



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ
ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

Διπλωματική Εργασία

**Δίκτυα Έξυπνης Σκόνης (Smart Dust): Μελέτη και ανάλυση
αρχιτεκτονικών, πρωτοκόλλων και τεχνολογιών. Το πρόβλημα
της τοπικής ανίχνευσης και διάδοσης της πληροφορίας και το
πρωτόκολλο MP-DFR.**

Πευκιανάκης Μ. Ιωάννης Α.Μ 2713
pefkian@ceid.upatras.gr

Επιβλέπων: Νικολετσέας Σωτήρης
Λέκτορας Πανεπιστημίου Πατρών

ΠΑΤΡΑ, ΙΟΥΝΙΟΣ 2006

ΠΡΟΛΟΓΟΣ

Οι πρόσφατες εξελίξεις στις τεχνολογίες, των μικρό-ηλεκτρονικών-μηχανικών συστημάτων ή MEMS (micro-electro-mechanical systems), των ασύρματων τεχνολογιών και των ψηφιακών ηλεκτρονικών, έδωσε τη δυνατότητα ανάπτυξης χαμηλού κόστους, χαμηλής ενέργειας και μεγάλης λειτουργικότητας δικτυακών κόμβων αισθητήρων που είναι μικρού μεγέθους και μπορούν να επικοινωνούν σε μικρές αποστάσεις. Η κατασκευή λοιπόν αυτών των μικροσκοπικών κόμβων οι οποίοι διαθέτουν αισθητήρες, μικρό-επεξεργαστές και έχουν τη δυνατότητα επικοινωνίας, οδήγησε στη ιδέα ανάπτυξης των ασύρματων δικτύων αισθητήρων ή δικτύων έξυπνης σκόνης τα οποία βασίζονται στην ύπαρξη μεγάλου αριθμού τέτοιων κόμβων και στην δυνατότητα που έχουν να «συνεργάζονται». Τα δίκτυα έξυπνης σκόνης λοιπόν, αποτελούν μια νέα κατηγορία δικτύων που απευθύνονται σε μια πληθώρα εφαρμογών και έχουν γίνει αντικείμενο εντατικής έρευνας τα τελευταία έξι χρόνια.

Αυτή η διπλωματική εργασία αποτελεί μια πλήρης μελέτη των δικτύων έξυπνης σκόνης, τόσο αναφορικά με τις τεχνολογίες και τις εφαρμογές τους, όσο και από τη πλευρά της αρχιτεκτονικής αυτών των δικτύων και των πρωτοκόλλων αιχμής. Συγκεκριμένα η εργασία αυτή αποτελείται από έξι βασικά μέρη.

Μέρος 1^ο: Το 1^ο μέρος αποτελεί μια εισαγωγή στα δίκτυα έξυπνης σκόνης και στις εφαρμογές τους. Επίσης σε αυτό το μέρος περιγράφουμε το πώς αυτά τα δίκτυα διαφοροποιούνται από τα ad hoc κινητά δίκτυα.

Μέρος 2^ο: Το 2^ο μέρος αποτελεί μελέτη των αρχιτεκτονικών των δικτύων έξυπνης. Αρχικά αναλύουμε την αρχιτεκτονική των κόμβων, δηλαδή τα συστατικά τους στοιχεία, ενώ στη συνέχεια την αρχιτεκτονική σε επίπεδο δικτύου, αναφέροντας τα πιο βασικά δικτυακά μοντέλα. Το μοντέλο των πολλαπλών sinks που παρουσιάζεται σε αυτό το μέρος, είναι ένα δικτυακό μοντέλο που αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής εργασίας. Τέλος σε αυτό το μέρος αναφερόμαστε και σε τεχνολογίες κόμβων αισθητήρων αλλά και σε λειτουργικά συστήματα και περιβάλλοντα υλοποίησης για αυτά τα δίκτυα.

Μέρος 3^ο: Το 3^ο αποτελεί το εναρκτήριο μέρος για τη μελέτη και ανάλυση πρωτοκόλλων, και αναφέρεται στα πρωτόκολλα ανακάλυψης τοπολογίας και αυτό-διαμόρφωσης.

Μέρος 4^ο: Το 4^ο μέρος αποτελεί μελέτη των πρωτοκόλλων Localization ή καλύτερα πρωτοκόλλων ανακάλυψης της θέσης για κάθε κόμβο του δικτύου. Τα πρωτόκολλα αυτά σε συνδυασμό με τα πρωτόκολλα ανακάλυψης τοπολογίας και αυτό-διαμόρφωσης του προηγούμενου μέρους, είναι συνήθως απαραίτητα να τρέχουν ανεξάρτητα εφαρμογής, χαρακτηριστικό που τα κάνει πολύ σημαντικά.

Μέρος 5^ο: Το 5^ο μέρος αναφέρεται σε πρωτόκολλα δρομολόγησης της πληροφορίας. Περισσότερο βάρος δίνεται όμως σε ένα ειδικό πρόβλημα δρομολόγησης, στο πρόβλημα της «τοπικής ανίχνευσης και διάδοσης της πληροφορίας» (*Problem of Local Detection and Propagation*) το οποίο αναφέρεται στη τοπική ανίχνευση ενός σημαντικού γεγονότος και την αποτελεσματική τόσο χρονικά όσο και ενεργειακά μετάδοση του, σε ένα κέντρο ελέγχου. Για το πρόβλημα του *Local Detection and Propagation*, πέρα από το ότι αναλύουμε μια σειρά πρωτοκόλλων, παρουσιάζουμε το MP-DFR ένα πρωτόκολλο που αναπτύχθηκε στα πλαίσια αυτής της διπλωματικής εργασίας.

Μέρος 6^ο: Το 6^ο μέρος αποτελεί το πειραματικό μέρος. Πιο συγκεκριμένα εκεί παραθέτουμε τα συγκριτικά αποτελέσματα των πρωτοκόλλων που παρουσιάστηκαν στο 5^ο μέρος και βέβαια κάνουμε μια μικρή ανάλυση του περιβάλλοντος υλοποίησης, δηλαδή του simulator *SimDust*.

Τα θέματα που παρουσιάζονται σε αυτή τη διπλωματική εργασία αντιπροσωπεύουν ένα μεγάλο μέρος της υπάρχουσας βιβλιογραφίας και έτσι αποτελούν χρήσιμο οδηγό κατανόησης των δικτύων έξυπνης σκόνης ακόμα και για κάποιον που δεν είναι έμπειρος σε σχετικά θέματα. Ελπίζουμε λοιπόν ότι θα αποτελέσει ουσιαστική βοήθεια για κάποιον που σκέφτεται να ασχοληθεί με τα δίκτυα έξυπνης σκόνης στο μέλλον. Καλή ανάγνωση!

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΕΧΟΜΕΝΑ	5
Μέρος 1 ^ο : ΔΙΚΤΥΑ ΕΞΥΠΙΝΗΣ ΣΚΟΝΗΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ.....	10
1 Εισαγωγή	11
1.1 Εφαρμογές των WSNs.....	12
1.2 Τύποι Εφαρμογών των WSNs	14
1.3 Προκλήσεις για τα WSNs.....	16
1.3.1 Χαρακτηριστικές απαιτήσεις των WSNs	16
1.3.2 Απαιτούμενοι μηχανισμοί	17
1.4 Κινητά Ad Hoc Δίκτυα & WSNs	19
1.5 Τεχνολογίες για τα WSNs	22
Μέρος 2 ^ο : ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ.....	23
2 Αρχιτεκτονική Κόμβου	24
2.1 Hardware Συστατικά του Κόμβου.....	24
2.1.1 Σύνοψη υλικού	24
2.1.2 Ελεγκτής.....	26
2.1.3 Μνήμη	27
2.1.4 Συσκευή επικοινωνίας.....	27
2.1.5 Αισθητήρες.....	29
2.1.6 Τροφοδοσία ενέργειας για τους κόμβους αισθητήρων.	30
2.2 Κατανάλωση Ενέργειας Κόμβου.....	31
2.2.1 Επικοινωνία	32
2.2.2 Επεξεργασία δεδομένων.....	32
2.3 Λειτουργικά Συστήματα και Περιβάλλοντα για WSNs	33
2.3.1 Ενσωματωμένα λειτουργικά συστήματα.....	33
2.3.2 TinyOS και nesC	34
2.4 Παραδείγματα Κόμβων Αισθητήρων.....	39
2.4.1 Η οικογένεια “Mica Mote”	39
2.4.2 Κόμβοι EYES	41
2.4.3 BTnodes.....	41
2.4.4 Scatterweb	42

3	Αρχιτεκτονική Δικτύου	43
3.1	Διαφορετικά Σενάρια για Δίκτυα Αισθητήρων	43
3.1.1	Τύποι πηγών και sinks	43
3.1.2	Single-hop vs multihop δικτύων.....	44
3.1.3	Πολλαπλοί sinks και sources	45
3.1.4	Τρεις τύποι κινητικότητας	46
3.2	Παράγοντες που Επηρεάζουν το Σχεδιασμό ενός WSN	48
3.2.1	Τεχνολογικές προδιαγραφές.....	48
3.2.2	Κόστος παραγωγής.....	49
3.2.3	Μέσο μετάδοσης.....	50
3.2.4	Ενέργεια.....	50
3.2.5	Ανοχή στα σφάλματα	50
3.2.6	Τοπολογία δικτύου	51
3.2.7	Παράταξη κόμβων (deployment)	51
3.3	Μοντέλα Δικτύων.....	51
3.3.1	Μοντέλο πηγής-Sink	52
3.3.2	Μοντέλο πηγής-Τείχους	53
3.3.3	Μοντέλο <i>Lattice</i> κατανομής.....	54
3.3.4	Μοντέλο πολλαπλών Walls ή Sinks.....	55
3.3.5	Μοντέλο κατανάλωσης ενέργειας	58
Μέρος 3 ^ο : ΑΝΑΚΑΛΥΨΗ ΤΟΠΟΛΟΓΙΑΣ & ΑΥΤΟ-ΔΙΑΜΟΡΦΩΣΗ.....		60
4	Ανακάλυψη Τοπολογίας στα Δίκτυα Αισθητήρων	61
4.1	Διαχείριση WSNs	61
4.2	Topology Discovery	63
4.2.1	Προσεγγίσεις για ανακάλυψη τοπολογίας.....	63
4.3	Απόκριση Ομάδας και Αλγόριθμος TopDisc	66
4.3.1	Διάδοση της αίτησης χρησιμοποιώντας τρία χρώματα	67
4.3.2	Διάδοση της αίτησης χρησιμοποιώντας τέσσερα χρώματα.....	69
4.3.3	Μηχανισμός απόκρισης του TopDisc	71
4.3.4	Εφαρμογές του TopDisc.....	72
5	Αυτό-Διαμόρφωση και ASCENT.....	72
5.1	ASCENT.....	74
5.1.1	Neighbor discovery phase (Φάση ανακάλυψης γείτονα)	75

5.1.2	Join decision phase (Φάση απόφασης συμμετοχής).....	75
5.1.3	Active & adaptive phase (Ενεργή και προσαρμοστική φάση)	76
Μέρος 4 ^ο : ΠΡΩΤΟΚΟΛΛΑ LOCALIZATION		78
6	Εισαγωγή στο Localization	79
7	Localization Hardware	80
7.1	Beacon Κόμβοι (Κόμβοι Άγκυρας).....	80
7.2	Ένδειξη Ισχύος Λαμβανόμενου Σήματος (RSSI).....	81
7.3	Υπολογισμός Απόστασης με την Μέτρηση των Hops	81
7.4	Διαφορά Χρόνου κατά την Άφιξη (TDoA).....	82
7.5	Γωνία Άφιξης (AoA).....	83
8	Θέματα στο Σχεδιασμό Localization Αλγορίθμων.....	84
8.1	Περιορισμοί Πόρων.....	84
8.2	Πυκνότητα (Density) Κόμβων	84
8.3	Μη Κυρτές (Non Convex) Τοπολογίες	85
8.4	Φυσικά Εμπόδια και Ανωμαλίες του Εδάφους	85
8.5	Ταξινόμηση Αλγορίθμων Localization	85
9	Κεντριοποιημένοι Αλγορίθμοι	87
9.1	MDS-MAP	87
10	Beacon-Based Κατανεμημένοι Αλγόριθμοι.....	88
10.1	Diffusion.....	88
10.2	Bounding Box.....	89
10.3	Gradient.....	90
10.4	APIT	91
10.5	Localization με ένα Mobile Beacon	93
10.5.1	Ιδέα του mobile beacon	93
10.5.2	Αλγόριθμος του mobile beacon.....	94
10.5.3	Τροχιά του mobile beacon.....	96
10.5.4	Πειραματικά αποτελέσματα του αλγορίθμου.....	98
11	Relaxation-Based Κατανεμημένοι Αλγόριθμοι.....	99
12	Coordinate System Stitching	100
13	Παρατηρήσεις Κριτική	101
14	Μελλοντικές Κατευθύνσεις.....	102
Μέρος 5 ^ο : ΠΡΩΤΟΚΟΛΛΑ ΔΡΟΜΟΛΟΓΗΣΗΣ		103

15	Πρωτόκολλα Δρομολόγησης στα WSNs	104
15.1	Κατηγοριοποίηση Routing Πρωτοκόλλων	105
15.2	Πρόβλημα Τοπικής Ανίχνευσης και Διάδοσης	107
16	Πρωτόκολλο LTP	108
16.1	Το Πρόβλημα.....	108
16.2	Local Target Protocol	109
17	Πρωτόκολλο PFR	112
17.1	Το Πρόβλημα.....	113
17.2	Λειτουργία του Πρωτοκόλλου	114
17.2.1	Το υπό-πρωτόκολλο «φ-υπολογισμού».....	116
18	Πρωτόκολλο MP - DFR	117
18.1	Χαρακτηριστικά Παραδείγματα Εφαρμογής	117
18.2	Service Differentiation	119
18.3	Σχετική Έρευνα	120
18.4	Ορισμός Προβλήματος.....	122
18.5	Multipath Diffusion Forwarding Protocol (MP-DFR)	123
18.5.1	Πιθανότητα αποτυχίας διάδοσης της πληροφορίας.....	130
18.6	Ιδιότητες του MP-DFR.....	133
18.6.1	Ορθότητα του MP-DFR	133
18.6.2	Αποδοτικότητα ενέργειας του MP-DFR	134
Μέρος 6 ^ο : ΕΞΟΜΟΙΩΣΗ & ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ		137
19	Εξομοιοτής SimDust	138
19.1	Πλατφόρμα Υλοποίησης	138
19.2	Οργάνωση Κώδικα στον SimDust.....	139
19.3	Σημαντικές Κλάσεις για τη Λειτουργία του SimDust.....	140
19.3.1	Η κλάση Execution.....	140
19.3.2	Η κλάση Particle.....	141
19.3.3	Η κλάση Network	141
19.3.4	Η κλάση Info	142
19.3.5	Η κλάση Message	142
19.3.6	Η κλάση Event.....	142
19.3.7	Η κλάση Cache_info_Obj.....	142
19.3.8	Η κλάση Experiment	143

19.3.9	Η κλάση Animation.....	143
19.3.10	Τα αρχεία misc.h και misc.cpp.....	143
19.4	Υλοποίηση του Multi-Sink Μοντέλου και του MP-DFR	143
19.4.1	Υλοποίηση multi-sink μοντέλου	143
19.4.2	Υλοποίηση του MP-DFR	144
19.5	Χρήση του Simulator.....	148
20	Πειραματικά Αποτελέσματα	149
20.1	MP-DFR vs MP-DFRa	150
20.2	MP-DFR vs LTP vs PFR σε Ιδανικό Δίκτυο.....	151
20.3	Fault Tolerance για τα MP-DFR, LTP και PFR	152
20.4	Δημιουργία Μονοπατιών στο MP-DFR	153
20.5	Φόρτος Δικτύου στα MP-DFR, LTP και PFR.....	154
	Αγγλικοί Όροι.....	160

ΜΕΡΟΣ 1^ο : ΔΙΚΤΥΑ ΕΞΥΠΝΗΣ ΣΚΟΝΗΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ

1 ΕΙΣΑΓΩΓΗ

Ο πιο συνηθισμένος τύπος επεξεργασίας της πληροφορίας αναφέρεται σε μεγάλες, γενικού σκοπού υπολογιστικές μηχανές που ποικίλουν από τους κλασσικούς ισχυρούς κεντρικούς υπολογιστές, μέχρι και τους σύγχρονους φορητούς υπολογιστές. Σε πάρα πολλές εφαρμογές, όπως οι εφαρμογές γραφείου, αυτές οι υπολογιστικές μηχανές χρησιμοποιούνται κυρίως για την επεξεργασία πληροφορίας που εστιάζεται στον άνθρωπο-χρήστη και σχετίζεται μόνο έμμεσα με το φυσικό περιβάλλον. Υπάρχουν όμως κάποιες άλλες κατηγορίες εφαρμογών όπου το φυσικό περιβάλλον είναι το επίκεντρο της προσοχής και που οι διάφοροι υπολογισμοί αναφέρονται στην άσκηση ελέγχου σε κάποιες φυσικές διαδικασίες. Ένα τέτοιο παράδειγμα είναι ο έλεγχος των χημικών διεργασιών σε ένα εργοστάσιο έτσι ώστε να διατηρείται η πίεση και η θερμοκρασία. Σε αυτό το παράδειγμα η έννοια του «υπολογισμού» είναι συνυφασμένη με αυτήν του «ελέγχου» και είναι ενσωματωμένη «*embedded*» στο φυσικό σύστημα. Αντίθετα από τη προηγούμενη κλάση συστημάτων, τέτοια ενσωματωμένα συστήματα «*embedded systems*» δεν λειτουργούν αλληλεπιδρώντας με τον άνθρωπο χρήστη, αλλά είναι επιφορτισμένα στο να διεκπεραιώσουν κάποια εργασία ελέγχου ως κομμάτια ενός μεγαλύτερου συστήματος.

Η ιδέα των *ενσωματωμένων συστημάτων* είναι πολύ γνωστή και εφαρμόζεται ευρέως στις επιστήμες εφαρμοσμένης μηχανικής. Τα ενσωματωμένα συστήματα χρησιμοποιούνται όμως πάρα πολύ και στη καθημερινή μας ζωή. Λίγα είναι τα νοικοκυριά στα οποία ο ενσωματωμένος υπολογισμός δεν είναι παρόν για να ελέγχει τη λειτουργία του πλυντηρίου, του video ή του κινητού τηλεφώνου. Σε αυτές τις εφαρμογές τα ενσωματωμένα συστήματα «συναντούν» τα συστήματα που βασίζονται στην ανθρώπινη αλληλεπίδραση.

Τα νέα τεχνολογικά επιτεύγματα όμως αξιοποιούν την ιδέα των ενσωματωμένων συστημάτων ακόμα παρά πέρα. Αυτά μπορούν όχι μόνο να αποτελούν εξοπλισμό μεγάλων αντικειμένων όπως το πλυντήριο, αλλά να ελέγχουν και να παρακολουθούν πιο πολύπλοκες λειτουργίες της κατοικίας και του εργασιακού χώρου. Οι ιδέες αυτού του τύπου του υπολογισμού αποτελούν όλο και μεγαλύτερο κομμάτι της καθημερινότητας και συνιστούν την «*Περιβαλλοντολογική Νοημοσύνη (Ambient Intelligence)*» όπου πολλές διαφορετικές συσκευές συγκεντρώνουν και επεξεργάζονται πληροφορίες από πολλές διαφορετικές πηγές που έχουν να κάνουν και με τον έλεγχο φυσικών διεργασιών και μπορούν να αλληλεπιδράσουν άμεσα με το χρήστη.

Για να γίνει απόλυτα κατανοητή η ιδέα της «*Περιβαλλοντολογικής Νοημοσύνης (Ambient Intelligence)*» εκτός από την έννοια του ελέγχου και του υπολογισμού, βασική έννοια είναι και η *επικοινωνία*. Όλες αυτές οι πηγές πληροφορίας πρέπει να είναι σε θέση να μεταφέρουν αυτήν εκεί που είναι απαραίτητη (σε κάποιο χρήστη ή κέντρο ελέγχου) και να συνεργάζονται έτσι ώστε να παρέχουν μια ακριβή εικόνα του πραγματικού κόσμου όπως απαιτείται. Η επικοινωνία αυτή για πολλές εφαρμογές είναι αδύνατη με τη χρήση καλωδίου. Έτσι και για τα ενσωματωμένα συστήματα εισάγεται η ιδέα της ασύρματης επικοινωνίας.

Μια καινούργια κλάση δικτύων που μπορεί να αλληλεπιδρά με το φυσικό περιβάλλον έχει εμφανιστεί τα τελευταία χρόνια. Αυτά είναι τα ασύρματα δίκτυα

αισθητήρων ή Wireless Sensor Networks (WSN). Αυτά τα δίκτυα αποτελούνται από κόμβους που είναι σε θέση να αλληλεπιδρούν με το περιβάλλον μέσω αισθητήρων και να επικοινωνούν ασύρματα μεταξύ τους, φέροντας εις πέρας εργασίες τις οποίες δε θα μπορούσε να διεκπεραιώσει μόνος του ένας κόμβος. Οι κόμβοι αυτοί γενικά διαθέτουν τουλάχιστον κάποιες δυνατότητες υπολογισμού, ασύρματης επικοινωνίας και κάποιους αισθητήρες για να αλληλεπιδρούν με το περιβάλλον. Τα WSN's είναι ιδιαίτερα ισχυρά διότι μπορούν να εφαρμοστούν σε πολλές πραγματικού κόσμου εφαρμογές. Εξαιτίας της ευελιξίας και της ποικιλίας των εφαρμογών που αυτά τα δίκτυα αναφέρονται, δεν υπάρχει κάποιο σύνολο απαιτήσεων και χαρακτηριστικών βάση των οποίων αυτά μπορούν να ταξινομηθούν. Για παράδειγμα σε κάποιες WSN εφαρμογές οι κόμβοι του δικτύου δεν μπορούν να συνδεθούν σε μια σταθερή πηγή ενέργειας και έτσι βασίζονται στη χρήση μπαταριών. Σε τέτοιες εφαρμογές η διατήρηση ενέργειας είναι πολύτιμη. Σε άλλες όμως εφαρμογές η τροφοδοσία ενέργειας μπορεί να μην είναι τόσο σημαντική και άλλες μετρικές όπως η ακρίβεια των αποτελεσμάτων να έχει τη μεγαλύτερη σημασία. Συμπερασματικά λοιπόν, το κόστος, το μέγεθος και η αρχιτεκτονική κάθε κόμβου είναι άμεσα συνυφασμένη με την εφαρμογή. Γενικότερα στενά συσχετισμένη είναι η χωρητικότητα της μπαταρίας με το μέγεθος του κόμβου, ενώ η τιμή του κάθε κόμβου εξαρτάται από τη ποιότητα των αισθητήρων. Όμως αν είναι διαθέσιμος ένα μεγάλος αριθμός κόμβων μπορεί το δίκτυο με μικρή κατανάλωση ενέργειας και καλή ποιότητα να παρακολουθήσει κάποιο φαινόμενο χωρίς απαραίτητα να αποτελείται από πολύ ακριβούς κόμβους.

Για να γίνει πιο κατανοητή η χρηστικότητα των ασύρματων δικτύων αισθητήρων θα παρουσιάσουμε στη παρακάτω παράγραφο κάποιες βασικές εφαρμογές τους.

1.1 ΕΦΑΡΜΟΓΕΣ ΤΩΝ WSNs

Ο ισχυρισμός των υποστηρικτών των WSNs είναι ότι ο καινούργιος αυτός τύπος δικτύων θα εξυπηρετήσει πολλές από τις υπάρχουσες περιοχές εφαρμογών αλλά επίσης θα δημιουργήσει ακόμα περισσότερες. Επειδή οι κόμβοι των δικτύων αυτών διαθέτουν αισθητήρες σε συνδυασμό με δυνατότητες υπολογισμού και ασύρματης επικοινωνίας, μπορούν να κατασκευαστούν διαφορετικού τύπου εφαρμογές με πολλούς διαφορετικούς τύπους κόμβων, όπου ακόμα διαφορετικών δυνατοτήτων κόμβοι θα χρησιμοποιούνται στην ίδια εφαρμογή. Κάποιες από τη μεγάλη ποικιλία εφαρμογών είναι οι εξής:

1. **Εφαρμογές Πρόληψης Καταστροφής:** Ένα τυπικό σενάριο αυτής της κατηγορίας των εφαρμογών είναι η ανίχνευση πυρκαγιάς. Οι κόμβοι αισθητήρων σε αυτή τη περίπτωση μπορεί να είναι εξοπλισμένοι με θερμομέτρα και επίσης μπορεί να είναι σε θέση να υπολογίσουν τη τοποθεσία τους, τρέχοντας κάποιο αλγόριθμο «εντοπισμού» (localization). Τους κόμβους αυτούς στη συνέχεια μπορούμε να τους πετάξουμε από ένα αεροπλάνο και να απλωθούν σε ένα δάσος. Ομαδικά αυτοί οι κόμβοι μπορούν να σχηματίσουν ένα «χάρτη θερμοκρασίας» και να ενημερώνουν τους πυροσβέστες οι οποίοι πιθανόν να διαθέτουν κάποιο PDA, για τυχόν πολύ υψηλές θερμοκρασίες που υπονοούν πυρκαγιά.

2. **Εφαρμογές Παρακολούθησης του Περιβάλλοντος:** Τα WSNs μπορούν να χρησιμοποιηθούν για αν ελέγχουν το περιβάλλον όπως για παράδειγμα αναφορικά με τους χημικούς ρύπους σε ένα χώρο ταφής σκουπιδιών.
3. **Ευφυή Κτίρια:** Σε ένα κτίριο συχνά καταναλώνονται μεγάλα ποσά ενέργειας εξαιτίας της λανθασμένης χρήσης των συσκευών Air Conditioning (HVAC). Μία αποδοτικότερη, ακριβέστερη πραγματικού χρόνου παρακολούθηση της θερμοκρασίας, της ροής του αέρα, της υγρασίας καθώς και άλλων φυσικών παραμέτρων σε ένα κτίριο με τη βοήθεια των WSNs μπορεί να βοηθήσει σε μεγάλο βαθμό την εξοικονόμηση ενέργειας. Επίσης ένα άλλο σενάριο είναι η παρακολούθηση των επιπέδων της μηχανικής πίεσης των κτιρίων σε σεισμικά ενεργές ζώνες. Ανεξάρτητα της εφαρμογής, ο βασικός στόχος των WSNs θα είναι η παρακολούθηση των φυσικών παραμέτρων. Οι κόμβοι σε αυτές τις εφαρμογές μπορούν να τοποθετηθούν σε υπάρχοντα κτίρια ή να ενσωματωθούν σε κτίρια υπό κατασκευή.
4. **Facility Management:** Αυτή η περίπτωση αναφέρεται σε εφαρμογές που δεν περιλαμβάνουν την παρακολούθηση ενός μοναδικού κτιρίου, αλλά μιας μεγαλύτερης περιοχής. Παράδειγμα τέτοιων εφαρμογών είναι η είσοδος σε κάποιο κτίριο χωρίς τη χρήση κλειδιού όπου οι άνθρωποι θα φοράνε κάποιο σήμα-πομπό που θα δίνει τη δυνατότητα σε ένα WSN να ελέγχει αν ο κάτοχος του έχει το δικαίωμα πρόσβασης σε κάποιο χώρο του κτιρίου. Αυτή η εφαρμογή θα είναι πολύ χρήσιμη στη αποτροπή εισβολών από το να εισέλθουν σε κάποιο χώρο.
5. **Εφαρμογές στη Γεωργία:** Η εφαρμογή των WSNs στη γεωργία μπορεί να επιτρέψει την ακριβή άρδευση και λίπανση με την τοποθέτηση στις καλλιεργήσιμες εκτάσεις κόμβων που είναι σε θέση να ανιχνεύσουν υγρασία και σύσταση του εδάφους. Σε τέτοιες περιπτώσεις ένας μικρός αριθμός κόμβων μπορεί να θεωρηθεί επαρκής όπως ένας κόμβος ανά $100m \times 100m$ έκτασης.
6. **Ιατρική και Υγειονομική Περίθαλψη:** Η χρήση των WSNs σε εφαρμογές ιατρικής περίθαλψης είναι σίγουρα πολύ ωφέλιμη αλλά και ηθικά αμφισβητούμενη. Το εύρος της χρήσης των WSNs κυμαίνεται από εφαρμογές όπου οι κόμβοι αισθητήρων είναι άμεσα εφαρμοσμένοι στον ασθενή για να παρακολουθούν την υγεία του (εδώ η ασύρματη επικοινωνία των κόμβων είναι ιδιαίτερα σημαντική) και συνεισφέρουν ακόμα και στη χορήγηση της φαρμακευτικής αγωγής του, μέχρι και για εφαρμογές όπου οι κόμβοι συνιστούν ένα σύστημα παρακολούθησης (tracking system) των γιατρών και ασθενών στο νοσοκομείο.
7. **Logistics:** Σε πολλές Logistics εφαρμογές είναι απαραίτητο να εξοπλίσουμε τα διάφορα αντικείμενα (ένα δέμα για παράδειγμα) με απλούς αισθητήρες που θα είναι σε θέση να ανιχνεύουν τη θέση αυτών των αντικειμένων κατά τη διάρκεια κάποιας μεταφοράς. Σε αυτές τις εφαρμογές συχνά δε χρειάζεται κάποιος κόμβος αισθητήρων που να επικοινωνεί ενεργά. Μία παθητική ανάγνωση δεδομένων είναι συχνά επαρκής όπως για παράδειγμα στη περίπτωση που μια βαλίτσα κινείται στη ζώνη μεταφορών σε ένα αεροδρόμιο και περνάει από συγκεκριμένα σημεία ελέγχου. Μια τέτοια παθητική ανάγνωση είναι πολύ πιο απλή και φθηνή από μία ενεργή επικοινωνία (ανταλλαγή μηνυμάτων) που μπορεί να απαιτούν οι εφαρμογές που συζητήσαμε παραπάνω.

8. **Στρατιωτικές εφαρμογές:** Τα WSNs μπορεί να είναι ένα αναπόσπαστο κομμάτι των στρατιωτικών συστημάτων. Η πολύ γρήγορη τοποθέτηση των κόμβων στη περιοχή ενδιαφέροντος (π.χ ρίψη από αεροπλάνο), η δυνατότητα αυτό-οργάνωσης και η ανοχή σε λάθη (fault tolerance) είναι χαρακτηριστικά που κάνουν αυτού του τύπου τα δίκτυα εξαιρετικά σε τέτοιου τύπου εφαρμογές. Από τη στιγμή που τα δίκτυα έξυπνης σκόνης βασίζονται στο πολύ πυκνό άπλωμα πολύ χαμηλού κόστους κόμβων αισθητήρων στη περιοχή ενδιαφέροντος, ακόμα και η καταστροφή μερικών κόμβων από τα εχθρικά στρατεύματα δεν θα επηρεάσει σημαντικά τη συνολική απόδοση του δικτύου. Κάποιες στρατιωτικές εφαρμογές είναι η παρακολούθηση του συμμαχικού εξοπλισμού και των πολεμοφοδίων, η παρακολούθηση των αντίπαλων δυνάμεων στην εχθρική περιοχή, η αξιολόγηση των καταστροφών κατά τη διάρκεια της μάχης, η ανίχνευση χημικής, βιολογικής και πυρηνικής επίθεσης.

Αν και κάποιες από τις παραπάνω εφαρμογές θα μπορούσαν να υλοποιηθούν χωρίς την βοήθεια ασύρματων δικτύων αισθητήρων τέτοιες λύσεις αισθητήρων θα «λιμοκτονούσαν» όπως χαρακτηριστικά αναφέρεται στο [1]. Τα WSNs μπορούν σε μεγάλο βαθμό να παρέχουν τις απαραίτητες πληροφορίες με την απαιτούμενη ακρίβεια, έγκαιρα και με την εξοικονόμηση σημαντικού ποσού ενέργειας.

1.2 ΤΥΠΟΙ ΕΦΑΡΜΟΓΩΝ ΤΩΝ WSNs

Ο περισσότερες από τις εφαρμογές που περιγράψαμε παραπάνω μοιράζονται κάποια κοινά χαρακτηριστικά. Στις περισσότερες από αυτές υπάρχει μια σαφής διαφορά μεταξύ των πηγών των δεδομένων ή *sources* (των κόμβων δηλαδή που ανιχνεύουν τα γεγονότα) και των κόμβων «βυθών» ή *sinks* (κόμβοι στους οποίους τα γεγονότα παραδίδονται). Οι *sinks* κάποιες φορές είναι κομμάτι του δικτύου και κάποιες άλλες είναι συστήματα εκτός δικτύου (π.χ PDAs που επικοινωνούν με το WSN). Επίσης συνήθως ο αριθμός των *sources* είναι μεγαλύτερος από τους *sinks* και ο *sink* επίσης αγνοεί ή δεν τον ενδιαφέρει η ταυτότητα των *source* κόμβων. Τα δεδομένα είναι αυτά που έχουν πολύ μεγάλη σημασία σε ένα ασύρματο δίκτυο αισθητήρων.

Κάποια μοντέλα αλληλεπίδρασης μεταξύ των κόμβων *source* και *sinks* είναι τα εξής:

1. **Ανίχνευση Γεγονότων:** Οι κόμβοι αισθητήρων σε αυτή τη περίπτωση πρέπει να αναφέρουν στον/στους *sink(s)* την ανίχνευση ενός συγκεκριμένου γεγονότος. Τα πιο απλά γεγονότα μπορούν να ανιχνευθούν τοπικά από ένα μοναδικό κόμβο (π.χ όταν κάποιο όριο θερμοκρασίας ξεπεραστεί), ενώ τα πιο σύνθετα απαιτούν την συνεργασία περισσότερων γειτονικών ή ακόμα και πιο απομονωμένων κόμβων έτσι ώστε να αποφασιστεί αν κάποιο γεγονός έχει πραγματικά συνέβη. Εάν πάρα πολλά γεγονότα συμβούν, η ταξινόμηση τους μπορεί να είναι ένα θέμα που θα πρέπει να λυθεί.

2. **Περιοδικές Μετρήσεις:** Σε αυτή τη περίπτωση οι κόμβοι πρέπει περιοδικά να αναφέρουν κάποιες μετρήσεις που παίρνουν. Η συχνότητα αυτών των αναφορών εξαρτάται από την εφαρμογή.
3. **Προσέγγιση Συναρτήσεων:** Ο τρόπος με τον οποίο μία φυσική τιμή όπως η θερμοκρασία μεταβάλλεται διαφέρει από ένα τόπο σε έναν άλλο και μπορεί να θεωρηθεί ότι είναι *συνάρτηση* της τοποθεσίας. Ένα WSN μπορεί να χρησιμοποιηθεί για να προσεγγίσει μία τέτοια άγνωστη συνάρτηση χρησιμοποιώντας έναν περιορισμένο αριθμό δειγμάτων που θα λαμβάνονται από κάθε κόμβο αισθητήρων ξεχωριστά. Βέβαια αυτή η προσεγγιστική συνάρτηση θα πρέπει να είναι διαθέσιμη και στον sink. Το πόσο συχνά αυτή η συνάρτηση θα ανανεώνεται εξαρτάται από το tradeoff επιθυμητής ακρίβειας και κατανάλωσης ενέργειας.
4. **Παρακολούθηση:** Η πηγή (source) κάποιου γεγονότος μπορεί να είναι κινητή (π.χ. ένα σενάριο όπου ένας εισβολέας πρέπει να παρακολουθείται). Το WSN μπορεί να χρησιμοποιηθεί για να αναφέρει τη τοποθεσία του source στον/στους sink(s) ή ακόμα να αναφέρει εκτιμήσεις για κατεύθυνση κίνησης ή ταχύτητα. Για να γίνει αυτό αποδοτικά οι κόμβοι αισθητήρων πρέπει να συνεργαστούν πριν στείλουν οποιαδήποτε αναφορά στον sink.

