

Ατομική Διπλωματική Εργασία

**ΚΙΝΗΤΙΚΟΤΗΤΑ ΓΙΑ ΕΛΕΓΧΟ ΠΕΡΙΟΧΗΣ ΣΕ ΑΣΥΡΜΑΤΑ
ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ**

Πάρης Ιωνά

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2010

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Κινητικότητα για Έλεγχο περιοχής σε Ασύρματα Δίκτυα Αισθητήρων

Πάρης Ιωνά

Επιβλέπων Καθηγητής

Βάσος Βασιλείου

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

Μάιος 2010

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Βάσο Βασιλείου για την καθοδήγηση και υποστήριξη που μου παρείχε σε όλη την διάρκεια της εκπόνησης αυτής της εργασίας, αλλά και την επίτευξη μιας άψογης και εποικοδομητικής συνεργασίας. Τέλος θα ήθελα να ευχαριστήσω την οικογένεια μου που με στήριξε σε όλη την διάρκεια των σπουδών μου.

Περίληψη

Ο τομέας των Ασύρματων Δικτύων Αισθητήρων παρουσιάζει τα τελευταία χρόνια μεγάλη ζήτηση σε ερευνητικό επίπεδο. Η εισαγωγή νέων τεχνολογιών και η ανάπτυξη των ήδη υπάρχων έδωσε πρόσφορο έδαφος για μεγάλες αλλαγές και βελτιώσεις σε αυτό τον τομέα. Η διπλωματική αυτή εργασία έχει ως στόχο την γενική παρουσίαση των ασύρματων δικτύων αισθητήρων. Πιο συγκεκριμένα θα ασχοληθεί με την κινητικότητα στα ασύρματα δίκτυα αισθητήρων καθώς επίσης και την παρουσίαση δύο διαφορετικών ομάδων αλγορίθμων σχετικών με πρωτόκολλα κινητικότητας. Θα μελετηθούν τα ήδη υπάρχοντα πρωτόκολλα και θα παρουσιαστούν και στη συνέχεια θα παρουσιαστούν αναλυτικά ένας αλγόριθμος για κάθε ομάδα που χρησιμοποιεί ρομπότ για την εξυπηρέτηση του δικτύου σε θέματα κινητικότητας.

Περιεχόμενα

Κεφάλαιο 1	Εισαγωγή.....	1
	1.1 Ασύρματα Δίκτυα Αισθητήρων	1
	1.2 Κόμβοι Αισθητήρων	4
	1.3 Προβλήματα-Προκλήσεις στα Ασύρματα Δίκτυα Αισθητήρων	7
Κεφάλαιο 2	Κινητικότητα στα Ασύρματα Δίκτυα Αισθητήρων.....	9
	2.1 Γενικά	9
	2.2 Μορφές κινητικότητας	10
Κεφάλαιο 3	Πρωτόκολλα κινητικότητας	12
	3.1 Αλγόριθμοι κάλυψης περιοχής	12
	3.2 Αλγόριθμοι ελέγχου περιμέτρου	17
	3.3 Αλγόριθμοι ομαδοποίησης	19
Κεφάλαιο 4	Προβλήματα προς επίλυση.....	21
	4.1 Γενικά	21
	4.2 Έλεγχος-Περιπολία περιοχής	24
	4.3 Ομαδοποίηση κόμβων σε κινητό ασύρματο δίκτυο	26
	4.4 Αποτελέσματα-Ανάλυση αποτελεσμάτων	33
Κεφάλαιο 5	Μελλοντική δουλειά.....	46
	5.1 Γενικές βελτιώσεις-επεκτάσεις-εισηγήσεις	46
Κεφάλαιο 6	Συμπεράσματα.....	49
	6.1 Γενική Σύνοψη	49
	6.2 Συμπεράσματα	50
Βιβλιογραφία		52
Παράρτημα Α.....		A-1

Κεφάλαιο 1

Εισαγωγή

1.1 Ασύρματα Δίκτυα Αισθητήρων	1
1.2 Κόμβοι Αισθητήρων	4
1.3 Προβλήματα-Προκλήσεις στα Ασύρματα Δίκτυα Αισθητήρων	7

1.1 Ασύρματα Δίκτυα Αισθητήρων

Τα Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks, WSNs), είναι ένας τομέας που προέκυψε από την πρόοδο των τομέων των επικοινωνιών και των μικροηλεκτρονικών συστημάτων. Τα δίκτυα αυτά αποτελούνται από μικρό ή μεγάλο αριθμό κόμβων-αισθητήρων, κινητών ή μη, που αποστέλλουν σε μια κεντρική μονάδα (sink) πληθώρα δεδομένων για επεξεργασία και λήψη αποφάσεων. Τα δεδομένα αυτά μπορεί να είναι θερμοκρασία, φωτεινότητα, ατμοσφαιρική πίεση, κίνηση αντικειμένων και πολλά άλλα περιβαλλοντικά μεγέθη. Όλα αυτά μπορούν να μετρώνται από ένα αισθητήρα στο μέγεθος ενός νομίσματος, όπως βλέπουμε και στο Σχήμα 1.1.



Σχήμα 1.1: Κόμβος αισθητήρα Mica2dot σε σχέση με νόμισμα δύο Ευρώ

Τα WSNs έχουν αρκετά χαρακτηριστικά που τα καθιστούν μοναδικά και ταυτόχρονα μια ξεχωριστή ομάδα από τα συμβατικά ασύρματα δίκτυα. Μερικά από αυτά είναι:

- **Application specific:** Ένα ήδη υπάρχον δίκτυο ασύρματων αισθητήρων που έχει ήδη αναπτυχθεί και σχεδιαστεί για μια συγκεκριμένη εφαρμογή, είναι πολύ δύσκολο να αλλάξει και να χρησιμοποιηθεί για άλλη εφαρμογή, χωρίς να αλλάξουν σε αυτό σε μεγάλο βαθμό ο αλγόριθμος λειτουργίας καθώς επίσης και η διάταξη των αισθητήρων και το υλικό που χρησιμοποιούμε. Για παράδειγμα για μια εφαρμογή στην οποία αισθητήρες βρίσκονται πάνω σε ρομπότ που αλληλεπιδρούν μεταξύ τους χρησιμοποιούμε τελείως διαφορετική λογική και σχεδιασμό από μια δεύτερη εφαρμογή όπου οι αισθητήρες είναι στατικοί και κάνουν κάποιο είδος περιοδικής μέτρησης, για παράδειγμα θερμοκρασίας σε ένα εργοστάσιο.
- **Κλίμακα μεγέθους:** Ένα WSN μπορεί να απαρτίζεται από μερικές δεκάδες μέχρι αρκετές χιλιάδες ή και περισσότερους κόμβους αισθητήρων. Άρα η αρχιτεκτονική και τα πρωτόκολλα που χρησιμοποιούμε πρέπει να είναι σε θέση να αντεπεξέλθουν με αυτούς τους αριθμούς.
- **Περιορισμένη ενέργεια και Χρόνος ζωής:** Λόγω του μικρού μεγέθους των κόμβων αισθητήρων και του τρόπου που χρησιμοποιούνται (για παράδειγμα ρίψη από αεροπλάνο σε ένα δάσος που καίγεται για συλλογή πληροφοριών), η ενέργεια που διαθέτει κάθε κόμβος αισθητήρα και κατ' επέκταση ο χρόνος ζωής ολόκληρου του δικτύου είναι περιορισμένα. Ο χρόνος ζωής είναι αντιστρόφως ανάλογος της ποιότητας λειτουργίας που έχουμε στο WSN κάθε φορά, έτσι πρέπει να γίνεται σωστή οριοθέτηση των στόχων και του τρόπου λειτουργίας του δικτύου για κάθε εφαρμογή.
- **Επεκτασιμότητα και Ανοχή σφαλμάτων:** Σε ένα δίκτυο αισθητήρων έχουμε πάρα πολύ συχνά αλλαγή στον αριθμό των κόμβων αισθητήρων. Αυτό συμβαίνει βασικά για δύο λόγους. Πρώτο από την εισαγωγή νέων κόμβων στο δίκτυο και δεύτερο από την απώλεια κόμβων είτε λόγω καταστροφής τους είτε λόγω εξάντλησης της ενέργειάς τους. Άρα οι αλγόριθμοι που χρησιμοποιούνται πρέπει να έχουν μηχανισμούς για να μπορούν να αντεπεξέλθουν σε αυτές τις καταστάσεις χωρίς προβλήματα.
- **Quality of service:** Σε ένα ασύρματο δίκτυο αισθητήρων για την ποιότητα της υπηρεσίας δεν λαμβάνονται υπόψη οι ίδιοι παράμετροι όπως στα συμβατικά ασύρματα δίκτυα. Δεν έχουμε πρωταρχικό σκοπό την ταχύτητα και το ρυθμό

μετάδοσης αλλά την αξιόπιστη μεταφορά όλων των πακέτων χωρίς πολλές αναμεταδόσεις.

- **Απλότητα:** Από τη στιγμή που κάθε κόμβος έχει περιορισμένες υπολογιστικές δυνατότητες και πόρους, το λειτουργικό καθώς και ο αλγόριθμος που χρησιμοποιείται δεν θα πρέπει να έχουν πολύπλοκες απαιτήσεις.
- **Αυτόνομη λειτουργία:** Αφού μιλούμε για ασύρματα δίκτυα αισθητήρων, θα πρέπει από τη στιγμή που θα γίνεται η τοποθέτηση τους και θα ξεκινήσει η λειτουργία τους θα πρέπει πλέον να μπορούν να λειτουργήσουν αυτόνομα χωρίς την συμμετοχή προγραμματιστή ή κάποιου χρήστη, σε περίπτωση σημαντικών μεταβολών στα μετρούμενα μεγέθη.

Μεταξύ του κόμβου που λαμβάνει τα δεδομένα (sink) και των κόμβων αισθητήρων που τα λαμβάνουν και τα αποστέλλουν έχουμε διάφορα σχέδια αλληλεπίδρασης. Τα πιο τυπικά είναι:

- **Εντοπισμός Συμβάντος(Event detection):** Οι κόμβοι αποστέλλουν στο sink μόνο όταν εντοπίσουν το γεγονός για το οποίο είναι προγραμματισμένοι, για παράδειγμα σε περίπτωση δικτύου που ελέγχει για πυρκαγιά σε ένα δάσος ή ένα εργοστάσιο.
- **Περιοδικές μετρήσεις:** Οι κόμβοι είναι ρυθμισμένοι έτσι ώστε να αποστέλλουν στα sink ανά τακτά χρονικά διαστήματα τις μετρήσεις που έχουν προγραμματιστεί να παίρνουν, όπως για παράδειγμα στην περίπτωση της καταγραφής της θερμοκρασίας σε ένα χώρο.
- **Παρακολούθηση:** Με την είσοδο κάποιου “εισβολέα” στο χώρο όπου επιβλέπετε από το WSN, αμέσως τίθεται σε συναγερμό το σύστημα με άμεσα μηνύματα στο sink.
- **Edge detection:** Σε αυτή την περίπτωση βρίσκουμε οριακές τιμές ή άλλες δομές για μια κατάσταση και τις αποστέλλουμε στο sink. Παράδειγμα είναι η εύρεση μέγιστης ή ελάχιστης θερμοκρασίας σε ένα χώρο συνολικά. Αυτό προϋποθέτει και μια μικρή επεξεργασία δεδομένων στον κόμβο αισθητήρα προτού στείλει παρακάτω τα δεδομένα του.
- **Function approximation:** Εδώ χρησιμοποιούμε το ασύρματο δίκτυο αισθητήρων για να προσεγγίσουμε μια κατάσταση ενός χώρου ή ακόμα και του χρόνου. Ένα απλό παράδειγμα είναι η κατασκευή χάρτη θερμοκρασίας σε ένα δωμάτιο.

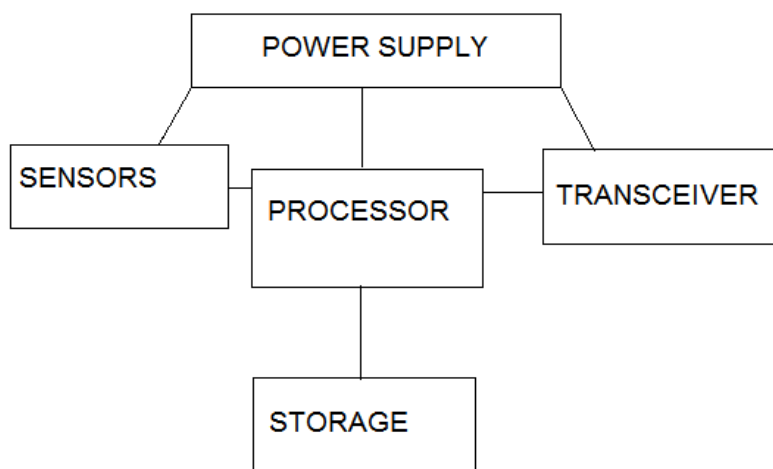
Τις περισσότερες φορές χρησιμοποιούμε τα ασύρματα δίκτυα αισθητήρων για να κάνουμε ένα είδος παρακολούθησης ή ελέγχου. Όμως δεν περιορίζονται οι εφαρμογές τους μόνο σε

αυτά. Σημαντικοί τομείς για εφαρμογές WSNs είναι οι επιχειρήσεις βοήθειας για φυσικές καταστροφές, όπως στην περίπτωση δασικής πυρκαγιάς, η καταγραφή βιοποικιλότητας, με τη χρήση των κόμβων αισθητήρων για την παρακολούθηση του ζωικού βασιλείου, η διαχείριση εγκαταστάσεων, η επιτήρηση μηχανών, η ιατρική και η υγιεινή(παρακολούθηση με τη χρήση WSNs από απόσταση ηλικιωμένων ανθρώπων) καθώς επίσης και σε πληθώρα στρατιωτικών εφαρμογών, για παρακολούθηση του εχθρού ή την φρούρηση χωρίς την χρήση ανθρώπινου δυναμικού μιας περιοχής. Ακόμη μπορούμε να χρησιμοποιήσουμε τα WSNs σε τομείς της γεωργίας, σε έξυπνα κτήρια και σε εφαρμογές λογιστικών.

Όσον αφορά τώρα τον τρόπο που γίνεται η ανάπτυξη των ασύρματων δικτύων αισθητήρων έχουμε τρεις διαφορετικούς τρόπους, κατά κύριο λόγο. Μπορούμε να έχουμε μια τυχαία κατανομή των κόμβων από αεροπλάνο για παράδειγμα σε μια μεγάλη περιοχή. Ακόμη μπορούμε πιο απλά να έχουμε τοποθέτηση των κόμβων σε προκαθορισμένα σημεία που θέλουμε, συνήθως με γεωγραφικά κριτήρια. Τελευταίος αλλά πιο αποδοτικός τρόπος ανάπτυξης είναι η χρήση κινητών κόμβων, σε δίκτυα που αλλάζουν δυναμικά οι θέσεις των κόμβων σύμφωνα με τις ανάγκες της εφαρμογής, με τη χρήση για παράδειγμα ρομπότ για την “έξυπνη” κίνηση τους στο χώρο.

1.2 Κόμβοι Αισθητήρων

Αν και υπάρχουν πολλά μοντέλα και σχέδια για κόμβους-αισθητήρες, υπάρχει μια βασική αρχιτεκτονική που όλοι ακολουθούν σε μεγάλο βαθμό, και την βλέπουμε και στο Σχήμα 1.2.



Σχήμα 1.2: Βασική δομή αρχιτεκτονικής ενός κόμβου-αισθητήρα

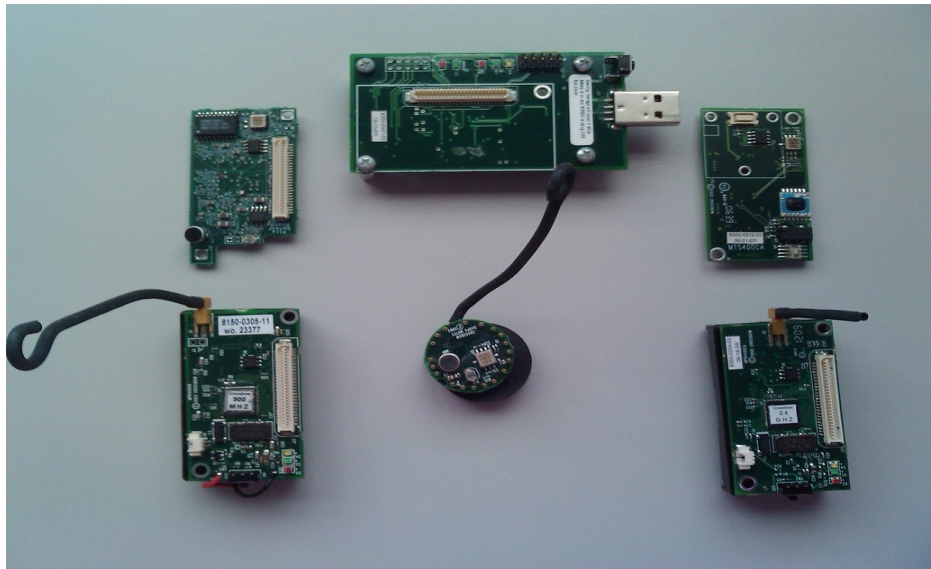
Τα κύρια μέρη ενός κόμβου-αισθητήρα, είναι ο μικροεπεξεργαστής, ο πομποδέκτης, η εξωτερική μνήμη, η πηγή ενέργειας και ένας ή περισσότεροι αισθητήρες [11].

- Ο **μικροεπεξεργαστής (microprocessor)** είναι ο πυρήνας του κόμβου. Δέχεται τα δεδομένα από τους αισθητήρες ή από τους άλλους κόμβους, τα επεξεργάζεται και αποφασίζει πότε και που θα τα στείλει. Χρησιμοποιούνται μικροεπεξεργαστές επειδή είναι η καλύτερη λύση όσο αφορά εξοικονόμηση ενέργειας, αφού μπορούμε να τους προγραμματίσουμε εύκολα να μπαίνουν σε κατάσταση ύπνου(sleep mode), όταν δεν είναι ανάγκη να δουλέψουν και έτσι δεν σπαταλιέται άσκοπα ενέργεια.
- Ο **πομποδέκτης (transceiver)** είναι υπεύθυνος για την αμφίδρομη επικοινωνία των κόμβων. Εκτός από την απλή μετάδοση των δεδομένων, πολλοί πομποδέκτες αναλαμβάνουν και άλλες εργασίες που αφορούν το πρωτόκολλο επικοινωνίας, μειώνοντας έτσι το φόρτο εργασίας του μικροεπεξεργαστή. Υπάρχουν διάφοροι τρόποι επικοινωνίας μεταξύ των κόμβων. Οι πιο διαδεδομένοι είναι με ραδιοσυχνότητα (RF), υπέρυθρες ακτίνες (infrared) και laser. Ο πιο κατάλληλος για επικοινωνία από τους τρεις τρόπους είναι με την χρήση των ραδιοσυχνοτήτων. Οι άλλοι δύο τρόποι, αν και δεν χρειάζονται αντένα για να αποστείλουν και να λάβουν τα δεδομένα, εντούτοις έχουν περιορισμένη ραδιοφωνική αναμετάδοση και χρειάζονται αρκετές ρυθμίσεις για να μπορούν να λειτουργήσουν καλά (ευθυγράμμιση κατά κύριο λόγο). Από την άλλη με τις ραδιοσυχνότητες μπορούμε να έχουμε μετάδοση και παραλαβή πακέτων προς όλες τις κατευθύνσεις χωρίς πρόβλημα.
- Η **μνήμη** σε ένα κόμβο αισθητήρα χρησιμοποιείται για να προγραμματίσουμε τον κόμβο, για να φυλάξει δεδομένα της εφαρμογής καθώς και για να αποθηκεύονται εκεί αναγνωριστικά του κόμβου. Συνήθως η μνήμη αυτή εμφανίζεται σαν ενσωματωμένη στον μικροεπεξεργαστή ή ακόμα και σαν επιπρόσθετη flash μνήμη.
- Η **πηγή ενέργειας** σε ένα κόμβο αισθητήρα μπορεί να είναι είτε κάποια μπαταρία είτε κάποιοι πυκνωτές που συγκερατούν ενέργεια. Τα τελευταία χρόνια έχουν αναπτυχθεί κόμβοι με τη δυνατότητα να παίρνουν ενέργεια από το περιβάλλον τους και να ανανεώνουν τα αποθέματα που έχουν, όπως με χρήση της ηλιακής ενέργειας.

Όσον αφορά τους πομποδέκτες, αυτοί έχουν τέσσερις βασικές λειτουργικές καταστάσεις:

- **Κατάσταση Εκπομπής (Transmit state):** ο πομποδέκτης είναι ενεργός και αποστέλλει τα δεδομένα που έχει ο κόμβος αισθητήρα. Βασικά ενεργό είναι το μέρος που στέλνει τα δεδομένα και όχι το μέρος που λαμβάνει.

- **Κατάσταση Λήψης (Receive state):** πάλι ο πομποδέκτης είναι ενεργός, όμως σε αυτή την περίπτωση ενεργό είναι το μέρος του πομποδέκτη που είναι υπεύθυνο για την λήψη μηνυμάτων από τους άλλους κόμβους. Έτσι γίνεται η παραλαβή των δεδομένων.
- **Ανενεργή Κατάσταση (Idle state):** Σε αυτή την κατάσταση ο πομποδέκτης αν και είναι έτοιμος να λάβει δεδομένα, δεν λαμβάνει τίποτα. Ορισμένα μέρη του είναι ενεργά αλλά και αρκετά είναι τελείως ανενεργά.
- **Κατάσταση Ύπνου (Sleep state):** εδώ έχουμε τα περισσότερα μέρη του πομποδέκτη ανενεργά. Σε αντίθεση με την idle state, εδώ ο πομποδέκτης δεν είναι έτοιμος να λάβει δεδομένα. Ξυπνά όμως ανά τακτά χρονικά διαστήματα και βλέπει αν υπάρχει τίποτα που τον ενδιαφέρει και μετά ξαναμπαίνει στην κατάσταση ύπνου, για όσο το δυνατόν μεγαλύτερη εξοικονόμηση ενέργειας.



Σχήμα 1.3: Κόμβοι αισθητήρες της οικογένειας Mica, μαζί με το programming board τους (πάνω στο κέντρο)

Μπορούμε να κατατάξουμε τους αισθητήρες σε τρεις ομάδες.

