπως ήταν αναμενόμενο, καθώς αυξάνεται η τιμή που δίνουμε στην παράμετρο tct η ενέργεια που απαιτεί το δίκτυο μειώνεται. Ο λόγος για τη μείωση της ενέργειας είναι ότι, όσο μεγαλώνουν οι tct τιμές, η πιθανότητα να αποσταλεί μια συναθροιστική τιμή μειώνεται, αφού όπως γνωρίζουμε για να στείλει ένας κόμβος τη νέα μέτρηση που έχει, πρέπει να διαφέρει απ' την τελευταία απεσταλμένη τιμή του περισσότερο από tct.



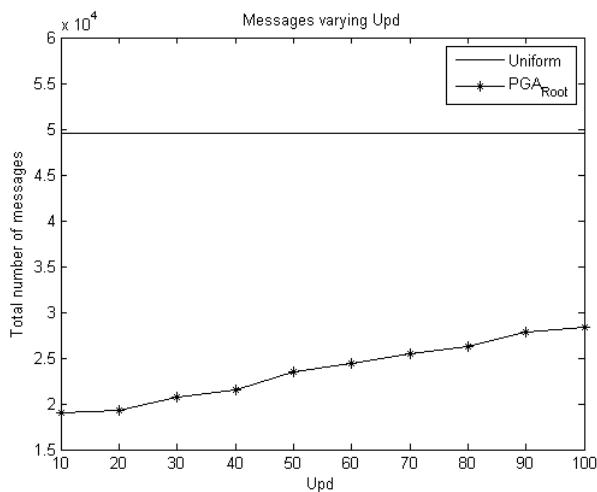
Σχήμα 6.7: TiNA με tct  $\in [0\%, 25\%]$ .

### 6.2.3 PGA και Uniform

Αναφορικά με το πείραμα του σχήματος 6.8, χρησιμοποιήθηκαν 340 κόμβοι-αισθητήρες, 1000 εποχές και η τοπολογία του δέντρου συνάθροισης δεδομένων ήταν η Tall. Ο shrink factor ήταν 0.6 και το συνολικό σφάλμα της εφαρμογής (E\_Global) ήταν 500.

Στη γραφική του σχήματος 6.8 παρατηρείται η συμπεριφορά των αλγορίθμων Uniform και PGA\_Root καθώς μεταβάλλεται ο παράγοντας Upd. Ο κατακόρυφος και ο οριζόντιος άξονας αναφέρονται στα συνολικά απεσταλμένα μηνύματα εντός του δικτύου και στις τιμές που παίρνει το Upd αντίστοιχα. Τα αποτελέσματα του αλγορίθμου Uniform είναι ίδια σε όλες τις περιπτώσεις, εφόσον ο προαναφερόμενος αλγόριθμος δεν σχετίζεται με τη συγκεκριμένη παράμετρο.

Στην ενότητα 5.6 είδαμε ότι οι αλγόριθμοι PGA (PGA\_Root και PGA\_Local) προσαρμόζουν τα φίλτρα των κόμβων κάθε Upd εποχές. Στο σχήμα 6.8 παρατηρείται ότι όσο το Upd αυξάνεται, αποστέλλονται εντός του δικτύου περισσότερα μηνύματα. Αυτό οφείλεται στο γεγονός ότι όσο αυξάνεται η παράμετρος Upd, μειώνονται οι φάσεις ανανέωσης και ο αλγόριθμος δεν προλαβαίνει να προσαρμόσει κατάλληλα τα φίλτρα στους κόμβους.

