ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ

Τμήμα Ηλεκτρονικών Μηχανικών & Μηχανικών Υπολογιστών

“ΠΑΡΑΛΛΗΛΗ ΕΠΕΞΕΡΓΑΣΙΑ ΠΟΛΛΑΠΛΩΝ ΣΥΝΑΘΡΟΙΣΤΙΚΩΝ ΕΠΕΡΩΤΗΣΕΩΝ ΣΕ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ ΜΕ ΚΙΝΟΥΜΕΝΟΥΣ ΚΟΜΒΟΥΣ”

Μπούσγος Σεραφείμ

Επιτροπή:
Δεληγιαννάκης Αντώνιος (Επιβλέπων)
Λαγουδάκης Μιχαήλ
Μανιά Αικατερίνη

Ευχαριστίες

Θα ήθελα πρωτίστως να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Δεληγιαννάκη Αντώνιο για την υποστήριξή του, την καθοδήγηση και τις συμβουλές του αλλά και για την εμπιστοσύνη που έδειξε στις ικανότητές μου. Επίσης, θα ήθελα να ευχαριστήσω τον Επίκουρο Καθηγητή κ. Λαγουδάκη Μιχαήλ και την Επίκουρη Καθηγήτρια κ. Μανιά Αικατερίνη για τη συμμετοχή τους ως μέλη της εξεταστικής επιτροπής.

Για τη φιλία και τη στήριξή τους θα ήθελα να ευχαριστήσω και όλους τους φίλους και συναδέλφους που ήταν δίπλα μου σε όλες τις φάσεις της ζωής μου και με στήριξαν ανιδιοτελώς σε μεγάλο βαθμό.

Τέλος, θα ήθελα να πω ένα μεγάλο ευχαριστώ και να εκφράσω την αγάπη και την ευγνωμοσύνη μου στους γονείς μου και όλη την οικογένειά μου, που είναι πάντα δίπλα μου και με στηρίζουν στα όνειρα και τις αποφάσεις μου.

"Την διπλωματική αυτή εργασία την αφιερώνω στους γονείς μου.Σας ευχαριστώ πολύ για την αγάπη σας."

Περιεχόμενα

Περίληψη

Ο τομέας των Ασύρματων Δικτύων Αισθητήρων παρουσιάζει τα τελευταία χρόνια πολύ μεγάλη ζήτηση σε ερευνητικό επίπεδο. Η εισαγωγή νέων τεχνολογιών και η ανάπτυξη των ήδη υπαρχουσών έδωσε πρόσφορο έδαφος για μεγάλες αλλαγές και βελτιώσεις σε αυτό τον τομέα. Στα δίκτυα αισθητήρων μία από τις απαιτήσεις είναι η μακροβιότητα του δικτύου, το οποίο επιτυγχάνεται με την ελαχιστοποίηση της κατανάλωσης ενέργειας σε κάθε κόμβο αισθητήρα. Πολλοί ερευνητές επικεντρώνονται στον σχεδιασμό αλγορίθμων δρομολόγησης που θα καταναλώνουν λιγότερη ενέργεια και επομένως, θα παρατείνουν τη διάρκεια ζωής του δικτύου .

Η διπλωματική αυτή εργασία ασχολείται με την παράλληλη επεξεργασία πολλαπλών συναθροιστικών επερωτήσεων σε ένα δίκτυο αισθητήρων με κινούμενους κόμβους. Ο στόχος λοιπόν είναι, τόσο να βρεθούν οι αποδοτικές διαδρομές που ελαχιστοποιούν το κόστος επικοινωνίας στην εκτέλεση πολλών συναθροιστικών επερωτήσεων, με την κατάλληλη επεξεργασία και την σωστή δρομολόγηση των μετρήσεων των κόμβων, όσο και να μειωθεί ο αριθμός των αστοχιών σύνδεσης μεταξύ των κόμβων (link failures). Σε αυτή τη διπλωματική εργασία παρουσιάζουμε έναν αλγόριθμο (STS-Mobile) που επιδιώκει να ελαχιστοποιήσει την κατανάλωσης ενέργειας, λαμβάνοντας υπόψιν του τόσο την ταχύτητα των κόμβων, όσο και το κόστος αναδιοργάνωσης του δικτύου σε περιπτώσεις απωλειών σύνδεσης. Οι συνδέσεις που δημιουργούνται μεταξύ των κόμβων των υποδέντρων βασίζονται κυρίως στον χρόνο που διαρκούν. Σκοπός λοιπόν της εργασίας αυτής είναι η βέλτιστη διαμόρφωση ενός δικτύου με κινούμενους κόμβους έτσι ώστε να παρατηρούνται λιγότερα link failures, η συγκέντρωση των αποτελεσμάτων με όσο το δυνατόν λιγότερα μηνύματα και η βέλτιστη αναδιαμόρφωση του δικτύου όταν link failures λαμβάνουν χώρα. Μια λεπτομερής πειραματική αξιολόγηση παρουσιάζει τα οφέλη των προτεινόμενων τεχνικών.

Εισαγωγή

Με τον όρο ασύρματο δίκτυο αισθητήρων εννοούμε ένα πλήθος υπολογιστικών κόμβων, οι οποίοι διαθέτουν μικρό φυσικό μέγεθος, δυνατότητες ασύρματης επικοινωνίας και φέρουν αισθητήρες που τους επιτρέπουν να παρατηρούν διάφορα φυσικά μεγέθη. Το πλήθος των κόμβων, οι υπολογιστικές τους δυνατότητες, το μεγεθός τους, το είδος των αισθητήρων τους κτλ., είναι όλοι τους παράγοντες που μεταβάλλονται και μπορούν να διαφέρουν ανάλογα με την εφαρμογή για την οποία προορίζονται. Έτσι, μια εφαρμογή ασύρματων αισθητήρων μπορεί να χρησιμοποιήσει από μερικές δεκάδες έως και χιλιάδες τέτοιους κόμβους και το μεγεθός τους μπορεί να ποικίλει από μικροσκοπικό έως ενός μικρού υπολογιστικού συστήματος.

Ένας τυπικός τρόπος της εξαγωγής πληροφοριών από ένα δίκτυο αισθητήρων ξεκινάει με τη διάδοση των επερωτήσεων από έναν κόμβο βάσης (gateway) προς τους κόμβους αισθητήρων, ζητώντας τους να ελέγξουν περιοδικά το περιβάλλον, και να επιστρέψουν τα συνολικά αποτελέσματα σε τακτά διαστήματα, συχνά αποκαλούμενα ως εποχές. Στην εργασία αυτή επικεντρωνόμαστε στην εκτέλεση πολλαπλών επερωτημάτων συνάθροισης. Το δίκτυο αισθητήρων εκτελεί τη συνάθροιση καθοδηγώντας τα στοιχεία από τους αισθητήρες πηγής μέσω των ενδιάμεσων κόμβων προς την gateway. Στην περίπτωση αυτή είναι χρήσιμο για τους κόμβους που βρίσκονται στην ίδια ομάδα (απαντάνε στο ίδιο υποσύνολο επερωτήσεων), να οργανωθούν ανάλογα στην τοπολογία του δικτύου, δηλαδή να χωριστούν σε ομάδες. Το πλεονέκτημα της προσέγγισης αυτής είναι ότι οι κόμβοι με παιδιά της ίδιας ομάδας μπορούν να συναθροίσουν τις τιμές των παιδιών τους σε μία τελική τιμή [26].

Μια γενική αντίληψη, ότι οι κόμβοι αισθητήρων είναι στατικοί και παραμένουν σταθερά στη θέση τους, κυριαρχούσε για μεγάλο χρονικό διάστημα. Ωστόσο,ορισμένες εφαρμογές, όπως παρακολούθηση των οικοτόπων, της άγριας πανίδας, περιπτώσεις διάσωσης, ιατρικής περίθαλψης και περιπτώσεις αντιμετώπισης καταστροφών, έχουν επιτρέψει την κινητικότητα σε WSN. Η κίνηση όμως των κόμβων προκαλεί προβλήματα στην δομή των δέντρων, αφού παρατηρούνται πολλές αστοχίες σύνδεσης μεταξύ των κόμβων. Το κόστος διάδοσης των επερωτήσεων στο δίκτυο υποτίθεται ότι έχει ένα δευτεροβάθμιο ρόλο για τις μακροπρόθεσμες επερωτήσεις, δεδομένου ότι η διάδοση επερώτησης πραγματοποιείται μία φορά, ενώ η διάδοση αποτελέσματος εμφανίζεται επανειλημμένα στους περιοδικούς κύκλους (εποχές) και οι αστοχίες σύνδεσης είναι ένα συχνό φαινόμενο.

Στην εργασία αυτή αρχικά ορίζουμε τι είναι αισθητήρας, τι είναι ένα δίκτυο αισθητήρων, παρουσιάζουμε εφαρμογές τους και αναλύουμε τα προβλήματα και τους περιορισμούς τους. Στην συνέχεια, αναφερόμαστε στην κινητικότητα στα ασύρματα δίκτυα αισθητήρων, στις μορφές κινητικότητας, στις δυνατότητες που προσφέρονται από την κίνηση των κόμβων και παρουσιάζουμε συγκεκριμένα μοντέλα κινούμενων κόμβων. Έπειτα, αναλύουμε τις γλώσσες επερωτήσεων και επικεντρωνόμαστε κυρίως στις συναρτήσεις συνάθροισης. Στο 4^{ο} κεφάλαιο παρουσιάζουμε τρείς αλγορίθμους που χρησιμοποιούν τεχνικές συνάθροισης μέσα στο δίκτυο, με σκοπό την ελάχιστη κατανάλωση ενέργειας. Οι αλγόριθμοι αυτοί (ECReduced, STG και STS) δεν λαμβάνουν υπόψιν τους την κίνηση των κόμβων, αλλά μόνο την τοποθεσία τους και τον αριθμό των hops. Αντίθετα, ο προτεινόμενος αλγόριθμος (STS-Mobile) που αναλύουμε στο Κεφάλαιο 5, βασίζεται στην κίνηση των κόμβων και στον χρόνο που δύο κόμβοι μπορούν να είναι συνδεδεμένοι. Για να αναδείξουμε τη λειτουργικότητα του αλγορίθμου STS-Mobile προσομοιώσαμε την λειτουργία του, όπως αναλύουμε στο Κεφάλαιο

6, και τον συγκρίνουμε με τον αλγόριθμο STS. Η προσομοίωση περιλαμβάνει και γραφικό περιβάλλον, όπου μπορούμε να δούμε τη φάση διαμόρφωσης του δικτύου, τη φάση διάδοσης αποτελέσματος και την αναδιαμόρφωσή του σε περιπτώσεις αστοχίας σύνδεσης. Τέλος, στο Κεφάλαιο 7 παρουσιάζονται τα γραφήματα των πειραματικών μετρήσεων, όπου φαίνονται τα πλεονεκτήματα του αλγορίθμου STS-Mobile.

Κεφάλαιο 1-Δίκτυο Αισθητήρων

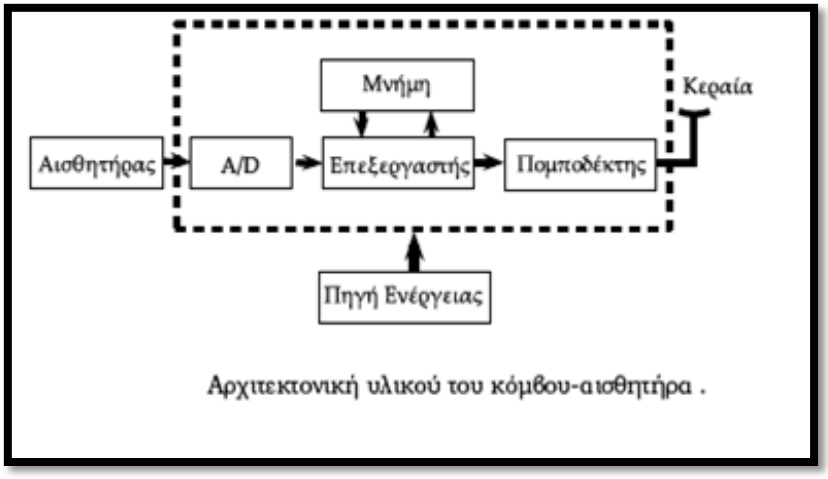
1.1 Αισθητήρες

Είναι σημαντικό να αποσαφηνίσουμε τη διαφορά ενός αισθητήρα και ενός κόμβου του δικτύου που έχει ενσωματωμένο κάποιο αισθητήρα. Ο αισθητήρας από μόνος του είναι η συσκευή εκείνη που αναλαμβάνει τη μετατροπή των σημάτων του φυσικού κόσμου σε ηλεκτρικά σήματα. Οι αισθητήρες αποτελούν ένα πολύ σημαντικό κομμάτι στην επιστήμη των ηλεκτρονικών, εφόσον συνδέουν οποιοδήποτε κύκλωμα ελέγχου με το εξωτερικό περιβάλλον. Τα πεδία εφαρμογής βρίσκονται όπου απαιτείται η αλληλεπίδραση ενός κυκλώματος ελέγχου ή ενός αυτοματισμού με το εξωτερικό περιβάλλον. Οι συχνότερες εφαρμογές είναι σε βιομηχανικούς αυτοματισμούς, σε συστήματα ελέγχου αυτοκινήτων, πλοίων, αεροπλάνων, σε αυτοματισμούς οικιακών συσκευών κλπ. Παραδοσιακά οι αισθητήρες αποτελούν αρκετά ογκώδεις συσκευές με συνηθέστερους τους αισθητήρες θερμοκρασίας, πίεσης, ροής και τους ανιχνευτές ακτινοβολίας. Τις τελευταίες δεκαετίες, λόγο της ανάπτυξης της τεχνολογίας της μικρομηχανικής, η ανάπτυξη τόσο των αισθητήρων, όσο και των εφαρμογών τους έχει γνωρίσει ραγδαία αύξηση (με προοπτικές ακόμα μεγαλύτερης ανάπτυξης). Η μικρομηχανική αποτελεί την επέκταση της μικροηλεκτρονικής στον χώρο των αισθητήρων, εκμεταλλευόμενη τις τεράστιες τεχνολογικές δυνατότητες που η μικροηλεκτρονική προσφέρει. Με την ανάπτυξη των μικρομηχανικών τεχνικών, αφενός βελτιστοποιήθηκαν τα χαρακτηριστικά των αισθητήρων και αφετέρου μειώθηκε δραματικά το κόστος και το μέγεθός τους.΄Ετσι, κατέστη εφικτή η ανάπτυξη νέων πρωτότυπων μικρομηχανικών δομών, σχεδιασμένων για ειδικές εφαρμογές όπως αυτές των δικτύων αισθητήρων που μελετάμε στην παρούσα εργασία.΄Ενας αισθητήρας έχει την δυνατότητα να μετρήσει κάποιο χαρακτηριστικό του χώρου που τον περιβάλλει όπως:

- Θερμοκρασία
- Υγρασία
- Χημική σύσταση
- Κίνηση, δονήσεις
- ΄Ηχος
- Συνθήκες φωτός

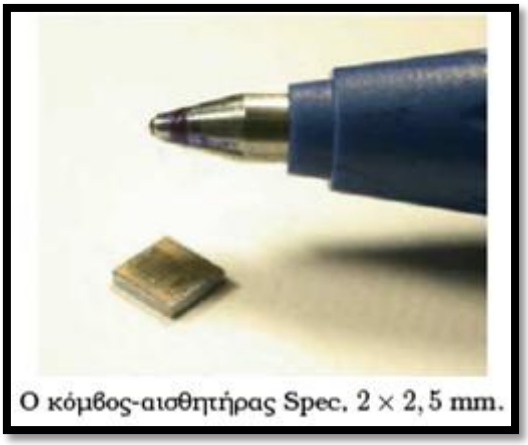
Στα πλαίσια των δικτύων αισθητήρων, ο όρος αισθητήρας συνήθως χρησιμοποιείται για να υποδηλώσει μια λειτουργική μονάδα του δικτύου που εκτός από τη λειτουργία της αίσθησης συμπεριλαμβάνει όλο εκείνο το υλικό και λογισμικό που την καθιστούν αυτόνομο κόμβο του δικτύου.

Μια γενική εικόνα της αρχιτεκτονικής του υλικού ενός κόμβου-αισθητήρα παρουσιάζεται στην Εικόνα 1.



Εικόνα 1

Παρατηρούμε ότι τα βασικότερα τμήματα του υλικού ενός κόμβου-αισθητήρα είναι το σύστημα μετάδοσης, οι αισθητήρες , η μονάδα επεξεργασίας και η μνήμη. Το σύστημα μετάδοσης είναι το πλέον σημαντικό σύστημα ενός κόμβου,μιας και αποτελεί το τμήμα με την μεγαλύτερη κατανάλωση ενέργειας, επηρεάζοντας έτσι την απόδοση του κόμβου,αλλά και τη συνολική απόδοση του δικτύου. Θέματα που απασχολούν την έρευνα στον τομέα του συστήματος μετάδοσης σχετίζονται με την ακτίνα εκπομπής, τον τύπο διαμόρφωσης που χρησιμοποιείται καθώς και τον ρυθμό μετάδοσης δεδομένων.Οι αισθητήρες αποτελούν τη διεπαφή εκείνη που αναλαμβάνει τη μετατροπή των σημάτων του φυσικού κόσμου σε μορφή για τις ηλεκτρονικές συσκευές. Κατά την επιλογή ενός αισθητήρα πρέπει να λαμβάνονται υπόψη οι απαιτήσεις της εφαρμογής για την οποία έχει αναπτυχθεί το δίκτυο,ο ρυθμός δειγματοληψίας του αισθητήρα και οι απαιτήσεις σε ενέργεια. Ένας κόμβος μπορεί να περιλαμβάνει περισσότερους του ενός αισθητήρες. Ο μετατροπέας αναλογικού σε ψηφιακό (Analog/Digital A/D) αναλαμβάνει τη μετατροπή των αναλογικών σημάτων του αισθητήρα σε ψηφιακά, ώστε να αξιοποιηθούν στη συνέχεια από τη μονάδα επεξεργασίας.Η μονάδα επεξεργασίας διαδραματίζει καθοριστικό ρόλο στην λειτουργία και τη συμπεριφορά ενός κόμβου-αισθητήρα και του δίνει τη δυνατότητα εκτέλεσης πολύπλοκων λειτουργιών. Στους σύγχρονους μικροελεγκτές ενσωματώνονται μνήμες τύπου flash και RAM, A/D μετατροπείς και ψηφιακά I/O σε ένα ολοκληρωμένο κύκλωμα χαμηλού κόστους. Η επιλογή του μικροελεγκτή στηρίζεται επιπλέον σε παράγοντες,όπως η κατανάλωση ενέργειας, οι απαιτήσεις σε τάση λειτουργίας,το κόστος,η υποστήριξη περιφερειακών,ο χρόνος αφύπνισης και η ταχύτητά του.



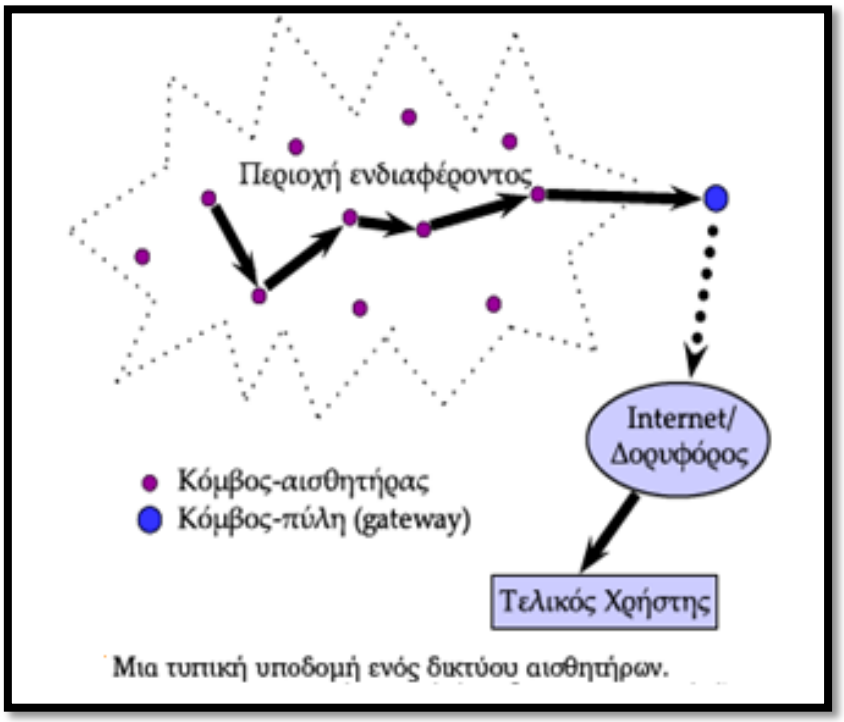
Εικόνα 2

Εικόνα 3

Τα λειτουργικά συστήματα για τους ασύρματους κόμβους-αισθητήρες είναι λιγότερα σύνθετα από τα γενικής χρήσης λειτουργικά συστήματα και λόγω των μικρότερων απαιτήσεων των εφαρμογών, αλλά και των περιορισμένων διαθέσιμων πόρων που εισάγει το υλικό των δικτύων αισθητήρων. Διάφορα λειτουργικά συστήματα έχουν αναπτυχθεί για τον σκοπό αυτό αναφέροντας ενδεικτικά τα παρακάτω TinyOS [1] , Contiki [2] , MANTIS [3] , Btnut [4] , και το nanoRK [5]. Το TinyOS αποτελεί το γνωστότερο λειτουργικό σύστημα ειδικά σχεδιασμένο για ασύρματα ενσωματωμένα συστήματα. Έχει ένα προγραμματιστικό μοντέλο προσαρμοσμένο για εφαρμογές «οδηγούμενες από συμβάντα» (eventdriven), ενώ ο πυρήνας του απαιτεί από κοινού μόνο 400 bytes κώδικα και μνήμης δεδομένων. Όλη η οργανωτική δομή του TinyOS, οι βιβλιοθήκες και οι εφαρμογές είναι γραμμένες στη γλώσσα προγραμματισμού NesC. Η γλώσσα NesC, η οποία αναπτύχθηκε από ερευνητές του Πανεπιστημίου του Berkeley, αντιπροσωπεύει ένα νέο πολλά υποσχόμενο πεδίο για τους σχεδιαστές εφαρμογών δικτύων αισθητήρων. Είναι κατάλληλα σχεδιασμένη για ενσωματωμένα συστήματα δικτύων και υποστηρίζει ένα προγραμματιστικό μοντέλο που ενσωματώνει αντιδραστικότητα με το περιβάλλον και δυνατότητες ελέγχου της επικοινωνίας.

1.2 Ορισμός ενός Wireless Sensor Network

Ένα ασύρματο δίκτυο αισθητήρων (Wireless Sensor Network [6]) , όπως βλέπουμε και στην Εικόνα 4 ,είναι ένα ασύρματο δίκτυο που αποτελείται από αυτόνομες συσκευές, κατανεμημένες στο χώρο, οι οποίες χρησιμοποιούν αισθητήρες με σκοπό τη συλλογική α-πεικόνιση και επεξεργασία φυσικών ή περιβαλλοντολογικών μεγεθών, όπως η θερμοκρασία, ο ήχος, η δόνηση, η πίεση, ή η κίνηση σε διάφορες τοποθεσίες. Η ανάπτυξη των ασύρματων δικτύων αισθητήρων παρακινήθηκε αρχικά από στρατιωτικές εφαρμογές, όπως την παρακολούθηση των πεδίων βολής. Πλέον χρησιμοποιούνται από το ευρύτερο κοινό σε μία πληθώρα εφαρμογών, που περιλαμβάνει παρακολούθηση περιβάλλοντος και κατοικίας, ιατρικές εφαρμογές, οικιακούς και βιομηχανικούς αυτοματισμούς και έλεγχο κυκλοφορίας.Ένα τυπικό δίκτυο αισθητήρων αποτελείται πολλές φορές από χιλιάδες τέτοιους κόμβους, κατανεμημένους στο χώρο που παρακολουθούν κάποιο φαινόμενο, είτε τυχαία, είτε σύμφωνα με κάποια προκαθορισμένη κατανομή. Παρόλα αυτά, η συνήθως δυναμική φύση των δικτύων αυτών (όσον αφορά στην υποδομή), προϋποθέτει ότι τα πρωτόκολλα και οι αλγόριθμοι των δικτύων αισθητήρων πρέπει να διαθέτουν αυτό-οργανωτικές δυνατότητες βασιζόμενες στην συνεργατική λειτουργία των επιμέρους κόμβων. Το κόστος των κόμβων είναι κυμαινόμενο, ανάλογα με το μέγεθος του δικτύου και την πολυπλοκότητα των μεμονωμένων κόμβων. Οι περιορισμοί μεγέθους και κόστους έχουν σαν αποτέλεσμα αντίστοιχους περιορισμούς στην ενέργεια, τη μνήμη και την υπολογιστική ισχύ.



Εικόνα 4

1.3 Εφαρμογές δικτύων αισθητήρων

Η τεχνολογία των Ασύρματων Δικτύων Αισθητήρων μπορεί να εφαρμοστεί σε πολλές εφαρμογές του πραγματικού κόσμου λόγω των παρακάτω χαρακτηριστικών:

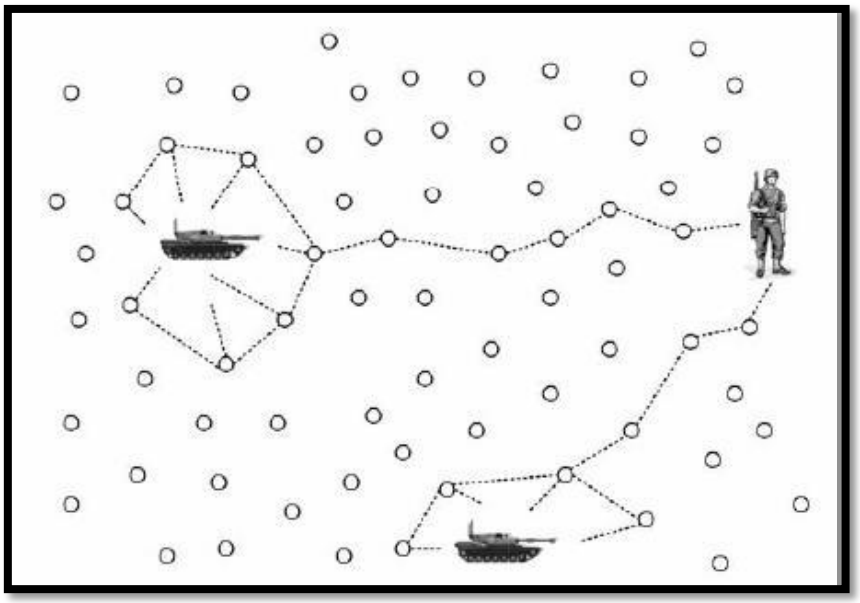
Ανάπτυξη οπουδήποτε και οποτεδήποτε. Τα ασύρματα δίκτυα περιέχουν κόμβους που δεν χρειάζονται ανθρώπινη παρακολούθηση για τη σωστή λειτουργία τους. Η τοποθέτηση των κόμβων μπορεί να γίνει και στις πιο επικίνδυνες περιοχές, ενώ η αποστολή τους μπορεί να επιτευχθεί σε οποιοδήποτε χρόνο.
Μεγαλύτερη αντοχή στα σφάλματα. Αυτό επιτυγχάνεται με την πυκνή ανάπτυξη του δικτύου WSN. Αν ένα μικρό ποσοστό κόμβων σταματήσει να λειτουργεί, τότε το δίκτυο μπορεί ακόμα να παράγει ικανοποιητικά αποτελέσματα.
Βελτιωμένη ακρίβεια. Ένας μικρός αριθμός μικροσκοπικών κόμβων μπορεί να έχει μεγαλύτερη ακρίβεια από έναν μεγαλύτερο κόμβο.
Μικρότερο κόστος. Λόγω του μικρότερου μεγέθους και της χαμηλότερης τιμής, ένα δίκτυο WSN είναι πιο οικονομικό από τα ενσύρματα δίκτυα αισθητήρων. Ένας άλλος παράγοντας που επηρεάζει την τιμή του δικτύου είναι η ευκολία ανάπτυξής τους.

Α. Στρατιωτικές εφαρμογές
Τα δίκτυα αισθητήρων έχουν γίνει αναπόσπαστο κομμάτι των στρατιωτικών επιχειρήσεων στα συστήματα διαταγών, ελέγχου, επικοινωνιών, υπολογισμών, πληροφοριών, επίβλεψης, αναγνώρισης και στόχευσης. Στο πεδίο της μάχης, δημιουργείται μια τάση οι στόχοι να γίνονται μικρότεροι σε μέγεθος, λιγότερο αναγνωρίσιμοι, με μεγαλύτερη ταχύτητα και να κινούνται συνήθως σε πολύ εχθρικό περιβάλλον. Για να μπορέσουμε να γνωρίζουμε την θέση και τη δύναμη των εχθρικών δυνάμεων, μπορούμε να τοποθετήσουμε πυκνές παρατάξεις αισθητήρων κοντά στον υποτιθέμενο στόχο. Λόγω των ικανοτήτων τους να είναι μη ελεγχόμενα από ανθρώπους, της εύκολης ανάπτυξης, της αυτό-οργάνωσης και της αντοχής σε σφάλματα, τα δίκτυα αισθητήρων μπορούν να παρέχουν άφθονα και διασταυρωμένα δεδομένα. Επίσης, οι αισθητήρες μπορούν να διασπαρθούν με αεροπορικά μέσα, με πυραύλους και τορπίλες ώστε να ξεπεράσουν κάποια εμπόδια και να οδηγηθούν στο ακριβές σημείο ανίχνευσης για την πιο αποτελεσματική εκπλήρωση της αποστολής τους. Η καταστροφή ορισμένων από τους αισθητήρες δεν επηρεάζει τη στρατιωτική επιχείρηση σε τέτοιο βαθμό, λόγω της πυκνής ανάπτυξης και της δυνατότητας αυτό-οργάνωσης. Ορισμένες χαρακτηριστικές εφαρμογές των δικτύων αισθητήρων είναι οι παρακάτω:

- Παρακολούθηση εξοπλισμού και πυρομαχικών.
- Παρακολούθηση του πεδίου της μάχης.
- Αναγνώριση των εχθρικών δυνάμεων και του εδάφους.
- Κατάδειξη στόχων [7] .
- Εκτίμηση ζημιών.
- Ανίχνευση και αναγνώριση μολυσμένης περιοχής.

Σενάριο

- Στρατιώτες κινούνται εντός κατοικημένης περιοχής(Εικόνα 5) χωρίς την γνώσης της
- Πιο πριν αισθητήρες ρίχνονται στην περιοχή
- Οι αισθητήρες συνεργάζονται
- Σε ανίχνευση στόχων, δεδομένα μεταφέρονται στους στρατιώτες με ακριβή πληροφορία της τοποθεσίας



Εικόνα 5

Β. Περιβαλλοντολογικές εφαρμογές.

Με τη διασπορά χιλιάδων μικροσκοπικών αισθητήρων σε μια γεωγραφική περιοχή, μπορούμε να παρακολουθούμε ή να ελέγχουμε το περιβάλλον. Έτσι μπορούμε να ανιχνεύουμε πλημμύρες, να επιβλέπουμε τον αέρα και την παροχή νερού, να ελέγχουμε τοπικά το κλίμα, να επιβλέπουμε τις καλλιέργειες για πιθανό κίνδυνο καταστροφών, να ανιχνεύουμε πυρκαγιές, να εξερευνούμε για αποθέματα μεταλλευμάτων κ.ά.

Γ.Έλεγχος οικοσυστήματος.

Τα ιδιαίτερα χαρακτηριστικά των δικτύων αισθητήρων (αυτό-οργάνωση, έλλειψη ελέγχου από τον άνθρωπο, πυκνή ανάπτυξη) επιτρέπουν την χρησιμοποίησή τους στον έλεγχο των οικοσυστημάτων [8] ,γιατί παρέχουν πληροφορίες σε πολλές περιβαλλοντολογικές καταστάσεις. Έτσι,εξασφαλίζεται η μακροχρόνια αναγνώριση, καταγραφή και ανάλυση των ενδιαφερόμενων γεγονότων. Η μακροχρόνια συλλογή δεδομένων μπορεί να βοηθήσει τους επιστήμονες να αναγνωρίσουν,να εντοπίσουν και να ανιχνεύσουν φαινόμενα σε περιοχές ενδιαφέροντος.

Εικόνα 6

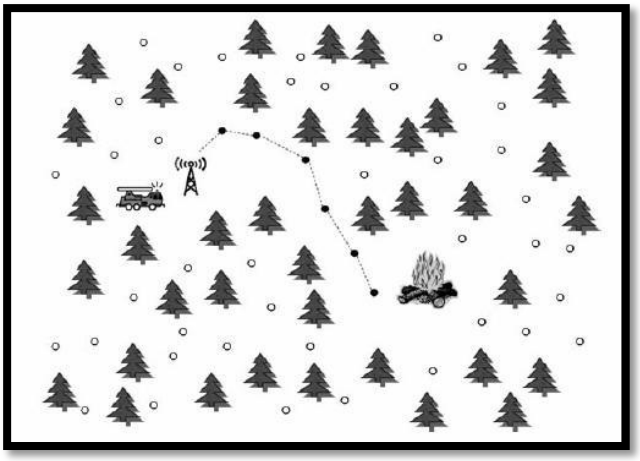
Δ.Έλεγχος κλίματος σε μεγάλα κτηριακά συγκροτήματα.

Η συνεχής δημιουργία όλο και μεγαλύτερων κτηρίων και εγκαταστάσεων (όπως εμπορικών κέντρων, ουρανοξυστών κ.ά.), έχει δημιουργήσει την ανάγκη για καλύτερο έλεγχο του κλίματος στο εσωτερικό τους [9]. Έτσι, μία περιήγηση σε ένα μεγάλο εμπορικό κέντρο μας δείχνει ότι η θερμοκρασία δεν είναι παντού ιδανική, π.χ. αλλού είναι χαμηλή και αλλού υψηλή, αλλού έχει υψηλότερη υγρασία και αλλού όχι. Για αυτούς και για άλλους λόγους υγιεινής πρέπει να εξασφαλίσουμε ένα ευχάριστο χώρο. Η δημιουργία τόσο ενσύρματων όσο και ασύρματων δικτύων σε αυτούς τους χώρους είναι ένας τρόπος ανίχνευσης και αντιμετώπισης των προβλημάτων που αναφέραμε. Συνήθως προτιμάται η ανάπτυξη WSN,διότι είναι πιο εύκαμπτα από τα ενσύρματα.