- **Παθητικοί - omnidirectional:** είναι οι αισθητήρες που μετρούν ένα φυσικό μέγεθος προς όλες τις κατευθύνσεις, χωρίς να επέμβουν στο περιβάλλον.
- **Παθητικοί – narrow-beam:** σε αντίθεση με την πρώτη ομάδα, σε αυτή το φυσικό μέγεθος που μετριέται περιορίζεται σε μια γωνία ή κατεύθυνση. Πιο χαρακτηριστικό παράδειγμα είναι όταν έχουμε μια κάμερα που κοιτάζει μόνο σε ένα χώρο.
- **Ενεργητικοί:** αντίθετα με τις δύο πιο πάνω ομάδες, αυτοί οι αισθητήρες για να πάρουν μια μέτρηση πρέπει να επέμβουν στο περιβάλλον (πχ. Αισθητήρες σεισμικών δονήσεων) .

1.3 Προβλήματα-Προκλήσεις στα Ασύρματα Δίκτυα Αισθητήρων

Λόγω της φύσης τους τα ασύρματα δίκτυα αισθητήρων έχουν πολλά προβλήματα και προκλήσεις που πρέπει να αντιμετωπιστούν. Τα σημαντικότερα και πιο κοινά από αυτά θα αναπτυχθούν πιο κάτω.

- Σημαντικότερο από όλα είναι το ζήτημα της ενέργειας. Λόγω της φύσης τους και της δουλειάς για την οποία είναι σχεδιασμένοι, δεν μπορούν να έχουν μεγάλες ποσότητες αποθηκευμένης ενέργειας. Έτσι πρέπει να δίνεται μεγάλη προσοχή στο πως μπορούμε να εξοικονομήσουμε όσο το δυνατό περισσότερη ενέργεια μέσα από τις εφαρμογές που αναπτύσσουμε. Ακόμη αρχίζουν τώρα να κυκλοφορούν αισθητήρες που μπορούν να αναπληρώνουν την αποθηκευμένη τους ενέργεια από ανανεώσιμες πηγές ενέργειας, όπως την ηλιακή.
- Έχουμε περιορισμένες δυνατότητες όσον αφορά την υπολογιστική δύναμη και τις άλλες δυνατότητες των κόμβων αισθητήρων και κατ' επέκταση των WSNs, αφού πρέπει να είναι μικροί οι κόμβοι και σχετικά φτηνοί αφού οι αριθμοί που χρειαζόμαστε πρακτικά είναι πολύ μεγάλοι.
- Τα ασύρματα δίκτυα αισθητήρων είναι ευάλωτα σε “θανάτους” κόμβων τους, συχνά από εξάντληση των αποθεμάτων ενέργειας ή άλλων περιβαλλοντικών αλλαγών που μπορούν να τα επηρεάσουν. Έτσι οι εφαρμογές πρέπει να είναι έτοιμες να αντιμετωπίσουν αποτελεσματικά την αλλαγή στην τοπολογία του δικτύου και να μπορούν δυναμικά να διαχειρίζονται τις απώλειες χωρίς μεγάλα προβλήματα.
- Εκτός από το να είναι ανεκτικά στις απώλειες, τα δίκτυα ασύρματων αισθητήρων θα πρέπει επίσης να μπορούν να επεκταθούν αν χρειαστεί. Έτσι όλες οι εφαρμογές πρέπει να μπορούν να δεχτούν ανά πάσα στιγμή επιπρόσθετους κόμβους-αισθητήρες στο ήδη υπάρχον δίκτυο τους και να τους εντάξουν ομαλά και χωρίς προβλήματα.
- Λόγω του γεγονότος ότι συνήθως έχουμε πάρα πολλούς κόμβους-αισθητήρες σε ένα δίκτυο που είναι διασκορπισμένοι συνήθως σε αφιλόξενα περιβάλλοντα με πολύ μεγάλη κλίμακα θορύβου, πρέπει να υπάρχουν μηχανισμοί για να μπορούμε να έχουμε άμεση απόκριση από τους κόμβους, αφού οι εφαρμογές που τρέχουμε είναι σχεδόν πάντα εφαρμογές πραγματικού χρόνου.
- Το πόσο πυκνοί είναι οι αισθητήρες μας μέσα στο δίκτυο μπορεί επίσης να επηρεάσει την επίδοση του. Αν έχουμε περισσότερους από ό,τι θα έπρεπε, θα παρατηρούνται

φαινόμενα συμφόρησης μέσα στο ίδιο το δίκτυο με αποτέλεσμα να χάνουμε πακέτα, άρα να έχουμε περισσότερα λάθη στα αποτελέσματα μας.

Η ατομική αυτή διπλωματική εργασία θα ασχοληθεί με τα θέματα τις κάλυψης τρυπών σε ένα ασύρματο δίκτυο αισθητήρων, με τον εντοπισμό κόμβων με κοινά χαρακτηριστικά σε ένα δίκτυο και την περιμετρική περιπολία ενός χώρου από μια ομάδα κόμβων. Όλοι οι αλγόριθμοι έχουν αναπτυχθεί σε γλώσσα προγραμματισμού C, με την ενσωμάτωση της βιβλιοθήκης CP142.

Μέσα στα πλαίσια αυτής της διπλωματικής εργασίας εξοικειώθηκα και με την χρήση του λογισμικού TinyOs, αν και δεν το χρησιμοποίησα σε κάποια από τις εφαρμογές που έχω αναπτύξει. Υπάρχει ένα οδηγός για το πως μπορεί κάποιος να εγκαταστήσει το λογισμικό αυτό πάνω σε ένα υπολογιστή καθώς και το πως μπορεί να τρέξει μια ενδεικτική εφαρμογή, πάνω σε αισθητήρες MicaZ, στο παράρτημα A-1. Ακόμη υπάρχουν στο παράρτημα και παραπομπές σε ιστοσελίδες για τυχόν προβλήματα που μπορεί να εντοπιστούν, που εισηγούνται τρόπους επίλυσής τους.

Κεφάλαιο 2

Κινητικότητα στα Ασύρματα Δίκτυα Αισθητήρων

2.1 Γενικά	9
2.2 Μορφές κινητικότητας	10

2.1 Γενικά

Στις μέρες μας όλο και πιο σύγχρονοι κόμβοι αισθητήρων αναπτύσσονται. Όμως δεν είναι μόνο οι κόμβοι που αναπτύσσονται, αλλά και γενικά τα ασύρματα δίκτυα αισθητήρων. Πλέον, έχουμε την εισαγωγή της κινητικότητας μέσα σε αυτά τα δίκτυα, πράγμα που μας δίνει ακόμη περισσότερες δυνατότητες στα χέρια μας για να τις εκμεταλλευτούμε και να βάλουμε ακόμη περισσότερο στις ζωές μας τα WSNs. Έχουμε κόμβους τώρα να εφαρμόζονται πάνω σε ρομπότ, ζώα, αυτοκίνητα, ακόμη και ανθρώπους που σκοπό έχουν την καταγραφή διαφόρων περιβαλλοντικών αλλά και βιολογικών μεγεθών για παρατήρηση και μελέτη σε πολλούς τομείς.

Η κινητικότητα μπορεί να βοηθήσει σε πολλούς τομείς ένα ασύρματο δίκτυο αισθητήρων. Μπορεί να δώσει λύσεις στα προβλήματα κάλυψης ενός δικτύου καθώς και να προσφέρει πιο εύκολα κάλυψη σε ένα δίκτυο στο οποίο δεν υπάρχουν αρκετοί αισθητήρες για να καλύψουν στατικά ολόκληρο το χώρο που τους αναλογεί. Από την άλλη μπορούμε να συλλέξουμε πληροφορίες από απομακρυσμένους κόμβους με μια κινητή μονάδα συλλογής πληροφοριών αντί να αναγκάζουμε τους κόμβους να κάνουν μακρινές αποστολές δεδομένων με αποτέλεσμα να χάνουν άσκοπα πολύτιμη ενέργεια. Με την κίνηση ακόμη μπορούμε να ακολουθούμε ένα κινούμενο στόχο, από τον οποίο θέλουμε να συλλέξουμε πληροφορίες και όμως δεν μπορούμε να τον πλησιάσουμε για να του βάλουμε πάνω του αισθητήρες. Ακόμη μπορούμε να χρησιμοποιήσουμε ζώα σαν κινητήρια δύναμη, ενσωματώνοντας πάνω τους αισθητήρες για να μπορούμε να ελέγχουμε το φυσικό τους περιβάλλον, τα δάση για περιπτώσεις πυρκαγιάς καθώς και τα ίδια τα ζώα.

2.2 Μορφές κινητικότητας

Την κινητικότητα μπορούμε να την συναντήσουμε σε πολλές μορφές σε ένα ασύρματο δίκτυο αισθητήρων, όπως προαναφέρθηκε πιο πάνω. Σύμφωνα με το [11] μπορεί να έχουμε τρεις βασικές μορφές:

- **Node mobility:** σε αυτή την περίπτωση έχουμε τους κόμβους-αισθητήρα να είναι από μόνοι τους κινητοί. Εκτελούν αυτοί την κίνηση στο δίκτυο, αλλάζοντας την τοπολογία του πολύ συχνά. Ένα απλό παράδειγμα αυτής της μορφής κίνησης είναι όταν αντί να καλύπτουμε ένα χώρο με πάρα πολλούς κινητούς κόμβους, χρησιμοποιούμε αισθητά πιο λίγους κινητούς κόμβους, που με κάποιες προκαθορισμένες τροχιές, καλύπτουν πάλι ολόκληρο το χώρο.

- **Sink mobility:** αντί να έχουμε κίνηση από όλους τους κόμβους έχουμε κίνηση μόνο από τον sink, τον κόμβο δηλαδή που συλλέγει τα αποτελέσματα. Με αυτό τον τρόπο μπορούμε να εξοικονομήσουμε ενέργεια αφού δεν πρέπει έτσι να σπαταλούμε τόση ενέργεια στο να αποστέλλουμε σε μακρινές αποστάσεις τα μηνύματα απομακρυσμένων κόμβων, αλλά απλά να τα στέλνουμε μόνο όταν φτάσει το sink κοντά στον node. Αυτή η μορφή κινητικότητας μπορεί να έχει πολύ καλή εφαρμογή στην συλλογή πληροφοριών από κόμβους που έχουν αναπτυχθεί για παράδειγμα σε ένα δάσος για την παρακολούθηση διαφόρων ζώων ή την συλλογή πληροφοριών για το δάσος. Ένα ρομπότ, που ήδη γνωρίζει τις θέσεις των αισθητήρων θα στέλνεται για να συλλέξει ότι πληροφορία έχουν οι κόμβοι και θα επιστρέφει μετά στη βάση.

- **Event mobility:** η τρίτη και τελευταία βασική μορφή κινητικότητας σύμφωνα με το [2], σε ένα δίκτυο ασύρματων αισθητήρων είναι όταν μετακινείται το ίδιο το “πρόβλημα” που θέλουμε να παρακολουθήσουμε. Έχουμε για παράδειγμα ένα ζώο που θέλουμε να παρακολουθούμε συνέχεια σε μια μεγάλη περιοχή που καλύπτεται από πάρα πολλούς κόμβους. Το ζώο αυτό δεν είναι δυνατό να βρίσκεται σε ολόκληρη την περιοχή σε μια χρονική στιγμή. Ανά πάσα στιγμή μπορεί να κινηθεί και να βρεθεί σε άλλο σημείο. Έτσι ενεργεί είναι μόνο οι κόμβοι που βρίσκονται κοντά στο ζώο, αποστέλλοντας τις πληροφορίες που θέλουμε. Σε αντίθετη περίπτωση, όταν δηλαδή το ζώο αυτό φύγει από τον τομέα τους μπαίνουν σε κατάσταση ύπνου, εξοικονομώντας έτσι πολύτιμη ενέργεια. Ακόμη εκτός από την παρακολούθηση ζώων μπορεί να βρούμε παραδείγματα για αυτό το είδος κίνησης σε στρατιωτικές εφαρμογές παρακολούθησης. Βασικά όπου έχουμε ένα γεγονός που κινείται μέσα στο δίκτυο και θέλουμε να καταγράφουμε τις κινήσεις του και την συμπεριφορά του, μπορούμε να βρούμε και μια εφαρμογή με event mobility.

Με το να έχουμε κινητικότητα στα ασύρματα δίκτυα αισθητήρων μπορούμε να πετύχουμε πολλά πράγματα. Πράγματα που μας βοηθούν να βελτιώσουμε τόσο την επίδοση όσο και την αξιοπιστία και μακροζωία των δικτύων μας.

Μπορούμε εύκολα να μεγαλώσουμε το δίκτυο μας με την χρήση της κίνησης. Δεν είμαστε αναγκασμένοι να τοποθετούμε εμείς τους κόμβους αλλά από μόνοι τους πλέον μπορούν να μουν στις θέσεις που θέλουμε αφού πλέον έχουν τη δυνατότητα να κινηθούν από μόνοι τους. Ακόμη μπορεί να κάνουν και την αρχική τοποθέτηση σε ένα χώρο από μόνοι τους, για παράδειγμα αν τους ρίξουμε με αεροπλάνο σε ένα δάσος που καίγεται για να παίρνουμε στοιχεία για την πυρκαγιά, και βάζοντας τους δεδομένο τα όρια το δάσους θα μπορούν οι κόμβοι να ακροβολιστούν στο δάσος από μόνοι τους δίνοντας μας τις μετρήσεις που χρειαζόμαστε. Σε αυτό τον τομέα μπορούμε να προσθέσουμε την ικανότητα ενός δικτύου να μπορεί να επιδιορθωθεί από μόνο του σε περιπτώσεις νεκρώσεις κάποιων κόμβων του. Με την προσθήκη κάποιων επιπλέον κινητών κόμβων κατά τη δημιουργία του δικτύου, όταν εντοπιστεί μια τρύπα θα μπορούν με την κίνηση τους οι επιπλέον κόμβοι να την καλύπτουν για να μπορεί το δίκτυο να λειτουργεί χωρίς προβλήματα.

Ακόμη μπορούμε να καλύψουμε με λίγους κινητούς κόμβους μεγάλες περιοχές. Αν αυτές τις περιοχές τις καλύπταμε με τους συμβατικούς ακίνητους κόμβους τότε θα θέλαμε εξαιρετικά πιο μεγάλο αριθμό κόμβων από ότι με τους κινητούς. Καλύπτουμε βασικά τον πλήθος αυτών των κόμβων με την κίνηση που θα εκτελούν στο χώρο οι κινητοί.

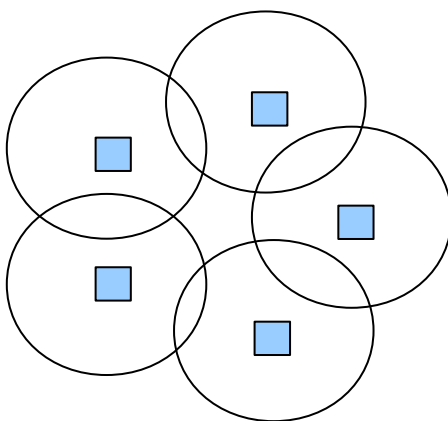
Κεφάλαιο 3

Πρωτόκολλα κινητικότητας

3.1 Αλγόριθμοι κάλυψης περιοχής	12
3.2 Αλγόριθμοι ελέγχου περιμέτρου	17
3.3 Αλγόριθμοι ομαδοποίησης	19

3.1 Αλγόριθμοι κάλυψης περιοχής

Ένα ασύρματο δίκτυο αισθητήρων μπορεί να παρατηρηθεί το πρόβλημα των “τρυπών” κάλυψης της περιοχής τους, της ύπαρξης δηλαδή περιοχών εντός του δικτύου όπου κανένας από τους κόμβους-αισθητήρες δεν μπορεί να συλλέξει πληροφορίες. Αυτές οι τρύπες μπορεί να επέλθουν από την τυχαία εναέρια τοποθέτηση των αισθητήρων, από την ύπαρξη εμποδίων στην περιοχή καθώς και από το θάνατο ενός ή περισσότερων αισθητήρων του δικτύου.



Σχήμα 3.1: Παράδειγμα τρύπας σε δίκτυο πέντε κόμβων-αισθητήρων

Οι σημαντικότεροι αλγόριθμοι κάλυψης περιοχής που υπάρχουν σήμερα χωρίζονται σε τριών ειδών κατηγορίες, σύμφωνα με το σε ποιο είδος WSN εφαρμόζονται. Σε στατικά, σε κινητά ή υβριδικά WSNs (ύπαρξη και κινητών και στατικών κόμβων αισθητήρων). Οι πιο σημαντικοί από αυτούς τους αλγόριθμους θα παρουσιαστούν πιο κάτω.

3.1.1 Αλγόριθμοι σε στατικά δίκτυα αισθητήρων

Είναι οι αλγόριθμοι που εφαρμόζονται σε WSNs όπου έχουμε μόνο στατικούς κόμβους αισθητήρων.

- **Coverage configuration protocol (CCP)**

Αλγόριθμος που ρυθμίζει αυτόματα το βαθμό κάλυψης, αφού παρέχει ελαστικότητα στη ρύθμιση του WSN [20]. Όταν $R_c \geq R_s$ (όπου R_c το εύρος επικοινωνίας και R_s το εύρος αίσθησης) κάθε κόμβος τρέχει τον K_s αλγόριθμο κάλυψης. Σαν δεδομένο έχουμε ότι ζητείται κάλυψη βαθμού K_s . Αν κάθε τοποθεσία μέσα στο R_s κάποιου κόμβου, καλύπτεται από άλλους γειτονικούς κόμβους οι οποίοι είναι ήδη ενεργοί, ο συγκεκριμένος κόμβος προγραμματίζεται να κοιμάται. Σε περιπτώσεις που έχουμε ότι $R_c < R_s$ έχουμε την εισαγωγή του SPAN αφού ο CCP δεν εγγυάται την συνδεσιμότητα μαζί με την κάλυψη. Το SPAN είναι ένας κατανεμημένος αλγόριθμος που συντονίζει multi-hop ad-hoc WSNs και χωρίς να μειώνει σημαντικά τη χωρητικότητα ή τη συνδεσιμότητα του δικτύου, μειώνει την κατανάλωση ενέργειας. Είναι βασικά μια προσθήκη στον CCP έτσι ώστε να καλύπτονται όλες οι πιθανές περιπτώσεις που διέπουν τη σχέση μεταξύ R_c και R_s .

- **K-UC & K-NC**

Αλγόριθμοι που αναπτύσσονται στο [4], οι οποίοι έχουν πολυωνυμικό χρόνο για να ελέγχουν αν υπάρχει ο ελάχιστος αριθμός κόμβων αισθητήρων για να καλύψει κάθε κομμάτι της περιοχής του δικτύου. Και οι δύο αλγόριθμοι χρειάζονται τις πληροφορίες του που βρίσκεται ο κάθε κόμβος αισθητήρα για να μπορεί να υπολογίσει το αν υπάρχει η επιθυμητή κάλυψη. Ο K-UC υποθέτει ότι κάθε κόμβος έχει ένα ομοιόμορφο δίσκο αίσθησης σε αντίθεση με τον K-NC, ο οποίος υποθέτει ότι κάθε κόμβος έχει δικό του ξεχωριστό και μη ομοιόμορφο εύρος αίσθησης. Στον K-UC, θα υπολογιστεί από ένα κόμβο η κάλυψη ενός γειτονικού κόμβου μόνο αν ο γείτονας αυτός βρίσκεται μέσα σε μια περιοχή η οποία δεν υπερβαίνει τις δύο φορές το εύρος αίσθησης του κόμβου. Από την άλλη στο K-NC βλέπουμε να λαμβάνονται υπόψη διαφορετικοί κανόνες για το πότε ένας κόμβος βρίσκεται μέσα στο εύρος αίσθησης κάποιου δεύτερου κόμβου και ως εκ τούτου διαφορετικοί κανόνες και για το πότε ένας κόμβος πρέπει να καλύψει την κάλυψη των γειτονικών του κόμβων αισθητήρων.

Μετά έχουμε και στις δύο περιπτώσεις υπολογισμό της κάλυψης της περιμέτρου του κάθε κόμβου από το εύρος αίσθησης των γειτονικών του κόμβων. Έτσι ο κάθε κόμβος μπορεί να

ελέγξει να υπάρχει η απαιτούμενη κάλυψη στην περίμετρό του. Μια κεντρική οντότητα μαζεύει τις πληροφορίες για τις τρύπες κάλυψης από το δίκτυο για να μπορεί μετά να στείλει κόμβους όπου χρειάζεται επιπλέον κάλυψη για να καλυφτούν οι τρύπες αυτές.

- **Differentiated**

Αρχικά σε αυτό τον αλγόριθμο έχουμε τους κόμβους να παίρνουν τη θέση τους και να κάνουν συγχρονισμό με τους γείτονές τους [18]. Αυτό αποτελεί την φάση της αρχικοποίησης. Μετά ακολουθεί η φάση της αίσθησης, η οποία επαναλαμβάνεται πολλές φορές σε ίσα χρονικά διαστήματα. Εδώ κάθε κόμβος αισθητήρα χωρίζει την περιοχή του σε grids και μετά διαφημίζει το σημείο αναφοράς του με τους χρόνους αρχής και τέλους αυτού του σημείου. Έτσι κάθε κόμβος ταξινομεί τους γείτονες που καλύπτουν ένα συγκεκριμένο grid, με βάση το σημείο αναφοράς τους σε ένα γύρο σε αύξουσα σειρά. Με βάση αυτή την ταξινόμηση ο κάθε κόμβος μπορεί να υπολογίσει το πόσο πρέπει να είναι ενεργός έτσι ώστε να μπορεί ολόκληρο το grid να καλύπτεται στον επιθυμητό βαθμό.

3.1.2 Αλγόριθμοι σε κινητά δίκτυα αισθητήρων

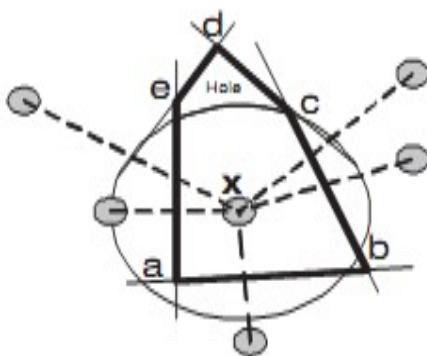
Είναι οι αλγόριθμοι που εφαρμόζονται σε WSNs όπου έχουμε μόνο κινητούς κόμβους αισθητήρων.