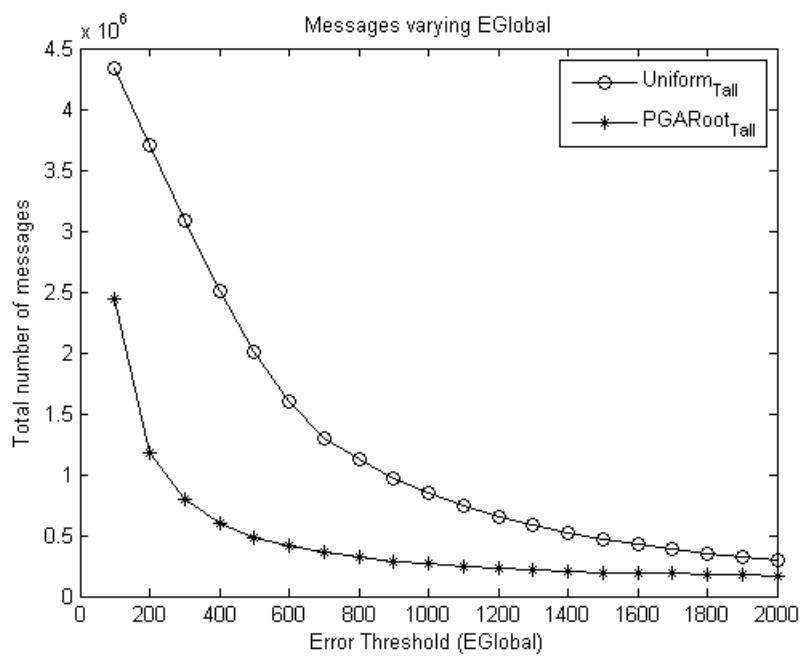


Σχήμα 6.8: Συμπεριφορά των αλγορίθμων Uniform και PGA\_Root αλλάζοντας την παράμετρο Upd.

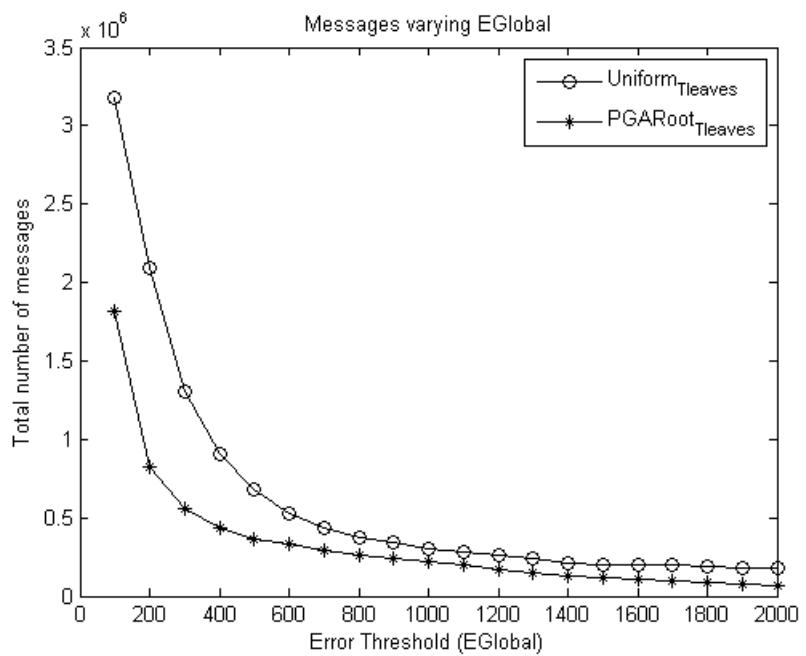
Η γραφική 6.9 παρουσιάζει τη συμπεριφορά των αλγορίθμων Uniform και PGA\_Root αλλάζοντας το συνολικό σφάλμα της εφαρμογής για τις τοπολογίες Tall και Tleaves αντίστοιχα. Οι κόμβοι-αισθητήρες στο παρόν πείραμα είναι 340, η παράμετρος Upd είναι 50 και το πλήθος των εποχών ισούται με 40000. Επίσης, ο shrink factor είναι ίσος με 0.6.