Τα παραπάνω μοντέλα αλληλεπίδρασης μπορούν να συνδυαστούν και με περιορισμούς στο χρόνο και στο χώρο (π.χ. η αναφορά γεγονότων μπορεί να πρέπει να γίνει σε ένα συγκεκριμένο χρονικό διάστημα, και από μία επίσης συγκεκριμένη περιοχή). Αυτές βέβαια οι απαιτήσεις μπορεί να αλλάζουν με τον χρόνο. Έτσι οι sinks θα πρέπει να είναι σε θέση να ενημερώνουν του κόμβους του δικτύου για τις αλλαγές των απαιτήσεων κατά τη διάρκεια λειτουργίας του δικτύου. Τέλος αυτά τα μοντέλα μπορεί να αναφέρονται μόνο σε μια αίτηση (ερώτημα) του sink στο δίκτυο για κάποια πληροφορία (*'one-shot queries'*).

Τα παραπάνω παραδείγματα αποκαλύπτουν επίσης ότι υπάρχουν πάρα πολλοί τρόποι για να απλωθούν οι κόμβοι στο χώρο ενδιαφέροντος. Για παράδειγμα μπορούμε να τοποθετήσουμε τους κόμβους αισθητήρων σε συγκεκριμένες θέσεις, ή ακόμα και με τυχαία ρίψη αυτών από ένα αεροπλάνο, να παρακολουθήσουμε τις θερμοκρασίες ενός δάσους και να αποτρέψουμε μια πιθανή πυρκαγιά. Επίσης οι κόμβοι αισθητήρων μπορεί να είναι οι ίδιοι κινητοί και να βρίσκουν τις κατάλληλες θέσεις ώστε να ανιχνεύουν καλύτερα το στόχο τους.

Η διαφορετικότητα των εφαρμογών επηρεάζει επίσης τον τρόπο με τον οποίο θα συντηρείται το δίκτυο. Έτσι προκύπτουν τα εξής ερωτήματα: Υφίσταται η έννοια της διατήρησης δικτύου για τα WSNs, μιας και τα δίκτυα αυτά είναι ad-hoc με μικρό χρόνο ζωής και η αποστολή τους περιορίζεται στη παρακολούθηση κάποιου φαινομένου για πολύ μικρό διάστημα; Μήπως αυτά τα δίκτυα πρέπει αδιάσπαστα και συνεχώς να παρακολουθούν κάποιο φαινόμενο και έτσι η συντήρηση είναι πολύ σημαντική;

Ιδιαίτερα σχετικό ζήτημα με τη συντήρηση του δικτύου είναι και οι επιλογές για παροχή ενέργειας στους κόμβους του. Σε κάποιες εφαρμογές η παροχή ενέργειας μέσω σύρματος είναι εφικτή και έτσι δεν υπάρχει τέτοιο ζήτημα. Για κόμβους όμως που λειτουργούν αυτόνομα, το πρόβλημα της ενέργειας μπορεί να είναι είτε τετριμμένο (για εφαρμογές μερικών ημερών) είτε ιδιαίτερη πρόκληση ιδιαίτερα σε εφαρμογές που δεν είναι δυνατή η συντήρηση του δικτύου, και που όμως αυτό πρέπει να λειτουργεί για πολύ

καιρό. Προφανώς ο στόχος για κόμβους με προσιτή τιμή και μικρό μέγεθος παίζει σημαντικό ρόλο στο σχεδιασμό τροφοδοσίας ενέργειας.

1.3 ΠΡΟΚΛΗΣΕΙΣ ΓΙΑ ΤΑ WSNs

Τα WSNs αναφέρονται σε ένα τόσο μεγάλο σύνολο εφαρμογών που είναι εξαιρετικά δύσκολο να τα αντιληφθούμε μονοσήμαντα. Παρόλα αυτά εμφανίζουν κάποια κοινά γνωρίσματα, που έχουν να κάνουν με τα ιδιαίτερα χαρακτηριστικά και μηχανισμούς αυτών των δικτύων. Η πλήρης κατανόηση αυτών των χαρακτηριστικών και μηχανισμών είναι ιδιαίτερη πρόκληση για τα ασύρματα δίκτυα αισθητήρων.

1.3.1 Χαρακτηριστικές απαιτήσεις των WSNs

Τα παρακάτω χαρακτηριστικά είναι κοινά στα περισσότερα παραδείγματα εφαρμογών που περιγράψαμε παραπάνω:

1. **Τύπος Υπηρεσίας (ToS):** Ο τύπος υπηρεσίας που προσφέρεται από ένα δίκτυο επικοινωνίας είναι προφανής, μεταφέρει bit από τα ένα μέρος στο άλλο. Για ένα WSN η μεταφορά πληροφορίας είναι απλά μια λειτουργία και όχι ο αντικειμενικός σκοπός του δικτύου. Το WSN λοιπόν αναμένεται να παρέχει χρήσιμες πληροφορίες για κάποιο γεγονός και φαινόμενα που συμβαίνουν στο περιβάλλον. Έτσι λοιπόν προκύπτει μία διαφορετική αντίληψη για τις υπηρεσίες που μπορεί να παρέχει ένα δίκτυο.
2. **Ποιότητα Υπηρεσιών (QoS):** Πολύ στενά συσχετισμένη με το τύπο υπηρεσιών που προσφέρει ένα δίκτυο είναι και η ποιότητα υπηρεσιών του. Παραδοσιακά οι απαιτήσεις σε ποιότητα υπηρεσιών όπως για παράδειγμα ένα ανώτατο όριο καθυστέρησης ή περιορισμένο εύρος ζώνης δεν έχουν ιδιαίτερη σημασία όταν οι εφαρμογές είναι ανεκτικές στις καθυστερήσεις ή όταν το εύρος ζώνης των μεταδιδόμενων δεδομένων είναι αρκετά μικρό. Στα WSNs σε κάποιες περιπτώσεις η περιστασιακή μετάδοση ενός μόνο πακέτου είναι υπέρ-αρκετή. Σε κάποιες άλλες όμως εφαρμογές υπάρχουν απαιτήσεις για υψηλή αξιοπιστία μετάδοσης δεδομένων ή ακόμα οι καθυστερήσεις είναι ιδιαίτερα σημαντικές όπως όταν για παράδειγμα ένα WSN πρέπει να ανιχνεύσει πυρκαγιά σε κάποιο δάσος. Στην ποιότητα υπηρεσιών στα WSNs σημαντικό ρόλο παίζει η ποιότητα της πληροφορίας που φτάνει στους sinks.
3. **Ανοχή σε Σφάλματα (Fault Tolerance):** Από τη στιγμή που κάποιοι κόμβοι μπορεί να εξαντλήσουν τα αποθέματα ενέργειας τους ή να καταστραφούν ή ακόμα να μην μπορούν να επικοινωνήσουν εξαιτίας παρεμβολών και θορύβου στο ασύρματο κανάλι, είναι πολύ σημαντικό το WSN σαν ολοκληρωμένη οντότητα να μπορεί να αντεπεξέλθει στα παραπάνω προβλήματα.
4. **Χρόνο Ζωής του Δικτύου:** Σε πολλούς τύπους εφαρμογών οι κόμβοι του δικτύου πρέπει να βασιστούν σε περιορισμένες πηγές ενέργειας (π.χ.

- χρησιμοποιώντας μπαταρίες). Η αντικατάσταση των πηγών ενέργειας (π.χ. επαναφόρτιση των μπαταριών) μπορεί να είναι πρακτικά αδύνατη, την ίδια στιγμή που το δίκτυο πρέπει να στηρίζει την εφαρμογή για όσο γίνεται μεγαλύτερο χρονικό διάστημα. Έτσι μπορούμε να καταλάβουμε ότι ο χρόνος ζωής του δικτύου είναι ιδιαίτερα σημαντικός και επιβάλλει σχεδιασμό πρωτοκόλλων που είναι ιδιαίτερα αποδοτικά στο τομέα της ενέργειας.
5. **Κλιμακωσιμότητα (Scalability):** Από τη στιγμή που τα WSNs αποτελούνται από ένα μεγάλο αριθμό κόμβων, τα πρωτόκολλα και οι αρχιτεκτονικές των WSNs πρέπει να μπορούν να διαχειρίζονται ένα ιδιαίτερα μεταβαλλόμενο αριθμό κόμβων.
 6. **Συντηρησιμότητα (Maintainability):** Αφού και το φυσικό περιβάλλον του WSN και το ίδιο το WSN (καταστροφή κόμβων, εξάντληση ενέργειας, νέες εφαρμογές) αλλάζει συνεχώς το σύστημα πρέπει να είναι σε θέση να προσαρμόζεται. Πρέπει λοιπόν να είναι σε θέση να παρακολουθεί την δικιά του κατάσταση και να αλλάζει τις λειτουργικές παραμέτρους ή να επιλέγει διαφορετικά trade-offs (να παρέχει μικρότερης ποιότητας πληροφορία όταν οι ενεργειακοί πόροι είναι περιορισμένοι).
 7. **Programmability:** Οι κόμβοι δεν θα πρέπει να είναι σε θέση απλά να επεξεργάζονται πληροφορίες, αλλά επίσης θα πρέπει να αντιδρούν ευέλικτα όταν τους αναθέτει ο διαχειριστής του δικτύου να κάνουν κάποια διαφορετική εργασία. Αυτό σημαίνει ότι οι κόμβοι αυτοί θα πρέπει να μπορούν να προγραμματίζονται και να ξανά-προγραμματίζονται ότι το δίκτυο πρέπει να διεκπεραιώσει μία καινούργια εφαρμογή.
 8. **Πυκνότητα Δικτύου:** Σε ένα WSN ο αριθμός των κόμβων ανά μονάδα επιφανείας (*πυκνότητα δικτύου*) μπορεί να διαφέρει αρκετά σε διαφορετικά WSNs. Επίσης η πυκνότητα του δικτύου μπορεί να μην είναι ομοιογενής σε ολόκληρο το δίκτυο.

1.3.2 Απαιτούμενοι μηχανισμοί

Για να μπορούμε να αντιληφθούμε καλύτερα τις νέες απαιτήσεις και δυνατότητες των WSNs πρέπει να σχεδιάσουμε νέες εφαρμογές και πρωτόκολλα που θα βασίζονται σε διαφορετικές ιδέες. Ιδιαίτερη πρόκληση είναι να βρούμε μηχανισμούς που θα εστιάζουν στις ιδιαιτερότητες κάθε εφαρμογής ξεχωριστά και θα ικανοποιούν συγκεκριμένες απαιτήσεις σε ποιότητα υπηρεσίας, χρόνο ζωής και συντηρησιμότητα. Από την άλλη πλευρά αυτοί οι μηχανισμοί θα πρέπει να είναι συγχρόνως και γενικοί σε ένα ευρύτερο σύνολο εφαρμογών. Αυτό γιατί η κατασκευή εξολοκλήρου κάποιου καινούργιου πρωτοκόλλου για κάθε νέα εφαρμογή που προκύπτει θα είναι σίγουρα οικονομικά ανέφικτη.

Κάποιοι μηχανισμοί οι οποίοι αποτελούν τυπικό μέρος των WSNs δικτύων είναι οι εξής:

1. **Πολλαπλών Αλμάτων (Multihop) Ασύρματη Επικοινωνία:** Ενώ η ασύρματη επικοινωνία είναι βασική τεχνική μετάδοσης, η άμεση επικοινωνία μεταξύ πομπού και δέκτη εμφανίζει πολλούς περιορισμούς. Συγκεκριμένα η ασύρματη επικοινωνία σε πολύ μακρινές αποστάσεις είναι μόνο δυνατή με μια υψηλής ισχύος μετάδοση, με τη κατανάλωση ενός απαγορευτικού ποσού ενέργειας. Η μετάδοση της πληροφορίας μέσω πολλαπλών κόμβων (multihop propagation) μπορεί να συμβάλλει δραστικά στην εξοικονόμηση ενέργειας.
2. **Αποδοτικής Ενέργειας Λειτουργίες:** Για να μπορεί το δίκτυο να επιτύχει μεγάλους χρόνους ζωής οι αποδοτικής ενέργειας λειτουργίες είναι ιδιαίτερα σημαντικές. Οι αποδοτικής ενέργειας λειτουργίες περιλαμβάνουν ενεργειακά αποδοτική μεταφορά δεδομένων ανάμεσα στους κόμβους και επίσης αποδοτική διαχείριση της πληροφορίας που ανιχνεύει το WSN από το περιβάλλον. Επίσης ένα πολύ σημαντικό θέμα είναι η μη-ομοιογενής κατανάλωση ενέργειας και η δημιουργία hot-spot.
3. **Αυτό-διαμόρφωση (auto-configuration):** Ένα WSN δίκτυο θα πρέπει να ρυθμίζει και να διαμορφώνει τις παραμέτρους του αυτόνομα και όχι να περιμένει κάποια εξωτερική υποστήριξη μιας και ο μεγάλος αριθμός των κόμβων, και πεδίο ενδιαφέροντος που απλώνεται το δίκτυο κάνουν αυτήν την επιλογή απαγορευτική. Για παράδειγμα, οι κόμβοι θα πρέπει να είναι σε θέση να καθορίσουν τις γεωγραφικές τους θέσεις ανταλλάσσοντας μηνύματα μεταξύ τους, σε μια διαδικασία που είναι γνωστή ως *self-location*. Επίσης το δίκτυο πρέπει να μπορεί να αντεπεξέλθει σε περιπτώσεις καταστροφής κόμβων ή σε περιπτώσεις αύξησης της πυκνότητας του δικτύου με εισαγωγή καινούργιων κόμβων.
4. **Συνεργασία και Ενδοδοικτυακή Επεξεργασία:** Σε κάποιες εφαρμογές ένας μόνο κόμβος αισθητήρων δεν μπορεί να αποφασίσει αν ένα γεγονός έχει συμβεί και έτσι απαιτείται η συνεργασία πολλών κόμβων για να ανιχνεύσουν ένα γεγονός και επίσης απαιτείται η συγκέντρωση δεδομένων από πολλούς κόμβους για είναι οι πληροφορίες για αυτό το γεγονός επαρκής. Οι πληροφορίες λοιπόν μπορούν να επεξεργαστούν από το ίδιο το δίκτυο με διάφορους τρόπους και με τη συνεργασία των κόμβων, και δε χρειάζεται να μεταφερθούν στα «άκρα» του δικτύου για να γίνει αυτή η διαδικασία. Ένα τέτοιο παράδειγμα είναι η απόφαση της υψηλότερης ή της μέγιστης θερμοκρασίας σε μια περιοχή, και αναφορά αυτής σε κάποιο sink. Για να γίνει αυτός ο υπολογισμός αποδοτικά, οι θερμοκρασίες που ανιχνεύονται από κάθε ένα κόμβο ξεχωριστά μπορούν να συγκεντρώνονται καθώς διαδίδονται μέσα από το δίκτυο, μειώνοντας έτσι τη ποσότητα δεδομένων που θα έπρεπε να μεταδοθούν στον sink, βελτιώνοντας έτσι την αποδοτικότητα σε ενέργεια. Το πώς θα γίνεται αυτή η συγκέντρωση είναι θέμα υπό συζήτηση.
5. **Δίκτυα Προσανατολισμένα στα Δεδομένα:** Τα παραδοσιακά δίκτυα επικοινωνιών προσανατολίζονται συνήθως στο να μεταφέρουν δεδομένα μεταξύ δύο συγκεκριμένων συσκευών, όπου σε κάθε μία έχει ανατεθεί μία διεύθυνση και άρα τέτοια δίκτυα μπορούν να χαρακτηριστούν ως δίκτυα βασισμένα σε διευθύνσεις (*address-centric*). Αντίθετα σε ένα WSN όπου πολλοί κόμβοι τυπικά απλώνονται αυθαίρετα για να προστατέψουν το δίκτυο από αποτυχίες κόμβων και να αντισταθμίσουν τη χαμηλή δυνατότητα «αίσθησης» κάθε κόμβου αισθητήρων ατομικά, η ταυτότητα κάθε συγκεκριμένου κόμβου δεν έχει σημασία. Αυτό που έχει πραγματικά σημασία σε ένα WSN είναι οι μετρήσεις κάθε κόμβου και όχι η

ταυτότητα κάθε κόμβου που παρείχε αυτές. Έτσι μεταβαίνουμε από τα *address-centric* δίκτυα στα *data-centric*.

6. **Τοπικότητα:** Στη περίπτωση των WSN η αρχή της τοπικότητας πρέπει να εφαρμοστεί για να διατηρηθεί η κλιμακωσιμότητα του δικτύου. Οι κόμβοι οι οποίοι διαθέτουν περιορισμένους πόρους σε μνήμη, πρέπει να διατηρούν κατά την εφαρμογή κάποιου πρωτοκόλλου την κατάσταση το πολύ των γειτονικών τους κόμβων. Αυτό θα επιτρέπει στο δίκτυο να ενσωματώνει μεγάλους αριθμούς κόμβων χωρίς να χρειάζεται καθένας από αυτούς να διαθέτει μεγάλη επεξεργαστική ισχύ και μνήμη. Τα πως βέβαια θα συνδυαστεί η αρχή της τοπικότητας με τη σχεδίαση αποδοτικών πρωτοκόλλων είναι ένα ανοιχτό θέμα.
7. **Trade-offs:** Τα WSNs καλούνται να διαχειριστούν μεγάλα trade-offs μεταξύ αντιφατικών στόχων και κατά τη σχεδίαση πρωτοκόλλων και κατά το χρόνο εκτέλεσης. Τέτοια trade-offs όπως έχουμε αναφέρει παραπάνω μπορεί να είναι: α) η μεγαλύτερη κατανάλωση ενέργειας μπορεί να προσφέρει μεγαλύτερη ακρίβεια αποτελεσμάτων, β) ο εξασφάλιση «μακροζωίας» για ολόκληρο το δίκτυο μπορεί να επηρεάσει τη ζωή κάποιων συγκεκριμένων κόμβων. Επίσης ένα άλλο trade-off εισάγεται από τη πυκνότητα των κόμβων στο δίκτυο. Ανάλογα με την εφαρμογή, την κατανομή και τις αποτυχίες των κόμβων κατά την εκτέλεση, η πυκνότητα του δικτύου μπορεί να αλλάξει σημαντικά και κάθε πρωτόκολλο πρέπει να είναι σε θέση να χειριστεί αυτές τις διαφορετικές καταστάσεις.

Οι παραπάνω μηχανισμοί πρέπει να ληφθούν σοβαρά υπόψη για τον σχεδιασμό πρωτοκόλλων σε WSNs.

1.4 ΚΙΝΗΤΑ Ad Hoc ΔΙΚΤΥΑ & WSNs

Ένα ad hoc δίκτυο είναι ένα δίκτυο που εγκαθίσταται για να εξυπηρετήσει μία γρήγορα εμφανιζόμενη επικοινωνιακή ανάγκη. Το απλούστερο παράδειγμα ενός ad hoc δικτύου είναι ίσως ένα σύνολο υπολογιστών που συνδέονται μέσω καλωδίων για να σχηματίσουν ένα μικρό δίκτυο, όπως π.χ. κάποιο φορητοί υπολογιστές σε μία αίθουσα συναντήσεων. Σε αυτό το παράδειγμα υπάρχει η ιδέα της αυτό-διαμόρφωσης (self-configuration) αφού το δίκτυο αναμένεται να λειτουργήσει χωρίς χειροκίνητη διαμόρφωση.

Συνήθως όμως η ιδέα του MANET σχετίζεται με την ασύρματη επικοινωνία και συγκεκριμένα με την ασύρματη multihop επικοινωνία. Όπως άλλωστε υπονοούν και τα αρχικά του, το MANET είναι αναφέρεται σε κινητούς κόμβους. Παραδείγματα τέτοιων δικτύων βρίσκουν εφαρμογές σε πεδία καταστροφών (οι πυροσβέστες επικοινωνούν μεταξύ τους) ή σε μεγάλα κατασκευαστικά πεδία όπου η κατασκευή access points (πόσο μάλλον καλωδίων) είναι αδύνατη. Σε τέτοια λοιπόν δίκτυα, πολλοί κόμβοι ατομικά σχηματίζουν ένα μεγάλο δίκτυο, όπου τα πακέτα μεταφέρονται μέσω πολλών κόμβων (multihop propagation) για να φτάσουν τελικά στο κόμβο παραλήπτη, επιτρέποντας τη κατασκευή δικτύων για μεγάλες γεωγραφικά περιοχές όπου η άμεση επικοινωνία πομπού-δέκτη είναι αδύνατη. Οι δύο βασικές προκλήσεις που έχουν να αντιμετωπίσουν τα MANETs είναι η οργάνωση του δικτύου μιας και οι κόμβοι αλλάζουν συνεχώς θέση,

και οι περιορισμοί που εισάγει το ασύρματο κανάλι. Η σχετική έρευνα που αναφέρεται στις δύο παραπάνω προκλήσεις είναι μία πολύ ενεργή περιοχή. Πολύ διάσημο είναι το βιβλίο [3].

Τα γενικότερα αυτά θέματα υπάρχουν και στα WSNs και στα MANETs. Όμως υπάρχουν και ιδιαίτερες διαφορές μεταξύ αυτών των τύπων δικτύων που ορίζουν διαφορετικές ερευνητικές κατευθύνσεις.

1. **Εφαρμογές και Εξοπλισμός:** Τα MANETs αναφέρονται σε διαφορετικές εφαρμογές και άρα σε διαφορετικό εξοπλισμό από τα WSNs. Σε ένα MANET ένα τερματικό μπορεί να είναι αρκετά ισχυρό (laptop ή PDA) με αρκετά αποθέματα ενέργειας. Αυτός ο εξοπλισμός είναι απαραίτητος μιας και στις MANET εφαρμογές συνήθως εμπλέκονται άνθρωποι. Για παράδειγμα ένα MANET μπορεί να χρησιμοποιηθεί για φωνητική επικοινωνία μεταξύ δύο ομότιμων χρηστών ή για απομακρυσμένη επικοινωνία με κάποιον Web Server. Συμπερασματικά ο εξοπλισμός πρέπει να είναι ισχυρός ώστε να υποστηρίξει αυτές τις εφαρμογές.
2. **Συγκεκριμένες Εφαρμογές:** Εξαιτίας του πολύ μεγάλου αριθμού συνδυασμών των τεχνολογιών αισθητήρων, υπολογισμού και επικοινωνίας πολλά σενάρια εφαρμογών για ένα WSN είναι πιθανά. Είναι λοιπόν απίθανο να υπάρχει ένα μοναδικό μοντέλο για τόσες πολλές εφαρμογές. Για παράδειγμα η διαφορετική πυκνότητα που μπορεί να έχει ένα WSN μπορεί να απαιτεί το σχεδιασμό διαφορετικών πρωτοκόλλων ή τουλάχιστον προσαρμοστικών πρωτοκόλλων. Αν και αυτή η ποικιλομορφία των εφαρμογών υπάρχει και στα MANETs, δεν είναι όμως τόσο έντονη.
3. **Αλληλεπίδραση με το Περιβάλλον:** Από τη στιγμή που τα WSNs πρέπει να αλληλεπιδράσουν με το περιβάλλον, τα χαρακτηριστικά της κίνησης δεδομένων μέσα στο δίκτυο είναι πολύ διαφορετικά από αυτά των δικτύων που αλληλεπιδρούν με τον άνθρωπο. Μια συνέπεια αυτού είναι ότι τα WSNs μπορεί να παρουσιάζουν πολύ μικρούς ρυθμούς μετάδοσης δεδομένων για μεγάλο χρονικό διάστημα, αλλά να εμφανίσουν καταγιστική (*bursty*) κίνηση όταν συμβεί κάποιο γεγονός. Οι πολύ μεγάλες λοιπόν περίοδοι (μήνες) αδράνειας, μπορεί να δώσουν τη θέση τους σε πολύ μικρές περιόδους (λεπτά ή δευτερόλεπτα) υψηλής δραστηριότητας «πιέζοντας» τη χωρητικότητα του δικτύου στα όρια της. Από την άλλη πλευρά τα MANETs χρησιμοποιούνται για να υποστηρίξουν πιο παραδοσιακές εφαρμογές (Ιστός, φωνή κ.τ.λ.) και έτσι συγκριτικά έχουν πιο κατανοητά χαρακτηριστικά κίνησης δεδομένων.
4. **Κλιμακωσιμότητα:** Συνήθως τα WSNs πρέπει να ενσωματώσουν μεγαλύτερους αριθμούς (χιλιάδες ή εκατοντάδες χιλιάδες) από κόμβους σε αντίθεση με τα MANETs, και έτσι απαιτούν πιο κλιμακωτό (*scalable*) σχεδιασμό. Σαν αποτέλεσμα η ανάθεση σε κάθε κόμβο αισθητήρων ενός μοναδικού αναγνωριστικού, έχει ιδιαίτερο κόστος (είτε κατά παραγωγή των κόμβων, είτε κατά το χρόνο εκτέλεσης) και αποτελεί βάρος που πρέπει να αποφευχθεί. Έτσι λοιπόν τα MANET's πρωτόκολλα που χρησιμοποιούν αναγνωριστικά δεν εφαρμόζονται στα WSNs.
5. **Ενέργεια:** Στα WSNs αλλά και στα MANETs η ενέργεια αποτελεί πολύτιμο και συνάμα περιορισμένο πόρο. Όμως το θέμα της ενέργειας στα WSNs και κατά προέκταση ο χρόνος ζωής του δικτύου, είναι περισσότερο φλέγων μιας και σε μια

WSN εφαρμογή η δυνατότητα να αντικατάστασης ή επαναφόρτισης της μπαταρίας του κόμβου πολλές φορές δεν υπάρχει.

6. **Αυτό-διαμόρφωση (self-configuration):** Όπως και στα ad hoc έτσι και στα δίκτυα αισθητήρων οι κόμβοι πρέπει να οργανωθούν μόνοι τους σε δίκτυο και έτσι οι αυτό-διαμόρφωση αποτελεί κοινό χαρακτηριστικό και για τους δύο τύπους δικτύων. Εξαιτίας όμως του μεγάλου αριθμού κόμβων και της περιορισμένη ενέργειας, η αυτό-διαμόρφωση στα WSNs γίνεται πολλές φορές πιο πολύπλοκη.
7. **QoS:** Τα θέματα ποιότητας υπηρεσίας σε ένα MANET πηγάζουν από τις παραδοσιακές εφαρμογές (π.χ καλή ποιότητα μεταδιδόμενης φωνής). Αντίθετα στα WSNs υπάρχουν νέα δεδομένα στο θέμα του QoS όπως αναφέραμε και παραπάνω.
8. **Απλότητα:** Από τη στιγμή που οι κόμβοι αισθητήρων πρέπει να είναι απλοί και μπορούν να διαθέτουν περιορισμένα αποθέματα ενέργειας, το λειτουργικό σύστημα και το λογισμικό δικτύωσης πρέπει να είναι απλούστερα από αυτά ενός desktop υπολογιστή. Αυτή η απλότητα μπορεί όμως να απαιτεί την παραβίαση του μοντέλου επιπέδων για το λογισμικό δικτύωσης αφού η αφαιρετικότητα του μοντέλου επιπέδων μπορεί να έχει μεγάλο κόστος. Επίσης άλλες πηγές όπως η μνήμη που σχετίζεται άμεσα με τα απαιτητικά σε μνήμη πρωτόκολλα δρομολόγησης στα MANETs, δεν είναι διαθέσιμη σε μεγάλες ποσότητες στα WSNs, κάνοντας επιτακτική την ανάγκη διαφορετικών σχεδιαστικών λύσεων.
9. **Κινητικότητα (mobility):** Το πρόβλημα της κινητικότητας στα MANETs προκαλείται από τη κίνηση των κόμβων που σαν συνέπεια έχει να αλλάζει τις multihop διαδρομές μεταξύ των κόμβων. Στα WSNs το ίδιο πρόβλημα μπορεί να υπάρχει αν οι κόμβοι αισθητήρων είναι κινητοί. Όμως υπάρχουν επίσης δύο επιπλέον όψεις της κινητικότητας στη περίπτωση των WSNs.

Στη 1^η περίπτωση το δίκτυο αισθητήρων μπορεί να χρησιμοποιείται για να ανιχνεύει και να παρατηρεί φυσικά φαινόμενα (π.χ ανίχνευση κάποιου εισβολέα). Αυτό φαινόμενο που μπορεί να είναι μια ακολουθία γεγονότων που συμβαίνουν στο δίκτυο, προκαλεί επίσης μια τοπική επεξεργασία των δεδομένων όπως για παράδειγμα για να αποφασιστεί αν πράγματι υπάρχει εισβολέας. Όμως τι θα συμβεί αν το φαινόμενο μετακινηθεί. Ιδανικά τα δεδομένα που συγκεντρώνονται σε ένα μέρος του δικτύου θα πρέπει να είναι διαθέσιμα στο σημείο που θα κινηθεί ο στόχος (π.χ. εισβολέας).

Στη 2^η περίπτωση οι sinks του δικτύου (κόμβοι στους οποίους πρέπει να παραδοθούν τα γεγονότα που ανιχνεύτηκαν) μπορεί να είναι επίσης κινητοί. Βασικά το σενάριο αυτό δεν φαίνεται να έχει διαφορά από αυτό των κινητών κόμβων σε ένα MANET, αλλά όμως μπορεί να προκαλέσει προβλήματα σε πρωτόκολλα που δουλεύουν αποδοτικά σε στατικά περιβάλλοντα.

Τέλος και στα WSNs και στα MANETs η κινητικότητα είναι συσχετισμένη με την έννοια ότι μία ομάδα κόμβων κινείται με το ίδιο τρόπο. Σε ένα MANET ένα παράδειγμα συσχετιζόμενης κίνησης είναι μία ομάδα ανθρώπων που ταξιδεύουν μαζί, ενώ σε ένα WSN οι κόμβοι μπορούν να κινούνται παρασυρόμενοι από μία καταγίδα ή ένα ποτάμι.

Ανακεφαλαιώνοντας μπορούμε να παρατηρήσουμε ότι υπάρχουν αρκετές ομοιότητες μεταξύ των δύο τύπων δικτύων. Όμως το γεγονός ότι τα WSNs αναφέρονται

σε μια πολύ μεγάλη ποικιλία εφαρμογών, αλληλεπιδρούν με το φυσικό περιβάλλον και εισάγουν πολλά trade-offs, τα κάνει να ξεχωρίζουν από τα MANETs.

1.5 ΤΕΧΝΟΛΟΓΙΕΣ ΓΙΑ ΤΑ WSNs

Η κατασκευή ασύρματων δικτύων αισθητήρων έχει γίνει δυνατή μετά την ανάπτυξη συγκεκριμένων τεχνολογιών. Πρωταρχική από αυτές είναι η σμίκρυνση του hardware. Τα μικρότερα μεγέθη των chips έχουν οδηγήσει στη μείωση της κατανάλωσης ενέργειας σε ένα επίπεδο όπου η κατασκευή των WSNs είναι πλέον εφικτή. Αυτή η σμίκρυνση αναφέρεται κυρίως στους μικρό-ελεγκτές, στα chip μνήμης αλλά και ράδιο-modem που είναι υπεύθυνα για την ασύρματη τεχνολογία. Πέρα από οικονομία στην ενέργεια οι νέες τεχνολογίες κοστίζουν λιγότερο, κάνοντας εφικτή τη κατασκευή δικτύων με πολύ μεγάλο αριθμό κόμβων. Φυσικά οι παραπάνω τεχνολογίες συνδυάζονται και με την ανάπτυξη στη τεχνολογία αισθητήρων για την οποία θα μιλήσουμε παρακάτω.

Τα παραπάνω βασικά τεχνολογικά χαρακτηριστικά ενός κόμβου αισθητήρων θα πρέπει να συνοδεύονται και από ένα παροχέα ενέργειας. Έτσι λοιπόν και ανάλογα με την εφαρμογή, απαιτούνται υψηλής χωρητικότητας και μεγάλης διάρκειας μπαταρίες, που θα έχουν αμελητέο ρυθμό αυτό-εκφορτισμού (self-discharge rate) και οι οποίες θα μπορούν να παρέχουν μικρές ποσότητες ρεύματος. Ιδανικά ένας κόμβος αισθητήρων πρέπει να έχει επίσης μια συσκευή επαναφόρτισης της μπαταρίας ο οποία θα χρησιμοποιεί το περιβάλλον (π.χ. ηλιακά κύτταρα).

Το συμπληρωματικό των τεχνολογιών hardware είναι το software. Η πρώτη ερώτηση που πρέπει να απαντηθεί σε αυτό το σημείο είναι λειτουργικότητα ενός κόμβου (το λειτουργικό του σύστημα, το περιβάλλον εκτέλεσης). Το περιβάλλον του κόμβου πρέπει να υποστηρίζει ανταλλαγή πληροφοριών μεταξύ επιπέδων, τμηματικότητα (modularity) και να μπορεί να επαναθέτει εργασίες. Επίσης αυτή η αρχιτεκτονική software ενός κόμβου πρέπει να επεκταθεί σε μια δικτυακή αρχιτεκτονική όπου το κύριο ερώτημα είναι πως θα διαχωριστούν οι εργασίες ανάμεσα στους κόμβους του δικτύου. Τέλος πολύ σημαντικός είναι ο τρόπος σχεδιασμού πρωτοκόλλων επικοινωνίας.

ΜΕΡΟΣ 2^ο : ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ

2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΟΜΒΟΥ

Για να χτίσεις ένα ασύρματο δίκτυο αισθητήρων είναι να είναι διαθέσιμοι οι κόμβοι αισθητήρων. Αυτή οι κόμβοι πρέπει να πληρούν τις απαιτήσεις της συγκεκριμένης εφαρμογής. Πιθανόν θα πρέπει να είναι μικροί, φθηνοί, ενεργειακά αποδοτικοί, θα πρέπει να είναι εξοπλισμένοι με τους κατάλληλους αισθητήρες, θα πρέπει να έχουν τους απαραίτητους πόρους σε μνήμη και υπολογιστική ισχύ και τέλος θα πρέπει να είναι σε θέση να επικοινωνούν μεταξύ τους. Σε αυτό λοιπόν το κεφάλαιο θα περιγράψουμε τα hardware συστατικά τέτοιων κόμβων και κάποιες τεχνολογίες για κόμβους αισθητήρων. Θα αναφερθούμε όμως στην αρχιτεκτονική του κόμβου ως ξεχωριστή οντότητα. Στο επόμενο κεφάλαιο θα αναλύσουμε την αρχιτεκτονική των δικτύων αισθητήρων.