- **Χρήση διαγραμμάτων Voronoi**

Για να ανακαλύψουμε την ύπαρξη τρυπών σε ένα WSN με κινητούς κόμβους, κάθε κόμβος, με τη χρήση κάποιας τεχνικής καθορισμού τοποθεσίας, πρέπει να ξέρει που βρίσκονται οι γείτονες του έτσι ώστε να μπορεί να δημιουργήσει το δικό του διάγραμμα Voronoi. Έτσι μετά ολόκληρη η περιοχή του WSN χωρίζεται σε πολύγωνα Voronoi, με το κάθε ένα από αυτά να έχει ένα μόνο κόμβο αισθητήρα. Αυτός ο αισθητήρας βρίσκεται πιο κοντά σε κάθε σημείο του πολυγώνου από κάθε άλλο αισθητήρα. Μετά ο κάθε κόμβος ξεχωριστά συγκρίνει το πεδίο στο οποίο μπορεί να αισθανθεί με την περιοχή που καταλαμβάνει το πολύγωνο Voronoi έτσι ώστε να υπολογίσει τις τοπικές τρύπες που μπορεί να υπάρξουν. Για να καλυφθούν αυτές οι τρύπες έχουν προταθεί τρεις διαφορετικοί αλγόριθμοι στο [8].

Vector Based (VEC): Ένας κόμβος προσπαθεί να κρατήσει την απόσταση του από κάθε κόμβο ίση με μια μέση απόσταση που έχει με αυτό τον κόμβο και αφού την έχει υπολογίσει πριν υποθέτοντας ότι όλοι οι κόμβοι είναι ομοιόμορφα κατανεμημένοι μέσα στο δίκτυο.

Minmax: Παρόμοιος σε λειτουργία με τον VOC. Εκτός όμως από την κίνηση του κόμβου προς την κορυφή του δικού του πολυγώνου Voronoi, λαμβάνει υπόψη και τις αποστάσεις από τις άλλες κορυφές για να βρει τη θέση μέσα στο πολύγωνο η οποία να ελαχιστοποιεί την απόσταση από την μακρινότερη κορυφή. Έτσι δεν χρησιμοποιεί την άπληστη λογική του VOC και έχει καλύτερα αποτελέσματα αν και έχει περισσότερους υπολογισμούς.



Σχήμα 3.2: Χρήση διαγραμμάτων Voronoi για ανίχνευση τρυπών[8]

Αλγόριθμος ο οποίος ασχολείται με την κάλυψη τρυπών κάλυψης που δημιουργούνται από την εξάντληση της ενέργειας των κόμβων αισθητήρων στο δίκτυο [17]. Αρχικά κάθε κόμβος μαθαίνει τους γείτονες του και υπολογίζει την περιοχή κάλυψής του. Αν κάποιος κόμβος αισθητήρα έχει χαμηλά αποθέματα ενέργειας, στέλνει ένα Panic Request (μήνυμα πανικού). Οι γειτονικοί του κόμβοι του στέλνουν Panic Reply, για το αν μπορούν να μετακινηθούν στην περιοχή χωρίς να χάσουν την υπάρχουσα κάλυψή τους, μαζί με το ενεργειακό τους απόθεμα και μικρότερη απαιτούμενη απόσταση που θα διανύσουν για να φτάσουν στον τελικό τους προορισμό. Μετά ο κόμβος που «πεθαίνει» επιλέγει τον πιο κατάλληλο κόμβο και τον ειδοποιεί να προχωρήσει με την κίνησή του.

15

- **Distributed Self-Spreading algorithm (DSSA)**

Εμπνευσμένος από την ισορροπία των μορίων. Εκτελείται σε κάθε κόμβο του δικτύου, οι οποίοι αρχικά είναι διασκορπισμένοι τυχαία και κινούνται στη συνέχεια με βάση εικονικές δυνάμεις που ασκούνται πάνω τους από τους γείτονές τους. Αυτές οι δυνάμεις εξαρτώνται από την απόσταση τους από τους γείτονές τους καθώς και από την τοπική πυκνότητα των αισθητήρων [13].

- **Intelligent Deployment and Clustering Algorithm (IDCA)**

Ο αλγόριθμος αυτός συγκρίνει την τοπική πυκνότητα των κόμβων με την πυκνότητα που θα εμφανιζόταν στο δίκτυο αν αναπτυσσόταν ομοιόμορφα [13]. Με τα αποτελέσματα που θα έχει μετά, ένας κόμβος αισθητήρα θα επιλέξει την κατάλληλη μέθοδο clustering. Έτσι ένας κόμβος πρέπει να κινηθεί ή όχι, βάση της σχετικής υπόλοιπης ενέργειας των αισθητήρων, με απώτερο σκοπό με την ομοιόμορφη διανομή των κόμβων στο δίκτυο να μειωθεί η διαφορά στην υπολειπόμενη ενέργεια.

- **Potential Fields**

Εδώ έχουμε την προϋπόθεση ότι κάθε αισθητήρας μπορεί να καθορίσει την απόσταση και την κατεύθυνση προς όλους τους κόμβους αισθητήρων καθώς και τα εμπόδια που βρίσκονται στο χώρο του δικτύου. Με αυτά ο κάθε κόμβος, συμπεριφερόμενος πάλι σαν εικονικό μόριο, μπορεί να καθορίσει τις εικονικές δυνάμεις που ασκούνται πάνω του και να τις μετατρέψει σε ένα πίνακα ελέγχου. Αυτό τον πίνακα θα τον χρησιμοποιεί για να κινηθεί μέσα στον χώρο του δικτύου. Έτσι δεν χρειάζεται να έχουμε κάποια κεντρική μονάδα ελέγχου και συντονισμού στο δίκτυο που να υποδεικνύει τις κινήσεις των κόμβων [3].

3.1.2 Αλγόριθμοι σε υβριδικά δίκτυα αισθητήρων

Είναι οι αλγόριθμοι που εφαρμόζονται σε WSNs όπου έχουμε και κινητούς και σταθερούς κόμβους αισθητήρων.

- **Bidding Protocol**

Οι σταθεροί κόμβοι υπολογίζουν τις τοπικές τρύπες κάλυψης χρησιμοποιώντας διαγράμματα Voronoi σε αντίθεση με τους κινητούς κόμβους αισθητήρων που υπολογίζουν τις τρύπες κάλυψης μόνο όταν αποφασίσουν να κινηθούν. Υπάρχει μια μορφή πλειοδότηση μεταξύ των κινητών και των σταθερών κόμβων. Οι σταθεροί στέλνουν με μηνύματα broadcast σε

απόσταση μέχρι και δύο βήματα (hops) το μέγεθος των τοπικών τους τρυπών κάλυψης στους κινητούς κόμβους που βρίσκονται στις μακρινότερες περιοχές των πολυγώνων Voronoi τους. Μετά οι κινητοί κόμβοι, με βάση την μεγαλύτερη πλειοδότηση που έχουν παραλαβή και σε συνάρτηση του αν η τρύπα που θα δημιουργήσουν με το να μετακινηθούν από την τωρινή τους θέση είναι μεγαλύτερη ή όχι από αυτή που θα καλυφθεί, αποφασίζουν αν θα κινηθούν αναλόγως [7].

- **COVEN**

Ακολουθείται μια στρατηγική δύο φάσεων εδώ, σύμφωνα με το [6]. Υπάρχει και η προϋπόθεση ότι κάθε κόμβος γνωρίζει τη θέση του. Στην πρώτη φάση κατασκευάζεται ένα διάγραμμα Voronoi. Οι κόμβοι που βρίσκονται κοντά στα όρια αίσθησης του δικτύου, βρίσκουν το που είναι, i αυτά θεωρώντας τα άκρα του Voronoi πολυγώνου τους σαν τα τη γραμμή των ορίων του δικτύου. Στη δεύτερη φάση απλά υπολογίζεται το πόσοι κόμβοι χρειάζονται επιπλέον για να καλυφθεί ικανοποιητικά το δίκτυο. Αν έχουμε αραιή αρχική τοποθέτηση των κόμβων, τότε μετά την ανάπτυξη των επιπλέον κόμβων, επιλέγονται μερικοί από αυτούς για να εντοπίσουν απομονωμένους κόμβους στο δίκτυο, με το να κινηθούν. Μετά τον εντοπισμό αυτών των κόμβων αισθητήρων, ζητάτε από την εφαρμογή η ανάπτυξη και άλλων κόμβων στις περιοχές των απομονωμένων.

3.2 Αλγόριθμοι ελέγχου περιμέτρου

Όσον αφορά αυτή την περιοχή, δεν υπάρχουν πολλά υλοποιημένα πρωτόκολλα για ακίνητους αισθητήρες. Όμως συστήματα πολλών ρομπότ ή έξυπνων πρακτόρων που συνεργάζονται για να περιπολήσουν μια καθορισμένη περιοχή έχουν μελετηθεί με διάφορες οπτικές γωνίες. Έχουν δοθεί θεωρητικές και εμπειρικές λύσεις για να διασφαλίσουν την ποιότητα αυτής της περιπολίας. Όμως ο ορισμός της ποιότητας εξαρτάτε από πολλά πράγματα. Οι περισσότερες μελέτες επικεντρώνονται στη συχνότητα των επισκέψεων στην περιοχή, έτσι ένα αποδοτικό πρωτόκολλο θα μπορούσε να ήταν ένα που εγγυάται μεγάλη συχνότητα επισκέψεων από τα ρομπότ σε κάθε μέρος της ελεγχόμενης περιοχής. Στην περίπτωση που τα ρομπότ δουλεύουν σε ένα περιβάλλον που πρέπει να το προστατεύσουν από εισβολείς, αποδοτικότητα ορίζεται η ικανότητα της περιπόλου να χειρίζεται αυτούς τους εισβολείς. Πιο κάτω θα δούμε μερικές χαρακτηριστικές εφαρμογές που έχουν προταθεί.

Στο [14] και το [15] έχουμε μια προσπάθεια της τοποθέτησης σημείων ελέγχου ασφάλειας. Έχουμε την υπόθεση σε αυτά τα δύο άρθρα ότι οι πράκτορες εργάζονται σε ένα περιβάλλον

όπου ο αντίπαλος μπορεί να ερευνήσει κάθε πιθανή συμπεριφορά της περιπόλου. Στο [15] οι πράκτορες που έχουν κάνουν χρήση πολιτικής με πολλές τυχαίες τοποθετήσεις και κινήσεις για να μεγαλώσουν την πιθανότητα να βρουν τους αντιπάλους. Όμως με αυτή την πολιτική, ακόμη και με ένα πράκτορα, το πρόβλημα που περιγράφουν μπορεί να λυθεί σε εκθετικό χρόνο, αφού παραθέτουν ευρετικούς αλγορίθμους και για τις περιπτώσεις με ένα ή πολλαπλούς πράκτορες. Από την άλλη στο [14] έχουμε την επιπλέον μελέτη του προβλήματος τοποθέτησης σημείων ελέγχου στην περίπτωση που η συμπεριφορά των αντιπάλων κόμβων είναι άγνωστη, παραθέτοντας πάλι ευρετικούς αλγορίθμους για τη βέλτιστη επιλογή στρατηγικής από τους πράκτορες που περιπολούν την περιοχή.

Στο [10] έχουμε μια θεωρητική δουλειά από τους Shieh και Calvert, βασισμένη στις λύσεις που δίνει η υπολογιστική γεωμετρία. Με αυτή τη δουλειά προσπαθούν να βρουν τα βέλτιστα σημεία για έλεγχο για τα ρομπότ που κάνουν περιπολία σε μια περιοχή. Προσπαθούν να μεγιστοποιήσουν το πόσο μπορούν να δουν τα ρομπότ στην περιοχή που περιπολούν και βρίσκουν αλγόριθμους που λύνουν κατά προσέγγιση το πρόβλημα, αφού πρώτα δείχνουν πως είναι πρόβλημα δυσκολίας NP-Hard, δηλαδή δεν έχει λύση.

Η πρώτη θεωρητική ανάλυση μια περιπόλου με πολλά ρομπότ δύνεται στο [21]. Μας εισαγάγει του idleness, η οποία είναι η διάρκεια στην οποία κάθε σημείο στην περιπολούμενη περιοχή δεν έχει επισκεφτεί. Μετά, βάση αυτής της έννοιας, αναλύονται δύο τύποι σχεδίων περιπολίας με πολλαπλά ρομπότ. Η περιπολία της περιοχής σε υποπεριοχές, με κάθε περιοχή να δέχεται επανειλημμένα επισκέψεις από ένα ρομπότ και η κυκλική περιπολία, όπου δίνεται μια διαδρομή σε ολόκληρη την περιοχή και όλα τα ρομπότ επισκέπτονται όλα τα σημεία της περιοχής με σειρά. Μετά μας αποδεικνύεται ότι στη δεύτερη προσέγγιση, η συχνότητα των επισκέψεων σε κάθε σημείο στην περιοχή είναι αρκετά μεγαλύτερη. Αυτή η διαπίστωση, για το ότι στην κυκλική προσέγγιση έχουμε καλύτερα πλεονεκτήματα γίνεται και στο [1], με βάση το κριτήριο του idleness.

Τέλος στο [22] μας προσφέρετε ένα νέο κριτήριο βελτιστοποίησης κριτηρίων για αξιολόγηση αλγορίθμων περιπολίας. Μας παρουσιάζετε ένας αλγόριθμος στον οποίο το κάθε σημείο της περιπολούμενής περιοχής επισκέπτεται από τα ρομπότ με την ίδια βέλτιστη συχνότητα. Η δουλειά αυτή βασίζεται στη δημιουργία ενός κύκλου περιπολίας ο οποίος επισκέπτεται όλα τα σημεία της περιοχής στον ελάχιστο χρόνο και τα ρομπότ απλά κινούνται ίσες αποστάσεις σε αυτό τον κύκλο περιπολίας.

3.3 Αλγόριθμοι ομαδοποίησης

Το να ομαδοποιήσουμε ομάδες κόμβων είναι ένα από τα πιο σημαντικά προβλήματα που έχουμε. Ένας απλός ορισμός του clustering θα μπορούσε να είναι: η διαδικασία της οργάνωσης διάφορων αντικειμένων σε ομάδες, των οποίων τα μέλη είναι κοινά με κάποιο τρόπο. Στα WSNs, με αυτή την ομαδοποίηση στοχεύουμε στην μείωση της κατανάλωσης ενέργειας. Αυτό γιατί με την ομαδοποίηση των αισθητήρων, μια ομάδα θα στέλνει στον συντονιστή της και αυτό θα είναι υπεύθυνος για την παραπέρα διάδοση των δεδομένων τους. Έχουν προταθεί πολλών ειδών αλγόριθμοι, με τους περισσότερους να είναι Heuristics και να στοχεύουν στην δημιουργία του μικρότερου αριθμού από ομάδες έτσι ώστε κάθε κόμβος να είναι το πολύ d βήματα μακριά από τον επικεφαλής της ομάδας. Οι περισσότεροι από αυτούς έχουν πολυπλοκότητα $O(n)$, με n τον αριθμό των κόμβων. Ακόμη αρκετοί από αυτούς απαιτούν συγχρονισμό μεταξύ των αισθητήρων, πράγμα που τους κάνει κατάλληλους μόνο για δίκτυα με μικρό αριθμό κόμβων.

Πιο κάτω θα μελετήσουμε μερικούς από αυτούς τους αλγορίθμους ομαδοποίησης(clustering algorithms).

- **Linked Cluster Algorithm(LCA & LCA2)**

Σε αυτόν τον αλγόριθμο (LCA) [5] ένας κόμβος γίνεται ο επικεφαλής της ομάδας εάν έχει την μεγαλύτερη τιμή για ταυτότητα από τους γείτονες του που βρίσκονται ένα Hop μακριά του. Αυτός ο αλγόριθμος βελτιώνεται από τον LCA2 [2], όπου σε αυτόν διαλέγεται σαν επικεφαλής ο κόμβος με το μικρότερο Id από όλους τους κόμβους, που δεν είναι ούτε επικεφαλείς ούτε απέχουν ένα βήμα μακριά από άλλους επικεφαλείς.

- **Weighted Clustering Algorithm (WCA)**

Ο WCA επιλέγει ένα κόμβο σαν επικεφαλής ομάδας βάση στον αριθμό των γειτόνων, την ενέργεια μετάδοσης, την μπαταρία και τον βαθμό κινητικότητας του κόμβου. Επίσης περιορίζει τον αριθμό των κόμβων σε ένα cluster έτσι ώστε η απόδοση των πρωτοκόλλων MAC να μην υποβαθμίζεται [12].

- **Distributed Clustering Algorithm (DCA)**

Αυτό ο αλγόριθμος χρησιμοποιεί βάρη για να διαλέξει τους επικεφαλείς κάθε ομάδας [16]. Αυτά τα βάρη είναι γενικά και μπορούν να προσδιοριστούν ανάλογα με την εφαρμογή. Διαλέγει τον κόμβο με το ψηλότερο βάρος μεταξύ των γειτόνων με απόσταση ένα βήμα σαν

αρχηγό της ομάδας αυτής. Είναι κατάλληλος ο DCA για δίκτυα στα οποία οι κόμβοι είναι στατικοί ή κινούνται με μικρές ταχύτητες. Ο DMAC (Distributed and mobility-adaptive clustering) τροποποιεί τον DCA για να επιτρέπει την κίνηση των κόμβων, στη διάρκεια ή μετά την φάση της δημιουργίας των clusters.

- **Algorithm for Cluster Establishment (ACE)**

Σε αυτό τον αλγόριθμο, που αναπτύσσεται στο [9], έχουμε ομαδοποίηση των κόμβων μέσα σε ένα σταθερό αριθμό επαναλήψεων, χρησιμοποιώντας τον βαθμό του κόμβου σαν κύρια παράμετρο. Μια από τις αδυναμίες του ACE είναι ότι διαλέγει τυχαία τον υποψήφιο κόμβο σε κάθε επανάληψη, γεγονός το οποίο δημιουργεί διαφορετικά αποτελέσματα κάθε φορά στο ίδιο δίκτυο. Επίσης χρησιμοποιώντας δύο παραμέτρους που προσαρμόζονται από τον χρήστη ελέγχει τον σχηματισμό καινούριων ομάδων, με αποτέλεσμα η απόδοσή του να βασίζεται σε αυτές, που καθορίζονται βάση του μεγέθους και του σχήματος του δικτύου αισθητήρων.

- **Low Energy Adaptive Clustering Hierarchy (LEACH)**

Στόχος του είναι να παρέχει συνάθροιση πληροφοριών για δίκτυα αισθητήρων με το να παρέχει επικοινωνία που να είναι ενεργειακά επαρκής χωρίς να αγνοεί μερικούς κόμβους περισσότερο από άλλους. Σε κάθε κύκλο διαλέγονται διαφορετικοί επικεφαλείς για τις ομάδες που υπάρχουν. Ο κάθε κόμβος τρέχει ένα στοχαστικό αλγόριθμο για να αποφασίσει αν θα πρέπει να γίνει επικεφαλής σε κάθε συγκεκριμένο κύκλο. Οι κόμβοι που ήδη ήταν επικεφαλείς δεν μπορούν να ξαναγίνουν για P χρόνο, όπου P το επιθυμητό ποσοστό αρχηγών στο δίκτυο. Στο τέλος κάθε γύρου όσοι δεν είναι επικεφαλείς, διαλέγουν τη πιο κοντινή ομάδα και μπαίνουν σε αυτή. Έπειτα οι επικεφαλείς των ομάδων διαμορφώνουν ένα πρόγραμμα για το πώς κάθε κόμβος στην ομάδα τους θα μεταδώσει τα δεδομένα του [19].

Κεφάλαιο 4

Προβλήματα προς επίλυση

4.1 Γενικά	21
4.2 Έλεγχος-Περιπολία περιοχής	24
4.3 Ομαδοποίηση κόμβων σε κινητό ασύρματο δίκτυο	26
4.4 Αποτελέσματα-Ανάλυση αποτελεσμάτων	33

4.1 Γενικά

Σε αυτό το κεφάλαιο θα αναπτυχθούν τα τρία προβλήματα με τα οποία ασχολήθηκα στην ατομική διπλωματική μου εργασία. Με λίγα λόγια είχα να υλοποιήσω ένα αλγόριθμο για κάθε μια από τις περιοχές που μελέτησα καθ' όλη τη διάρκεια της εργασίας μου. Πιο συγκεκριμένα, έπρεπε να αναπτύξω μια εφαρμογή για παρακολούθηση περιμέτρου από αριθμό κόμβων με τυχαία αρχική τοποθέτηση σε μια περιοχή, μια για την ομαδοποίηση κόμβων σε ένα δίκτυο και μια για αναγνώριση από δύο ομάδες κόμβων των κόμβων-συμμάχων τους και ομαδοποίηση τους κατά την τυχαία τους κίνηση σε ένα χώρο. Αν και μελετήθηκε και το θέμα της κάλυψης μιας περιοχής με κίνηση σε ασύρματα δίκτυα αισθητήρων, εν τούτοις δεν αναπτύχθηκε κάποιος αλγόριθμος.

Και οι τρεις εφαρμογές έχουν αναπτυχθεί σε γραφικό περιβάλλον, σε γλώσσα προγραμματισμού C, με την ενσωμάτωση της βιβλιοθήκης GP142, η οποία ενσωματώνει στη γλώσσα C συναρτήσεις για την ανάπτυξη γραφικών. Περισσότερες πληροφορίες τόσο για την χρήση, όσο και για τις δυνατότητες της βιβλιοθήκης αυτής, μπορεί κάποιος να πάρει από το εγχειρίδιο χρήσης της βιβλιοθήκης όπως και πιο κάτω. Ο γραφικός χώρος ο οποίος χρησιμοποιείται στις εφαρμογές είναι πάντα ένα τετράγωνο, για πιο εύκολη κατανόηση του αλγορίθμου και για εύκολη αναγνώριση των γνωρισμάτων του προβλήματος κάθε φορά.

Οι εφαρμογές όλες είναι σχεδιασμένες και δοκιμασμένες σε λειτουργικό LINUX (Ubuntu 10.4), με τη χρήση του editor KATE. Για να μπορέσουν να τρέξουν οι εφαρμογές πρέπει

πρώτα να είμαστε σίγουροι πως έχουμε εγκαταστήσει κάποια πακέτα εφαρμογών και πως έχουμε όλα τα απαραίτητα αρχεία στον φάκελο.

Πρώτα από όλα πρέπει να βεβαιωθούμε ότι έχουμε εγκαταστήσει το Xlib.h για να μπορούμε να χρησιμοποιήσουμε τις εφαρμογές. Για να γίνει αυτό, ανοίγουμε ένα τερματικό και εκτελούμε την ακόλουθη εντολή

- **sudo apt-get install libx11-dev**

Φυσικά για να μπορούμε να την εκτελέσουμε θα πρέπει να έχουμε δικαιώματα χρήστη πάνω στην μηχανή Linux που θα χρησιμοποιήσουμε.