Απ'το σχήμα 6.9 μπορεί να διαπιστωθεί ότι όσο αυξάνεται το συνολικό σφάλμα που δέχεται η εφαρμογή, τόσο λιγότερα μηνύματα ανταλλάζονται. Αυτό είναι αναμενόμενο, αφού οι κόμβοι θα έχουν μεγαλύτερο φίλτρο και δεν θα αποστέλλουν τιμές με μικρή απόκλιση μεταξύ τους. Επιπλέον, παρατηρείται ότι η απόδοση του αλγορίθμου PGA\_Root είναι πολύ καλύτερη απ'αυτή του Uniform. Αυτό οφείλεται στο γεγονός, ότι ο PGA\_Root προσαρμόζει τα φίλτρα στους κόμβους ανάλογα με την λειτουργία τους στο δίκτυο (αν αποστέλλουν πολλά ή λίγα μηνύματα) σε αντίθεση με τον Uniform, που κρατάει τα φίλτρα σταθερά στους κόμβους.

Συγκρίνοντας τις δύο τοπολογίες του δικτύου στις εικόνες 6.9(a) και 6.9(b) διαπιστώνεται ότι στην τοπολογία Tleaves (οι ενεργοί κόμβοι στο συγκεκριμένο πείραμα ήταν 161) ανταλλάζονται λιγότερα μηνύματα εντός του δικτύου. Το αναμενόμενο θα ήταν να παρουσιάζονται λιγότερα μηνύματα στην τοπολογία που έχει τα περισσότερα φίλτρα. Έστω όμως ότι έχουμε τον αλγόριθμο Uniform και δύο δέντρα συνάθροισης δεδομένων με τις τοπολογίες Tall και Tleaves αντίστοιχα. Οι κόμβοι-φύλλα και στις δύο τοπολογίες θεωρούμε ότι παίρνουν τις ίδιες τιμές. Περισσότερες μετρήσεις απότομης - φύλλα θα αποσταλούν απ'το δέντρο συνάθροισης της τοπολογίας Tall, αφού τα φίλτρα των συγκεκριμένων κόμβων ( $E_i = E_{Global} \setminus 340$ ) θα είναι μικρότερο απότοιχα φίλτρα της τοπολογίας Tleaves (  $E_i = E_{Global} \setminus 161$ ). Αυτό έχει ως αποτέλεσμα να μεταφερθούν περισσότερες μετρήσεις απότοιχα φύλλα του δέντρου συνάθροισης δεδομένων της τοπολογίας Tall σε σύγκριση με την τοπολογία Tleaves.



(a) Τοπολογία Tall



(b) Τοπολογία Leaves

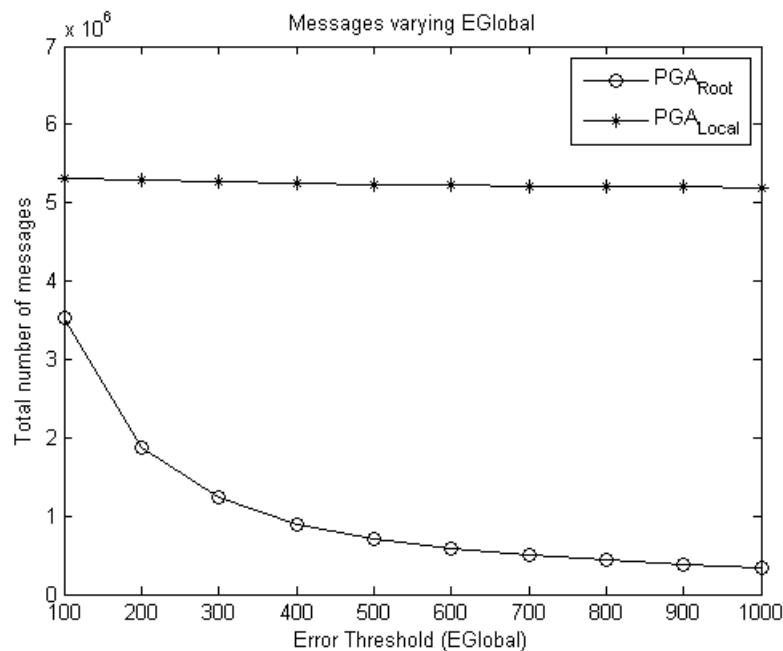
Σχήμα 6.9: Απόδοση των αλγορίθμων PGA\_Root και Uniform για τις δύο τοπολογίες του δικτύου.