2.1 HARDWARE ΣΥΣΤΑΤΙΚΑ ΤΟΥ ΚΟΜΒΟΥ

Αρχικά θα δώσουμε μια περίληψη του υλικού ενός κόμβου αισθητήρων και στη συνέχεια θα προσπαθήσουμε να περιγράψουμε κάθε συστατικό του με μεγαλύτερη λεπτομέρεια.

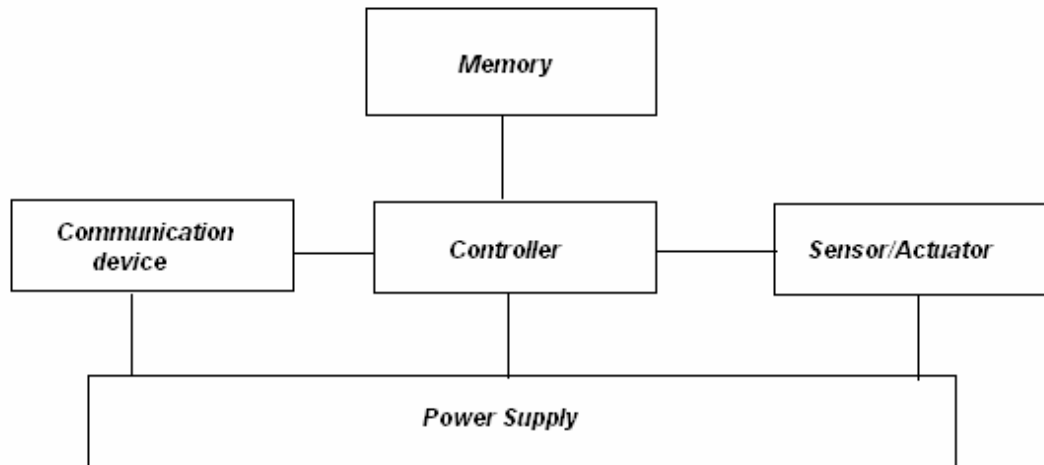
2.1.1 Σύνοψη υλικού

Όταν επιλέγονται τα hardware συστατικά ενός κόμβου αισθητήρων, είναι σαφές ότι οι απαιτήσεις της εφαρμογής παίζουν σημαντικό ρόλο αφού καθορίζουν το κόστος, το μέγεθος, την κατανάλωση ενέργειας και τις δυνατότητες επικοινωνίας ενός κόμβου. Για παράδειγμα σε κάποιες ακραίες περιπτώσεις, ολόκληρος ο κόμβος αισθητήρων θα πρέπει να είναι μικρότερος από 1 cm, να ζυγίζει λιγότερο από 100 g, να είναι φθηνότερος από 1\$ και να δαπανά ενέργεια λιγότερη από 100 μ W. Σε ακόμα πιο ακραίες περιπτώσεις υπάρχει ο ισχυρισμός ότι οι κόμβοι μπορεί να έχουν το μέγεθος κόκκων άμμου. Για αυτό τα δίκτυα αισθητήρων πολλές φορές αναφέρονται και ως *Δίκτυα Έξυπνης Σκόνης* ή *Smart Dust Networks*. Επίσης οι κόμβοι του δικτύου αναφέρονται και ως *grain particles* (η *σωματίδια σκόνης*). Συνήθως δεν είναι το μέγεθος ο πιο σημαντικός παράγοντας αλλά το κόστος και η κατανάλωση ενέργειας.

Αν και δεν υπάρχει βεβαίως ένα μόνο διαθέσιμο πρότυπο, ούτε η απαίτηση ένα πρότυπο αρχιτεκτονικής να καλύπτει όλες τις δυνατές εφαρμογές, σε αυτό το κεφάλαιο θα μελετήσουμε μια τυπική αρχιτεκτονική κόμβου αισθητήρων. Έτσι λοιπόν ένας τυπικός κόμβος αισθητήρων αποτελείται από πέντε κύρια τμήματα (σχήμα 2.1):

1. **Controller:** Ο ελεγκτής (controller) επεξεργάζεται τα σχετικά δεδομένα και είναι ικανός να εκτελέσει και κώδικα.

2. **Memory:** Κάποια μνήμη είναι απαραίτητη για να αποθηκεύει προγράμματα και ενδιάμεσα δεδομένα. Συνήθως διαφορετικοί τύποι μνήμης χρησιμοποιούνται για δεδομένα και για προγράμματα.
3. **Sensors και Actuators:** Αποτελούν τη διεπαφή με τον φυσικό κόσμο. Οι αισθητήρες λοιπόν είναι οι συσκευές που μπορούν να ελέγξουν ή να παρατηρήσουν φυσικές παραμέτρους του περιβάλλοντος.
4. **Communication:** Για να μπορούν οι κόμβοι να αποτελούν δίκτυο απαιτούν συσκευές που θα στέλνουν και θα λαμβάνουν πληροφορίες μέσω του ασύρματου καναλιού.



Σχήμα 2.1. Περίληψη των hardware συστατικών ενός κόμβου αισθητήρων.

5. **Power Supply:** Κάποια πηγή ενέργειας όπως η μπαταρία θα πρέπει να είναι διαθέσιμη σε κάθε κόμβο. Επίσης κάποιος τρόπος επαναφόρτισης μέσω του περιβάλλοντος όπως ηλιακά κύτταρα θα πρέπει να διαθέτει ιδανικά κάθε κόμβος.

Κάθε συστατικό από τα παραπάνω πρέπει να ισορροπήσει το trade-off μεταξύ μικρής κατανάλωσης ενέργειας και εκπλήρωση του έργου που πρέπει να φέρει εις πέρας ο κόμβος. Για παράδειγμα και η συσκευή επικοινωνίας αλλά και ο ελεγκτής πρέπει να είναι απενεργοποιημένοι για όσο καιρό αυτό είναι δυνατό. Για να ξυπνήσει ξανά από τη περίοδο sleep, μπορεί να χρησιμοποιήσει ένα εξ' αρχής προγραμματισμένο χρονόμετρο το οποίο θα τον ενεργοποιήσει μετά από κάποιο χρόνο. Επίσης οι αισθητήρες μπορεί να προγραμματιστούν έτσι ώστε να δημιουργούν μια interrupt κάθε φορά που ανιχνεύεται ένα γεγονός ή η συσκευή επικοινωνίας δέχεται μια εισερχόμενη μετάδοση.

Για να υλοποιηθούν αυτές λειτουργίες ειδοποίησης απαιτείται κατάλληλη διασύνδεση μεταξύ των διάφορων συστατικών υλικού. Επίσης είναι απαραίτητο τα συστατικά αυτά μέρη να ανταλλάζουν και πληροφορίες ελέγχου αλλά και δεδομένων μέσω αυτών των διασυνδέσεων. Οι διασυνδέσεις αυτές μπορούν να υλοποιηθούν πολύ απλά, όπως για παράδειγμα ένας αισθητήρας μπορεί να αναφέρει μια αναλογική τιμή στον ελεγκτή ή με κάποιο έξυπνο τρόπο να έκανε μια προεπεξεργασία στα δεδομένα που

ανίχνευσε και αν αυτά αντιστοιχούσαν σε ένα πραγματικό γεγονός τότε να ξυπνούσε τον ελεγκτή. Μια τέτοια προεπεξεργασία θα ήταν σημαντική για την οικονομία ενέργειας.

2.1.2 Ελεγκτής

Ο ελεγκτής (controller) μπορεί να θεωρηθεί ο πυρήνας του ασύρματου κόμβου αισθητήρων. Έτσι αυτός μπορεί να συλλέξει δεδομένα από τους αισθητήρες, να τα επεξεργαστεί, να αποφασίσει πότε και που πρέπει να στείλει, να λάβει δεδομένα από άλλους κόμβους του δικτύου και να αποφασίσει τέλος τις παραμέτρους του αισθητήρα. Επίσης είναι υπεύθυνος να εκτελεί μια ποικιλία προγραμμάτων, τα οποία ποικίλλουν από πρωτόκολλα επικοινωνίας και επεξεργασίας σήματος, μέχρι προγράμματα εφαρμογών. Ουσιαστικά ο controller είναι η κεντρική μονάδα επεξεργασίας (CPU) του κόμβου.

Μια τέτοια ποικιλία καθηκόντων επεξεργασίας μπορούν να εκτελεστούν σε διάφορες αρχιτεκτονικές ελεγκτών όπου η κάθε μια εισάγει διαφορετικά trade-offs σχετικά με την ευελιξία, τις επιδόσεις, την αποδοτικότητα ενέργειας και το κόστος.

Μία λύση για είναι η χρήση γενικού σκοπού επεξεργαστές όπως αυτούς που χρησιμοποιούνται στους desktop υπολογιστές. Όμως αυτοί οι επεξεργαστές τροφοδοτούνται με μεγάλα ποσά ενέργειας και έτσι η κατανάλωση είναι απαγορευτική. Υπάρχουν επίσης και μικρότεροι επεξεργαστές οι οποίοι χρησιμοποιούνται γενικότερα στα ενσωματωμένα συστήματα και αναφέρονται συνήθως ως *μικρό-ελεγκτές* (*microcontrollers*). Κάποια βασικά χαρακτηριστικά των μικρό-ελεγκτών που τους κάνουν να ταιριάζουν ιδιαίτερα με τα ενσωματωμένα συστήματα είναι η ευελιξία σύνδεσης τους με άλλες συσκευές (όπως αισθητήρες), το σύνολο εντολών τους είναι κατάλληλο για επεξεργασία σήματος ευαίσθητη στο χρόνο και τέλος δεν απαιτούν μεγάλα ποσά ενέργειας. Επίσης ένα ακόμα πλεονέκτημα είναι ότι έχουν ενσωματωμένη μνήμη. Τέλος οι μικρό-ελεγκτές μπορούν να προγραμματιστούν εύκολα χαρακτηριστικό που τους δίνει ιδιαίτερη ευελιξία. Οι μικρό-ελεγκτές είναι λοιπόν κατάλληλοι για τα WSNs εξαιτίας των παραπάνω, αλλά και από την επιπλέον δυνατότητα τους να μειώνουν την κατανάλωση ενέργειας μπαίνοντας σε μια κατάσταση ύπνου (sleep-state) στην οποία βρίσκονται σε λειτουργία μόνο κάποια κομμάτια του ελεγκτή. Μία από τις κύριες διαφορές μεταξύ των συστημάτων γενικού σκοπού και των microcontrollers είναι ότι οι δεύτεροι δεν διαθέτουν μονάδα διαχείρισης μνήμης, γεγονός που περιορίζει τη λειτουργικότητα της μνήμης. Για παράδειγμα είναι πολύ δύσκολο ή ακόμα και αδύνατο να εξασφαλίσουμε protected ή virtual memory στη περίπτωση του microcontroller.

Μία ειδική περίπτωση προγραμματιζόμενων επεξεργαστών είναι οι *Ψηφιακοί Επεξεργαστές Σήματος* (DSPs). Αυτοί οι επεξεργαστές είναι κατασκευασμένοι όσον αφορά την αρχιτεκτονική και το σύνολο εντολών τους, ώστε να επεξεργάζονται μεγάλες ποσότητες διανυσματικών δεδομένων, όπως συμβαίνει στις εφαρμογές επεξεργασίας σήματος. Σε ένα ασύρματο κόμβο αισθητήρων ένας DSP μπορεί να χρησιμοποιηθεί για να επεξεργαστεί δεδομένα που προέρχονται από απλά αναλογική ασύρματη συσκευή και να εξάγει μία ψηφιακή ροή δεδομένων. Σε ευρυζωνικές ασύρματες επικοινωνίες οι DSPs αποτελούν μία επιτυχώς χρησιμοποιούμενη πλατφόρμα. Στα WSNs όμως οι απαιτήσεις σε ασύρματη επικοινωνία είναι μικρότερες και οι εργασίες που έχουν να κάνουν με την

επεξεργασία σήματος και είναι σχετικές με την ανίχνευση της πληροφορίας δεν είναι τόσο πολύπλοκες. Συμπερασματικά τα πλεονεκτήματα των DSPs δεν αξιοποιούνται σε ένα WSN κόμβο και άρα αυτή η τεχνολογία δεν χρησιμοποιείται.

Μία άλλη εναλλακτική για τη περίπτωση του controller είναι η χρήση των *Field Programmable Gate Arrays (FPGAs)* ή των *Application-Specific Integrated Circuits (ASICs)*. Ένα FPGA μπορεί να επανά-προγραμματιστεί έτσι ώστε να προσαρμοστεί στις μεταβαλλόμενες απαιτήσεις, πράγμα όμως που θα σημαίνει σπατάλη χρόνου και ενέργειας. Ένας ASIC είναι ένας ειδικευμένος επεξεργαστής σχεδιασμένος για μια συγκεκριμένη εφαρμογή όπως για παράδειγμα για ένα υψηλής ταχύτητας δρομολογητή. Το trade-off σε αυτή τη περίπτωση είναι η απώλεια ευελιξίας σε αντάλλαγμα με επιδόσεις και αποδοτικότητα στην κατανάλωση ενέργειας.

Για μία «αφοσιωμένη» WSN εφαρμογή, όπου τα καθήκοντα ενός κόμβου αισθητήρων δεν αλλάζουν κατά τη διάρκεια της ζωής του, και ο αριθμός των κόμβων είναι αρκετά μεγάλος ώστε να δικαιολογεί την επένδυση στους ASICs, αυτοί οι επεξεργαστές μπορεί να είναι μία πολύ καλή λύση. Στο παρόν στάδιο της τεχνολογίας όμως, η μεγάλη ευελιξία και η απλότητα στη χρήση των μικρό-ελεγκτών, τους καθιστά τη προτιμότερη λύση.

2.1.3 Μνήμη

Όσον αφορά τη μνήμη, προφανώς ο κόμβος αισθητήρων χρειάζεται μια μνήμη τυχαίας προσπέλασης (RAM) για να αποθηκεύει τα δεδομένα που ανιχνεύει, τα πακέτα από άλλου κόμβους κ.τ.λ. Αν και η RAM είναι γρήγορη, το κύριο μειονέκτημα της είναι ότι χάνει τα περιεχόμενα της αν η τροφοδοσία ενέργειας διακοπεί. Ο κώδικας του προγράμματος μπορεί να αποθηκευτεί ROM ή πιο τυπικά στην EEPROM ή στην flash μνήμη (μοιάζει με την EEPROM απλά επιτρέπει στα δεδομένα να γράφονται ή να διαγράφονται κατά blocks, αντί ένα byte τη φορά). Η flash μνήμη μπορεί επίσης να χρησιμοποιηθεί σαν ενδιάμεσος αποθήκη για τα δεδομένα δε περίπτωση που η RAM είναι ανεπαρκής ή η τροφοδοσία έχει διακοπεί για κάποιο χρονικό διάστημα. Οι καθυστερήσεις προσπέλασης της flash μνήμης και η υψηλή απαίτηση ενέργειας πρέπει να ληφθούν υπόψη.

Το να έχει κυρίως η RAM το κατάλληλο μέγεθος είναι ιδιαίτερα σημαντικό και από άποψη κόστους αλλά και εξοικονόμησης ενέργειας. Πάντως γενικότερα οι απαιτήσεις μνήμης εξαρτώνται ιδιαίτερα από την εφαρμογή.

2.1.4 Συσσκευή επικοινωνίας

Μέσο Μετάδοσης

Η συσκευή επικοινωνίας (communication device) είναι απαραίτητη για την ανταλλαγή δεδομένων μεταξύ των κόμβων. Σε κάποιες περιπτώσεις (λίγες) η ενσύρματη

επικοινωνία μπορεί να προτιμάται. Σε αυτή τη περίπτωση η σύσταση της συσκευής επικοινωνίας είναι τετριμμένη.

Ο ασύρματος τρόπος επικοινωνίας είναι σαφώς πιο ενδιαφέρον. Η πρώτη επιλογή που πρέπει να κάνουμε σε αυτή τη περίπτωση είναι αυτή του ασύρματου μέσου. Η επιλογή αυτή περιλαμβάνει τις ράδιο-συχνότητες, ή την οπτική επικοινωνία ή τον υπέρηχο. Άλλα μέσα όπως μαγνητική επαγωγή χρησιμοποιούνται σε πολύ ειδικές περιπτώσεις. Από τις παραπάνω επιλογές η επικοινωνία με ράδιο-συχνότητες (RF) ταιριάζει περισσότερο σε εφαρμογές WSN. Η RF επικοινωνία είναι σε θέση να παρέχει μεγάλο εύρος εκπομπής, υψηλούς ρυθμούς δεδομένων, αποδεκτούς ρυθμούς σφαλμάτων με λογική κατανάλωση ενέργειας. Επίσης δεν απαιτεί LOS επικοινωνία μεταξύ του πομπού και του δέκτη.

Για ένα πρακτικό RF ασύρματο σύστημα, η συχνότητα του φέροντος πρέπει να επιλεγεί με μεγάλη προσοχή. Στα ασύρματα δίκτυα αισθητήρων οι συχνότητες επικοινωνίας που χρησιμοποιούνται είναι ανάμεσα στα 433 MHz και 2.4 GHz.

Transceivers

Για να επιτευχθεί η επικοινωνία και ο πομπός και ο δέκτης είναι απαραίτητος σε ένα κόμβο αισθητήρων. Το βασικό καθήκον τους είναι να μετατρέπουν μια ροή από bit που προέρχονται από τον ελεγκτή σε ράδιο-κύματα και το αντίστροφο. Για πρακτικούς λόγους είναι περισσότερο βολικό είναι να χρησιμοποιείται μία συσκευή που ενσωματώνει τις λειτουργίες και του πομπού και του δέκτη. Αυτές οι συσκευές είναι γνωστές ως *transceivers*. Συνήθως υποστηρίζεται half-duplex επικοινωνία, μιας και η ταυτόχρονη εκπομπή και λήψη δεν είναι πρακτική στις περισσότερες περιπτώσεις.

Μία μεγάλη ποικιλία φθηνών transceivers είναι εμπορικά διαθέσιμη οι οποίοι ενσωματώνουν όλα τα απαραίτητα κυκλώματα για εκπομπή και λήψη (διαμορφωτές, αποδιαμορφωτές, ενισχυτές, φίλτρα κ.α). Στη συνέχεια θα περιγράψουμε τη βασική δομή ενός transceiver.

Η βασική δομή του transceiver αποτελείται από τη RF πρόσοψη (front end) και τμήμα βασική ζώνης (baseband):

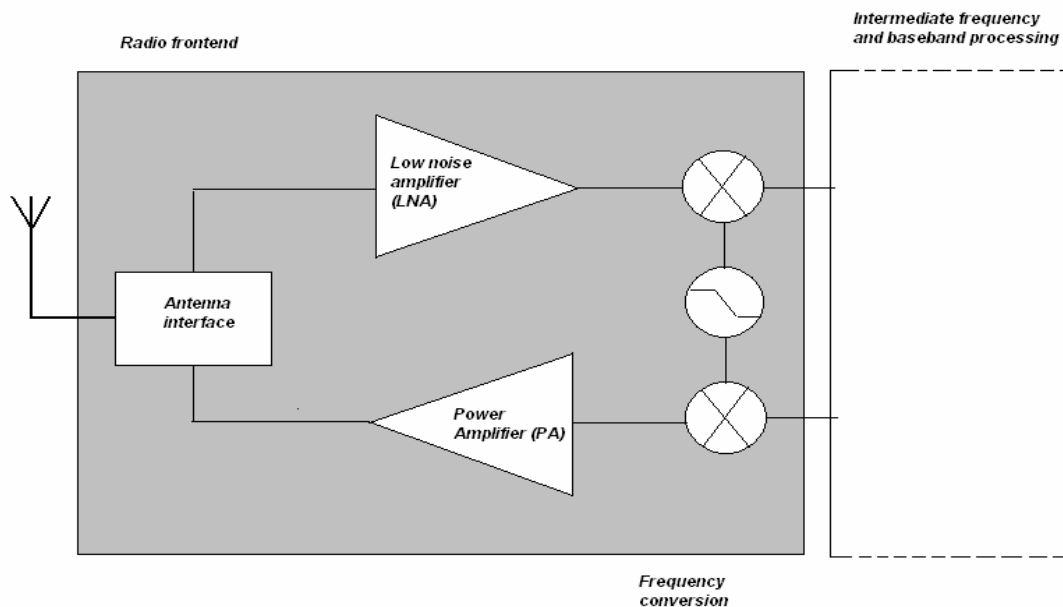
- Η RF πρόσοψη εκτελεί την επεξεργασία του αναλογικού σήματος στη ζώνη της ράδιο-συχνότητας, ενώ
- Ο βασικής ζώνης επεξεργαστής (baseband processor) εκτελεί την ανάλυση σήματος στο ψηφιακό πεδίο και επικοινωνεί με τον επεξεργαστή του κόμβου αισθητήρων ή με κάποιο άλλο ψηφιακό κύκλωμα.

Μεταξύ αυτών των μερών πραγματοποιείται μία μετατροπή συχνοτήτων είτε άμεσα, είτε μέσω μιας ή περισσότερων ενδιάμεσων συχνοτήτων. Το όριο ανάμεσα στο ψηφιακό και το αναλογικό πεδίο ορίζεται από τους (Ψηφιακό σε Αναλογικό) DAC μετατροπείς και τους (Αναλογικό σε Ψηφιακό) ADC μετατροπείς.

Η RF πρόσοψη όπως αναφέραμε εκτελεί την επεξεργασία του αναλογικού σήματος στη ζώνη της ράδιο-συχνότητας όπως για παράδειγμα στη μπάντα των 2.4 GHz. Τα βασικά συστατικά αυτού του τμήματος φαίνονται στο σχήμα 2.2 και είναι:

- Ο ενισχυτής ενέργειας (PA) από τη βασική ζώνη ή κάποια ενδιάμεση συχνότητα, ενισχύει το σήμα για να μεταδοθεί από τη κεραία.
- Ο χαμηλού θορύβου ενισχυτής (LNA) ενισχύει τα εισερχόμενα σήματα σε επίπεδα κατάλληλα για περαιτέρω επεξεργασία χωρίς να μειώσει σημαντικά το λόγο σήματος προς θόρυβο (Signal to Noise Ratio-SNR). Το εύρος της ισχύος των λαμβανόμενων σημάτων ποικίλει ανάλογα με το πόσο κοντά είναι κάθε κόμβος. Χωρίς κάποιες ενέργειες διαχείρισης, ο LNA μπορεί να είναι ανοιχτός συνεχώς και να καταναλώνει ένα μεγάλο ποσό ενέργειας.
- Στοιχεία όπως τοπικοί ή ελεγχόμενης τάσης ταλαντωτές μπορούν να χρησιμοποιηθούν για μετατροπή συχνότητας από το RF φάσμα σε ενδιάμεσες συχνότητες ή στη βασική ζώνη.

Ανάλογα με την αρχιτεκτονική της RF πρόσοψης, είναι δυνατόν να υπάρχουν και άλλα στοιχεία, όπως για παράδειγμα φίλτρα.



Σχήμα 2.2. RF Πρόσοψη.

2.1.5 Αισθητήρες

Οι αισθητήρες (sensors) μπορεί να χωριστούν σε τρεις κύριες κατηγορίες:

1. **Παθητικοί μη Κατευθυντικοί Αισθητήρες:** Αυτοί οι αισθητήρες είναι σε θέση να μετρήσουν μια φυσική ποσότητα χωρίς να «εμβαθύνουν» στη μελέτη του περιβάλλοντος και έτσι από αυτήν την άποψη είναι παθητικοί. Κάποιοι από αυτούς τους αισθητήρες είναι επίσης αυτό-τροφοδοτούμενοι με την έννοια ότι αντλούν την ενέργεια που χρειάζονται από το περιβάλλον. Σε αυτές

τις μετρήσεις των κόμβων αυτών δεν υπάρχει η ιδέα της «κατεύθυνσης». Τυπικά παραδείγματα τέτοιων αισθητήρων αποτελούν τα θερμόμετρα, οι αισθητήρες φωτός, τα μικρόφωνα, οι χημικοί αισθητήρες που είναι ευαίσθητοι σε συγκεκριμένες ουσίες, οι ανιχνευτές καπνού κ.τ.λ.

2. **Παθητικοί Στενής Δέσμης Αισθητήρες:** Αυτοί οι αισθητήρες είναι παθητικοί, όμως σε αντίθεση με τη προηγούμενη κατηγορία έχουν την αίσθηση της κατεύθυνσης στις μετρήσεις τους. Ένα τυπικό παράδειγμα είναι η κάμερα η οποία «παίρνει μετρήσεις» με βάση μια συγκεκριμένη κατεύθυνση, αλλά μπορεί και να περιστραφεί αν αυτό είναι απαραίτητο.
3. **Ενεργοί Αισθητήρες:** Η τελευταία κατηγορία αισθητήρων μελετάει ενεργά το περιβάλλον όπως για παράδειγμα κάποιοι τύποι αισθητήρων σεισμού που παράγουν κύματα δονήσεων με τη βοήθεια μικρών εκρήξεων. Τέτοιοι αισθητήρες είναι περισσότερο πολύπλοκοι.

Πρακτικά οι αισθητήρες όλων των παραπάνω τύπων είναι διαθέσιμοι σε πολλές διαφορετικές μορφές και παρουσιάζουν ιδιαιτερότητες. Κάποια προφανή trade-offs είναι η ακρίβεια αίσθησης, η κατανάλωση ενέργειας, το κόστος και το μέγεθος.

Γενικότερα η περισσότερη θεωρητική δουλειά σε WSNs θεωρεί παθητικούς μη κατευθυντικούς αισθητήρες. Μία υπόθεση που υπάρχει γενικότερα στην υπάρχουσα βιβλιογραφία είναι ότι κάθε κόμβος έχει *μία περιοχή κάλυψης* όπου μπορεί αξιόπιστα και με ακρίβεια να αναφέρει την πληροφορία που ανίχνευσε. Αν και αυτή η υπόθεση μπορεί να θεωρηθεί υπερβολική, μπορεί να είναι και αρκετά πρακτική. Είναι πολλές φορές δυνατόν να υποθέσουμε με βάση τις γνώσεις για κάποια εφαρμογή, ότι κάποιες φυσικές ποσότητες μεταβάλλονται γρηγορότερα ή λιγότερο γρήγορα με την απόσταση. Για παράδειγμα η θερμοκρασία ή η πίεση είναι αδύνατον να είναι διαφορετικές σε απόσταση μερικών μέτρων. Έτσι επιτρέποντας κάποιες αναπόφευκτες ανακρίβειες κατά τις μετρήσεις μπορούμε να θεωρήσουμε μια *ακτίνα κάλυψης* μέσα στην οποία οι τιμές του κόμβου αισθητήρων θεωρούνται έγκαιρες.

2.1.6 Τροφοδοσία ενέργειας για τους κόμβους αισθητήρων.

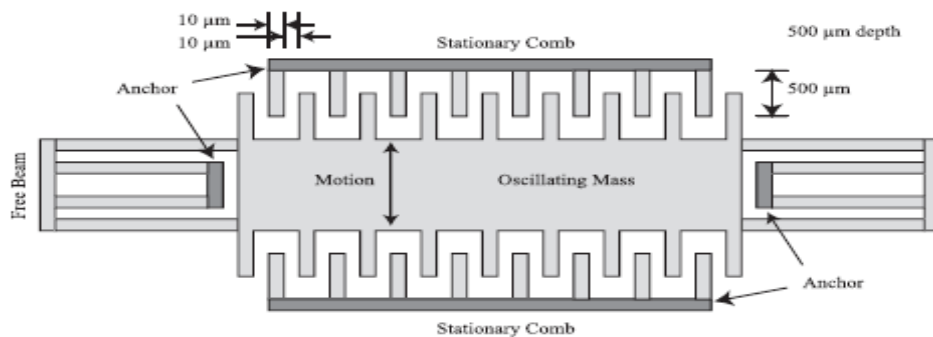
Η τροφοδοσία για τα WSNs είναι ιδιαίτερα σημαντική και έχει δύο διαφορετικές όψεις. Η πρώτη αναφέρεται στην αποθήκευση και στη παροχή ενέργειας στην κατάλληλη μορφή. Η δεύτερη αναφέρεται στο τρόπο που μπορεί να γίνει αναπλήρωση της ενέργειας που έχει καταναλωθεί χρησιμοποιώντας μία εξωτερική σε σχέση με το κόμβο πηγή ενέργειας.

Η αποθήκευση της ενέργειας γίνεται συνήθως με τη χρήση μπαταριών. Όμως η κατασκευή μπαταριών είναι ξεχωριστό κομμάτι της επιστήμης και βιομηχανίας και έτσι δεν θα ασχοληθούμε περισσότερο.

Καθώς οι απαιτήσεις των κόμβων σε ενέργεια μειώνονται γίνεται δυνατόν οι κόμβοι να μπορούν να αναπληρώσουν την ενέργεια που καταναλώνουν αντλώντας ενέργεια από το περιβάλλον. Ήδη πολλά σχήματα έχουν προταθεί για να μετατρέπουν την ενέργεια του περιβάλλοντος σε ηλεκτρική ενέργεια. Η ενέργεια που θα

συσσωρεύεται με αυτό τον τρόπο, μπορεί να χρησιμοποιείται και να αποθηκεύεται στα ηλεκτρικά κυκλώματα του κόμβου, και έτσι δεν θα υπάρχει πλέον η ανάγκη για χρήση μπαταριών. Κάποιες από τις πιο δημοφιλείς πηγές ενέργειας από το περιβάλλον περιλαμβάνουν την ηλιακή ενέργεια, RF, και τις μηχανικές δονήσεις [4].

Όπως αναφέραμε ένας τρόπος παραγωγής ενέργειας είναι οι μηχανικές δονήσεις. Η διαθέσιμη ενέργεια που μπορεί να παραχθεί εξαρτάται από το πλάτος και την συχνότητα της δόνησης και κυμαίνεται από $0.1 \mu\text{W}/\text{cm}^3$ έως $10,000 \mu\text{W}/\text{cm}^3$ σε κάποιες οριακές περιπτώσεις. Η μετατροπή της ενέργειας από τις δονήσεις σε ηλεκτρική, βασίζεται σε ήλεκτρο-μαγνητικές, ήλεκτρο-στατικές και πιεζοηλεκτρικές αρχές. Στο σχήμα 2.3 φαίνεται ένα παράδειγμα μιας τέτοια γεννήτριας με μεταβλητό πυκνωτή ([5]). Με την απαραίτητη ρύθμιση αυτή η συσκευή μπορεί να παράγει $10\mu\text{W}$ ενέργειας.



Σχήμα 2.3. Μετατροπείας δονήσεων σε ηλεκτρική ενέργεια.

2.2 ΚΑΤΑΝΑΛΩΣΗ ΕΝΕΡΓΕΙΑΣ ΚΟΜΒΟΥ.

Ο ασύρματος κόμβος αισθητήρων αφού είναι μία μικρό-ηλεκτρονική συσκευή μπορεί να εφοδιάζεται με περιορισμένη ποσότητα ενέργειας ($<0.5 \text{ Ah}$, 1.2 V). Σε κάποια σενάρια εφαρμογών, η αναπλήρωση των πηγών ενέργειας όπως έχουμε αναφέρει είναι αδύνατη. Έτσι ο χρόνος ζωής των κόμβων εξαρτάται άμεσα από τη μπαταρία. Σε ένα multihop δίκτυο αισθητήρων κάθε κόμβος έχει διπλό ρόλο, ως ανιχνευτής δεδομένων αλλά και ως δρομολογητής δεδομένων. Η δυσλειτουργία κάποιων κόμβων λοιπόν, μπορεί να προκαλέσει σημαντικές τοπολογικές αλλαγές και μπορεί να οδηγήσει σε επανά-δρομολόγηση των πακέτων και σε αναδιοργάνωση του δικτύου. Έτσι λοιπόν η διατήρηση και η διαχείριση της ενέργειας αποκτούν μεγαλύτερη βαρύτητα. Βασικός λοιπόν στόχος των ερευνητών είναι ο σχεδιασμός ενεργειακά αποδοτικών αλγορίθμων.

Τα βασικά καθήκοντα ενός κόμβου αισθητήρων είναι να ανιχνεύει γεγονότα, να εφαρμόζει μια τοπική γρήγορη επεξεργασία αυτών και να τα μεταδίδει. Άρα λοιπόν θα μπορούσαμε να κατατάξουμε τους τρόπους κατανάλωσης ενέργειας σε τρεις κατηγορίες: 1) *Ανίχνευση Γεγονότων* 2) *Επικοινωνία* 3) *Επεξεργασία Δεδομένων*.

Η ισχύς που καταναλώνεται για την ανίχνευση γεγονότων ποικίλλει ανάλογα με την εφαρμογή. Η σποραδική ανίχνευση, καταναλώνει λιγότερη ενέργεια από την συνεχή παρακολούθηση κάποιου φαινομένου. Η πολυπλοκότητα επίσης ανίχνευσης παίζει ένα

σημαντικό ρόλο για την κατανάλωση ενέργειας. Στις παρακάτω παραγράφους θα αναλύσουμε την κατανάλωση ενέργειας για επικοινωνία και επεξεργασία δεδομένων.

2.2.1 Επικοινωνία

Ένας κόμβος καταναλώνει μεγάλα ποσά ενέργειας για επικοινωνία. Στην επικοινωνία συμπεριλαμβάνεται και η εκπομπή και η λήψη δεδομένων. Μπορεί να αποδειχτεί ότι για μικρού εύρους μετάδοση με χαμηλή ενέργεια ακτινοβολίας (~ 0 dbm), η κατανάλωση ενέργειας είναι η ίδια και για την εκπομπή και για τη λήψη. Οι συνθέτες συχνότητας, οι ταλαντωτές ελέγχου τάσης, τα PLLs και οι ενισχυτές καταναλώνουν μεγάλα ποσά ενέργειας. Είναι πολύ σημαντικό στον υπολογισμό που κάνουμε για την ενέργεια να υπολογίσουμε και την ενέργεια «εκκίνησης (start-up)» του κόμβου η οποία αν και καταναλώνεται για μερικές εκατοντάδες μsec δεν είναι καθόλου αμελητέα. Η start-up ενέργεια μπορεί κυρίως να αποδοθεί στο χρόνο κλειδώματος του PLL κυκλώματος. Όσο το μέγεθος του μεταδιδόμενου πακέτου μειώνεται η start-up κατανάλωση ενέργειας γίνεται μεγαλύτερη από την ενέργεια που καταναλώνεται όσο ο κόμβος είναι ενεργός. Έτσι λοιπόν σε αυτή τη περίπτωση είναι αναποδοτικό να εναλλάσσουμε τον κόμβο μεταξύ καταστάσεων ON και OFF, επειδή με αυτό τον τρόπο θα καταναλώνεται περισσότερη ενέργεια, από το να είχαμε το κόμβο συνέχεια στο ON.