Ε. Πρόληψη καταστροφών και παροχή βοήθειας

Τα δίκτυα αισθητήρων μπορούν να είναι αποτελεσματικά σε επείγουσες καταστάσεις και περιοχές καταστροφής . Η ακριβής ανίχνευση μιας περιοχής που εκτελείται από τα δίκτυα αισθητήρων μπορεί να είναι κρίσιμη σε επιχειρήσεις διάσωσης, όπως ανεύρεση θυμάτων, εκτίμηση κινδύνου και αναγνώριση ή εντοπισμός παγιδευμένου προσωπικού. Για παράδειγμα, τα δίκτυα αισθητήρων μπορούν να αναπτυχθούν σε μεγάλα κτίρια κατά την κατασκευή των κτιρίων, να ριχτούν στην περιοχή διάσωσης, ή να χρησιμοποιηθούν ήδη τοποθετημένοι αισθητήρες σε μια περιοχή καταστροφής. Είναι επίσης χρήσιμο να αναπτυχθούν δίκτυα αισθητήρων για αποστολές ανίχνευσης μακράς διάρκειας, όπως ανίχνευση και παρακολούθηση αστοχίας υλικού, ώστε να παρθούν κατάλληλα μέτρα για την αποφυγή ατυχημάτων. Παρόλα τα σημαντικά μέτρα που λαμβάνονται για την ανίχνευση φωτιάς, οι φωτιές σε ακαλλιέργητες και δασικές περιοχές δημιουργούν μεγάλες καταστροφές κάθε χρόνο, τόσο σε άψυχο όσο και έμψυχο υλικό. Επειδή οι καιρικές συνθήκες κατά τη διάρκεια μιας φωτιάς είναι προβλέψιμες, μπορούμε εύκολα να προβλέψουμε την ύπαρξη μιας πυρκαγιάς(Εικόνα 7) κατά την περίοδο επικινδυνότητας για φωτιά. Λόγω της τυχαίας και της πυκνής ανάπτυξής τους, τα δίκτυα αισθητήρων είναι μια καλή επιλογή για την ανίχνευση και αναφορά για φωτιά. Διασκορπίζοντας μαζικά δίκτυα αισθητήρων σε επικίνδυνες περιοχές μπορεί να γίνει πιο αποδοτική η ειδοποίηση για φωτιά και η προέλευσή της.Επίσης ,

χρησιμοποιούνται και εντοπισμό άλλων φυσικών καταστροφών όπως οι πλημμύρες.Για παράδειγμα αναφέρουμε το σύστημα "ALERT" [10] που αναπτύχθηκε από τις ΗΠΑ για την υπηρεσία του σκοπού αυτού.



Εικόνα 7

ΣΤ. Ιατρική φροντίδα

Τα δίκτυα αισθητήρων είναι χρήσιμα στην παροχή άμεσης και αποτελεσματικής ιατρικής βοήθειας και θα οδηγήσουν σε ένα πιο υγιές περιβάλλον για τον άνθρωπο. Ορισμένες από τις χρήσεις σ' αυτό το πεδίο περιλαμβάνουν την απομακρυσμένη ανίχνευση ιών. Πολλές περιοχές, βασανιζόμενες από ασθένειες, είναι φτωχές σε αξιόπιστες τηλεπικοινωνίες. Η ανάπτυξη μεγάλου αριθμού ασύρματων αισθητήρων σε τέτοιες περιοχές, μπορεί να βοηθήσει στη συλλογή και μετάδοση σημαντικών πληροφοριών, όπως μια ασθένεια και τα χαρακτηριστικά του μολυσμένου πληθυσμού, η αναγνώριση χαρακτηριστικών της περιοχής, ο έλεγχος περιβαλλοντολογικών συνθηκών όπως η υγρασία και το ύψος της βροχής που επιτρέπουν την εξάπλωση ιών και νοσογόνων οργανισμών. Τα δίκτυα αισθητήρων μπορούν να χρησιμοποιηθούν για την πρόβλεψη ξεσπάσματος πολλών μεταδοτικών ασθενειών.Επίσης,χρησιμοποιούνται για την ολοκληρωμένη παρακολούθηση ασθενών. Η χρήση συσκευών αισθητήρων για την ανίχνευση πιθανών μολυσμένων ατόμων μπορεί να είναι μια αποτελεσματική μέθοδος για την αποφυγή εξάπλωσης μεταδοτικών ασθενειών. Επιπλέον, οι γηραιότεροι πολίτες μπορούν να φέρουν ασύρματους αισθητήρες(Εικόνα 8) [11] [12] , οι οποίοι παρακολουθούν συνεχώς τους χτύπους της καρδιάς, την πίεση κ.ά. Σε αντικανονικές καταστάσεις, ένας προειδοποιητικός ήχος υπενθυμίζει τον ασθενή να ειδοποιήσει τον γιατρό του ή μία αυτόματη υπενθύμιση στέλνεται στο κέντρο υγείας. Σημαντικό για τους αισθητήρες αυτούς είναι ότι είναι ασύρματοι και έτσι προσφέρουν ελευθερία κίνησης στους ασθενείς.Επιπλέον επιτυγχάνουν την παρακολούθηση των ασθενών στο οικείο περιβάλλον τους και όχι στα νοσοκομεία βελτιώνοντας έτσι την ποιότητα ζωής τους. Επιπλέον,τα δίκτυα αισθητήρων μπορούν να χρησιμοποιηθούν για την παρακολούθηση της εύρυθμης λειτουργίας ενός νοσοκομείου ή ενός κέντρου υγείας. Έτσι, αισθητήρες τοποθετούνται σε καίρια σημεία των κτιρίων,ώστε να ελέγχουν την παροχή ρεύματος, την παροχή οξυγόνου, την θερμοκρασία δωματίων κ.ά.Μπορεί επίσης να χρησιμοποιηθούν και για την παρακολούθηση ηλικιωμένων

Εικόνα 8

Ζ. Οικιακές εφαρμογές

Τα δίκτυα αισθητήρων μπορούν να παίξουν σημαντικό ρόλο στη δημιουργία ενός πιο βολικού και έξυπνου χώρου διαμονής για τον άνθρωπο. Ορισμένες χρήσιμες εφαρμογές δίδονται παρακάτω:

Καταμέτρηση αγαθών από απόσταση: Τα δίκτυα αισθητήρων μπορούν να χρησιμοποιηθούν στην καταμέτρηση εξ' αποστάσεως των αγαθών γενικής χρήσης (όπως το νερό, ο ηλεκτρισμός κ.ά.) και στη συνέχεια να μεταδώσουν τα αποτελέσματα μέσω ασύρματων συνδέσεων .Ένα απλό παράδειγμα αυτής της χρήσης είναι η τοποθέτηση ασύρματων αισθητήρων στα παρκόμετρα που θα ειδοποιούν τους χρήστες τους για την λήξη του χρόνου παρκαρίσματος.

Έξυπνος χώρος: Με τις σύγχρονες τεχνολογικές ανακαλύψεις, γίνεται δυνατή η ενσωμάτωση διάφορων ασύρματων αισθητήρων σε έπιπλα και οικιακές συσκευές, οι οποίοι συνεργάζονται μεταξύ τους και δημιουργούν ένα αυτόνομο δίκτυο. Για παράδειγμα, ένα έξυπνο ψυγείο μπορεί δημιουργήσει ένα μενού ανάλογα με τα υπάρχοντα αγαθά και να μεταδώσει τις σχετικές παραμέτρους ψησίματος σε ένα έξυπνο φούρνο,ο οποίος θα επιλέξει την κατάλληλη θερμοκρασία και τον χρόνο ψησίματος.

Adaptive House [13]: Το σπίτι μαθαίνει τις συμπεριφορές των ενοίκων και προσαρμόζει τις πηγές ενέργειας.Λόγου χάρη,ανάβει το καλοριφέρ,όταν εκτιμάει ότι έρχεται σπίτι ο ένοικος, ελέγχει το φωτισμό ανάλογα με το στάδιο της ημέρας και την τοποθεσία του ενοίκου κ.α.

Η. Επιστημονικές εξερευνήσεις

Η αποτελεσματική ανάπτυξη και λειτουργία των αυτορρυθμιζόμενων δικτύων αισθητήρων ανοίγει νέους ορίζοντες στην εξερεύνηση σε υψηλότερα αλλά και χαμηλότερα περιβάλλοντα, όπως το διάστημα και οι αβαθείς ωκεανοί. Έτσι, η εξερεύνηση των ωκεανών μπορεί να γίνει εύκολη με διασπορά αισθητήρων σε μεγάλα βάθη,οι οποίοι θα συλλέγουν πληροφορίες και θα τις μεταδίδουν στον απομακρυσμένο σταθμό βάσης.

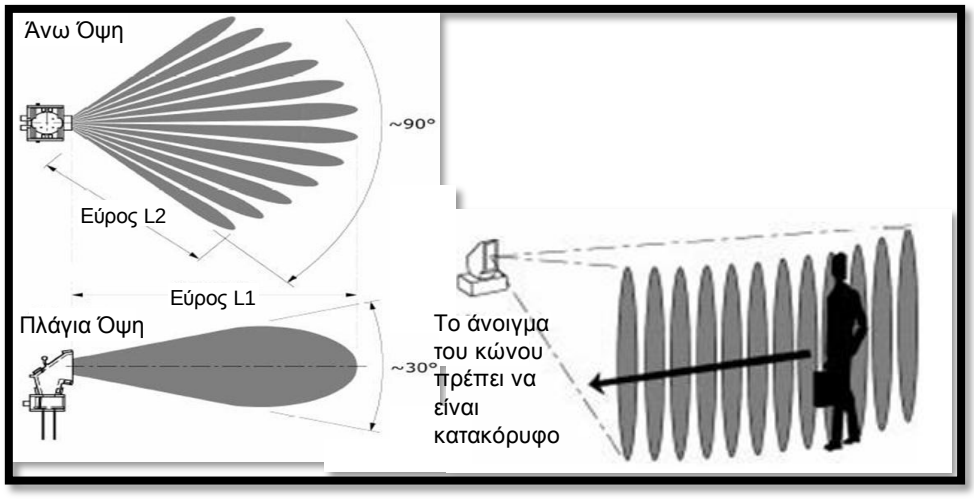
Θ. Αλληλεπίδραση με το περιβάλλον

Αναπτύσσοντας φθηνές και μικροσκοπικές συσκευές αισθητήρων, ελεγκτών κίνησης σε παιχνίδια και άλλα παιδικά αντικείμενα μπορούμε να δημιουργήσουμε έξυπνα νηπιαγωγεία για να προάγουμε την παιδική ανάπτυξη. Αυτά τα συστήματα παρέχουν ένα περιβάλλον με αλληλεπίδραση ατόμου-φυσικού κόσμου και όχι των κλασικών μορφών αλληλεπίδρασης ατόμου-ατόμου ή ατόμου-υπολογιστή. Αυτό γιατί επιτρέπει προσωπική αντίληψη για κάθε παιδί, προσαρμογή στη δυναμική για τις δραστηριότητες των παιδιών και μόνιμη και διακριτική συλλογή πληροφοριών για τις ενέργειες των παιδιών. Επίσης, τα δίκτυα αισθητήρων μπορούν να χρησιμοποιηθούν για τη δημιουργία αλληλεπιδρώντων μουσείων. Τα παιδιά μπορούν να συμμετέχουν ενεργά στα πειράματα και να λαμβάνουν απαντήσεις με το άγγιγμα ενός αντικειμένου που είναι εφοδιασμένο με αισθητήρες. Τέλος, μία ακόμα εφαρμογή των δικτύων αισθητήρων είναι η χρήση τους ως πλατφόρμες ψηφοφορίας .

Ι. Επίβλεψη

Η άμεση και από απόσταση επίβλεψη αποτελεί έναν σημαντικό τομέα όπου έχουν εφαρμογή τα συστήματα αισθητήρων. Για παράδειγμα, ένας μεγάλος αριθμός ακουστικών αισθητήρων μπορεί να χρησιμοποιηθεί για να ανιχνευθούν και να ανακαλυφθούν στόχοι και εισβολείς σ' έναν ορισμένο χώρο ασφαλείας. Τα δίκτυα αισθητήρων μπορούν να αναπτυχθούν σε μεγάλα κτίρια, κατοικημένες περιοχές, αεροδρόμια, σιδηροδρομικούς σταθμούς κ.α. για να ανιχνεύουν εισβολείς και να αναφέρουν στο κέντρο ελέγχου άμεσα,ώστε η ανακάλυψή τους να γίνει γρήγορα. Παρόμοια, η ανάπτυξη αισθητήρων καπνού σε στρατηγικά επιλεγμένες θέσεις των σπιτιών, γραφείων ή εργοστασίων είναι σημαντική στην πρόληψη κατά των πυρκαγιών και στην ανίχνευση της εξάπλωσης μιας πυρκαγιάς.Για τη διευκόλυνση των επιχειρήσεων Αναζήτησης και Διάσωσης προσώπων σχεδιάστηκε και υλοποιήθηκς το σύστημα "CenWits" (Connection-less sensor-based trackingsystem using witness) [14] .Ο στόχος του συστήματος αυτού είναι να μπορεί να προσφέρει τη πλέον πρόσφατη πληροφορία για το που εθεάθη το αγνούμενο πρόσωπο.

PIR: Διαφορικός αισθητήρας. Ανιχνεύει στόχους καθώς διασχίζουν τις ακτίνες που παράγει(Εικόνα 9)



Εικόνα 9

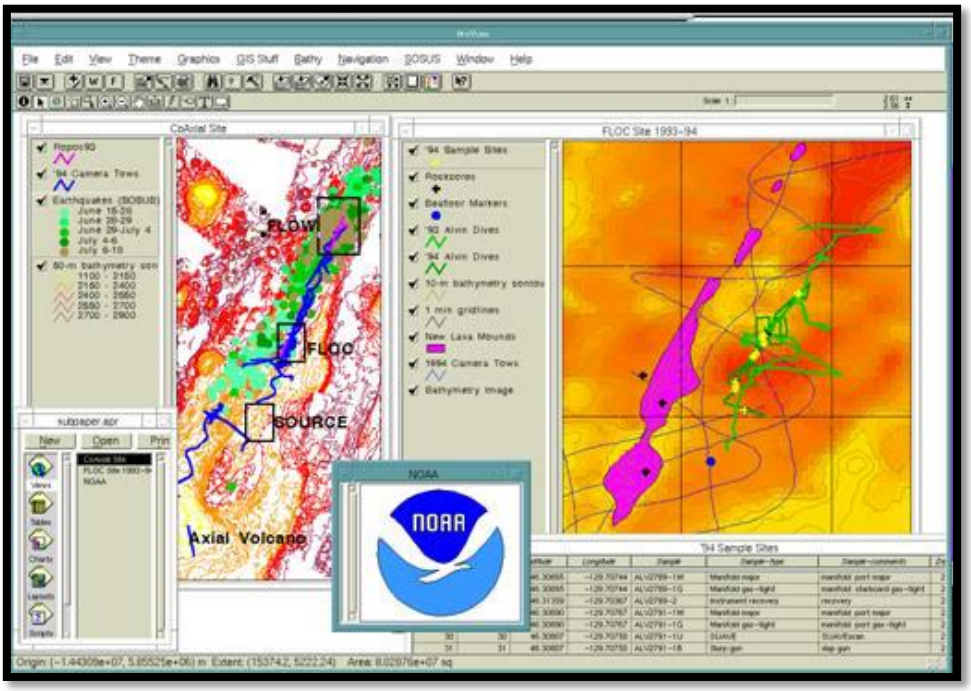
K. Underwater sensors networks

Τα υποβρύχια δίκτυα αισθητήρων [15] είναι μια νέα εφαρμογή των δικτύων αισθητήρων τα οποία χρήζουν ιδιαίτερης μελέτης. Συνήθως χρησιμοποιούνται για στρατιωτικές και επιστημονικές εφαρμογές. Αποτελούνται από μεταβλητό αριθμό αισθητήρων και οχημάτων που αναπτύσσονται για να εκτελέσουν αποστολές ανίχνευσης κίνησης σε μια δεδομένη περιοχή του βυθού .Ο master node είναι υπεύθυνος για τη συλλογή των δεδομένων,τη μεταφορά τους στο κέντρο ελέγχου στην ακτή και για τον έλεγχο των υπολοίπων κόμβων. Η κίνηση του δικτύου αποτελείται από ασύγχρονη μεταφορά δεδομένων, απομακρυσμένη εποπτεία του δικτύου και έλεγχο ωκεανογραφικών αισθητήρων. Το κύριο κίνητρο για τη δημιουργία υποβρύχιων δικτύων αισθητήρων είναι η σχετικά εύκολη ανάπτυξή τους λόγω της έλλειψης καλωδίων και της μη παρέμβασης πλοίων στην επίτευξη των στόχων. Υπάρχουν πολλές εφαρμογές των υποβρύχιων συστημάτων,όπως η ανάπτυξη κατάλληλων συστημάτων επιτήρησης, η ανίχνευση υποβρύχιων στόχων και η προώθηση κρίσιμων πληροφοριών. Οι περιβαλλοντολογικές εφαρμογές περιλαμβάνουν ανίχνευση φυσικών ενδείξεων (όπως περιεκτικότητα σε αλάτι, πίεση και θερμοκρασία) και χημικών-βιολογικών ενδείξεων (όπως επίπεδα βακτηρίων, επίπεδα μόλυνσης και επικίνδυνοι βιολογικοί ή χημικοί παράγοντες σε πηγάδια και δεξαμενές). Τα υποβρύχια δίκτυα βρίσκουν επίσης εφαρμογή και στη συλλογή ωκεανογραφικών δεδομένων, στην εκμετάλλευση υπογείων κοιτασμάτων και στην παρεμπόδιση καταστροφών από σεισμική δραστηριότητα ή την εμφάνιση τσουνάμι. Σε σύγκριση με τα ραδιοφωνικά κύματα, ο ήχος έχει καλύτερα χαρακτηριστικά διάδοσης στο νερό. Έτσι, η χρήση του ήχου είναι πιο βολική για την υποβρύχια επικοινωνία. Γι’ αυτό οι υποβρύχιοι αισθητήρες χρησιμοποιούν ακουστικά μόντεμ για την επικοινωνία τους. Μειονεκτήματα αυτής της τεχνολογίας είναι η μεγάλη χρονική καθυστέρηση, η μικρή ταχύτητα μετάδοσης και ο μεγάλος θόρυβος λόγω της υποβρύχιας επικοινωνίας.

Η πρώτη γνωστή εφαρμογή δικτύων αισθητήρων υπήρξε το SOSUS (Sound Surveillance System, Σύστημα Ηχητικής Παρακολούθησης). Το παραπάνω δίκτυο χρησιμοποιήθηκε στις αρχές της δεκαετίας του 1950, κατά τη διάρκεια του ψυχρού πολέμου, για την ανίχνευση και τον εντοπισμό Σοβιετικών υποβρυχίων(Εικόνα 10) με τη βοήθεια ειδικών ακουστικών αισθητήρων (υδρόφωνα).

Εικόνα 10

Το SOSUS παραμένει ακόμη σε λειτουργία, για ειρηνικούς σκοπούς και χρησιμοποιείται κυρίως για την παρακολούθηση διάφορων φαινομένων(Εικόνα 11) όπως σεισμικές δραστηριότητες, παρακολούθηση φαλαινών κλπ.



Εικόνα 11

1.4 Προβλήματα και Περιορισμοί των Ασύρματων Δικτύων Αισθητήρων

Τα Ασύρματα Δίκτυα Αισθητήρων παρουσιάζουν ένα εντελώς διαφορετικό σύνολο από περιορισμούς σε σύγκριση με τους περιορισμούς που παρουσιάζονται στα παραδοσιακά δίκτυα. Το πιο σημαντικό από αυτά είναι η ενέργεια. Αυτά τα δίκτυα αποτελούνται από συσκευές που πρέπει να είναι ενεργές αρκετή ώρα με μικρές μπαταρίες.
Πηγές Κατανάλωσης Ενέργειας:

- Άνοιγμα/κλείσιμο ασυρμάτου-Ενέργεια εξαρτάται από το μοντέλο του ασυρμάτου
- Μετάδοση δεδομένων-Εξαρτάται ανά bit από ισχύς μετάδοσης (πχ, τι εμβέλεια θέλουμε να έχει το μήνυμα), μοντέλο ασυρμάτου
- Λήψη δεδομένων-Αναμονή για δεδομένα (χωρίς κάτι να λαμβάνεται εκείνη την ώρα)

Η ενέργεια στις τρείς παραπάνω περιπτώσεις εξαρτάται και από το χρόνο.Επίσης, αντιμετωπίζουν κάποια προβλήματα λόγω των περιορισμών που υπάρχουν στους αισθητήρες, όπως για παράδειγμα, οι περιορισμένοι πόροι, οι χαμηλές υπολογιστικές δυνατότητες και η μικρή μνήμη.Ευτυχώς, οι δυνατότητες των αισθητήρων έχουν αυξηθεί σημαντικά τα τελευταία χρόνια και επιπλέον έχουν δημιουργηθεί αλγόριθμοι με μικρές απαιτήσεις μνήμης και μικρή πληροφορία/αισθητήρα.
Αντίθετα με τους υπόλοιπους υπολογιστές, οι αισθητήρες συχνά πρέπει να τοποθετηθούν σε δύσκολες περιοχές, όπου οι τεχνικοί που θα τους εφαρμόσουν έχουν μειωμένη ορατότητα.
Ακόμα κάτι που γίνεται δύσκολα στους κόμβους αισθητήρων είναι να καθοριστεί για πιο λόγω χάθηκε ένα πακέτο. Οι πιθανοί λόγοι είναι λόγω υπερχείλισης της ουράς, λόγω έλλειψης ενέργειας, ή λόγω έλλειψης ασφάλειας. Για παράδειγμα, κάποιος εξωγενής παράγοντας μετακίνησε τον αισθητήρα, όπως κάποιο πουλί. Ακόμα κάποιος πιθανός λόγος για το χάσιμο πακέτων είναι η συμφόρηση στο δίκτυο, επομένως κάποια πακέτα να μην μπορούν να μεταφερθούν και ειδικότερα, αυτά που βρίσκονται πιο μακριά από τον κόμβο gateway.
Τα Ασύρματα Δίκτυα Αισθητήρων πρέπει να αντιμετωπίσουν και το πρόβλημα για το coverage. Δηλαδή, πόσο καλά μια περιοχή παρακολουθείται από τους κόμβους αισθητήρες.
Επίσης πρέπει να αντιμετωπίσουν και το χαμηλό ασύρματο επικοινωνιακό bandwidth λόγω της ασύρματης επικοινωνίας τους.Λόγω αυτών και άλλων περιορισμών και προβλημάτων που έχουν να αντιμετωπίσουν τα Ασύρματα Δίκτυα Αισθητήρων υπάρχει ανάγκη για καινοτόμα συστήματα, πρωτόκολλα και αλγορίθμους.

Κεφάλαιο 2-Κινητικότητα στα Ασύρματα Δίκτυα Αισθητήρων

2.1 Γενικά

Πρόσφατα μια νέα κατηγορία για σημαντικές εφαρμογές δικτύων αισθητήρων εμφανίζεται ,όπου η κινητικότητα είναι θεμελιώδες χαρακτηριστικό για τα συστήματα που εξετάζονται. Σε τέτοιες εφαρμογές οι αισθητήρες συνάπτονται πάνω σε αυτοκίνητα, robots, ζώα ή ανθρώπους που κινούνται γύρω σε μεγάλες γεωγραφικές περιοχές.

2.2 Μορφές Κινητικότητας

Στα ασύρματα δίκτυα αισθητήρων η κινητικότητα μπορεί να εμφανιστεί σε τρεις μορφές [17]:

1. Node mobility: Οι ασύρματοι κόμβοι αισθητήρων είναι καθεαυτό κινητοί. Το δίκτυο πρέπει να αναδιοργανώνεται αρκετά συχνά για να μπορεί να λειτουργήσει σωστά.

2. Gateway mobility: Ο κόμβος gateway (ο κόμβος στον οποίο καταλήγουν οι μετρήσεις των υπόλοιπων κόμβων) είναι κινούμενος.

3. Event mobility: Σε εφαρμογές όπως η ανίχνευση γεγονότος και γενικά σε εφαρμογές καταδίωξης, η αιτία που προκαλεί το γεγονός ή το καταδιωκόμενο αντικείμενο μετακινείται. Σε τέτοιου είδους σενάρια, είναι σημαντικό να καλύπτεται συνέχεια το παρατηρούμενο γεγονός από ένα ικανοποιητικό αριθμό αισθητήρων. Όπου υπάρχει υψηλότερη δραστηριότητα, οι αισθητήρες ενεργοποιούνται γύρω από το αντικείμενο, για να το παρατηρήσουν. Ακολούθως, αφού τελειώσουν, αλλάζουν από την ενεργή κατάσταση σε κατάσταση ύπνου για εξοικονόμηση ενέργειας. Όσο μετακινείται μέσα στο δίκτυο το αντικείμενο που προκαλεί το γεγονός, συνοδεύεται από μια περιοχή με δραστηριότητα των κόμβων που βρίσκονται γύρο από αυτό.

2.3 Δυνατότητες κινητικότητας

Η κινητικότητα στα ασύρματα δίκτυα αισθητήρων επιτρέπει τη δημιουργία ενός νέου μεγάλου συνόλου από δυνατότητες. Η ικανότητα της ενεργούς αλλαγής της τοποθεσίας μπορεί να λύσει πολλές από τις σχεδιαστικές προκλήσεις των στατικών ασύρματων δικτύων αισθητήρων.Πλεονεκτήματα της κινητικότητας είναι:
Ανάπτυξη δικτύου: Η κινητικότητα μπορεί να χρησιμοποιηθεί έτσι ώστε να τοποθετηθούν οι αισθητήρες σε βέλτιστες τοποθεσίες π.χ. μεγιστοποιώντας την κάλυψη της περιοχής του δικτύου.

Τοπολογική Προσαρμοστικότητα: Γίνεται καλύτερος έλεγχος όσον αφορά την τοπολογία του δικτύου. Οι διάφορες μεταβολές στο περιβάλλον αλλάζουν το ρυθμό παραγωγής των δεδομένων των κόμβων αισθητήρων με αποτέλεσμα τη μεταβολή και της κυκλοφοριακής συμφόρησης στο δίκτυο. Σε περιπτώσεις μεγάλης κυκλοφοριακής συμφόρησης τα στατικά δίκτυα πιθανών να μην μπορούν να καλύψουν τις νέες απαιτήσεις κυκλοφορίας. Με την κινητικότητα των αισθητήρων το δίκτυο μπορεί να προσαρμοστεί στις δυναμικές καταστάσεις κατά το χρόνο εκτέλεσης.

Επιδιόρθωση δικτύου: Πολλές φορές, μπορεί να δημιουργηθούν τρύπες στην κάλυψη του δικτύου λόγω βλάβης συγκεκριμένων κόμβων, είτε από έλλειψη ενέργειας των κόμβων, είτε από καταστροφής τους. Η τρύπα αυτή μπορεί να αποκατασταθεί προγραμματίζοντας κινητούς κόμβους να μετακινηθούν σε αυτές τις περιοχές,έτσι ώστε να επιδιορθωθεί το δίκτυο.

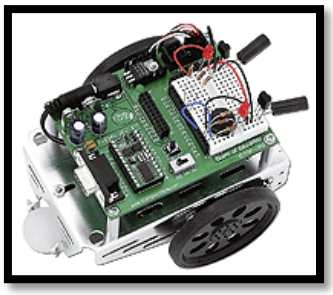
Εντοπισμός γεγονότος: Υπάρχει περίπτωση η αρχική ανάπτυξη της τοπολογίας του ασύρματου δικτύου αισθητήρων να μην είναι βέλτιστη για να ανιχνεύσει έγκαιρα τα γεγονότα. Το πρόβλημα αυτό μπορεί να διορθωθεί χρησιμοποιώντας κινητούς αισθητήρες οι οποίοι θα επισκέπτονται ανά περιόδους ή δυναμικά τις περιοχές με υψηλή πιθανότητα εμφάνισης γεγονότος.

Απομόνωση από ελαττωματικούς ή κακόβουλους κόμβους: Η απόδοση του ελαττωματικού κόμβου μπορεί να συγκριθεί με την απόδοση κάποιου γνωστού λειτουργικού κόμβου και στη συνέχεια να μεταφερθεί ο λειτουργικός κόμβος στη θέση του ελαττωματικού. Με παρόμοιο τρόπο μπορούν να εφαρμοστούν τεχνικές για την απομόνωση των κακόβουλων κόμβων. Αυτό θα έχει ως αποτέλεσμα την αύξηση της αξιοπιστίας του δικτύου και την απομόνωση κόμβων που ίσως να επηρεάσουν αρνητικά το δίκτυο.

2.4 Κινητοί Κόμβοι στα Ασύρματα Δίκτυα Αισθητήρων

Για να ερευνηθεί η κινητικότητα στα Ασύρματα Δίκτυα Αισθητήρων χρησιμοποιούνται συνήθως κάποιες πλατφόρμες – robots. Πάνω στις πλατφόρμες αυτές ενσωματώνονται οι κόμβοι αισθητήρων που πρέπει να είναι κινητοί.

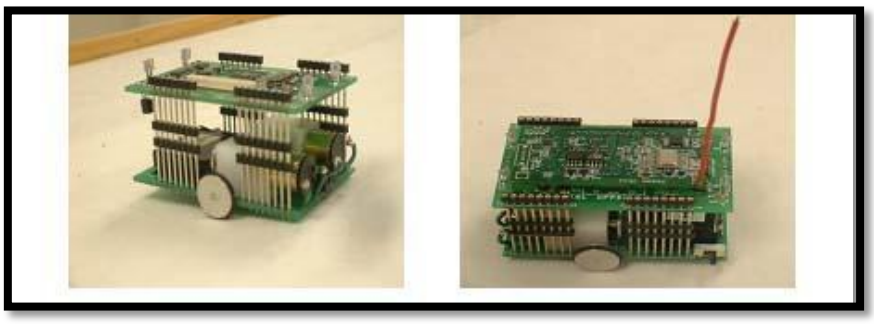
BOE-BOT(Εικόνα 12)



Εικόνα 12

Το Boe-bot [18] της parallax αποτελείται από ένα υψηλής ποιότητας αλουμινένιο σκελετό, το οποίο παρέχει μια εύρωστη πλατφόρμα για τις σέρβο μηχανές και τον πίνακα κυκλωμάτων. Οι τρύπες και οι αυλακώσεις του, μπορούν να χρησιμοποιηθούν για να προστεθεί διαφόρων ειδών ρομποτικό εξοπλισμό. Το πράσινο κύριο κύκλωμα, που είναι τοποθετημένο στο πάνω μέρος του robot ονομάζεται Board of Education. Ο μικροεπεξεργαστής που μπορεί να συνδεθεί σε μια υποδοχή στον πράσινο πίνακα κυκλωμάτων ονομάζεται BASIC Stamp και μπορεί να προγραμματιστεί στη γλώσσα προγραμματισμού PBASIC. Ακόμη, όλες οι εισόδοι – εξόδοι για την επικοινωνία με άλλα εξαρτήματα μπορούν να υλοποιηθούν πάνω στο breadboard που παρέχει. Το ρομπότ μπορεί να προγραμματιστεί για να ακολουθήσει μια γραμμή, να ακολουθήσει το φως, ή να περιπλανηθεί αυτόνομα αποφεύγοντας τα αντικείμενα τα οποία μπορεί να συναντήσει στο δρόμο του.

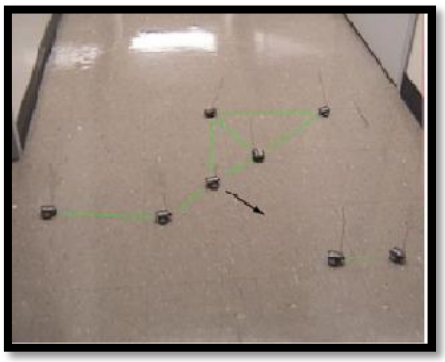
ROBOMOTE(Εικόνα 13)



Εικόνα 13

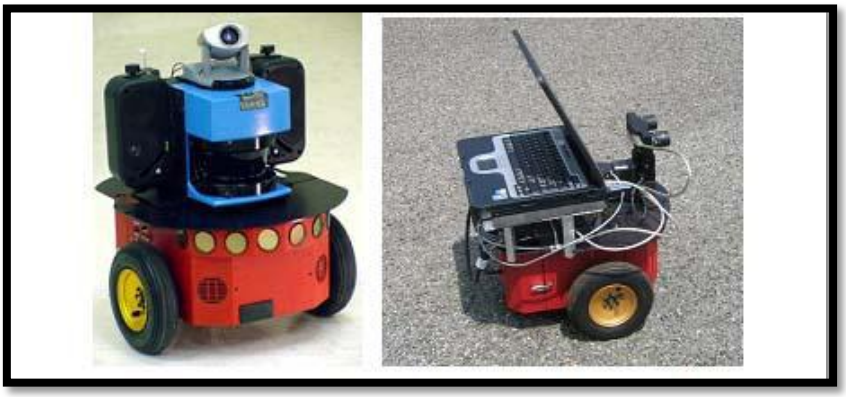
Το robomote [19] έχει σχεδιαστεί ως μια πλατφόρμα για πειραματισμό στα ασύρματα δίκτυα αισθητήρων. Οι σχεδιαστικοί στόχοι αυτής της πλατφόρμας ήταν η ευκολία ανάπτυξής της

και το χαμηλό κόστος της. Αποτελείται από ένα μικροεπεξεργαστή Atmel8535 ο οποίος είναι ένας 8-bit AVR RISC MCU με 8k bytes προγραμματιζόμενη Flash , 512 bytes EEPROM και 512 bytes εσωτερική SRAM. Επίσης το robomote έχει δυο motors, μια πυξίδα για να μπορεί να υπολογίζει την πορεία του και αισθητήρες IR. Ο κόμβος αισθητήρα επικοινωνεί με το robomote για να το κατευθύνει.Ένα από τα γνωστά πειράματα που έχουν γίνει με αυτή την πλατφόρμα είναι η καταγραφή γεγονότων από το περιβάλλον με κίνηση εμπνευσμένη από τον τρόπο που κινούνται τα βακτηρίδια (biased random walk-Εικόνα 14).



Εικόνα 14

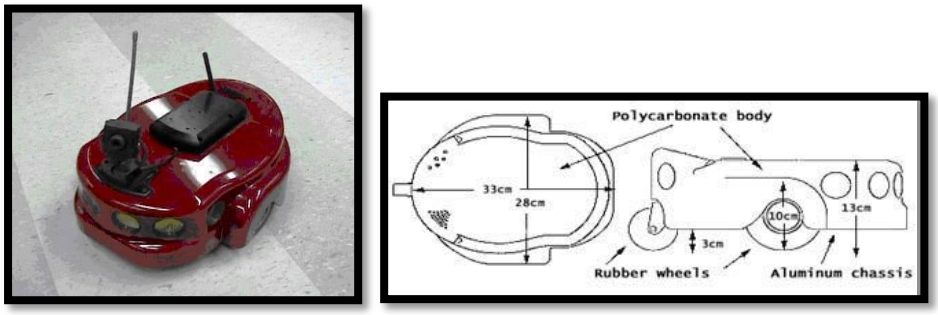
Pioneer P3-DX (Εικόνα 15)



Εικόνα 15

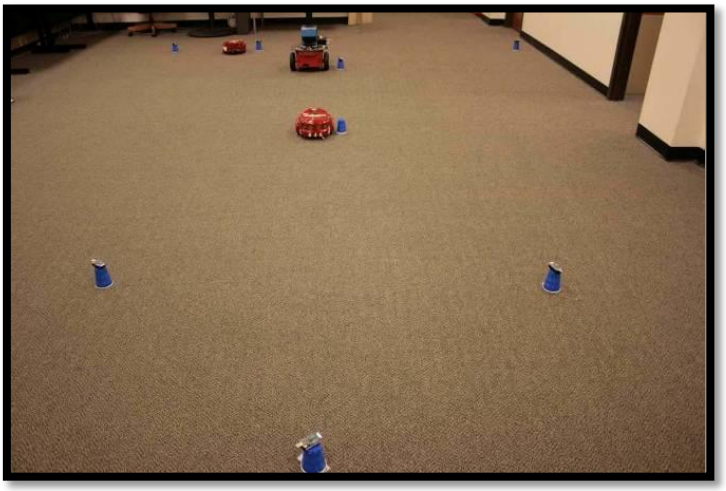
Το Pioneer 3DX (P3DX) από τη Mobile Robots είναι ένα robot,το οποίο μπορεί να διαθέτει ένα on-board pc με ένα μεγάλο εύρος από αισθητήρες και μπορεί να επικοινωνεί μέσω WiFi (Wireless Ethernet).