#### 6.2.4 PGA\_Root και PGA\_Local

Στη συγκεκριμένη ενότητα παρατηρείται η απόδοση του PGA αλγορίθμου μοιράζοντας το πλεόνασμα λάθους με τους δύο διαφορετικούς τρόπους που αναπτύχθηκαν στο εδάφιο 5.6.4.

Το συγκεκριμένο πείραμα πραγματοποιήθηκε με 300 αισθητήρες, για 40000 εποχές, με Upd παράμετρο ίση με 50 και shrink factor ίσο με 0.6. Η τοπολογία που χρησιμοποιήθηκε ήταν η Tall. Στη γραφική του σχήματος 6.10 έχουμε την απεικόνιση των συνολικών απεσταλμένων μηνυμάτων συναρτήσει του συνολικού σφάλματος εφαρμογής.

Διαπιστώνεται απ'το σχήμα 6.10 ότι η απόδοση του αλγορίθμου PGA\_Root είναι σαφώς καλύτερη απ'την απόδοση του αλγορίθμου PGA\_Local. Ο αλγόριθμος PGA\_Local συνίσταται σε περιπτώσεις, όπου η κατανομή των δεδομένων στους κόμβους - αισθητήρες είναι σχετικά στατική.



Σχήμα 6.10: Σύγκριση αλγορίθμων PGA\_Root και PGA\_Local.

# Βιβλιογραφία

- [1] S.Madden, M.Franklin, J.Hellerstein and W.Hong. *TAG: A tiny aggregation service for ad-hoc sensor networks.* In Proc. of OSDI, Dec. 2002.
- [2] M.A. Sharaf, J.Beaver, A.Labrinidis, P.K. Chrysanthis. *TiNA: a scheme for temporal coherency-aware in-network aggregation.* In Proc.of ACM MobiDE Workshop, Sep.2003
- [3] A.Deligiannakis, Y.Kotidis, N.Roussopoulos. *Processing approximate aggregate queries in wireless sensor networks.* Information Systems, vol. 31, no. 8, pp. 770-792, Dec. 2006.
- [4] "The Handbook of Computer Networks", J.Willey & Sons, Dec.2007.
- [5] "Beginning Google Maps API 3", G.Svennerberg, 2010.
- [6] *Ad-hoc network:* [http://el.wikipedia.org/wiki/Wireless\\_Network#Ad\\_hoc\\_.CE.B4.CE.AF.CE.BA.CF.84.CF.85.CE.B1](http://el.wikipedia.org/wiki/Wireless_Network#Ad_hoc_.CE.B4.CE.AF.CE.BA.CF.84.CF.85.CE.B1).
- [7] *Google Maps:* <http://code.google.com/intl/el-GR/apis/maps/documentation/javascript/>.
- [8] Συντεταγμένες Πάρνηθας: <http://www.easypedia.gr/el/articles/%CF%80/%CE%AC/%CF%81/%CE%A0%CE%AC%CF%81%CE%BD%CE%B7%CE%B8%CE%B1.html>.
- [9] *Point\_In\_Polygon Algorithm-PolygonDetermining whether a point is inside a complex polygon.* <http://www.alienryderflex.com/polygon/>.
- [10] *Haversine formula:* [http://en.wikipedia.org/wiki/Haversine\\_formula](http://en.wikipedia.org/wiki/Haversine_formula).
- [11] *Circle's cartesian coordinates:* <http://en.wikipedia.org/wiki/Circle>.
- [12] *Query:* <http://en.wikipedia.org/wiki/Query>.