Στο [6] υπάρχει ένα τύπος για την κατανάλωση ενέργειας (P_c) για επικοινωνία με ράδιο κύματα όπου:

$$P_c = N_T [P_T (T_{on} + T_{st}) + P_{out} (T_{on})] + N_R [P_R (R_{on} + R_{st})]$$

Στο τύπο, $P_{T/R}$ είναι η ενέργεια που καταναλώνεται από το πομπό/δέκτη ενώ P_{out} είναι η ενέργεια εξόδου του πομπού. Επίσης T/R_{on} είναι ο χρόνος του πομπού/δέκτη που είναι ON και T/R_{st} είναι ο start-up χρόνος του πομπού/δέκτη. Η ποσότητα $N_{T/R}$, δείχνει πόσες φορές ο πομπός/δέκτης μεταβαίνει στην κατάσταση ON στη μονάδα του χρόνου, γεγονός που εξαρτάται από την εφαρμογή και το MAC. Η ποσότητα T_{on} μπορεί να γραφτεί ως το λόγο L/R όπου L είναι το μέγεθος του πακέτου, και R είναι ο ρυθμός των δεδομένων. Οι σημερινοί χαμηλής ενέργειας transceivers έχουν P_T και P_R περίπου 20 dbm και P_{out} κοντά στα 0 dbm. Στόχος των ερευνητών είναι το P_c να γίνει -20 dbm.

2.2.2 Επεξεργασία δεδομένων

Η κατανάλωση ενέργειας για την επεξεργασία δεδομένων, είναι πολύ μικρότερη συγκριτικά με αυτή που καταναλώνεται στην επικοινωνία. Για να καταλάβουμε τη σύγκριση ας υποθέσουμε τώρα ότι έχουμε εξασθένηση Rayleigh και τέταρτης δύναμης απώλεια ενέργειας με την απόσταση. Τότε το ενεργειακό κόστος για μετάδοση πληροφορίας 1 KB για μια απόσταση 100 m, είναι περίπου το ίδιο με το να εκτελέσουμε 3.000.000 εντολές σε ένα επεξεργαστή των 100.000.000 εντολών το δευτερόλεπτο.

Όπως έχουμε αναφέρει ένας κόμβος πρέπει να διαθέτει υπολογιστικές δυνατότητες και να μπορεί να αλληλεπιδρά με τους γύρω του. Οι διάφοροι περιορισμοί στο μέγεθος και το κόστος μας έχουν οδηγήσει στο να επιλέξουμε τεχνολογία CMOS για την κατασκευή των μικρό-επεξεργαστών του κόμβου. Δυστυχώς όμως αυτή η επιλογή επιβάλλει κάποιους περιορισμούς στην αποδοτικότητα ενέργειας. Ένα CMOS τρανζίστορ καταναλώνει ενέργεια κάθε φορά που μπαίνει σε λειτουργία (switching). Αυτή η ενέργεια είναι ανάλογη με την συχνότητα που γίνεται αυτό (switching), με τη χωρητικότητα της συσκευής και με το τετράγωνο της συχνότητας εναλλαγής τάσεων. Μειώνοντας λοιπόν τη τροφοδοσία τάσης, μπορούμε αποδοτικά να μειώσουμε τη κατανάλωση ενέργειας. Η δυναμική κλιμάκωση της τάσης, έχει ως στόχο να προσαρμόζει τη τροφοδοσία ενέργειας του επεξεργαστή και τη συχνότητα λειτουργίας με το φόρτο εργασίας του κόμβου. Όταν μικρό-επεξεργαστής διαχειρίζεται αποτελεσματικά το χρονικά μεταβαλλόμενο υπολογιστικό φόρτο, μειώνοντας τη συχνότητα λειτουργίας σε περιόδους αδράνειας, αυτό έχει σαν αποτέλεσμα μία γραμμική μείωση της κατανάλωσης ενέργειας. Όταν όμως μειώνει τη τάση λειτουργίας έχουμε σαν αποτέλεσμα τετραγωνική μείωση της ενέργειας. Βέβαια από την άλλη πλευρά αυτό μπορεί να έχει ως αποτέλεσμα ο επεξεργαστής να μη μπορεί να δουλεύει στο μέγιστο ρυθμό που έχει τη δυνατότητα. Πολύ σημαντικά ποσά ενέργειας μπορεί να εξοικονομηθούν αν αναγνωριστεί ότι η λειτουργία του επεξεργαστή στη μέγιστη απόδοση, δεν είναι πάντα επιθυμητή και ότι η τάση και η συχνότητα λειτουργίας του επεξεργαστή πρέπει να προσαρμόζονται στις τρέχουσες απαιτήσεις.

Γενικότερα, η κατανάλωση ενέργειας κατά την επεξεργασία δεδομένων (P_p) μπορεί να εκφραστεί ως:

$$P_p = CV_{dd}^2 f + V_{dd} I_0 e^{V_{dd}/nV_T}$$

όπου C είναι η χωρητικότητα, V_{dd} είναι η αλλαγή τάσης και f είναι η συχνότητα του switching.

Πρέπει να σημειώσουμε τέλος, ότι υπάρχουν επιπλέον κυκλώματα σε κάθε κόμβο για κωδικοποίηση και αποκωδικοποίηση δεδομένων. Επίσης κάποια επιπλέον κυκλώματα που είναι απαραίτητα για συγκεκριμένες εφαρμογές είναι πιθανόν να χρησιμοποιηθούν. Πρέπει λοιπόν να είμαστε ιδιαίτερα προσεκτικοί στο σχεδιασμό πρωτοκόλλων και αλγορίθμων στο θέμα της ενέργειας.

2.3 ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΠΕΡΙΒΑΛΛΟΝΤΑ ΓΙΑ WSNs

2.3.1 Ενσωματωμένα λειτουργικά συστήματα

Παραδοσιακά η δουλειά ενός λειτουργικού συστήματος είναι ο έλεγχος και η προστασία των πόρων του συστήματος (συμπεριλαμβάνοντας λειτουργίες εισόδου εξόδου), η σωστή διαχείριση τους από πολλούς χρήστες καθώς επίσης η υποστήριξη ταυτόχρονης εκτέλεσης πολλαπλών διεργασιών και η επικοινωνία μεταξύ τους. Όμως όλες αυτές οι

λειτουργικότητες είναι μερικώς χρήσιμες σε ένα ενσωματωμένο σύστημα μιας και ο κώδικας προς εκτέλεση είναι πολύ πιο περιορισμένος από αυτόν ενός συστήματος γενικού σκοπού. Άλλωστε με βάση αυτά που αναφέραμε παραπάνω, αυτά τα συστήματα δεν έχουν τους απαιτούμενους πόρους για να υποστηρίξουν ένα «πλήρες» λειτουργικό σύστημα.

Ένα λειτουργικό σύστημα ή ένα περιβάλλον εργασίας για τα WSNs πρέπει να υποστηρίζει τις συγκεκριμένες ανάγκες του συστήματος. Συγκεκριμένα η ανάγκη για εξοικονόμηση ενέργειας απαιτεί λειτουργικότητα για διαχείριση ενέργειας, όπως για παράδειγμα ελεγχόμενη απενεργοποίηση κάποιων συστατικών του συστήματος όταν αυτά δεν είναι απαραίτητα, ή τεχνικές δυναμικής κλιμάκωσης της τάσης (DVS). Επίσης κάποια περιφερειακά (π.χ. αισθητήρες) πρέπει να ελέγχονται εύκολα και αποδοτικά από το λειτουργικό σύστημα.

Όλα τα παραπάνω φανερώνουν την απαίτηση για ένα κατάλληλο προγραμματιστικό μοντέλο, για ένα σαφή τρόπο δόμησης μιας στοίβας πρωτοκόλλου και για την ανάπτυξη τεχνικών για αποτελεσματική διαχείριση ενέργειας. Στην επόμενη παράγραφο παρουσιάζουμε το *TinyOS*, το δημοφιλέστερο λειτουργικό σύστημα για ασύρματα δίκτυα αισθητήρων και τη γλώσσα προγραμματισμού *nesC* στην οποία βασίζεται.

2.3.2 TinyOS και nesC

Γενικότερα το *TinyOS* είναι ένα λειτουργικό σύστημα για ενσωματωμένα (embedded) συστήματα και είναι γραμμένο στη γλώσσα προγραμματισμού *nesC* σαν ένα σύνολο από επιμέρους τμήματα και διεργασίες που είναι σε θέση να συνεργάζονται. Εκτός από *nesC*, κάποια κομμάτια του *TinyOS* είναι γραμμένα και σε C (κυρίως εργαλεία και βιβλιοθήκες) καθώς και Java. Ο βασικός σχεδιαστικός στόχος αυτού του λειτουργικού συστήματος ήταν να μπορεί να ενσωματωθεί στους κόμβους, και να μπορεί να λειτουργήσει με τους σημαντικούς περιορισμούς μνήμης που διακρίνουν τους κόμβους αισθητήρων. Πριν κλείσουμε αυτή την εισαγωγική παράγραφο αξίζει να σημειώσουμε ότι το *TinyOS* είναι προϊόν συνεργασίας του πανεπιστημίου Berkeley μαζί με το ερευνητικό τμήμα της Intel και ότι τώρα αναπτύσσονται οι σειρές 2.x αυτού.

Το *TinyOS* αναφέρεται κατ' αρχήν σε προγραμματισμό βασισμένο στα γεγονότα (*event based programming*), μιας και η ιδέα λειτουργίας των WSNs είναι να δρουν διαφορετικά στα γεγονότα που συμβαίνουν στο περιβάλλον. Το σύστημα σε αυτή τη περίπτωση περιμένει για κάποιο γεγονός να συμβεί, όπου ένα γεγονός μπορεί να είναι κάποια διαθέσιμα δεδομένα από κάποιο αισθητήρα, η άφιξη ενός πακέτου ή η λήξη ενός χρονομέτρου. Στη συνέχεια ένα τέτοιο γεγονός υπόκειται σε επεξεργασία από μία μικρή ακολουθία εντολών, η οποία αποθηκεύει ότι πράγματι έχει συμβεί κάποιο γεγονός, καθώς και κάποιες επιπλέον πληροφορίες όπως για παράδειγμα οι τιμές που ανίχνευσε ο αισθητήρας κ.τ.λ. Πάντως η πραγματική επεξεργασία της πληροφορίας δεν γίνεται σε αυτές τις ρουτίνες που διαχειρίζονται την εμφάνιση του γεγονότος.

Το *TinyOS* εκτός από το προγραμματισμό βασισμένο στα γεγονότα, υποστηρίζει και τμηματικότητα (*modularity*) υπό την έννοια ότι αποτελείται από πολλά συστατικά

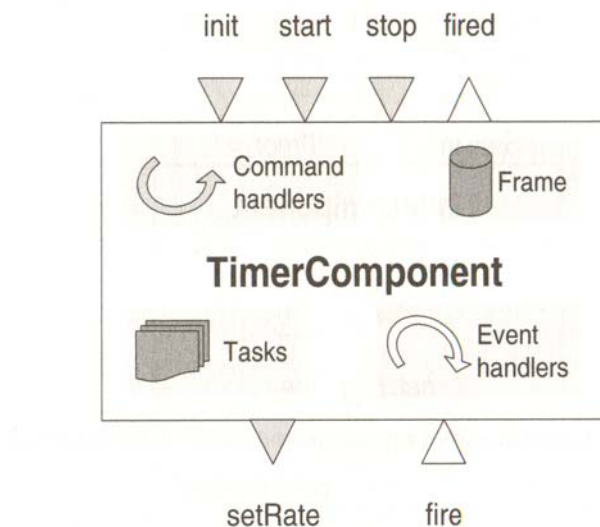
στοιχεία (components). Κάθε συστατικό στοιχείο περιέχει και αντίστοιχη λειτουργικότητα, όπως για παράδειγμα μπορεί να διαχειρίζεται τις ράδιο-παρεμβολές, ή να υπολογίζει τις διαδρομές για τη δρομολόγηση πακέτων. Ένα συστατικό στοιχείο αποτελείται από ένα πλαίσιο (*frame*) που αποθηκεύει τη κατάσταση της πληροφορίας, το κώδικα που υλοποιεί τις απαραίτητες εργασίες (*tasks*), χειριστές για τα γεγονότα (*events*) και εντολές (*commands*). Τα γεγονότα αλλά και οι εντολές ανταλλάσσονται μεταξύ των συστατικών στοιχείων. Επίσης τα συστατικά στοιχεία ταξινομούνται ιεραρχικά, από τα χαμηλού επιπέδου συστατικά που βρίσκονται σε επίπεδο hardware μέχρι αυτά σε επίπεδο εφαρμογών. Τα γεγονότα ξεκινούν από το επίπεδο του hardware και περνούν προς τα συστατικά υψηλότερου επιπέδου. Αντίθετα οι εντολές, περνούν από τα υψηλότερα επίπεδα στα χαμηλότερα.

Το σχήμα 2.4 απεικονίζει πως θα ήταν ένα χρονόμετρο (component) σε μια πιο αφηρημένη έκδοση. Το χρονόμετρο αυτό λοιπόν μπορεί να κατανοήσει τρεις εντολές (“*init*”, “*start*”, “*stop*”) και μπορεί να διαχειριστεί ένα γεγονός (“*fire*”) που θα λάβει από ένα άλλο συστατικό στοιχείο. Επίσης μπορεί να εκδώσει “*setRate*” σε αυτό το συστατικό και μπορεί και το ίδιο να εκπέμψει ένα “*fired*” γεγονός.

Αυτό που πρέπει να σημειώσουμε είναι ότι το TinyOS, βασιζόμενο στην ιδέα του event-based προγραμματισμού, δεν αναθέτει τόσο σημαντικά καθήκοντα στους χειριστές εντολών (commands) και γεγονότων (events). Συγκεκριμένα οι εντολές υλοποιούν απλώς μια αίτηση για κάποια εργασία (*task*) η οποία πρέπει να εκτελεστεί από τα κατώτερα στην ιεραρχία συστατικά στοιχεία. Ομοίως ένας χειριστής γεγονότων, απλά αποθηκεύει τη κατάσταση της πληροφορίας σε ένα πλαίσιο (*frame*) και επίσης δρομολογεί μία εργασία (*task*) για να εκτελεστεί αργότερα. Μπορεί επίσης να στείλει εντολές και σε άλλα συστατικά στοιχεία ή να προωθήσει ένα γεγονός. Η ουσιαστική υπολογιστική δουλειά λοιπόν γίνεται μέσα στις εργασίες (*tasks*).

Στο TinyOS οι εργασίες (*tasks*) συνεχίζουν να τρέχουν μέχρι να ολοκληρωθούν και μπορούν να διακοπούν μόνο από κάποιους χειριστές γεγονότων. Τη «διαιτησία» ανάμεσα στις εργασίες (μπορεί να έχουν δημιουργηθεί πολλές εργασίες προς εκτέλεση, εξαιτίας των πολλαπλών γεγονότων που έχουν συμβεί) αναλαμβάνει να διεκπεραιώσει ένας ενεργειακά αποδοτικός FIFO προγραμματιστής διεργασιών ή καλύτερα *scheduler*, ο οποίος απενεργοποιεί το κόμβο του δικτύου όταν δεν εκτελείτε κάποια εργασία, ή όταν δεν υπάρχει κάποια εργασία στην ουρά.

Όπως προαναφέραμε όλες οι εργασίες (*tasks*) πρέπει να ολοκληρωθούν για να μπορούν να δώσουν σε άλλες τη θέση τους στην ουρά. Όμως δεν έχει γίνει σαφές πως ένα συστατικό στοιχείο (component) θα καταλάβει ότι ένα άλλο συστατικό στοιχείο έχει ολοκληρώσει την εργασία του. Για παράδειγμα, πώς ένα ARQ πρωτόκολλο μαθαίνει από το MAC πρωτόκολλο αν ένα πακέτο έχει σταλθεί επιτυχώς; Η ιδέα είναι να διαχωρίσουμε αυτή τη διαδικασία ενημέρωσης σε δύο φάσεις. Η 1^η φάση αναφέρεται στην αποστολή της εργασίας που πιθανό να έχει συμβεί, ενώ η 2^η είναι η σαφή πληροφόρηση για την έκβαση της εργασίας, η ο οποία αποτελεί και ξεχωριστό γεγονός. Αυτή η προσέγγιση του προγραμματισμού σε χωριστές φάσεις (***split-phase programming***) απαιτεί λοιπόν μία εντολή (ερώτηση δηλαδή για κάποιο task) και ένα αντίστοιχο γεγονός πληροφόρησης (εάν δεν απαιτείται επιβεβαίωση για κάποια εντολή, τότε δεν απαιτείται και αντίστοιχο γεγονός πληροφόρησης).



Σχήμα 2.4. Παράδειγμα Timer Component.

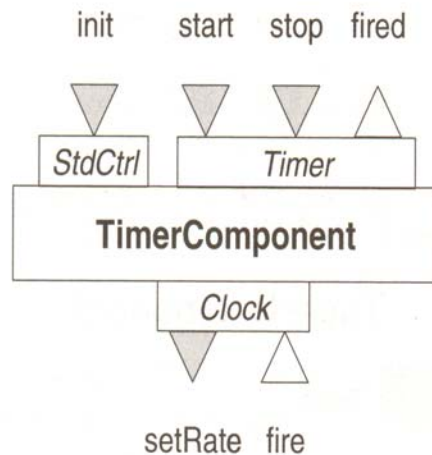
Έχοντας λοιπόν εντολές και γεγονότα ως το μοναδικό τρόπο αλληλεπίδρασης ανάμεσα σε δύο συστατικά στοιχεία, σε συνδυασμό με τη χρήση *split-phase* προγραμματισμού, προκύπτει ένας μεγάλος αριθμός γεγονότων και εντολών ακόμα και σε ένα όχι και τόσο μεγάλο πρόγραμμα. Έτσι είναι απαραίτητο να εισάγουμε *αφαιρετικότητα* στο πρόγραμμά μας για να μπορούμε να το οργανώσουμε αποδοτικά. Για να γίνουμε πιο σαφείς, το σύνολο των εντολών που ένα συστατικό στοιχείο καταλαβαίνει, και το σύνολο των γεγονότων που ένα συστατικό στοιχείο μπορεί να παράγει, αποτελούν ένα *interface* για τα άλλα συστατικά στοιχεία του υψηλότερου ιεραρχικά επιπέδου. Διαφορετικά, ένα συστατικό στοιχείο μπορεί να καλέσει εντολές από το χαμηλότερο του συστατικό στοιχείο, και επίσης μπορεί να λάβει τα γεγονότα που αυτό το χαμηλότερο συστατικό στοιχείο εκπέμπει. Με αυτό λοιπόν τον τρόπο χτίζεται η διεπαφή (*interface*) ανάμεσα σε δύο συστατικά στοιχεία.

Η γλώσσα nesC κάνει δυνατή την υλοποίηση μια τέτοιας ιδέας, αφού επιτρέπει στον προγραμματιστή να ορίσει *τύπους interface*, καθώς και εντολές και γεγονότα που ανήκουν σε αυτό. Έτσι λοιπόν η nesC επιτρέπει εύκολα να υλοποιηθεί το *split-phase* προγραμματιστικό μοντέλο, τοποθετώντας τις εντολές και τα αντίστοιχα γεγονότα στο ίδιο *interface*. Τα συστατικά στοιχεία στη συνέχεια, παρέχουν *interface* στους χρήστες τους, ενώ με τη σειρά τους χρησιμοποιούν τα *interface* των κατώτερων συστατικών στοιχείων.

Το σχήμα 2.5 δείχνει πώς το χρονόμετρο του προηγούμενου παραδείγματος μπορεί να αναδιοργανωθεί και να χρησιμοποιεί ένα *interface* ρολογιού (*clock interface*) που ουσιαστικά αποτελείται από τα *StdCtrl* και *Timer* interfaces. Ο αντίστοιχος nesC κώδικας φαίνεται στο πίνακα 2.1. Να σημειώσουμε εδώ ότι το συστατικό στοιχείο *TimerComponent* ορίζεται ως *module* αφού αποτελεί ένα πρωτογενές (*primitive*) συστατικό στοιχείο που άμεσα περιέχει εργασίες (*tasks*) και διαχειριστές (*handlers*).

Τέτοια πρωτογενή συστατικά στοιχεία και *modules*, μπορούν να συνδυαστούν σε μεγαλύτερες διαμορφώσεις (*configurations*) «καλωδιώνοντας» (συνδέοντας) τα

κατάλληλα interfaces μεταξύ τους. Για να μπορεί να γίνει αυτή η σύνδεση, μόνο τα συστατικά στοιχεία που έχουν τους σωστούς τύπους interface μπορούν να ενωθούν (αυτός ο έλεγχος γίνεται από τον compiler). Το σχήμα 2.6 δείχνει πως ο `TimerComponent` και ένα επιπλέον συστατικό στοιχείο το `HWClock`, μπορούν να συνδεθούν και να σχηματίσουν ένα καινούργιο συστατικό στοιχείο (`CompleteTimer`), εκθέτοντας μόνο προς τα έξω τα `StdCtrl` και `Timer` interfaces. Επίσης ο πίνακας 2.2 δείχνει τον αντίστοιχο nesC κώδικα. Πρέπει να σημειώσουμε εδώ ότι και τα modules και τα *configurations* αποτελούν συστατικά στοιχεία.



Σχήμα 2.5. Timer Component με χρήση Interfaces

Χρησιμοποιώντας τους ορισμούς, τις υλοποιήσεις και τις συνδέσεις των συστατικών στοιχείων (*components*), το TinyOS και η nesC αποτελούν μία πολύ ισχυρή και εύχρηστη βάση για την υλοποίηση των βασικών λειτουργιών ενός λειτουργικού συστήματος, καθώς και για την υλοποίηση stacks πρωτοκόλλων επικοινωνίας και εφαρμογών. Η εμπειρία έχει δείξει μέχρι τώρα ότι πράγματι οι προγραμματιστές μπορούν εύκολα να σχεδιάσουν επιμέρους συστατικά στοιχεία (*components*), τα οποία μπορούν να συνδυαστούν και να ικανοποιήσουν διαφορετικές λειτουργικότητες. Έτσι υλοποιείται και η ιδέα της τμηματικότητας (*modularity*) που αναφέραμε στην αρχή της παραγράφου. Τέλος για όλα τα παραπάνω, το μέγεθος του κώδικα αλλά και της απαιτούμενης μνήμης διατηρείται αρκετά μικρό.

Γενικά το TinyOS μπορεί να θεωρηθεί ως η πρότυπη πλατφόρμα υλοποίησης για τα WSNs. Μάλιστα έχει αρχίσει να γίνεται διαθέσιμο και μπορεί να ενσωματωθεί σε ένα αυξανόμενο αριθμό από κόμβους «*motes*» πέρα από αυτά που προοριζόταν αρχικά. Τέλος, έχουν αναπτυχθεί ένα πλήθος επεκτάσεων, εφαρμογών και πρωτοκόλλων για τους κόμβους αισθητήρων.

Πίνακας 2.1. Ορισμός Modules και Interfaces

```

interface StdCtrl {
    command result_t init();
}

interface Timer {
    command result_t start (char type, uint32_t interval);
    command result_t stop ();
    event result_t fired();
}

interface Clock {
    command result_t setRate (char interval, char scale);
    event result_t fire ();
}

module TimerComponent {
    provides {
        interface StdCtrl;
        interface Timer;
    }
    uses interface Clock as Clk;
}

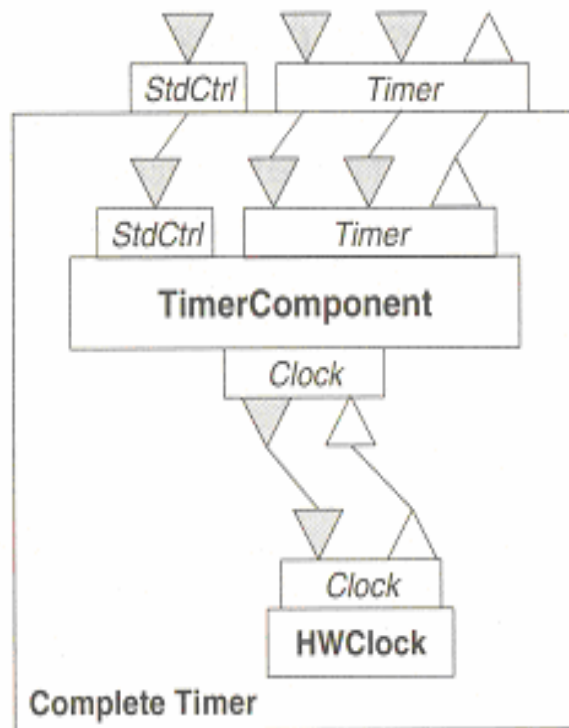
```

Πίνακας 2.2. *Configuration* ως συνδυασμός συστατικών στοιχείων.

```

configuration CompleteTimer {
    provides {
        interface StdCtrl;
        interface Timer;
    }
    implementation {
        components TimerComponent, HWClock;
        StdCtrl = TimerComponent.HWClock;
        Timer = TimerComponent.Timer;
        TimerComponent.Clk = HWClock.Clock;
    }
}

```



Σχήμα 2.6. Ένα μεγαλύτερο *configuration* ως συνδυασμό δύο συστατικών στοιχείων.

2.4 ΠΑΡΑΔΕΙΓΜΑΤΑ ΚΟΜΒΩΝ ΑΙΣΘΗΤΗΡΩΝ

Ήδη είναι διαθέσιμος ένας αρκετά μεγάλος αριθμός πραγματικών κόμβων αισθητήρων για χρήση τόσο στην έρευνα όσο και σε εφαρμογές. Πάντα εξαρτώμενοι από τις απαιτήσεις της συγκεκριμένης εφαρμογής, οι κόμβοι αυτοί διαφοροποιούνται ως προς την ενέργεια που διαθέτουν, το μέγεθος κ.τ.λ. Γενικά οι κόμβοι αυτοί είναι πολύ μικροί και παρομοιάζονται με μικρούς κόκκους σκόνης. Για αυτό και συχνά αναφέρονται ως *motes*. Σε αυτή παράγραφο θα αναφέρουμε κάποιους δημοφιλής κόμβους.

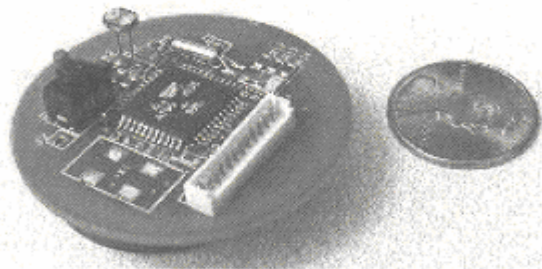
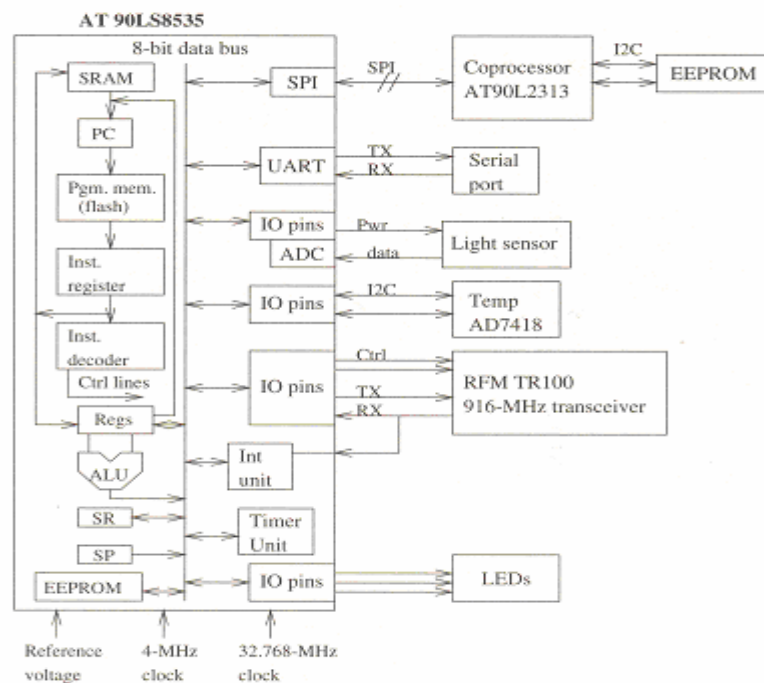
2.4.1 Η οικογένεια “Mica Mote”

Ξεκινώντας στα τέλη της δεκαετίας του 1990, η οικογένεια αυτή των κόμβων προέκυψε από ερευνά του πανεπιστημίου της Καλιφόρνια Berkeley σε συνδυασμό με το ερευνητικό τμήμα της Intel. Οι κόμβοι αυτοί είναι ευρέως γνωστοί ως *Mica motes*, και υπάρχουν αρκετές διαφορετικές εκδόσεις όπως οι Mica, Mica2, Mica2Dot. Οι κόμβοι αυτοί είναι εμπορικά διαθέσιμοι από την εταιρία Crossbow (<http://www.xbow.com>) σε

όλες τις διαφορετικές εκδόσεις τους. Το σύνθηες λειτουργικό σύστημα για τέτοιες πλατφόρμες είναι το TinyOS.

Ένα παράδειγμα για το πώς σχηματικά είναι ένας τέτοιος κόμβος, καθώς και μια φωτογραφία (σε σύγκριση με το μέγεθος ενός κέρματος) του φαίνονται στο σχήμα 2.7. Βασικά συστατικά στοιχεία του κόμβου είναι ο μικρό-ελεγκτής της οικογένειας Atmel, ένα απλό ράδιο-modem (RFM TR 1000) και αρκετές συνδέσεις. Είναι επίσης δυνατόν στη κύρια πλακέτα του κόμβου να συνδέσουμε επιπλέον αισθητήρες, όπως για παράδειγμα αισθητήρες υγρασίας, και έτσι ο κόμβος να μπορεί να απευθύνεται σε περισσότερες εφαρμογές. Τέλος να σημειώσουμε ότι οι κόμβοι στον ελεγκτή μέσω I2C διάυλο ή μέσω SPI ανάλογα την έκδοση.

Μία άλλη οικογένεια οι MEDUSA-II κόμβοι, μοιράζονται τα ίδια συστατικά στοιχεία και έχουν περίπου τον ίδιο σχεδιασμό με τους MICA.



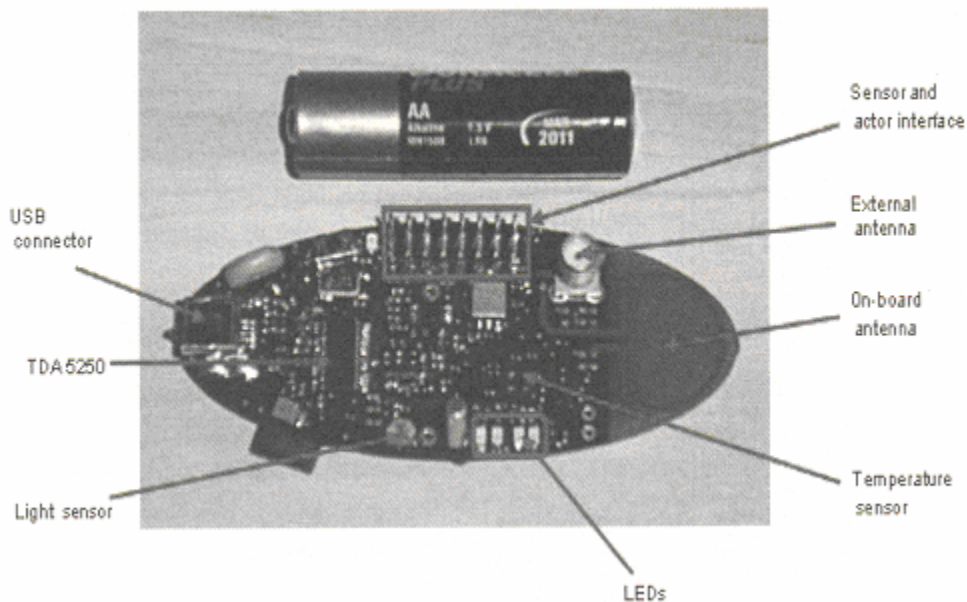
Σχήμα 2.7. Φωτογραφία και σχήμα του κόμβου Mica.

2.4.2 Κόμβοι EYES

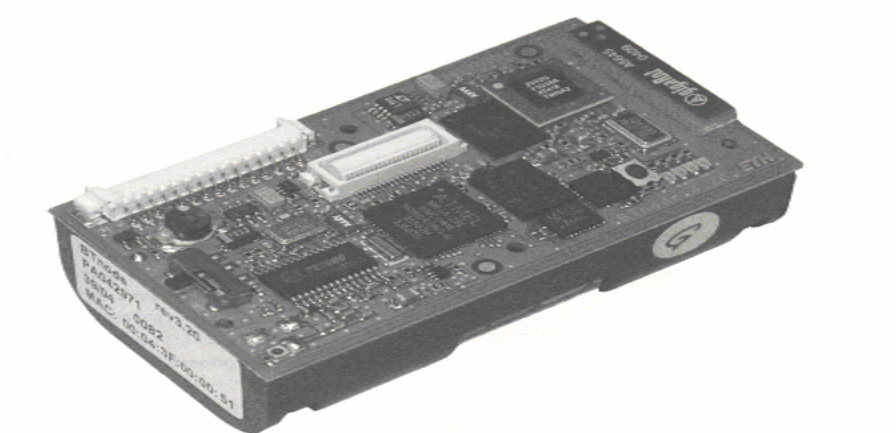
Οι κόμβοι που αναπτύχθηκαν στα πλαίσια του project EYES (<http://www.eyes.eu.org>) (*Energy Efficient Sensor Networks*) είναι επίσης ένα παράδειγμα κόμβων αισθητήρων (σχήμα 2.8). Αυτοί οι κόμβοι είναι εφοδιασμένοι με ένα *Texas Instrument MSP 430* μικρό-ελεγκτή, ένα Infineon ράδιο-modem TDA 5250 και ένα ελεγκτή για τη μεταδιδόμενη ισχύ. Ο κόμβος τέλος, έχει τη δυνατότητα να συνδεθεί με USB σε ένα PC και επίσης να ενσωματώσει επιπλέον τύπους αισθητήρων.

2.4.3 BTnodes

Οι “BTnodes” (σχήμα 2.9) είναι αποτέλεσμα πολλών ερευνητικών εργασιών που έχουν γίνει στο ETH Zurich. Αυτοί οι κόμβοι χρησιμοποιούν ATmega 128L microcontroller, 64+180 kB RAM και 128 kB μνήμη FLASH. Σε αντίθεση με τους περισσότερους κόμβους αισθητήρων (αλλά ομοίως με του κόμβους που αναπτύσσονται από την Intel), οι κόμβοι αυτοί μπορούν να χρησιμοποιήσουν τεχνολογία Bluetooth για επικοινωνία, με πομπό-δέκτες Chipcon CC1000 ανάμεσα στα 433 και 915 MHz.