Μετά θα πρέπει να βεβαιωθούμε ότι υπάρχουν εκτός από τα αρχεία με τους κώδικες, που υπάρχουν πάνω στο CD της διπλωματικής αυτής εργασίας, και τα ακόλουθα τρία αρχεία, που περιέχουν τις συναρτήσεις για την βιβλιοθήκη GP142 που χρησιμοποιήθηκε για την ανάπτυξη του γραφικού περιβάλλοντος. Αυτά είναι τα ακόλουθα:

- gp142.c
- gp142.h
- gp142lib.h

Ακολούθως για να κάνουμε compile τους κώδικες χρειαζόμαστε να εκτελέσουμε την εντολή:

- `gcc -o <executable> gp142.c <source_file> -g -L/usr/X11R6/lib -lX11 -lc -lm`

Στη θέση του <executable>, βάζουμε το όνομα του εκτελέσιμου αρχείου που θα δημιουργηθεί και στη θέση του <source_file> το όνομα του κώδικα που θέλουμε να τρέξουμε κάθε φορά. Τέλος για να τρέξουμε το πρόγραμμα τρέχουμε την εντολή

- `./<executable>`

Μετά στο τερματικό θα μας ζητά να δώσουμε μερικά δεδομένα για να ξεκινήσει το πρόγραμμα. Απλά δακτυλογραφούμε κάθε φορά το ζητούμενο και πατάμε στη συνέχεια enter για να εισάγουμε τα δεδομένα. Βλέπουμε και στο Σχήμα 4.1 ένα παράδειγμα για την εφαρμογή της κάλυψης περιμέτρου από το τερματικό.

```
File Edit View Terminal Help
paris@paris-desktop:~/Desktop/diplomatiki$ gcc -o perimeter gp142.c perimeter.c
-g -L/usr/X11R6/lib -lX11 -lc -lm
paris@paris-desktop:~/Desktop/diplomatiki$ ./perimeter
Give the number of sensors you want to start with
3
Give the number of rows and columns you want to start with(the area is square s
o give one value)
█
```

Σχήμα 4.1: Παράδειγμα όψης τερματικού για την εφαρμογή του ελέγχου περιμέτρου

Animate	Recording off
Stop	Playback off
SingleStep	SingleScript
Logging off	Quit

Σχήμα 4.2: Το παράθυρο με τα κουμπιά ελέγχου του gp142

Μόλις εισαγάγουμε όλα τα στοιχεία θα ανοίξουν δύο παράθυρα στον υπολογιστή μας. Ένα με το πλέγμα και τους κόμβους στην αρχική τους θέση και ένα με τα κουμπιά ελέγχου του gp142. Βασικό είναι να ξέρουμε τι κάνουν κάποια από τα κουμπιά του παραθύρου ελέγχου.

Όπως βλέπουμε και στο Σχήμα 4.2, έχουμε οκτώ κουμπιά που έχουν τις εξής λειτουργίες:

- **Animate:** ξεκινά την προσομοίωση της εφαρμογής
- **Stop:** σταματά την προσομοίωση
- **SingleStep:** προχωρά την προσομοίωση κατά ένα στιγμιότυπο μόνο
- **Logging off/on:** Με το να ενεργοποιήσουμε το κουμπί αυτό μπορούμε να παίρνουμε δεδομένα σχετικά με τις κλήσεις όλων των συναρτήσεων της βιβλιοθήκης gp142 πάνω στο τερματικό. Είναι χρήσιμο μόνο για την αποσφαλμάτωση του κώδικα.
- **Recording off/on:** με αυτό το κουμπί μπορούμε να αποθηκεύουμε μια υποακολουθεία στιγμιοτύπων, όταν είναι ενεργοποιημένο στο αρχείο gp142.scr
- **Playback off/on:** αναπαράγουμε ότι υπάρχει στο αρχείο gp142.scr
- **SingleScript:** αναπαράγουμε το επόμενο στιγμιότυπο από το αρχείο gp142.scr
- **Quit:** κλείνουμε τα παράθυρα του gp142.

4.2 Έλεγχος-Περιπολία περιοχής

4.2.1 Γενικά για το πρόβλημα

Για αυτή την εφαρμογή, το γενικό σενάριο είναι ότι έχουμε ένα χώρο τον οποίο θέλουμε να βάλουμε κινητούς κόμβους-αισθητήρες, με σκοπό τον περιμετρικό έλεγχο της. Αρχικά οι κόμβοι αναπτύσσονται τυχαία μέσα στον χώρο και σκοπός τους είναι να βρουν το κοντινότερο σημείο στο οποίο πρέπει να πάει ο κάθε ένας, έτσι ώστε με το που θα ξεκινήσει η κίνησή τους να ελέγχουν με τον καλύτερο δυνατό τρόπο την περιοχή στην οποία έχουν αναπτυχθεί. Βασική προϋπόθεση είναι ότι αρχικά οι κόμβοι γνωρίζουν τα όρια τις περιοχής καθώς επίσης και το πόσοι επιπλέον κόμβοι υπάρχουν στο σύνολό τους ανεπτυγμένοι. Για τους σκοπούς των παραδειγμάτων στα σχήματα που ακολουθούν χρησιμοποιήθηκαν τρεις κόμβοι και σαν μέγεθος πλευράς του πλέγματος είκοσι τετράγωνα.

4.2.2 Αλγόριθμος

Αρχικά γίνεται από τον χρήστη η εισαγωγή των επιθυμητών μεταβλητών για το πρόβλημα. Συγκεκριμένα εισάγεται ο αριθμός των κόμβων που θα χρησιμοποιηθεί κάθε φορά καθώς επίσης και το μέγεθος της περιοχής. Όσον αφορά το τελευταίο, χρειάζεται μόνο η εισαγωγή μιας πλευράς τετραγώνου, αφού όπως έχει προαναφερθεί χρησιμοποιείται τετράγωνος γραφικός χώρος. Με βάση τον αριθμό των κόμβων και την περίμετρο του χώρου, γίνεται ο υπολογισμός των αρχικών θέσεων στις οποίες πρέπει να πάνε οι κόμβοι για να ξεκινήσουν την κίνησή τους. Ο κάθε κόμβος προχωρά και λαμβάνει τη θέση του και μετά όλοι μαζί ξεκινούν την περιμετρική περιπολία του χώρου.

Αλγοριθμικά Βήματα

Διάβασε τα δεδομένα από τον χρήστη

Τοποθέτησε τυχαία τους κόμβους στον χώρο

Βάση του αριθμού των κόμβων και της περιμέτρου

Υπολόγισε τις τελικές θέσεις των κόμβων στην περίμετρος

Βάση των τελικών θέσεων

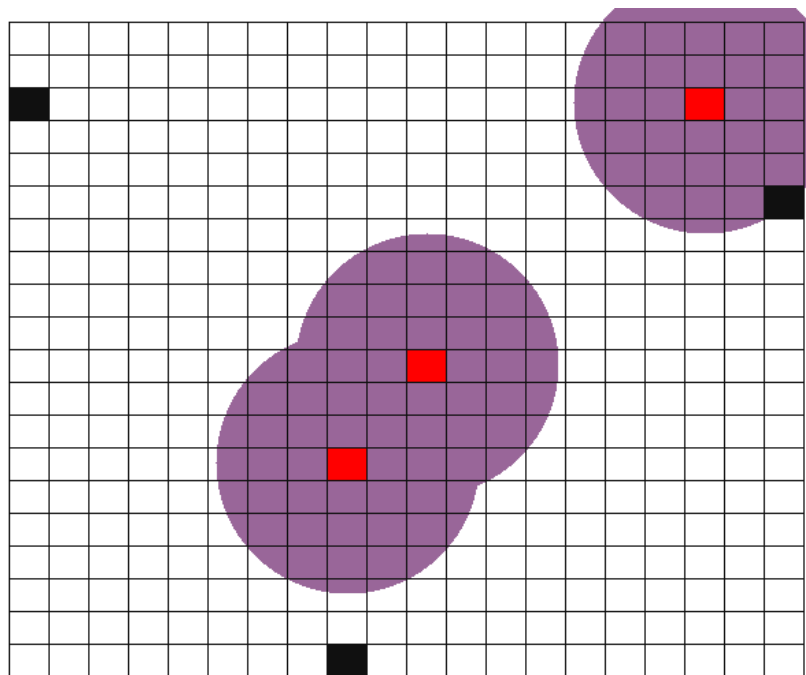
Υπολόγισε ποιος κόμβος πάει σε κάθε θέση

Ξεκίνα την κίνηση των κόμβων προς τις τελικές τους θέσεις

Μόλις φτάσουν όλοι οι κόμβοι στον προορισμό τους ξεκίνα την περιμετρική κάλυψη δεξιόστροφα

4.2.3 Επεξήγηση αλγορίθμου

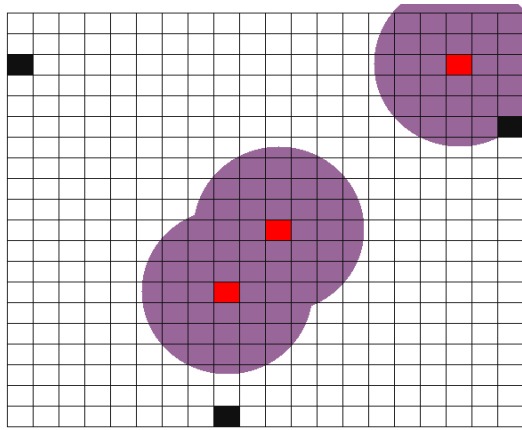
Όπως έχει προαναφερθεί ο χρήστης το μόνο που έχει να κάνει είναι να εισάγει στο πρόγραμμα τον αριθμό των κόμβων που θέλει να χρησιμοποιηθούν και το μέγεθος της μιας πλευράς του τετραγωνικού εικονικού χώρου στον οποίο θέλουμε να εφαρμόσουμε την περιμετρική κάλυψη. Οι κόμβοι τοποθετούνται μετά από το πρόγραμμα σε τυχαίες θέσεις μέσα στον χώρο, που αναπαριστάτε σαν ένα πλέγμα.



Σχήμα 4.3: Αρχική τυχαία τοποθέτηση των κόμβων και επιλογή κατάλληλων θέσεων για αρχική τοποθέτηση πριν την περιμετρική κίνηση

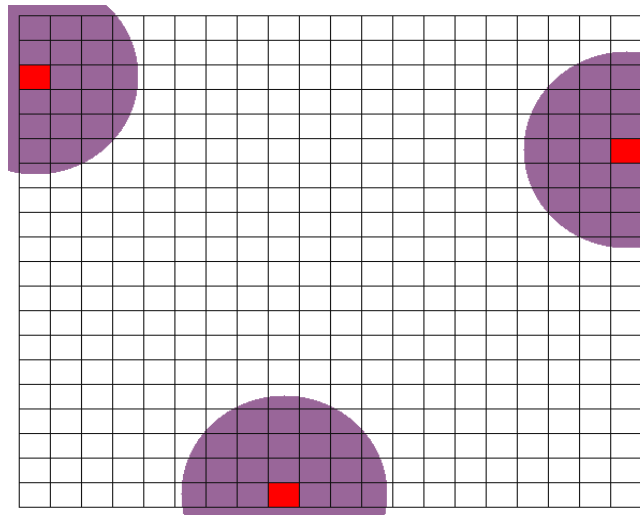
Το πρόγραμμα αμέσως υπολογίζει μετά βάση της συνολικής περιμέτρου του χώρου μας και του αριθμού των κόμβων, τις καλύτερες θέσεις στις οποίες πρέπει να πάνε οι κόμβοι για να έχουμε ομοιόμορφη περιμετρική κάλυψη. Βασικά διαιρείται η περίμετρος με τον αριθμό των κόμβων και με βάση το αποτέλεσμα, το πρώτο τελικό σημείο θα είναι στην απόσταση $1 \cdot (\text{πηλίκο τις διαίρεσης})$, ο δεύτερος στο $2 \cdot (\text{πηλίκο τις διαίρεσης})$ και το K στο $K \cdot (\text{πηλίκο τις διαίρεσης})$, όπου K ο δοσμένος αριθμός των κόμβων.

Μετά γίνεται η κίνηση των κόμβων προς τα σημεία, όμως η κίνηση αυτή είναι η μικρότερη δυνατή συνολικά για όλους τους κόμβους. Βασικά δεν πάει ο κάθε κόμβος στο πιο κοντινό του σημείο, αλλά επιλέγετε για κάθε σημείο ο κόμβος ο οποίος βρίσκεται πιο κοντά του για να πάει εκεί. Έτσι εξοικονομούμε, έστω και λίγη ενέργεια για αυτή την κίνηση.



Σχήμα 4.4: Επιλογή των πιο κοντινών κόμβων στα τελικά σημεία και κίνηση προς αυτά

Μετά ξεκινά η κίνηση των K κόμβων προς τα σημεία που έχουν σαν τελικό προορισμό για να ξεκινήσουν μετά την περιμετρική τους κίνηση στο χώρο. Μόλις όλοι οι κόμβοι τοποθετηθούν ξεκινούν την περιπολία τους στην περίμετρο.



Σχήμα 4.5: Οι κόμβοι έχουν φτάσει στις τελικές τους θέσεις και ξεκινούν την περιμετρική τους κίνηση(δεξιόστροφα)

4.3 Ομαδοποίηση κόμβων σε κινητό ασύρματο δίκτυο

4.3.1 Γενικά για το πρόβλημα

Δύο είναι τα κύρια ζητήματα αυτής της εφαρμογής, και για αυτό αναπτύχθηκαν δύο αλγόριθμοι. Πρώτα από όλα έχουμε το ζήτημα της εύρεσης άλλων κόμβων αισθητήρων και την ομαδοποίηση τους μέσα σε ένα καθορισμένο χώρο με την βοήθεια της κίνησής τους. Δηλαδή μετά από μια τυχαία κατανομή τους σε ένα χώρο και με δεδομένο ότι γνωρίζουν τα όρια της περιοχής, καθώς επίσης και το συνολικό αριθμό των άλλων κόμβων-αισθητήρων, να

μπορούν αυτοί οι κόμβοι-αισθητήρες να βρουν τους άλλους κόμβους και να συνεχίζουν μαζί όλοι την κίνησή τους μέχρι να βρεθούν όλοι.

Το δεύτερο θέμα-πρόκληση που εξετάστηκε για αυτή την κατηγορία αλγορίθμων, ήταν το θέμα του να μπορούν οι κόμβοι-αισθητήρες να ομαδοποιηθούν βάση κάποιων κοινών τους χαρακτηριστικών. Ένα σενάριο είναι να έχουμε δύο αντίπαλες ομάδες κόμβων διεσπαρμένες στον χώρο και με την κίνησή τους να καταλήξουν να έχουν χωριστεί, εκτελώντας ταυτόχρονα ξεχωριστές τροχιές.

Και για τους δύο πιο πάνω αλγορίθμους που έχουν αναπτυχθεί σε γραφικό περιβάλλον, έχουμε ένα κόμβο ο οποίος χρίζεται εκ των προτέρων ως αρχηγός. Έτσι οι υπόλοιποι κόμβοι-αισθητήρες θα μπορούν, όταν βρουν τον αρχηγό τους, να ομαδοποιηθούν μαζί του και να συνεχίσουν μαζί πλέον την κίνησή τους μέχρι να βρουν όλα τα μέλη της ομάδας τους. Από την άλλη, όταν βρουν οποιοδήποτε άλλο κόμβο, συμπεριλαμβανομένων και κόμβων της δικής τους ομάδας, που όμως δεν έχουν ομαδοποιηθούν με τον αρχηγό τους, τότε αλλάζουν και οι δύο πορεία. Αυτό μας προσφέρει περισσότερες πιθανότητες για να μπορέσουν όλοι οι κόμβοι να βρεθούν μέσα στην πορεία του αρχηγού τους ή στη συνέχεια στην πορεία της ομάδας που θα σχηματιστεί.

4.3.2 Αλγόριθμος για εύρεση άλλων κόμβων με κίνηση

Αρχικά γίνεται από τον χρήστη η εισαγωγή των επιθυμητών μεταβλητών για το πρόβλημα. Συγκεκριμένα εισάγεται ο αριθμός των κόμβων που θα χρησιμοποιηθεί κάθε φορά καθώς επίσης και το μέγεθος της περιοχής. Όσον αφορά το τελευταίο, χρειάζεται μόνο η εισαγωγή μιας πλευράς τετραγώνου, αφού όπως έχει προαναφερθεί χρησιμοποιείται τετράγωνος γραφικός χώρος. Ένα από τους κόμβους επιλέγεται τυχαία σαν αρχηγός για να έχουμε ένα σημείο αναφοράς, όπου θα πρέπει να μαζευτούν όλοι οι άλλοι κόμβοι.

Οι κόμβοι-αισθητήρες τοποθετούνται τυχαία στο χώρο και τους δίνεται μια τυχαία αρχική κατεύθυνση. Με το ξεκίνημα της κίνησής τους, ελέγχουν κάθε ένας ξεχωριστά αν υπάρχει στην καινούρια τους θέση κάποιος άλλο κόμβος, με βάση την εμβέλεια τους. Αν υπάρχει και είναι ο αρχηγός ή κάποιος άλλος κόμβος που ανήκει ήδη στην ομάδα του αρχηγού, συνεχίζει μαζί τους και αυτός. Αν είναι ο αρχηγός που βρίσκει κάποιο άλλο κόμβο, δεν αλλάζει την πορεία του ο αρχηγός αλλά ο άλλος κόμβος. Αν συναντηθούν δύο άλλοι κόμβοι, αλλάζουν κατευθύνσεις με σκοπό να βρουν μετά την ομάδα που θα σχηματιστεί.

Αλγοριθμικά Βήματα

Διάβασε τα δεδομένα από τον χρήστη

Τοποθέτησε τυχαία τους κόμβους στο χώρο

Επέλεξε τον αρχηγό της ομάδας

Ξεκίνησε την κίνηση των κόμβων-αισθητήρων

 Αν βρεις άλλο κόμβο στην εμβέλεια σου

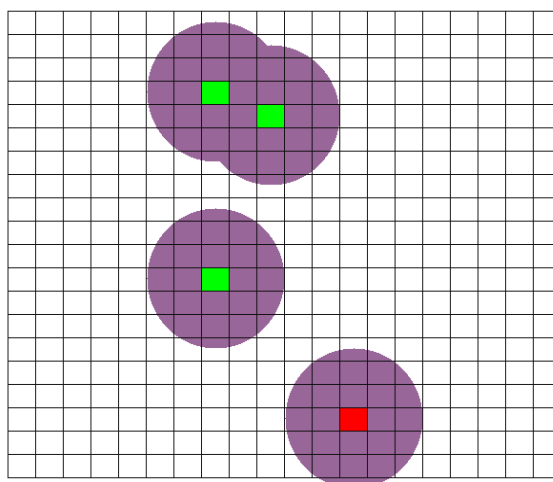
 Αν είναι ο αρχηγός ακόλουθα την τροχιά του

 Διαφορετικά άλλαξε κατεύθυνση

4.3.3 Επεξήγηση αλγορίθμου

Όπως είχαμε και στις προηγούμενες περιπτώσεις των αλγορίθμων που έχουν αναπτυχθεί για αυτή την διπλωματική εργασία, ο χρήστης πρέπει να εισάγει από το τερματικό τον αριθμό των κόμβων-αισθητήρων που επιθυμεί να χρησιμοποιήσει καθώς και το μέγεθος του πλέγματος, και πιο συγκεκριμένα το μέγεθος μιας από τις πλευρές του. Φυσικά απαραίτητη προϋπόθεση για τον αλγόριθμο είναι να γνωρίζουν οι κόμβοι τα όρια της περιοχής στην οποία βρίσκονται.

Οι κόμβοι τοποθετούνται αυθαίρετα στο χώρο. Με το ξεκίνημα της κίνησης τους τους δίνεται από μια τυχαία κατεύθυνση από το πρόγραμμα. Ο αρχηγός της ομάδας διακρίνεται με διαφορετικό χρώμα από τους υπόλοιπους όπως βλέπουμε και στο Σχήμα 4.6.

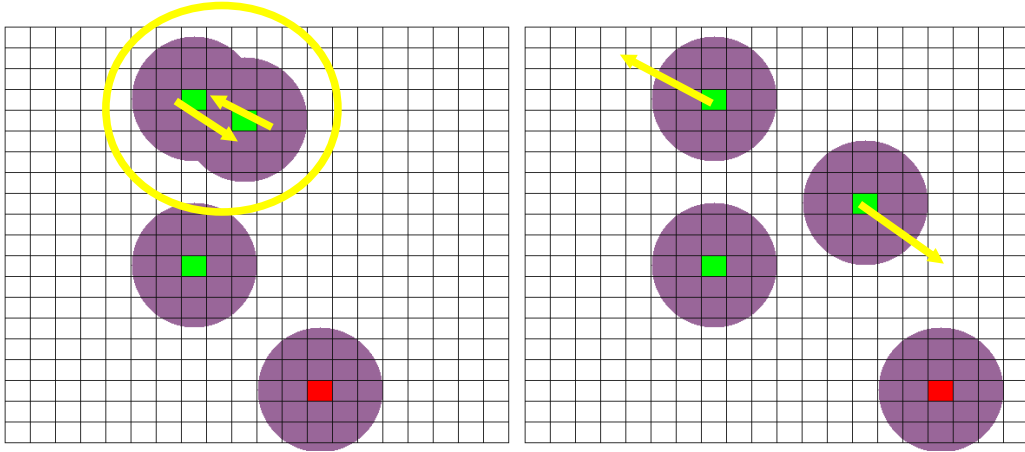


Σχήμα 4.6: Αρχικοποίηση του προγράμματος με 4 κόμβους. Διακρίνεται ο αρχηγός με κόκκινο χρώμα.

Για να ελέγχουμε το αν κάποιος κόμβος μπει στην εμβέλεια κάποιου άλλου, έχουμε ένα πίνακα που είναι αρχικοποιημένος με 0. Ο πίνακας αυτός είναι ίσος με το χώρο που χρησιμοποιούμε. Για κάθε κόμβο που τοποθετείτε στο πλέγμα, αλλάζει την αντίστοιχη τιμή του τετραγώνου που βρίσκεται ο κόμβος σε ένα αριθμό μεγαλύτερο από ένα, και στο χώρο γύρω του σχηματίζει ένα κατειλημμένο κυκλικό χώρο με τιμή ένα. Ο κυκλικός αυτός χώρος αντιπροσωπεύει την ακτίνα λήψης και αποστολής κάθε κόμβου και η ακτίνα του είναι ίση με την τιμή RANGE, η οποία αντιπροσωπεύει με την σειρά της την εμβέλεια που θέλουμε να έχουν οι κόμβοι μας. Με την κίνηση του κόμβου στο χώρο, αλλάζει και η θέση του στον πίνακα ελέγχου. Σε κάθε γύρο της προσομοίωσης γίνεται έλεγχος για το αν υπάρχει άλλος κόμβος στην εμβέλεια του κάθε κόμβου που ελέγχουμε.