AmigoBot(Εικόνα 16)



Εικόνα 16

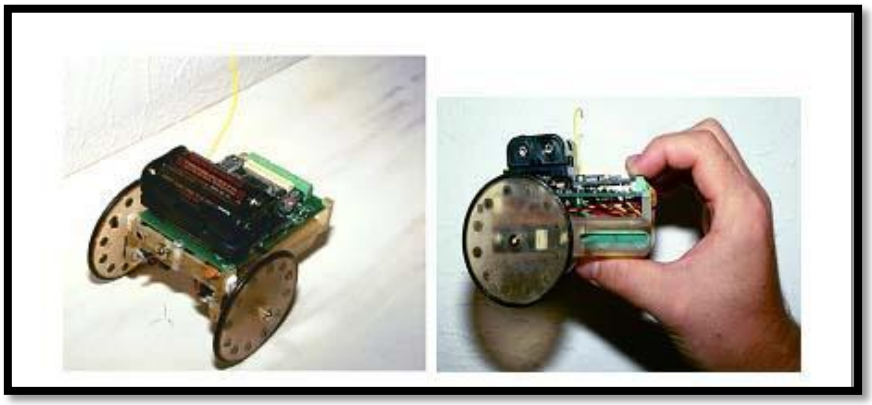
Το AmigoBot είναι ένα έξυπνο robot με μια παντοδύναμη μηχανή κίνησης. Έχει ενσωματωμένο ένα μικροεπεξεργαστή με το λειτουργικό σύστημα AmigOS και αισθητήρες οι οποίοι του επιτρέπουν να αισθάνεται τι βρίσκεται γύρω από αυτό,ώστε να κινείται με ασφάλεια στο περιβάλλον γύρω του αποφεύγοντας εμπόδια. Σε συνδυασμό με ένα state-of-the-art λογισμική ρομποτικής, όπως το ActivMedia Robotics' Interface for Applications (ARIA) και το λογισμικό Saphira να τρέχει στον υπολογιστή, το AmigoBot μπορεί να καθορίσει την τοποθεσία του και να βρεί δρομολόγιο από μια θέση σε μια άλλη. Τα δύο αυτά robot (AmigoBot και P3DX) χρησιμοποιήθηκαν σε πειράματα(Εικόνα 17) που αφορούν επιδιόρθωση συνδεσιμότητας σε σύρματα δίκτυα αισθητήρων [20] . Τα robot αυτά κατευθύνονται από ασύρματους κόμβους αισθητήρων για την επιδιόρθωση της συνδεσιμότητας του δικτύου.



Εικόνα 17

Συνεργασία AmigoBot και Pioneer P3-DX για επιδιόρθωση συνδεσιμότητας

MICAbots(Εικόνα 18)



Εικόνα 18

Τα MICAbots έχουν σχεδιαστεί για εργαστηριακά πειράματα και αποτελούνται από δύο motors, ένα μικροεπεξεργαστή, μια βάση πάνω στην οποία τοποθετούνται οι μπαταρίες και ο κόμβος αισθητήρα. Τα robot αυτά έχουν μικρό μέγεθος (8.6 cm x 6.1 cm x 2.1 cm το μέγεθος της βάσης τους) και χαμηλό κόστος (λιγότερα από $350) , γιατί κατασκευάζονται με υλικά που είναι διαθέσιμα στην αγορά. Χρησιμοποιούν τις πλατφόρμες MICA του Berkeley για την επεξεργασία των δεδομένων και την επικοινωνία και προγραμματίζονται σε περιβάλλον TinyOs.

Εικόνα 19

CotsBots(Εικόνα 20)

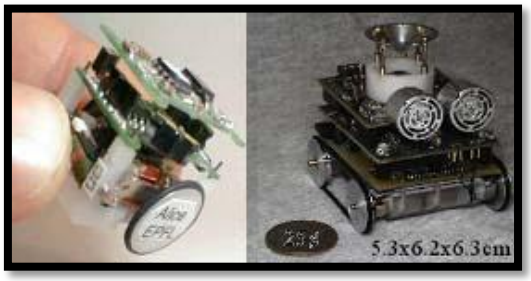
Εικόνα 20

Τα CotsBots [21] είναι mobile robots πάνω στην οποία μπορούν να ερευνηθούν αλγόριθμοι συνεργασίας και κατανεμημένης αίσθησης σε μεγάλα ασύρματα δίκτυα αισθητήρων. Έχουν μικρό μέγεθος (13cm x 6.5cm μέγεθος της βάσης) και κοστίζουν λιγότερο από $200. Τα robot αυτά είναι εξοπλισμένα με ενσωματωμένο επεξεργαστή, ασύρματη επικοινωνία και ένα σκελετό αυτοκινήτου (Kyosho mini-z) για να έχει τη δυνατότητα να κινείται. Τα

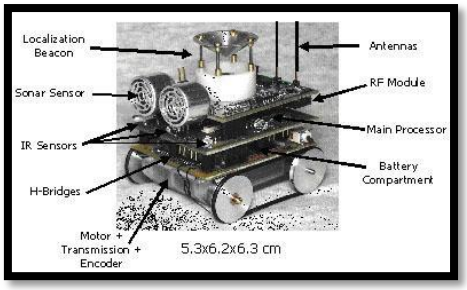
προγράμματα του είναι γραμμένα σε περιβάλλον TinyOs.Το motor board το οποίο ελέγχει τις κινήσεις του CotsBot αποτελείται από:

- Atmel ATmega8L
- 1-8MHz CPU
- 8KB Program Memory
- 1KB RAM
- 2 Discrete H-Bridge Circuits
- Speed and Direction Control of Motors
- up to 4A, 30V load
- Power Monitoring
- Accelerometer ADXL202e
- 51-pin bus

Millibots (The Millibot Team-Εικόνα 21 και 22)



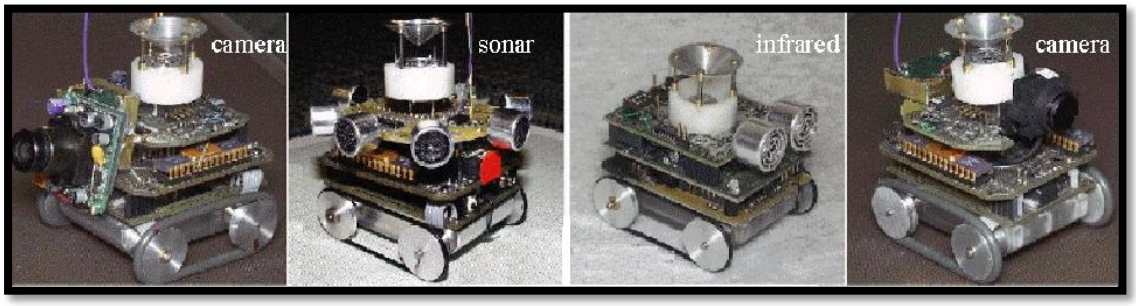
Εικόνα 21



Εικόνα 22

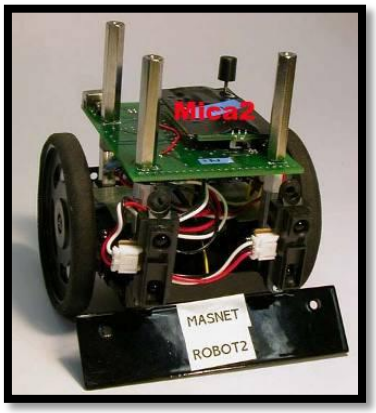
Τα millibots [22] είναι μικρά ημι-αυτόνομα ή και αυτόνομα robots. Λόγω του μικρού τους μεγέθους (7 X 7 X 7cm), είναι αδύνατο να χωρέσουν σε ένα robot όλοι οι απαραίτητοι αισθητήρες που χρειάζονται για την ολοκλήρωση μιας αποστολής. Ακόμη λόγω του μικρού μεγέθους των αισθητήρων και της μικρής ισχύς του robot, οι αισθητήρες έχουν περιορισμένες δυνατότητες. Γι' αυτό το λόγο, τα robot αυτά δουλεύουν σε ομάδες, στις οποίες το κάθε robot έχει τις δικές του ιδιαιτερότητες και όλα μαζί συνεργάζονται για να φέρουν εις πέρας μια αποστολή. Επομένως η συνεργασία είναι απαραίτητη και πρέπει να υπάρχει συντονισμός στο μάζεμα των πληροφοριών από το περιβάλλον. Η επιλογή των αισθητήρων που θα διαθέτει το κάθε robot(Εικόνα 23) και η συγκρότηση ομάδας εξαρτάται από τις ανάγκες της αποστολής. Π.χ κάποια robots μπορεί να είναι εξοπλισμένα με αισθητήρες για την εξερεύνηση της ομάδας, άλλα να είναι εξοπλισμένα με μεγαλύτερη υπολογιστική ισχύ για την επεξεργασία των δεδομένων αίσθησης και της δημιουργίας του τοπικού χάρτη της περιοχής και άλλα robots μπορεί να είναι εξοπλισμένα με καλύτερα motors και με πιο ανεπτυγμένη κινητικότητα. Για να επιτευχθεί η ειδίκευση του κάθε robot χωρίς να χρειαστεί μεγάλη

ποσότητα από robots, τα millibots έχουν κατασκευαστεί,ώστε να είναι διαμορφώσιμα. Ακόμα και η πλατφόρμα κινητικότητας είναι διαμορφώσιμη και μπορεί να επιλεχθεί ανάλογα με την επιφάνεια στην οποία θα αναπτυχθούν. Ένα σημαντικό συστατικό των millibots είναι το σύστημα καθορισμού τοποθεσίας τους που βασίζεται σε υπέρηχους. Αυτό το σύστημα έχει ένα σημαντικό πλεονέκτημα σε σχέση με τα υπάρχοντα συστήματα,γιατί δεν χρειάζεται σταθερά beacons. Χρησιμοποιώντας τα millibots είτε ως beacons, είτε ως localization receivers εναλλακτικά, η ομάδα ως σύνολο μπορεί να κινηθεί και να επανατοποθετηθεί διατηρώντας ακριβείς τους υπολογισμούς καθορισμού τοποθεσίας. Ο ακριβής εντοπισμός της τοποθεσίας των robot είναι πολύ σημαντικός για εργασίες οι οποίες περιλαμβάνουν εξερεύνηση και mapping. Μερικές εφαρμογές στις οποίες μπορούν να χρησιμοποιηθούν τα millibots περιλαμβάνουν:Εντοπισμό-Χαρτογράφηση-Διάσωση.



Εικόνα 23

MASMOTES(Εικόνα 24)

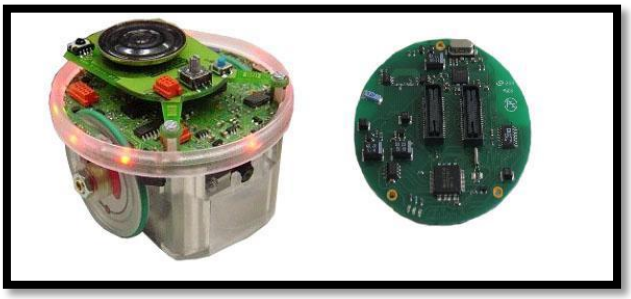


Εικόνα 24

Τα MAS-motes [23] είναι κατασκευασμένα βασισμένα στους αισθητήρες micaZ της Crossbow Technology Inc. Τα robot αυτά είναι φθηνά και δεν έχουν υψηλό κόστος συντήρησης. Είναι γερά και μικρά έτσι ώστε να αντέχουν σε αντίξοες συνθήκες. Οι δυο τροχοί που διαθέτουν έχουν ξεχωριστό motor. Επίσης,είναι εξοπλισμένα με τρεις αισθητήρες IR (δυο μπροστά και ένας πίσω) για αποφυγή των συγκρούσεων,καθώς και 2 φωτεινούς αντιστάτες οι οποίοι είναι γυρισμένοι προς τα κάτω για να ανιχνεύουν τη συγκέντρωση ομίχλης κάτω από τα MAS-mote.

Εικόνα 25

E-PUCK(Εικόνα 26)



Εικόνα 26

Το E-PUCK είναι ένα mini κινητό robot το οποίο δημιουργήθηκε από το Swiss Federal Institute of Technology για εκπαιδευτικούς σκοπούς. Τώρα είναι εμπορικά διαθέσιμο από την K-Team. Το E-PUCK έχει ένα επεξεργαστή dsPIC στα 60MHz (15MIPS) με πυρήνα DSP για επεξεργασία σημάτων, 8 proximity sensors και φωτός (ambient), ένα 3D accelerometer, 3 μικρόφωνα omni-directonal, μια VGA κάμερα, και ένα δέκτη IR για ασύρματο έλεγχο. Επίσης,έχει 8 κόκκινα LEDS, ένα πράσινο body light, ένα κόκκινο LED μεγαλύτερης έντασης στο μπροστινό μέρος και ένα μεγάφωνο το οποίο μπορεί να αναπαράγει αρχεία WAV. Παρέχει επικοινωνία μέσω RS232 και Bluetooth. Ενέργεια του παρέχουν 5 WH LiIon μπαταρίες. Προγραμματίζεται σε γλώσσα C (μεταγλωττιστής GNU GCC).Για να μπορέσει το E-PACK να επικοινωνεί ασύρματα έχει δημιουργηθεί ένα radio communication module,έτσι ώστε να μπορούν να διεξαχθούν πειράματα τα οποία απαιτούν ασύρματη επικοινωνία. Τα E-PACK έχουν χρησιμοποιηθεί σε πειράματα για Collective Decision και Isolated Collective Decision [24] .

Εικόνα 27

Κεφάλαιο 3 - Μοντελοποίηση Δεδομένων και Γλώσσες Επερωτήσεων

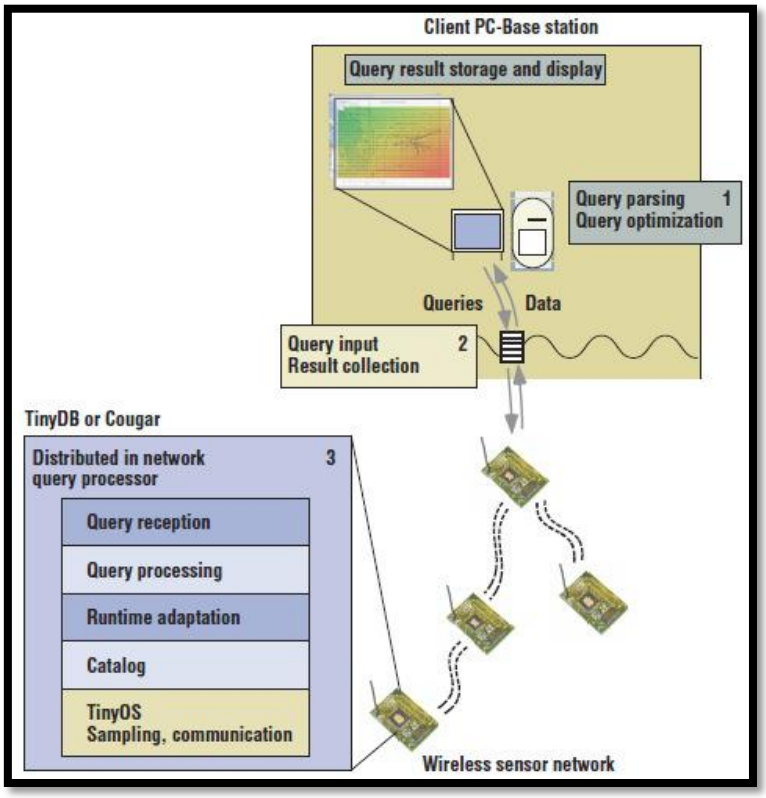
Όπως αναφέρθηκε, παρόλους τους περιορισμούς που έχουν τα δίκτυα αυτά, έχουν να υπολογίσουν και να επεξεργαστούν δεκάδες δεδομένα και να τα στείλουν σε συγκεκριμένες στιγμές. Για την επίτευξη αυτού, θα πρέπει να σχεδιαστεί και υλοποιηθεί μια αρχιτεκτονική πάνω στην οποία να μπορεί να βασιστεί μια τέτοια συλλογή δεδομένων.

3.1 Αρχιτεκτονική επεξεργαστών επερωτημάτων

Η αρχιτεκτονική επεξεργαστών επερωτημάτων αποτελείται από δύο βασικά μέρη(Εικόνα 28)

Α) Λογισμικό σε επίπεδο υπηρέτη (Server-side software) το οποίο εκτελείται στον υπολογιστή του χρήστη (σταθμός βάσης). Στην απλούστερή του μορφή το λογισμικό αυτό κάνει γραμματική ανάλυση στο επερώτημα, το διανέμει στο δίκτυο και συλλέγει τα αποτελέσματα όπως επιστρέφουν από το δίκτυο.

Β) Λογισμικό σε επίπεδο αισθητήρα (Sensor-side software) το οποίο εκτελείται στους αισθητήρες .Αυτό το λογισμικό αποτελείται από συνιστώσες λογισμικού οι οποίες είναι υλοποιημένες σε μια πλατφόρμα λειτουργικού που ονομάζεται TinyOS και είναι εγκατεστημένο σε κάθε κόμβο του δικτύου. Ένας από τους κόμβους (που ονομάζεται και gateway) επικοινωνεί με τον σταθμό – βάση.



Εικόνα 28

3.2 Γλώσσα Επερωτήσεων

Αφού έχουμε μοντελοποιήσει ως ένα βαθμό την αρχιτεκτονική ενός συστήματος επεξεργασίας επερωτημάτων δικτύων αισθητήρων, ήρθε η ώρα να μοντελοποιήσουμε τον τρόπο με τον οποίο αργότερα ο χρήστης θα δημιουργήσει τα επερωτήματα ώστε να συλλέξει μετρήσεις και αποτελέσματα.Γενικά στη σύνταξη ενός επερωτήματος για δίκτυα αισθητήρων ακολουθείται η κλασσική δηλωτική μορφή επερωτημάτων (declarative queries) σε απλή μορφή και με ορισμένες επεκτάσεις όπως φαίνεται παρακάτω

SELECT {attributes, aggregates}
FROM {Sensordata S}
WHERE {predicate}
GROUP BY {attributes}
HAVING {predicate}
DURATION time interval
EVERY time span e

Το παραπάνω πρότυπο επερωτήματος μπορεί να χρησιμοποιηθεί στο παρακάτω παράδειγμα που αφορά την μέτρηση συγκέντρωσης ενός αερίου:

SELECT AVG(R.concentration)
FROM ChemicalSensor R
WHERE R.loc IN region
HAVING AVG(R.concentration) > T
DURATION (now, now + 3600)
EVERY 10

Παρατηρούμε ότι η ειδοποιός διαφορά της παραπάνω σύνταξης με την σύνταξη της SQL είναι η υποστήριξη χρονικής διάρκειας και επανάληψης ενός επερωτήματος. Αυτό επιτυγχάνεται με τις προτάσεις DURATION και EVERY όπου ορίζουμε τον χρόνο ζωής του επερωτήματος και τον ρυθμό εκτέλεσης του επερωτήματος αντίστοιχα. Μια έλλειψη που έχει ο τρόπος αυτός εκτέλεσης επερωτημάτων είναι η ελλιπείς υποστήριξη πυροδότησης γεγονότων (trigger events). Αντί τη στιγμή που συμβεί κάτι να δημιουργήσει μια ειδοποίηση, ελέγχει ανά τακτά χρονικά διαστήματα τις μετρήσεις.Η γλώσσα αυτή λόγο του τρόπου με τον οποίο γεννά επερωτήματα στους κόμβους του δικτύου ονομάστηκε Acquisitional Query Language.Παρακάτω εξηγούμε τον τρόπο που υλοποιούμε ειδοποιήσεις με βάση γεγονότα που συμβαίνουν στο δίκτυο αισθητήρων [25] .

Επερωτήματα βασισμένα σε γεγονότα (Event-Based Queries)

Τα γεγονότα σε μια βάση δεδομένων παράγονται με δύο τρόπους, είτε από ένα άλλο επερώτημα είτε από το λειτουργικό σύστημα μέσω μιας τρίτης εφαρμογής η οποία είναι εγκατεστημένη στον κόμβο .Για παράδειγμα το παρακάτω επερώτημα:

```
ON EVENT bird-detect(loc):
SELECT AVG(light), AVG(temp), event.loc
FROM sensors AS s
WHERE dist(s.loc, event.loc) < 10m
SAMPLE INTERVAL 2 s FOR 30 s
```

χρησιμοποιείται για να αναφέρει τη μέση θερμοκρασία και φωτεινότητα τη στιγμή που ορισμένοι αισθητήρες εντόπισαν ένα πτηνό. Άρα κάθε φορά που εντοπίζεται ένα πτηνό, το επερώτημα αυτό ξεκινά από τον κόμβο στον οποίο ενεργοποιήθηκε το συμβάν και μετρήσεις από τους γειτονικούς κόμβους λαμβάνουν χώρα κάθε 2 δευτερόλεπτα και για συνολικό χρονικό διάστημα 30 δευτερολέπτων.Η δυνατότητα παραγωγής και επεξεργασίας γεγονότων παίζει καθοριστικό ρόλο στη εξοικονόμηση ενέργειας στα δίκτυα αισθητήρων καθώς επιτρέπει στο δίκτυο να είναι σε αδράνεια για αρκετό χρονικό διάστημα έως ότου συμβεί το επιθυμητό γεγονός, αντί να ελέγχει συνεχώς αν υπάρχουν δεδομένα να λάβει.

Επερωτήματα βασισμένα σε χρόνο ζωής (Lifetime-Based Queries)

Στην περίπτωση των επερωτημάτων αυτών, αντί να έχουμε την πρόταση SAMPLE INTERVAL, έχουμε την πρόταση LIFETIME <x> όπου x είναι η διάρκεια σε χρόνο (Μέρες, εβδομάδες, μήνες). Ο τρόπος αυτός είναι πολύ χρήσιμος σε περιβάλλοντα που δεν μας ενδιαφέρει τόσο οι μικροαλλαγές που συμβαίνουν στο περιβάλλον αλλά το βάθος χρόνου για το οποίο συλλέγουμε μετρήσεις. Έτσι έχουμε:

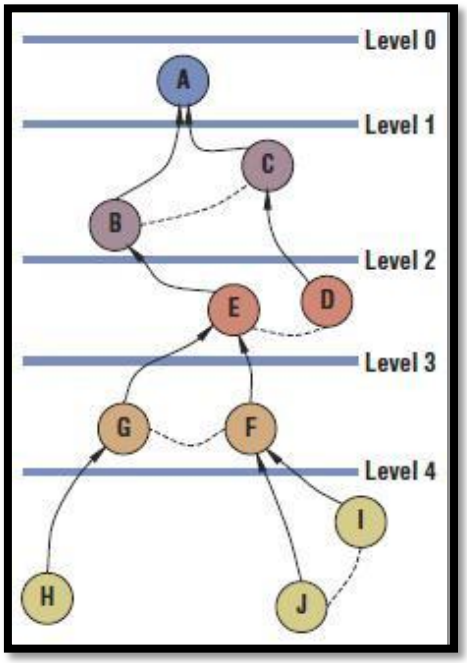
```
SELECT nodeid, accel
FROM sensors
LIFETIME 30 days
```

Σε αυτό το επερώτημα συλλέγουμε δείγματα για τουλάχιστον 30 μέρες σε ρυθμό τέτοιο ώστε να ικανοποιείται ο περιορισμός των ημερών ζωής του επερωτήματος. Για να υπολογίσει το ρυθμό αυτό η βάση δεδομένων υπολογίζει το κόστος σε ενέργεια για την μετάδοση και τη συλλογή των μετρήσεων με βάση το ενεργειακό της απόθεμα. Αυτό θα πρέπει να γίνει για κάθε κόμβο ξεχωριστά εκτός αν θεωρήσουμε ότι τα κόστη είναι ίδια. Συνήθως ο ρυθμός αυτός υπολογίζεται μέσω μιας φόρμουλας βασισμένης στο κόστος (cost-based formula).

3.3 Διασπορά Επερωτήσεων και Συλλογή Αποτελεσμάτων

Μετά τη δημιουργία και βελτιστοποίηση της επερώτησης το σύστημα θα πρέπει να την διαχύσει στο δίκτυο. Σε αυτό χρησιμεύει ένα δένδρο δρομολόγησης το οποίο ξεκινάει αρχής γενομένου από τον σταθμό βάση ή από ένα επιλεγμένο σημείο το οποίο ονομάζεται σημείο αποθήκευσης (storage point) και δημιουργείται καθώς όλοι οι κόμβοι παιδιά του δένδρου μαθαίνουν το επερώτημα και το προωθούν στα δικά τους παιδιά (1 επίπεδο κάθε φορά) έως ότου όλο το δίκτυο να μάθει το επερώτημα.Κάθε μήνυμα που μεταδίδεται στο δίκτυο περιέχει ορισμένες πληροφορίες όπως το επίπεδο του δένδρου, την απόσταση δηλαδή από τη

ρίζα και ο γονέας είναι υπεύθυνος για την προώθηση των αποτελεσμάτων του επερωτήματος στον κόμβο βάση. Αν οι κόμβοι τηρούν παραπάνω από έναν γονείς τότε το δίκτυο έχει πολλαπλά τέτοια δένδρα δρομολόγησης τα οποία μπορούν να υποστηρίξουν περισσότερα ταυτόχρονα επερωτήματα. Αυτού του είδους η τοπολογία επικοινωνίας ονομάζεται δρομολόγηση βασισμένη σε δένδρα (tree-based routing-Εικόνα 29). Να σημειωθεί στο σημείο αυτό τέλος, ότι ξεχωριστή μελέτη αποτελεί η επιλογή του κατάλληλου γονέα για κάθε κόμβο καθώς στην πράξη είναι πολύ σημαντικό για την απόδοση ενός δικτύου αισθητήρων.



Εικόνα 29

Τοπολογία ενος δικτύου σε δένδρο δρομολόγησης.Τα συμπαγή τόξα δείχνουν πρός τον γονέα ενώ οι διακεκομμένες γραμμές υποδηλώνουν ότι οι κόμβοι έχουν επαφή ο ένας με τον άλλο αλλά δεν μπορούν να δρομολογήσουν μεταξύ τους.

3.4 Συναρτήσεις συνάθροισης

Στην πλειοψηφία των περιπτώσεων, οι επερωτήσεις περιλαμβάνουν στη σύνταξή τους συναρτήσεις συνάθροισης. Στην κατηγορία αυτή ανήκουν οι συναρτήσεις, οι οποίες χρησιμοποιούν (συγκεντρώνουν) δεδομένα από πολλές εγγραφές και επιστρέφουν μία τιμή. Συναθροίζουν δηλαδή πολλές τιμές σε μία τελική τιμή. Παραδείγματα συναθροιστικών συναρτήσεων είναι οι COUNT, MIN και MAX. Στο σημείο αυτό υπεισέρχεται η έννοια της συνάθροισης μέσα στο δίκτυο Για την επεξεργασία των επερωτήσεων, υπάρχουν δύο μεθοδολογίες: η συγκεντρωτική (centralized) και η κατανεμημένη (distributed).Στην πρώτη περίπτωση, η επεξεργασία πραγματοποιείται σε στάδια δύο. Στο πρώτο στάδιο, οι κόμβοι στέλνουν τα δεδομένα μέσα από το δίκτυο προς το σταθμό βάσης. Στο δεύτερο στάδιο, ο σταθμός βάσης αναλαμβάνει την επεξεργασία των δεδομένων που έχουν ληφθεί.Η σύγχρονη τάση, όμως, απαιτεί τη χρήση της κατανεμημένης επεξεργασίας. Σκοπός της είναι η

μεταφορά μέρους του υπολογιστικού φόρτου από το σταθμό βάσης προς τους κόμβους του δικτύου, οι οποίοι αναλαμβάνουν τη μερική επεξεργασία των δεδομένων.

Χαρακτηριστικά συναρτήσεων συνάθροισης

Οι συναρτήσεις συνάθροισης διακρίνονται βάσει των παρακάτω χαρακτηριστικών [26]:

1. Ευαισθησία σε διπλοεγγραφές .Το χαρακτηριστικό αυτό καθορίζει αν μία συνάρτηση συνάθροισης θα επιστρέψει το ίδιο αποτέλεσμα όταν ένα σύνολο δεδομένων περιέχει τουλάχιστον δύο ίδιες τιμές. Παραδείγματα τέτοιων συναρτήσεων αποτελούν οι MEDIAN, AVERAGE και COUNT. Αντίθετα, οι συναρτήσεις MIN, MAX και COUNT DISTINCT δεν ανήκουν στην κατηγορία αυτή. Για παράδειγμα, αν η συνάρτηση AVERAGE λάβει την τιμή 5 δύο φορές, το αποτέλεσμά της θα συγκλίνει ακόμα περισσότερο προς την τιμή 5. Αντίθετα, η συνάρτηση MAX δεν επηρεάζεται, αφού θα επιστρέψει πάλι την τιμή 5.

2. Ανεκτικότητα στις απώλειες. Οι ενδεικτικές (exemplary) συναρτήσεις επιστρέφουν πάντα μία αντιπροσωπευτική τιμή που περιέχεται στο σύνολο δεδομένων,ενώ οι περιληπτικές(summary) εκτελούν διάφορους υπολογισμούς σε ολόκληρο το σύνολο δεδομένων και επιστρέφουν την τελική τιμή. Οι περιληπτικές τιμές (όπως τα αποτελέσματα και AVERAGE των COUNT) είναι πιο εύκολο να υπολογιστούν, ακόμα και σε ένα δίκτυο με απώλεια πληροφορίας, όταν δεν λαμβάνονται όλα τα πακέτα. Από την άλλη πλευρά, οι ενδεικτικές συναθροίσεις μπορεί να επιστρέψουν ανακριβείς τιμές, ακόμα και στην περίπτωση που χαθούν λίγα μόνο πακέτα, γι'αυτό είναι απαραίτητο κάθε τιμή να λαμβάνει μέρος στη συνάθροιση. Παραδείγματα ενδεικτικών συναθροίσεων αποτελούν οι συναρτήσεις MIN, MAX και MEDIAN.

3. Μονοτονία. Στην κατηγορία αυτή ανήκουν συναθροίσεις που επιτρέπουν όσο το δυνατόν πιο νωρίς τον έλεγχο της τιμής ενός πεδίου. Για παράδειγμα, έστω ότι ο χρήστης ζητάει να μάθει τη μέγιστη τιμή της θερμοκρασίας στο δίκτυο. Οι κόμβοι φύλλα μεταδίδουν την πληροφορία τους, οι γειτονικοί κόμβοι ακούνε τους κόμβους φύλλα, αλλά μεταδίδουν τις δικές τους τιμές μόνο στην περίπτωση που είναι μεγαλύτερες από την τρέχουσα μέγιστη τιμή. Με αυτό τον τρόπο, μειώνεται σημαντικά ο αριθμός μηνυμάτων που αποστέλλεται στο σταθμό βάσης, ενώ δεν επηρεάζεται το αποτέλεσμα. Άλλο παράδειγμα είναι ο υπολογισμός της τιμής MIN από μία σειρά εγγραφών ταξινομημένη σε αύξουσα σειρά. Τότε, λαμβάνεται η τιμή μόνο από την πρώτη εγγραφή με αποτέλεσμα να μειώνεται σημαντική η χρήση υπολογιστικού φόρτου.

4. Πληροφορία εγγραφής μερικής κατάστασης. Το ποσό της πληροφορίας μερικής κατάστασης που απαιτείται, διαφέρει ανάμεσα στις συναρτήσεις συνάθροισης. Συναρτήσεις όπως οι SUM και COUNT, χρειάζονται εγγραφές μερικής κατάστασης, οι οποίες έχουν ίδιο μέγεθος με την τελική συνάθροιση. Η συνάρτηση AVERAGE χρειάζεται μία εγγραφή μερικής κατάστασης, που θα περιέχει δύο τιμές, της SUM και της COUNT μαζί. Άλλες

συναρτήσεις όπως η MEDIAN, χρειάζονται να μεταδοθεί στο σταθμό βάσης ολόκληρο το σύνολο δεδομένων

Συνάθροιση μέσα στο δίκτυο

Μία επερώτηση που ζητάει συγκεντρωτικά δεδομένα, εισάγεται στο δίκτυο αισθητήρων στον κεντρικό κόμβο του δικτύου, το κόμβο gateway. Στη συνέχεια ο κόμβος gateway προωθεί την επερώτηση στους άλλους κόμβους του δικτύου. Το απλούστερο αλλά λιγότερο βέλτιστο σχέδιο είναι εκείνο που στηρίζεται στη μέθοδο συγκεντρωτικής επεξεργασίας. Σύμφωνα με το σχέδιο αυτό, κάθε κόμβος, αφού υπολογίσει τα δεδομένα βάσει της επερώτησης που έλαβε, τα αναφέρει στον κόμβο gateway ώστε να πραγματοποιηθεί η επεξεργασία τους.Όταν ο gateway λάβει όλα τα πακέτα πληροφοριών απ 'όλους τους κόμβους,συναθροίζει όλα τα δεδομένα σε μία τελική τιμή και την αναφέρει στο χρήστη.Η μέθοδος αυτή, η οποία είναι γνωστή ως άμεση μεταφορά (direct delivery), έχει αρκετά μειονεκτήματα. Ένα από αυτά είναι το γεγονός ότι πρέπει να μεταφερθεί μεγάλος αριθμός πακέτων στον gateway. Αφού κάθε κόμβος στέλνει τις δικές του πληροφορίες, υπάρχει τουλάχιστον ένα πακέτο δεδομένων για κάθε κόμβο.
Επίσης, κάποιοι κόμβοι δεν έχουν τη δυνατότητα να επικοινωνήσουν άμεσα με τον gateway, με αποτέλεσμα τα πακέτα τους να πρέπει να μεταδοθούν από άλλους (γειτονικούς) κόμβους μέχρι να φτάσουν στον τελικό προορισμό. Μειονέκτημα αποτελεί επίσης το μέγεθος του κάθε πακέτου, το οποίο είναι αρκετά μικρό αφού περιέχει τα δεδομένα ενός μόνο κόμβου. Με αυτό τον τρόπο, αυξάνεται ο αριθμός μικρών πακέτων που μεταδίδονται προς το σταθμό βάση και μειώνεται η διάρκεια ζωής του δικτύου. Προκειμένου να διατηρηθεί η ενέργεια και το εύρος ζώνης σε χαμηλά επίπεδα, κρίνεται χρήσιμο η επεξεργασία των δεδομένων να υλοποιηθεί μέσα στο ίδιο το δίκτυο. Κάνουμε δηλαδή χρήση της κατανεμημένης επεξεργασίας επερωτήσεων. Η συνάθροιση μέσα στο δίκτυο αποτελεί μηχανισμό που έχει στόχο τη μείωση της κατανάλωσης ενέργειας και χρήσης εύρους ζώνης που απαιτούνται για την εκτέλεση της επερώτησης του χρήστη, δίνοντας στους ενδιάμεσους κόμβους τη δυνατότητα να συναθροίζουν τα δεδομένων των αισθητήρων. Η τεχνική αυτή παρέχει διάφορα πλεονεκτήματα σε σχέση με τη μέθοδο direct delivery:

1. Μειώνεται ο αριθμός των πακέτων που πρέπει να σταλούν μέσα από το δίκτυο. Όσο τα πακέτα δεδομένων αναπαράγονται από τους κόμβους, μπορεί να συνδυαστούν μαζί με άλλα πακέτα, περιέχοντας τις αντίστοιχες τιμές συνάθροισης. Άλλωστε, ο χρήστης δεν ενδιαφέρεται για μεμονωμένες τιμές, οπότε δεν υπάρχει απώλεια στην τιμή του αποτελέσματος που λαμβάνει.