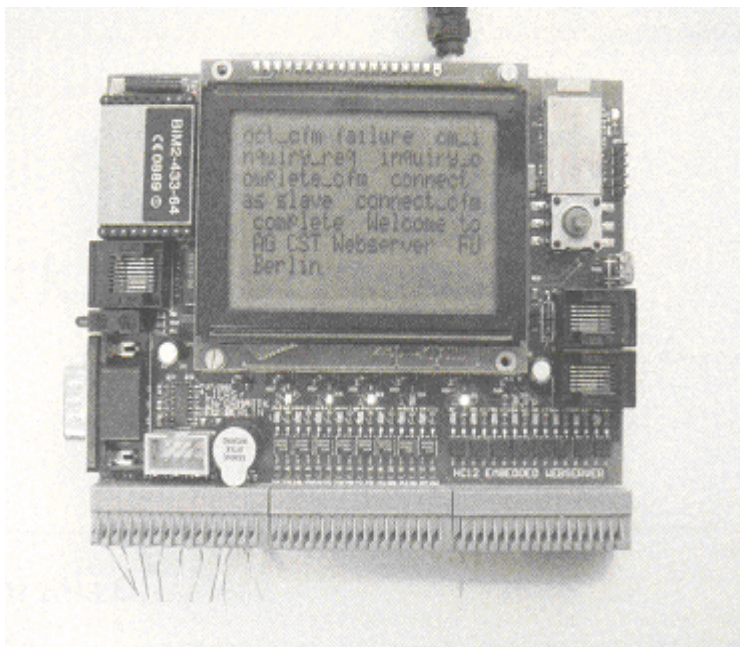
Σχήμα 2.8. EYE κόμβος αισθητήρων.



Σχήμα 2.9. BTnode.

2.4.4 Scatterweb

Η πλατφόρμα Scatterweb αναπτύχθηκε από την ομάδα Computer Systems & Telematics στο πανεπιστήμιο Freie του Βερολίνου (σχήμα 2.10). Η πλατφόρμα αυτή αποτελεί τη βάση ουσιαστικά για μία ολόκληρη οικογένεια από κόμβους, που ξεκινούν από κόμβους αισθητήρων (MSP 430 microcontroller) και φτάνουν μέχρι web servers για ενσωματωμένα συστήματα οι οποίοι και διαθέτουν ένα μεγάλο εύρος τρόπων διασύνδεσης.



Σχήμα 2.10. Ένας Scatterweb ενσωματωμένος Web Server.

3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΚΤΥΟΥ

Το κεφάλαιο αυτό μας εισάγει στις βασικές αρχές που συνεπάγονται τη μετατροπή των ανεξάρτητων κόμβων αισθητήρων σε δίκτυο αισθητήρων. Στη 1^η παράγραφο του κεφαλαίου θα αναφέρουμε τα διαφορετικά σενάρια που διαφοροποιούν τα WSNs μεταξύ τους. Στη συνέχεια θα παρουσιάσουμε τους παράγοντες που επηρεάζουν το σχεδιασμό ενός δικτύου αισθητήρων. Τέλος θα αναφερθούμε σε υπάρχοντα δικτυακά μοντέλα, και θα παρουσιάσουμε ένα καινούργιο μοντέλο με πολλαπλά κέντρα αναφοράς (multiple-sinks).

3.1 ΔΙΑΦΟΡΕΤΙΚΑ ΣΕΝΑΡΙΑ ΓΙΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ

3.1.1 Τύποι πηγών και sinks

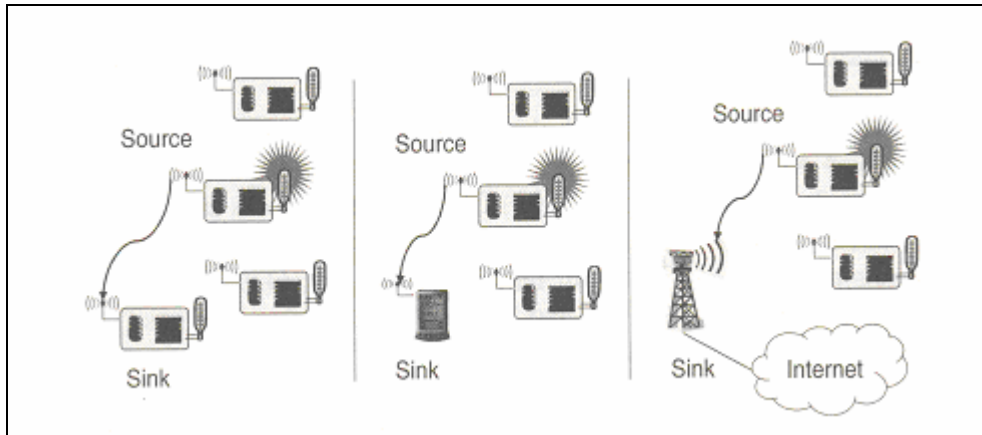
Σε προηγούμενο κεφάλαιο παρουσιάζοντας διαφορετικούς τύπους εφαρμογών, παρατηρήσαμε και διαφορετικούς τρόπους αλληλεπίδρασης μεταξύ πηγών και κέντρων ελέγχου (sinks) και έτσι είδαμε ποια είναι η έννοια της πηγής των γεγονότων (source) και των κέντρων ελέγχου (sinks). Μία πηγή (source) λοιπόν, είναι κάθε οντότητα (τυπικά κόμβος του δικτύου) που μπορεί να ανιχνεύσει και να παρέχει πληροφορίες. Επίσης μπορεί να είναι ένας *actuator* κόμβος, δηλαδή ένας κόμβος ο οποίος θα λάβει την απόφαση με βάση τα διαθέσιμα δεδομένα και θα κάνει τις κατάλληλες ενέργειες στο πεδίο ενδιαφέροντος.

Από την άλλη πλευρά ο *sink* είναι η οντότητα που απαιτεί τις πληροφορίες. Στην πραγματικότητα υπάρχουν τρεις επιλογές για ένα sink:

1. Ο sink μπορεί να ανήκει στο δίκτυο αισθητήρων (ως sensor ή actuator), ή να είναι μια οντότητα εκτός του δικτύου.
2. Ο sink μπορεί να είναι μια πραγματική συσκευή όπως ένα PDA και να αλληλεπιδρά με το δίκτυο.
3. Ο sink μπορεί να είναι μια πύλη για ένα μεγαλύτερο δίκτυο όπως το Internet, όπου ουσιαστικά η αίτηση για κάποια πληροφορία έρχεται από ένα κόμβο αρκετά απομακρυσμένο ο οποίος συνδέεται έμμεσα με το δίκτυο αισθητήρων.

Οι κύριοι αυτοί τύποι των sinks φαίνονται στο σχήμα 3.1.

Πολύ σημαντικό για τις πηγές και τους sinks όπως θα δούμε και παρακάτω είναι αν αυτοί κινούνται (mobility). Στην αρχιτεκτονική του δικτύου δεν μας ενδιαφέρει πως επεξεργάζονται οι κόμβοι την πληροφορία αλλά περισσότερο η σχετική τους θέση.

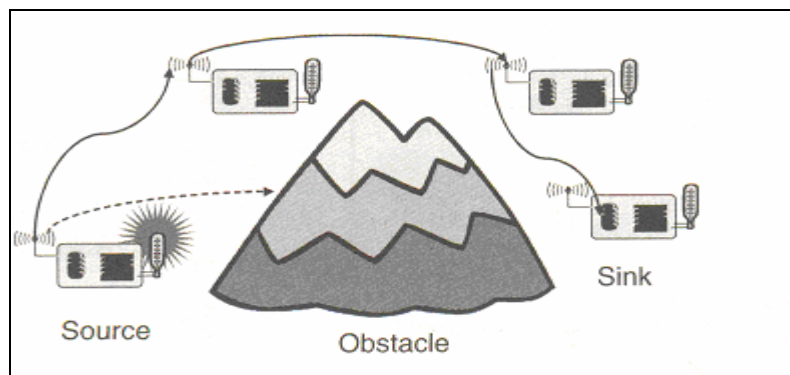


Σχήμα 3.1. Τρεις τύποι sinks σε ένα απλό single-hop δίκτυο.

3.1.2 Single-hop vs multihop δικτύων

Με βάση αυτά που ισχύουν στη ράδιο-επικοινωνία και το περιορισμό ενέργειας που αυτή επιβάλλει, εισάγεται άμεσα ένας περιορισμός στη δυνατή απόσταση μεταξύ πομπού και δέκτη. Εξαιτίας αυτών των περιορισμών στην απόσταση, δεν είναι συνήθως δυνατή η άμεση επικοινωνία μεταξύ πηγής και sink, ιδιαίτερα στα WSNs όπου προβλέπεται ότι θα καλύπτουν μεγάλη γεωγραφική απόσταση (περιβαλλοντολογικές εφαρμογές) ή ότι θα λειτουργούν σε περιβάλλοντα που μπορεί να υπάρχει μεγάλη εξασθένηση σήματος.

Ένας προφανής τρόπος για να μπορέσουμε να ξεπεράσουμε αυτό το περιορισμό της απόστασης, είναι να στέλνουμε τη πληροφορία στον sink μέσω πολλαπλών κόμβων (*multihop*). Ο τύπος αυτός λοιπόν των multihop δικτύων (σχήμα 3.2) είναι πολύ ελκυστικός για τα WSNs, όπου το ρόλο των ενδιάμεσων κόμβων έχουν οι ίδιοι οι κόμβοι αισθητήρων, χωρίς να χρειάζεται έτσι επιπλέον εξοπλισμός. Ανάλογα με την εφαρμογή, η πιθανότητα να υπάρχει ένας ενδιάμεσος κόμβος αισθητήρων στο σωστό μέρος είναι αρκετά μεγάλη (όταν για παράδειγμα έχουμε ομοιόμορφη κατανομή κόμβων), αλλά παρόλα αυτά δεν μπορεί κανείς να εγγυηθεί ότι θα υπάρχει πάντοτε κάποια διαδρομή από τη πηγή στο sink, ή ότι αυτή η διαδρομή θα είναι μικρή.



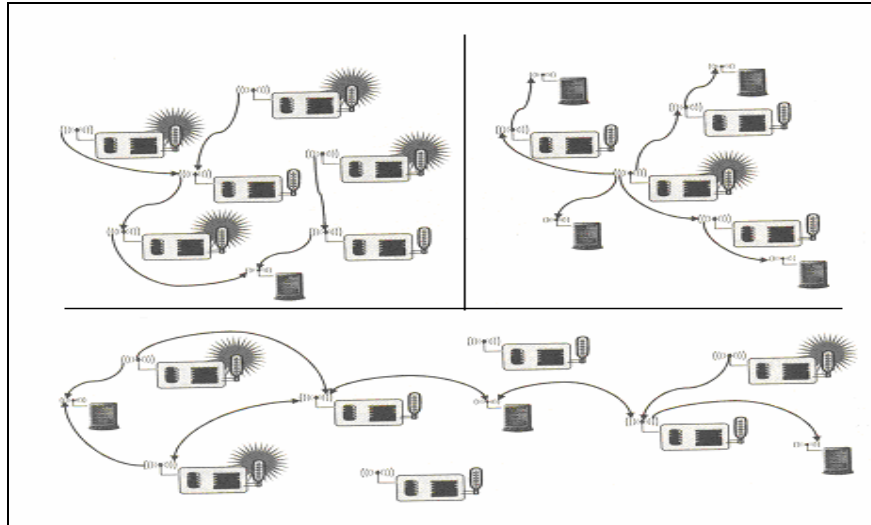
Σχήμα 3.2. Multihop Δίκτυα: Από τη στιγμή που η άμεση επικοινωνία είναι αδύνατη λόγω απόστασης ή εμποδίων, ο multihop τρόπος επικοινωνίας μπορεί να λύσει το πρόβλημα.

Ενώ η ιδέα της multihop διάδοσης είναι μία καλή τακτική διάδοσης στις περιπτώσεις που έχουμε μεγάλες αποστάσεις ή εμπόδια, έχει αποδειχτεί ότι είναι ωφέλιμη και στη κατανάλωση ενέργειας. Η ιδέα πίσω από αυτόν τον ισχυρισμό, είναι ότι η εξασθένηση των ράδιο-σημάτων είναι τουλάχιστον ανάλογη του τετραγώνου της απόστασης (συνήθως αρκετά μεγαλύτερη) στα περισσότερα περιβάλλοντα, και ότι μειώνοντας την απόσταση μετάδοσης θα εξοικονομούμαι αρκετή ενέργεια. Συγκεκριμένα, όταν στοχεύουμε σε ένα σταθερό SNR σε όλους του δέκτες (ας υποθέσουμε αμελητέους ρυθμούς σφαλμάτων), η εκπεμπόμενη ενέργεια που απαιτείται για άμεση επικοινωνία σε μια απόσταση d , είναι cd^a (όπου c είναι μια σταθερά, $a \geq 2$ ο συντελεστής απώλειας μονοπατιού). Εκπέμποντας λοιπόν σε μια απόσταση $d/2$, η ενέργεια μειώνεται στα $c(d/2)^a$.

Όμως ο παραπάνω υπολογισμός θεωρεί μόνο εκπεμπόμενη ενέργεια και όχι πραγματικά καταναλισκόμενη ενέργεια, αφού ας μην ξεχνάμε ότι ενέργεια καταναλώνεται και στον ενδιάμεσο κόμβο. Για να μπορούμε λοιπόν να έχουμε τη συνολική κατανάλωση ενέργειας πρέπει να λάβουμε υπόψη αυτά που αναφέραμε στη παράγραφο 2.2. Επίσης είναι αρκετά εύκολο να δείχτεί ότι μεγάλο ποσό ενέργειας καταναλώνεται άσκοπα αν η απόσταση d είναι παρά πολύ μικρή. Μόνο λοιπόν για μεγαλύτερες αποστάσεις d μπορούμε να δείξουμε ότι το κόστος της εκπεμπόμενης ενέργειας υπερτερεί αυτού που καταναλώνεται στα ηλεκτρονικά του πομπού και του δέκτη. Ανάμεσα λοιπόν στην επιλογή της άμεσης και της multihop επικοινωνίας, υπεισέρχονται πολλές παράμετροι που αναφέρονται στη συσκευή και στο περιβάλλον. Γενικά όμως η multihop επικοινωνία προτιμάτε αναλογικά με τη κατανάλωση ενέργειας στις περισσότερες εφαρμογές.

3.1.3 Πολλαπλοί sinks και sources

Στις περισσότερες εφαρμογές απαιτείται η χρήση πολλαπλών πηγών για να μπορείτε να παρακολουθείται αποδοτικά το περιβάλλον, ενώ υπάρχουν επίσης εφαρμογές που χρησιμοποιούνται και πολλαπλοί sinks. Κάποιες από τις πιο περίπλοκες περιπτώσεις, είναι όταν υπάρχουν πολλές πηγές και sinks και όπου όλες ή κάποιες πληροφορίες πρέπει να φτάσουν σε όλους ή κάποιους sinks. Στο σχήμα 3.3 φαίνονται οι παραπάνω περιπτώσεις. Σε επόμενη παράγραφο που θα παρουσιάσουμε κάποια βασικά μοντέλα δικτύων, θα αναλύσουμε το μοντέλο των *multiple sinks* που εισήγαγαν οι Ι. Πευκιανάκης, Ι. Χατζηγιαννάκης, και Σ. Νικολετσέας.



Σχήμα 3.3. Πολλαπλές πηγές και sinks. Σημειώνουμε ότι στο τελευταίο μισό του σχήματος και οι sinks και οι πηγές στέλνουν δεδομένα και στην αριστερή και στη δεξιά πλευρά του δικτύου.

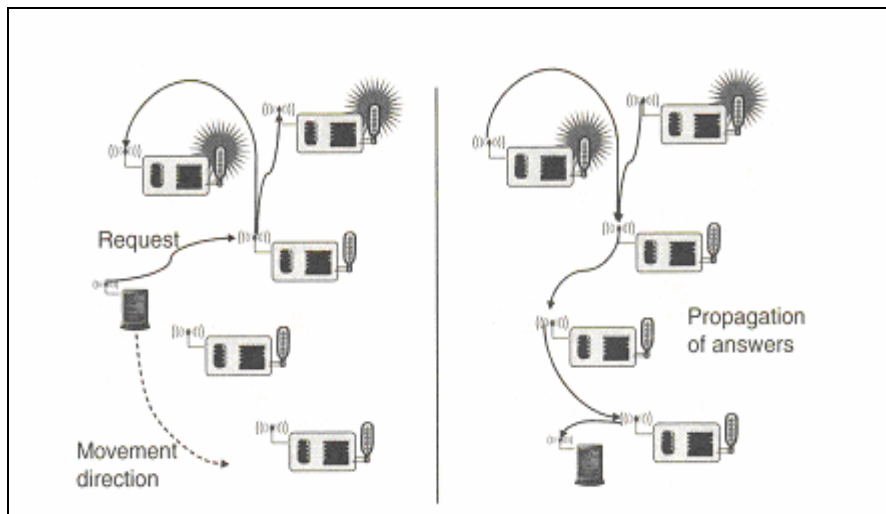
3.1.4 Τρεις τύποι κινητικότητας

Στα σενάρια που περιγράφηκαν παραπάνω, όλοι οι συμμετέχοντες στο δίκτυο ήταν στατικοί. Αλλά ένα από τα μεγαλύτερα πλεονεκτήματα της ασύρματης επικοινωνίας είναι ότι μπορεί να υποστηρίξει κινητούς κόμβους. Στα WSNs η ιδέα της κινητικότητας (*mobility*) μπορεί να εμφανιστεί σε τρεις κύριες μορφές:

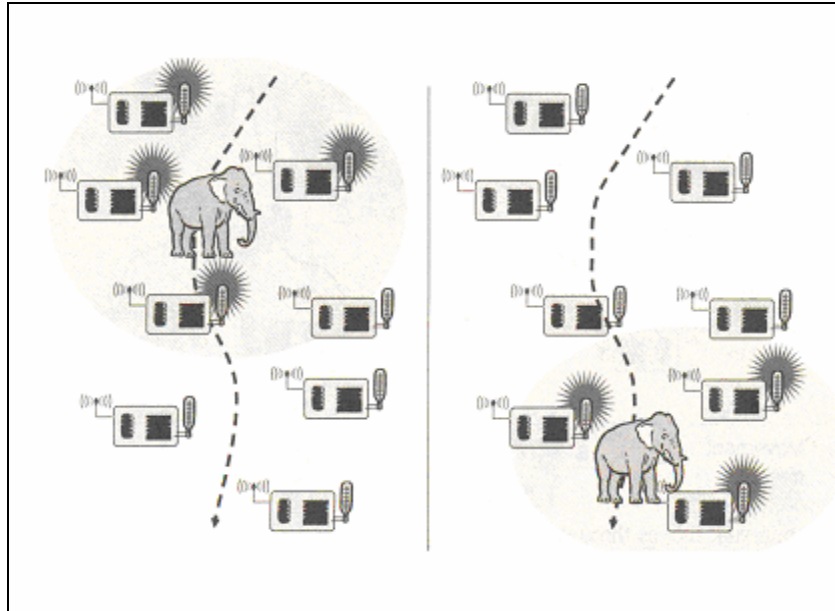
1. **Κινητικότητα Κόμβου:** Οι ασύρματοι κόμβοι αισθητήρων μπορεί να είναι κινητοί. Η έννοια της κινητικότητας αυτών των κόμβων εξαρτάται βέβαια πάρα πολύ από την εφαρμογή. Σε παραδείγματα εφαρμογών παρακολούθησης του περιβάλλοντος, η κινητικότητα των κόμβων δεν είναι απαραίτητη. Σε εφαρμογές παρακολούθησης της ζωής των ζώων ενός αγροκτήματος (π.χ. οι κόμβοι μπορεί να έχουν προσκολληθεί σε κάποιο ζώο), η κινητικότητα των κόμβων είναι απαραίτητη.
2. **Κινητικότητα Sink:** Οι sinks που συλλέγουν τις πληροφορίες μπορεί να είναι επίσης κινητοί (σχήμα 3.4). Ενώ αυτή η περίπτωση μπορεί να ουσιαστικά να ανήκει στην ιδέα κινητικότητας των κόμβων που αναφέραμε παραπάνω, υπάρχει περίπτωση ο sink να μην είναι κομμάτι του WSN, αλλά για παράδειγμα μπορεί να είναι ένας άνθρωπος ο οποίος κάνει αίτηση για κάποιες πληροφορίες με τη χρήση ενός PDA καθώς περπατάει σε ένα «έξυπνο» (δικτυωμένο δηλαδή) κτίριο. Σε μια τέτοια απλή περίπτωση, μπορεί ο sink (άνθρωπος με το PDA) να αλληλεπιδράσει με το WSN μένοντας ακίνητος σε ένα σημείο, και να κινηθεί μόνο όταν έχει τελειώσει αυτή η αλληλεπίδραση. Σε κάποιες άλλες περιπτώσεις, μπορεί διαδοχικές αλληλεπιδράσεις με το WSN, να θεωρηθούν ως ξεχωριστές και ασυσχέτιστες αιτήσεις για πληροφορία. Επίσης εάν ο sink επιτρέπεται να

αλληλεπιδράσει με όλους του κόμβους του δικτύου ή μόνο με κάποιους συγκεκριμένους, είναι μία σχεδιαστική επιλογή των επιπέδων του πρωτοκόλλου. Τέλος μία πολύ ενδιαφέρουσα περίπτωση είναι αν τα γεγονότα είναι τοπικά διαθέσιμα ή εάν πρέπει να ανακτηθούν από διαφορετικό μέρος του δικτύου.

3. **Κινητικότητα Γεγονότων:** Σε εφαρμογές όπως ανίχνευση γεγονότων ή παρακολούθηση κάποιου στόχου, και η αιτία των γεγονότων αλλά και ο στόχος μπορεί να είναι κινητός. Σε τέτοια σενάρια, είναι (συνήθως) σημαντικό ότι το γεγονός υπό παρατήρηση να καλύπτεται από έναν επαρκή αριθμό από κόμβους όλη την ώρα. Έτσι λοιπόν οι κόμβοι που είναι γύρω από το αντικείμενο, πρέπει να ξυπνάνε, να το παρακολουθούν, και όταν αυτό απομακρυνθεί, αυτοί να επιστρέφουν στη κατάσταση sleep. Έτσι λοιπόν καθώς το γεγονός κινείται μέσα στο δίκτυο, «ακολουθείται» από μία περιοχή ενεργοποίησης του δικτύου. Αυτό το μοντέλο αποκαλείται και “*frisbee*” και εισάγεται στο [8]. Η ιδέα αυτής της κινητικότητας των γεγονότων φαίνεται στο σχήμα 3.5, όπου ο σκοπός του δικτύου είναι να ανιχνεύσει και να παρατηρήσει ένα ελέφαντα που κινείται. Σε αυτό το παράδειγμα οι κόμβοι που δεν βρίσκονται στην ενεργή περιοχή παρακολούθησης, μπορεί να είναι σε μια κατάσταση sleep, και να γίνονται ενεργοί μόνο όταν χρειάζεται να γίνουν ενδιάμεσοι για να διαβιβαστεί πληροφορία στο sink.



Σχήμα 3.4. Ένας κινητός sink κινείται μέσα στο δίκτυο καθώς οι πληροφορίες ανακτώνται για αυτόν.



Σχήμα 3.5. Περιοχή κόμβων αισθητήρων που ανιχνεύουν ένα γεγονός (ένα ελέφαντα). Οι διακεκομμένες γραμμές δείχνουν τη πορεία του ελέφαντα. Η σκιασμένη περιοχή αναφέρεται στην ενεργή περιοχή του δικτύου.

Τα πρωτόκολλα επικοινωνίας για τα WSNs, θα πρέπει να είναι σε θέση να ανταποκριθούν σε αυτούς τους τύπους κινητικότητας. Ιδιαίτερα η κινητικότητα των γεγονότων, παρουσιάζει ποικιλομορφία και διαφοροποιείται πάρα πολύ ανάλογα με την εφαρμογή.

3.2 ΠΑΡΑΓΟΝΤΕΣ ΠΟΥ ΕΠΗΡΕΑΖΟΥΝ ΤΟ ΣΧΕΔΙΑΣΜΟ ΕΝΟΣ WSN

Αν και εξαιτίας της πληθώρας εφαρμογών των WSNs υπάρχουν πολλοί διαφορετικοί παράγοντες που επηρεάζουν το σχεδιασμό του δικτύου, θα προσπαθήσουμε να εστιάσουμε στους πιο σημαντικούς οι οποίοι αποτελούν και οδηγό για σχεδιασμό πρωτοκόλλων και αλγορίθμων για τα WSNs. Οι παράγοντες αυτοί επίσης μπορεί να αποτελέσουν μέτρα σύγκρισης μεταξύ των διαφορετικών πρωτοκόλλων.

3.2.1 Τεχνολογικές προδιαγραφές

Κάθε κόμβος αισθητήρων ή καλύτερα κάθε *smart dust* συσκευή (ή και *grain particle*), μπορεί να έχει το μέγεθος κυβικής ίντσας και να περιέχει τα τέσσερα βασικά υποσυστήματα : α) τροφοδότη ενέργειας, β) επεξεργαστή, γ) αισθητήρα, δ) transceiver.

Όπως έχουμε προαναφέρει, ένα πολύ μεγάλος περιορισμός για τη κατασκευή κυκλωμάτων για αυτούς του κόμβους είναι η τάση λειτουργίας. Από τη στιγμή που αυτές οι συσκευές λειτουργούν με μπαταρίες και επειδή τα RF chips απαιτούν τάση λειτουργίας στα 3 Volts, οι κατασκευαστές εστιάζουν σε πηγές των τριών Volts (συγκεκριμένα σε μπαταρίες λιθίου, που λειτουργούν στην επιθυμητή τάση με 285 *mWatt-hrs/gram* πυκνότητα ενέργειας και με μέγιστη ένταση στα 10 *mA* στα 10.5 *grams*).

Για να μπορούν οι κόμβοι του δικτύου να λειτουργούν για μεγάλο χρονικό διάστημα, έχουν σχεδιαστεί να εισέρχονται σε μια κατάσταση «ύπνου» (sleep mode), έτσι ώστε να σώζουν ενέργεια. Όλοι οι μικρό-ελεγκτές του εμπορίου έχουν τη δυνατότητα να ρίχνουν τη τάση λειτουργίας του (sleep mode) και να καταναλώνουν κατά μέσο όρο 1 μW .

Οι αισθητήρες σε ένα WSN παρέχουν ακατέργαστα δεδομένα στους κόμβους, με βάση τα οποία οι κόμβοι θα πρέπει να κατανοήσουν το περιβάλλον και να δράσουν ανάλογα. Οι αισθητήρες γενικότερα μπορούν να χωριστούν σε δύο κατηγορίες: α) αυτοί που παρακολουθούν τον καιρό β) αυτοί που παρακολουθούν κίνηση. Κάποιοι τύποι αισθητήρων όπως έχουμε προαναφέρει είναι το μαγνητόμετρο, το επιταχυνσιόμετρο, οι αισθητήρες φωτός, θερμοκρασίας, πίεσης και υγρασίας. Οι αισθητήρες συνήθως λειτουργούν σε τάσεις από 2.7 μέχρι 9 Volts, ενώ καταναλώνουν 200-650 μA .

Τέλος, όσον αφορά τις τεχνολογίες επικοινωνίας, κάποιος μπορεί να επιλέξει κατά τη κατασκευή του δικτύου του, ανάμεσα σε τρεις διαφορετικές. Ο TR1000 RF Monolithic transceiver έχει σχεδιαστεί για μικρού εύρους εφαρμογές και καταναλώνει κατά μέσο όρο 18 *mWatts* ενέργεια, ενώ παρέχει ρυθμό μετάδοσης δεδομένων στα 19.2 *kbps* σε αποστάσεις 5-30 *m*. Ο Bluetooth συμβατός Philstar PH2401 transceiver, καταναλώνει λιγότερο από 20 *mWatts* ενέργεια, και επιτυγχάνει ρυθμούς 1 *Mbit/s* σε αποστάσεις 10 μέτρων. Τέλος, μία εναλλακτική στην RF επικοινωνία είναι η μονής κατεύθυνσης ασύγχρονη laser επικοινωνία, η οποία μπορεί να παρέχει ρυθμό μεγαλύτερο των 5 *kbps* σε απόσταση 21.4 *km*.

3.2.2 Κόστος παραγωγής

Από τη στιγμή που ένα WSN αποτελείται από ένα πολύ μεγάλο αριθμό κόμβων, το κόστος κάθε κόμβου είναι πολύ σημαντικό για το συνολικό κόστος κατασκευής και συντήρησης του δικτύου. Αν το κόστος λοιπόν ενός WSN είναι μεγαλύτερο από το κόστος να παρατάξουμε «χειροκίνητα αισθητήρες», τότε δεν δικαιολογεί την ύπαρξη του. Σαν αποτέλεσμα το κόστος κάθε κόμβου πρέπει να διατηρηθεί σε χαμηλά επίπεδα. Η αγορά σήμερα διαθέτει ένα Bluetooth σύστημα (που θεωρείται φθηνή συσκευή) με λιγότερο από 10 euro, όμως το κόστος ενός κόμβου αισθητήρα θα πρέπει να είναι κάτω του ενός euro για να μπορεί να είναι εφικτή η κατασκευή μεγάλων δικτύων.

3.2.3 Μέσο μετάδοσης

Σε ένα WSN οι κόμβοι είναι συνδεδεμένοι μεταξύ τους ασύρματα. Οι σύνδεσμοι μεταξύ κόμβων μπορούν να σχηματιστούν είτε με ράδιο-σήματα, είτε με υπέρυθρες, είτε με οπτικά μέσα. Για να είναι δυνατή η καθολική λειτουργία αυτών των δικτύων, πρέπει το ασύρματο μέσο μετάδοσης να οριστεί για να γίνει διαθέσιμο παγκοσμίως. Ένα μεγάλο πάντως μέρος του διαθέσιμου hardware για τους κόμβους αισθητήρων βασίζεται στο σχεδιασμό RF κυκλωμάτων. Για παράδειγμα η χαμηλής ισχύος συσκευή αισθητήρων που παρουσιάζεται στο [10] χρησιμοποιεί μονού καναλιού RF transceiver που λειτουργεί στα 916 MHz.

Ένας άλλος τρόπος για επικοινωνία μεταξύ κόμβων είναι μέσω υπέρυθρων. Αυτού του τύπου η επικοινωνία δεν χρειάζεται άδεια και είναι αρκετά ανθεκτική στις παρεμβολές από ηλεκτρικές συσκευές. Επίσης οι transceivers υπέρυθρων είναι φθηνότεροι και κατασκευάζονται ευκολότερα. Οι επικοινωνία με υπέρυθρες, αλλά και με οπτικό τρόπο, απαιτεί οπτική επαφή (LOS επικοινωνία) μεταξύ πομπού και δέκτη.

3.2.4 Ενέργεια

Όπως έχουμε αναφέρει παραπάνω οι κόμβοι αισθητήρων μπορούν να εφοδιαστούν μόνο με περιορισμένη πηγή ενέργειας ($< 0.5 Ah, 1.2 V$). Επίσης σε κάποια σενάρια εφαρμογών η αναπλήρωση ενέργειας είναι αδύνατη. Σε ένα multihop WSN κάθε κόμβος άλλοτε έχει το ρόλο της πηγής πληροφορίας και άλλοτε το ρόλο του δρομολογητή. Οι δυσλειτουργία κάποιων κόμβων του δικτύου μπορεί να προκαλέσει σημαντικές αλλαγές στη τοπολογία του δικτύου και να κάνει απαραίτητη την επανά-δρομολόγηση πακέτων ή την αναδιοργάνωση του δικτύου. Έτσι η σωστή διαχείριση και συντήρηση των πηγών ενέργειας γίνεται ακόμα πιο σημαντική. Για τους παραπάνω λόγους οι ερευνητές προσπαθούν να σχεδιάζουν πρωτόκολλα και αλγόριθμους που είναι αποδοτικά ως προς την ενέργεια

3.2.5 Ανοχή στα σφάλματα

Οι κόμβοι αισθητήρων μπορεί να αποτύχουν ή να μπλοκάρουν είτε εξαιτίας έλλειψης ενέργειας, είτε κάποιας φυσικής παρεμβολής είτε εξαιτίας κάποιας βλάβης. Η αποτυχία ενός ή περισσότερων κόμβων, δεν πρέπει να επηρεάζει σημαντικά τη συνολική λειτουργία του δικτύου. Αυτό το θέμα είναι γνωστό ως *fault tolerance* (ανοχή στα λάθη) στα WSNs. Εξ' ορισμού λοιπόν, *fault tolerance* είναι η ικανότητα που έχει ένα δίκτυο να διατηρεί τη λειτουργικότητα του ακόμα και αν κάποιοι κόμβοι τεθούν εκτός λειτουργίας.

3.2.6 Τοπολογία δικτύου

Από μερικές εκατοντάδες, μέχρι μερικές χιλιάδες κόμβοι απλώνονται στο πεδίο του δικτύου. Περίπου η πυκνότητα των κόμβων είναι $20 \text{ κόμβοι}/m^3$. Η πυκνή αυτή παράταξη (deployment) ενός μεγάλου αριθμού κόμβων απαιτεί προσεκτική διαχείριση της τοπολογίας του δικτύου. Η τοπολογία ενός WSN πρέπει να εξεταστεί σε τρεις φάσεις:

1. *Φάση προ-παράταξης (pre-deployment) και παράταξης*: Οι κόμβοι αισθητήρων μπορούν να απλωθούν μαζικά και άτακτα στο πεδίο του δικτύου, ή να τοποθετηθούν ένας προς ένα. Μπορούν για παράδειγμα να πεταχτούν από ένα αεροπλάνο στο πεδίο ενδιαφέροντος, ή να τοποθετηθούν ένας προς ένα από ένα αεροπλάνο ή ρομπότ.
2. *Φάση μετά-παράταξης (post-deployment)*: Μετά την παράταξη των κόμβων, η τοπολογία μπορεί να αλλάξει εξαιτίας: α) της θέσης, β) της προσβασιμότητας (κινούμενα εμπόδια, θόρυβος κ.α.), γ) της διαθέσιμης ενέργειας, δ) της δυσλειτουργίας των κόμβων.
3. *Φάση επαναδιάταξης (redeployment) επιπλέον κόμβων*: Επιπλέον κόμβοι μπορούν να προστεθούν στο δίκτυο οποιαδήποτε στιγμή, για να αντικαταστήσουν κάποιους άλλους που δυσλειτουργούν, ή επειδή το απαιτεί η εφαρμογή.

3.2.7 Παράταξη κόμβων (deployment)

Οι κόμβοι αισθητήρων είναι πυκνά απλωμένοι μέσα ή πολύ κοντά στο πεδίο ενδιαφέροντος. Πολλές φορές τα WSN λειτουργούν χωρίς επιτήρηση σε απομακρυσμένες γεωγραφικά περιοχές. Το πεδίο ενδιαφέροντος για να ένα WSN μπορεί να είναι ο πυθμένας ενός ωκεανού, μία χημικά και βιολογικά μολυσμένη περιοχή, ένα πεδίο μάχης ή ακόμα ένα σπίτι ή μεγάλο κτίριο.