Ο κάθε κόμβος εκτελεί την δική του ανεξάρτητη κίνηση. Μόλις βρεθεί στον δρόμο του κάποιος άλλος κόμβος, τότε ελέγχει αν αυτός είναι είτε ο αρχηγός είτε κάποιος κόμβος που είναι ήδη ομαδοποιημένες με τον αρχηγό. Αν είναι μια από αυτές τις περιπτώσεις, τότε προσκολλάται και αυτός σε όσους ήδη ανήκουν στην ομάδα και συνεχίζει μαζί τους την κοινή πορεία. Στην περίπτωση που βρει κάποιο άλλο μη ομαδοποιημένο κόμβο, τότε και οι δύο αλλάζουν πορεία.

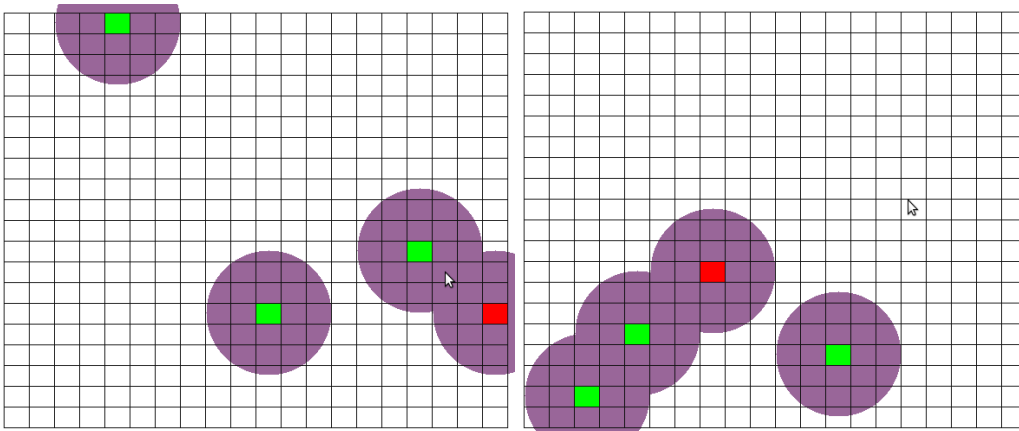
Αυτή η αλλαγή πορείας γίνεται για δύο λόγους. Πρώτο, γιατί παρατήρησα ότι είναι πιο καλό να ομαδοποιώ τους κόμβους σε μια ομάδα μόνο και όχι σε πολλές μικρές που να ενώνονται μετά σε μια μεγάλη. Στις αρχικές δοκιμές που έκανα διαπίστωσα πως σε πάρα πολλές περιπτώσεις, όταν δημιουργούσα πολλές μικρές ομάδες και μετά περίμενα να ενωθούν όλες σε μια, δεν υπήρχε ποτέ συνάντηση των υποομάδων, με αποτέλεσμα να κινούνται αέναα στο χώρο χωρίς στο τέλος να σχηματιστεί μια και μόνο ομάδα. Από την άλλη, με το σχηματισμό μιας μόνο ομάδας, οι περιπτώσεις να μην ενωθούν όλοι οι κόμβοι σε μια ομάδα ήταν αισθητά πιο λίγες. Ακόμη και στις περιπτώσεις που δεν συμπληρωνόταν η ομάδα, ελάχιστοι μεμονωμένοι κόμβοι έμεναν εκτός. Δεύτερος λόγος, παρεμφερής με τον πρώτο, είναι για να μπορούν και αυτοί οι μεμονωμένοι κόμβοι που μένουν εκτός από την τελική ομάδα, να έχουν μια νέα πορεία. Έτσι έχουμε περισσότερες πιθανότητες να συναντήσουν σε κάποια φάση την ομάδα και να προσκολληθούν σε αυτή. Στο Σχήμα 4.7 βλέπουμε μια περίπτωση συνάντησης δύο κόμβων που δεν ανήκουν στην τελική ομάδα και την επόμενη τους κίνηση, μετά την αλλαγή της κατεύθυνσης της φοράς τους.



Σχήμα 4.7: Συνάντηση δύο κόμβων και αλλαγή πορείας τους. Αριστερά βλέπουμε τους δύο κόμβους που είναι κυκλωμένοι να συναντώνται και δεξιά βλέπουμε την αλλαγή πορείας και των δύο προς αντίθετες κατευθύνσεις.

Ο αρχηγός της ομάδας δεν αλλάζει ποτέ την κίνηση του. Ότι και αν συμβεί συνεχίζει απρόσκοπτα την αρχική του τροχιά στο χώρο. Αν βρεθεί στον δρόμο του κάποιος κόμβος, απλά προσκολλάται πάνω του ο άλλος κόμβος και συνεχίζουν ομαδοποιημένοι πλέον την κοινή τους κίνηση.

Σχήμα 4.8: Αριστερά ο κόμβος-αρχηγός ομαδοποιείται με τον πρώτο κόμβο που βρίσκεται στο



δρόμο του και μαζί συνεχίζουν μέχρι να βρουν τους επόμενους κόμβους. Δεξιά έχει πλέον προσκολληθεί και δεύτερος κόμβος στην ομάδα σε επόμενο στιγμιότυπο.

Όταν στο τέλος σχηματιστεί η ομάδα συνεχίζει την κοινή της πορεία στο χώρο χωρίς διακοπή. Εννοείτε πως στη συνέχεια οι κόμβοι σε μια πραγματική εφαρμογή θα μπορούν να συνεχίσουν με κάποια άλλη εργασία, όπως για παράδειγμα να τρέξουν τον αλγόριθμο για την περιμετρική περιπολία που αναπτύχθηκε πιο πάνω.

4.3.4 Αλγόριθμος για ομαδοποίηση κόμβων βάση όμοιων χαρακτηριστικών με κίνηση

Πρόκειται για επέκταση βασικά του προηγούμενου αλγορίθμου. Σκοπός του είναι να ομαδοποιούνται κόμβοι σε ένα δίκτυο αισθητήρων βάση κοινών τους χαρακτηριστικών. Αρχικά γίνεται από τον χρήστη η εισαγωγή των επιθυμητών μεταβλητών για το πρόβλημα. Συγκεκριμένα εισάγεται ο αριθμός των κόμβων που θα χρησιμοποιηθεί κάθε φορά καθώς επίσης και το μέγεθος της περιοχής.

Ο αριθμός των κόμβων που θα εισαχθεί αντιπροσωπεύει όλους τους κόμβους που θα χρησιμοποιηθούν. Βασικά αυτός ο αριθμός χωρίζεται στα δύο και οι δύο ομάδες έχουν ίσο αριθμό κόμβων(αν δοθεί φυσικά ζυγός αριθμός στην είσοδο του προγράμματος). Ένας κόμβος από κάθε ομάδα επιλέγεται τυχαία για αρχηγός.

Οι κόμβοι-αισθητήρες τοποθετούνται τυχαία στο χώρο και τους δίνεται μια τυχαία αρχική κατεύθυνση. Με το ξεκίνημα της κίνησής τους, ελέγχουν κάθε ένας ξεχωριστά αν υπάρχει στην καινούρια τους θέση κάποιος άλλο κόμβος, με βάση την εμβέλεια τους. Αν υπάρχει και είναι ο αρχηγός της ομάδας με τα ίδια χαρακτηριστικά με αυτόν ή κάποιος άλλος κόμβος που ανήκει ήδη στην ομάδα του αρχηγού του, συνεχίζει μαζί τους και αυτός. Αν είναι ο αρχηγός που βρίσκει κάποιο άλλο κόμβο, δεν αλλάζει την πορεία του ο αρχηγός αλλά ο άλλος κόμβος. Αν συναντηθούν δύο άλλοι κόμβοι οι οποίοι δεν έχουν προσαρτηθεί σε καμιά ομάδα, αλλάζουν κατευθύνσεις με σκοπό να βρουν μετά την ομάδα που θα σχηματιστεί, είτε έχουν κοινά χαρακτηριστικά είτε όχι.

Αλγοριθμικά Βήματα

Διάβασε τα δεδομένα από τον χρήστη

Τοποθέτησε τυχαία τους κόμβους στο χώρο

Επέλεξε τον αρχηγό κάθε ομάδας

Ξεκίνησε την κίνηση των κόμβων-αισθητήρων

 Αν βρεις άλλο κόμβο στην εμβέλεια σου

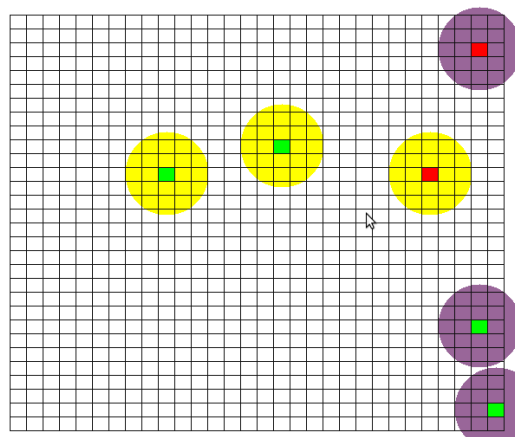
 Αν ανήκει στην ομάδα σου έλεγξε

 Αν είναι ο αρχηγός της ομάδας σου ακόλουθα την τροχιά του

 Διαφορετικά άλλαξε κατεύθυνση

4.3.5 Επεξήγηση αλγορίθμου

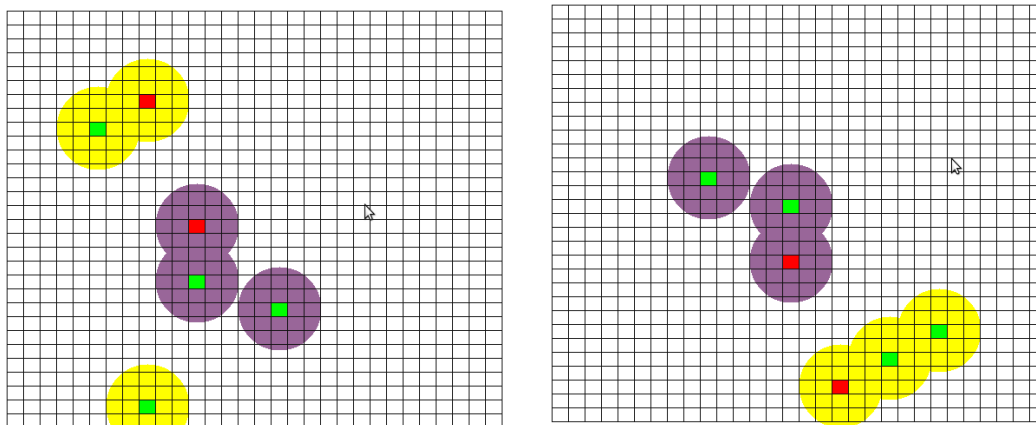
Ο αλγόριθμος έχει σχεδιαστεί για την ομαδοποίηση δύο ομάδων με διαφορετικά χαρακτηριστικά. Έχουμε πάλι την εισαγωγή από τον χρήστη των στοιχείων που χρειάζεται το πρόγραμμα για να ξεκινήσει να λειτουργεί. Βασικά εισάγονται από το τερματικό ο αριθμός των κόμβων και το μέγεθος του πλέγματος που θα χρησιμοποιηθεί. Υπάρχει όμως μια ουσιώδης διαφορά για αυτό το σενάριο από το προηγούμενο. Βασικά ο αριθμός που θα εισαχθεί από τον χρήστη αντιπροσωπεύει τον συνολικό αριθμό κόμβων στην προσομοίωση, δηλαδή τον αριθμό και των δύο ομάδων που θα υπάρχουν μαζί. Οι δύο ομάδες παίρνουν από μισούς κόμβους, οι οποίοι τοποθετούνται τυχαία στο χώρο προτού ξεκινήσει η κίνησή τους. Για κάθε ομάδα επιλέγεται πάλι ένας αρχηγός, για να γίνει η ομαδοποίηση της κάθε μιας βάση αυτού. Όπως φαίνεται και στο Σχήμα 4.9, οι αρχηγοί ξεχωρίζουν από το κόκκινο χρώμα σε σχέση με το πράσινο που έχουν όλοι οι άλλοι κόμβοι. Επίσης οι δύο διαφορετικές ομάδες ξεχωρίζουν από τα διαφορετικά χρώματα που έχουν οι κόμβοι τους σαν χρώμα στην εμβέλεια του σήματος τους.



Σχήμα 4.9: Αρχικοποίηση του προγράμματος με 6 κόμβους σαν είσοδο. Διακρίνονται οι 2 ομάδες των τριών κόμβων με κίτρινο και μοβ χρώμα σαν αναπαράσταση της εμβέλειας των κόμβων και οι αρχηγοί της κάθε ομάδας με κόκκινο χρώμα κόμβου.

Όπως και στον προηγούμενο αλγόριθμο, έτσι και τώρα ανατίθεται στον κάθε κόμβο μια τυχαία αρχική διεύθυνση για κίνηση και ξεκινά η προσομοίωση. Μόλις ένας κόμβος βρει μέσα στην εμβέλεια του εμβέλεια άλλου κόμβου, ελέγχει πρώτα από όλα αν αυτός ο κόμβος ανήκει στην ίδια ομάδα με αυτόν. Αν όχι αλλάζουν και οι δύο την πορεία τους. Στην περίπτωση που είναι της ίδια ομάδας με αυτόν, τότε πραγματοποιεί ακόμη ένα επιπλέον έλεγχο. Βλέπει αν ο κόμβος που έχει συναντήσει είναι ο αρχηγός του ή κόμβος που έχει ήδη ομαδοποιηθεί μαζί με τον αρχηγό του. Αν όντως ισχύει αυτό το σενάριο τότε μπαίνει και αυτός στην ομάδα, αλλάζοντας την φορά της κίνησής του και κάνοντας την ίδια με την κίνηση της ομάδας. Αν πάλι βρεθεί με κάποιο κόμβο της ομάδας του, που όμως δεν ανήκει

στην τελική ομάδα τότε αλλάζουν και οι δύο κατευθύνσεις και συνεχίζουν το ψάξιμο για εύρεση της ομάδας.



Σχήμα 4.10: Αριστερά ο τελικός σχηματισμός της μοβ ομάδας. Δεξιά βλέπουμε ότι η κίνηση συνεχίζεται μέχρι να βρεθούν και όλοι οι κόμβοι της κίτρινης ομάδας.

Όπως και πριν, πάλι έχουμε ένα πίνακα δευτερεύον για να ελέγχουμε σε κάθε γύρο αν υπάρχει στην εμβέλεια του κάθε κόμβου κάποιος άλλος κόμβος. Αυτή την φορά όμως είναι φυσικά να πρέπει να κοιτάζουμε και σε πια ομάδα ανήκει ο άλλος κόμβος εκτός του αν είναι ο αρχηγός της ομάδας ή όχι.

Αξίζει να σημειωθεί ότι οι ομάδες, δεν αντιδρούν όταν βρεθούν η μια με την άλλη. Θα ήταν πολύ ζημιογόνο να αλλάζουν σε κάθε συνάντησή τους κατεύθυνση όλοι οι κόμβοι της κάθε ομάδας. Αν από την άλλη συναντηθεί μια ομάδα με ένα κόμβο της άλλης ομάδας, τότε ο μεμονωμένος αυτός κόμβος αλλάζει την κίνησή του μόνο.

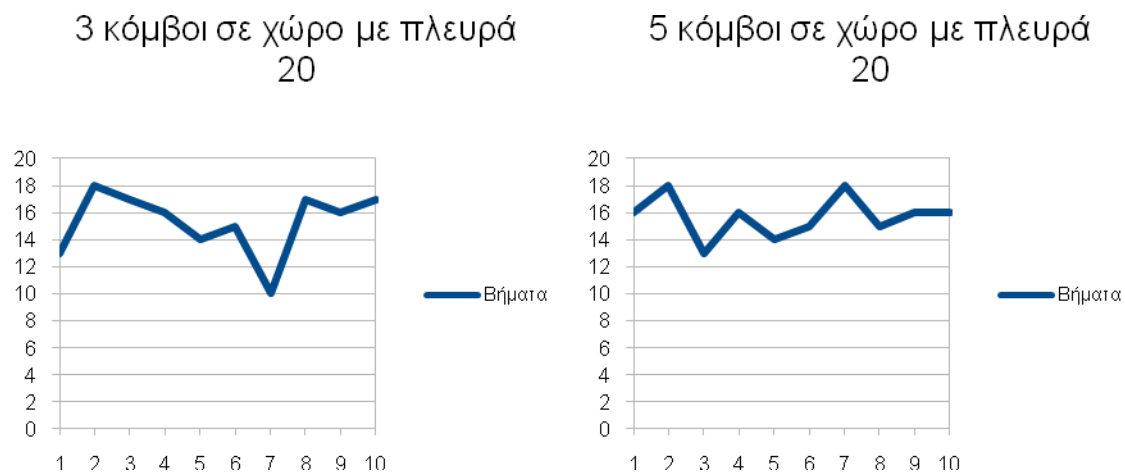
4.4 Αποτελέσματα-Ανάλυση

Για να μπορέσουμε να αξιολογήσουμε τους αλγόριθμους που αναπτύχθηκαν, πρέπει να εκτελέσουμε αρκετές μετρήσεις για να δούμε πόσο αποτελεσματικοί είναι. Για τις ανάγκες αυτής της αξιολόγησης έτρεξα διάφορα αντιπροσωπευτικά σενάρια για τον κάθε αλγόριθμο πολλές φορές, μετρώντας κάθε φορά τα πόσα βήματα χρειάστηκαν για να ολοκληρώσει ο αλγόριθμος την εργασία που έχει να κάνει.

4.4.1 Έλεγχος-Περιπολία περιοχής

Για αυτό το πρόγραμμα έτρεξα δύο σενάρια για έλεγχο σε σχέση με το μέγεθος της περιοχής. Πρώτα έτρεξα την προσομοίωση για περιοχή με περίμετρο 80 τετράγωνα και μετά με περίμετρο 120. Για κάθε ένα από τα δύο σενάρια είχα διαφορετικές εκτελέσεις με διαφορετικό αριθμό κόμβων. Για το σενάριο με την περίμετρο των 80 τετραγώνων, έκανα την μια δοκιμή με 3 κόμβους και την άλλη με 5. Αντίστοιχα στο άλλο σενάριο είχα την πρώτη δοκιμή με 10 κόμβους και την επόμενη με 20. Με αυτό τον τρόπο ήθελα να δοκιμάσω το πόσο γρήγορα μπορούσε να τερματίσει ο αλγόριθμος σε διάφορες συνθήκες. Θεωρούσα σημείο τερματισμού τη στιγμή που και ο τελευταίος κόμβος έφτανε στην τελική του θέση πάνω στην περίμετρο του χώρου και ήταν πλέον όλοι έτοιμοι να ξεκινήσουν την περιμετρική περιπολία τους.

Σενάριο με χώρο περιμέτρου 80 τετραγώνων

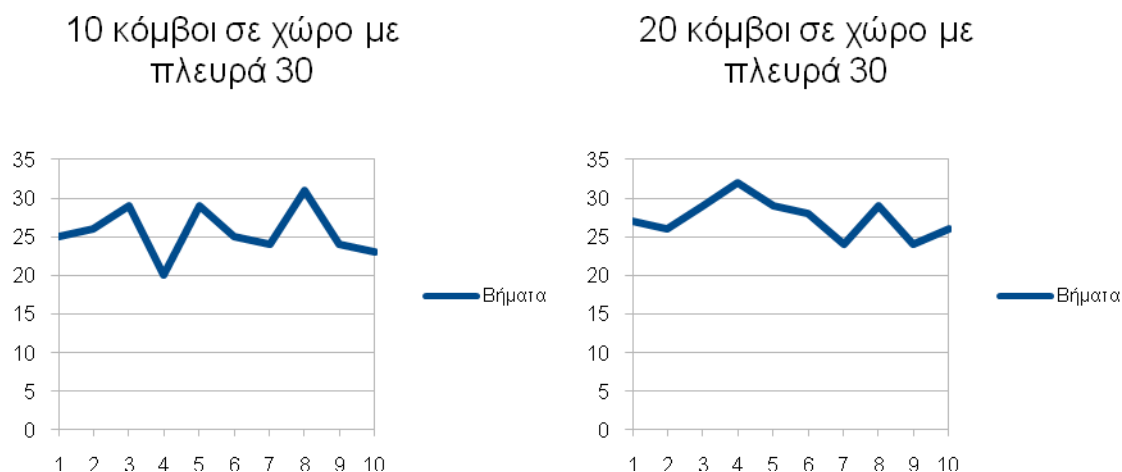


Σχήμα 4.11: Γραφικές παραστάσεις για περιμετρική κάλυψη σε χώρο 80 τετραγώνων

Στις πιο πάνω γραφικές παραστάσεις βλέπουμε τον αριθμό των βημάτων που χρειάστηκε για 10 φορές το πρόγραμμα για να ολοκληρώσει τις εργασίες του. Αριστερά παρατηρούμε την γραφική παράσταση για το σενάριο με τους 3 κόμβους. Όπως βλέπουμε πάντα χρειάζεται πιο λίγα βήματα από τα 20 τετράγωνα που είναι η μια πλευρά του χώρου. Κατά μέσο όρο είχαμε 15,3 βήματα σε κάθε επανάληψη του αλγορίθμου. Αυτό μας δίνει ένα αρκετά καλό και γρήγορο αποτέλεσμα για αυτό το πρόβλημα. Μπορούμε να παρατηρήσουμε ακόμη σχετικά μεγάλες διακυμάνσεις στα βήματα που χρειάζονταν κάθε φορά. Αυτό φυσικά οφείλετε στο ότι οι κόμβοι ξεκινούσαν κάθε φορά από τυχαία σημεία.

Στην δεξιά γραφική παράσταση βλέπουμε τα αποτελέσματα για το σενάριο με τους 5 κόμβους μέσα στο χώρο. Όπως βλέπουμε έχουμε πάλι παρόμοια αποτελέσματα με την προσομοίωση με τους 3 κόμβους. Έχουμε ένα μέσο όρο 15,7 βημάτων για την ολοκλήρωση του προβλήματος κάθε φορά. Η μικρή αύξηση που παρατηρείτε οφείλετε στην αύξηση των κόμβων στο δίκτυο. Πάλι όμως παραμένει σε πολύ ικανοποιητικά όρια.