2. Μειώνεται κατά πολύ η πιθανότητα σύγκρουσης των πακέτων. Αφού γίνεται αποστολή λιγότερων πακέτων, είναι λιγότερο πιθανό για έναν αισθητήρα να χρειαστεί να στείλει πάλι κάποιο μήνυμα το οποίο μπορεί να χάθηκε σε σύγκρουση πακέτων.

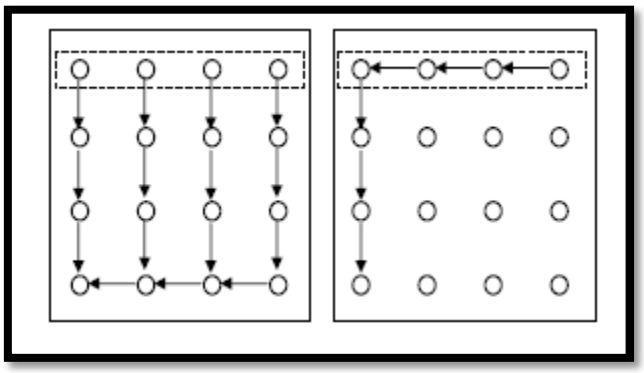
3. Μειώνεται το μέγεθος της περίσσειας πληροφορίας (πλεονασμός) που λαμβάνει ο gateway. Στην περίπτωση που μεταδίδονται μεμονωμένα δεδομένα από κάθε κόμβο, αυξάνεται η πιθανότητα οι ενδιάμεσοι κόμβοι να λάβουν το ίδιο μήνυμα από γειτονικούς κόμβους και ο σταθμός βάσης να λάβει τελικά πολλαπλές αντιγραφές του ίδιο μηνύματος και να χρειάζεται να διαγράψει την περίσσεια πληροφορία.
4. Αυξάνεται η ακρίβεια των αποτελεσμάτων. Οι αισθητήρες, οι οποίοι αποσυνδέονται προσωρινά από το δίκτυο, δεν έχουν τη δυνατότητα να λάβουν και να επεξεργαστούν μία επερώτηση. Στην περίπτωση αυτή, ο κόμβος γονέας παρατηρεί ότι το παιδί του δεν επέστρεψε κάποια τιμή και μπορεί να την υπολογίσει βασιζόμενος σε προηγούμενα δεδομένα. Αυτό είναι πάρα πολύ χρήσιμο σε περιβάλλοντα στα οποία οι τιμές για κάποια μεγέθη δύσκολα αλλάζουν σημαντικά σε μικρά χρονικά διαστήματα (π.χ. η θερμοκρασία σε ένα δωμάτιο).

Κεφάλαιο 4- Οι αλγόριθμοι ECReduced, STG και STS

Ένας τυπικός τρόπος της εξαγωγής πληροφοριών από ένα δίκτυο αισθητήρων είναι να διαδώσει τις συναθροιστικές επερωτήσεις από τον κόμβο gateway στους κόμβους αισθητήρες, ζητώντας τους να ελέγξουν περιοδικά το περιβάλλον, και να επιστρέψουν τα αποτελέσματα τους σε τακτά χρονικά διαστήματα. Ένα παράδειγμα τέτοιων συναθροιστικών επερωτήσεων είναι “select avg(temperature) from Sensors where loc in Region every 10 min”. Δεδομένου ότι οι κόμβοι λειτουργούν με μπαταρίες, η ενεργειακή συντήρηση είναι σημαντική γιατί προσκρούει άμεσα στη διάρκεια ζωής του δικτύου. Οι πρόσφατες μελέτες έχουν δείξει ότι η ραδιοεπικοινωνία είναι σημαντικά ακριβότερη από τον υπολογισμό ή την μέτρηση στις περισσότερες υπάρχουσες πλατφόρμες κόμβων αισθητήρων. Ως εκ τούτου, κύρια επιδίωξη στο σχεδιασμό των αλγορίθμων επεξεργασίας επερώτησης , είναι να ελαχιστοποιηθεί το κόστος επικοινωνίας της αποστολής των αποτελεσμάτων από τους κόμβους-αισθητήρες στον κόμβο gateway.Το κόστος διάδοσης επερώτησης στο δίκτυο υποτίθεται ότι έχει έναν δευτεροβάθμιο ρόλο για μακροπρόθεσμες επερωτήσεις, δεδομένου ότι η διάδοση εμφανίζεται μια φορά, ενώ η διάδοση αποτελέσματος εμφανίζεται επανειλημμένα σε τακτά χρονικά διαστήματα. Το κόστος επικοινωνίας της διάδοσης αποτελέσματος εξουσιάζει έτσι το κόστος επικοινωνίας της διάδοσης επερώτησης. Αφότου κατασκευάζεται ένα δέντρο, οι κόμβοι αισθητήρων προωθούν προς τα εμπρός τις μετρήσεις τους κατά μήκος των πορειών του δέντρου.Η πιο πρόσφατη έρευνα έχει εστιάσει στη βελτιστοποίηση πολλών συναθροιστικών επερωτήσεων (optimizing multiple aggregate queries).Ο στόχος είναι να βρεθούν οι αποδοτικές διαδρομές που ελαχιστοποιούν το κόστος επικοινωνίας στην εκτέλεση πολλών συναθροιστικών επερωτήσεων, με τη μελέτη της αλληλεπίδρασης μεταξύ της επεξεργασίας και της δρομολόγησης των τιμών της επερώτησης.

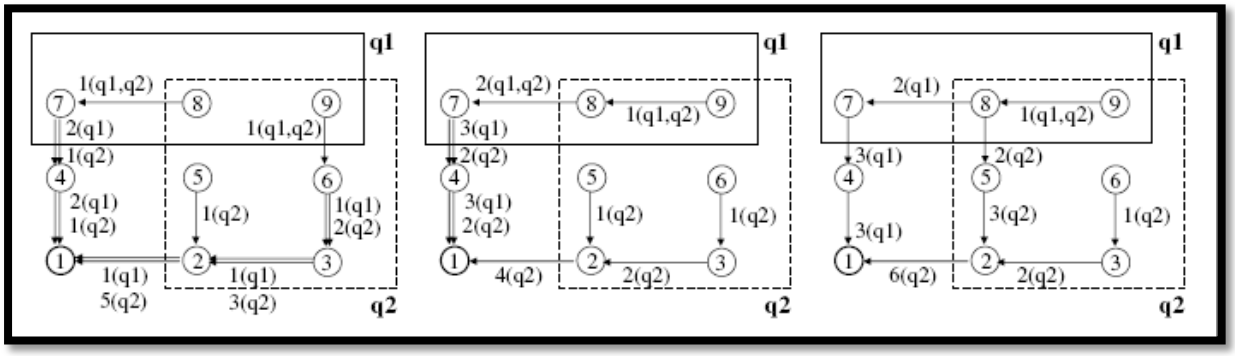
Παραδείγματα

Τα πλεονεκτήματα από την σωστή επιλογή ενός σχεδίου δρομολόγησης και επεξεργασίας για την εκτέλεση των συναθροιστικών επερωτήσεων παρουσιάζονται στα ακόλουθα παραδείγματα.



Εικόνα 30

Η εικόνα 30 παρουσιάζει ένα παράδειγμα μιας ενιαίας συναθροιστικής επερώτησης, η οποία ζητά το άθροισμα(sum) όλων των αναγνώσεων στη διακεκομμένη ορθογώνια περιοχή. Προσέξτε ότι ένας συνολικός αριθμός 15 μηνυμάτων στέλνονται στο αριστερό δέντρο, ενώ μόνο 6 μηνύματα διαβιβάζονται κατά μήκος του προσεκτικά επιλεγμένου δεξιού δέντρου. Το δεξιό δέντρο δρομολόγησης είναι καλύτερο από άποψη συνολικού κόστους επικοινωνίας . Επίσης, το όφελος του δεύτερου σχεδίου είναι ότι αθροίζει όλες τις αναγνώσεις μιας επερώτησης νωρίς και αποφεύγει να στέλνει μετρήσεις μέσα από δυσχερή μονοπάτια.



Εικόνα 31

Η εικόνα 31 δείχνει τα οφέλη για την δημιουργία ενός κατάλληλου σχεδίου εκτέλεσης στην περίπτωση πολλών επερωτήσεων αρίθμησης(count) . Για την ευκολία στις γραφικές παραστάσεις περιλαμβάνονται επίσης IDs κόμβων και τα μηνύματα που διαβιβάζονται μέσω των συνδέσεων δικτύων. Τα μηνύματα έχουν το σχήμα u(q1, . . . , qn), που δείχνει ότι την τιμή u πού συμβάλλει στις ερωτήσεις q1. . . , qn. Τα τεμνόμενα ορθογώνια διαιρούν φυσικά το δίκτυο σε μικρότερα τμήματα. Ένα τμήμα S(segment) είναι ένα σύνολο κόμβων, που καλύπτονται από το ίδιο σύνολο επερωτήσεων .Το αριστερό σχέδιο δεν εκμεταλλεύεται τις κοινές τιμές των τμημάτων και επομένως αποτυγχάνει να αθροίσει μαζί τις αναγνώσεις (των κόμβων 8 και 9). Το δεύτερο σχέδιο παρουσιάζει μικρότερο κόστος επικοινωνίας, επειδή εκμεταλλεύεται τις κοινές τιμές των τμημάτων, ωστόσο δεν αθροίζει τις τιμές που καλύπτουν ίδια επερώτηση και τις διαβιβάζει χωριστά μέχρι τον κόμβο gateway. Αυτή η συμπεριφορά είναι παρόμοια με το δεύτερο αλγόριθμο που θα παρουσιάσουμε στην συνέχεια αποκαλούμενο SegmentToGateway(STG). Το δεξιό σχέδιο έχει μια βέλτιστη συμπεριφορά επειδή εκμεταλλεύεται τις κοινές τιμές τμημάτων και αποφεύγει να στέλνει τιμές που καλύπτουν ίδια επερώτηση μέσα από μακροχρόνια μονοπάτια. Ο κόμβος 8 στέλνει τις τιμές σε δύο γονείς(5 και 7) και συγχωνεύονται με άλλες δύο της ίδιας επερώτησης .Με αυτόν τον τρόπο λοιπόν, μόνο δύο ξεχωριστά αποτελέσματα στέλνονται στον κόμβο gateway .Αυτός ο τρόπος δρομολόγησης τιμών περιγράφεται από τον αλγόριθμο SegmentToSegment (STS).
Και οι τρείς αλγόριθμοι αποτελούνται από δύο φάσεις: (ι) μια φάση διαμόρφωσης δικτύων και (ii) μια διάδοση αποτελέσματος Ο ρόλος της πρώτης φάσης είναι να δημιουργηθούν οι διαδρομές για να προετοιμάσει το έδαφος για τη δεύτερη φάση, δηλ. την αποστολή των αποτελεσμάτων στον κόμβο gateway σε περιοδικά διαστήματα.

1.Ο αλγόριθμος ECReduced [27]

Φάση διαμόρφωσης δικτύου :Οι συναθροιστικές επερωτήσεις διαδίδονται στο δίκτυο και κάθε κόμβος επιλέγει ως γονέα του(parent node) το γείτονα στην κοντυνότερη πορεία προς στον κόμβο gateway. Εάν υπάρχουν περισσότεροι από ένας υποψήφιοι γονείς , ο κόμβος επιλέγει το γονέα του με έναν από τους ακόλουθους τρόπους: (ι) τυχαία, (ii) ο πρώτος κόμβος από τον οποίο έλαβε μια επερώτηση, (iii) ο κόμβος με τον οποίο διατηρεί με συνέπεια την καλύτερη επικοινωνία.

Φάση διάδοσης αποτελέσματος: Οι διαδρομές των αποτελεσμάτων επερώτησης προκαθορίζονται στη φάση διαμόρφωσης δικτύου.Κάθε κόμβος στέλνει τις μετρήσεις του στον κόμβο parent.Αν ένα ζευγάρι εισαγωγής (value,CV) αναφέρεται σε q επερωτήσεις ,διασπάται σε q διαφορετικά ζευγάρια όπου το καθένα αντιστοιχεί σε μια επερώτηση.Ζευγάρια εισαγωγής που αντιστοιχούν στην ίδια επερώτηση συγχωνεύονται ,μετατρέπονται σε ζευγάρια εξαγωγής και στέλνονται στον κόμβο parent.

2.Ο αλγόριθμος SegmentToGateway (STG) [27]

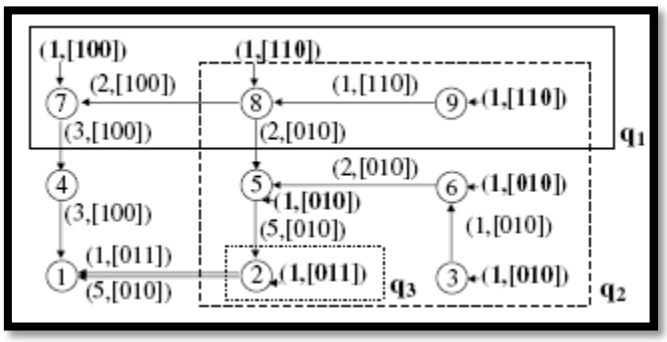
Ο αλγόριθμος STG εκμεταλλεύεται το γεγονός ότι τα τεμνόμενα ορθογώνια επερώτησης διαιρούν φυσικά το δίκτυο σε μικρότερα τμήματα.Ορισμοί σημαντικών εννοιών για την κατανόηση της STG δίνονται παρακάτω:

- Ένα τμήμα S(segment) είναι ένα σύνολο κόμβων που καλύπτονται από το ίδιο σύνολο επερωτήσεων. Παραδείγματος χάριν, οι επερωτήσεις στο στην εικόνα 32 διαμορφώνουν πέντε τμήματα {s1, s4}, {s2}, {s3, s5, s6},{s7} και {s8, s9}.Ένα τμήμα S (ή ένας κόμβος n στο S) αντιπροσωπεύεται από ένα bit-vector που ονομάζεται SGVector και δείχνει ποιες επερωτήσεις καλύπτουν τους κόμβους του S (π.χ., SGVector({s3,s5,s6})=[010]).
- Ο STG εκτελεί τη συνάθροιση των μετρήσεων των αισθητήρων με την δημιουργία ενός δέντρου ανά τμήμα, αντί της δημιουργίας ενός δέντρου ανά επερώτηση, ή ενός δέντρου για όλες τις επερωτήσεις.Τα δέντρα των τμημάτων έχουν ως ρίζα τον κόμβο SGLeader , δηλ. τον κόμβο με το μικρότερο hopCount(κοντινότερη πορεία προς στον κόμβο gateway). SGDistance είναι ο αριθμός των hops απο τον SGLeader στον κόμβο gateway. distToSGLeader είναι ο αριθμός των hops απο έναν κόμβο στον SGLeader του. Για παράδειγμα,SGDistance (s6) =2 και distToSGLeader (s6) =1.

Φάση διαμόρφωσης δικτύου: Αυτή η φάση είναι παρόμοια με την αντίστοιχη του αλγορίθμου ECReduced, εκτός από το ότι σε αυτήν την περίπτωση κάθε κόμβος προσδιορίζει όχι μόνο έναν γείτονα γονέα(parent-Εικόνα 33.Γραμμές κώδικα απο 1 εώς 9) αλλά και ένα SGParent (δηλ., ένας γείτονας σε μια πορεία στο SGLeader- Εικόνα 33.Γραμμές κώδικα απο 11 εώς 48). Επάνω στη λήψη ενός μηνύματος αναγνωριστικών σημάτων ,ένας κόμβος ενημερώνει τον τοπικό κατάλογο γειτόνων και, εάν είναι απαραίτητο, της αξίας hopCount (όπως σε ECReduced). Το επόμενο βήμα εξαρτάται από εάν το αναγνωριστικό σήμα στέλνεται από έναν κόμβο στο ίδιο ή σε ένα διαφορετικό τμήμα. Στην πρώτη περίπτωση(Εικόνα 33.Γραμμές κώδικα απο 11 εώς 29), ο κόμβος συγκρίνει τα στοιχεία που έχει για τον SGLeader με αυτά στο αναγνωριστικό σήμα. Εάν το αναγνωριστικό σήμα ξέρει για ένα SGLeader πιό κοντά στην gateway (με μικρότερο SGDistance), η μεταβλητή SGDistance ενημερώνεται και ο κόμβος που έστειλε το αναγνωριστικό μήνυμα επιλέγεται για να είναι ο

SGParent. Στην δεύτερη περίπτωση (Εικόνα 33.Γραμμές κώδικα απο 31 εώς 48)όπου ένας κόμβος λαμβάνει ένα αναγνωριστικό σήμα απο ένα κόμβο σε ένα διαφορετικό τμήμα, συνειδητοποιεί ότι είναι στα σύνορα του τμήματος και έτσι είναι υποψήφιος κόμβος για να γίνει SGLeader.Εκλέγεται για να είναι SGLeader εάν το hopCount του είναι μικρότερο από το SGDistance.Το μήνυμα αναγνωριστικών σημάτων ενημερώνεται αναλόγως και ξαναστέλνεται

Φάση διάδοσης αποτελέσματος: Στο τέλος της φάσης διαμόρφωσης του δικτύου, κάθε κόμβος ξέρει τον γονέα του και τον SGParent του. Στη φάση διάδοσης αποτελέσματος, κάθε κόμβος συγχωνεύει τα ζευγάρια εισαγωγής (value, CV) με ίδιο CV αθροίζοντας τις τιμές. Ένα ζευγάρι εξαγωγής(value, CV) στέλνεται στον SGParent εάν και μόνο εάν το CV είναι ίσο με το SGVector του τρέχοντος κόμβου. Τα υπόλοιπα ζευγάρια εξαγωγής (value, CV) διαβιβάζονται στον κόμβο parent. Οι γείτονες κόμβοι του κόμβου gateway στέλνουν όλα τα μηνύματά τους χωρίς εξαίρεση άμεσα στην gateway.Έτσι λοιπόν ο αλγόριθμος STG αθροίζει τις τιμές όλων των κόμβων μέσα σε κάθε τμήμα χωριστά ακολουθώντας ένα μίνι-δέντρο που καταλήγει στον SGLeader. Οι υπόλοιπες τιμές (των οποίων τα CVs δεν είναι ίσα με το SGVector) διαβιβάζονται στους κόμβους parent (αντί του SGParent) και φθάνουν στον κόμβο gateway μέσω της κοντινότερης πορείας.



Εικόνα 32

3.Ο αλγόριθμος SegmentToSegment (STS) [27]

Αν και ο STG λειτουργεί καλά από άποψη να συγχωνεύσει τις αναγνώσεις του ίδιου τμήματος, αποτυγχάνει συχνά στο να συγχωνεύσει μετρήσεις που ανήκουν στην ίδια επερώτηση που προέρχονται από τα διαφορετικά τμήματα. Στη χειρότερη περίπτωση, αυτές οι μετρήσεις διαδίδονται από τους κόμβους SGLeader στον κόμβο gateway απο μακριά ανεπιθύμητα μονοπάτια. Ο STS εξετάζει την αδυναμία του STG με την αποστολή των μηνυμάτων προς τους γείτονες που είναι πιθανό να τα συγχωνεύσουν με άλλα.

Φάση διαμόρφωσης δικτύου: Η φάση διαμόρφωσης του STS είναι παρόμοια με την αντίστοιχη φάση του STG εκτός από το ότι κάθε κόμβος επιλέγει ως SGParent έναν κόμβο στην κοντινότερη (αντί σε οποιεσδήποτε) πορεία προς στον SGLeader (Στην Εικόνα 33 βλέπουμε την διαφορά ανάμεσα στον αλγόριθμο STS και στον STG).Ο αριθμός των hops από τον κόμβο προς τον SGLeader του, αποθηκεύεται στην μεταβλητή distToSGLeader.Όταν ένας κόμβος SGLeader μεταδίδει ένα μήνυμα θέτει την μεταβλητή distToSGLeader ίση με μηδέν. Μόλις ληφθεί ένα αναγνωριστικό μήνυμα ο κόμβος συγκρίνει την τιμή της τοπικής μεταβλητής distToSGLeader με την αντίστοιχη του μηνύματος αυξημένη κατά ένα. Εάν η τιμή της τοπικής μεταβλητής είναι μεγαλύτερη,τότε ο κόμβος αποστολέας επιλέγεται ως SGParent και η τοπική μεταβλητή distToSGLeader αλλάζει .

Φάση διάδοσης αποτελέσματος: Αρχικά, κάθε κόμβος μετατρέπει τα ζευγάρια εισαγωγής (value, CV) σε ζευγάρια εξαγωγής ακριβώς όπως στον ECReduced και στον STG.Έπειτα έχουμε δύο νέα βήματα: 1) Οι κόμβοι επιλέγουν έναν κατάλληλο γείτονα για να διαβιβάσουν κάθε ζευγάρι εξαγωγής, και 2) Διαχωρισμός μηνυμάτων , ο οποίος χωρίζει το ζευγάρι εξαγωγής πρίν το διαβιβάζει . Αναλυτικά το πρώτο βήμα : Ο βασικός στόχος του πρώτου βήματος είναι να διαβιβαστούν τα ζευγάρια εξαγωγής (value, CV) προς τους κόμβους που πιθανός να τα συγχωνεύσουν με άλλα. Το πρώτο (value, CV) ζευγάρι που εξετάζεται είναι αυτό που συμβάλλει στις περισσότερες ερωτήσεις (με το μέγιστο αριθμό 1 bit στο CV).
Η διαδικασία ταιριάσματος με τον καλύτερο κόμβο γειτόνων είναι:
Βήμα 1.1: Για να εξασφαλίσουν οι κόμβοι ότι τα μηνύματα δεν διαβιβάζονται μακρυά από τον κόμβο gateway, μόνο οι γείτονες πιό κοντά στον κόμβο gateway από τον τρέχοντα κόμβο θεωρούνται κατάλληλοι ,δηλ. Lexicographically (hopCount, SGDistance, distToSGLeader, xCoord, yCoord). Έτσι λοιπόν , ο κόμβος s3 επιλέγει τον s2(Εικόνα 32) για να στείλει μήνυμα . Σαν εξαίρεση, ένας κόμβος εξετάζει επίσης τους γείτονες στο ίδιο τμήμα που δεν είναι πιό στενοί στην gateway, εάν (i) αυτοί είναι πιό κοντά στον SGLeader τους και (ii) όλες οι επερωτήσεις που καλύπτουν αυτούς τους κόμβους συμπεριλαμβάνονται επίσης στο CV των μηνυμάτων.Για παράδειγμα, ο s3 εξετάζει και τον s6 για να διαβιβάσει την μέτρηση του (1, [010]) επειδή distToSGLeader (s6) <distToSGLeader (s3) και το SGVector(s6) = [010] εναι κοινό με το CV [010].
Βήμα 1.2: Μεταξύ των γειτόνων που επιλέγονται στο βήμα 1.1, εξετάστε μόνο εκείνους που ταιριάζουν καλύτερα με το CV δηλ. όποιοι καλύπτονται από το μέγιστο αριθμό κοινών επερωτήσεων με CV. Εάν αυτός ο αριθμός είναι 0 ή ο κόμβος είναι δίπλα στην gateway, στέλνει το μήνυμα στον κόμβο parent. Ο κόμβος s3 έχει δύο υποψήφιους γείτονες για να στείλει (1,[010]),τον κομβο s2 και τον κόμβο s6, (από το βήμα 1.1). Τα SGVectors [011] και [010] των s2 και s6 έχουν μια κοινή επερώτηση με το CV ([010]). Μεταξύ των γειτόνων με ίσο αριθμό κοινών επερωτήσεων, οι κόμβοι επιλέγουν αυτόν με τον ελάχιστο αριθμό ερωτήσεων (s6).
Βήμα 1.3: Μεταξύ των γειτόνων που επιλέγονται στο βήμα 1.2, οι κόμβοι επιλέγουν αυτόν με μικρότερο lexicographically (SGDistance, distToSGLeader, xCoord, yCoord). Για παράδειγμα,ο s8 έχει δύο υποψήφιους γείτονες τον s5 και τον s7 για να στείλει το ζευγάρι εξαγωγης (2, [110]). Και οι δύο έχουν SGDistance ίσο με 2 και distToSGLeader ίσο με 0 (και οι δύο κόμβοι είναι ηγέτες τμήματος), έτσι το s7 επιλέγεται επειδή έχει μικρότερη συντεταγμένη X.
Βήμα 2: Διαχωρισμός μηνυμάτων. Η λογική πίσω από αυτό το βήμα είναι ότι συχνά η διάσπαση των ζευγαριών αποδίδει καλύτερα. Για παράδειγμα,έστω p να είναι το ζευγάρι που εξετάζεται να σταλεί στο προηγούμενο βήμα. Το ζευγάρι p μπορεί να χωρίστει σε δύο ζευγάρια p1 και p2, βασισμένα στο SGVector του επιλεγμένου γείτονα. Υποθέστε ότι ο κόμβος s8 επιλέγει τον s7 για να διαβιβάσει το p = (2, [110]) στο προηγούμενο βήμα. Προσέξτε ότι το CV του p καλύπτει περισσότερες επερωτήσεις (q1 και q2) από το SGVector του επιλεγμένου γείτονα (SGVetorc (s7) =[100],που δείχνει ότι ο s7 καλύπτεται μόνο από την q1 επερώτηση). Σε αυτήν την περίπτωση, το p μπορεί να χωριστεί σε δύο ζευγάρια, ένα που συμβάλλει στις κοινές επερωτήσεις p1 = (2, [100]), και άλλο που συμβάλλει στην επερώτηση που απομένει p2 = (2, [010]). Το ζευγάρι p1 στέλνεται στον επιλεγμένο γείτονα και το p2 επανεισάγεται στον κατάλογο ζευγαριών εξαγωγής. Κατά τη διάρκεια της εισαγωγής, τα ζευγάρια με ίσο CVs συγχωνεύονται.Εάν ο κατάλογος CV με τα ζευγάρια για αποστολή δεν είναι κενός, τα βήματα 1 και 2 επαναλαμβάνονται.
Με τη βοήθεια της προσεκτικής δρομολόγησης μηνυμάτων, της συγχώνευσης και του διαχωρισμού, ο STS εξασφαλίζει ότι όλες οι μετρήσεις κοινών επερωτήσεων συγχωνεύονται μαζί προτού να αφήσουν την περιοχή επερώτησης, προσφέροντας κατά συνέπεια μεγαλύτερα οφέλη σε σχέση με τα STG και ECReduced.

Κώδικας java για την φάση διαμόρφωσης δικτύου του αλγορίθμου STG και του αλγορίθμου STS.Οι παρακάτω γραμμές κώδικα χωρίς αυτές που έιναι υπογραμμισμένες αντιστοιχούν στον STG και όλος ο κώδικας στον STS.

```
1   if(( (source.getHopcount()+1)<thisNode.getHopcount()
2        || ((source.getHopcount()+1)==thisNode.getHopcount()
3        && source.getMyLocation().
4        strictlyCloser(allNodes.get(thisNode.getParentId()).getMyLocation()))){
5
6              thisNode.setHopcount(source.getHopcount()+1);
7              thisNode.setParentId(source.getID());
8
9              }
10
11  if(equalVectors(thisNode.getSGVector(),source.getSGVector())){
12
13
14      if((thisNode.getSGDistance()>source.getSGDistance())
15      || (thisNode.getSGDistance()==source.getSGDistance()
16      && (source.getDistanceToSGLeader()+1)<thisNode.getDistanceToSGLeader())
17      || (thisNode.getSGDistance()==source.getSGDistance()
18      && (source.getDistanceToSGLeader()+1)==thisNode.getDistanceToSGLeader()
19      && source. getleaderLoc().strictlyCloser(thisNode.getleaderLoc()))){
20
21         thisNode.setSGParentId(source.getID());
22         thisNode.setSGDistance(source.getSGDistance());
23         thisNode.setDistanceToSGLeader((source.getDistanceToSGLeader()+1));
24         thisNode.setleaderLoc(source.getleaderLoc());
25
26                         }
27
28
29                 }
30
31  else if(equalVectors(thisNode.getSGVector(),source.getSGVector()==false){
32
33       if( (thisNode.getSGDistance()>thisNode.getHopcount())
34            || ((thisNode.getSGDistance()==thisNode.getHopcount())
35            && source.getID()==thisNode.getParentId()
36            &&  source.getID()!=thisNode.getSGParentId()
37            &&thisNode.getMyLocation().strictlyCloser(thisNode.getleaderLoc()))){
38
39                 thisNode.setSGParentId(source.getID());
40                 thisNode.setSGDistance(thisNode.getHopcount());
41                 thisNode.setDistanceToSGLeader(0);
42                 thisNode.setleaderLoc(thisNode.getMyLocation());
43
44                         }
45
46
47          }
48
```

Εικόνα 33

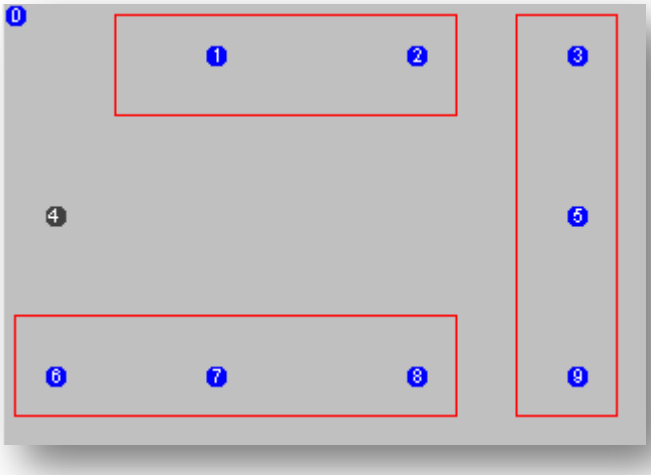
Κεφάλαιο 5-Ο αλγόριθμος SegmentToSegment With Mobile Nodes(STS-Mobile)

Η κινητικότητα των κόμβων αισθητήρων σε ασύρματο δίκτυο αισθητήρων (WSN) έχει δημιουργήσει νέες προκλήσεις ιδιαίτερα σε σχέση κατανάλωση ενέργειας καθώς παρατηρούνται πολλά link failures.Ορισμένες πραγματικές εφαρμογές επιβάλλουν σε συνδυασμό περιβάλλοντα των σταθερών και κινητών κόμβων αισθητήρων στο ίδιο δίκτυο, ενώ άλλες απαιτούν ένα πλήρες κινητό περιβάλλον απο αισθητήρες. Ως εκ τούτου, κύρια επιδίωξη στο σχεδιασμό των αλγορίθμων επεξεργασίας πολλαπλών επερωτήσεων με κινούμενους κόμβους , είναι να ελαχιστοποιηθεί το κόστος επικοινωνίας λόγω των πολλών link failures.Όπως προαναφέραμε,το κόστος διάδοσης επερώτησης στο δίκτυο υποτίθεται ότι έχει έναν δευτεροβάθμιο ρόλο για μακροπρόθεσμες επερωτήσεις, δεδομένου ότι η διάδοση εμφανίζεται μιά φορά, ενώ η διάδοση αποτελέσματος εμφανίζεται επανειλημμένα σε τακτά χρονικά διαστήματα. Ο στόχος λοιπόν είναι τόσο να βρεθούν οι αποδοτικές διαδρομές που ελαχιστοποιούν το κόστος επικοινωνίας στην εκτέλεση πολλών συναθροιστικών επερωτήσεων, με τη μελέτη της αλληλεπίδρασης μεταξύ της επεξεργασίας και της δρομολόγησης των τιμών της επερώτησης όσο και να μειωθεί ο αριθμός των link failures.

Στην συνέχεια παρουσιάζουμε ένα παράδειγμα για το πώς λειτουργεί ο αλγόριθμος STS σε ένα δίκτυο αισθητήρων με κινούμενους κόμβους και τα λάθη που παρουσιάζονται.

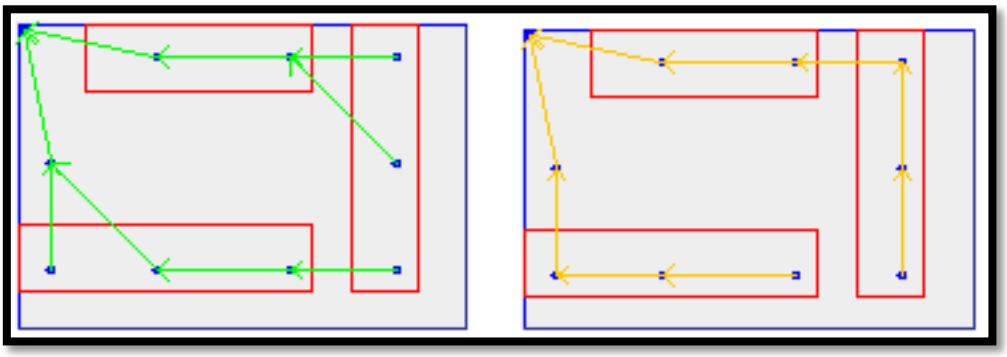
Παράδειγμα

Έχουμε ένα δίκτυο αισθητήρων που αποτελείται από 10 κόμβους(Εικόνα 34).Ο κόμβος 0 είναι ο κόμβος gateway που διαδίδει τις επερωτήσεις στους κόμβους και στον οποίο καταλήγουν οι μετρήσεις.Παρατηρούμε ότι το δίκτυο καλύπτουν τρείς συναθροιστικές επερωτήσεις οι οποίες δημιουργούν τέσσερα segments {s1, s2}, {s3, s5, s9} , {s6,s7,s8} και {s4}.Οι επερωτήσεις ζητούν από τους κόμβους να λαμβάνουν και να στέλνουν μετρήσεις ανα Χ δευτερόλεπτα.Οι μπλέ κόμβοι είναι οι ακίνητοι κόμβοι και ο μαύρος κόμβος(4) έναι κινούμενος.



Εικόνα 34

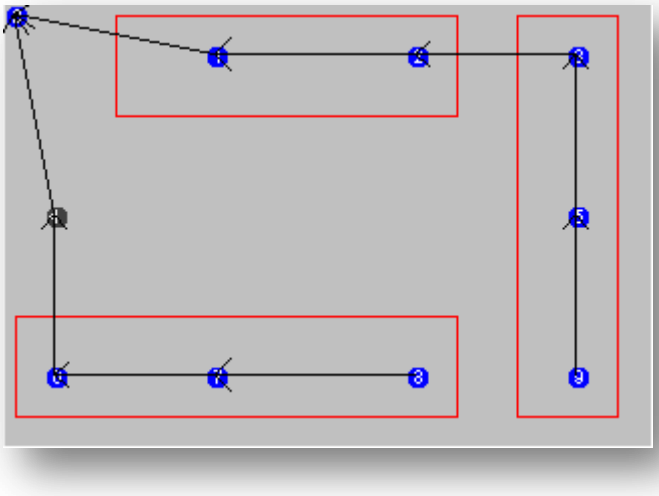
Διαμόρφωση δικτύου σύμφωνα με τον αλγόριθμο STS



Εικόνα 35

Τα βέλη στο αριστερό μέρος της εικόνα 35 δείχνουν τις επιλογές των κόμβων ως προς τον parent κόμβο και τα βέλη στο δεξιό μέρος δείχνουν τις επιλογές των κόμβων ως προς τον SGParent κόμβο.