3.3 ΜΟΝΤΕΛΑ ΔΙΚΤΥΩΝ

Η μοντελοποίηση των Smart Dust δικτύων, χρησιμοποιείται για τη θεωρητική ανάλυση και για την ανάπτυξη πρωτοκόλλων. Η έννοια του «μοντέλου», περικλείει τη διάταξη των κόμβων στο χώρο, τη θέση του κέντρου ελέγχου (*sink*) και τις δυνατότητες επικοινωνίας που έχουν οι κόμβοι μεταξύ τους και με τον *sink*. Ως γενικευμένο μοντέλο για τα δίκτυα έξυπνης σκόνης, μπορούμε να θεωρήσουμε τη διασπορά ενός μεγάλου αριθμού από κόμβους αισθητήρων οι οποίοι όπως έχουμε αναφέρει έχουν πολύ μικρό μέγεθος και για αυτό χαρακτηρίζονται ως *grain particles* (η *σωματίδια σκόνης*).

Οι κόμβοι αισθητήρων θεωρούνται όμοιοι μεταξύ τους και ισότιμοι όσον αφορά τις δυνατότητες και τις λειτουργίες που επιτελούν. Διαθέτουν όλοι τους transceivers, αισθητήρες για ανίχνευση του περιβάλλοντος και φυσικά επεξεργαστή για ανάλυση των δεδομένων. Επιπρόσθετα είναι εφοδιασμένοι με μπαταρίες (είναι ισότιμοι και στα ποσά

ενέργειας), οι οποίες όμως σταδιακά εξαντλούνται. Επίσης τα περισσότερα σενάρια εφαρμογών αναφέρονται σε κόμβους που είναι ακίνητοι (στα δικτυακά μοντέλα που ακολουθούν οι κόμβοι θεωρούνται ακίνητοι).

Οι βασικές λειτουργίες των κόμβων αισθητήρων είναι η συλλογή πληροφοριών από το περιβάλλον με τη βοήθεια των αισθητήρων που διαθέτουν, και η προώθηση τους στον sink. Ο κάθε κόμβος μπορεί να έχει ένα σύνολο από διαφορετικούς αισθητήρες και να αντιλαμβάνεται μια ποικιλία φαινομένων που συμβαίνουν γύρω του. Επιπρόσθετα, κάθε κόμβος αναλαμβάνει να δημιουργήσει πληροφορίες σχετιζόμενες με τις μετρήσεις των αισθητήρων του αλλά και να δρομολογήσει κάθε εισερχόμενη πληροφορία στην περίπτωση της multihop διάδοσης. Η κύρια διαθέσιμη μέθοδος μετάδοσης στους κόμβους, είναι η διάδοση μηνυμάτων με τη τεχνική broadcast σε συγκεκριμένη ακτίνα R . Κατά κανόνα, όλοι οι κόμβοι που βρίσκονται εντός εμβέλειας R , θα λάβουν το μήνυμα (θεωρούμε ιδανικό MAC). Μια παραλλαγή αυτού του μοντέλου επιτρέπει τη διάδοση με τεχνική broadcast υπό γωνία. Αυτό σημαίνει ότι το μεταδιδόμενο μήνυμα, θα το λάβουν μόνο εκείνοι οι κόμβοι που βρίσκονται στο κυκλικό τομέα που ορίζεται από την ακτίνα R , και τις επίκεντρες γωνίες $+\alpha$ και $-\alpha$ ως προς μια ευθεία αναφοράς.

Κάποια πρωτόκολλα προβλέπουν ότι οι κόμβοι μπορούν αν χρησιμοποιούν οπτικά μέσα επικοινωνίας (π.χ. laser) για κατευθυνόμενη μετάδοση προς σημείο-κόμβο. Έτσι μπορεί να επιτευχθεί η κατευθυνόμενη επικοινωνία που αναφέραμε παραπάνω. Η μετάδοση γίνεται με απώτερο σκοπό η πληροφορία να φτάσει στο sink. Ο sink θεωρείται ότι έχει ανεξάντλητους υπολογιστικούς και ενεργειακούς πόρους και αντιπροσωπεύει τις αρχές εκείνες που είναι υπεύθυνες για την ανάπτυξη του δικτύου και ενδιαφέρονται να συλλέξουν τα δεδομένα από αυτό.

Τέλος αξ σημειωθεί ότι πολλές φορές οι κόμβοι είναι απαραίτητο να γνωρίζουν τη τοποθεσία τους, γιατί η πληροφορία που στέλνουν στο sink πρέπει να συνοδεύεται από τις συντεταγμένες που έγινε το γεγονός. Για αυτό πιθανόν να χρειάζεται να τρέχει κάποιος *localization* αλγόριθμος.

3.3.1 Μοντέλο πηγής-Sink

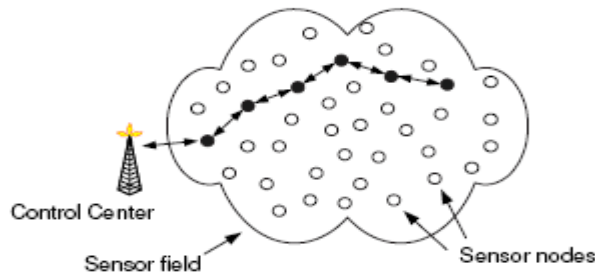
Το μοντέλο πηγής-sink είναι ένα πολύ δημοφιλές μοντέλο και χρησιμοποιείται πάρα πολύ στα πρωτόκολλα για διάδοση πληροφορίας στα WSNs όπως για παράδειγμα στα [11], [12]. Σε αυτό το μοντέλο οι κόμβοι του δικτύου είναι διασκορπισμένοι σε μια επίπεδη επιφάνεια (2-dimensional plane) και αποτελούν σημεία αυτής, χωρίς να έχουν κάποιο μέγεθος. Η πυκνότητα του δικτύου θεωρείται αρκετά μεγάλη και επίσης κάθε κόμβος έχει ακτίνα μετάδοσης R που συνήθως είναι αρκετά μικρότερη από τη διάμετρο του δικτύου. Το κέντρο ελέγχου (sink) αναπαρίσταται επίσης με ένα σημείο στο επίπεδο.

Για να μπορούν οι κόμβοι να εκπέμπουν μηνύματα προς τον sink, θα πρέπει πριν εφαρμοστεί οποιοδήποτε πρωτόκολλο, να πραγματοποιηθεί μία φάση «ανακάλυψης» όπου οι κόμβοι θα ανακαλύψουν τη κατεύθυνση όπου κείται ο sink. Στη φάση αυτή συνήθως ανταλλάσσονται μηνύματα ελέγχου μεταξύ των κόμβων, συνήθως με μηχανισμούς πλημμύρας (flooding) του δικτύου. Επίσης σε αυτή τη φάση μπορεί να μάθει ο κόμβος επιπλέον τη σχετική απόσταση του από τους γειτονικούς του κόμβους,

καθώς επίσης και να υπολογίσει την κατεύθυνση του λαμβανόμενου μηνύματος ελέγχου με τη χρήση τεχνολογία κατευθυντικών κεραιών.

Ο παρακάτω ορισμός συνοψίζει την ιδέα του μοντέλου πηγής-sink:

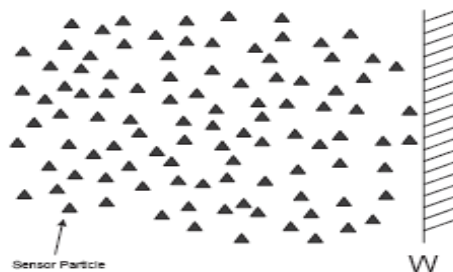
Ορισμός 1: Ας υποθέσουμε ότι d είναι η πυκνότητα (μετριέται ως ο αριθμός των particles/m²) των κόμβων-particles σε μια περιοχή. Επίσης έστω R το μέγιστο εύρος μετάδοσης κάθε particle. Ένας Sink S , ορίζεται σαν ένα σημείο στο επίπεδο του δικτύου. Η μετάδοση κάθε particle που βρίσκεται σε απόσταση R από τον sink S , θα παραληφθεί από αυτόν.



Σχήμα 3.6. Μοντέλο Πηγής-Sink.

3.3.2 Μοντέλο πηγής-Τείχους

Το μοντέλο «πηγής-τείχους» ή *Source-Wall* προτάθηκε από τους Ι. Χατζηγιαννάκη, Σ. Νικολετσέα και Π. Σπυράκη στο [13]. Η βασική διαφορά που παρουσιάζει σχετικά με το προηγούμενο μοντέλο, είναι ότι αντί ο sink να είναι ένα απλό σημείο στο επίπεδο, είναι ένα μεγάλου μήκους τείχος το οποίο αντιπροσωπεύει το κέντρο έλεγχου. Οποιαδήποτε αποστολή μηνύματος προς το τείχος, από κόμβο που είναι εντός απόστασης R από αυτό, σημαίνει αυτόματα και λήψη του μηνύματος από το κέντρο ελέγχου. Μια πιθανή υλοποίηση του τείχους αυτού θα ήταν μια συστοιχία από κέντρα ελέγχου κατά μήκος του τείχους, που θα ήταν σε θέση να επικοινωνούν. Μια εναλλακτική υλοποίηση θα ήταν ένας κινητός κόμβος που θα είχε το ρόλο του κέντρου ελέγχου και θα μπορούσε να κινηθεί κατά μήκος του τείχους με μεγάλη ταχύτητα και να συλλέξει τις πληροφορίες. Στην ανάλυση των πρωτοκόλλων (π.χ. LTP) όπως θα δούμε παρακάτω, το Wall θεωρείται άπειρου μήκους.



Σχήμα 3.7. Μοντέλο Πηγής-Τείχους.

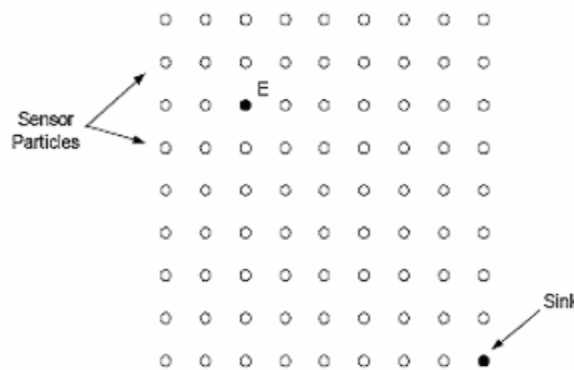
3.3.3 Μοντέλο *Lattice* κατανομής

Το μοντέλο αυτό εισήγαγαν οι Ι. Χατζηγιαννάκης, Τ. Δημητρίου, Σ. Νικολετσέας και Π. Σπυράκης για τη γεωμετρική ανάλυση του PFR ([11]). Το μοντέλο *Lattice* αναφέρεται στην διασπορά των particles στο δισδιάστατο επίπεδο όπως φαίνεται στο σχήμα 3.8. Αυτό το μοντέλο αναφέρεται σε προσεγγίσεις όπου το δίκτυο δεν απλώνεται άτακτα, αλλά εγκαθίσταται κάθε κόμβος είτε από ρομπότ είτε από άνθρωπο στο δισδιάστατο επίπεδο. Ένα τέτοιο μοντέλο θα ήταν πιθανόν χρήσιμο σε εφαρμογές γεωργίας όπου οι άνθρωποι ή τα ρομπότ θα τοποθετούσαν τους κόμβους αισθητήρων με ένα ομοιόμορφο τρόπο ώστε να μπορούσαν να παρατηρήσουν το περιβάλλον ενδιαφέροντος με ομογενή τρόπο. Πιθανότατα και άλλες εφαρμογές (π.χ. στρατιωτικές) θα μπορούσαν να χρησιμοποιήσουν μια τέτοια τύπου grid κατανομή. Να σημειώσουμε εδώ ότι ένας τέτοιος τύπος δικτύου μπορεί να προκύψει με μεγάλη πιθανότητα, αν απλώσουμε τυχαία ένα πολύ μεγάλο αριθμό κόμβων, όπως μπορεί ναδειχτεί με random γεωμετρική ανάλυση ([14]).

Το μοντέλο υποθέτει μία $n \times n$ lattice κατανομή με n^2 κόμβους, καθένας από τους οποίους έχει εύρος μετάδοσης R και επίσης καθένας απέχει από τον άλλο (κάθετα και οριζόντια) κατά δ . Έτσι λοιπόν κάθε particle περιβάλλεται από οκτώ γείτονες.

Υπάρχει ένα μοναδικό σημείο στο lattice, το οποίο είναι ο sink και το οποίο αντιπροσωπεύει το κέντρο ελέγχου στο οποίο πρέπει να φτάσει η πληροφορία. Στο μοντέλο αυτό ο sink θεωρείται στατικός, αλλά πιθανόν να μπορεί να κινηθεί ελάχιστα ώστε να συλλέξει δεδομένα τα οποία έφτασαν πολύ κοντά, αλλά όχι σε απόσταση R από αυτόν.

Τέλος, τα particles είναι σε θέση να γνωρίζουν τη θέση τους στο lattice καθώς επίσης και τη τοποθεσία του sink (π.χ. χρησιμοποιώντας GPS πληροφορίες, ή απλά τη γωνία της ευθείας που συνδέει αυτά με το sink σε σχέση με ένα κοινό σύστημα συντεταγμένων).



Σχήμα 3.8. Smart Dust Lattice.

3.3.4 Μοντέλο πολλαπλών Walls ή Sinks

Το μοντέλο των πολλαπλών τειχών (*Multiple-Wall* ή *Multiple-Sink Model*) προτάθηκε από τους Ι. Πευκιανάκη, Ι. Χατζηγιαννάκη, Σ. Νικολετσέα ([35]), και αποτελεί μία γενίκευση του μοντέλου *πηγής-τείχους*. Έτσι δανείζεται τη δομή του μοντέλου *πηγής-τείχους* καθώς επίσης υιοθετεί την ιδέα των *grain particles* όπως παρουσιάστηκαν στις προηγούμενες τρεις παραγράφους.

Το *Multiple-Sink* μοντέλο (έτσι θα το αναφέρουμε από εδώ και πέρα), δεν θεωρεί ότι υπάρχει μόνο ένα τείχος (Wall) σε κάποια πλευρά του επιπέδου του δικτύου, αλλά ότι υπάρχουν πολλαπλοί *sinks* που είναι σε θέση να καλύψουν περιμετρικά, μερικώς ή ολικώς, το επίπεδο του δικτύου. Κάθε *sink* μπορεί να καλύψει ένα μέρος της περιμέτρου του δικτύου, και είναι υπεύθυνος να συλλέγει τα δεδομένα που φθάνουν στα όρια του δικτύου, δηλαδή στους «*συννοριακούς*» (“*boundary*”) κόμβους.

Ορισμός 2: Ας υποθέσουμε ότι το R το μέγιστο εύρος μετάδοσης κάθε κόμβου. Ένας *συννοριακός* (*boundary*) κόμβος, είναι ένα κόμβος που είναι εντός ακτίνας R από κάποιον *sink*.

Μια γραφική αναπαράσταση του μοντέλου των πολλαπλών *sinks* φαίνεται στο σχήμα 3.9. Οι χρωματισμένοι κόμβοι, είναι οι *boundary* κόμβοι, ενώ τα βέλη οριοθετούν το πεδίο δράσης κάθε *sink*. Να σημειώσουμε εδώ ότι στο σχήμα 3.9 θεωρούμε ότι οι κόμβοι καλύπτουν ολόκληρη τη περίμετρο του δικτύου. Όμως ανάλογα με την εφαρμογή, ένα τέτοιο σενάριο μπορεί να μην είναι εφικτό, και έτσι θεωρήσουμε μερική κάλυψη της περιμέτρου του δικτύου. Να σημειώσουμε επίσης, ότι θα μπορούσε να υπήρχε μόνο ένας κινητός *sink* (π.χ. ένα ρομπότ που θα κινούνταν και θα έπαιζε το ρόλο του κέντρου ελέγχου) που θα κάλυπτε ολικώς ή μερικώς τη περίμετρο του δικτύου. Πρακτικά όμως μία τέτοια περίπτωση δεν θα ήταν ρεαλιστική, μιας και ένας μόνο κινητός *sink* δεν θα μπορούσε να συλλέξει όλα τα δεδομένα με επιτυχία ([29]).

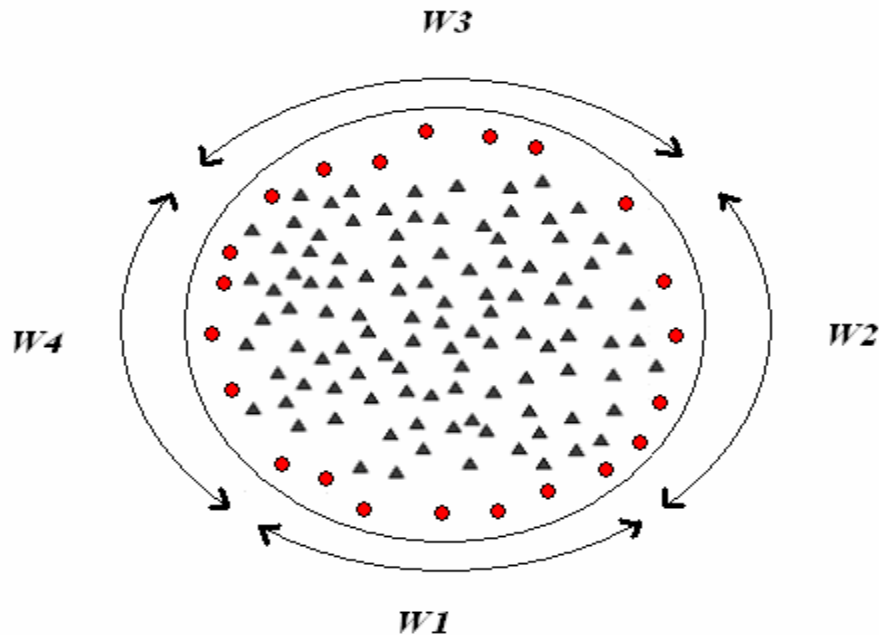
Η ιδέα των πολλαπλών *sink* μπορεί να υλοποιηθεί πρακτικά κάτω από ένα διαφορετικό σενάριο. Οι *boundary* κόμβοι μπορούν να θεωρηθούν κόμβοι με μεγαλύτερες δυνατότητες από αυτές που αναφέρθηκαν στο παραπάνω ορισμό. Τα *boundary particles* δηλαδή, μπορεί να είναι δευτερεύοντες στατικοί *sinks*, με μόνιμη ή ανανεώσιμη τροφοδοσία ενέργειας (π.χ. επαναφορτιζόμενες μπαταρίες), και με μεγαλύτερο εύρος μετάδοσης. Έτσι λοιπόν, όταν λάβουν ένα γεγονός που πρέπει να προωθηθεί σε κάποιο κέντρο ελέγχου, μπορούν να το μεταδώσουν στον κοντινότερο *sink* (ή ακόμα σε κάποιο άλλο *boundary particle* που είναι κοντότερα σε κάποιο *sink*). Επίσης τα *boundary particles* μπορεί να τοποθετηθούν εκ των προτέρων στο πεδίο ενδιαφέροντος από άνθρωπο ή ρομπότ. Χρησιμοποιώντας λοιπόν *boundary* κόμβους με τα παραπάνω χαρακτηριστικά, θα χρειαστούμε πιθανόν λιγότερους κινητούς *sinks* (άρα μικρότερο κόστος), και επίσης θα μπορούμε να εφαρμόσουμε το μοντέλο μας σε εφαρμογές όπου η τοποθέτηση κινητών κόμβων σε κάθε σημείο του δικτύου είναι αδύνατη (π.χ. σε ένα πεδίο μάχης). Βέβαια όλα τα παραπάνω δυνατά σενάρια υλοποίησης δεν επηρεάζουν την βασική ιδέα και τα χαρακτηριστικά του μοντέλου μας.

Για να μπορέσουμε να δώσουμε μια γεωμετρική υπόσταση στο μοντέλο μας, και για να γίνει εφικτή η ανάλυση του πρωτοκόλλου MP-DFR που θα παρουσιάσουμε

παρακάτω, θα θεωρήσουμε ότι το smart dust δίκτυο μας καλύπτει μία τετραγωνική περιοχή όπου τα διαφορετικά Walls αναφέρονται στις τέσσερις πλευρές του τετραγώνου. Να σημειώσουμε εδώ ότι το Wall έχει τα ίδια χαρακτηριστικά με αυτά που αναφέραμε στο μοντέλο πηγής-τείχους καθώς επίσης και κάθε Wall W_i για $i=1,..,4$ (λόγω των τεσσάρων πλευρών του τετραγώνου) μπορεί να υλοποιηθεί με τη βοήθεια πολλών κινητών sink. Ο αριθμός των sinks σε κάθε W_i μπορεί να είναι διαφορετικός, ανάλογα με τις ανάγκες του δικτύου (σχήμα 3.10).

Για το μοντέλο των πολλαπλών sinks, μπορούμε να υποθέσουμε ότι αρχικά τρέχει μια φάση εγκατάστασης (set-up phase) κατά τη διάρκεια της οποίας οι κόμβοι απλώνονται στο πεδίο ενδιαφέροντος. Κάθε grain particle που υπάρχει στο δίκτυο θεωρούμε ότι έχει τις εξής δυνατότητες:

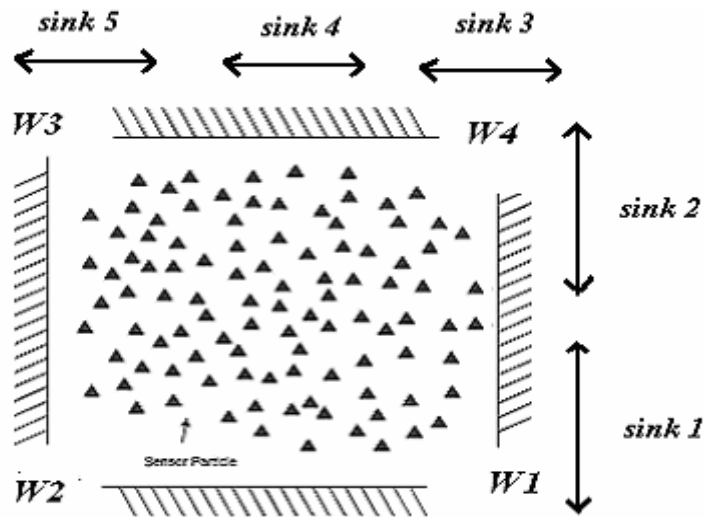
1. Μπορεί να εκτιμήσει την κατεύθυνση της λαμβανόμενης μετάδοσης (πρακτικά αυτό μπορεί να γίνει με τη χρήση κεραιών αίσθησης κατεύθυνσης).
2. Μπορεί να εκτιμήσει τη κάθετη απόσταση από ένα Wall W_i (με τη χρήση τεχνικών μέτρησης της εξασθένησης του λαμβανόμενου σήματος).
3. Όλες οι particles έχουν ένα καθολικό σύστημα συντεταγμένων.
4. Με την ανταλλαγή μηνυμάτων ελέγχου (μικρά μηνύματα που μεταδίδονται μόνο μια φορά) κατά τη διάρκεια της φάσης εγκατάστασης, κάθε particle γνωρίζει την κατεύθυνση προς κάθε διαφορετικό Wall W_i . Υποθέτοντας λοιπόν ότι κάθε particle έχει την αίσθηση της κατεύθυνσης με τη χρήση των μηνυμάτων ελέγχου, τότε θα μπορεί να γνωρίζει την τοποθεσία κάθε W_i .



Σχήμα 3.9. Multi-sink model

Τέλος, υποθέτουμε ότι οι κινητοί sinks που υλοποιούν τους Walls είναι εφοδιασμένοι με ισχυρούς πομπούς, και έτσι μπορούν να επικοινωνούν ο ένας με τον άλλο (πιθανόν μέσω κάποιου ενδιάμεσου sink). Κρίνοντας από αυτά που αναφέραμε παραπάνω, κάθε particle είναι σε θέση να μεταδώσει κάποιο σημαντικό γεγονός που ανίχνευσε σε κάποιο sink. Επίσης όπως προαναφέραμε, αν τα δεδομένα φτάσουν στους boundary κόμβους, τότε ο sink μπορεί να τα συλλέξει.

Κάποιες από τις παραπάνω υποθέσεις είναι ρεαλιστικές (π.χ. είναι πράγματι δυνατό να εφαρμόσουμε «έξυπνες» κεραίες σε αυτούς τους μικροσκοπικούς κόμβους ([30]) και κάποιες άλλες αναμένεται να γίνουν στο προσεχές μέλλον. Επίσης κάποιες από τις παραπάνω υποθέσεις μπορούν να ικανοποιούνται με την εφαρμογή εναλλακτικών λύσεων, όπως για παράδειγμα οι κόμβοι για να προσδιορίσουν τις συντεταγμένες τους, μπορούν να τρέξουν μια φάση προ-επεξεργασίας και να εκτελέσουν μια localization τεχνική, για να προσδιορίσουν κάποιες εικονικές (*fictitious*) συντεταγμένες ([31]) και έτσι χαλαρώνουμε την υπόθεση 3. Έχοντας τώρα υπολογίσει τις συντεταγμένες μπορούμε να προσδιορίσουμε τη κατεύθυνση της λαμβανόμενης μετάδοσης (χαλαρώνουμε την υπόθεση 1) και την απόσταση (χαλαρώνουμε την υπόθεση 2). Σε κάθε περίπτωση πάντως, οι παραπάνω υποθέσεις είναι απλουστευτικές και βοηθούν στην ανάλυση των πρωτοκόλλων που βασίζονται σε αυτό το μοντέλο. Επίσης δεν μπορούν να χαρακτηριστούν υπερβολικές μιας και δεν υποθέτουν χρήση GPS, ή σφαιρικής γνώσης της δομής του δικτύου.



Σχήμα 3.10. Γεωμετρική αναπαράσταση Multi-sink μοντέλου. Κάθε W_i μπορεί να υλοποιείται από ένα διαφορετικό αριθμό κινητών sinks.

Αριθμός των Sinks: Ο αριθμός των κινητών sinks στο multiple-sink μοντέλο μας, είναι ιδιαίτερα σημαντικός για την αποδοτικότητα του μοντέλου. Αν θεωρήσουμε ένα μόνο κινητό sink ο οποίος κινείται αργά πάνω-κάτω κατά μήκος μια ευθείας γραμμής, τότε αυτός μπορεί να μην είναι επαρκής για τη συλλογή όλων των πληροφοριών. Έτσι μπορεί να οδηγηθούμε σε απώλειες δεδομένων εξαιτίας κάποιων buffer overflows στα boundary

particles. Επίσης αν ο αριθμός των κινητών sinks είναι ιδιαίτερα μικρός, οι αποστάσεις που αυτοί καλούνται να καλύψουν είναι αρκετά μεγάλες. Έτσι η χωρητικότητα της μπαταρίας τους μπορεί να μην είναι επαρκής για τη μετακίνηση για τόση μεγάλη απόσταση και έτσι πιθανόν να χρειαστούν στα μέσα της διαδρομής επανατροφοδότηση ενέργειας.

Για να υπολογίσουμε τον κατάλληλο αριθμό sinks για να υλοποιηθεί ένα τέτοιο μοντέλο, θα βασιστούμε στο [29] το οποίο αναφέρεται στο ίδιο σενάριο. Το σενάριο αυτό είναι λοιπόν ότι ένας κινητός sink (data mule) κινείται κατά μήκος μιας ευθείας γραμμής και συλλέγει τα δεδομένα. Σύμφωνα με το [29] λοιπόν, ο αριθμός των sinks που απαιτούνται κατά μήκος αυτής της ευθείας γραμμής θα είναι $\lceil RTT / buffer_fill_time \rceil$ όπου:

- *num_nodes* είναι κόμβοι που παρατάσσονται σε μια επιφάνεια $l \times l$ μονάδων.
- Ένας κινητός sink κινείται κατά μήκος μια ευθείας γραμμής με ταχύτητα *s*.
- Ο χρόνος που απαιτείται για να γεμίσει το buffer κάποιου κόμβου είναι *buffer_fill_time*.
- Ο χρόνος που κάνει ένας mobile sink για να αδειάσει το buffer κάποιου κόμβου είναι *service_time* και δίνεται από τη σχέση *buffer_size/communication_data_rate*.
- RTT ή Round Trip Time για κάποιο κινητό sink θα δίνεται από τη σχέση: $(l/s) + (num_nodes \times service_time) + (l/s)$.

Περισσότερες πληροφορίες για τη παραπάνω ανάλυση μπορεί να δει κάποιος στο [29]. Μία διαφορά μεταξύ του παραπάνω σεναρίου και του δικού μας μοντέλου, είναι ότι στο [29] κάθε sink εξυπηρετεί ένα συγκεκριμένο και ίσο αριθμό από κόμβους. Όμως από την άλλη πλευρά στο μοντέλο μας, κάθε sink μπορεί να συλλέξει δεδομένα που έχουν προέλθει από οποιοδήποτε particle στο δίκτυο.

Το μοντέλο των πολλαπλών sinks είναι απλό ρεαλιστικό και απεικονίζει με ακρίβεια τις τεχνολογικές προδιαγραφές ενός πραγματικού smart dust δικτύου. Ανάλογα με την εφαρμογή, ο αριθμός των Wall (άρα και των sinks) μπορεί να αλλάξει δυναμικά χωρίς να επηρεάζει τη στιβαρότητα του μοντέλου. Επίσης το μοντέλο δεν απαιτεί από τα particles να διαθέτουν GPS, πράγμα που το κάνει ακόμα πιο ρεαλιστικό.

3.3.5 Μοντέλο κατανάλωσης ενέργειας

Όπως έχουμε ήδη αναφέρει στη παράγραφο 2.2, για τη κατανάλωση ενέργειας σε κάθε κόμβο, πρέπει να λάβουμε υπόψη τις λειτουργίες για την ανίχνευση του περιβάλλοντος, την επεξεργασία πληροφοριών και ανταλλαγή και λήψη μηνυμάτων κατά τη διάρκεια της επικοινωνίας.

Γενικά η ενέργεια για την επεξεργασία της πληροφορίας συνυπολογίζεται με την ενέργεια που καταναλώνεται για την ανίχνευση και θεωρείται ίση με ένα ποσό E_{elec} για δεδομένο χρονικό διάστημα ([15]). Για να γίνει η αποστολή ενός μηνύματος k -bit σε απόσταση d , υποθέτουμε ότι απαιτείται ενέργεια που περιγράφεται από τη σχέση:

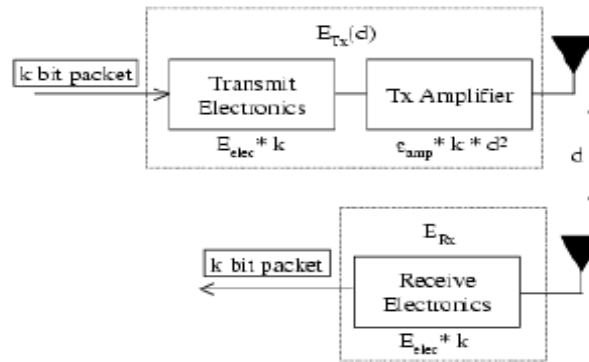
$$E_{T_x}(k, d) = E_{elec} \cdot k + e_{amp} \cdot k \cdot d^2$$

Η ενέργεια που δαπανάται για τη λήψη του ίδιου μηνύματος δίνεται από τη σχέση:

$$E_{R,r}(k) = E_{elec} \cdot k$$

Σημειώνουμε ότι το ποσό E_{elec} είναι η ενέργεια που απαιτείται για τη λειτουργία του transceiver, ενώ e_{amp} είναι η ενέργεια που απαιτείται από το πομπό για την ενίσχυση του μεταδιδόμενου σήματος, ώστε να επιτευχθεί μια επαρκής τιμή για το λόγο σήματος προς θόρυβο (SNR). Επίσης είναι φανερό από το τύπο ότι η ενέργεια εκπομπής θεωρείται ανάλογη του τετραγώνου της απόστασης. Από τους παραπάνω τύπους, μπορεί κάποιος να συμπεράνει ότι για μικρές αποστάσεις, η ενέργεια εκπομπής και λήψης συγκλίνουν στην ίδια τιμή. Τέλος το κανάλι επικοινωνίας που υποθέτουμε, θεωρείται συμμετρικό υπό την έννοια ότι η ενέργεια για τη μετάδοση από τον κόμβο A στον B απαιτείται το ίδιο ποσό ενέργειας με αυτή από τον B στον A .

Ένας κόμβος αισθητήρων έχει τη δυνατότητα να λειτουργήσει σε κατάσταση χαμηλής ενέργειας, σταματώντας τη λειτουργία κάποιων δαπανηρών κυκλωμάτων (π.χ. transceiver) ή περιορίζοντας τη λειτουργία του επεξεργαστή. Στη κατάσταση αυτή υποθέτουμε ότι η δαπάνη ενέργειας είναι αμελητέα, ή προσεγγιστικά μηδενική. Πάντως η επαναφορά από μια κατάσταση sleep σε μια κατάσταση κανονική, κοστίζει κάποιο ποσό ενέργειας που θεωρείται πολλαπλάσιο του E_{elec} .



Σχήμα 3.11. Μοντέλο κατανάλωσης ενέργειας στο transceiver RF.

ΜΕΡΟΣ 3^ο : ΑΝΑΚΑΛΥΨΗ ΤΟΠΟΛΟΓΙΑΣ & ΑΥΤΟ-ΔΙΑΜΟΡΦΩΣΗ

4 ΑΝΑΚΑΛΥΨΗ ΤΟΠΟΛΟΓΙΑΣ ΣΤΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ

Οι αρχιτεκτονικές των δικτύων αισθητήρων που απαιτούνται για τις διάφορες εφαρμογές μπορεί να αναφέρονται σε κατανεμημένα με υψηλή πολυπλοκότητα δίκτυα τα οποία αποτελούνται από εκατοντάδες κόμβους αισθητήρων. Οι περιορισμοί που επιβάλλονται το μεγάλο μέγεθος των δικτύων και το εξωτερικό περιβάλλον, είναι γνωστά και ως *dynamics*. Η αύξηση των *dynamics* του συστήματος (*system dynamics*) συμβαίνει εξαιτίας του μεγάλου αριθμού των κόμβων. Πιο συγκεκριμένα ο μεγάλος αριθμός των κόμβων κάνει απαγορευτικό το «χειροκίνητο» configuration καθενός κόμβου ξεχωριστά. Επίσης η αύξηση των *dynamics* του εξωτερικού περιβάλλοντος (*environmental dynamics*) κάνει αδύνατο το pre-configuration κάθε κόμβου. Για παράδειγμα η τοπολογία του δικτύου δεν είναι εκ των προτέρων γνωστή ώστε να μπορεί να προγραμματιστεί κάθε κόμβος ανάλογα. Από τα παραπάνω μπορούμε να συμπεράνουμε ότι η συμπεριφορά του δικτύου θα είναι τελείως απρόβλεπτη, εξαιτίας της τυχαιότητας της κατάστασης κάθε κόμβου και της δομής του δικτύου.