Σενάριο με χώρο περιμέτρου 120 τετραγώνων



Σχήμα 4.12: Γραφικές παραστάσεις για περιμετρική κάλυψη σε χώρο 120 τετραγώνων

Στα πιο πάνω γραφήματα βλέπουμε πάλι τον αριθμό των βημάτων που χρειάστηκαν για κάθε μια από τις 10 φορές που έτρεξαν οι προσομοιώσεις του προγράμματος για την περιμετρική κάλυψη σε ένα χώρο με μήκος πλευράς 30 τετράγωνα, συνολικής περιμέτρου δηλαδή 120 τετραγώνων. Αριστερά βρίσκεται η γραφική παράσταση για την προσομοίωση με τους 10 κόμβους. Βλέπουμε ότι έχουμε ένα μέσο όρο βημάτων στα 25,6 για τις δοκιμές που έγιναν. Βλέπουμε ακόμη ότι υπάρχουν 2 σχετικά αποκλίσεις στην τέταρτη και την όγδοη δοκιμή. Αυτές οι διακυμάνσεις συμβαίνουν λόγω της τυχαίας τοποθέτησης των κόμβων και σε συνάρτηση με το ότι έχουμε σχετικά μικρό αριθμό κόμβων, μπορεί να υπάρχει τόση διαφορά (μια διαφορά δέκα βημάτων) αφού μπορεί να είχαμε δύο ακραία σενάρια, στα οποία οι κόμβοι ήταν την μια φορά σε σχετικά πολύ μικρές αποστάσεις από τους τελικούς τους προορισμούς και την άλλη σε μια από τις περιπτώσεις με τις πιο μεγάλες σχετικά αποστάσεις που θα πρέπει να καλύψουν από την αρχική τους θέση. Κατά τα άλλα ο μέσος όρος βημάτων που βγήκε μέσα από αυτές τις δοκιμές είναι πολύ ικανοποιητικός αφού είναι μικρότερος σε μέσο όρο από μια πλευρά του τετραγωνικού χώρου.

Στη δεξιά γραφική παράσταση βλέπουμε τα αποτελέσματα για την δοκιμή με 20 κόμβους. Εδώ έχουμε ένα πιο σταθερό αριθμό βημάτων από πριν, όπως είχαμε και στο σενάριο με τον χώρο περιμέτρου 20X20 τετραγώνων. Επίσης έχουμε ελαφρώς πιο μεγάλο αριθμό βημάτων από ότι με τους 10 κόμβους στην ίδια περιοχή, αφού ο μέσος όρος μας τώρα είναι στα 27,4 βήματα κάθε φορά. Αυτά τα δύο οφείλονται στο γεγονός ότι έχουμε περισσότερους κόμβους στον χώρο. Και πάλι μπορούμε να πούμε ότι το αποτέλεσμα μας είναι καλό και ενθαρρυντικό.

Ανάλυση αποτελεσμάτων

Σε γενικές γραμμές τα πειράματα για αυτό το πρόγραμμα ήταν πολύ καλά. Σε όλες τις περιπτώσεις είχαμε σχετικά μικρό αριθμό βημάτων για ολοκλήρωση του προβλήματος και δεν παρατηρήθηκαν ιδιαίτερα προβλήματα. Φυσικά για να είναι λειτουργικός ο αλγόριθμος αυτός σε ένα κανονικό δίκτυο, εννοείτε πως πρέπει να έχουμε ένα κόμβο-συντονιστή που θα παίζει περίπου το ρόλο του προγράμματος. Βασικά αυτός θα πρέπει να αποφασίζει ποιος κόμβος θα πάει που και πότε πρέπει να ξεκινούν οι κόμβοι να κάνουν την περιπολία τους. Επίσης θα πρέπει να είναι υπεύθυνος να μαζεύει τα αποτελέσματα και τις μετρήσεις που καταγράφουν κατά τη κίνησή τους οι υπόλοιποι κόμβοι, είτε με μια μορφή κίνησης, όπως για παράδειγμα μια εσωτερική αντίστροφη τροχιά από αυτή που κάνουν οι κόμβοι που περιπολούν, είτε μένοντας σταθερός σε ένα σημείο από όπου θα μπορεί να πιάσει σε κάθε γύρο μηνύματα από τους άλλους, δηλαδή συμπεριφερόμενος σαν checkpoint.

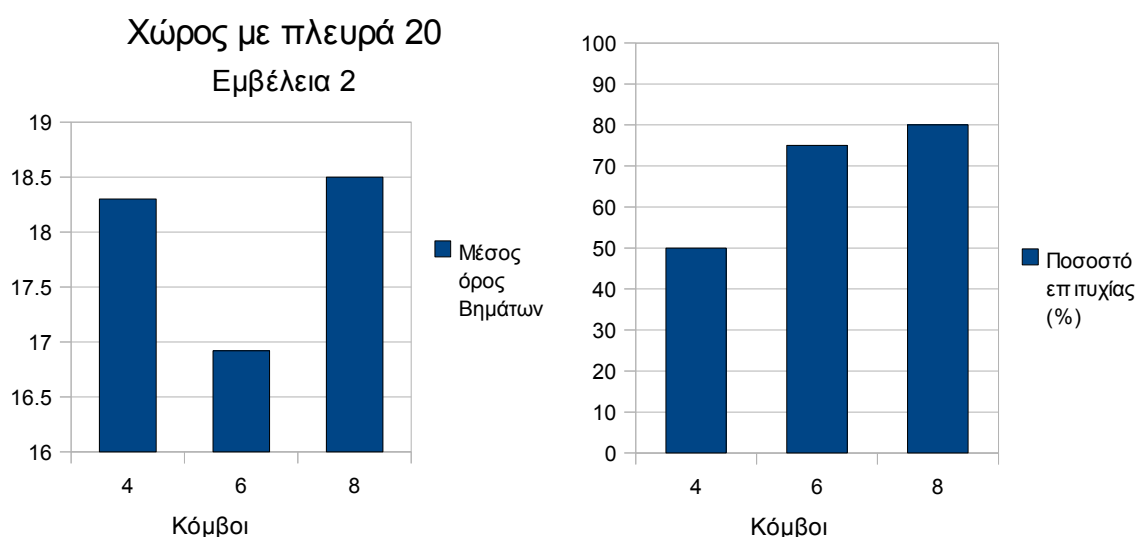
4.4.2 Αλγόριθμος για εύρεση άλλων κόμβων με κίνηση

Για το πρόγραμμα αυτό έγιναν δύο μετρήσεις με διαφορετικές παραμέτρους. Αρχικά έγιναν δοκιμές για χώρο με περίμετρο 80 τετραγώνων. Αυτές οι δοκιμές χωρίστηκαν σε δύο διαφορετικά σενάρια. Στο ένα είχαμε όλους τους κόμβους να έχουν εμβέλεια λήψης και αποστολής 2 τετραγώνων και στο άλλο 3. Για κάθε ένα από αυτά τα δύο σενάρια τώρα έγιναν δοκιμές με 4, 6 και 8 κόμβους μέσα στο χώρο. Μετά πάρθηκαν μετρήσεις για χώρο περιμέτρου 400 τετραγώνων, με 100 κόμβους κάθε φορά. Τα δύο διαφορετικά σενάρια εδώ υπονοούσαν εμβέλεια κόμβων 3 και 5 τετραγώνων αντίστοιχα. Σαν ολοκλήρωση του αλγορίθμου έχουμε την στιγμή που όλοι οι κόμβοι είναι πλέον ενωμένοι σε μια ομάδα.

Εκτός από τις γραφικές για τα πόσα βήματα χρειάστηκαν για να ολοκληρωθεί κάθε φορά το πρόβλημα χρειάστηκε να κατασκευαστούν και γραφικές παραστάσεις με το ποσοστό

επιτυχίας κάθε φορά, αφού υπήρχαν αρκετές φορές που ο αλγόριθμος δεν τερμάτιζε την λειτουργία του αλλά συνέχιζαν οι κόμβοι μια ατέρμον κίνηση στο χώρο. Για αυτό το σκοπό προτάθηκε και μια μικρή προσθήκη που υλοποιήθηκε στον επόμενο αλγόριθμο (αλγόριθμος για ομαδοποίηση κόμβων βάση όμοιων χαρακτηριστικών με κίνηση) και θα την δούμε στην ανάλυση των αποτελεσμάτων αυτού του αλγορίθμου στο κεφάλαιο 4.4.3, στην παράγραφο που αναλύονται τα αποτελέσματα από τις δοκιμές που πραγματοποιήθηκαν εκεί. Και για αυτό τον αλγόριθμο θα μπορούσε να γίνει η προσθήκη όμως κρίθηκε πιο χρήσιμο να γίνει μόνο στον επόμενο.

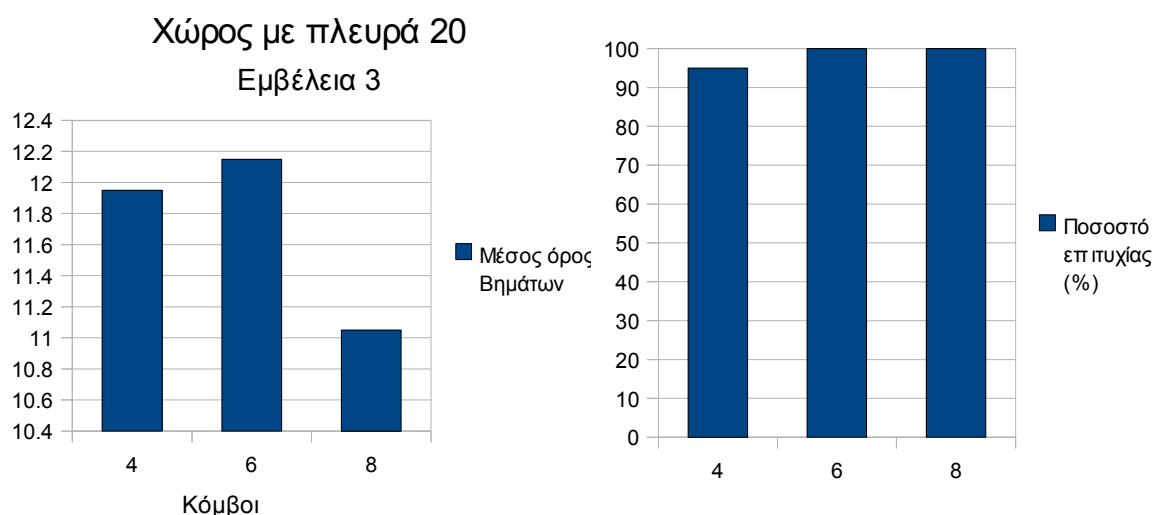
Σενάριο με χώρο περιμέτρου 80 τετραγώνων



Σχήμα 4.13: Γραφικές παραστάσεις για ομαδοποίηση σε χώρο 80 τετραγώνων

Εδώ βλέπουμε τις γραφικές παραστάσεις με τα αποτελέσματα από τις προσομοιώσεις για τον χώρο με περίμετρο 80 τετραγώνων για το σενάριο που όλοι οι κόμβοι έχουν εμβέλεια 2 τετράγωνα. Αριστερά έχουμε το γράφημα με τον μέσο αριθμό των βημάτων για τις περιπτώσεις με τους 4, 6 και 8 κόμβους στον χώρο μας. Βλέπουμε ότι έχουμε ελαφρώς πιο μεγάλο αριθμό βημάτων για τις περιπτώσεις των 4 και των 8 κόμβων από την περίπτωση των 6 κόμβων. Αυτό οφείλεται γιατί από τη μια στην περίπτωση των 4 κόμβων, έχουμε πιο μικρή συνολική εμβέλεια στην οποία μπορεί να ανιχνεύσουν ο ένας κόμβος τον άλλο και από την άλλη στην περίπτωση με τους 8 κόμβους περισσότερες συγκρούσεις μεταξύ κόμβων που δεν έχουν ομαδοποιηθεί ακόμη, με αποτέλεσμα να αλλάζουν φορά κίνησης άρα και να αργούν λίγο περισσότερο να βρουν την υπόλοιπη σχηματισμένη ομάδα.

Όμως κάτι πολύ σημαντικό φαίνεται στην δεύτερη γραφική παράσταση (δεξιά). Εδώ βλέπουμε το ποσοστό επιτυχίας των αλγορίθμων, δηλαδή το πόσες φορές το πρόγραμμα έφτανε στην ομαδοποίηση των κόμβων. Από ότι βλέπουμε για τους 4 κόμβους έχουμε μόνο 50% επιτυχία, και 75% και 80% για τα σενάρια με 6 και 8 κόμβους αντίστοιχα. Αυτές οι απώλειες οφείλονται βασικά στο ότι έχουμε πολύ μικρή ακτίνα εμβέλειας, με αποτέλεσμα να μην μπορούν πάντα να βρεθούν οι κόμβοι. Το πρόβλημα είναι πολύ μεγαλύτερο φυσικά στην περίπτωση των τεσσάρων κόμβων, αφού εκτός από μικρή εμβέλεια έχουμε και πιο λίγους κόμβους, άρα πολύ πιο μικρή περιοχή που μπορεί να ελεγχθεί κάθε φορά. Αυτό φυσικά είναι πρόβλημα. Με αφορμή αυτές τις μετρήσεις δοκίμασα μετά και τα υπόλοιπα σενάρια, πρώτα με μεγαλύτερη ακτίνα στον ίδιο χώρο και μετά σε πολύ μεγαλύτερο χώρο και με πολύ περισσότερους κόμβους γιατί αν και εκεί είναι παρόμοια τα αποτελέσματα ο το πρόγραμμα και παράλληλα και ο αλγόριθμος που έχω αναπτύξει δεν θα είναι καθόλου αποτελεσματικός.



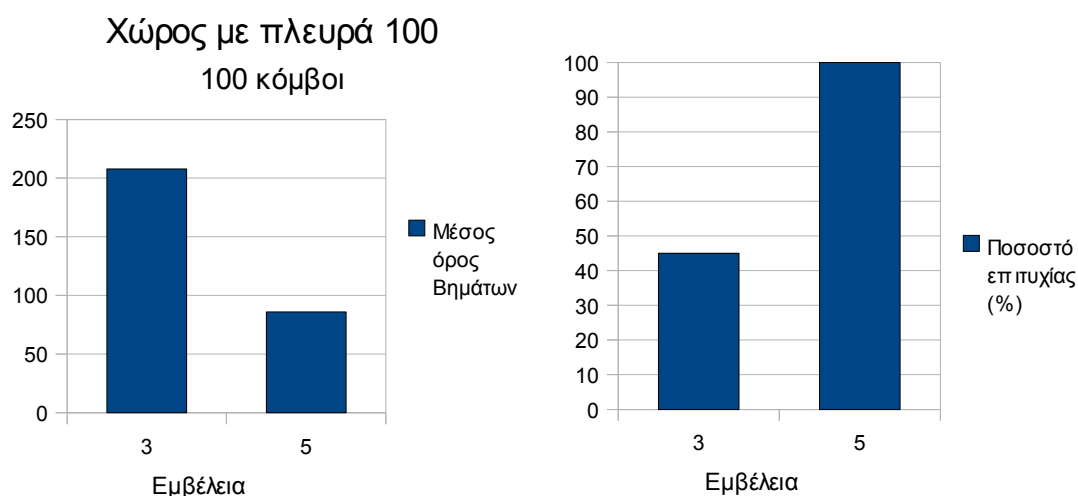
Σχήμα 4.14: Γραφικές παραστάσεις για ομαδοποίηση σε χώρο 80 τετραγώνων

Εδώ βλέπουμε τις γραφικές παραστάσεις με τα αποτελέσματα από τις προσομοιώσεις για τον χώρο με περίμετρο 80 τετραγώνων για το σενάριο που όλοι οι κόμβοι έχουν εμβέλεια 2 τετράγωνα. Αριστερά έχουμε πάλι την γραφική παράσταση με τον αριθμό τον μέσο αριθμό βημάτων που χρειάστηκε για την κάθε περίπτωση μέχρι να έχουμε λύση του προβλήματος μας. Αυτή τη φορά παρατηρούμε ότι έχουμε και στις τρεις περιπτώσεις ίδιο περίπου αποτελέσματα, με πιο χαμηλό μέσο όρο αυτή τη φορά για την περίπτωση που έχουμε να ομαδοποιήσουμε 8 κόμβους. Επίσης μπορούμε να παρατηρήσουμε ότι έχουμε αισθητά πιο μικρό αριθμό βημάτων από την προηγούμενη δοκιμή, που είχαμε κόμβους με ακτίνα 2 τετραγώνων. Αυτό οφείλετε στο γεγονός ότι πλέον μπορούμε να ελέγξουμε μεγαλύτερο

κομμάτι του χώρου σε κάθε περίπτωση, άρα μπορούμε σε πολύ πιο γρήγορο χρόνο να έχουμε εύρεση και σχηματισμό της ομάδας των κόμβων μας.

Στην γραφική παράσταση με τα ποσοστά επιτυχίας του προγράμματος που βλέπουμε στα δεξιά παρατηρούμε ότι πλέον έχουμε σχεδόν απόλυτη επιτυχία σε όλες τις περιπτώσεις. Μοναδικό ψεγάδι είναι ένα ποσοστό 5% αποτυχίας στο σενάριο που προνοούσε 4 μόνο κόμβους. Αυτές οι αποτυχίες συνέβηκαν λόγω του περιορισμένου χώρου που μπορούμε να καλύψουμε κάθε φορά με την εμβέλεια των 4 κόμβων, με αποτέλεσμα να υπάρχει μια πιθανότητα, έστω και πολύ μικρή, για να μην μπορούν με την κίνηση που κάνουν οι κόμβοι έτσι ώστε να μην συναντηθούν όλοι οι κόμβοι ποτέ. Φυσικά και τα αποτελέσματα είναι εξαιρετικά καλύτερα από ότι είχαμε στην προηγούμενη δοκιμή, και γενικά για τις περιπτώσεις που έχουμε πολλούς κόμβους έχουμε ένα πολύ αποτελεσματικό αλγόριθμο, θεωρητικά πάντα μιλώντας.

Σενάριο με χώρο περιμέτρου 400 τετραγώνων



Σχήμα 4.15: Γραφικές παραστάσεις για ομαδοποίηση σε χώρο 400 τετραγώνων

Πιο πάνω έχουμε τις γραφικές παραστάσεις με τα αποτελέσματα από τις προσομοιώσεις για τον χώρο με περίμετρο 400 τετραγώνων με 100 κόμβους. Από το γράφημα στα δεξιά, που παρουσιάζει τον μέσο όρο βημάτων μέχρι την ολοκλήρωση του προβλήματος στις περιπτώσεις με κόμβους με εμβέλεια με 3 και 5 τετράγωνα, μπορούμε αμέσως να καταλάβουμε το ότι ο αλγόριθμος είναι πολύ πιο αποδοτικός για τις περιπτώσεις που έχουμε πολύ περισσότερο χώρο να ελέγχεται κάθε φορά. Έχουμε ένα μέσο αριθμό λίγο πάνω από 200 βήματα (207,89) για την περίπτωση με την εμβέλεια των 3 τετραγώνων και από την

άλλη, στην περίπτωση με τα 5 τετράγωνα εμβέλεια έχουμε μέσο όρο 86 βημάτων για κάθε προσομοίωση.

Η μεγάλη διαφορά στην επίδοση του αλγορίθμου για κάθε σενάριο είναι ακόμη πιο εμφανής, όταν παρατηρήσουμε την γραφική παράσταση που παρουσιάζει τα ποσοστά επιτυχίας των δύο διαφορετικών σεναρίων. Βλέπουμε ότι για την περίπτωση που οι κόμβοι έχουν ακτίνα εμβέλειας 5 τετραγώνων, έχουμε απόλυτη επιτυχία, δηλαδή το πρόγραμμα πάντα να ομαδοποιεί και τους 100 κόμβους. Από την άλλη όμως οι αριθμοί είναι πολύ κακοί για το σενάριο με την εμβέλεια των 3 τετραγώνων. Το ποσοστό επιτυχίας δεν φτάνει ούτε το 50%. Είναι μόλις 45%. Αυτό δεν είναι καθόλου ικανοποιητικό νούμερο, αφού στην πράξη σημαίνει ότι θα έχουμε μικρές πιθανότητες να σχηματίσουμε μια ομάδα, έστω και με τις πολλές παραλήψεις από τυχόν εμπόδια και δυσκολίες που έχουμε κάνει για τις ανάγκες της προσομοίωσης αυτής. Όμως όπως έχει αναφερθεί και πιο πάνω, έχει προστεθεί στο πρόγραμμα για τον σχηματισμό 2 ομάδων στον ίδιο χώρο ένας επιπλέον έλεγχος που διασφαλίζει την απόλυτη επιτυχία του αλγορίθμου όπως θα δούμε.

Ανάλυση αποτελεσμάτων

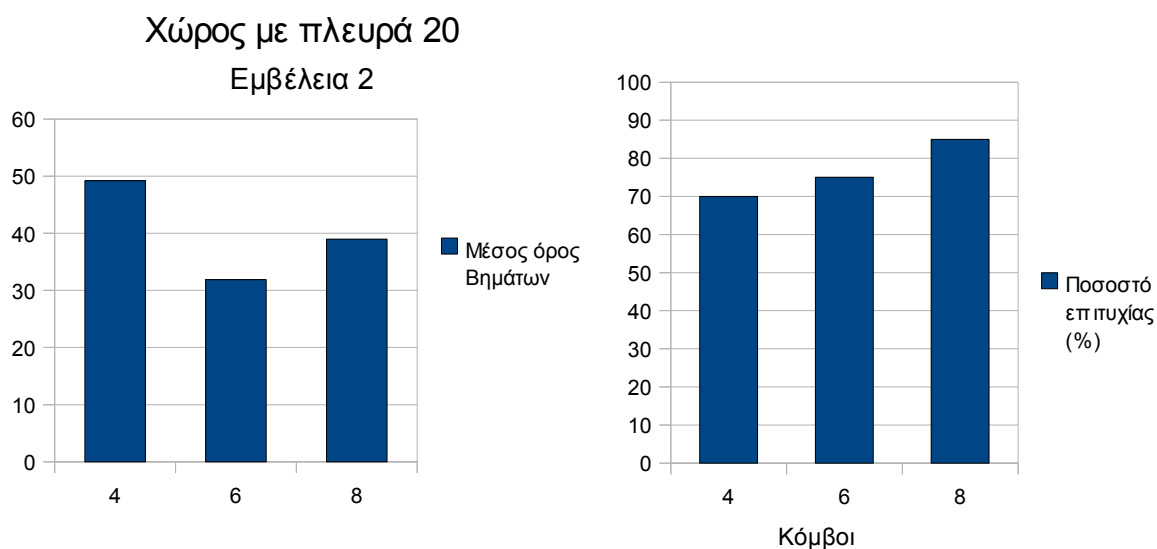
Από τα αποτελέσματα που έχουμε συλλέξει μέσω των δοκιμών που πραγματοποιήθηκαν μπορούμε πούμε ότι για περιπτώσεις με πολλούς κόμβους και μεγάλη εμβέλεια λήψης και παραλαβής για τους κόμβους ο αλγόριθμος δουλεύει πολύ καλά. Υπάρχει όμως πρόβλημα στις περιπτώσεις με τους λίγους κόμβους, που εντείνεται ακόμη περισσότερο όταν αυτοί δεν έχουν μεγάλη ακτίνα εμβέλειας. Για αυτό μπορεί να προσθέσουμε στον αλγόριθμο την βελτίωση για την οποία γίνεται εισήγηση στην ανάλυση των αποτελεσμάτων στο κεφάλαιο 4.4.3.