Φάση διάδοσης αποτελέσματος:



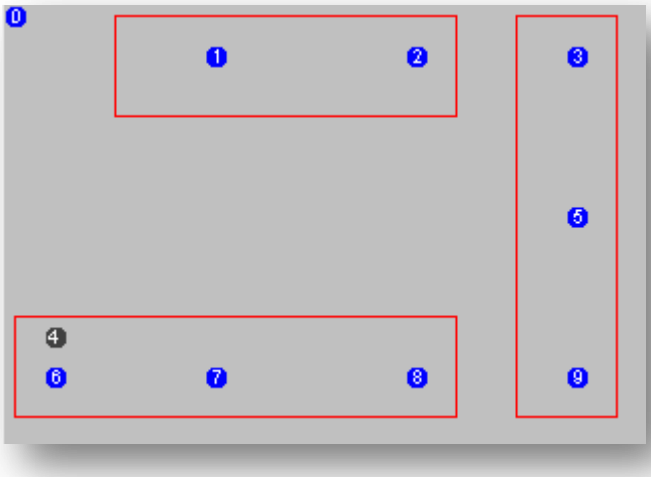
Εικόνα 36

----------Propagation Phase----------

The node 9 send the message:(Value: 1--CV:[0][0][1]) to Node: 5
The node 5 send the message:(Value: 2--CV:[0][0][1]) to Node: 3
The node 8 send the message:(Value: 1--CV:[1][0][0]) to Node: 7
The node 3 send the message:(Value: 3--CV:[0][0][1]) to parent Node: 2
The node 7 send the message:(Value: 2--CV:[1][0][0]) to Node: 6
The node 2 send the message:(Value: 1--CV:[0][1][0]) to Node: 1
The node 2 send the message:(Value: 3--CV:[0][0][1]) to parent Node: 1

The node 6 send the message:(Value: 3--CV:[1][0][0]) to parent Node: 4
The node 1 send the message:(Value: 2--CV:[0][1][0]) to parent Node: 0
The node 1 send the message:(Value: 3--CV:[0][0][1]) to parent Node: 0
The node 4 send the message:(Value: 3--CV:[1][0][0]) to parent Node: 0

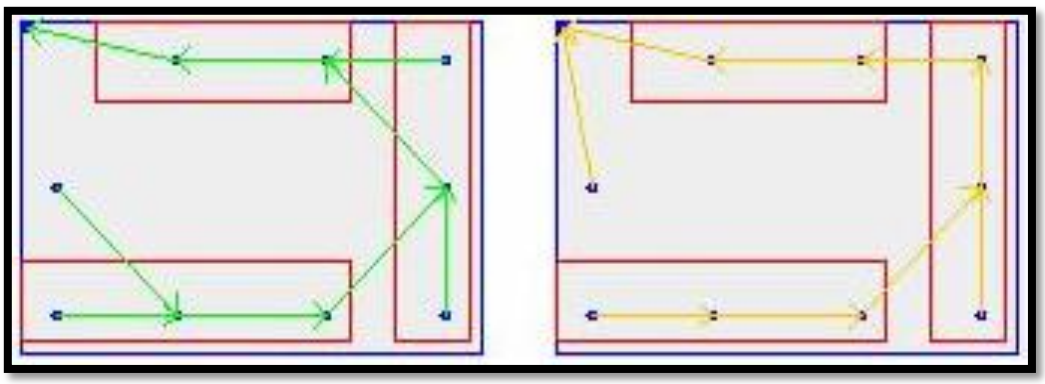
Παρατηρούμε ότι οι κόμβοι 8,7 και 6 στέλνουν τις μετρήσεις τους στον κόμβο gateway 0 διαμέσου του κόμβου 4.Μετά από Χ δευτερόλεπτα το δίκτυο μοιάζει ως εξής:



Εικόνα 37

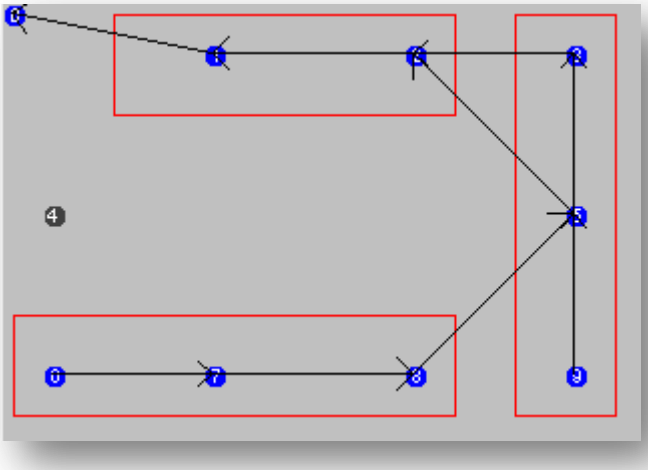
Ο κόμβος 4 έχει αλλάξει τοποθεσία και έχει εισέλθει στο τρίτο τμήμα(Εικόνα 37).Το πρόβλημα δεν είναι ότι ο κόμβος 4 έχει χάσει την σύνδεση του(link failure) με τον κόμβο 0 που ήταν και ο parent και ο SGParent του ,αλλα πλέον το υποδέντρο του τμήματος {s6,s7,s8} έχει χάσει την σύνδεση του με τον κόμβο gateway .Θα πρέπει λοιπόν να γίνει αναδιαμόρφωση του δικτύου για να μπορέσουν οι μετρήσεις του νέου τμήματος {s4,s6,s7,s8} να φτάσουν στον κόμβο gateway 0.Το μέγεθος των αναγνωριστικών μηνυμάτων που στέλνονται για την αναδιαμόρφωση είναι μεγαλύτερο απο το μέγεθος των μηνυμάτων που στέλνονται στην φάση διάδοσης αποτελέσματος και επιπλέον θα πρέπει να αλλάξει όλο το υποδέντρο του τμήματος, έτσι έχουμε μεγάλο επικοινωνιακό κόστος.Θα ήταν αποδοτικότερο λοιπόν η διαμόρφωση του δικτύου να γινόταν ως εξής(Εικόνα 38):

Διαμόρφωση δικτύου



Εικόνα 38

Φάση διάδοσης αποτελέσματος(Εικόνα 39):



Εικόνα 39

----------Propagation Phase----------

The node 6 send the message:(Value: 1--CV:[1][0][0][0]) to Node: 7
The node 7 send the message:(Value: 2--CV:[1][0][0][0]) to Node: 8
The node 8 send the message:(Value: 3--CV:[1][0][0][0]) to parent Node: 5
The node 9 send the message:(Value: 1--CV:[0][0][1][0]) to Node: 5
The node 5 send the message:(Value: 2--CV:[0][0][1][0]) to Node: 3
The node 5 send the message:(Value: 3--CV:[1][0][0][0]) to parent Node: 2
The node 3 send the message:(Value: 3--CV:[0][0][1][0]) to parent Node: 2
The node 2 send the message:(Value: 1--CV:[0][1][0][0]) to Node: 1
The node 2 send the message:(Value: 3--CV:[1][0][0][0]) to parent Node: 1
The node 2 send the message:(Value: 3--CV:[0][0][1][0]) to parent Node: 1
The node 1 send the message:(Value: 2--CV:[0][1][0][0]) to parent Node: 0
The node 1 send the message:(Value: 3--CV:[1][0][0][0]) to parent Node: 0
The node 1 send the message:(Value: 3--CV:[0][0][1][0]) to parent Node: 0

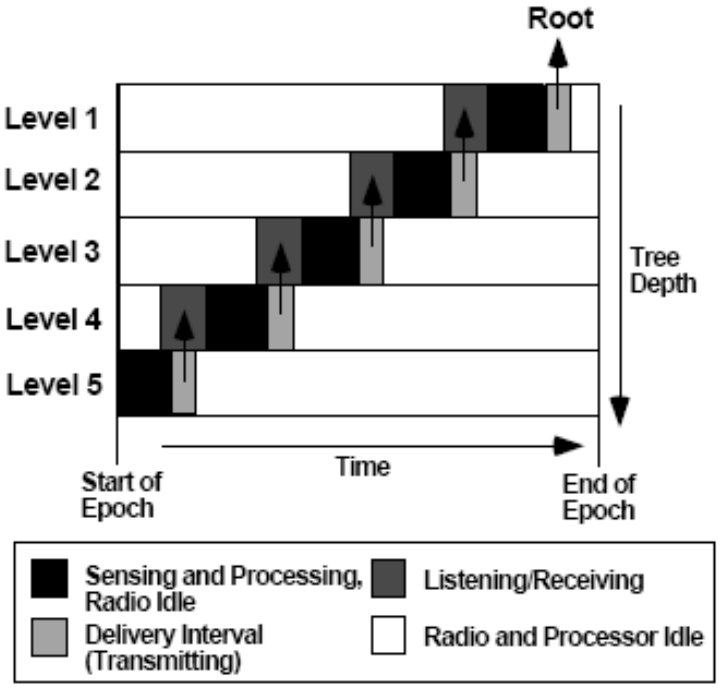
Παρατηρούμε λοιπόν ότι με αυτόν τον τρόπο διαμόρφωσης του δικτύου το μόνο πρόβλημα που θα προκύψει με την μετακίνηση του κόμβου 4 ,είναι ότι θα πρέπει ο κόμβος 4 να διαλέξει έναν κόμβο SGParent για να μπορέσει να στείλει την μέτρηση του.Επίσης,πρέπει να παρατηρήσουμε ότι στην φάση διάδοσης αποτελέσματος, οι μετρήσεις του τμήματος {s6,s7,s8} ακολουθούν μεγαλύτερο μονοπάτι(2 hops) για να καταλήξουν στον κόμβο gateway , σε σχέση με τον αλγόριθμο STS,αλλά όπως προαναφέραμε το μέγεθος των αναγνωριστικών μηνυμάτων που στέλνονται για την αναδιαμόρφωση είναι πολύ μεγαλύτερο

από το μέγεθος μηνυμάτων που στέλνονται στην φάση διάδοσης αποτελέσματος.Καταλήγουμε λοιπόν στο συμπέρασμα ότι ο δεύτερος αλγόριθμος διαμόρφωσης του δικτύου είναι αποδοτικότερος από τον STS.Ο αλγόριθμος αυτός ονομάζεται STS-Mobile.

Φάση διαμόρφωσης δικτύου STS-Mobile

Όπως προαναφέραμε στην φάση διαμόρφωσης του STS κάθε κόμβος επιλέγει ως parent έναν κόμβο στην κοντινότερη πορεία προς τον gateway και ως SGParent έναν κόμβο στην κοντινότερη πορεία προς τον SGLeader.Σε αυτό το σημείο βρίσκεται και η διαφορά ανάμεσα στον αλγόριθμο STS και στον STS-Mobile.Στον αλγόριθμο STS-Mobile κάθε κόμβος επιλέγει ως parent έναν κόμβο μέσω του οποίου θα συνδέεται με τον gateway με μικρότερο κόστος και ως SGParent έναν κόμβο μέσω του οποίου θα συνδέεται με τον SGLeader με μικρότερο κόστος.Οι κόμβοι με την βοήθεια των αναγνωριστικών μηνυμάτων γνωρίζουν την ταχύτητα και την θέση των γειτονικών κόμβων και έτσι μπορούν να υπολογίσουν το πόσο χρόνο μπορούν να επικοινωνούν μεταξύ τους.Βάση του χρόνου αυτού μπορούμε να ορίσουμε μία νέα παράμετρος που ονομάζουμε κόστος(cost).Αρχικά ας ορίσουμε κάποιες μεταβλητές και ύστερα ας δούμε πως υπολογίζουμε το κόστος. Θα πρέπει να τονίσουμε ότι ο τρόπος υπολογισμού του κόστους έγινε σύμφωνα με τις πειραματικές μετρήσεις.Χρησιμοποιήσαμε και άλλους τρόπους υπολογισμού του κόστους, οι οποίοι όμως απέφεραν χειρότερα αποτελέσματα.Για παράδειγμα, υπολογίσαμε το κόστος σύμφωνα με τον τύπο Cost=1/(epochsConnected*hops_from_Gateway).Η προσθήκη της δεύτερης παραμέτρου (hops_from_Gateway) , προσθέτει το γεγονός ότι οι κόμβοι που βρίσκονται πιο κοντά στον κόμβο gateway, μπορούν να αναδιοργανώσουν μεγαλύτερο κομμάτι του δικτύου(άρα και περισσότερα μηνύματα) , όμως τα πειραματικά αποτελέσματα δεν ήταν ικανοποιητικά.

numberOfEpochs:ο αριθμός των εποχών.Όπως φάινεται και απο την παρακάτω εικόνα, σε κάθε εποχή οι αισθητήρες δρομολογούν δεδομένα ή συναθροιστικές τιμές του υποδέντρου τους προς το χρήστη μέσω δέντρου συνάθροισης



Εικόνα 40

timeConnected:ο χρόνος που οι κόμβοι επικοινωνούν
timeEpochRepeat:ο χρόνος που επαναλμβάνεται κάθε εποχή
epochsConnected:Είναι το ακέραιο αποτέλεσμα της διαίρεσης της μεταβλητής timeConnected με την μεταβλητή timeEpochRepeat .Είναι δηλαδή ο αριθμός των εποχών που δύο γειτονικοί κόμβοι είναι συνδεδεμένοι.

Ο υπολογισμός του κόστους γίνεται ως εξής:

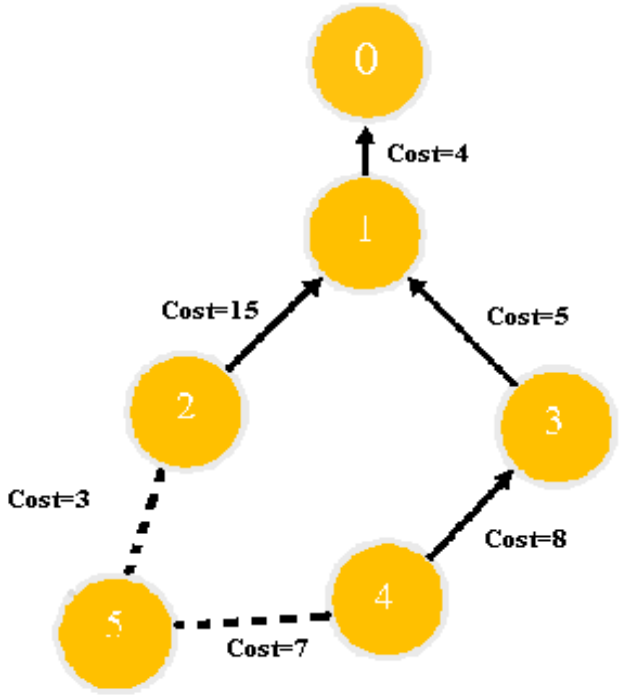
- Εάν και οι δύο κόμβοι είναι ακίνητοι τότε
  - **Cost=0**
- Εάν ένας ή και οι δύο είναι κινούμενοι τότε
  - Υπολογίζουμε τον χρόνο που οι δύο κόμβοι επικοινωνούν **timeConnected**
  - Υπολογίζουμε τον αριθμό των εποχών που οι δύο κόμβοι επικοινωνούν κάνοντας την διαίρεση
    epochsConnected =(INTEGER)(timeConnected / timeEpochRepeat)
  - **Cost =1/epochsConnected**

Όπως προαναφέραμε κάθε κόμβος επιλέγει ως parent(SGParent) έναν κόμβο μέσω του οποίου θα συνδέεται με τον gateway(SGLeader) για μεγαλύτερο χρονικό διάστημα.Έχοντας ως κριτήριο το κόστος ορίζουμε τις έννοιες:

Κόστος υποδέντρου:είναι το maximum κόστος ανάμεσα στους κόμβους ενός υποδέντρου.
CostToGateway: είναι το κόστος ενός κόμβου Α που ισούται με το κόστος υποδέντρου, που έχει ως φύλλο τον κόμβο Α και ρίζα τον gateway

SGCostToSGLeader: είναι το κόστος ενός κόμβου Α που ισούται με το κόστος ενός υποδέντρου, που έχει ως φύλλο τον κόμβο Α και ρίζα τον SGLeader του κόμβου Α.
SGCost: ισούται με την τιμή της μεταβλητής CostToGateway του SGLeader του.

Επάνω στη λήψη ενός μηνύματος αναγνωριστικών σημάτων ,αρχικά ο κόμβος ενημερώνει τον τοπικό κατάλογο γειτόνων.Στην συνέχεια υπολογίζει, ακολουθώντας τα βήματα που δείξαμε παραπάνω ,το κόστος ανάμεσα στον ίδιο και στον κόμβο που στέλνει το αναγνωριστικό μήνυμα και παίρνει την maximum τιμή (Εικόνα 42-Γραμμή κώδικα 1)μεταξύ του κόστους των δύο κόμβων και του κόστους υποδέντρου που έχει ως ρίζα τον gateway και ως φύλλο τον κόμβο που στέλνει το μήνυμα.Την maximum τιμή την συγκρίνει με την τοπική τιμή της μεταβλητής CostToGateway. Αν η maximum τιμή είναι μικρότερη τότε επιλέγει τον κόμβο που έστειλε το μήνυμα ως parent και η τιμή της τοπικής μεταβλητής αλλάζει και γινεται ίση με την maximum τιμή(Εικόνα 42-Γραμμή κώδικα απο 3 εώς 14).
Το επόμενο βήμα,η επιλογή δηλαδή SGParent(Εικόνα 42-Γραμμή κώδικα απο 16 εώς 58) , εξαρτάται από εάν το αναγνωριστικό σήμα στέλνεται από έναν κόμβο που βρίσκεται στο ίδιο ή σε ένα διαφορετικό τμήμα. Στην πρώτη περίπτωση, ο κόμβος συγκρίνει τα στοιχεία που έχει για τον SGLeader με αυτά στο αναγνωριστικό μήνυμα(Εικόνα 42-Γραμμή κώδικα απο 16 εώς 38).Εάν το αναγνωριστικό σήμα ξέρει για ένα SGLeader με μικρότερο SGCost, η τοπική μεταβλητή SGCost ενημερώνεται και ο κόμβος που έστειλε το αναγνωριστικό μήνυμα επιλέγεται για να είναι ο SGParent. Για τον υπολογισμό του κόστους SGCostToSGLeader δεν μας ενδιαφέρει ο χρόνος που δύο κόμβοι επικοινωνούν μέχρι να χαθεί η σύνδεση τους , αλλά ο χρόνος που επικοινωνούν βρισκόμενοι και οι δύο στο ίδιο segment.Στην συνέχεια ,όπως είδαμε και στην επιλογή του parent, παίρνει την maximum τιμή μεταξύ του κόστους των δύο κόμβων και του κόστους υποδέντρου που έχει ως ρίζα τον SGLeader και ως φύλλο τον κόμβο που στέλνει το μήνυμα.Την maximum τιμή την συγκρίνει με την τοπική τιμή της μεταβλητής SGCostToSGLeader. Αν η maximum τιμή είναι μικρότερη τότε επιλέγει τον κόμβο που έστειλε το μήνυμα ως SGParent(εφόσον έχουν το ίδιο SGCost) και η τιμή της τοπικής μεταβλητής αλλάζει και γινεται ίση με την maximum τιμή. Στην δεύτερη περίπτωση(Εικόνα 42-Γραμμή κώδικα απο 16 εώς 58), όπου ένας κόμβος λαμβάνει ένα αναγνωριστικό σήμα απο ένα κόμβο σε ένα διαφορετικό τμήμα, συνειδητοποιεί ότι είναι στα σύνορα του τμήματος και έτσι είναι υποψήφιος κόμβος για να γίνει SGLeader.Εκλέγεται για να είναι SGLeader εάν η τιμή του CostToGateway είναι μικρότερη από την τιμή του SGCost.Το μήνυμα αναγνωριστικών σημάτων ενημερώνεται αναλόγως και ξαναστέλνεται.



Ο κόμβος 5 έχει να επιλέξει για κόμβο parent ανάμεσα στους κόμβους 2 και 4.Σύμφωνα με τον αλγόριθμο STS ο κόμβος 5 θα επιλέξει για parent τον κόμβο 2, γιατί έχει hopCount ίσο με 2 (hopCount(2)<hopCount(4)).Αντίθετα,σύμφωνα με τον αλγόριθμο STS-Mobile θα επιλέξει για parent τον κόμβο 4 ,γιατί το maximum κόστος του δεξιού υποδέντρου είναι 8,ενώ το maximum κόστος του αριστερού είναι 15.

Εικόνα 41

Φάση διάδοσης αποτελέσματος STS-Mobile

Δεδομένου ότι έχει γίνει η διαμόρφωση του δικύου δεν λαμβάνουμε υπόψιν την κίνηση των κόμβων και το κόστος.Θεωρούμε ότι η φάση αυτή διαρκεί μικρό χρονικό διάστημα για να προκύψει link failure και οι μετρήσεις δρομολογούνται χωρίς πρόβλημα στον κόμβο gateway.Τα βήματα που ακολουθούνται είναι ακριβώς ίδια με αυτά του αλγρορίθμου STS.

Κώδικας java για την φάση διαμόρφωσης δικτύου του αλγορίθμου STS-Mobile.

```
1   MaxCost=maximumCost(source,thisNode);
2
3   if(( MaxCost<thisNode.getCost()
4      || (MaxCost==thisNode.getCost()
5      && (source.getHopcount()+1)<thisNode.getHopcount())
6      || (MaxCost==thisNode.getCost()
7      && (source.getHopcount()+1)==thisNode.getHopcount()
8      && source. getMyLocation().
9      strictlyCloser(allNodes.get(thisNode.getParentId()).getMyLocation()))){
10
11       thisNode.setCostToGateway(MaxCost);
12       thisNode.setHopcount(source.getHopcount()+1);
13       thisNode.setParentId(source.getID());
14  }
15
16  if(equalVectors(thisNode.getSGVector(),source.getSGVector())){
17
18
19    MaxSGCost = maximumSGCost(source,thisNode);
20
21    if ((thisNode.getSGCost()>source.getSGCost())
22        || (thisNode.getSGCost()==source.getSGCost()
23        && MaxSGCost<thisNode.getSGCostToSGLeader() )
24        || (thisNode.getSGCost()==source.getSGCost()
25        && MaxSGCost==thisNode.getSGCostToSGLeader()
26        && ((thisNode.getSGDistance()>source.getSGDistance())
27        || (thisNode.getSGDistance()==source.getSGDistance()
28        && (source.getDistanceToSGLeader()+1)<thisNode.getDistanceToSGLeader())
29        || (thisNode.getSGDistance()==source.getSGDistance()
30        && source. getleaderLoc().strictlyCloser(thisNode.getleaderLoc())))))){
31
32          thisNode.setSGCost(source.getSGCost());
33          thisNode.setSGCostToSGLeader(MaxSGCost);
34          thisNode.setSGParentId(source.getID());
35          thisNode.setSGDistance(source.getSGDistance());
36          thisNode.setDistanceToSGLeader((source.getDistanceToSGLeader()+1));
37          thisNode.setleaderLoc(source.getleaderLoc());
38          }
39  }else if(equalVectors(thisNode.getSGVector(),source.getSGVector())==false){
40
41         if((thisNode.getSGCost()>thisNode.getCost())
42            || (thisNode.getSGCost()==thisNode.getCost()
43            && thisNode.getSGDistance()>thisNode.getHopcount())
44            ||((thisNode.getSGCost()==thisNode.getCost())
45            && (thisNode.getSGDistance()==thisNode.getHopcount())
46            && source.getID()==thisNode.getParentId()
47            &&  source.getID()!=thisNode.getSGParentId()
48            && thisNode.getMyLocation().strictlyCloser(thisNode.getleaderLoc()))){
49
50               thisNode.setSGCost(thisNode.getCost());
51               thisNode.setSGCostToSGLeader(0);
52               thisNode.setSGParentId(source.getID());
53               thisNode.setSGDistance(thisNode.getHopcount());
54               thisNode.setDistanceToSGLeader(0);
55               thisNode.setleaderLoc(thisNode.getMyLocation());
56
57         }
58  }
59
60
61
62
```

Εικόνα 42

Ανδιαμόρφωση δικτύου σε περίπτωση αστοχίας σύνδεσης(link Failure)

Όπως προαναφέραμε,κύρια επιδίωξη στο σχεδιασμό των αλγορίθμων επεξεργασίας πολλαπλών επερωτήσεων με κινούμενους κόμβους , είναι να ελαχιστοποιηθεί το κόστος επικοινωνίας λόγω των πολλών link failures.Η μείωση των link failures επιτυγχάνεται με την σωστή διαμόρφωση του δικτύου ακολουθώντας τον παραπάνω κώδικα.Σε αυτο το σημείο είναι σημαντικό να παρουσιάσουμε τα βήματα που ακολουθεί ένας κόμβος όταν χάνεται η επικοινωνία με τον κόμβο parent και SGParent.
Ο έλεγχος για αστοχίες σύνδεσης(link failures) γίνεται λίγο πρίν αρχίσει μια εποχή έτσι ώστε το δίκτυο να είναι σωστά διαμορφωμένο και όλοι οι κόμβοι να μπορούν να στείλουν τις μετρήσεις στον κόμβο gateway.Αυτο το χρονικό σημείο για τον έλεγχο και την αναδιαμόρφωση είναι το βέλτιστο λόγω της κίνησης των κόμβων.Η χρονική διάρκεια μιας εποχής είναι πολύ μικρή έτσι ώστε να προκύψει link failure κατα την φάση διάδοσης αποτελέσματος.Αστοχία σύνδεσης παρατηρείται όταν ένας κόμβος χάνει την επικοινωνία και με τον κόμβο parent και με τον κόμβο SGParent. Θα μπορούσαμε να υποθέσουμε ότι link failure έχουμε αν χαθεί η επικοινωνία είτε με τον parent είτε με τον SGParent ,αλλά αυτή η περίπτωση οδηγεί σε πολλά link failures ,δηλαδή σε μεγάλο κόστος.Αρχικά,ας ορίσουμε κάποιες σημαντικές έννοιες :

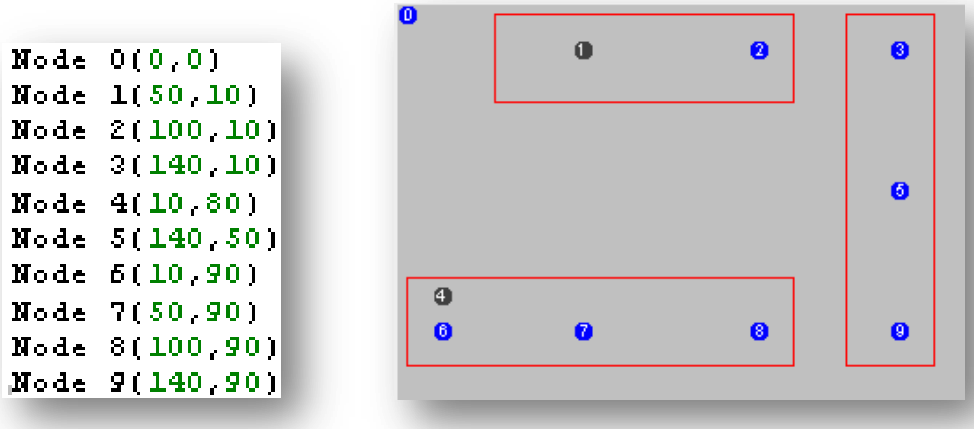
neighborsVector: Κάθε κόμβος όταν δέχεται αναγνωριστικά μηνύματα από τους γείτονες του αποθηκεύει τις τιμές των χαρακτηριστικών τους στον πίνακα neighbors.Όπως θα δούμε η χρήση του πίνακα είναι πολύ σημαντική σε περιπτώσεις αστοχίας σύνδεσης.
changedVariables: Στην περίπτωση που ο κόμβος αλλάξει τιμές από αυτές που είχε όταν έκανε τελευταία φορά broadcast, είτε επειδή άλλαξε parent ή SGParent είτε επειδή άλλαξε τιμές ο parent ή ο SGParent του(και έχει ενημερωθεί) τότε η μεταβλητή του changedVariables γίνετε ίση με true.
relation: Όταν ένας κόμβος A λάβει ένα αναγνωριστικό μήνυμα απο έναν γείτονα κόμβο B ,ελέγχει αν κόμβος B που έστειλε το μήνυμα ή οι κόμβοι στο υποδέντρο του B τον εχουν διαλέξει ως parent ή ως SGParent και αν ισχύει θέτουμε την μεταβλητή relation ίση με true.Ο έλεγχος αυτός είναι χρήσιμος γιατί αν η μετβλητή relation είναι ίση με true τότε ο κόμβος A δεν διαλέγει τον κόμβο B για parent ή για SGParent.Με αυτον τον τρόπο αποφεύγεται η δημιουργία κυκλικών διαδρομών (paths) που δεν καταλήγουν στον κόμβο gateway ή στον κόμβο SGLeader του κάθε κόμβου και όλες οι μετρήσεις καταλήγουν επιτυχώς στον gateway.

Τα βήματα που ακολουθεί ένας κόμβος με link failure είναι τα εξής:

1) Προσπελαύνει τον πίνακα neighborsVector και διαλέγει parent ή SGParent(εφόσον δεν είναι κενός).Αν ο πίνακας neighborsVector δεν είναι κενός και ο κόμβος διαλέξει parent και SGParent θα κάνει broadcast μόνο αν η changedVariables έιναι ίση με true
2) Αν ο πίνακας είναι κενός τότε ο κόμβος κάνει broadcast και ζητάει από τους νέους γείτονες του να του στείλουν τα στοιχεία τους για να επιλέξει parent και SGParent.Ο πίνακας μπορεί να είναι κενός είτε γιατί ο κόμβος έχει απομακρυνθεί από τους προηγούμενους γείτονες κόμβους που του είχαν στείλει μηνύματα, είτε γιατί οι γείτονες κόμβοι έχουν απομακρυνθεί από την εμβέλεια του κόμβου.
3) Αν ο κόμβος έχει προσπελάσει τον πίνακα αλλά δεν έχει επιλέξει parent και SGParent τότε ζητάει από τους νέους γείτονες του να του στείλουν τα στοιχεία τους για να επιλέξει parent και SGParent.Ο κόμβος δεν έχει επιλέξει parent και SGParent είτε γιατί έχει relation ίση με true με τους κόμβους που έχει αποθηκεύσει , είτε γιατί οι κόμβοι έχουν και αυτοί link Failure.
4) Αν ο κόμβος έχει λάβει όλα τα αναγνωριστικά μηνύματα απο τους γείτονες κόμβους και έχει επιλέξει parent και SGParent τότε κάνει broadcast.
5) Αν ο κόμβος Α έχει λάβει όλα τα αναγνωριστικά μηνύματα απο τους γείτονες κόμβους αλλά δεν έχει επιλέξει parent και SGParent, τότε ζητάει από έναν κόμβο γείτονα που έχει διαλέξει τον κόμβο Α ως parent ή ως SGParent(εξαρτάται ποιος έχει το μικρότερο hopCount) να επιλέξει άλλον κόμβο parent ή SGParent.Με αυτό τον τρόπο ο κόμβος Α δεν έχει πλέον relation με αυτόν τον κόμβο και μπορεί να τον διαλέξει ως parent και SGParent.