Με βάση λοιπόν όσα αναφέραμε παραπάνω μπορούμε να καταλάβουμε ότι η διαμόρφωση (configuration) και η διαχείριση (management) τέτοιων δικτύων είναι εξαιρετικά δύσκολη. Επιθυμητό λοιπόν θα ήταν αυτά τα δίκτυα να είναι σε θέση να αυτό-διαμορφώνονται (self-configuration) και να αυτό-διαχειρίζονται έτσι ώστε να είναι προσαρμοστικά στις διαφορετικές συνθήκες που επιβάλλει κάθε ξεχωριστή εφαρμογή.

Στο κεφάλαιο αυτό, αρχικά θα αναλύσουμε κάποια θέματα σχετικά με διαχείριση δικτύων αισθητήρων και θα περιγράψουμε κάποια πιθανά δικτυακά μοντέλα πάνω στα οποία τα διαφορετικά πρωτόκολλα μπορούν να εφαρμόσουν συγκεκριμένες λειτουργίες διαχείρισης, κατάλληλες για δίκτυα αισθητήρων. Στη συνέχεια θα αναλύσουμε ένα αποδοτικό αλγόριθμο «Ανακάλυψης Τοπολογίας» (ή “Topology Discovery”) τον “TopDisc” και θα παρουσιάσουμε τις εφαρμογές που μπορεί να έχει η γνώση της τοπολογίας σε ένα δίκτυο αισθητήρων.

Το κεφαλαίο αυτό συμπληρώνεται με το επόμενο κεφάλαιο που αναφέρεται σε θέματα αυτό-διαμόρφωσης του δικτύου (self-configuration). Έτσι ο συνδυασμός των δύο αυτών κεφαλαίων θα μας δώσει μια σαφή εικόνα για το πώς τα WSNs μπορούν να οργανωθούν χωρίς τη παρέμβαση του διαχειριστή τους.

4.1 ΔΙΑΧΕΙΡΙΣΗ WSNs

Τα δίκτυα αισθητήρων έχουν διαφορετική αρχιτεκτονική από τα κανονικά ενσύρματα δίκτυα. Οι κόμβοι σχεδιάζονται με μικρό κόστος και μικρό συντελεστή μορφής (form factor) για εύκολη τοποθέτηση σε μεγάλο αριθμό και έτσι έχουν περιορισμένη μνήμη, επεξεργαστή και μπαταρία. Οι περιορισμοί στην ενέργεια περιορίζουν επίσης και το εύρος επικοινωνίας αυτών των συσκευών. Οι κόμβοι πρέπει να διατηρούν μόνο τοπική

γνώση του δικτύου, αφού η αποθήκευση πληροφορίας για όλο το δίκτυο δεν θα ήταν scalable.

Επίσης η συμπεριφορά του δικτύου μπορεί να είναι απρόβλεπτη σε υψηλό βαθμό, εξαιτίας των χαρακτηριστικών λειτουργίας των κόμβων και του τυχαίου τρόπου με τον οποίο το δίκτυο δημιουργείται.

Οι παραπάνω παράγοντες υπονοούν ότι χρειαζόμαστε διαφορετικά μοντέλα για την αναπαράσταση της *τρέχουσας κατάστασης* του δικτύου. Ένα μοντέλο για δίκτυα αισθητήρων πρέπει να ενσωματώνει τα συγκεκριμένα χαρακτηριστικά που περιγράψαμε παραπάνω. Μερικά πιθανά μοντέλα για τα WSNs είναι τα εξής:

- *Τοπολογία Δικτύου*: Περιγράφει τον τρέχων χάρτη συνδεσιμότητας (connectivity/reachability map) του δικτύου και μπορεί να βοηθήσει σε λειτουργίες δρομολόγησης και σε μελλοντική παράταξη των κόμβων.
- *Ενεργειακός Χάρτης*: Δίνει τα ενεργειακά επίπεδα των κόμβων σε διαφορετικά τμήματα του δικτύου. Σε συνδυασμό με την Τοπολογία Δικτύου, μπορεί να χρησιμοποιηθεί για να αναγνώριση των “αδύναμων περιοχών” του δικτύου.
- *Usage Pattern*: Περιγράφει την κίνηση του δικτύου σε σχέση με περιόδους εργασίας των κόμβων και το ποσό των δεδομένων που μεταφέρονται ανά μονάδα χρόνου.
- *Μοντέλο Κόστους*: Απεικονίζει το δίκτυο σε σχέση με το κόστος εξοπλισμού, το ενεργειακό κόστος και το ανθρώπινο κόστος για την συντήρηση του δικτύου στα επιθυμητά επίπεδα λειτουργίας.
- *Μη ντετερμινιστικά μοντέλα*: Τα δίκτυα αισθητήρων είναι σε υψηλό βαθμό απρόβλεπτα και αναξιόπιστα. Τα στατιστικά και πιθανοτικά μοντέλα μπορούν να αποδειχθούν πολύ πιο αποτελεσματικά στην εκτίμηση της συμπεριφοράς του δικτύου από ότι τα ντετερμινιστικά μοντέλα.

Τα παραπάνω μοντέλα μπορούν να αποτελέσουν την Βάση Διαχείρισης Πληροφοριών (Management Information Base - MIB) για τα δίκτυα αισθητήρων. Για να ενημερώσουμε την MIB με την τρέχουσα κατάσταση του δικτύου, ένας κόμβος παρακολούθησης (monitoring node) πρέπει να μετρήσει διάφορες παραμέτρους του δικτύου.

Τα μοντέλα που αναφέραμε μπορούν να χρησιμοποιηθούν για διάφορες λειτουργίες διαχείρισης δικτύων. Ορίζουμε κάποιες από αυτές τις λειτουργίες παρακάτω:

- *Παράταξη των Αισθητήρων*: Τυπικά οι αισθητήρες μπορούν να τοποθετηθούν τυχαία χωρίς γνώση εκ των προτέρων για το έδαφος. Μελλοντικές παρατάξεις των αισθητήρων θα εξαρτώνται από την παρούσα κατάσταση του δικτύου.

- *Παρακολούθηση των Καταστάσεων του Δικτύου χρησιμοποιώντας τα Μοντέλα Δικτύου:* Λήψη περιοδικών μετρήσεων για τη λήψη ποικίλων καταστάσεων όπως συνδεσιμότητα του δικτύου, ενεργειακός χάρτης κ.α.
- *Συντήρηση Δικτύου:* Παρακολουθώντας το δίκτυο, μπορούμε να εντοπίσουμε προβληματικές περιοχές στο δίκτυο. Στη συνέχεια μπορούμε να πάρουμε διορθωτικά μέτρα όπως να τοποθετήσουμε επιπλέον αισθητήρες σε αυτήν την περιοχή.
- *Πρόβλεψη Μελλοντικών Καταστάσεων του Δικτύου:* Από περιοδικές μετρήσεις των καταστάσεων δικτύου μπορεί να καθοριστεί η δυναμική συμπεριφορά του δικτύου και να προβλεφθούν μελλοντικές καταστάσεις. Αυτό είναι χρήσιμο για να προβλέπουμε αποτυχίες του δικτύου και να παίρνουμε προληπτικά μέτρα.
- *Σχεδιασμός Δικτύων Αισθητήρων:* Μπορούν να χρησιμοποιηθούν τα μοντέλα Κόστους και Usage Pattern για το σχεδιασμό αρχιτεκτονικών των δικτύων αισθητήρων.

4.2 TOPOLOGY DISCOVERY

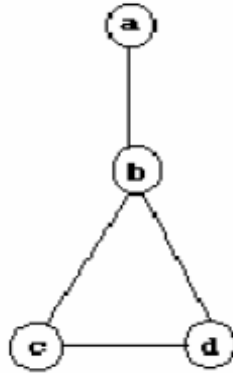
Σε αυτήν την ενότητα θα περιγράψουμε αλγορίθμους για ανακάλυψη τοπολογίας (topology discovery) που χρησιμοποιούνται στα δίκτυα αισθητήρων. Σκοπός μας είναι να κατασκευάσουμε την τοπολογία όλου του δικτύου από την πλευρά ενός μόνο κόμβου. Διαιρούμε τους αλγόριθμους σε τρία στάδια εκτέλεσης:

1. Ένας κόμβος παρακολούθησης (monitoring node) που χρειάζεται την τοπολογία του δικτύου αρχικοποιεί μία “Αίτηση για Ανακάλυψη Τοπολογίας” (*Topology Discovery Request*)
2. Η αίτηση διασχίζει το δίκτυο και φθάνει σε όλους του κόμβους που είναι σε λειτουργία (active nodes). Υποθέτουμε ότι κάθε κόμβος προωθεί την αίτηση ακριβώς μία φορά.
3. Ξεκινά ένας μηχανισμός *απόκρισης* για την αποστολή της πληροφορίας με την τοπολογία στον αρχικό κόμβο.

4.2.1 Προσεγγίσεις για ανακάλυψη τοπολογίας

Θεωρούμε τρεις μεθόδους που μπορούν να χρησιμοποιηθούν για την ανακάλυψη τοπολογίας. Σημειώνουμε ότι όταν μία *Αίτηση για Ανακάλυψη Τοπολογίας* διασχίζει το δίκτυο, κάθε κόμβος παίρνει πληροφορία για τους γειτονικούς του κόμβους. Σύμφωνα με τον μηχανισμό απόκρισης, κάθε κόμβος μπορεί να στείλει πίσω απόκριση με την λίστα των γειτόνων του. Για να επεξηγήσουμε τον μηχανισμό απόκρισης αυτών των τριών

μεθόδων, θεωρούμε το δίκτυο του σχήματος 4.1 με τον κόμβο a να είναι ο κόμβος αρχικοποίησης. Η *Αίτηση για Ανακάλυψη Τοπολογίας* φτάνει στον κόμβο b από τον a, και στους κόμβους c και d από τον κόμβο b. Οι αιτήσεις προωθούνται μόνο μία φορά.



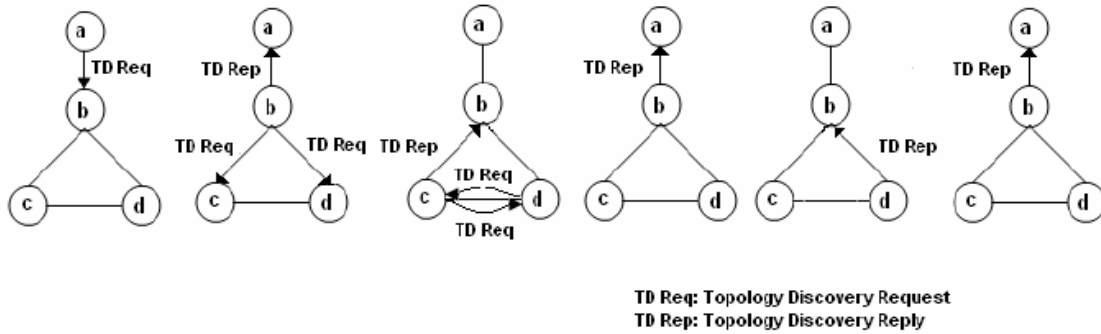
Σχήμα 4.1. Παράδειγμα για την επεξήγηση των διαφορών μεταξύ των διαφορετικών προσεγγίσεων για ανακάλυψη τοπολογίας

1. Άμεση Απόκριση (Direct Response)

Καταρχήν πλημμυρίζουμε το δίκτυο με την “Αίτηση για Ανακάλυψη Τοπολογίας”. Όταν ένας κόμβος λάβει την αίτηση, την προωθεί στους γείτονές του (στους κόμβους που είναι μέσα στην εμβέλειά του) και στέλνει μία απόκριση στον κόμβο από τον οποίο έλαβε τη αίτηση. Ο μηχανισμός απόκρισης για τους κόμβους του σχήματος 4.1 είναι ο εξής:

- Ο κόμβος b στέλνει απόκριση στον κόμβο a.
- Ο κόμβος c στέλνει απόκριση στον κόμβο b. Ο κόμβος b προωθεί την απόκριση στον κόμβο a.
- Ο κόμβος d στέλνει απόκριση στον κόμβο b. Ο κόμβος b προωθεί την απόκριση στον κόμβο a.
- Ο κόμβος a έχει την τοπολογία.

Σημειώνουμε ότι αν και οι κόμβοι γονείς μπορούν να ακούσουν τα «παιδιά» τους όταν προωθούν μία αίτηση (για παράδειγμα ο κόμβος a μπορεί να ξέρει για τον b όταν αυτός προωθεί την αίτηση) αυτό δεν είναι χρήσιμο αφού η πληροφορία της γειτονιάς τους είναι ατελής. Έτσι ένα αποκλειστικό πακέτο απόκρισης χρειάζεται για να στείλουμε την πληροφορία της γειτονιάς.

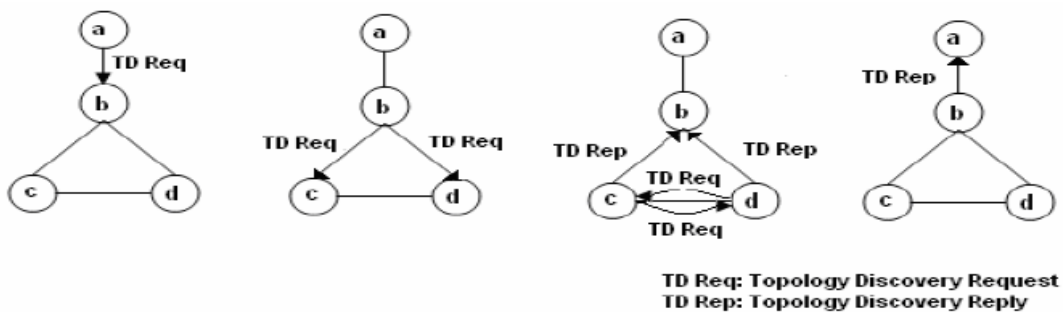


Σχήμα 4.2. Άμεση Απόκριση

2. Συνολική Απόκριση (Aggregated Response)

Όλοι οι ενεργοί κόμβοι στέλνουν μία Αίτηση για Ανακάλυψη Τοπολογίας αλλά περιμένουν τους κόμβους παιδιά να στείλουν απόκριση πριν στείλουν τις δικές τους αποκρίσεις. Αφού προωθήσει την Αίτηση για Ανακάλυψη Τοπολογίας, ένας κόμβος περιμένει να μάθει την γειτονιά του (neighborhood list) και τα παιδιά του ακούγοντας το κανάλι επικοινωνίας. Στη συνέχεια περιμένει να λάβει αποκρίσεις από τα παιδιά του. Όταν τις λάβει συναθροίζει τα δεδομένα και τα στέλνει στον πατέρα του. Ο μηχανισμός απόκρισης για τους κόμβους του σχήματος 4.1 είναι ο εξής:

- Οι κόμβοι c και d προωθούν την Αίτηση για Ανακάλυψη Τοπολογίας. Ο κόμβος b τους ακούει και συμπεραίνει ότι είναι παιδιά του.
- Ο κόμβος c στέλνει απόκριση στον κόμβο b. Ο κόμβος d στέλνει απόκριση στον κόμβο b.
- Ο κόμβος b συναθροίζει την πληροφορία από τους c, d και τον εαυτό του και προωθεί την απόκριση στον κόμβο a.
- Ο κόμβος a έχει την πλήρη τοπολογία.



Σχήμα 4.3. Συνολική Απόκριση

3. Απόκριση Ομάδας (Clustered Response)

Σε αυτήν την προσέγγιση το δίκτυο διαίρεται σε ομάδες (clusters). Κάθε ομάδα αντιπροσωπεύεται από έναν κόμβο (ονομάζεται αρχηγός ομάδας – cluster head) και κάθε κόμβος είναι μέλος τουλάχιστον μίας ομάδας. Έτσι κάθε κόμβος είναι στην εμβέλεια τουλάχιστον ενός αρχηγού ομάδας. Η απόκριση δημιουργείται μόνο από τους αρχηγούς ομάδας, οι οποίοι στέλνουν πληροφορίες για τους κόμβους της ομάδας τους. Όμοια με την συνολική απόκριση που είδαμε παραπάνω, οι αρχηγοί ομάδας μπορούν να συναθροίσουν πληροφορία από άλλους αρχηγούς ομάδας πριν στείλουν την απόκρισή τους. Ο μηχανισμός απόκρισης για τους κόμβους του σχήματος 4.1 είναι ο εξής:

- Υποθέστε ότι ο κόμβος b είναι ο αρχηγός ομάδας και οι κόμβοι c και d είναι μέλη της ομάδας.
- Οι κόμβοι c και d δεν στέλνουν απόκριση.
- Μόνο ο κόμβος b στέλνει απόκριση στον κόμβο a .
- Ο κόμβος a δεν παίρνει πληροφορία για τον σύνδεσμο $c \leftrightarrow d$.

Σημειώνουμε ότι η πληροφορία μπορεί να μην είναι πλήρης με την προσέγγιση της απόκρισης ομάδας. Οι μέθοδοι της άμεσης και συνολικής απόκρισης δίνουν μία ακριβή εικόνα της τοπολογίας του δικτύου. Η απόκριση ομάδας δίνει έναν reachability χάρτη (χάρτη προσέγγισης), όπου αν μπορούμε να φτάσουμε σε όλους τους αρχηγούς ομάδας, τότε μπορούμε να φτάσουμε σε όλους τους άλλους κόμβους από έναν τουλάχιστον αρχηγό. Ο αλγόριθμος *TopDisc* είναι αλγόριθμος που ανήκει σε αυτήν την προσέγγιση.

Από την άλλη μεριά το overhead που γλιτώνουμε στην ανακάλυψη τοπολογίας με την προσέγγιση της *Απόκρισης Ομάδας* είναι πολύ περισσότερο από την *Άμεση* ή την *Συνολική Απόκριση*.

4.3 ΑΠΟΚΡΙΣΗ ΟΜΑΔΑΣ ΚΑΙ ΑΛΓΟΡΙΘΜΟΣ TOPDISC

Θεωρούμε ένα δίκτυο αισθητήρων ως έναν μη κατευθυνόμενο γράφο $G\{V, E\}$ με V κορυφές και E ακμές. Στις προσεγγίσεις της απόκρισης ομάδας, οι κόμβοι του γράφου χωρίζονται σε σύνολα κόμβων. Ένας αρχηγός ομάδας αντιπροσωπεύει κάθε σύνολο και οι κόμβοι που ανήκουν στο σύνολό του ανήκουν στην γειτονιά του.

Έστω C ένα σύνολο με αρχηγούς ομάδων.

Έστω V_i η λίστα γειτονιάς του κόμβου i , με $i \in C$.

Τότε ισχύουν οι επόμενες συνθήκες:

1. $V = \cup V_i$
2. $\forall x \in V_i, \text{edge}(x, i) \in E$

Σημειώνουμε ότι το κόστος επικοινωνίας για την *Απόκριση Ομάδας* εξαρτάται από τον αριθμό των ομάδων και το μήκος του μονοπατιού που συνδέει τις ομάδες. Έτσι για ελάχιστο overhead επικοινωνίας θα πρέπει να λύσουμε τα παρακάτω προβλήματα:

- Να βρούμε ένα ελάχιστο $|C|$ (σύνολο με αρχηγούς ομάδων), που πρέπει να στείλει απόκριση. (Set Cover Problem)
- Να κατασκευάσουμε ένα ελάχιστο δένδρο με το σύνολο των αρχηγών ομάδων. (The Steiner Tree Problem)

Τα παραπάνω, είναι προβλήματα συνδυαστικής βελτιστοποίησης. Επιπλέον για βέλτιστη λύση, χρειαζόμαστε να έχουμε ολική πληροφορία για το δίκτυο ενώ στην περίπτωση μας οι κόμβοι έχουν μόνο τοπική πληροφορία. Δίνουμε ευρετικές (heuristics) συναρτήσεις, που προσφέρουν προσεγγιστικές λύσεις στα προβλήματα. Ένας από τους αλγόριθμους που προτείνεται, ο *TopDisc*, είναι απλός και κατανεμημένος και έτσι μπορεί να εφαρμοστεί στα δίκτυα αισθητήρων.

TopDisc

Ο TopDisc αλγόριθμος για την εύρεση των αρχηγών ομάδων βασίζεται στον απλό άπληστο $\log(n)$ -προσεγγιστικό αλγόριθμο για την εύρεση ενός cover set. Σε κάθε φάση ένας κόμβος επιλέγεται από τους κόμβους που έχουν ανακαλυφθεί, ο οποίος πρέπει να καλύπτει τον μέγιστο αριθμό από τους κόμβους που δεν έχουν ακόμα ανακαλυφθεί. Αφού στην περίπτωση της ανακάλυψης τοπολογίας τα σύνολα των γειτόνων και οι κορυφές του γράφου δεν είναι γνωστά στο χρόνο εκτέλεσης, δεν μπορούμε να έχουμε μία άμεση εφαρμογή του αλγορίθμου. Αντίθετα, τα σύνολα γειτονιάς πρέπει να δημιουργηθούν όσο η *Αίτηση για Ανακάλυψη Τοπολογίας* διαδίδεται μέσω του δικτύου. Χρησιμοποιούμε δύο προσεγγίσεις χρωματισμού των κόμβων για την εύρεση του συνόλου των αρχηγών ομάδων κατά τη διάρκεια της διάδοσης της Αίτησης: Η πρώτη χρησιμοποιεί τρία χρώματα και η δεύτερη χρησιμοποιεί τέσσερα. Ο μηχανισμός Απόκρισης είναι ο ίδιος και για τις δύο προσεγγίσεις.

4.3.1 Διάδοση της αίτησης χρησιμοποιώντας τρία χρώματα

Για να βρούμε το σύνολο των αρχηγών συνόλων χρησιμοποιούμε τρία χρώματα. Όλοι οι κόμβοι που λαμβάνουν μία *Αίτηση για Ανακάλυψη Τοπολογίας* και είναι ζωντανοί για να στείλουν απόκριση, θεωρούνται ως κόμβοι που έχουν ανακαλυφθεί.

- **Άσπρος κόμβος:** Κόμβος που δεν έχει ακόμα ανακαλυφθεί ή που δεν έχει λάβει ακόμα Αίτηση για Ανακάλυψη Τοπολογίας.
- **Μαύρος κόμβος:** Κόμβος αρχηγός ομάδας, ο οποίος στέλνει απόκριση με το σύνολο των γειτόνων του στην Αίτηση για Ανακάλυψη Τοπολογίας.

- **Γκρι κόμβος:** Κόμβος που καλύπτεται από τουλάχιστον έναν μαύρο κόμβο, δηλαδή είναι γείτονας ενός μαύρου κόμβου.

Αρχικά όλοι οι κόμβοι είναι άσπροι. Όταν η Αίτηση για Ανακάλυψη Τοπολογίας διαδίδεται, κάθε κόμβος χρωματίζεται μαύρος ή γκρι. Στο τέλος της αρχικής φάσης του αλγορίθμου, κάθε κόμβος στο δίκτυο είναι είτε μαύρος κόμβος είτε γείτονας ενός τουλάχιστον μαύρου κόμβου (γκρι κόμβος). Όλοι οι κόμβοι κάνουν broadcast το πακέτο αίτησης ακριβώς μία φορά στην αρχική φάση του αλγορίθμου. Έτσι όλοι οι κόμβοι έχουν την πληροφορία για τους γείτονές τους απλώς ακούγοντας αυτές τις μεταδόσεις.

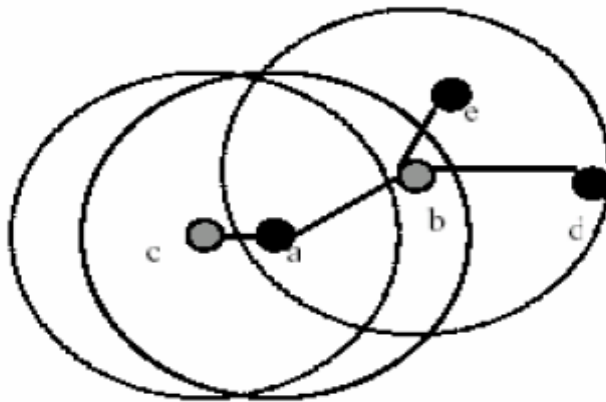
Χρησιμοποιούμε δύο ευρετικές συναρτήσεις (heuristics) με τις οποίες προσπαθούμε να πάρουμε το επόμενο σύνολο γειτόνων που ορίζεται από έναν κόμβο αρχηγό, ο οποίος πρέπει να καλύπτει τον μέγιστο αριθμό κόμβων που δεν έχουν ακόμα καλυφθεί. Η πρώτη χρησιμοποιεί έναν μηχανισμό χρωματισμού κόμβων για την εύρεση του απαιτούμενου συνόλου κόμβων. Η δεύτερη χρησιμοποιεί μία καθυστέρηση προώθησης αντιστρόφως ανάλογη με την απόσταση ανάμεσα στον αποστολέα και τον παραλήπτη. Αυτές οι δύο ευρετικές συναρτήσεις μας δίνουν μία λύση αρκετά κοντά στη λύση του κεντρικοποιημένου άπληστου αλγορίθμου για το cover set. Η διαδικασία αυτή φαίνεται παρακάτω:

- Ο κόμβος που αρχικοποιεί την Αίτηση για Ανακάλυψη Τοπολογίας χρωματίζεται μαύρος και κάνει broadcast την Αίτηση.
- Όλοι οι άσπροι κόμβοι γίνονται γκρι όταν λάβουν ένα πακέτο από έναν μαύρο κόμβο. Κάθε γκρι κόμβος κάνει broadcast την αίτηση σε όλους τους γείτονές του με μία τυχαία καθυστέρηση αντιστρόφως ανάλογη με την απόστασή του από τον μαύρο κόμβο από τον οποίο έλαβε το πακέτο.
- Όταν ένας άσπρος κόμβος λάβει ένα πακέτο από έναν γκρι κόμβο, γίνεται μαύρος με κάποια τυχαία καθυστέρηση. Εντωμεταξύ (στο μέσο χρόνο πριν να γίνει μαύρος) αν λάβει ένα πακέτο από έναν άλλο μαύρο κόμβο, χρωματίζεται γκρι. Πάλι η τυχαία καθυστέρηση είναι αντιστρόφως ανάλογη της απόστασης του από τον γκρι κόμβο από τον οποίο έλαβε το πακέτο Αίτησης.
- Από τη στιγμή που οι κόμβοι γίνουν μαύροι ή γκρι, αγνοούν τα άλλα πακέτα Αίτησης για Ανακάλυψη Τοπολογίας που λαμβάνουν.

Ένας καινούργιος μαύρος κόμβος πρέπει να επιλέγεται ώστε να καλύπτει τον μέγιστο αριθμό κόμβων που δεν έχουν ακόμα καλυφθεί. Αυτό επιτυγχάνεται έχοντας μία καθυστέρηση προώθησης, αντιστρόφως ανάλογη της απόστασης μεταξύ του κόμβου αποστολέα και του κόμβου παραλήπτη.

Η ιδέα να έχουμε μία καθυστέρηση προώθησης επεξηγείτε παρακάτω. Η περιοχή κάλυψης κάθε κόμβου είναι η κυκλική περιοχή που με κέντρο τον κόμβο και ακτίνα ίση με το εύρος επικοινωνίας. Τότε ο αριθμός των κόμβων που καλύπτονται από έναν κόμβο είναι ανάλογος της περιοχής κάλυψής του επί την τοπική πυκνότητα των κόμβων. Ο αριθμός των νέων κόμβων που καλύπτονται από έναν κόμβο που προωθεί (που κάνει broadcast την Αίτηση) είναι ανάλογος με την περιοχή κάλυψής του μείον την περιοχή

που έχει ήδη καλύψει. Αυτό απεικονίζεται στο σχήμα 4.4. Ο κόμβος *a* κάνει τους κόμβους *b* και *c* γκρι. Ο κόμβος *b* προωθεί (κάνει broadcast την Αίτηση) πριν τον *c* (αφού η απόσταση του *b* από τον *a* είναι μεγαλύτερη από την απόσταση του *c* από τον *a* και η καθυστέρηση είναι αντιστρόφως ανάλογη της απόστασης) και εξαιτίας αυτού περισσότεροι κόμβοι λαμβάνουν την αίτηση. Η καθυστέρηση θα κάνει τον *d* να γίνει μαύρος πιο πιθανά από τον *e*. Επίσης οι νέοι κόμβοι που καλύπτονται από τον *d* είναι πιθανόν να είναι περισσότεροι από αυτούς που καλύπτονται από τον *e*. Επιπλέον παρατηρούμε ότι ένας ενδιάμεσος κόμβος μεταξύ δύο μαύρων κόμβων (ο κόμβος *b* στο σχήμα 4.4) είναι πάντα μέσα στο εύρος και των δύο μαύρων κόμβων (δηλαδή δύο μαύροι κόμβοι απέχουν κατά ένα hop).



Σχήμα 4.4. Χρήση τριών χρωμάτων για καθυστερήσεις.

4.3.2 Διάδοση της αίτησης χρησιμοποιώντας τέσσερα χρώματα

Σε αυτήν την περίπτωση για να βρούμε τους αρχηγούς των ομάδων χρησιμοποιούμε τέσσερα χρώματα. Όλοι οι κόμβοι που λαμβάνουν μία *Αίτηση για Ανακάλυψη Τοπολογίας* και είναι ζωντανοί για να στείλουν απόκριση, θεωρούνται ως κόμβοι που έχουν ανακαλυφθεί. Και εδώ έχουμε άσπρους, μαύρους, και γκρι κόμβους που ορίζονται όπως στην προηγούμενη περίπτωση, αλλά επιπλέον έχουμε και τους *Dark Γκρι* κόμβους.

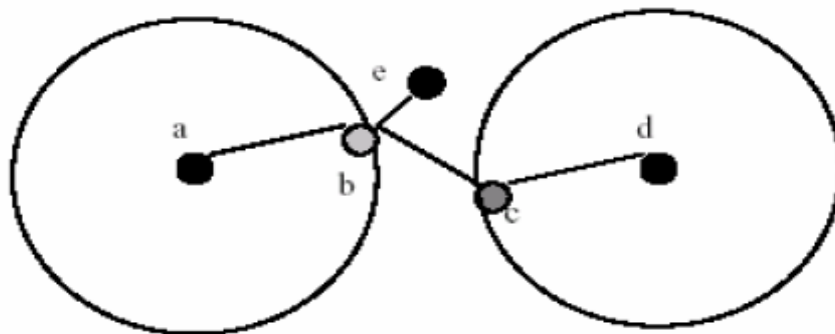
Dark Γκρι κόμβος: Κόμβος που έχει ανακαλυφθεί, και ο οποίος δεν καλύπτεται από κανέναν γειτονικό μαύρο κόμβο και έτσι απέχει δύο hops από τον μαύρο κόμβο. Ένας άσπρος κόμβος γίνεται dark γκρι όταν λαμβάνει μία αίτηση από έναν γκρι κόμβο.

Αρχικά όλοι οι κόμβοι είναι άσπροι. Όταν η Αίτηση για Ανακάλυψη Τοπολογίας διαδίδεται, κάθε κόμβος χρωματίζεται μαύρος, γκρι, ή dark γκρι. Στο τέλος, κάθε κόμβος στο δίκτυο είναι είτε μαύρος κόμβος είτε γείτονας ενός τουλάχιστον μαύρου κόμβου (γκρι κόμβος). Η διαδικασία αυτή φαίνεται παρακάτω:

- Ο κόμβος που αρχικοποιεί την Αίτηση για Ανακάλυψη Τοπολογίας χρωματίζεται μαύρος και κάνει broadcast την Αίτηση.
- Όλοι οι άσπροι κόμβοι γίνονται γκρι όταν λάβουν ένα πακέτο από έναν μαύρο κόμβο. Κάθε γκρι κόμβος κάνει broadcast την αίτηση σε όλους τους γείτονές του με μία τυχαία καθυστέρηση αντιστρόφως ανάλογη με την απόστασή του από τον μαύρο κόμβο από τον οποίο έλαβε το πακέτο.
- Όταν ένας άσπρος κόμβος λάβει ένα πακέτο από έναν γκρι κόμβο, γίνεται dark γκρι. Κάνει broadcast την Αίτηση σε όλους τους γείτονές του και ξεκινάει έναν μετρητή για να γίνει μαύρος κόμβος. Η καθυστέρηση προώθησης είναι αντιστρόφως ανάλογη με την απόστασή του από τον γκρι κόμβο από τον οποίο έλαβε το πακέτο
- Όταν ένας άσπρος κόμβος λάβει ένα πακέτο από έναν dark γκρι κόμβο, γίνεται μαύρος με κάποια τυχαία καθυστέρηση. Εντωμεταξύ (στο μέσο χρόνο πριν να γίνει μαύρος) αν λάβει ένα πακέτο από έναν άλλο μαύρο κόμβο, χρωματίζεται γκρι.
- Ένας dark γκρι κόμβος περιμένει κάποιο χρόνο έτσι ώστε ένας από τους γείτονές του να γίνει μαύρος. Όταν ο χρόνος λήξει (ο μετρητής μηδενιστεί) γίνεται μαύρος ο ίδιος γιατί δεν υπάρχει μαύρος κόμβος να τον καλύψει.
- Από τη στιγμή που οι κόμβοι γίνουν μαύροι ή γκρι, αγνοούν τα άλλα πακέτα Αίτησης για Ανακάλυψη Τοπολογίας που λαμβάνουν.

Η ιδέα για την χρήση τεσσάρων χρωμάτων αντί τριών είναι η εξής: Ένας καινούριος μαύρος κόμβος πρέπει να επιλεγεί ώστε να καλύπτει τον μέγιστο αριθμό κόμβων που δεν έχουν ακόμα καλυφθεί. Γι' αυτό θέλουμε τους μαύρους κόμβους να είναι χωρισμένοι ο ένας από τον άλλον δύο hops (όπως ο *a* και ο *b*) έτσι ώστε οι κόμβοι να ανήκουν στην γειτονιά (στην ομάδα) ενός μόνο μαύρου κόμβου. Αυτό μπορεί να μην είναι δυνατόν σε όλες τις περιπτώσεις και έτσι μερικοί μαύροι κόμβοι να απέχουν μόνο ένα hop ο ένας από τον άλλο (όπως ο κόμβος *e* στο σχήμα 4.5).

Ο αριθμός των ομάδων είναι λιγότερος από την προσέγγιση με τα τρία χρώματα. Όμως η προσέγγιση των τριών χρωμάτων δημιουργεί δέντρο TreC (δένδρο ομάδων) πιο κατάλληλο για εφαρμογές διαχείρισης δικτύων.