4.4.3 Αλγόριθμος για ομαδοποίηση κόμβων βάση όμοιων χαρακτηριστικών με κίνηση

Για τις ανάγκες αυτού του αλγορίθμου έγιναν ακριβώς τα ίδια σενάρια με το προηγούμενο. Δηλαδή δύο διαφορετικές μετρήσεις μια για χώρο περιμέτρου 80 τετραγώνων, με 4,6 και 8 κόμβους, με διαφορετικές εμβέλειες και μια με χώρο με περίμετρο 400 τετραγώνων, με 100 κόμβους, με εμβέλειες κόμβων 3 και 5 τετραγώνων. Αξίζει να σημειωθεί πως σε αυτή την περίπτωση, κάθε ομάδα παίρνει τους μισούς κόμβους από το σύνολο που χρησιμοποιούμε. Επίσης στην ανάλυση των αποτελεσμάτων θα έχουμε και μια μικρή μετατροπή του

αλγορίθμου για να μπορεί να τερματίζει πάντα και σύγκριση μεταξύ των αποτελεσμάτων που θα πάρουμε και αυτών με την πρώτη έκδοση.

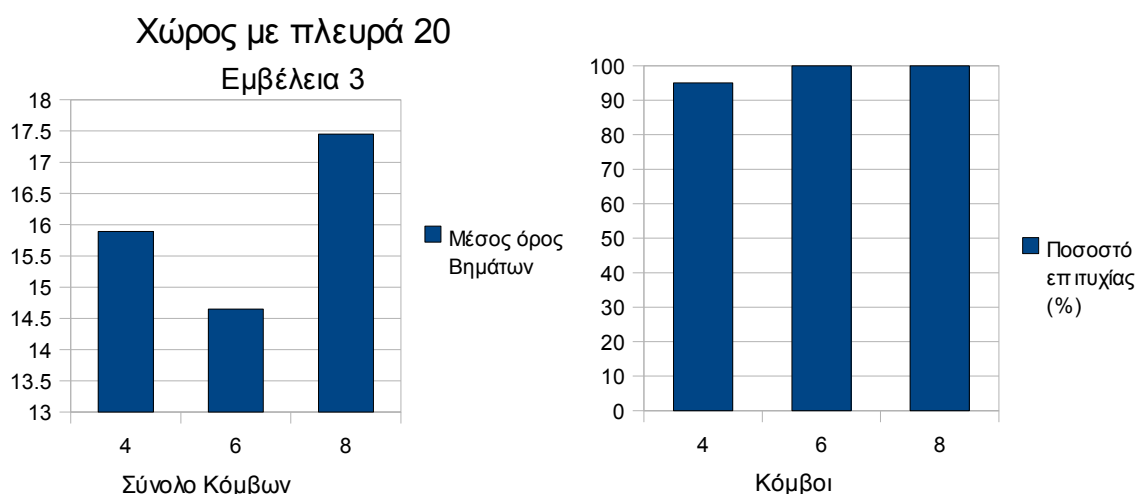
Σενάριο με χώρο περιμέτρου 80 τετραγώνων



Σχήμα 4.16: Γραφικές παραστάσεις για ομαδοποίηση 2 ομάδων σε χώρο 80 τετραγώνων

Εδώ βλέπουμε τις γραφικές παραστάσεις με τα αποτελέσματα από τις προσομοιώσεις για τον χώρο με περίμετρο 80 τετραγώνων για το σενάριο που όλοι οι κόμβοι έχουν εμβέλεια 2 τετράγωνα. Αριστερά έχουμε το γράφημα με τον μέσο αριθμό των βημάτων για τις περιπτώσεις με τους 4, 6 και 8 κόμβους στον χώρο μας. Παρατηρούμε τώρα πως έχουμε αισθητή διαφορά ανάμεσα στις 3 περιπτώσεις, με χειρότερη αυτή με τους 4 κόμβους, δηλαδή 2 για κάθε ομάδα, που χρειάζονται σχεδόν 50 βήματα ανά μέσο όρο οι κόμβοι κάθε φορά να ολοκληρωθεί το πρόβλημα, και με καλύτερη την περίπτωση με τους 6 κόμβους, με λίγο πάνω από 30 βήματα κάθε φορά. Το σενάριο με τους 8 κόμβους είναι περίπου στην μέση, με λίγο κάτω από 40 βήματα κάθε φορά. Έχουμε τώρα πολύ χειρότερα αποτελέσματα από ότι στο αλγόριθμο με την μια ομάδα, αφού πλέον έχουμε πολλές περισσότερες συγκρούσεις μεταξύ των κόμβων, αφού εκτός του ότι συγκρούονται με τους μη ομαδοποιημένους κόμβους της ομάδας τους, πλέον συγκρούονται και με τους κόμβους της άλλης ομάδας. Έτσι έχουμε πολύ μεγαλύτερες καθυστερήσεις για την ολοκλήρωση του αλγορίθμου. Τα πράγματα φυσικά είναι καλύτερα για τα σενάρια με τους περισσότερους κόμβους, αφού έχουμε αισθητά πιο χαμηλά αποτελέσματα από ότι στο σενάριο με τους 4.

Με μια ματιά και στην δεξιά γραφική παράσταση, αυτή δηλαδή που μας παρουσιάζει το πόσο αποτελεσματικοί είναι ο αλγόριθμος για κάθε προσομοίωση σε σχέση με το αν έφτασε στον σχηματισμό και των δύο ομάδων, μπορούμε να δούμε ότι έχουμε αποτελέσματα που κυμαίνονται από 70% για τους 4 κόμβους μέχρι λίγο πάνω από 80% για τους 8 κόμβους. Είναι καλύτερα από τον προηγούμενο αλγόριθμο για τον σχηματισμό μιας ομάδας, για τον ίδιο λόγο που αυτός ο αλγόριθμος είναι πολύ πιο αργός στην ολοκλήρωση του. Λόγω του μεγαλύτερου αριθμού συγκρούσεων, που συνεπάγεται πολλές αλλαγές στην κατεύθυνση. Από την άλλη όμως εξακολουθούμε να έχουμε αρκετά μεγάλο ποσοστό αποτυχίας εύρεσης σωστών αποτελεσμάτων, πράγμα που θα επιχειρήσουμε να λύσουμε στη συνέχεια.

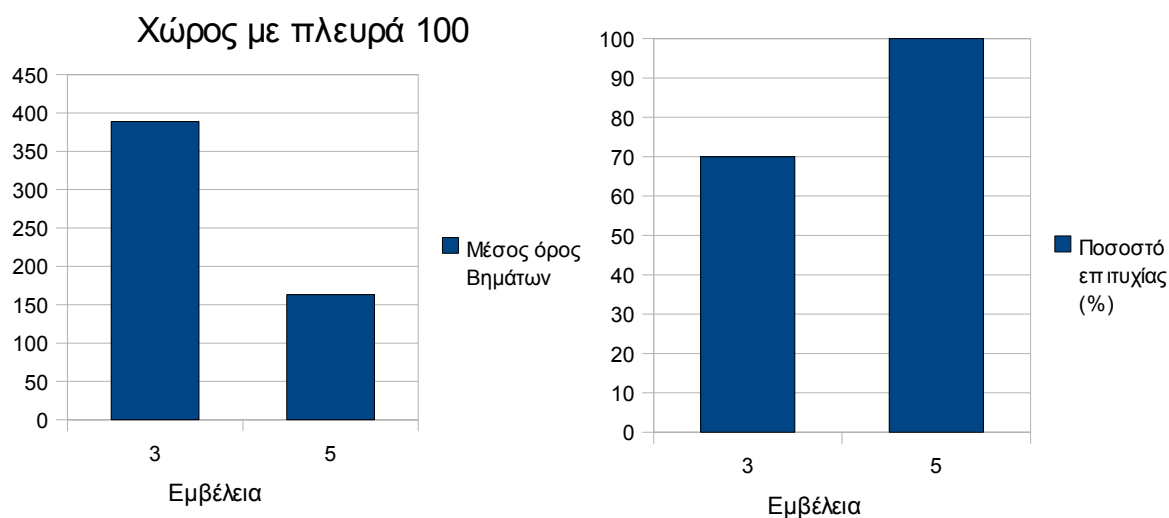


Σχήμα 4.17: Γραφικές παραστάσεις για ομαδοποίηση 2 ομάδων σε χώρο 80 τετραγώνων

Έχουμε τώρα τις γραφικές για το ίδιο σενάριο με εμβέλεια κόμβων αυτή τη φορά 3 τετράγωνα. Βλέπουμε στη γραφική παράσταση δεξιά ότι έχουμε τεράστια βελτίωση στον μέσο όρο βημάτων από ότι είχαμε στο σενάριο με την εμβέλεια των 2 τετραγώνων και πλέον μια πιο κοντινή απόσταση από τα αποτελέσματα του προηγούμενου αλγορίθμου. Έχουμε πάλι σαν καλύτερο αποτέλεσμα αυτό του σεναρίου με τους συνολικά έξι κόμβους, πράγμα πολύ λογικό.

Όταν παρατηρήσουμε και την γραφική παράσταση με τα ποσοστά επιτυχίας του αλγορίθμου, μπορούμε να δούμε πως εκτός από πολύ πιο γρήγορος από την προηγούμενη περίπτωση, ο αλγόριθμος είναι και πολύ πιο αποτελεσματικός για δίκτυα με κόμβους με πιο μεγάλη εμβέλεια. Η μόνη μικρή απώλεια, είναι ένα μικρό 5% στην περίπτωση που κάναμε την προσομοίωση με τους 4 κόμβους.

Σενάριο με χώρο περιμέτρου 400 τετραγώνων



Σχήμα 4.18: Γραφικές παραστάσεις για ομαδοποίηση 2 ομάδων σε χώρο 400 τετραγώνων

Πιο πάνω έχουμε τις γραφικές παραστάσεις με τα αποτελέσματα από τις προσομοιώσεις για τον χώρο με περίμετρο 400 τετραγώνων με 100 κόμβους. Παρατηρούμε από την γραφική παράσταση του μέσου αριθμού βημάτων στα αριστερά ότι με πιο μεγάλη εμβέλεια ο αλγόριθμος είναι κατά πολύ πιο γρήγορος από την περίπτωση όπου οι κόμβοι μας έχουν πιο μικρή εμβέλεια. Έχουμε χρόνο περισσότερο από 2 φορές πιο γρήγορο. Αυτό τονίζει ξανά το ότι ο αλγόριθμος που έχει αναπτυχθεί για αυτό το πρόβλημα είναι πολύ καλύτερος για περιπτώσεις που έχουμε μεγαλύτερες ακτίνες εμβέλειας στους κόμβους του δικτύου.

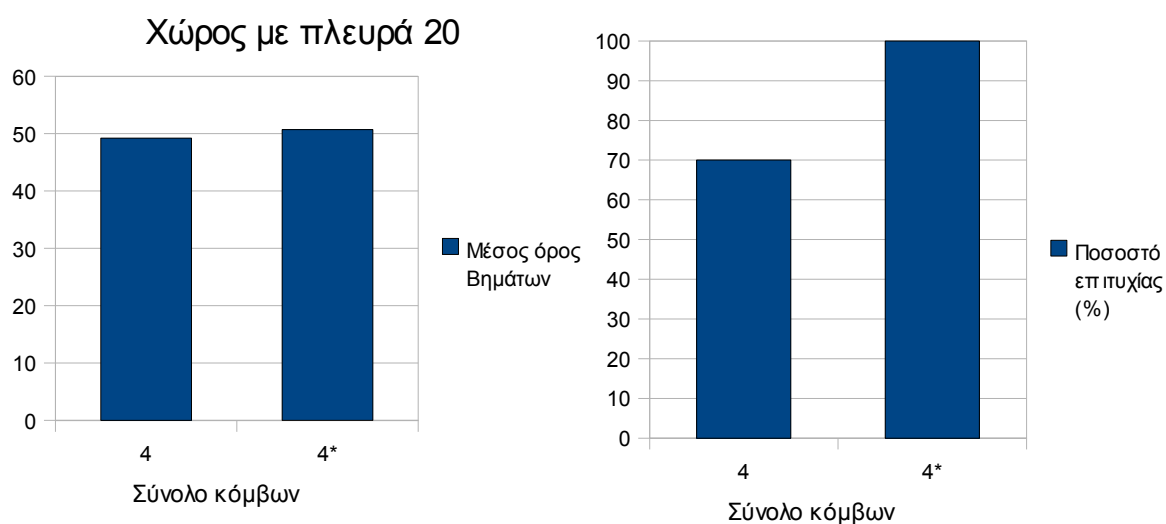
Ακόμη αν δούμε και την δεύτερη γραφική παράσταση στα δεξιά, βλέπουμε ότι εκτός από πιο γρήγορος, ο αλγόριθμος είναι και πιο αποδοτικός σε περιπτώσεις που έχουμε μεγαλύτερες εμβέλειες στους κόμβους μας, αφού όπως βλέπουμε έχουμε απόλυτη επιτυχία για το σενάριο με τα 5 τετράγωνα εμβέλεια σε σχέση με 70% επιτυχία στα σενάρια με τα 3.

Ανάλυση αποτελεσμάτων

Μέσα από τα σενάρια που τρέξαμε για τον αλγόριθμο ομαδοποίησης για ομάδες με κοινά χαρακτηριστικά, βλέπουμε ότι έχει πολύ καλά αποτελέσματα για μεγάλα δίκτυα, με κόμβους με μεγάλη εμβέλεια. Αντίθετα δεν είναι καθόλου καλά τα αποτελέσματα για τις περιπτώσεις όπου έχουμε δίκτυα με λίγους κόμβους και μικρή εμβέλεια σε αυτούς.

Επειδή όμως δεν θέλουμε τον αλγόριθμο να λειτουργεί μόνο σε μερικές περιπτώσεις θα κάνουμε μια μικρή μετατροπή. Βασικά αντί οι κόμβοι να συνεχίζουν συνέχεια στην ίδια πορεία, κάθε τακτά χρονικά διαστήματα θα αλλάζουν πορεία στον χώρο. Αυτό θα το κάνουν όμως μόνο οι κόμβοι οι οποίοι ακόμη δεν έχουν ομαδοποιηθεί πάνω στον αρχηγό και κινούνται μόνοι τους στον χώρο. Με αυτή την μικρή τροποποίηση σίγουρα θα αυξήσουμε τα ποσοστά επιτυχίας του αλγορίθμου. Για να το διαπιστώσουμε αυτό και στην πράξη, θα παρουσιαστούν στη συνέχεια αποτελέσματα από δύο διαφορετικά παραδείγματα, που θα συγκρίνουν τα αποτελέσματα του αλγορίθμου με την μικρή αυτή μετατροπή σε σχέση με τον παλιό.

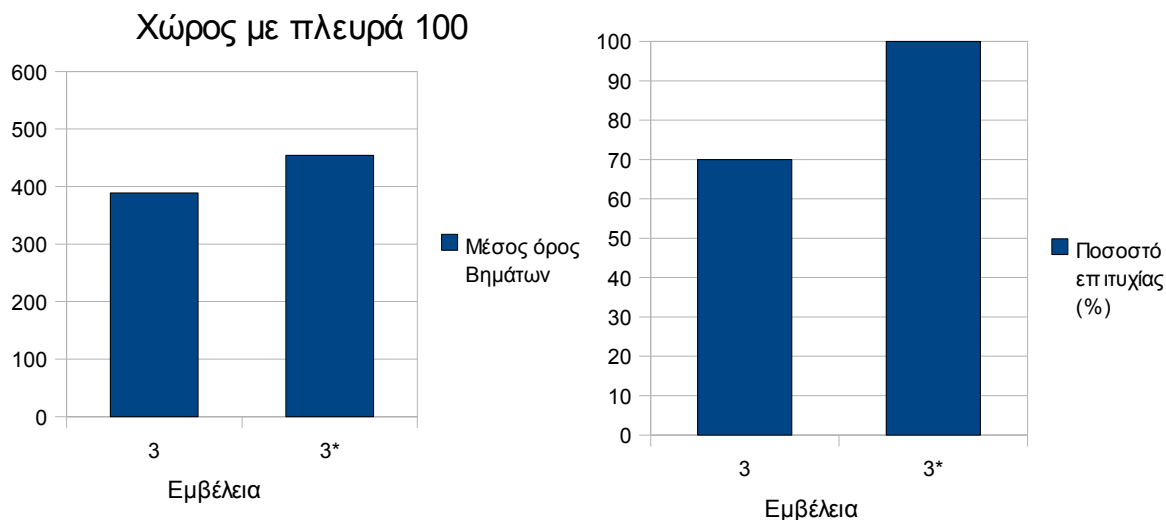
Σενάριο με χώρο περιμέτρου 80 τετραγώνων, εμβέλεια 2 και 4 κόμβους



Σχήμα 4.19: Γραφικές παραστάσεις για ομαδοποίηση 2 ομάδων σε χώρο 80 τετραγώνων με εισαγωγή βελτίωσης σε σύγκριση με πριν

Το 4* αντιπροσωπεύει το σενάριο με τον αλγόριθμο να έχει υποστεί την μικρή μετατροπή που έχει αναφερθεί λίγο πιο πάνω, στην παράγραφο με την ανάλυση των αποτελεσμάτων του κεφαλαίου 4.4.3, ενώ το 4 τα προηγούμενα αποτελέσματα. Βλέπουμε από την γραφική παράσταση δεξιά με το ποσοστό επιτυχίας ότι πλέον έχουμε πάντα ολοκλήρωση του αλγορίθμου πλέον με την προσθήκη της μικρής μετατροπής. Άρα επιτύχαμε τον σκοπό για τον οποίο εισάγαμε την μετατροπή. Από την άλλη μπορούμε να δούμε ότι προσθέσαμε ένα μικρό αντίτιμο στον αλγόριθμο, που εμφανίζεται με την μορφή μιας πολύ μικρής αύξησης στο μέσο όρο των βημάτων που χρειάζονται για να ολοκληρωθεί το πρόβλημα κάθε φορά. Όμως επειδή είναι τόσο πολύ μικρή, είναι σχεδόν αμελητέα, αφού το όφελος που μας προσφέρει σε αυτή την περίπτωση είναι πολύ μεγαλύτερο.

Σενάριο με χώρο περιμέτρου 400 τετραγώνων, εμβέλεια 3 και 100 κόμβους



Σχήμα 4.20: Γραφικές παραστάσεις για ομαδοποίηση 2 ομάδων σε χώρο 400 τετραγώνων με εισαγωγή βελτίωσης σε σύγκριση με πριν

Σε αυτές τις γραφικές παραστάσεις το 3* αντιπροσωπεύει το σενάριο με τον αλγόριθμο να έχει υποστεί την μικρή μετατροπή που έχει αναφερθεί λίγο πιο πάνω, στην παράγραφο με την ανάλυση των αποτελεσμάτων του κεφαλαίου 4.4.3, ενώ το 3 τα προηγούμενα αποτελέσματα. Παρατηρούμε πάλι ότι και σε αυτό το σενάριο η εισαγωγή της μετατροπής έχει κάνει τον αλγόριθμο να ολοκληρώνει πάντα, σε αντίθεση με το 70% των περιπτώσεων που είχαμε πριν. Ακόμη είναι εμφανές πως και σε πιο μεγάλο δίκτυο από πριν, έχουμε πάρα πολύ μικρή αύξηση στην μέση τιμή βημάτων που χρειάζεται ο αλγόριθμος μέχρι να ολοκληρώσει κάθε φορά. Από λίγο κάτω από 400 έχουμε τώρα περίπου 450. Αυτό φυσικά είναι πολύ καλό αποτέλεσμα.

Ανάλυση βελτίωσης

Από τα αποτελέσματα που είχαμε στην δοκιμή της βελτιωμένης έκδοσης του αλγορίθμου, μπορούμε να πούμε ότι πλέον έχουμε ένα πολύ καλό αλγόριθμο για όλα τα ήδη των δικτύων και όχι μόνο για τα μεγάλα που διαθέτουν κόμβους με μεγάλη ακτίνα εμβέλειας παραλαβής και λήψης δεδομένων.

Κεφάλαιο 5

Μελλοντική δουλεία

5.1 Γενικές βελτιώσεις-επεκτάσεις-εισηγήσεις

46

5.1 Γενικές βελτιώσεις-επεκτάσεις-εισηγήσεις

Υπάρχουν αρκετές βελτιώσεις που μπορούν να γίνουν στους αλγόριθμους που έχουν αναπτυχθεί σε αυτή την ατομική διπλωματική εργασία. Το πρώτο που θα ήταν καλό και κατά κάποιο τρόπο επιτακτικό να γίνει είναι η εφαρμογή τους σε πραγματικό δίκτυο αισθητήρων για να δούμε πόσο αποτελεσματικοί είναι όχι μόνο σε προσομοιώσεις αλλά και στην πραγματικότητα. Αυτό θα υποδείξει προβλήματα που μπορεί να υπάρχουν και να μην φαίνονται σε ένα περιβάλλον προσομοίωσης, όπως για παράδειγμα παρεμβολές που μπορεί να υπάρχουν μεταξύ των κόμβων-αισθητήρων. Επίσης με την χρήση του γραφικού περιβάλλοντος στη γλώσσα προγραμματισμού C, δεν μπορούμε να έχουμε κανένα αποτέλεσμα όσον αφορά το πόση ενέργεια καταναλώνεται από τους κόμβους και πόσο μπορούν να αντέξουν να πραγματοποιούν τα σενάρια που έχουν υλοποιηθεί. Φυσικά κατά το σχεδιασμό των αλγορίθμων έχει ληφθεί υπόψη το δεδομένο ότι δεν υπάρχει μεγάλη πηγή ενέργειας και αποφεύγονται σε όλες τις περιπτώσεις οι περιττές κινήσεις. Όμως αυτό δεν φτάνει για μια σωστή και έγκυρη εκτίμηση των τελικών ενεργειακών σπαταλών που θα γίνουν.