Εικόνα 43

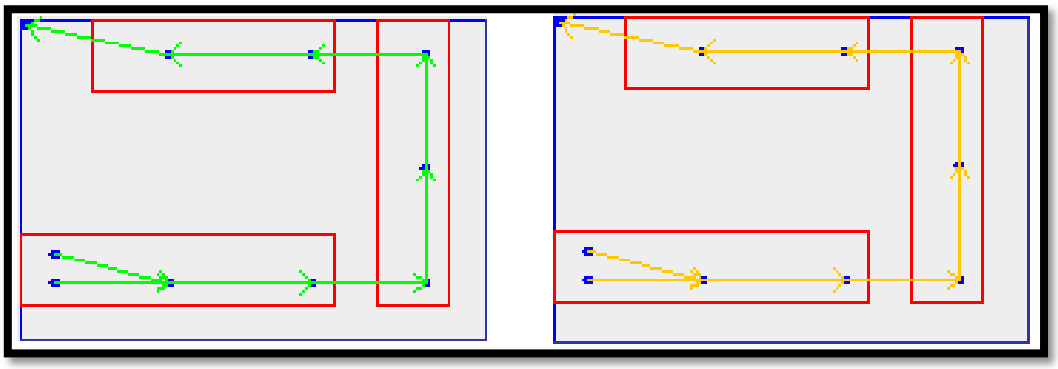
Παράδειγμα που δείχνουμε αναλυτικά πως λειτουργεί η αναδιαμόρφωση του δικτύου σε περιπτώσεις αστοχίας σύνδεσης .
Οι θέσεις των κόμβων είναι :



Εικόνα 44

-Το transmission radius των κόμβων είναι 55 μέτρα
-Οι πολλαπλές συναθροιστικές επερωτήσεις έχουν τα εξής χαρακτηριστικά (aggr x0, y0 xdim,ydim):
Query 1 (count 0,75 110,100)
Query 2(count 25,0 110,25)
Query 3(count 125,0 150,100)

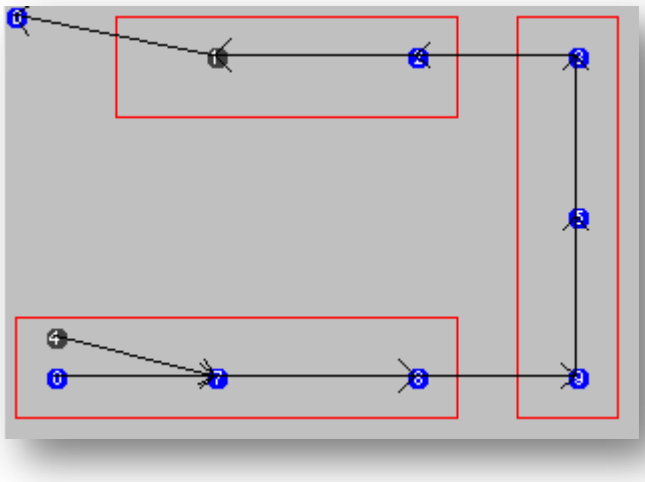
Η διαμόρφωση του δικτύου σύμφωνα με τον αλγόριθμο STS-Mobile είναι:



Εικόνα 45

Η διάδοση αποτελεσμάτων στην πρώτη εποχή γίνεται ως εξής:

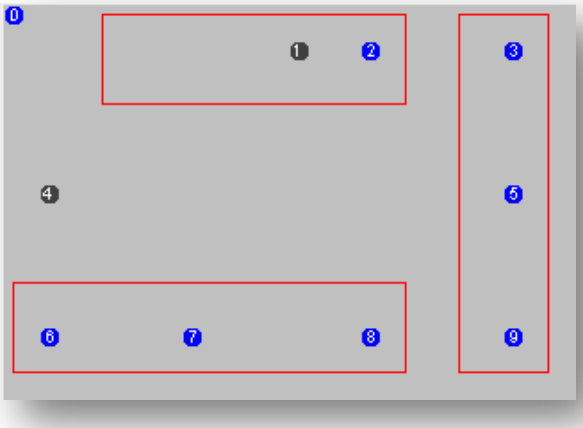
```
----------------------------------------
-----------EPOCH:1-----------
----------------------------------------

----------Propagation   Phase-----------

The node 4 send the message:(Value: 1--CV:[1][0][0][0]) to Node: 7
The node 6 send the message:(Value: 1--CV:[1][0][0][0]) to Node: 7
The node 7 send the message:(Value: 3--CV:[1][0][0][0]) to Node: 8
The node 8 send the message:(Value: 4--CV:[1][0][0][0]) to  parent Node:9
The node 9 send the message:(Value: 1--CV:[0][0][1][0]) to Node: 5
The node 9 send the message:(Value: 4--CV:[1][0][0][0]) to  parent Node:5
The node 5 send the message:(Value: 2--CV:[0][0][1][0]) to Node: 3
The node 5 send the message:(Value: 4--CV:[1][0][0][0]) to  parent Node:3
The node 3 send the message:(Value: 3--CV:[0][0][1][0]) to  parent Node:2
The node 3 send the message:(Value: 4--CV:[1][0][0][0]) to  parent Node:2
The node 2 send the message:(Value: 1--CV:[0][1][0][0]) to Node: 1
The node 2 send the message:(Value: 3--CV:[0][0][1][0]) to  parent Node:1
The node 2 send the message:(Value: 4--CV:[1][0][0][0]) to  parent Node:1
The node 1 send the message:(Value: 2--CV:[0][1][0][0]) to  parent Node:0
The node 1 send the message:(Value: 3--CV:[0][0][1][0]) to  parent Node:0
The node 1 send the message:(Value: 4--CV:[1][0][0][0]) to  parent Node:0
```



Εικόνα 46

Οι θέσεις των κόμβων μετά απο Χ δευτερόλεπτα :

```
Node 0(0,0)
Node 1(80,10)
Node 2(100,10)
Node 3(140,10)
Node 4(10,50)
Node 5(140,50)
Node 6(10,90)
Node 7(50,90)
Node 8(100,90)
Node 9(140,90)
```



Εικόνα 47

Έλεγχος για link failure:

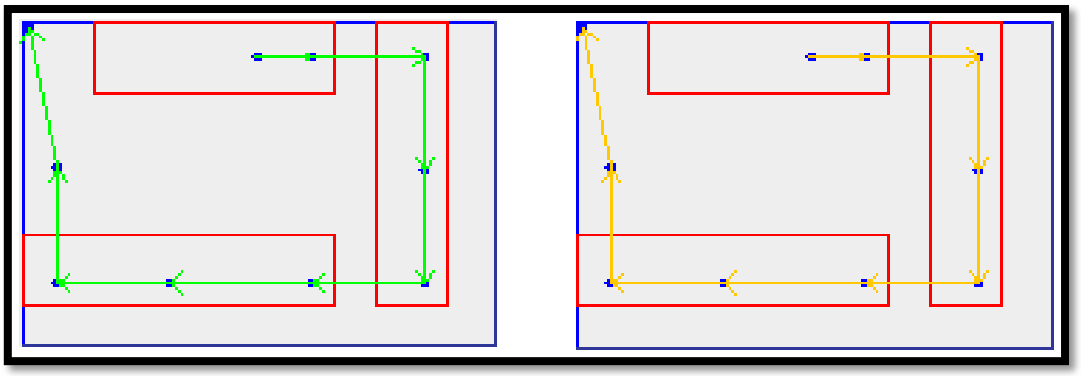
Ο κομβός 1 απομακρύνεται απο τον κόμβο 0(gateway) που ήταν και parent και SGParent του και έτσι χάνει την επικοινωνία μαζί του.Επίσης , ο κόμβος 4 απομακρύνεται απο τον κόμβο 7 που ήταν και parent και SGParent του και έχουμε αστοχία σύνδεσης.Θα πρέπει να γίνει αναδιαμόρφωση δικτύου ξεκινώντας απο τον κόμβο 4 γιατί βρίσκεται πιο κοντά στον κόμβο gateway.
Ακολουθώντας τα βήματα που περιγράψαμε παραπάνω:
Ο κόμβος 4 :
Ελέγχει τον πίνακα neighborsVector.Απο τον πίνακα επιλέγει ως parent και SGParent τον κόμβο 6.Ο κόμβος 4 δεν γνωρίζει τα στοιχεία του καινούργιου γείτονα 0 και έτσι δεν μπορεί να τον επιλέξει.Ο κόμβος 4 πλέον δεν έχει link failure και θα κάνει broadcast.
Ο κόμβος 1 :
Ελέγχει τον πίνακα neighborsVector.Ο κόμβος δεν επιλέγει SGParent και parent αν και έχει τα στοιχεία του κόμβου 2 ,γιατί ο κόμβος 2 τον είχε διαλέξει για parent και SGParent(relation ισο με true).Σύμφωνα λοιπόν με όσα προαναφέραμε, ο κόμβος 1 κάνει broadcast και ζητάει απο τους γείτονες να του στείλουν τα στοιχεία τους για να επιλέξει parent και SGParent.Ο μόνος γειτονάς του είναι ο κόμβος 2 τον οποίο όμως δεν μπορεί να διαλέξει.Του ζητάει λοιπόν να διαλέξει άλλον parent και SGParent. Έτσι λοιπόν και ο κόμβος 2 χάνει την συνδεσή του με τον κόμβο 1 και ακολουθεί και αυτός τα ίδια βήματα.
Ο κόμβος 2:
Ελέγχει τον πίνακα neighbors.Ο κόμβος δεν επιλέγει SGParent και parent αν και έχει τα στοιχεία του κόμβου 3 ,γιατί ο κόμβος 3 τον είχε διαλέξει για parent και SGParent(relation ισο με true).Έτσι λοιπόν και ο κόμβος 2 κάνει broadcast και ζητάει απο τους γείτονες να του

στείλουν τα στοιχεία τους για να επιλέξει parent και SGParent. Ο μόνος γειτονάς του είναι ο κόμβος 3 τον οποίο όμως δεν μπορεί να διαλέξει Όπως συνέβη και νωρίτερα, ζητάει απο τον κόμβο 3 να επιλέξει άλλον parent και SGParent.Ο κόμβος 3 χάνει την συνδεσή του με τον κόμβο 2 και ακολουθεί και αυτός τα ίδια βήματα.

Η ίδια διαδικασία συνεχίζεται εως ότου φτάσουμε στον κόμβο 4:

Ο κόμβος 4:
Όπως είδαμε στην αρχή ο κόμβος 4 διάλεξε νέο parent και SGParent τον κόμβο 6.Όμως ο κόμβος 6 του ζητάει να βρεί άλλον parent και SGParent.Ο κόμβος 4 επειδή δεν γνωρίζει τα στοιχεία άλλου γείτονα κάνει broadcast και ζητάει να του στείλουν μήνυμα οι γείτονες που δεν γνωρίζει τα στοιχεία τους.Ο κόμβος 0 του στέλνει μήνυμα και ο κόμβος 4 τον διαλέγει ως parent και SGParent.Με την σειρά του κάνει ξανά broadcast και ενημερώνει τον κόμβο 6.
Ο κόμβος 6:
Λαμβάνει το μήνυμα απο τον κόμβο 4 και τον διαλέγει για parent και SGParent.Στην συνέχεια κάνει και αυτός broadcast και ενημερώνει τον κόμβο 7.

Η ίδια διαδικασία συνεχίζεται εως ότου φτάσουμε στον κόμβο 1.Η τελική διαμόρφωση του δικτύου είναι:

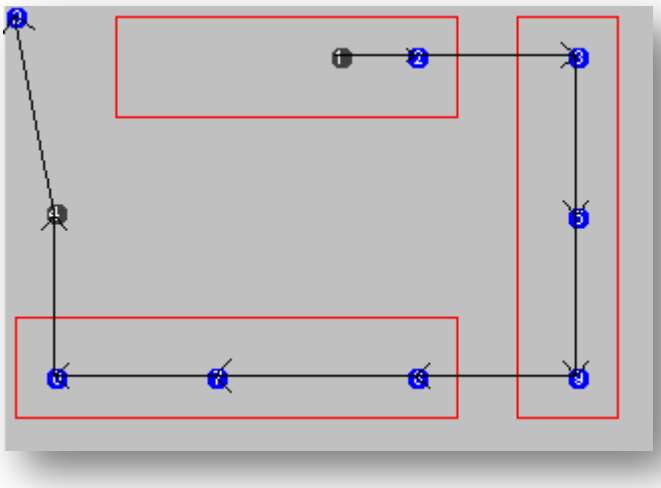


Εικόνα 48

Η διάδοση αποτελεσμάτων γίνεται ως εξής:

```
----------------------------------------
----------------------------------------
----------------------------------------

----------Propagation   Phase-----------

The node 1 send the message:(Value: 1--CV:[0][1][0][0]) to Node: 2
The node 2 send the message:(Value: 2--CV:[0][1][0][0]) to  parent Node:3
The node 3 send the message:(Value: 1--CV:[0][0][1][0]) to Node: 5
The node 3 send the message:(Value: 2--CV:[0][1][0][0]) to  parent Node:5
The node 5 send the message:(Value: 2--CV:[0][0][1][0]) to Node: 9
The node 5 send the message:(Value: 2--CV:[0][1][0][0]) to  parent Node:9
The node 9 send the message:(Value: 3--CV:[0][0][1][0]) to  parent Node:8
```

```
The node 9 send the message:(Value: 2--CV:[0][1][0][0]) to  parent Node:8
The node 8 send the message:(Value: 1--CV:[1][0][0][0]) to Node: 7
The node 8 send the message:(Value: 3--CV:[0][0][1][0]) to  parent Node:7
The node 8 send the message:(Value: 2--CV:[0][1][0][0]) to  parent Node:7
The node 7 send the message:(Value: 2--CV:[1][0][0][0]) to Node: 6
The node 7 send the message:(Value: 3--CV:[0][0][1][0]) to  parent Node:6
The node 7 send the message:(Value: 2--CV:[0][1][0][0]) to  parent Node:6
The node 6 send the message:(Value: 3--CV:[1][0][0][0]) to  parent Node:4
The node 6 send the message:(Value: 3--CV:[0][0][1][0]) to  parent Node:4
The node 6 send the message:(Value: 2--CV:[0][1][0][0]) to  parent Node:4
The node 4 send the message:(Value: 3--CV:[1][0][0][0]) to  parent Node:0
The node 4 send the message:(Value: 3--CV:[0][0][1][0]) to  parent Node:0
The node 4 send the message:(Value: 2--CV:[0][1][0][0]) to  parent Node:0
```



Εικόνα 49

Κεφάλαιο 6-Προσομοίωση

Η προσομοίωση έγινε σε γλώσσα java και χρησιμοποιήσαμε το eclipse , ένα ελεύθερης διανομής (freeware) ολοκληρωμένο περιβάλλον.Παρακάτω παρουσιάζουμε τις κλάσεις και τις πιο σημαντικές μεθόδους που έχουμε υλοποιήσει.

Η κλάση Main

Η κλάση Main περιέχει την μέθοδο main στην οποία αρχίζει και τελειώνει το πρόγραμμα.Η main αρχικά διαβάζει τα δεδομένα που εισάγει ο χρήστης και δημιουργεί τα αντικείμενα για την διαμόρφωση ενός ολοκληρωμένου sensor network.

Η κλάση SensorNetwork

Η κλάση SensorNetwork περιέχει τις σημαντικότερες μεθόδους για την προσομοίωση ενός Sensor Network.Οι μέθοδοι αυτοί είναι:

addNode:

Η μέθοδος addNode τοποθετεί τους κόμβους στον χώρο.Εφόσον δεν έχει οριστεί απο τον χρήστη, οι κόμβοι τοποθετούνται σε μορφή πλέγματος .Επίσης επιλέγει τυχαία ποιοι θα είναι οι κινούμενοι κόμβοι (εφόσον δεν έχει οριστεί απο τον χρήστη) ανάλογα με το ποσοστό που έχει θέσει ο χρήστης.

createSegmnets:

Η μέθοδος createSegments δέχεται ως όρισμα όλες τις επερωτήσεις και διαμορφώνει τα segments.Ενημερώνει δηλαδή την μεταβλητή των κόμβων SGVector.

nodeNeighbors:

Η μέθοδος αυτή ενημερώνει κάθε κόμβο για τους γειτονές του.

configurationPhaseA ΚΑΙ configurationPhaseB:

Στις δύο αυτές μεθόδους υλοποιείται ο αλγόριθμος διαμόρφωσης του δικτύου όπως και ο αλγόριθμος διαχείρισης αστοχίων σύνδεσης(link falures).Είναι οι δύο πιο σημαντικοί μέθοδοι του προγράμματος

maximumCost:

Η μέθοδος maximumCost υπολογίζει το κόστος ανάμεσα στον κόμβο που στέλνει το αναγνωριστικό μήνυμα και στον κόμβο που λαμβάνει το μήνυμα και επιστρέφει το μέγιστο κόστος ανάμεσα στο κόστος που υπολόγισε και στην τιμή της μεταβλητής CostToGateway του κόμβου που έστειλε το μήμυμα.

maximumSGCost:

Η μέθοδος maximumSGCost υπολογίζει το κόστος ανάμεσα στον κόμβο που στέλνει το αναγνωριστικό μήνυμα και στον κόμβο που λαμβάνει το μήνυμα και επιστρέφει το μέγιστο κόστος ανάμεσα στο κόστος που υπολόγισε και στην τιμή της μεταβλητής SGCostToSGLeader του κόμβου που έστειλε το μήμυμα.

propagationPhase :

Στην μέθοδο αυτή υλοποιείται η φάση διάδοσης αποτελέσματος όπως περιγράψαμε στην ανάλυση του αλγορίθμου STS.

randomMotion:

Η μέθοδος randomMotion καθορίζει όλη την κίνηση των κινούμενων κόμβων για όλες τις εποχές.

randomQueries:
Η μέθοδος randomQueries δημιουργεί τυχαίες επερωτήσεις εφόσον ο χρήστης δεν έχει ορίσει συγκεκριμένες.
checkForLinkFailures:
Η μέθοδος checkForLinkFailures ελέγχει το δίκτυο αισθητήρων για αστοχίες σύνδεσης
changedVariables:
Η μέθοδος changedVariables ελέγχει αν ο κόμβος έχει διαφορετικές τιμές στις μεταβλητές του σε σχέση με την τελευταία φορά που έκανε broadcast.
resetNodeVariables:
Η μέθοδος resetNodeVariables αυτή αρχικοποιεί τις τιμές των μεταβλητών ενός κόμβου.Συνήθως χρησιμποοιείται όταν σε έναν κόμβο παρατηρηθεί αστοχία σύνδεσεις.
relation:
Η μέθοδος αυτή ελέγχει την σχέση ανάμεσα στον κόμβο που λαμβάνει το μήνυμα και στον κόμβο που στέλνει το μήνυμα και δίνει τιμή στην μεταβλητή relation.
backwardLinkFailureSearch:
Η μέθοδος αυτή ελέγχει αν στο υποδέντρο που έχει ως φύλλο τον κόμβο που την καλεί, υπάρχει κόμβος που έχει link failure ή changedVariables ίση με true.

Η κλάση Node

Η κλαση Node αντιπροσωπεύει τους κόμβους ενός δικτύου αισθητήρων.Οι μεταβλητές ενός κόμβου είναι:

double velocityX
double velocityY
int[] SGVector
int nodeId
int hopcount
Coordinate myLoc
int parentId
int transRadius
int SGParentId
int SGDistance
int distToSGLeader
Coordinate leaderLoc
double CostToGateway
double SGCost
double SGCostToSGLeader
boolean broadcast
boolean linkFailure
int propagatePosition
Vector<Neighbor> neighbors
Vector<Pair> pairsMessage
Vector<Waypoint> waypoints
Node lastVarBroadcast
int LinkFailureSolutionType

Οι μεταβλητές στις οποίες δεν έχουμε αναφερθεί είναι η *propagatePosition* ,η οποία καθορίζει την σειρά με την οποία οι κόμβοι στέλνουν μηνύματα στην φάση διάδοσης αποτελέσαμτος ,η *lastVarBroadcast* στην οποία αποθηκεύονται οι τιμές του κόμβου την τελευταία φορά που έκανε broadcast και η *LinkFailureSolutionType* η οποία καθορίζει αν ο κόμβος θα χρησιμοποιεί ή όχι τον πίνακα *neighbors*, σε περιπτώσεις αστοχίας σύνδεσης.

Οι σημαντικότεροι μέθοδοι της κλάσης Node είναι:
getMessageFromNeighborNodesConfiguration:

Η μέθοδος αυτή επιστρέφει των αριθμό των γειτόνων απο τους οποίους έλαβε μήνυμα ο κόμβος.

checkFromTheNeighborsVector:

Η μέθοδος επιστρέφει των αριθμό των κόμβων που έχει ελένξει ο κόμβος απο τον πίνακα neighbors.

KnowledgeOfVarFromNeighborsVector:

Η μέθοδος αυτή επιστρέφει των αριθμό των γειτονικών κόμβων απο τους οποίους ο κόμβος γνωρίζει τα στοιχεία.

myValue:

Η μέθοδος αυτή ορίζει στον κόμβο την μετρησή του.

getOutPutPairs:

Η μέθοδος αυτή επιστρέφει τον πίνακα pairMessage που περιέχει τα μηνύματα εξαγωγής που πρόκειται να στείλει ο κόμβος στην φάση διάδοσης αποτελέσματος.

inPutPair:

Στην μέθοδο αυτή γίνετε συγχώνευση και διαχωρισμός των μηνυμάτων ,όπως περιγράψαμε στην φάση διάδοσης αποτελέσματος.

Η κλάση Pair

Η κλαση Pair προσομοιώνει τα ζευγάρια εισαγωγής ή εξαγωγής.Αποτελείται απο δύο μεταβλητές int value και int[] CV.Ο πίνακας CV έχει μέγεθος ίσο με των αριθμό των επερωτήσεων και οι τιμές που παίρνει είναι 1 ή 0 ανάλογα με τις επερωτήσεις στις οποίες συμβάλλει η value.Η κλάση Pair περιέχει μεθόδους που είναι πολυ χρήσιμοι για την σύγκριση ενός πίνακα CV και ενός SGVector ,την διάσπαση CV και γενικά την επεξεργασία πινάκων CV.

Η κλάση Coordinate

Η κλάση Coordinate προσομοιώνει την θέση ενός αντικειμένου στο χώρο.Αποτελείται απο δύο μεταβλητές x και y.Χρησιμοποιείται απο τις κλάσεις Node,Query,SensorNetwork όχι μόνο γιατί ορίζει την θέση τους και την διαστασή τους αλλα και γιατί περιέχει χρήσιμες μεθόδους.Οι μέθοδοι αυτοί υπολόγίζουν την απόσταση μεταξύ των κόμβων, σε ποιό segment ανήκει ένας κόμβος και βρίσκουν τους γείτονες ενός κόμβου .

Η κλάση Query

Η κλαση Query προσομοιώνει τις επερωτήσεις του sensor network.Οι μεταβλητές της κλάσης Query είναι :

Coordinate queryXoYo
Coordinate queryDim
everyTime

Οι μεταβλητές *queryXoYo* και *queryDim* δείχνουν που αρχίζει και που τελειώνει η επερώτηση,δηλαδή τον χώρο που καλύπτει και η μεταβλητή *everyTime* είναι ο χρόνος που επαναλαμβάνεται κάθε εποχή.

Η κλάση DrawSensorNetwork

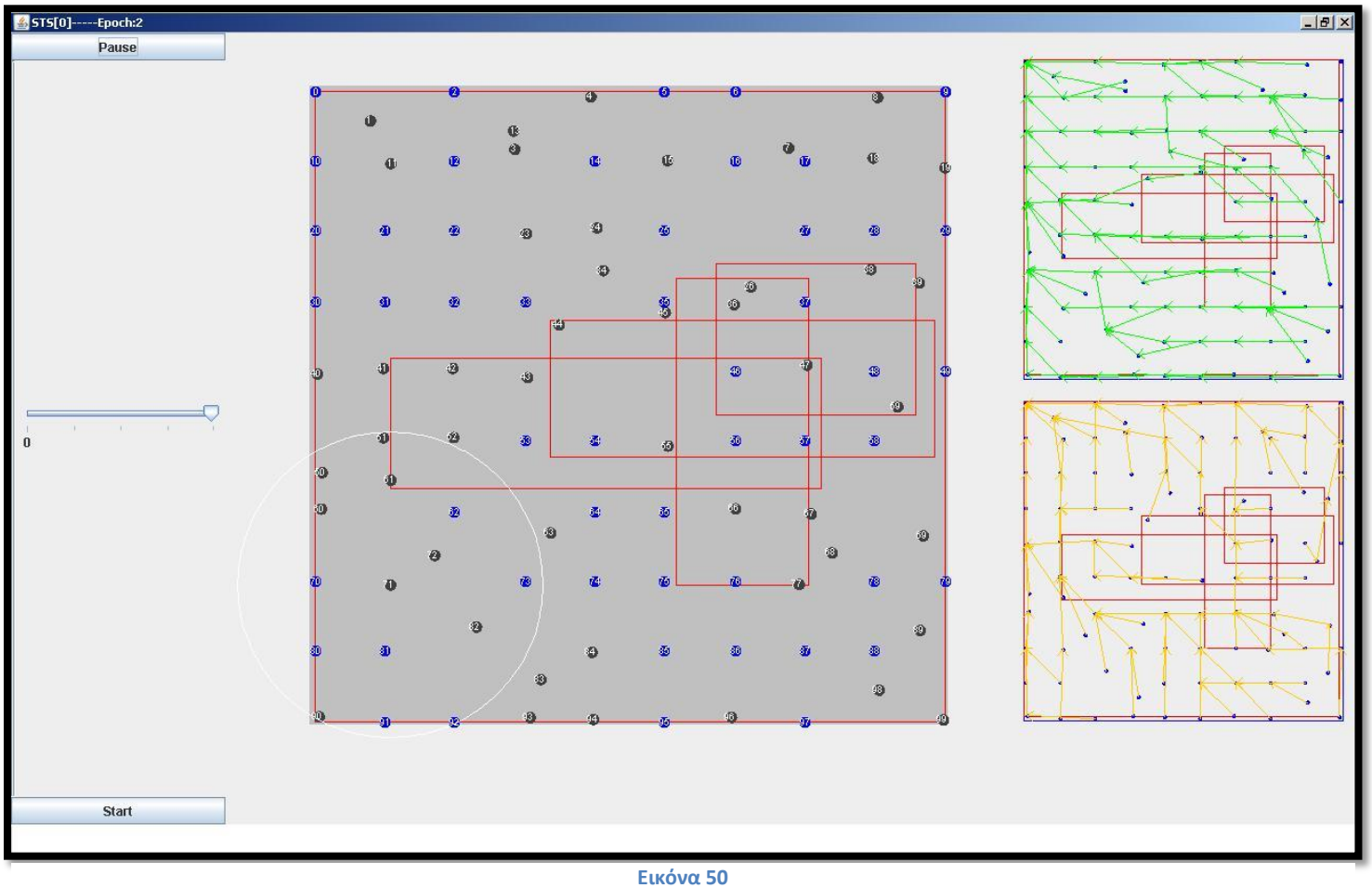
Η κλαση DrawSensorNetwork δημιουργεί ένα γραφικό περιβάλλον όπου μπορούμε να δούμε την φάση διαμόρφωσης δικτύου,τ ην φάση διάδοσης αποτελέσματος και την αναδιαμόρφωση του δικτύου

Η κλάση LineChart

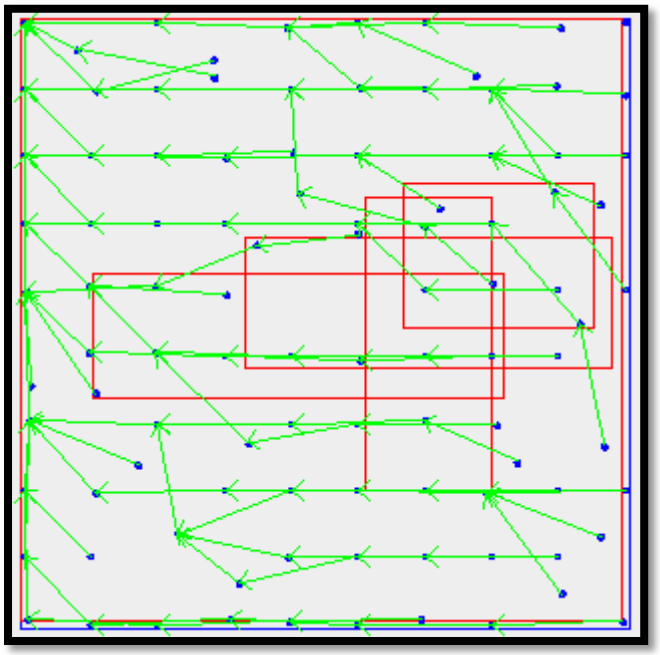
Η κλαση LineChart χρησιμοποιείται για την δημιουργία γραφημάτων.

Χαρακτηριστικά του προγράμματος προσομοίωσης.

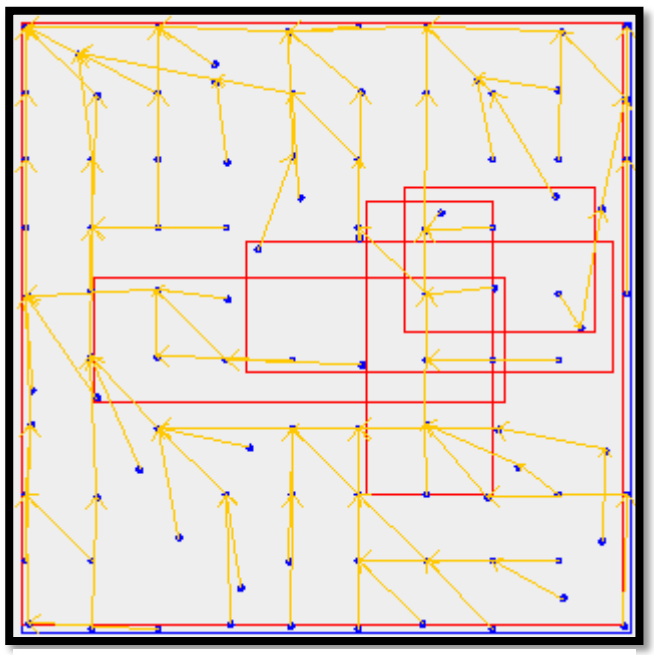
- Στο πρόγραμμα που έχουμε δημιουργήσει ο χρήστης έχει την επιλογή είτε να ορίσει συγκεκριμένες θέσεις στους κόμβους και να τους δώσει και ταχύτητα είτε απλά να εισάγει τον αριθμό των κόμβων που θέλει και το πρόγραμμα τους τοποθετεί αυτόματα σε μορφή πλέγματος.Στην δεύτερη περίπτωση θα πρέπει να ορίσει και το ποσοστό των κινούμενων κόμβων.
- Όπως συμβαίνει με τους κόμβους, έτσι και με τις επερωτήσεις ο χρήστης μπορεί να ορίσει συγκεκριμένες επερωτήσεις ή μπορεί να επιλέξει το πρόγραμμα να δημιουργήσει τυχαίες.
- Ο χρήστης μπορεί επίσης να ρυθμίσει τις διαστάσεις του δικτύου αισθητήρων,τον αριθμό των εποχών και το transmission range των κόμβων.Επιπλέον έχει την δυνατότητα να ρυθμίσει τους κόμβους ώστε να κινούνται με διαφορετικές ταχύτητες.
- Το πρόγραμμα όπως προαναφέραμε συγκρίνει δύο αλγορίθμους ,τον STS και τον STS-Mobile.Οι αλγόριθμοι έχουν την δυνατότητα αποθήκευσης στον πίνακα neighbors των στοιχείων των γειτονικών κόμβων.Ο πίνακας αυτός χρησιμοποιείται σε περιπτώσεις αστοχίας σύνδεσης.Ο χρήστης μπορεί να επιλέξει οι αλγόριθμοι να μην χρησιμοποιούν αυτόν τον πίνακα.Έτσι λοιπόν ,μπορούμε να παρατηρήσουμε την την λειτουργικότητα των κόμβων που δεν διαθέτουν αρκετό αποθηκευτικό χώρο.
- Ένα πολύ σημαντικό χαρακτηριστικό του προγράματος που υλοποιήσαμε είναι ότι κατα την διάρκεια εκτελεσής του δημιουργούναι text αρχεία τα οποία περιγράφουν λεπτομερώς όλες τις φάσεις των αλγορίθμων.
- Ένα εξίσου πολύ σημαντικό χαρακτηριστικό είναι η **γραφική προσομοίωση**(Εικόνα 50) του δικτύου αισθητήρων.Μπορούμε να δούμε όλο το δίκτυο αισθητήρων,τους κόμβους ,την κινησή τους ,τις επερωτήσεις και τα segments που δημιουργούνται.Μπορούμε επίσης να δούμε τις συνδέσεις των κόμβων,δηλαδή τις επιλογές τους ως προς τον parent(Εικόνα 51) και τον SGParent(Εικόνα 52).Στην φάση διαμόρφωσης του δικτύου όπως και στην φάση αναδιαμόρφωσης ,βλέπουμε τους κόμβους να κάνουν broadcast και στην φάση διάδοσης αποτελέσματος να στέλνουν τις μετρήσεις τους στους γείτονες κόμβους(Εικόνα 54).Ανα πάσα στιγμή μπορούμε να σταματήσουμε προσωρίνα,να επιβραδύνουμε ή να επιταχύνουμε την γραφική προσομοίωση και κάνοντας κλίκ πάνω σε οποιοδήποτε κόμβο μπορούμε να δούμε τα χαρακτηριστικά του(Εικόνα 53)..



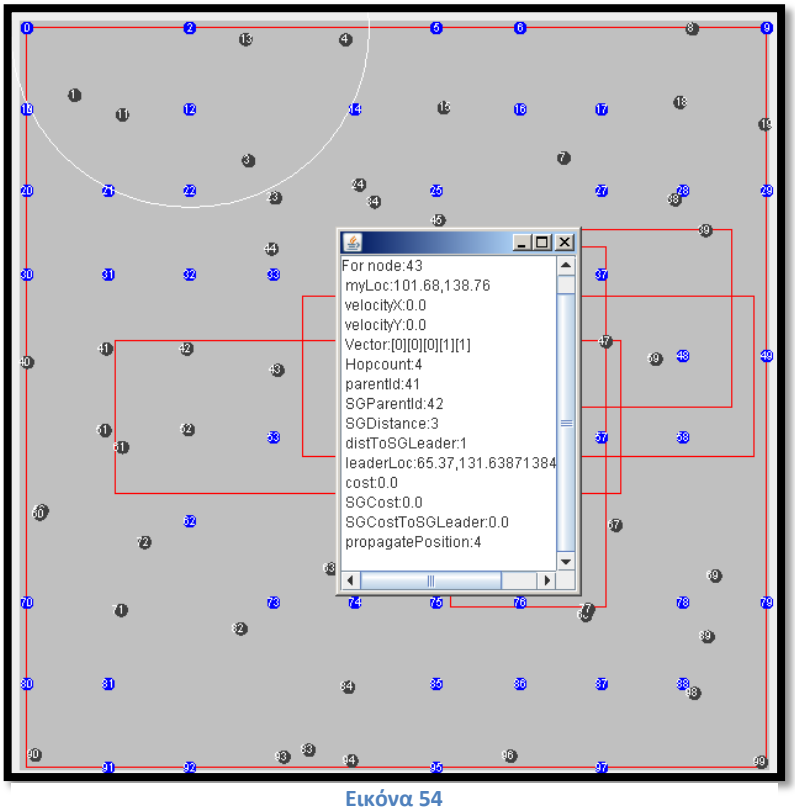
Εικόνα 50



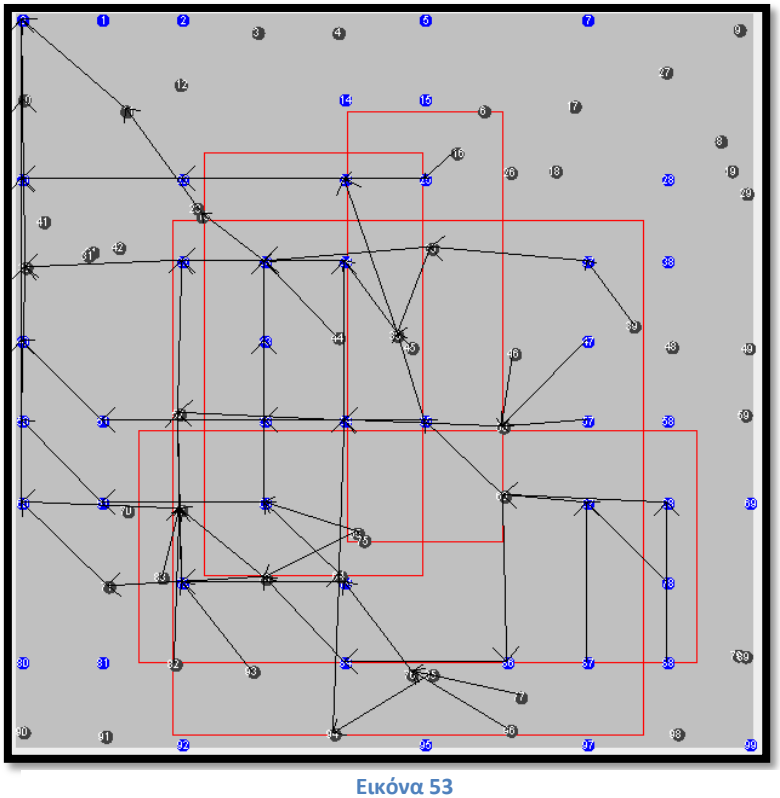
Εικόνα 51



Εικόνα 52



Εικόνα 54



Εικόνα 53

Κεφάλαιο 7-Πειραματικές Μετρήσεις

Μια εκτενής πειραματική αξιολόγηση πραγματοποιήθηκε για να γίνει η σύγκριση του προτεινόμενου αλγορίθμου STS-Mobile με τον STS.Για κάθε αλγόριθμο παίρνουμε δύο περιπτώσεις,μια περίπτωση είναι να χρησιμοποιεί τον πίνακα neighbors και άλλη είναι να μην τον χρησιμοποιεί.Έτσι λοιπόν συγκρίνουμε τέσσερις αλγορίθμους.Στα αποτελέσματα των πειραμάτων που ακολουθούν αναγράφονται οι επιδόσεις των τεσσάρων αλγορίθμων.Για κάθε περίπτωση μεταβάλουμε την τιμή μιας παραμέτρου και κρατάμε τις υπόλοιπες σταθερές. Έτσι λοιπόν παίρνουμε μετρήσεις μεταβάλλοντας :(i) το ποσοστό των κινούμενων κόμβων , (ii) την ταχύτητα των κόμβων, (iii) τον αριθμό των επερωτήσεων, (iv) το transmission range και (v) τον αριθμό των κόμβων.Για κάθε περίπτωση παρουσιάζουμε εφτά γραφήματα. Πριν αρχίσουμε την ανάλυση των γραφημάτων παρουσιάζουμε κάποιες υποθέσεις που έχουμε κάνει και κάποιους ορισμούς:

Έχουμε υποθέσει ότι τα αναγνωριστικά μηνύματα που στέλνονται ανάμεσα στους κόμβους στις φάσεις διαμόρφωσης και αναδιαμόρφωσης του δικτύου ,περιέχουν τις εξής μεταβλητές:

```
velocityX(16 bits)
velocityY(16 bits)
nodeId(16 bits)
hopcount(16 bits)
myLoc_X(16 bits)
myLoc_Y(16 bits)
parentid(16 bits)
SGParentId(16 bits)
SGDistance(16 bits)
distToSGLeader(16 bits)
leaderLoc(16 bits)
cost(16 bits)
SGCost(16 bits)
SGCostToSGLeader(16 bits)
linkFailure(1 bits)
SGVector(numberOfQueries bits)
```

Το μέγεθος ενός αναγνωριστικού μηνύματος είναι :

sizeOfConfigurationMessage=sizeOf(`velocityX`*)+sizeOf(*`velocityY`*)+sizeOf(*`nodeId`*)+sizeOf(*`hopcount`*)+sizeOf(*`myLoc_X`*)+sizeOf(*`myLoc_Y`*)+sizeOf(*`parentid`*)+sizeOf(*`SGParentId`*)+sizeOf(*`SGVector`*)+sizeOf(*`SGDistance`*)+sizeOf (*`distToSGLeader`*)+sizeOf(*`leaderLoc`*)+sizeOf(*`cost`*)+sizeOf(*`SGCost`*)+sizeOf(*`SGCostToSGLeader`*)+sizeOf(*`linkFailure`*)*

Ο συνολικός αριθμός των bits που στέλνονται στην φάση διαμόρφωση και στις φάσεις αναδιαμόρφωσης είναι :

*sumOfConfigurationBits=numberOfConfigurationMessages*sizeOfConfigurationMessage*

Επίσης έχουμε υποθέσει ότι τα μηνύματα που στέλνονται ανάμεσα στους κόμβους στην φάση διάδοσης αποτελέσματος αποτελούνται από τις εξής μεταβλητές:

```
value(16 bits)
CV(numberOfQueries bits)
```

Το μέγεθος ενός μηνύματος στην φάση αυτή είναι:

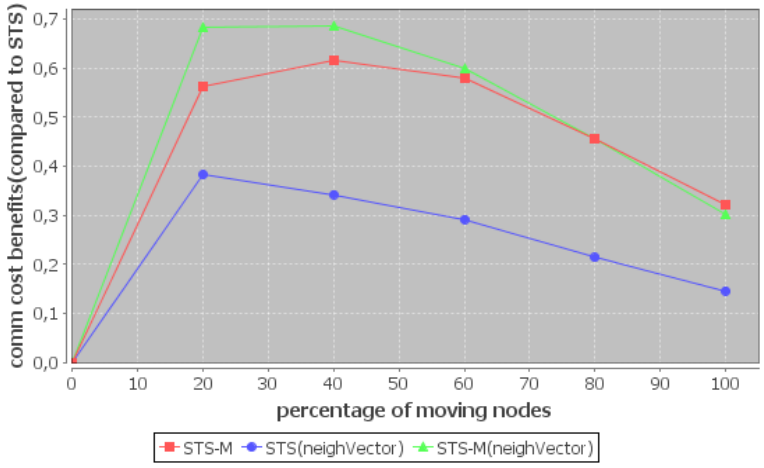
sizeOfPropagationMessage= sizeOf(`value`*)+sizeOf(*`CV`*)*

Ο συνολικός αριθμός των bits που στέλνονται στην φάση διάδοσης αποτελέσματος είναι:
*sumOfPropagationBits=numberOfPropagationMessages*sizeOfPropagationMessage;*
Ο συνολικός αριθμός των bits που στέλνονται είναι :
sumOfbits= sumOfConfigurationBits+ sumOfPropagationBits
Το όφελος ενός αλγορίθμου έναντι ενός άλλου υπολογίζεται ως εξής(Για παράδειγμα το όφελος του STS-M έναντι του αλγορίθμου STS):

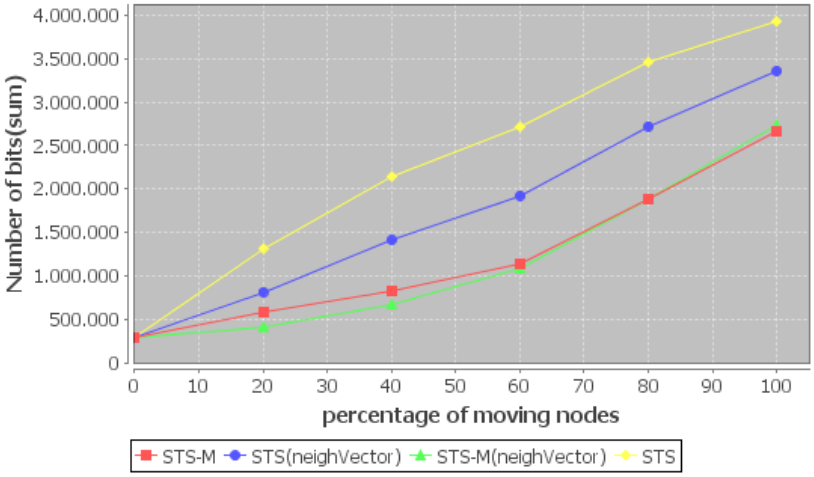
(sumOfbits (STS)- sumOfbits (STS-M))/ sumOfbits (STS)

Οι προεπιλεγμένες ρυθμίσεις προσομοίωσης είναι οι εξής: Αναπτύσσουμε 100 κόμβους σε ένα δίκτυο διαστάσεων 300×300m σε μορφή πλέγματος. Το transmission range έχει οριστεί στα 70m και ο αριθμός των επερωτήσεων που καλύπτουν το δίκτυο είναι πέντε.Η πρώτη επερώτηση καλύπτει όλο το δίκτυο και οι υπόλοιπες τέσσερις είναι τυχαίες.Η ταχύτητα των κινούμενων κόμβων ποικίλει και μπορεί να είναι μία από τις τέσσερις:0.05m/s,0.1m/s,0.5m/s και 0.9m/s με ίση πιθανότητα.Έτσι λοιπόν μεταβάλουμε κάθε φορά την τιμή μιας παραμέτρου και κρατάμε τις υπόλοιπες με την προεπιλεγμένη τιμή τους. Κάθε σημείο στα γραφήματα είναι ο μέσος όρος 40 επαναλήψεων και κάθε επανάληψη τρέχει για 100 εποχές.
Για κάθε παράμετρο που μεταβάλουμε την τιμή του παρουσιάσουμε 7 γραφήματα: (α)Το πρώτο γράφημα παρουσιάζει το όφελος των αλγορίθμων STS(με χρήση του πίνακα neighbors),STS-M (χωρίς την χρήση του πίνακα neighbors)και STS-M(με χρήση του πίνακα neighbors) έναντι του STS(την χρήση του πίνακα neighbors) .Τους αλγορίθμους που χρησιμοποιούν τον πίνακα neighbors βάζουμε δίπλα στην ονομασία τους την λέξη neighVector και αυτούς που δεν τον χρησιμοποιούν έχουμε απλά την ονομασία τους.(β)Στο δεύτερο γράφημα βλέπουμε τον συνολικό αριθμό bits που στέλνονται σε όλες τις φάσεις και στις 100 εποχές.(γ)Στο τρίτο γράφημα μπορούμε να δούμε τον αριθμό των bits που στέλνονται στις φάσεις διαμόρφωσης και στις φάσεις αναδιαμόρφωσης του δικτύου και στις 100 εποχές.(δ)Στο τέταρτο γράφημα παρατηρούμε τον συνολικό αριθμό bits που στέλνονται στις φάσεις διάδοσης αποτελέσματος και στις 100 εποχές.(ε)Το πέμπτο γράφημα δείχνει τον αριθμό των μηνυμάτων που στέλνονται στις φάσεις αναδιαμόρφωσης του δικτύου και στις 100 εποχές(δηλαδή τα μηνύματα που στέλνονται για να αντιμετωπιστούν οι αστοχίες σύνδεσης).(στ)Στο έκτο γράφημα παρουσιάζονται ο αριθμός των μηνυμάτων που στέλνονται στις φάσεις διάδοσης αποτελέσματος και στις 100 εποχές και τέλος στο (ζ) έβδομο γράφημα βλέπουμε τον αριθμό κόμβων που παρουσιάζουν αστοχία σύνδεσης ανά εποχή.

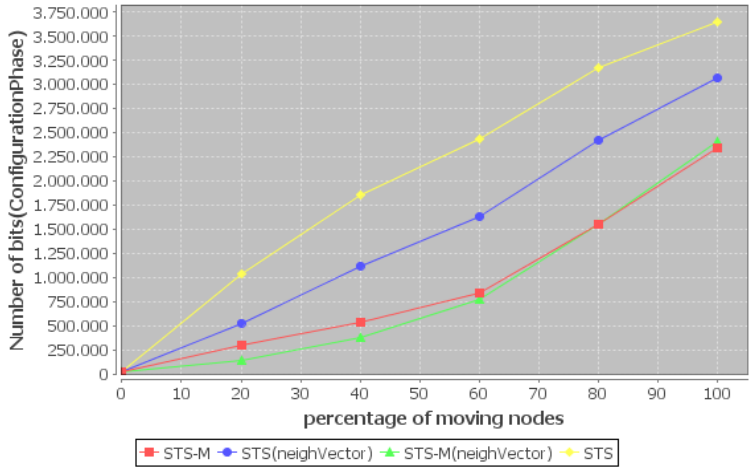
Μεταβάλλοντας το ποσοστό των κινούμενων κόμβων



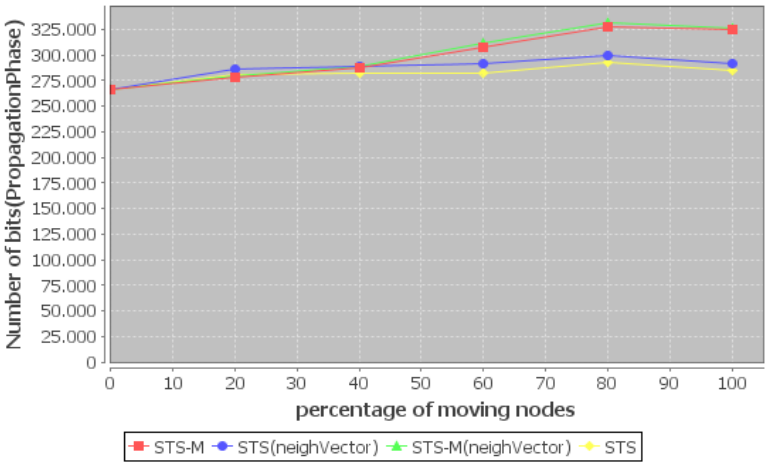
Γράφημα 1



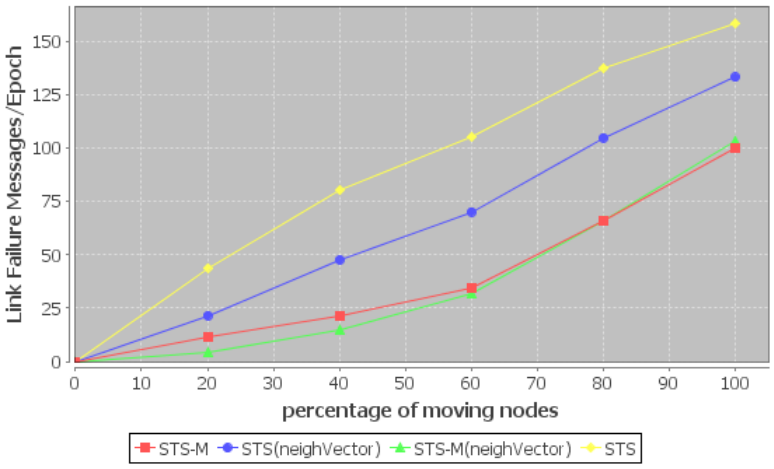
Γράφημα 2



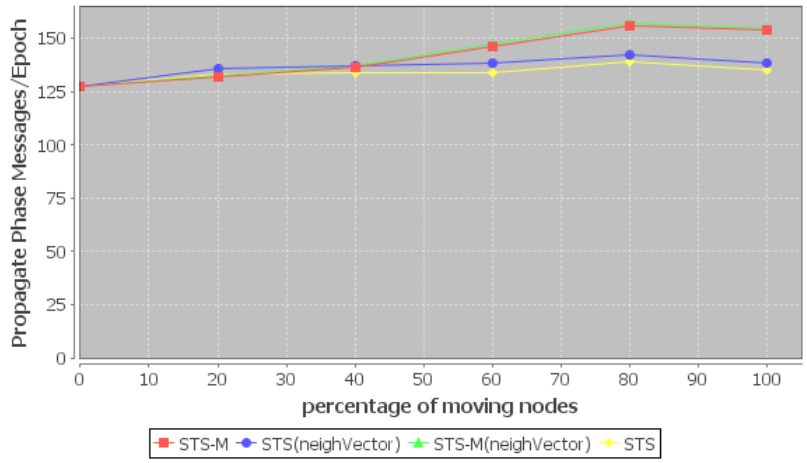
Γράφημα 3



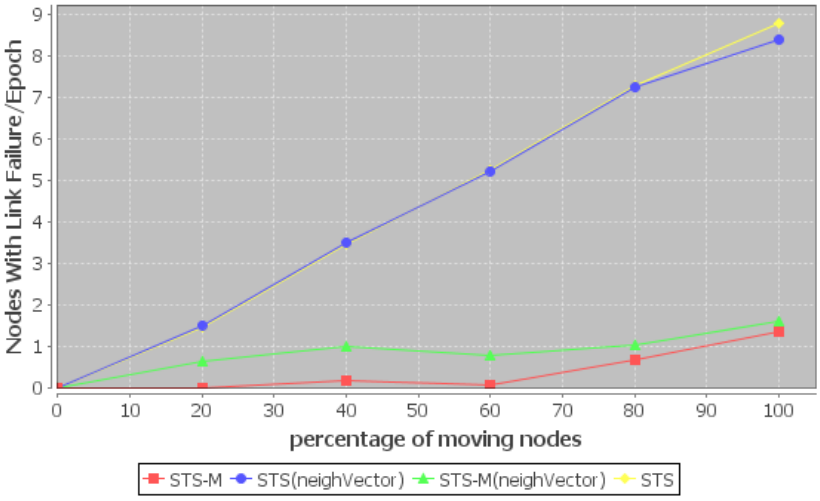
Γράφημα 4



Γράφημα 5



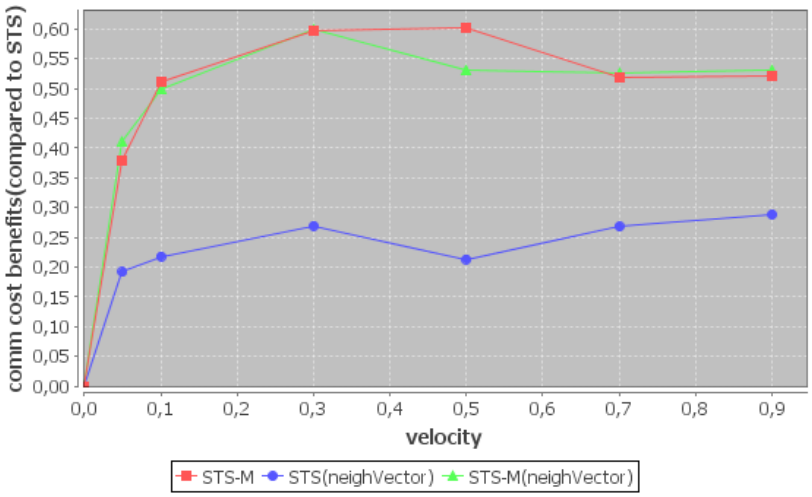
Γράφημα 6



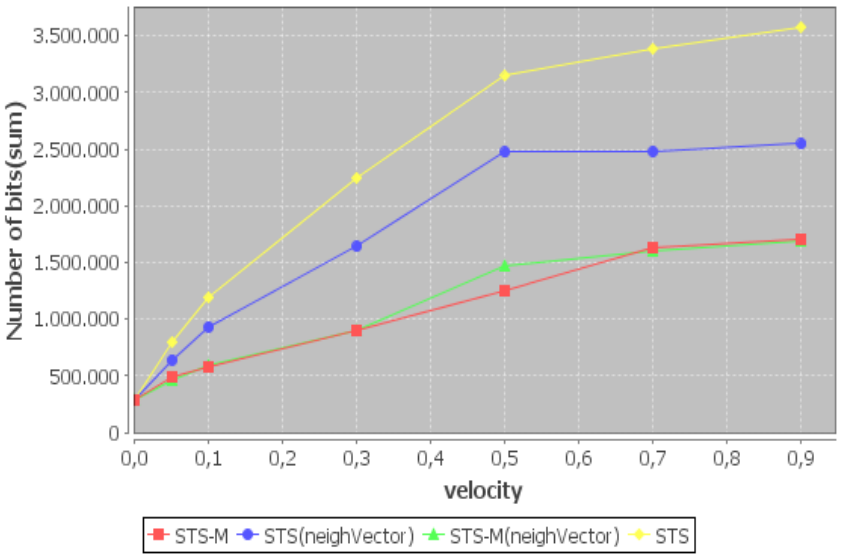
Γράφημα 7

Παρατηρήσεις:Στο γράφημα 1 βλέπουμε το όφελος και των τριών αλγορίθμων έναντι του αλγορίθμου STS.Αρχικά παρατηρούμε πόσο σημαντική είναι η χρήση του πίνακα neighbors για την αντιμετώπιση αστοχιών σύνδεσης και επίσης παρατηρούμε πόσο πιο αποδοτικός είναι ο αλγόριθμος STS-M έναντι του STS.Ωστόσο ,βλέπουμε ότι όταν το ποσοστό των κινούμενων κόμβων αρχίζει και ανεβαίνει πάνω από το 60% παρατηρούνται περισσότερα link failures ανά εποχή(γράφημα 7) στους αλγορίθμους STS-M και STS-M(neighVector),ενώ σε μικρότερα ποσοστά ο κόμβος STS-M άγγιζε το μηδέν 0 και ο STS-M(neighVector) ήταν κάτω από 1 linkFailure/epoch.Αυτό το γεγονός οδηγεί σε αύξηση του αριθμού των μηνυμάτων που στέλνονται για την αναδιαμόρφωση του δικτύου (γράφημα 5-μεγαλώνει η κλίση των αλγορίθμων STS-M και STS-M(neighVector)) έχοντας ως συνέπεια την μείωση της απόδοσης τους, παρ' όλα αυτά έχουν καλύτερη απόδοση από τους άλλους δύο.Στα γραφήματα 4 κα 6 παρατηρούμε ότι μετά το 60% οι αλγόριθμοι STS-M και STS-M(neighVector) στέλνουν περισσότερα μηνύματα στις φάσεις διάδοσης αποτελέσματος σε σχέση με τους υπόλοιπους(τα μηνύματα καταλήγουν στον gateway από δυσχερέστερα μονοπάτια), αλλά η αύξηση αυτή δεν επηρεάζει την απόδοση τους αφού οι αλγόριθμοι STS-M και STS-M(neighVector) στέλνουν λιγότερα μηνύματα για την αναδιαμόρφωση του δικτύου(Τα μηνύματα αυτά έχουν μεγαλύτερο μέγεθος από τα μηνύματα στην διάδοση αποτελέσματος).Είναι επίσης σημαντικό να τονίσουμε ότι από το 50% των κινούμενων κόμβων και μετά ,οι αλγόριθμοι STS-M και STS-M(neighVector) ταυτίζονται,δηλαδή η χρήση του πίνακα neighVector που αποθηκεύονται όλα τα στοιχεία των γειτονικών κόμβων δεν είναι απαραίτητη.Αυτό συμβαίνει γιατί όταν αυξάνονται οι κόμβοι που κινούνται , οι γείτονες ενός κόμβου αλλάζουν συνέχεια και έτσι τα στοιχεία που αποθηκεύονται στον πίνακα δεν είναι χρήσιμα,αφού οι γείτονες κόμβοι προηγούμενων εποχών έχουν απομακρυνθεί.

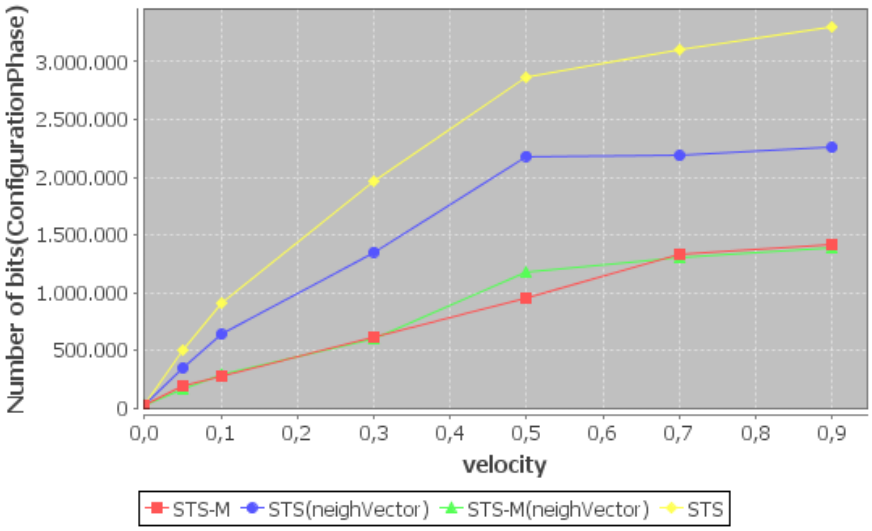
Μεταβάλλοντας την ταχύτητα των κόμβων



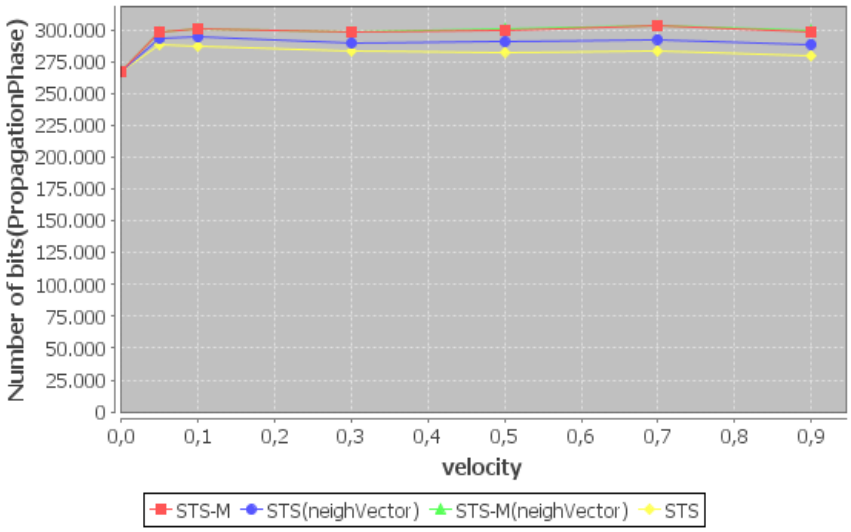
Γράφημα 8



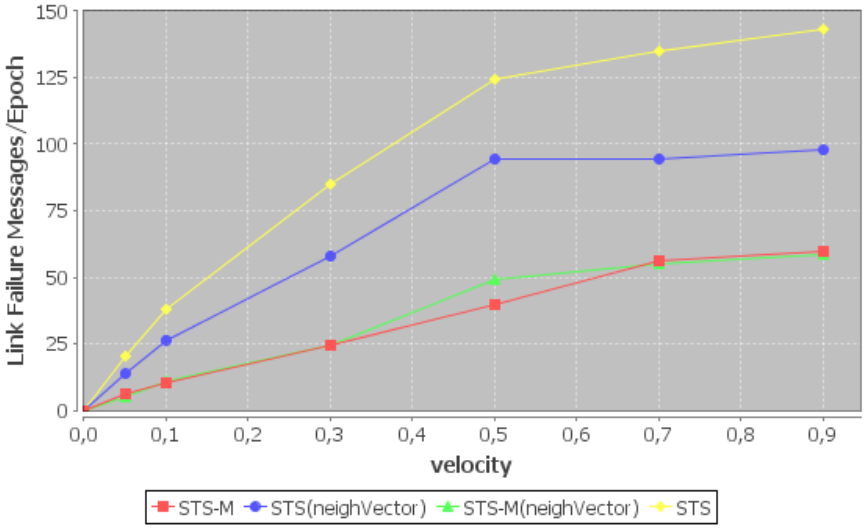
Γράφημα 9



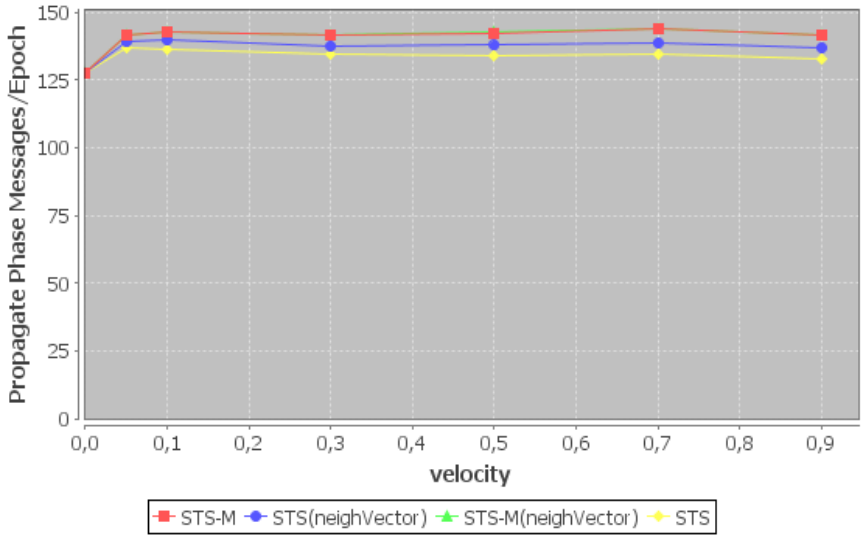
Γράφημα 10



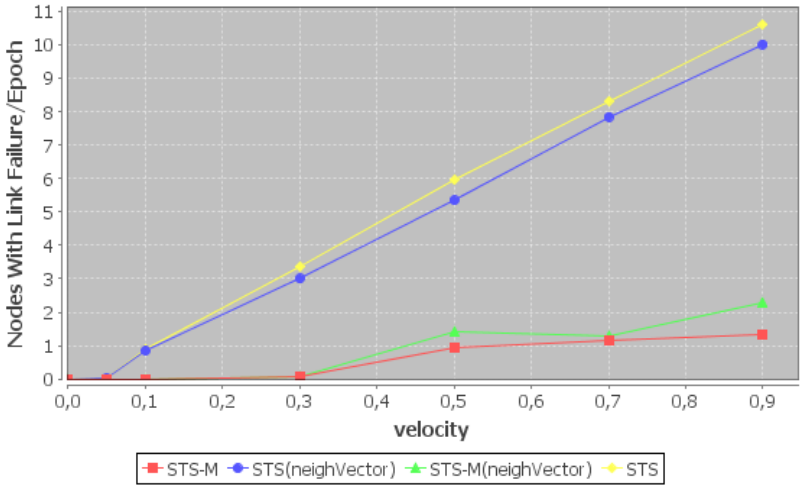
Γράφημα 11



Γράφημα 12



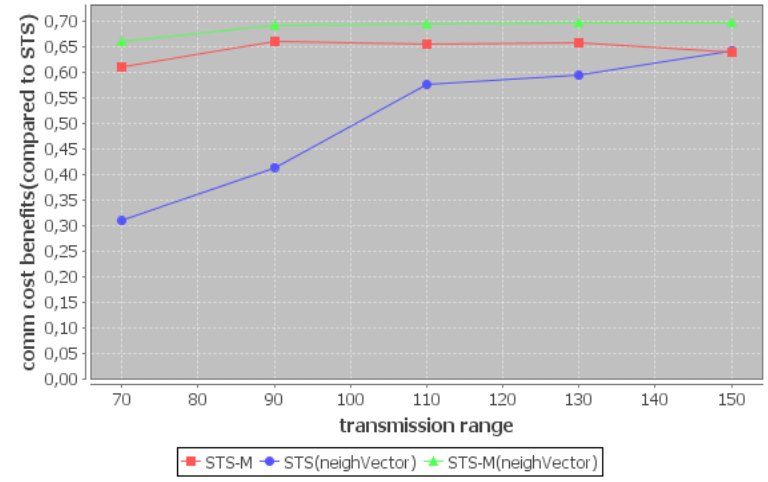
Γράφημα 13



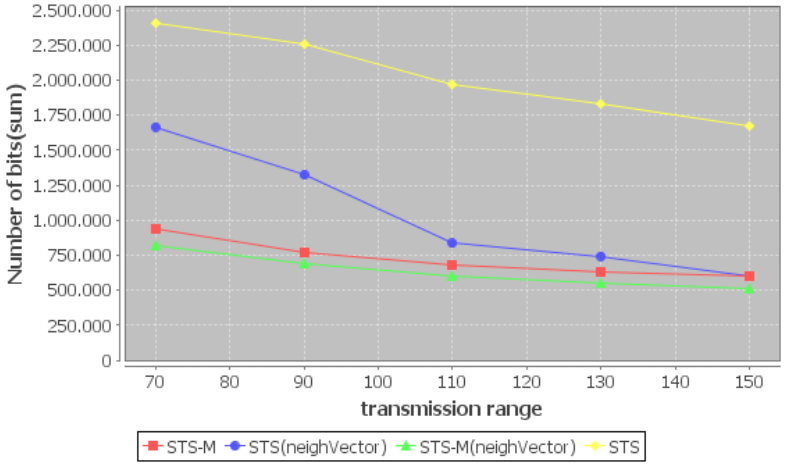
Γράφημα 14

Παρατηρήσεις:Αρχικά αυτό που παρατηρούμε είναι η πολύ καλή απόδοση του αλγορίθμου STS-M σε σχέση με τον STS(γράφημα 8) που παραμένει σχεδόν σταθερός για όλες ταχύτητες.Αυτό οφείλεται στον πολύ μικρό αριθμό των αστοχιών σύνδεσης ανά εποχή(γράφημα 14) ,δηλαδή στην πολύ καλή διαμόρφωση και αναδιαμόρφωση του δικτύου. Επίσης ένα πολύ σημαντικό αποτέλεσμα της σωστής διαμόρφωσης του δικτύου είναι ότι ενώ η ταχύτητα των κόμβων αυξάνεται, ο αριθμός των μηνυμάτων που στέλνονται στις φάσεις διάδοσης αποτελέσματος είναι σταθερός,έτσι όλοι οι αλγόριθμοι χρησιμοποιούν τα βέλτιστα μονοπάτια για να καταλήξουν οι μετρήσεις στον gateway. Εδώ λοιπόν βλέπουμε καθαρά ότι αλγόριθμος STS-M λειτουργεί πολύ καλά όχι μόνο στο να περιορίσει τις αστοχίες σύνδεσης ,αλλά και στην σωστή δρομολόγηση των μετρήσεων.

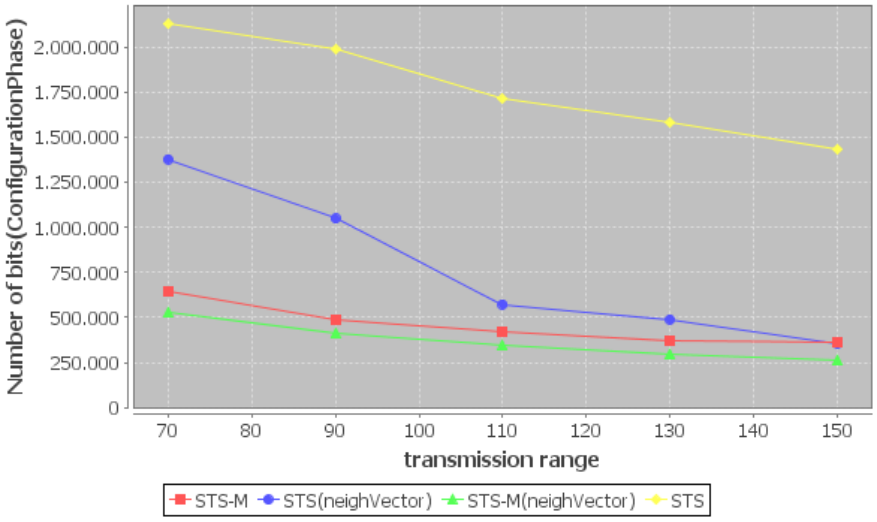
Μεταβάλλοντας το transmission range



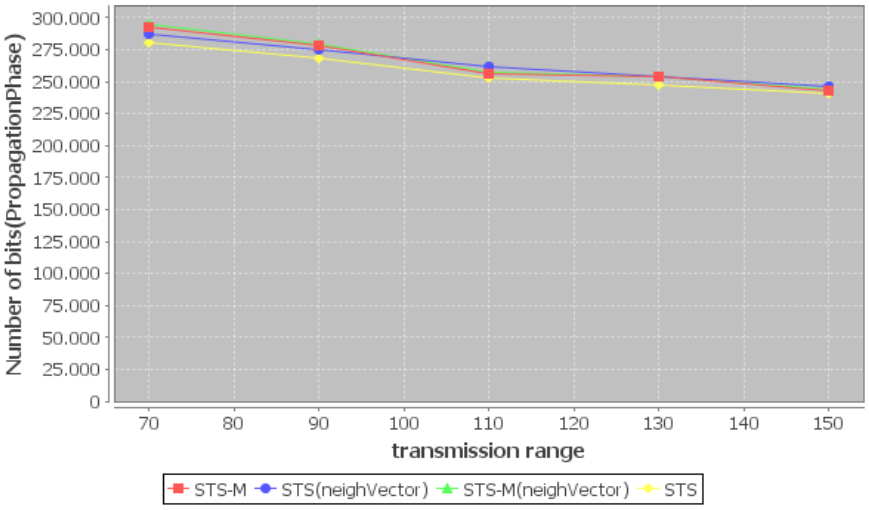
Γράφημα 15



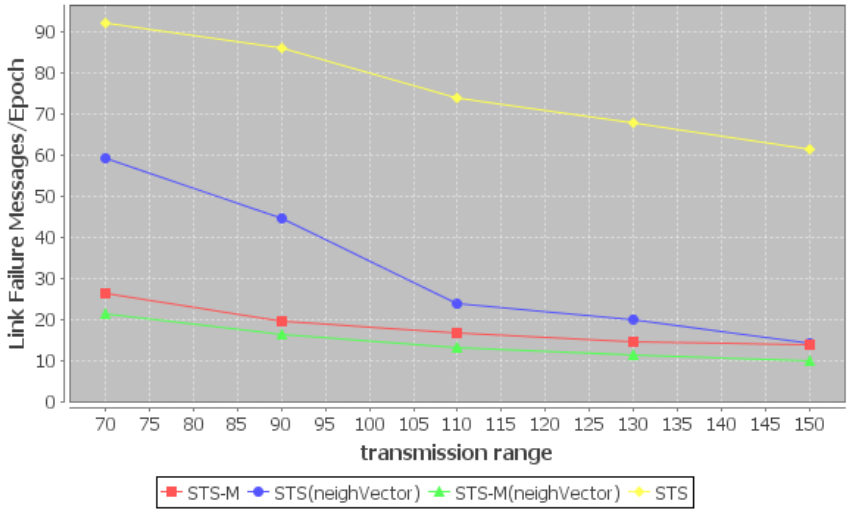
Γράφημα 16



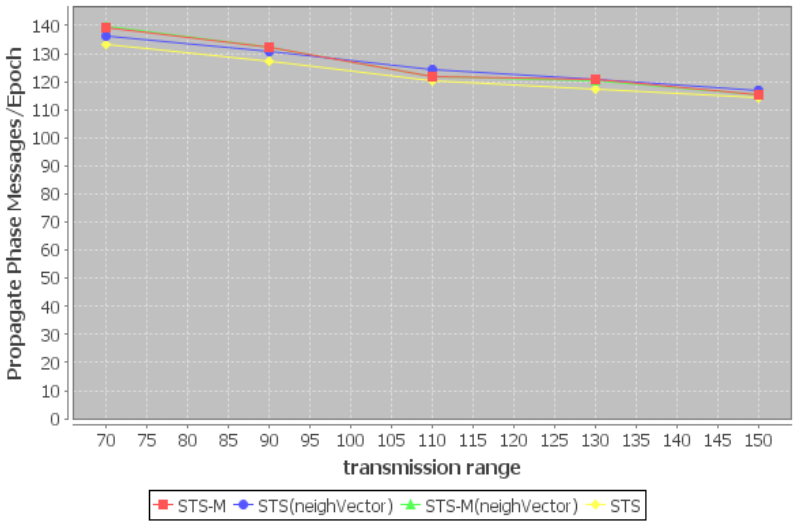
Γράφημα 17



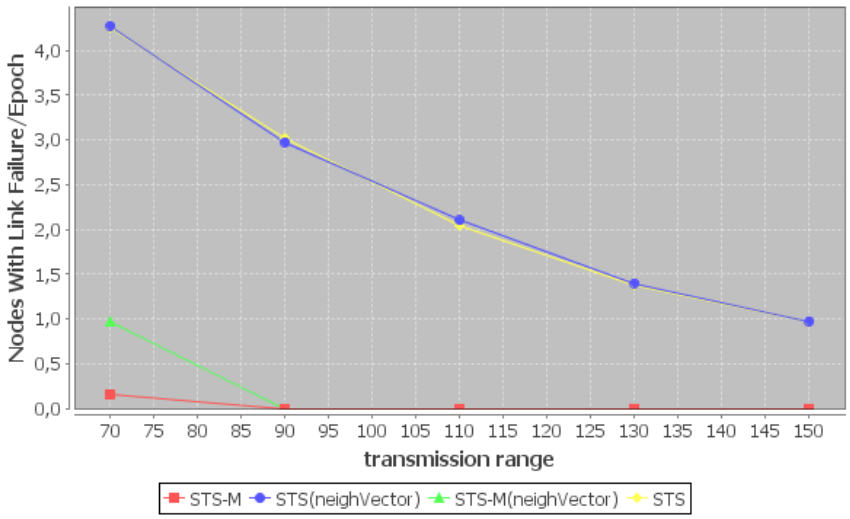
Γράφημα 18



Γράφημα 19



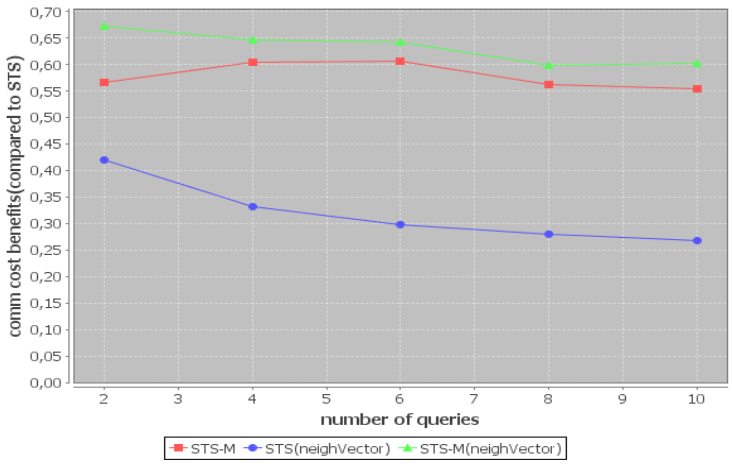
Γράφημα 20



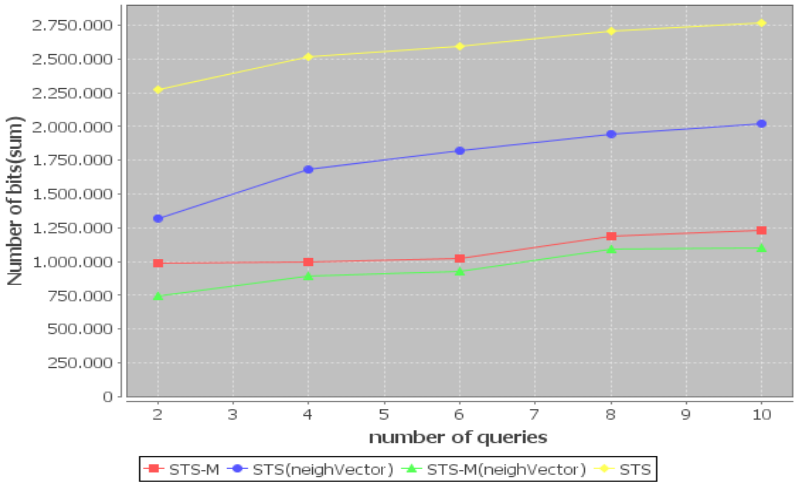
Γράφημα 21

Παρατηρήσεις:Παρατηρούμε ότι καθώς το transmission range αυξάνεται το όφελος των αλγορίθμων STS-M και STS-M(neighVector) παραμένει σταθερό και το όφελος STS(neighVector) αυξάνεται έναντι του STS(γράφημα 15).Αυτό συμβαίνει γιατί ο STS(neighVector) χρησιμοποιεί τον πίνακα neighbors για την αντιμετώπιση αστοχιών σύνδεσης,έτσι όταν αυξάνεται το transmission range αυξάνεται και ο αριθμός των γειτονικών κόμβων των οποίων τα στοιχεία έχει αποθηκευμένα.Σε περίπτωση αστοχίας σύνδεσης ,παρά το γεγονός ότι πολλοί γείτονες κόμβοι θα έχουν απομακρυνθεί ,θα έχει επιλογές στον πίνακα για να συνδεθεί.Επίσης ,λόγω της αύξησης του transmission range μειώνεται ο αριθμός των link failures (γράφημα 21) και μειώνεται ο αριθμός των μηνυμάτων που στέλνονται στις φάσεις διάδοσης αποτελέσματος .