Και στις τρεις εφαρμογές που αναπτύχθηκαν ο χώρος πάντα ήταν ένα τετραγωνικό πλέγμα. Φυσικά στην πράξη είναι σχεδόν αδύνατο να βρούμε πρακτική χρήση αυτών των εφαρμογών με χώρο τόσο τετραγωνισμένο. Έτσι πρέπει οι αλγόριθμοι να μπορούν να αντεπεξέλθουν και σε ακανόνιστα σχήματα χώρου για να είναι αποτελεσματικοί στο φυσικό κόσμο και όχι μόνο σε προσομοιώσεις. Μια ακόμη σημαντική απλοποίηση που έχει γίνει για τις ανάγκες των προσομοιώσεων είναι η υπόθεση πως όλοι οι κόμβοι έχουν την ίδια εμβέλεια τόσο

αποστολής όσο και λήψης δεδομένων. Πάλι αυτό το σενάριο δεν μπορεί να θεωρηθεί εφικτό σε πραγματικές συνθήκες, και για αυτό πρέπει να γίνουν και εδώ αρκετές βελτιώσεις.

Από την άλλη θα ήταν καλό, για την περίπτωση των αλγορίθμων που ασχολούνται με την περιμετρική κάλυψη και την ομαδοποίηση κόμβων να ενοποιηθούν σε ένα υβριδικό αλγόριθμο. Αυτό θα μας βοηθούσε στο να έχουμε στην αρχή τυχαίο αριθμό κόμβων από δύο ομάδες στο χώρο. Η μια από τις δύο μόνο να είναι όμως υπεύθυνη για την περιμετρική κάλυψη του χώρου. Έτσι να ξεκινά πρώτα ο αλγόριθμος για την εύρεση των δύο διαφορετικών ομάδων και μετά, η ομάδα που πρέπει να κάνει την περιμετρική κάλυψη να συνεχίζει με τον αλγόριθμο που έχει αναπτυχθεί για αυτό τον σκοπό. Οι κόμβοι της άλλης ομάδας θα πρέπει είτε να μένουν στην θέση που έχουν όλοι πλέον εισαχθεί στην ομάδα είτε μπορούμε να τους εκμεταλλευτούμε με άλλο τρόπο, όπως για παράδειγμα να καλύψουμε την περιοχή μέσα στον χώρο που έχουμε. Να εισάγουμε δηλαδή και ένα αλγόριθμο κάλυψης της περιοχής και να ακροβολιστεί η δεύτερη ομάδα μέσα στο χώρο, στις κατάλληλες τοποθεσίες έτσι ώστε να ελέγχουν ολόκληρη την περιοχή, είτε να ελέγχουν όση περιοχή δεν μπορεί να ελεγχθεί από τους κόμβους που κάνουν την περιμετρική περιπολία.

Για τον ίδιο αλγόριθμο, θα μπορούσε να υπήρχε μια επέκταση. Να έχουμε και ένα κόμβο που να έμενε σε μια εσωτερική τροχιά, σε αντίστροφη φορά από τους κόμβους που εκτελούν την περιπολία. Έτσι θα μπορούσε αυτός να συλλέγει τις πληροφορίες που μαζεύουν οι υπόλοιποι και να τις αποστέλλει για επεξεργασία, για να μπορούμε για παράδειγμα να ελέγχουμε την περιοχή και εμείς από ένα κεντρικό σημείο. Αλλιώς θα μπορούσαμε να έχουμε ένα κόμβο sink τοποθετημένο σε ένα σημείο της κοντά στην περίμετρο σταθερά, που να συλλέγει τις πληροφορίες των κόμβων που εκτελούν την περιφρούρηση της περιοχής.

Μια επιπλέον βελτιστοποίηση για τον αλγόριθμο εύρεσης ομάδων κόμβων με κοινά χαρακτηριστικά είναι να γίνει προέκταση του έτσι ώστε να μπορεί να υποστηρίζει προσομοίωση περισσότερων από δύο ομάδων κόμβων. Ο αλγόριθμος που έχει αναπτυχθεί στην εργασία αυτή υποστηρίζει την ύπαρξη μόνο δύο ομάδων. Έτσι δεν θα έχουμε κανένα περιορισμό όσο αφορά το πόσες ομάδες θα μπορούμε να δοκιμάσουμε για να δούμε αν είναι αποδοτικός ο αλγόριθμος και σε πιο μεγάλους αριθμούς από αυτές.

Και οι τρεις αυτές εφαρμογές μπορούν να χρησιμοποιηθούν στην πράξη για διάφορες εφαρμογές. Σχετικά με τον αλγόριθμο της περιμετρικής κάλυψης, μπορεί να εφαρμοστεί κάλλιστα με τη βοήθεια ρομπότ, επίγειων ή εναέριων για τη φρούρηση διάφορων περιοχών

στο περιβάλλον, που η φύση τους δεν βοηθά την ανθρώπινη επιτήρηση. Έτσι πολύ εύκολα θα μπορούμε να ρίξουμε τα ρομπότ αυτά με τους κόμβους-αισθητήρες πάνω τους στην περιοχή από ένα αεροπλάνο, και αυτοί από μόνοι τους γνωρίζοντας τα όρια της περιοχής να ξεκινήσουν από μόνοι τους την περιπολία γύρω από την περιοχή και να αποστέλλουν τις πληροφορίες που θέλουμε χωρίς κόπο και χωρίς να έχουμε φόβο για ανθρώπινες απώλειες.

Από την άλλη ο αλγόριθμος για εύρεση ομάδων θα μπορούσε να χρησιμοποιηθεί για παράδειγμα σε στρατιωτικές εφαρμογές. Μπορούμε να έχουμε κατασκοπικά ρομπότ τα οποία να μπορούν να εισχωρούν σε αντίπαλες περιοχές και να περισυλλέξουν διάφορα δεδομένα. Όταν βρεθούν αντιμέτωπα με εχθρικές δυνάμεις, δηλαδή να αισθανθούν μέσα στην εμβέλεια τους οτιδήποτε εκτός από άλλα ρομπότ τις δικής τους ομάδας να αλλάζουν την κίνησή τους για να μην τα βρει ο αντίπαλος. Από την άλλη όταν βρουν τον αρχηγό της ομάδας να κινηθούν μαζί του έτσι ώστε να γίνει και η συλλογή των πληροφοριών και να μπορούν να επιστρέψουν πίσω όλα μαζί, φέροντας τις πολύτιμες τους πληροφορίες για επεξεργασία.

Βασική προϋπόθεση για την χρήση των αλγορίθμων όμως είναι η βελτίωση τους και ο πειραματισμός τους σε πραγματικές συνθήκες. Δεν μπορούν διαφορετικά να βγουν ασφαλή συμπεράσματα για το πόσο καλές και χρήσιμες μπορεί να φανούν.

Κεφάλαιο 6

Συμπεράσματα

6.1 Γενική Σύνοψη	49
6.2 Συμπεράσματα	50

6.1 Γενική σύνοψη

Η ατομική διπλωματική αυτή εργασία καταπιάστηκε με θέματα γύρω από το χώρο των ασύρματων δικτύων αισθητήρων. Πιο συγκεκριμένα είδαμε στην αρχή μια γενική αναφορά για το τι είναι ένα ασύρματο δίκτυο αισθητήρα και από τι είναι ένας κόμβος αισθητήρα. Αναλύθηκαν τα χαρακτηριστικά των δικτύων αυτών, σχέδια αλληλεπίδρασης καθώς και τομείς που χρησιμοποιούνται σήμερα τα WSNs. Για τους αισθητήρες είδαμε σε τι ομάδες μπορούμε να τους κατατάξουμε, την βασική αρχιτεκτονική τους και μια περιγραφή του κάθε ενός βασικού στοιχείου τους καθώς και επίσης βασικές τους λειτουργίες.

Είδαμε τα ανοικτά ζητήματα που υπάρχουν στο χώρο αυτό και επικεντρωθήκαμε στην κινητικότητα στα ασύρματα δίκτυα αισθητήρων, με την οποία και ασχολείται η εργασία αυτή. Παρουσιάστηκαν οι τρόποι με τους οποίους μπορεί να παρουσιαστεί η κινητικότητα μέσα σε ένα ασύρματο δίκτυο αισθητήρων και ποιες δυνατότητες προσδίδουν αυτοί στο δίκτυο.

Μετά αναλύσαμε τρεις διαφορετικές ομάδες αλγορίθμων που χρησιμοποιούν κινητικότητα και συγκεκριμένα αλγόριθμους για κάλυψη περιοχής, για έλεγχο περιμέτρου και αλγόριθμους για ομαδοποίηση κόμβων-αισθητήρων σε ένα δίκτυο. Μελετήσαμε διάφορους υπάρχον αλγορίθμους από κάθε κατηγορία που προτείνονται σε διάφορα άρθρα, για να πάρουμε μια ιδέα για το τι υπάρχει ήδη και για να μπορέσουμε στη συνέχεια να λύσουμε τα δικά μας προβλήματα.

Μετέπειτα αναλύθηκαν τα τρία διαφορετικά υποπροβλήματα που υπήρχαν για το σκοπό της εργασίας αυτής. Για κάθε ένα από αυτά παρουσιάστηκαν οι λύσεις τους υλοποιημένες σε γραφικό περιβάλλον με τη βοήθεια της γλώσσας C και αναλυτική περιγραφή του αλγόριθμου που χρησιμοποιήθηκε για να γίνουν. Έγινε ανάλυση των αποτελεσμάτων από δοκιμαστικές προσομοιώσεις που έγιναν και σχολιασμός των αποτελεσμάτων.

Τέλος αναφέρθηκε η μελλοντική δουλειά που μπορεί να γίνει σχετικά με τα θέματα που ασχολήθηκε η ατομική διπλωματική αυτή εργασία καθώς και τομείς που μπορεί να εφαρμοστούν οι συγκεκριμένοι αλγόριθμοι.

6.2 Συμπεράσματα

Από την μελέτη που έχει γίνει για τις ανάγκες αυτής της διπλωματικής εργασίας, μπορώ να πω ότι τα ασύρματα δίκτυα αισθητήρων είναι μια ενδιαφέρουσα και πολλά υποσχόμενη τεχνολογία. Βέβαια υπάρχουν ακόμη πολλά προβλήματα, ειδικά στα θέματα κατανάλωσης ενέργειας, που όμως με την συνεχή ανάπτυξη της τεχνολογίας τα πράγματα θα γίνονται όλο και καλύτερα. Ακόμη με την εισαγωγή της κινητικότητας μπορούμε να βοηθήσουμε πάρα πολύ την αποτελεσματικότητα και τις δυνατότητες των WSNs.

Με την κίνηση πλέον οι αισθητήρες μπορούν να κάνουν πολλά περισσότερα πράγματα και ακόμη μπορεί να τα εκτελέσουν και πολύ καλύτερα από το να είναι πάντα σταθεροί σε ένα σημείο. Έχουμε μια πιο σφαιρική και ολοκληρωμένη κατανόηση του γεγονότος που θέλουμε να μετρήσουμε με την κίνηση των κόμβων, αφού μπορούμε να κάνουμε πολύ περισσότερα πράγματα με την κίνηση.

Όσον αφορά την κάλυψη μιας περιοχής, βλέπουμε τώρα ότι μπορούμε πολύ εύκολα με την εισαγωγή της κινητικότητας σε ένα δίκτυο αισθητήρων, να καλύψουμε τρύπες που δημιουργούνται από τον θάνατο κόμβων του, καθώς επίσης να καλύψουμε τρύπες σε περιπτώσεις που έχουμε μια τυχαία αρχική τοποθέτηση, χωρίς μεγάλα προβλήματα και επιπτώσεις, με τον κατάλληλο αλγόριθμο κάθε φορά.

Για τον περιμετρικό έλεγχο μιας περιοχής, βλέπουμε ότι μπορούμε πλέον με λίγους κινητούς κόμβους να καλύψουμε πολύ πιο εύκολα μια περιοχή, που με τους συμβατικούς στατικούς κόμβους, θα χρειαζόμασταν τεράστιους αριθμούς από αυτούς. Φυσικά αυτό προϋποθέτει την ύπαρξη και κάποιων σημείων ελέγχου για να μπορούμε να συλλέγουμε τις μετρήσεις που

παίρνουν κατά την περιπολία τους οι κόμβοι, όμως αυτό δεν αποτελεί πρόβλημα αφού υπάρχουν πολύ απλές λύσεις.

Επίσης, αναφορικά με την ομαδοποίηση των κόμβων σε μια περιοχή, χωρίς την κινητικότητα αυτό είναι σχεδόν αδύνατο έως ακατόρθωτο. Με την ομαδοποίηση των κόμβων μπορούμε να πετύχουμε στη συνέχεια πολλά πράγματα. Αυτό γιατί τις περισσότερες φορές πρέπει να εγκαταστήσουμε ένα ασύρματο δίκτυο αισθητήρων σε χώρους όπου δεν έχει πρόσβαση άνθρωπος. Έτσι χρειάζεται για παράδειγμα να τους ρίξουμε από αεροπλάνο. Με ένα απλό αλγόριθμο ομαδοποίησης, όπως αυτό που αναπτύχθηκε στην εργασία αυτή θα μπορούμε να ομαδοποιήσουμε τους κόμβους, και μετά να τους αφήσουμε να συντονιστούν για να πραγματοποιήσουν τις μετρήσεις που θέλουμε.

Τα ασύρματα δίκτυα αισθητήρων μπορούν κάλλιστα στο άμεσο μέλλον να βοηθήσουν έμπρακτα τον άνθρωπο σε χιλιάδες πρακτικές εφαρμογές, προσδίδοντας νέα δεδομένα και νέες προκλήσεις σε πάρα πολλούς τομείς. Με τη συνεχή έρευνα σε αυτό το πεδίο μπορούμε να προκαλέσουμε μια γενική πρόοδο, που βάση την να έχει αυτούς τους μικρούς κόμβους-αισθητήρες. Βασική προϋπόθεση φυσικά είναι να μπορούμε να εκμεταλλευτούμε τις ευκαιρίες που μας δίνει η τεχνολογία και να τις ενσωματώσουμε στα δίκτυα αυτά.

Βιβλιογραφία

- [1] A. Almeida, G. Ramalho, H. Santana, P. Tedesco, T. Menezes, V. Corruble, and Y. Chevaleyre. “*Recent advances on multi-agent patrolling*”. Lecture Notes in Computer Science, 3171:474–483, 2004
- [2] A. Ephremides, J.E. Wieselthier and D. J. Baker, “*A Design concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling*”, Proceeding of IEEE, Vol. 75, No. 1, pp. 56-73, 1987.
- [3] A. Howard, M. J Mataric, and G. S. Sukhatme. “*Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem.*” In 6th International Symposium on Distributed Autonomous Robotics Systems (DARS02), June 2002.
- [4] C. Huang and Y. Tseng. “*The Coverage Problem in a Wireless Sensor Network*”, September 2003.
- [5] D. J. Baker and A. Ephremides, “*The Architectural Organization of a Mobile Radio Network via a Distributed Algorithm*”, IEEE Transactions on Communications, Vol. 29, No. 11, pp. 1694-1701, November 1981.
- [6] G. Amitabha. “*Estimating Coverage Holes and Enhancing Coverage in Mixed Sensor Networks*”, 2004.
- [7] G. Wang, G. Cao, and T. La Porta.” *A bidding protocol for deploying mobile sensors.*” In 11th IEEE International Conference on Network Protocol ICNP ‘03, pp. 315–324, Nov 2003.
- [8] G. Wang, G. Cao, and T. La Porta. “*Movement-assisted sensor deployment*”, In IEEE INFOCOM 2004, June 2004.
- [9] H. Chan, A. Perrig, “*ACE: An Emergent Algorithm for Highly Uniform Cluster Formation*”. In 2004 European Workshop on Sensor Networks. pp. 154-171.

- [10] J. S. Shieh and T. W. Calvert. “*View and route planning for patrol and exploring robots*”. *Advanced Robotics*, 6(4):399–430, 1992.
- [11] K. Holger and A. Willig, “*Protocols and Architectures for Wireless Sensor Networks*”, John Wiley and Son, 2007
- [12] M. Chatterjee, S. K. Das, and D. Turgut, “*WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks*”, *Journal of Cluster Computing*, Special issue on Mobile Ad hoc Networking, No. 5, 2002, pp. 193-204.
- [13] N. Heo and P. K. Varshney.” *An intelligent deployment and clustering algorithm for a distributed mobile sensor network.*” In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pp. 4576–4581, Oct 2003.
- [14] P. Paruchuri, J. P. Pearce, M. Tambe, F. Ordonez, and S. Kraus. “*An efficient heuristic approach for security against multiple adversaries*”. In *AAMAS*, 2007.
- [15] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. “*Security in multiagent systems by policy randomization*”. In *AAMAS*, 2007.
- [16] S. Basagni, “*Distributed Clustering for Ad Hoc Networks*”, in *Proceedings of International Symposium on Parallel Architectures, Algorithms and Networks*, pp. 310-315, June 1999.
- [17] S. Ganeriwal, A. Kansal, and M. B. Srivastava.” *Self aware actuation for fault repair in sensor networks.*” In *Proceedings of the IEEE International Conference on Robotics and Automation(ICRA)*, May 2004.
- [18] T. Yan, T. He, and J. Stankovic. “*Differentiated surveillance for sensor networks.*”, In *Proceedings of the ACM SenSys 03*, Nov 2003.
- [19] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, “*Energy-efficient communication protocols for wireless microsensor networks*”, in: *Proceedings of the 33rd Hawaiian International Conference on Systems Science (HICSS-33)*, Maui, HI, USA, 4–7 January 2000.

- [20] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless and C. Gill. “*Integrated coverage and connectivity configuration in wireless sensor networks.*” In Proceedings of the ACM SenSys ‘03, pp 28–39, Nov 2003.
- [21] Y. Chevaleyre. “*Theoretical analysis of the multi-agent patrolling problem*”. In Proceedings of Intelligent Agent Technology (IAT), 2004.
- [22] Y. Elmaliach, N. Agmon, and G. A. Kaminka. “*Multi-robot area patrol under frequency constraints*”. In ICRA, 2007.

Παράρτημα Α

Οδηγός εγκατάστασης TinyOS για τρέξιμο ενδεικτικής εφαρμογής (Surge_Reliable) πάνω σε αισθητήρες Micaz

Οι οδηγίες που ακολουθούν είναι ενδεικτικές, και είναι από την προσωπική μου εμπειρία με το λειτουργικό αυτό. Μπορεί να μην ενδείκνυται να γίνουν για άλλες εφαρμογές αφού οι εφαρμογές δεν τρέχουν σε όλες τις εκδόσεις του λειτουργικού. Πριν να ξεκινήσετε θα ήταν καλό πρώτα να διαβάσετε το “Getting Started Guide” της Crossbow για να μπειτε στο νόημα. Η εγκατάσταση έχει γίνει από μέρους μου σε λειτουργικό Windows Xp, με τη χρήση του Cygwin.

Προμηθευτείτε το CD TinyOS+Surge.

Εγκαταστήστε το tinyos-1.1.0-1is.exe που βρίσκεται πάνω στο CD.

Αναβαθμίστε το σε tinyos-1.1.7(4-7)

Αντιγράψτε το tinyos-1.1.7July2004cvs-1.cygwin.noarch.rpm (μπορείτε να το κατεβάσετε από την ιστοσελίδα <http://www.tinyos.net/>) στο φάκελο \$ROOT/opt, όπου \$ROOT ο φάκελος όπου είναι εγκατεστημένο το tinyos1.1.0.

Πηγαίνετε στο \$ROOT/opt, και εκτελέστε την ακόλουθη εντολή στο Cygwin

rpm --force --ignoreos -Uvh tinyos-1.1.7July2004cvs-1.cygwin.noarch.rpm

Πηγαίνετε στο \$ROOT/opt, μετονομάστε το tinyos-1.x σε tinyos-1.1.0, και το tinyos-1.1.7 σε tinyos-1.x.

Αναβάθμιση εφαρμογών

Αντιγράψτε το Xbow.tgz στο \$ROOT/opt/tinyos-1.x/contrib, και αποσυμπιέστε το. Θα δείτε τον υποκατάλογο xbow.(αν μην μπορείτε να βρείτε αυτό το αρχείο υπάρχει πάνω στο CD με την ατομική διπλωματική μου εργασία)

Όλες οι καινούριες εφαρμογές πρέπει να βρίσκονται στον φάκελο \$ROOT/opt/tinyos-1.x/contrib.xbow/opt/apps.

Αναβαθμίστε τα environment variable:

alias mib510 = “MIB510=/dev/ttyS<n>”,

<n> είναι το serial port number, 0-COM1, 1-COM2.

Κάνετε Compile το Surge_Reliable και φορτώστε το στα motes:

cd \$ROOT/opt/tinyos-1.x/contrib.xbow/opt/apps/Surge_Reliable

(αλλάξτε το Makefile, φύγετε τα σχόλια για το micaz βάλτε σε σχόλια την γραμμή για το mica2)

Εκτελέστε μια από τις δύο ακόλουθες εντολές

make micaz install.0 mib510, COM1

make micaz reinstall.0 mib510, COM1

Ο αριθμός μετά το install είναι το node number για τα motes. Για την εφαρμογή μας το 0 είναι για το base station. Οι άλλοι κόμβοι παίρνουν οτιδήποτε μη μηδενικές τιμές.

Το group ID καθορίζεται στο \$ROOT/opt/tinyos-1.x/contrib./xbow/apps/MakeXbowLocal.

Εγκαταστήστε το Surge-View και τον SerialForwarder. Χρειάζονται για να μπορέσετε να τρέξει και να δείτε γραφικά τα αποτελέσματα της εφαρμογής. Βρίσκονται και αυτά πάνω στο CD της διπλωματικής μου εργασίας.

Για να τρέξετε τον SerialForwarder, πηγαίνετε στον φάκελο του Surge-View, κάνετε double click στο SerialForwarder. Και για mote communications γράψετε : serial@COM<n>:57600. Ο αριθμός <n> πρέπει να είναι ότι έχετε χρησιμοποιήσει στο make + 1.

Για να τρέξετε το Surge-View, ξεκινήστε ένα Command Prompt Window, πηγαίνετε στον φάκελο του Surge-View, και πληκτρολογήστε

Surge-View 7 or Surge-View 7 > log-file,

όπου 7 είναι το group ID που υπάρχει μέσα στο configuration file MakeXbowLocal.

Αν όλα πάνε καλά θα έχετε στην οθόνη σας τα αποτελέσματα σε λίγη ώρα. Προσέξτε ότι χρειάζονται λίγα λεπτά για να δείτε τις πρώτες μετρήσεις. Αν παραδόξως υπάρχουν προβλήματα μπορείτε να βρείτε μερικές λύσεις στην σελίδα <http://www.mobilab.unina.it/TinyOSMAC.htm>