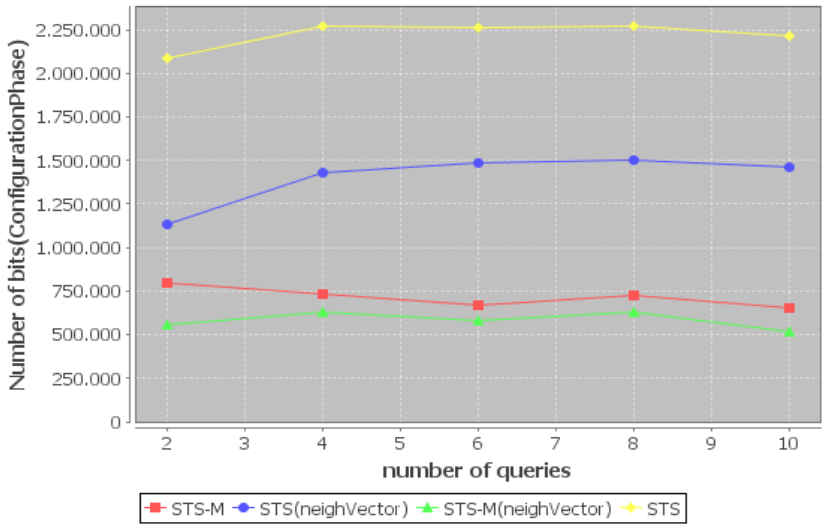
Μεταβάλλοντας τον αριθμό των επερωτήσεων



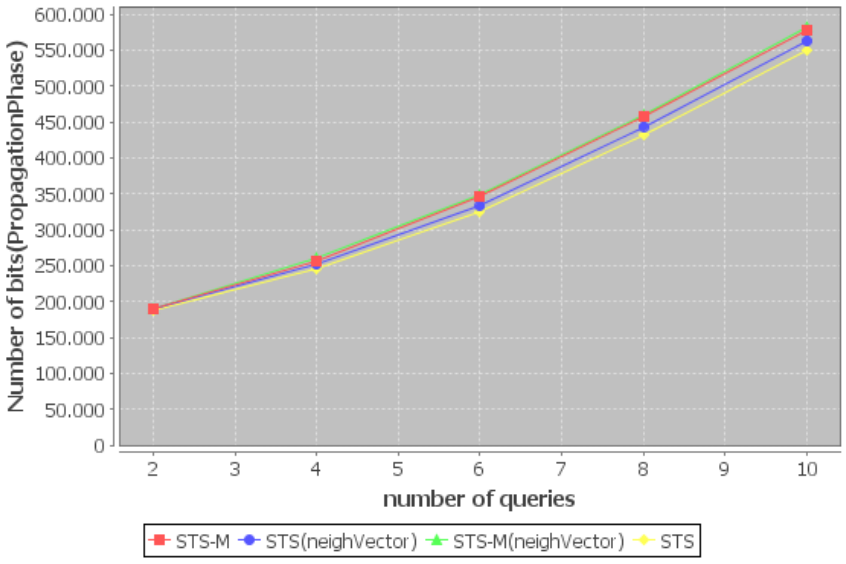
Γράφημα 22



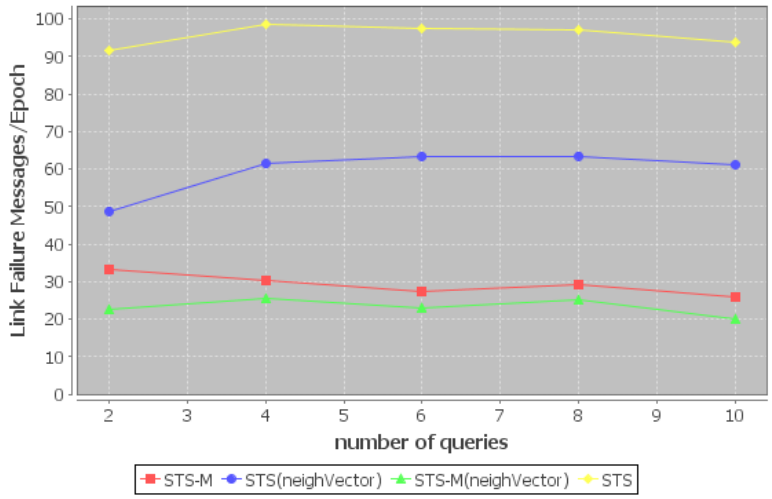
Γράφημα 23



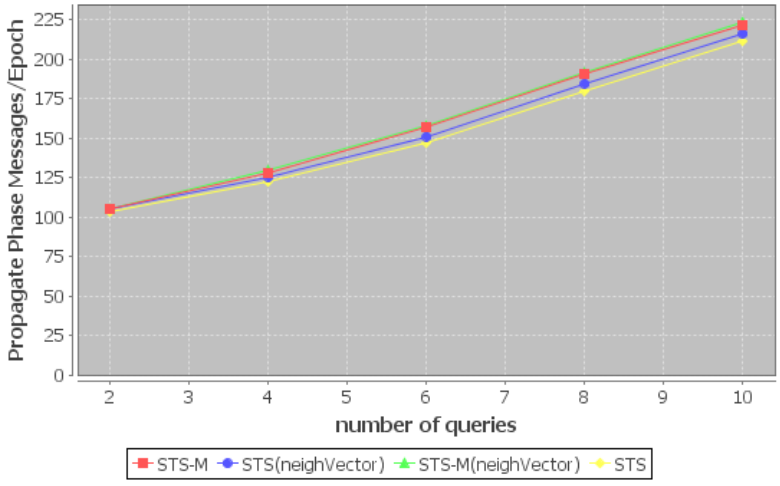
Γράφημα 24



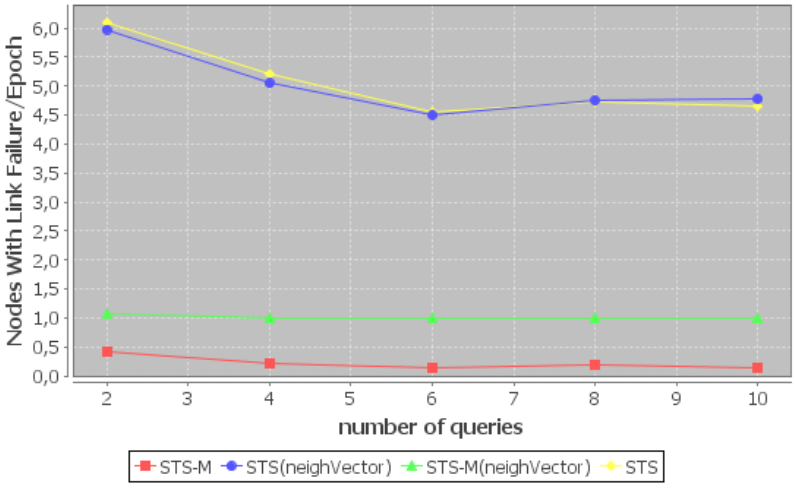
Γράφημα 25



Γράφημα 26



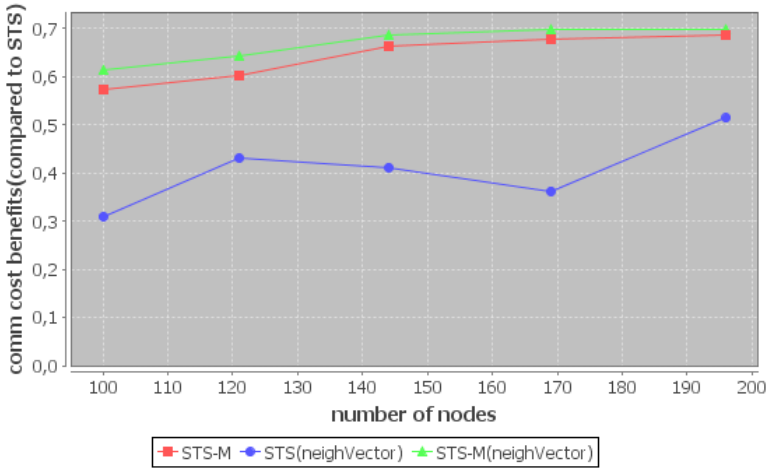
Γράφημα 27



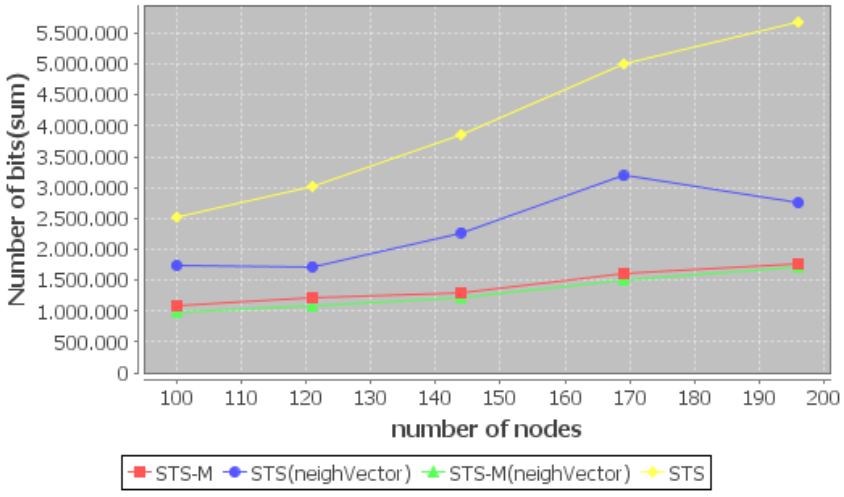
Γράφημα 28

Παρατηρήσεις: :Αρχικά αυτό που παρατηρούμε είναι η πολύ καλή απόδοση του αλγορίθμου STS-M σε σχέση με τον STS(γράφημα 22) που παραμένει σταθερή όσο αυξάνεται ο αριθμός των επερωτήσεων.Επίσης, παρατηρούμε ότι ο αριθμός των επερωτήσεων επηρεάζει τον αριθμό των μηνυμάτων που στέλνονται στην φάση διάδοσης αποτελέσματος(γράφημα 25 και 27) ,αλλά δεν επηρεάζει αισθητά τον αριθμό των μηνυμάτων για την διαμόρφωση του δικτύου,καθώς όπως βλέπουμε από το γράφημα 26, ο αριθμός των bits που στέλνονται για την αναδιαμόρφωση του παραμένει περίπου σταθερός.

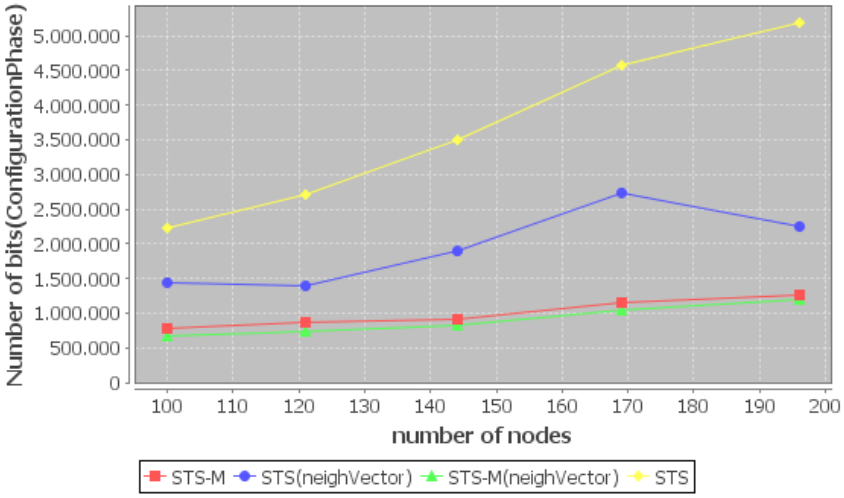
Μεταβάλλοντας τον αριθμό των κόμβων



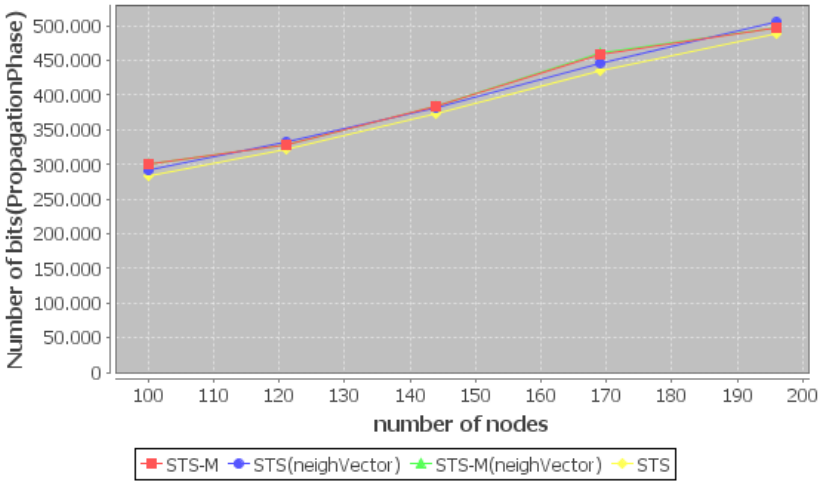
Γράφημα 29



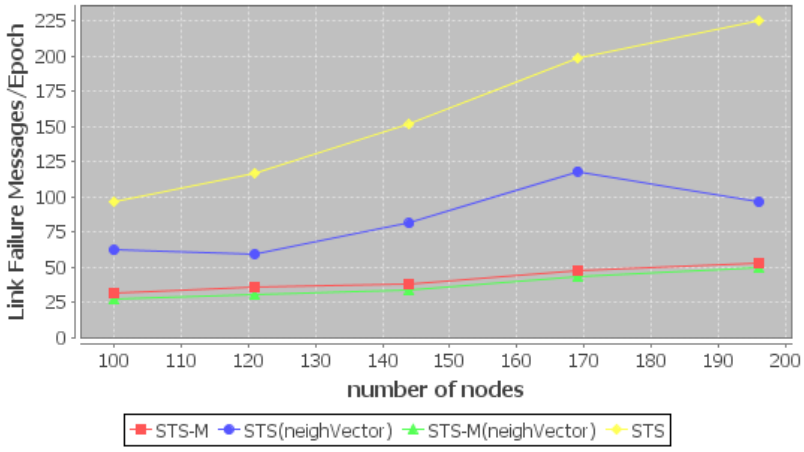
Γράφημα 30



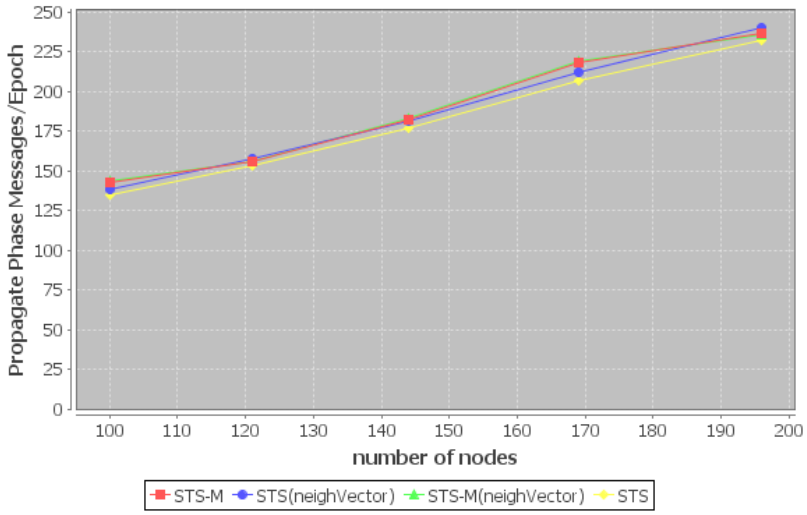
Γράφημα 31



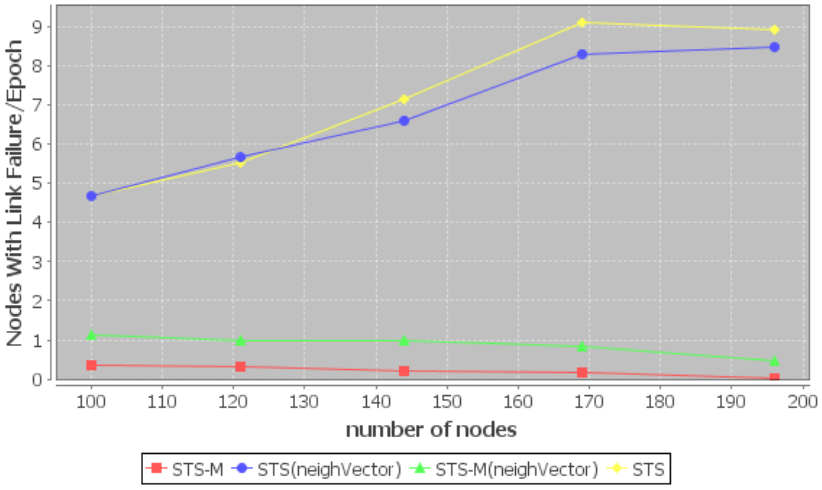
Γράφημα 32



Γράφημα 33



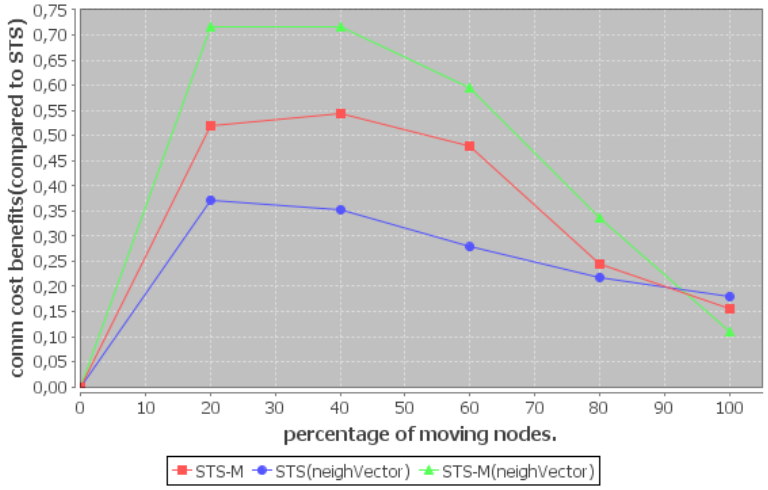
Γράφημα 34



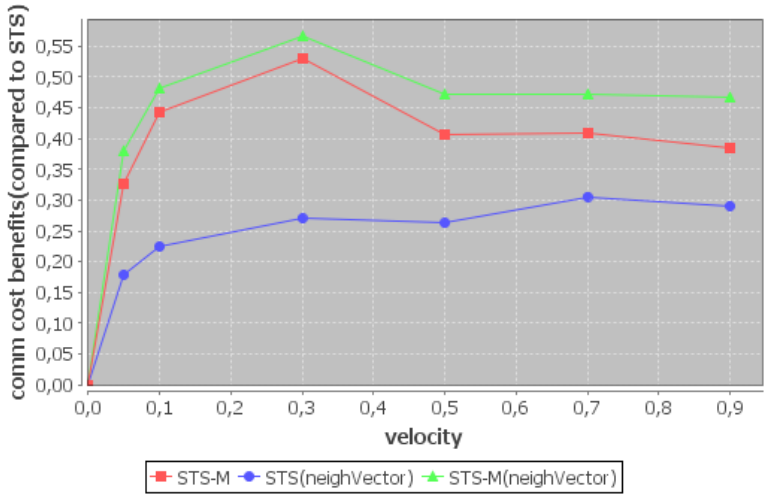
Γράφημα 35

Παρατηρήσεις:Παρατηρούμε ότι όσο αυξάνεται ο αριθμός των κόμβων οι αλγόριθμοι STS-M και STS-M(neighVector) παρουσιάζουν μεγαλύτερο όφελος έναντι του STS(γράφημα 29) και αυτό γιατί παραμένει σταθερός ο αριθμός των κόμβων που παρουσιάζουν αστοχία σύνδεσης ανά εποχή(υπάρχει και μικρή μείωση) ,σε αντίθεση με τον STS στον οποίον αυξάνεται ο αριθμός(γράφημα 35).Είναι σημαντικό να τονίσουμε ότι και στον αλγόριθμο STS(neighVector) αυξάνεται ο αριθμός των κόμβων με αστοχία σύνδεσης, αλλά δεν επηρεάζει την απόδοση του αφού χρησιμοποιεί τον πίνακα neighbors για την αντιμετώπιση τους.Έτσι ,επειδή αυξάνεται ο αριθμός των γειτονικών κόμβων των οποίων τα στοιχεία έχει αποθηκευμένα σε περίπτωση αστοχίας σύνδεσης ,παρά το γεγονός ότι πολλοί γείτονες κόμβοι θα έχουν απομακρυνθεί ,θα έχει επιλογές στον πίνακα για να συνδεθεί.

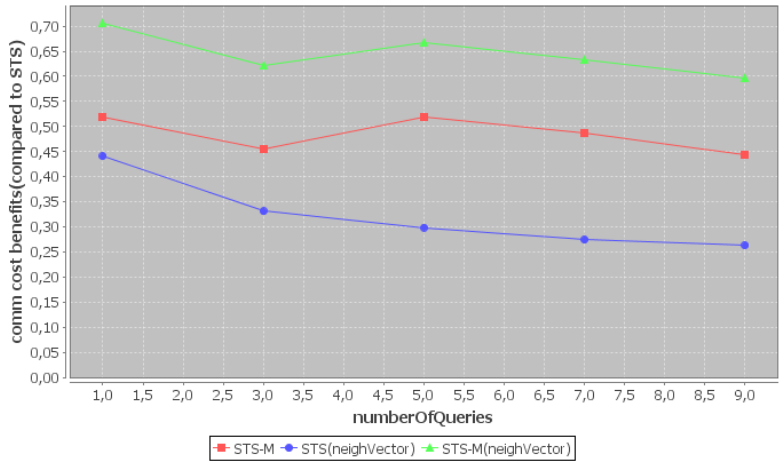
Προσθέτουμε και τις μετρήσεις που έχουμε πάρει χρησιμοποιώντας τον τύπο 1/(epochsConnected * hops_apo_gateway) για να δείξουμε ότι η απόδοση του επιλεγμένου τρόπου υπολογισμού του κόστους ήταν καλύτερη .Παρουσιάζουμε μόνο τα γραφήματα της απόδοσης.



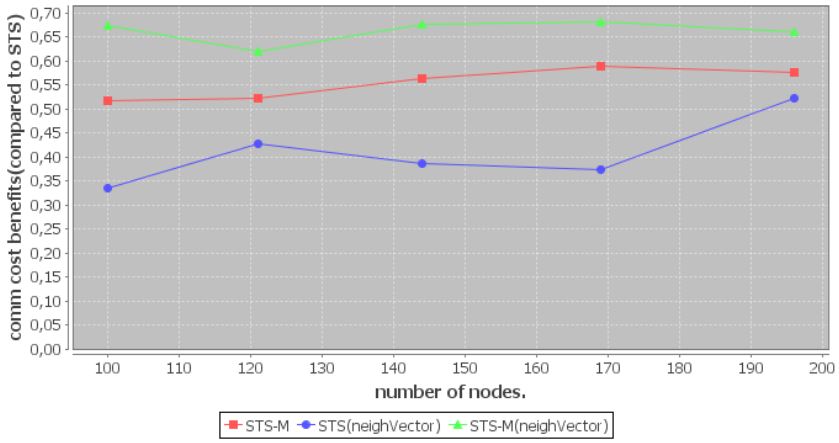
Γράφημα 36



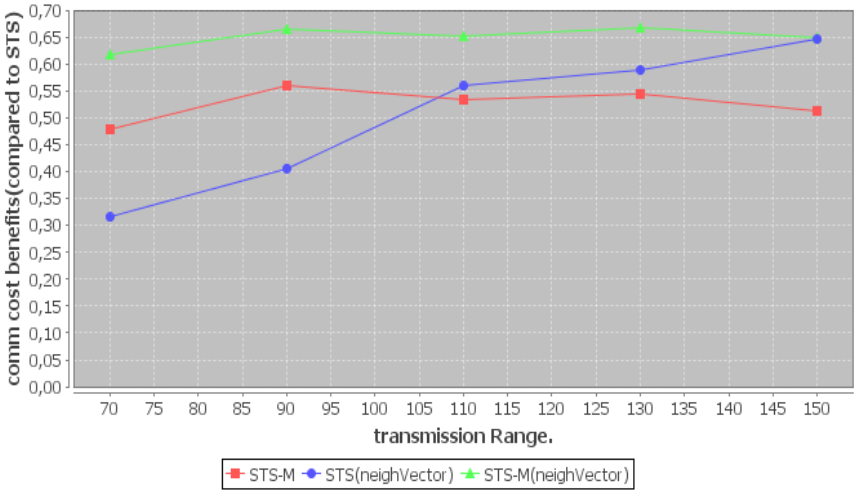
Γράφημα 37



Γράφημα 38



Γράφημα 39



Γράφημα 40

Συμπεράσματα

Η διπλωματική αυτή εργασία ασχολείται με την παράλληλη επεξεργασία πολλαπλών συναθροιστικών επερωτήσεων σε ένα δίκτυο αισθητήρων με κινούμενους κόμβους. Ο στόχος λοιπόν ήταν, τόσο να βρεθούν οι αποδοτικές διαδρομές που ελαχιστοποιούν το κόστος επικοινωνίας στην εκτέλεση πολλών συναθροιστικών επερωτήσεων, με την κατάλληλη επεξεργασία και την σωστή δρομολόγηση των μετρήσεων των κόμβων , όσο και να μειωθεί ο αριθμός των αστοχιών σύνδεσης μεταξύ των κόμβων (ink failures) που παρατηρούνται με την χρήση των ήδη υπάρχοντων αλγορίθμων.Για την υλοποίηση του στόχου μας δημιουργήσαμε έναν νέο αλγόριθμο , τον STS-M (Segment To Segmnet with Mobile Nodes) , ο οποίος βασίζεται σε μια νέα παράμετρο που ονομάζουμε Cost.Με βάση την νέα αυτή παράμετρο , ο αλγόριθμός μας διαμορφώνει κατάλληλα το δίκτυο αισθητήρων με αποτελέσμα να παρατηρούνται λιγότερες αστοχίες σύνδεσης ανάμεσα στους κόμβους , γίνεται βέλτιστη επεξεργασία των μετρήσεων και σωστή δρομολόγησή τους προς τον κόμβο gateway και το δίκτυο αναδιαμορφώνεται σε περιπτώσεις αστοχίας σύνδεσης με όσο το δυνατόν μικρότερο ενεργειακό κόστος.Η πολύ καλή απόδοση του αλγορίθμου μας φαίνεται απο τα πειραματικά αποτελέσματα,τα οποία αναδεικνύουν την καλή λειτουργικότητα του STS-M.

Βιβλιογραφία

[1] http://www.tinyos.net/

[2] http://www.sics.se/contiki/

[3] http://mantisos.org/

[4] http://www.btnode.ethz.ch/

[5] http://www.nanork.org/

[6] *http://en.wikipedia.org/wiki/Wireless_sensor_network*

[7] G. Simon, M. Maroti, A. Ledeczi, G. Balogh, B. Kusy, A. Nadas, G. Pap, J. Sallai, and K. Frampton. Sensor network-based countersniper system. In Second International Conference on Embedded Networked Sensor Systems (Sensys), 2004.

[8] P. Zhang, C.M. Sadler, S.A. Lyon, and M. Martonosi. Hardware design experiences in zebranet. In SenSys04, 2004.

[9] C. D. Kidd, R. Orr, G. D. Abowd, C. G. Atkeson, I. A. Essa, B. MacIntyre, E. D. Mynatt, T. Starner, and W. Newstetter. The aware home: A living laboratory for ubiquitous computing research. In CoBuild, pages 191-198, 1999.

[10]http://www.alertsystems.org

[11] B.G. Celler et al. An instrumentation system for the remote monitoring of changes in functional health status of the elderly. In International Conference IEEE-EMBS, 1994.

[12] G. Coyle et al. Home telecare for the elderly. Telemedicine and Telecare, 1:183-184, 1995.

[13]http://www.cs.colorado.edu/~mozer/index.php?dir=/Research/Projects/Adaptive%20house/

[14] J.H. Huang, S. Amjad, and S. Mishra. Cenwits: A sensor-based loosely coupled search and rescue system using witnesse. In Third International Conference on Embedded Networked Sensor Systems (Sensys), 2005.

[15] I. Vasilescu, K. Kotay, D. Rus, M. Dunbabin, and P. Corke. Data collection, storage, retrieval with an underwater sensor network. In Third International Conference on Embedded Networked Sensor Systems (Sensys), 2005.

[16]http:/courses.ced.tuc.gr/.ΗΜΜΥ (ΠΛΗ) 516
Επεξεργασία και Διαχείριση Δεδομένων σε Δίκτυα Αισθητήρων

[17] Holger Karl, Andreas Willig. “Protocols and Architectures For Wireless Sensor Networks” Ed. John Wiley & Sons Inc., pp. 1-88, 2005

[18] Parallax inc, “Boe-Bot Robot”, [Online]. Available: http://www.parallax.com/tabid/411/Default.aspx

[19] Karthik Dantu, Mohammad Rahimi, Hardik Shah, Sandeep Babel, Amit Dhariwal and Gaurav Sukhatme “Robomote: Enabling Mobility In Sensor Networks.”, 2004.

[20] Nuzhet Atay, Burchan Bayazit. “Mobile Wireless Sensor Network Connectivity Repair with K-Redundancy”, September 2007

[21] Sarah Bergbreiter, Jameson Lee, Dr. Kris Pister, “Cotsbots” [Online]. Available: http://www-bsac.eecs.berkeley.edu/projects/cotsbots/

[22] Luis E. Navvarro - Serment, Robert Grabowski, Christiaan J.J. Paredis, and Pradeep K. Khosla. “Millibots:The Development of a Frameworkand Algorithms for a Distributed Heterogeneous Robot Team”, December 2002.

[23] http://mechatronics.ece.usu.edu/mas-net/index_files/Pub_MASmote_A_Mobility_Node_for_MASnet.pdf

[24] Christopher M. Cianci, Xavier Raemy, Jim Pugh, and Alcherio MartinoliCommunication in a Swarm of Miniature. “Robots: The e-Puck as an Educational Tool for Swarm Robotics”, 2007.

[25] Samuel Madden , Michael J. Franklin , Joseph M. Hellerstein , Wei Hong, The design of an acquisitional query processor for sensor networks, Proceedings of the 2003 ACM SIGMOD international conference on Management of data, June 2003

[26] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, *TAG: A tiny aggregation service for ad-hoc sensor networks*, in Proceedings of the Symposium on Operating Systems Design and Implementation, OSDI, December 2002

[27] Niki Trigoni, A. G. (2008). Interplay of Processing and Routing in Aggregate Query Optimization for Sensor Networks. In *Distributed Computing and Networking* (pp. 401-415).