Σχήμα 4.5. Χρήση τεσσάρων χρωμάτων για καθυστέρηση

4.3.3 Μηχανισμός απόκρισης του TopDisc

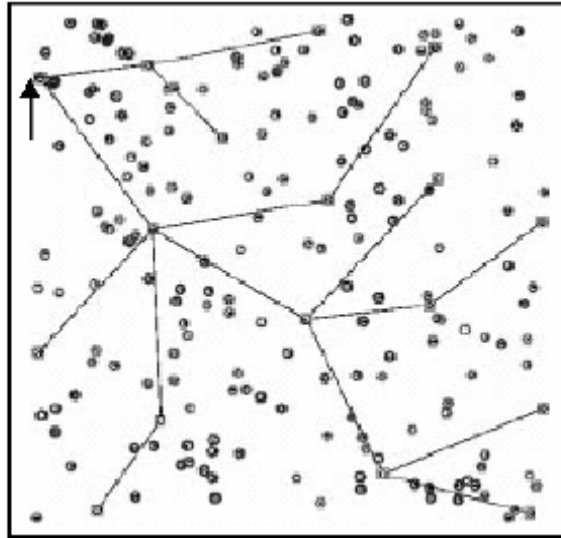
Στην πρώτη φάση του αλγορίθμου δώσαμε χρώματα στους κόμβους. Ο κόμβος αρχικοποίησης έγινε η ρίζα του δένδρου των μαύρων κόμβων στο οποίο οι πατέρες μαύροι κόμβοι είναι το πολύ δύο hops μακριά (χρησιμοποιώντας τέσσερα χρώματα) και ένα hop μακριά (χρησιμοποιώντας τρία χρώματα) από τους μαύρους κόμβους παιδιά τους. Κάθε κόμβος έχει τις ακόλουθες πληροφορίες στο τέλος αυτής της περιόδου:

- Μία ομάδα ορίζεται από τον μαύρο κόμβο, στην οποία είναι αρχηγός
- Ένας γκρι κόμβος γνωρίζει το *id* της ομάδας του.
- Κάθε κόμβος ξέρει τον μαύρο κόμβο πατέρα του, ο οποίος είναι ο τελευταίος μαύρος κόμβος από τον οποίο προωθήθηκε Αίτηση για Ανακάλυψη Τοπολογίας για να τον φτάσει.
- Κάθε μαύρος κόμβος γνωρίζει τον default κόμβο στον οποίο πρέπει να προωθήσει πακέτα για να φτάσει τον μαύρο κόμβο πατέρα του. Αυτός ο κόμβος είναι ο κόμβος από τον οποίο έλαβε την Αίτηση για Ανακάλυψη Τοπολογίας.
- Όλοι οι κόμβοι έχουν πληροφορία για τους γείτονές τους.

Χρησιμοποιώντας τις παραπάνω πληροφορίες, τα βήματα για την *TopDisc Απόκριση* παρουσιάζονται παρακάτω:

- Όταν ένας κόμβος γίνεται μαύρος, ξεκινάει έναν μετρητή για να στείλει Απόκριση στην Αίτηση για Ανακάλυψη Τοπολογίας. Κάθε μαύρος κόμβος περιμένει για αυτήν την χρονική περίοδο κατά την διάρκεια της οποίας λαμβάνει αποκρίσεις από τους μαύρους κόμβους παιδιά του.
- Συναθροίζει όλες τις λίστες γειτόνων από τα παιδιά του και του εαυτού του και όταν ο χρονική περίοδος που περιμένει (ο μετρητής μηδενιστεί) λήξει, προωθεί την συνολική λίστα γειτόνων στον πατέρα του μέσω του default κόμβου.

Για να δουλέψει σωστά ο αλγόριθμος θα πρέπει τα timeout να ορίζονται κατάλληλα. Για παράδειγμα θα πρέπει τα timeout των μαύρων κόμβων παιδιών να λήγουν πάντα πριν το timeout του πατέρα. Για αυτό δίνουμε τιμή στα timeout αντιστρόφως ανάλογη με τον αριθμό των hops που ένας μαύρος κόμβος απέχει από τον κόμβο αρχικοποίησης. Το σχήμα 4.6 δείχνει ένα τυπικό TreC (δένδρο ομάδων) που προέκυψε από τον TopDisc.



Σχήμα 4.6. TreC για 200 κόμβους και εύρος 20 μέτρα. Το βέλος αναπαριστά τον κόμβο αρχικοποίησης.

4.3.4 Εφαρμογές του TopDisc

Ο βασικός σκοπός της ανακάλυψης τοπολογίας είναι να παρέχει στον διαχειριστή δικτύου την τοπολογία του δικτύου. Ο μηχανισμός TopDisc παρέχει τον reachability χάρτη της περιοχής. Επιπλέον, με μία παραλλαγή, οι κόμβοι μπορούν όταν προωθούν την Αίτηση για Ανακάλυψη Τοπολογίας να στέλνουν και την διαθέσιμη ενέργειά τους. Έτσι τελικά μαζί με τον reachability χάρτη θα είναι γνωστός και ο ενεργειακός χάρτης της περιοχής. Εφόσον είναι γνωστές αυτές οι πληροφορίες στον διαχειριστή, μπορεί να γίνει δρομολόγηση η οποία θα είναι πιο στιβαρή (robust), ενεργειακά αποδοτική (energy efficient) και πιο γρήγορη.

5 ΑΥΤΟ-ΔΙΑΜΟΡΦΩΣΗ ΚΑΙ ASCENT

Σε αυτό το κεφάλαιο, θα προσεγγίσουμε το θέμα της αυτό-διαμόρφωσης (self-configuration) της αυτό-διαχείρισης από μία διαφορετική σκοπιά. Σε αυτή τη περίπτωση κάθε κόμβος θα πρέπει να είναι σε θέση να ανακαλύψει τη συμμετοχή του σε μία multi-hop δικτυακή τοπολογία και το τρόπο που πρέπει να συνεισφέρει ώστε για παράδειγμα το δίκτυο να μπορεί με στιβαρότητα και ενεργειακή αποδοτικότητα να προωθήσει (propagation) μια πληροφορία σε κάποιο σταθμό βάσης. Πιο συγκεκριμένα θα αναλύσουμε το πρωτόκολλο ASCENT που είναι σε θέση να δώσει με αποδοτικό τρόπο το ρόλο του κάθε κόμβου στο δίκτυο.

Το πρωτόκολλο ASCENT (Adaptive Self-Configuring Sensor Network Topologies) λοιπόν, είναι όπως φαίνεται και από το όνομα του ένα πρωτόκολλο για self-configuration του δικτύου αισθητήρων και στην ουσία καθοδηγεί κάποιο κόμβο για τον ενεργό ρόλο του στο δίκτυο. Πρέπει να σημειώσουμε ότι το ASCENT δεν είναι ένα πρωτόκολλο δρομολόγησης ή ένα πρωτόκολλο διασποράς δεδομένων στο δίκτυο. Το ASCENT απλά αποφασίζει ποιος κόμβος θα συμμετάσχει στην υποδομή δρομολόγησης του δικτύου. Δηλαδή ουσιαστικά αποφασίζει πιο κόμβος πρέπει να συνεισφέρει στη δρομολόγηση πακέτων στο δίκτυο έτσι ώστε να επιτευχθεί μικρή κατανάλωση ενέργειας και μακροζωία του δικτύου.

Τα χαρακτηριστικά του δικτύου στα οποία καλείται να προσαρμοστεί το πρωτόκολλο ASCENT είναι συνοπτικά τα εξής:

1. **Ad-hoc δομή δικτύου:** Δεν μπορούμε να περιμένουμε στο δίκτυο αισθητήρων οι κόμβοι να έχουν ομοιόμορφη κατανομή στο χώρο.
2. **Περιορισμοί ενέργειας:** Το δίκτυο πρέπει να κάνει οικονομία στους ενεργειακούς πόρους ώστε να αυξήσει τη διάρκεια ζωής τους.
3. **Dynamics:** Όπως αναφέραμε παραπάνω τόσο τα system όσο και τα environmental dynamics αποτελούν περιοριστικό παράδειγμα για τα sensor δίκτυα.

Με βάση τα όσα περιγράψαμε παραπάνω φαίνεται πως μπορεί να είναι πιο εύκολο να τοποθετήσουμε από την αρχή ένα μεγάλο αριθμό κόμβων παρά να «ενισχύσουμε» στη συνέχεια το δίκτυο με περισσότερους κόμβους. Όμως το ερώτημα είναι «Πόσο μεγάλος αριθμός κόμβος πρέπει να παραταχθεί;». Αν τοποθετήσουμε λίγους sensor κόμβους, η απόσταση μεταξύ αυτών θα είναι πολύ μεγάλη και έτσι ο ρυθμός απώλειας πακέτων θα αυξηθεί. Επίσης απαγορευτική θα είναι και η απαιτούμενη ενέργεια για τη διάδοση της πληροφορίας. Από την άλλη πλευρά, αν χρησιμοποιήσουμε ένα μεγάλο αριθμό κόμβων ταυτόχρονα, τότε μπορεί να καταναλώνουμε την ενέργεια των κόμβων άσκοπα, και επίσης μπορούμε να δημιουργήσουμε και παρεμβολές μεταξύ γειτονικών κόμβων. Το ASCENT λοιπόν είναι ένα προσαρμοστικό self-configuration πρωτόκολλο που προσπαθεί να βρει μια ισορροπία για τον ενεργητικό ρόλο των κόμβων.

Η ανάλυση του πρωτοκόλλου που παραθέτουμε παρακάτω κάνει κάποιες βασικές υποθέσεις. Αρχικά υποθέτει τη χρήση ενός πρωτοκόλλου CSMA για το MAC υπό-επίπεδο. Αυτή η υπόθεση εισάγει ξεκάθαρα το ενδεχόμενο για ανταγωνισμό πόρων όταν πάρα πολλοί γείτονες συμμετέχουν στο multi-hop δίκτυο. Μία άλλη παραδοχή είναι η αντίδραση του πρωτοκόλλου όταν υπάρχει μεγάλη απώλεια πακέτων. Τέλος το πρωτόκολλο ASCENT δεν ανιχνεύει ή αντιμετωπίζει ολόκληρα τμήματα του δικτύου μιας και θεωρούμε το δίκτυο αρκετά πυκνό ώστε να μην υπάρχουν κόμβοι απομονωμένοι από το υπόλοιπο κομμάτι του δικτύου.

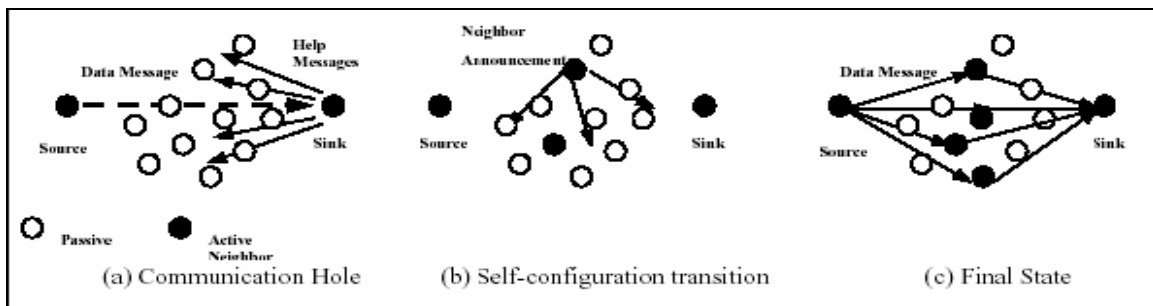
Πριν κλείσουμε αυτή τη παράγραφο πρέπει να σημειώσουμε ότι το ASCENT πρωτόκολλο δεν είναι απαραίτητο όταν η πυκνότητα (density) του δικτύου είναι μικρή, μιας και σε αυτή τη περίπτωση πρέπει να επιστρατευτούν όλοι οι κόμβοι για να συνεισφέρουν στη διάδοση της πληροφορίας. Μόνο αν το δίκτυο είναι πυκνό επιστρατεύουμε το ASCENT για να γίνεται οικονομία στην ενέργεια του δικτύου.

5.1 ASCENT

Ο αλγόριθμος ASCENT αποτελείται από φάσεις. Όταν ένας κόμβος αρχικοποιείται, μπαίνει σε μία φάση όπου “ακούει” (*listening-only phase*) η οποία ονομάζεται *neighbor discovery phase* (Φάση Ανακάλυψης Γειτόνων) και στην οποία κάθε κόμβος κάνει μία εκτίμηση του αριθμού των γειτόνων του που μεταδίδουν μηνύματα, βασιζόμενος σε τοπικές μετρήσεις. Μετά το τέλος αυτής της φάσης, ο κόμβος μπαίνει στη *join decision phase* (Φάση Απόφασης Συμμετοχής), όπου αποφασίζει αν θα συμμετέχει στο multi-hop δίκτυο αισθητήρων. Σε αυτή τη φάση, ο κόμβος μπορεί να συμμετέχει στο δίκτυο προσωρινά για μία περίοδο, για να ελέγξει αν βοηθάει στη βελτίωση της συνδεσιμότητας. Αν αποφασίσει να συμμετέχει για μεγαλύτερο χρονικό διάστημα, μπαίνει στη *active phase* (ενεργή φάση), και ξεκινάει να στέλνει μηνύματα ελέγχου δρομολόγησης και δεδομένων. Αν ο κόμβος αποφασίσει να μην συμμετέχει στο δίκτυο, μπαίνει σε στην *adaptive phase* (προσαρμοστική φάση), όπου σταματάει να λειτουργεί για κάποιο χρονικό διάστημα ή μειώνει το εύρος μετάδοσής του (την ισχύ του).

Το σχήμα 5.1, απεικονίζει τον αλγόριθμο ASCENT κατά τη διάρκεια της αρχικοποίησης σε μία περιοχή υψηλής πυκνότητας (με πολλούς κόμβους). Αρχικά., μόνο μερικοί κόμβοι συμμετέχουν στο multi-hop δίκτυο. Οι άλλοι απλά “ακούν” τα μηνύματα αλλά δεν μεταδίδουν. Η πηγή (source) ξεκινά να μεταδίδει δεδομένα στον προορισμό (sink), όπως φαίνεται στο 5.1(a). Επειδή ο προορισμός βρίσκεται πολύ μακριά από την πηγή, έχει μεγάλο message loss (απώλεια μηνύματος). Καλούμε αυτήν την κατάσταση *communication hole* (τρύπα επικοινωνίας), δηλαδή ο παραλήπτης έχει υψηλό message loss εξαιτίας της φτώχης συνδεσιμότητας με τον αποστολέα. Ο προορισμός τότε, ξεκινά να στέλνει *help messages* για να ειδοποιήσει τους γείτονες που βρίσκονται στη φάση που “ακούνε” (*listening-only phase*) να συμμετέχουν στο δίκτυο. Όταν ένας γείτονας λάβει ένα *help message*, μπορεί να αποφασίσει να συμμετέχει στο δίκτυο. Αυτό φαίνεται στο 5.1(b). Όταν ένας κόμβος συμμετάσχει στο δίκτυο, ξεκινά να στέλνει και να λαμβάνει μηνύματα, γίνεται ένας ενεργός γείτονας (*active neighbor*) και ειδοποιεί τους άλλους *passive* γείτονες (που μόνο “ακούνε”) ότι έγινε ενεργός, στέλνοντας τους ένα μήνυμα ανακοίνωσης γείτονα. Η διαδικασία αυτή συνεχίζεται μέχρι ο αριθμός των *passive* κόμβων που γίνονται *active* σταθεροποιηθεί σε μία συγκεκριμένη τιμή. Όταν αυτή η διαδικασία ολοκληρωθεί, οι νέοι *active* γείτονες που συμμετέχουν στο δίκτυο, κάνουν την μεταφορά δεδομένων από την πηγή στον προορισμό πιο αξιόπιστη.

Στη συνέχεια θα μιλήσουμε για τις φάσεις του ASCENT αλγορίθμου.



Σχήμα 5.1. Self-Configurable WSN.

5.1.1 Neighbor discovery phase (Φάση ανακάλυψης γείτονα)

Η αύξηση του αριθμού των γειτόνων μπορεί να οδηγήσει σε κατανάλωση ενέργειας εξαιτίας του ανταγωνισμού για τους πόρους (για το κανάλι). Έτσι είναι πολύ σημαντικό κάθε κόμβος να ξέρει πόσοι είναι περίπου οι γείτονες στην περιοχή του, έτσι ώστε να μπορεί να αυτό-ρυθμιστεί (self-configure), βασιζόμενος στον αριθμό των κόμβο που ανταγωνίζονται το κανάλι. Αυτό μας οδηγεί σε κάποια ερωτήματα όπως: Ποιος μπορεί να θεωρηθεί γείτονας και πώς μπορούμε να καθορίσουμε ότι ένας κόμβος είναι γείτονας χρησιμοποιώντας μόνο τοπική πληροφορία;

Ορίζουμε γείτονα έναν κόμβο από τον οποίο λαμβάνουμε ένα συγκεκριμένο ποσοστό μηνυμάτων σε ένα χρονικό διάστημα.

Όταν ένας κόμβος αρχικοποιείται, προσπαθεί να βρει μία προσέγγιση του αριθμού των active γειτόνων του. Οι κόμβοι σε αυτήν την φάση (neighbor discovery phase) είναι passive γείτονες, δηλαδή μόνο ακούν (λαμβάνουν) τα μηνύματα αλλά δεν κάνουν μεταδόσεις. Ο βασικός ψευδοκώδικας του αλγορίθμου που τρέχει μέχρι την λήψη ενός μηνύματος φαίνεται στο σχήμα 5.2(a). Έχουν αναπτυχθεί δύο εκδόσεις του αλγορίθμου ASCENT που βασίζονται σε διαφορετικά κατώφλια απώλειας (loss threshold levels). Στην πρώτη έκδοση έχουμε ένα σταθερό (fixed) κατώφλι απώλειας που δεν αλλάζει, ενώ στην δεύτερη το κατώφλι απώλειας αλλάζει ανάλογα με τον αριθμό των γειτόνων που υπολογίζονται. Αυτό φαίνεται στο 5.2(b).

<pre> while (end_discovery_phase()) { messg = check_message_queue(); if (messg) { add_neighbor_list (messg); messg = NULL; } num_neighbors = total_active_neighbors(); } </pre>	$\text{LossThreshold} = 1 - \frac{2}{N} \quad \text{for } N \geq 3$ $\text{LossThreshold} = 0.3 \quad \text{for } N < 3$ <i>N: number of active neighbors</i>
(a)	(b)

Σχήμα 5.2. (a) Neighbor Discovery. (b) Thresholds.

5.1.2 Join decision phase (Φάση απόφασης συμμετοχής)

Όπως έχουμε αναφέρει, σκοπός αυτής της φάσης είναι να αποφασίσει αν οι κόμβοι θα συμμετέχουν στο multi-hop δίκτυο αισθητήρων. Μπορούν να χρησιμοποιηθούν πολλές μεταβλητές για να μας βοηθήσουν σε αυτήν την απόφαση, όπως είναι η ενέργεια και το message loss.

Έχουμε πει ότι η αύξηση του αριθμού των γειτόνων μπορεί να οδηγήσει σε κατανάλωση ενέργειας εξαιτίας του ανταγωνισμού για τους πόρους (για το κανάλι).

Εξαιτίας αυτού, μία από τους μεταβλητές για απόφαση είναι το πόσο πολλοί είναι οι γείτονες (αριθμός των active γειτόνων). Επιπλέον, ένας από τους στόχους μας είναι να παρέχουμε πλήρη κάλυψη επικοινωνίας για το δίκτυο αισθητήρων. Έτσι λοιπόν η άλλη μεταβλητή που χρησιμοποιούμε για την απόφαση συμμετοχής είναι το data message loss. Εκτός από αυτές τις μεταβλητές που είναι τοπικές, χρησιμοποιούμε και εξωτερικά σήματα που προέρχονται από τους γείτονες, και είναι τα help messages, τα οποία έχουμε αναφέρει προηγουμένως.

Οι κόμβοι, πριν μπουν σε αυτήν την φάση, έχουν ήδη μία εκτίμηση του αριθμού των active γειτόνων (NB) τους καθώς και του data message loss (ML). Ο πίνακας που ακολουθεί δείχνει τη βασική λειτουργία της απόφασης. Κάθε μεταβλητή (ο αριθμός των active γειτόνων και το data message loss) μπορεί να πάρει δύο τιμές, high και low. Όταν ο αριθμός των active γειτόνων (NB) είναι μεγάλος (high) τότε αν προσθέσουμε και άλλον γείτονα στο δίκτυο (αν δηλαδή ένας passive κόμβος γίνει active) το μόνο που θα κάνουμε θα είναι να δημιουργήσουμε επιπλέον παρεμβολή και ανταγωνισμό για το κανάλι. Γι' αυτό το λόγο όταν το NB είναι υψηλό, όπως βλέπουμε και στον πίνακα, ο κόμβος δεν συμμετέχει στο δίκτυο. Στην περίπτωση που το NB είναι χαμηλό και το data message loss (ML) είναι υψηλό, υποθέτουμε ότι υπάρχει τρύπα επικοινωνίας (communication hole) και ο κόμβος συμμετέχει στο δίκτυο. Αν το NB είναι χαμηλό και το ML χαμηλό, ο κόμβος συμμετέχει στο δίκτυο μόνο αν έχει λάβει ("ακούσει") help messages και συμμετέχει προσωρινά για ένα χρονικό διάστημα. Αν περάσει αυτό το χρονικό διάστημα και το ML συνεχίζει να είναι χαμηλό, τότε ο κόμβος συμμετέχει μόνιμα στο δίκτυο και γίνεται active, αφού υποθέτει ότι συνεισφέρει θετικά. Αν όμως το ML μετά από αυτό το χρονικό διάστημα έχει γίνει υψηλό τότε ο κόμβος δεν συμμετέχει στο δίκτυο και περνάει στην *προσαρμοστική φάση (adaptive phase)*.

Πίνακας 5.1. Λειτουργία Join Decision.

Message Loss (ML)	Number of Neighbors (NB)	
	LOW	HIGH
LOW	If received help message, then temporarily joins	Do not join
HIGH	Join	Do not join

5.1.3 Active & adaptive phase (Ενεργή και προσαρμοστική φάση)

Όσοι κόμβοι είναι στη active κατάσταση στέλνουν και λαμβάνουν όλα τα μηνύματα ελέγχου δρομολόγησης. Όταν ένας κόμβος συμμετάσχει στο δίκτυο (γίνεται active) στέλνει αμέσως ένα μήνυμα ανακοίνωσης γείτονα με το οποίο ειδοποιεί τους passive γείτονες να αυξήσουν τους ενεργούς γείτονές τους κατά έναν. Όταν ένας κόμβος που συμμετέχει ήδη στο multi-hop δίκτυο ανιχνεύσει υψηλό data message loss στέλνει help messages στους passive κόμβους.

Όσοι κόμβοι αποφασίσουν να μην συμμετάσχουν στο δίκτυο, περνούν στην προσαρμοστική (adaptive) φάση. Οι κόμβοι που βρίσκονται σε αυτήν την φάση, σταματούν να λειτουργούν για ένα χρονικό διάστημα για να μην καταναλώνουν ενέργεια

και να παρατείνουν τη ζωή του δικτύου. Μπορούν επιπλέον να μειώσουν το εύρος μετάδοσής τους ώστε να μειώσουν τις παρεμβολές.

ΜΕΡΟΣ 4^ο : ΠΡΩΤΟΚΟΛΛΑ LOCALIZATION

6 ΕΙΣΑΓΩΓΗ ΣΤΟ LOCALIZATION

Τα πρόσφατα τεχνολογικά επιτεύγματα ενθάρρυναν την ανάπτυξη μικρών, χαμηλού κόστους συσκευών που ενσωματώνουν μικρό-αισθητήρες μικρής ισχύος, επεξεργαστές και δυνατότητα για ασύρματη επικοινωνία. Όταν παρατάζουμε έναν μεγάλο αριθμό από αυτούς τους μικρούς κόμβους σε μία περιοχή, αυτοί έχουν την ικανότητα να μετρούν διάφορα χαρακτηριστικά του φυσικού περιβάλλοντος (π.χ. πίεση, θερμοκρασία, κίνηση κ.α) με πολύ μεγάλη λεπτομέρεια. Έτσι λοιπόν αν καταφέρουμε να συντονίσουμε τους κόμβους αυτού δικτύου αισθητήρων (sensor network) θα είμαστε σε θέση να μελετούμε με ένα επαναστατικό τρόπο οποιοδήποτε φυσικό σύστημα.

Όπως έχουμε αναφέρει, υπάρχει μία πληθώρα εφαρμογών για τα δίκτυα αισθητήρων όπως: 1) Στρατιωτικές Εφαρμογές (επικοινωνίες, στρατιωτική παρακολούθηση κ.α), 2) Εφαρμογές Περιβάλλοντος (ανίχνευση πυρκαγιάς σε δάσος, παρακολούθηση καλλιεργειών κ.α), 3) Εφαρμογές Υγείας (παρακολούθηση γιατρών και ασθενών κ.α). 4) Εφαρμογές Κατοικίας (αυτόματο σπίτι κ.α). 5) Διάφορες Εμπορικές Εφαρμογές (ανίχνευση και παρακολούθηση κλοπών αυτοκινήτων κ.α). Όμως για όλες τις παραπάνω εφαρμογές είναι απαραίτητο να έχουμε ένα προσανατολισμό για το που βρίσκονται αυτοί οι κόμβοι σε σχέση με ένα γνωστό παγκόσμιο σύστημα συντεταγμένων. Με αυτό τον τρόπο κάθε πληροφορία που θα προέρχεται από το δίκτυο αισθητήρων θα χαρακτηρίζεται από μια γεωγραφική τοποθεσία ή καλύτερα από ένα σύνολο συντεταγμένων κάποιου γνωστού συστήματος συντεταγμένων, η οποία μπορεί να αναλυθεί και να αξιοποιηθεί ανάλογα με την εφαρμογή.

Τα Ad hoc δίκτυα αισθητήρων υπόκεινται σε πολλούς περιορισμούς κατά το σχεδιασμό τους. Από τη μία μεριά, το χαμηλό κόστος των κόμβων δίνει τη δυνατότητα μεγάλης κλιμάκωσης του δικτύου και υψηλού υπολογιστικού παραλληλισμού. Από την άλλη μεριά κάθε κόμβος έχει περιορισμένη ενέργεια, μικρή αξιοπιστία και τοπική επικοινωνία με ένα μικρό αριθμό γειτονικών κόμβων. Τα παραπάνω χαρακτηριστικά σε συνδυασμό με το είδος των εφαρμογών των sensor networks δεν επιτρέπουν προσεκτική, ομοιόμορφη τοποθέτηση των κόμβων στο χώρο του δικτύου. Επίσης η τοποθέτηση κάποιας τεχνολογίας GPS (Global Positioning System) στους κόμβους του sensor network είναι απαγορευτική λόγω κόστους. Συμπερασματικά λοιπόν το δίκτυο αισθητήρων πρέπει να είναι σε θέση να «αυτό-οργανώνεται» και να προσδιορίζει μόνο του τις συντεταγμένες των κόμβων του.

Σε αυτό το μέρος θα κάνουμε μία περιγραφή του υλικού και των τεχνολογιών που χρησιμοποιούνται στους αλγόριθμους εντοπισμού (localization algorithms) και επίσης θα περιγράψουμε τις κύριες κατηγορίες αυτών των αλγορίθμων. Ιδιαίτερη προσοχή θα δώσουμε στον αλγόριθμο των M. Sichitiu και V. Ramadurai [19] όπου θεωρούμε ότι έχουμε ένα μόνο κινητό beacon κόμβο. Τέλος θα κάνουμε μία σύντομη κριτική σε αυτούς τους localization αλγόριθμους και θα δώσουμε κάποιες μελλοντικές κατευθύνσεις.

7 LOCALIZATION HARDWARE

Το πρόβλημα εντοπισμού (localization problem) καλείται να λύσει δύο σημαντικά προβλήματα. Το πρώτο είναι το πρόβλημα ορισμού ενός συστήματος συντεταγμένων και καλύπτεται στη παράγραφο 7.1. Το δεύτερο πρόβλημα αναφέρεται στον υπολογισμό της απόστασης μεταξύ δύο sensor κόμβων (ranging πρόβλημα) και καλύπτεται στο επόμενο του κεφαλαίου.

7.1 BEACON ΚΟΜΒΟΙ (ΚΟΜΒΟΙ ΑΓΚΥΡΑΣ)

Στόχος του Localization είναι ο καθορισμός φυσικών συντεταγμένων για μια ομάδα κόμβων αισθητήρων. Αυτές οι συντεταγμένες μπορεί να είναι *καθολικές* (global), δηλαδή μπορεί να συμμορφώνονται με κοινά αποδεκτές συντεταγμένες κάποιου GPS συστήματος, ή να είναι *σχετικές* (relative) δηλαδή να αποτελούν έναν αυθαίρετο «άκαμπτο μετασχηματισμό» κάποιων GPS συντεταγμένων. Οι σχετικές συντεταγμένες έχουν μόνο νόημα τοπικά.

Οι κόμβοι *beacon* (ή κόμβοι άγκυρας όπως συχνά ονομάζονται) αποτελούν απαραίτητη προϋπόθεση για να γίνει localization κάπου δικτύου σε ένα σύστημα καθολικών συντεταγμένων. Οι beacon κόμβοι είναι λοιπόν συνηθισμένοι sensor κόμβοι οι οποίοι όμως ξέρουν τις καθολικές συντεταγμένες τους εκ των προτέρων. Αυτή η πληροφορία μπορεί να έχει γραφτεί στη μνήμη των beacon κόμβων (αν βέβαια είναι γνωστή η θέση τους εκ των προτέρων) ή οι beacon κόμβοι μπορεί να διαθέτουν ένα GPS δέκτη. Το ελάχιστο, τρεις μη-συγγραμικοί beacon κόμβοι απαιτούνται για να οριστούν οι καθολικές συντεταγμένες σε δίκτυο δύο διαστάσεων. Αν μιλάμε για τρισδιάστατες συντεταγμένες τότε απαιτούνται το λιγότερο τέσσερις μη-συγγραμικοί beacon κόμβοι.

Ένα πολύ σημαντικό ζήτημα για την αποδοτικότητα του localization αλγόριθμου είναι ο τρόπος με τον οποίο θα τοποθετηθούν οι beacon κόμβοι. Πολλά πειράματα έχουν δείξει ότι ο καθορισμός συντεταγμένων γίνεται πιο ακριβής εάν οι beacon κόμβοι τοποθετηθούν στο κυρτό περίγραμμα γύρω από το δίκτυο. Επίσης σημαντική βελτίωση παρατηρείται με την τοποθέτηση beacon κόμβων στη μέση του δικτύου.

Το βασικό και προφανές πλεονέκτημα της χρησιμοποίησης πολλών beacon κόμβων, είναι η απλούστευση του προσδιορισμού των συντεταγμένων των κόμβων του δικτύου. Όμως υπάρχουν και σοβαρά μειονεκτήματα. Κατ' αρχήν οι δέκτες GPS είναι εξαιρετικά ακριβοί. Επίσης οι δέκτες GPS καταναλώνουν μεγάλα ποσά ενέργειας για να λειτουργήσουν γεγονός που κάνει την εφαρμογή τους σε sensor κόμβους δύσκολη μιας και αυτοί έχουν μόνη πηγή ενέργειας μια μπαταρία. Όπως προαναφέραμε μια εναλλακτική λύση αντί του GPS είναι κόμβοι οι οποίοι θα είναι προγραμματισμένοι εκ των προτέρων για την τοποθεσία τους. Όμως αυτό είναι πολλές φορές αδύνατο, όπως στη περίπτωση που ρίχνουμε τους κόμβους αισθητήρων από ένα αεροπλάνο. Συμπερασματικά οι χρήση των beacon κόμβων είναι απαραίτητη για το προσδιορισμό καθολικών συντεταγμένων, αλλά όμως πρέπει να είναι περιορισμένη μιας και είναι πολύ ακριβή.

7.2 ΈΝΔΕΙΞΗ ΙΣΧΥΟΣ ΛΑΜΒΑΝΟΜΕΝΟΥ ΣΗΜΑΤΟΣ (RSSI)

Στα ασύρματα δίκτυα αισθητήρων κάθε κόμβος έχει ένα ραδιοπομπό. Το ερώτημα λοιπόν που προκύπτει είναι πως μπορούμε να εκμεταλλευτούμε αυτό τον εξοπλισμό για να μπορούμε να προσδιορίζουμε την απόσταση εκπομπής. Υπάρχουν λοιπόν δύο πολύ σημαντικές τεχνικές που υπολογίζουν την απόσταση μεταξύ κόμβων χρησιμοποιώντας την αποστολή και λήψη ράδιο-σημάτων. Σε αυτή τη παράγραφο θα αναλύσουμε τη τεχνική «Ένδειξη Ισχύος Λαμβανόμενου Σήματος» ή καλύτερα RSSI (Received Signal Strength Indication).

Σύμφωνα με τη θεωρία η ενέργεια ενός ράδιο-σήματος ελαττώνεται με το τετράγωνο της απόστασης (νόμος του αντίστροφου τετραγώνου). Έτσι λοιπόν κάθε κόμβος που «ακούει» μια ράδιο-μετάδοση, θα είναι σε θέση να χρησιμοποιήσει την ισχύ του λαμβανόμενου σήματος για να προσδιορίσει την απόσταση του από τον πομπό. Έτσι λοιπόν το RSSI προτείνει την εξής λύση. Αφού λοιπόν κάθε κόμβος αισθητήρων έχει ράδιο-πομπό, γιατί να μην χρησιμοποιήσει την ισχύ του σήματος για να προσδιορίσει την απόσταση (range);

Πρακτικά όμως ο υπολογισμός της απόστασης με την χρήση RSI περιέχει θόρυβο αρκετών μέτρων. Αυτός ο θόρυβος οφείλεται στο γεγονός ότι η διάδοση ράδιο-σημάτων είναι μη ομοιόμορφη σε πραγματικό περιβάλλον. Για παράδειγμα η μετάδοση ράδιο-σημάτων είναι διαφορετική πάνω από άσφαλτο και διαφορετική πάνω από γρασίδι. Επίσης στο θόρυβο συνεισφέρουν και κάποια φυσικά εμπόδια όπως τοίχοι, έπιπλα κ.α, που ανακλούν ή απορροφούν τα ράδιο-κύματα. Σαν αποτέλεσμα η πρόβλεψη της απόστασης με βάση την ισχύ του λαμβανόμενου σήματος δεν είναι ιδιαίτερα αξιόπιστη.

Το RSSI χρησιμοποιείται σε πολλούς localization αλγόριθμους και έτσι γίνονται προσπάθειες για την βελτίωση της ποιότητας των αποτελεσμάτων του.

7.3 ΥΠΟΛΟΓΙΣΜΟΣ ΑΠΟΣΤΑΣΗΣ ΜΕ ΤΗΝ ΜΕΤΡΗΣΗ ΤΩΝ HOPS

Μπορεί το RSSI να έχει αρκετές αδυναμίες, αλλά αυτό δεν σημαίνει ότι τα ράδιο-σημάτα που εκπέμπουν οι κόμβοι δεν μπορούν να χρησιμοποιηθούν για τον υπολογισμό της απόστασης. Η βασική παρατήρηση που πρέπει να γίνει είναι ότι αν δύο κόμβοι είναι σε θέση να επικοινωνούν μεταξύ τους με ράδιο-σημάτα, τότε η απόσταση τους είναι με μεγάλη πιθανότητα μικρότερη από R , όπου R είναι η μέγιστη απόσταση ή καλύτερα η εμβέλεια μεταξύ δύο sensor κόμβων. Αυτή η παρατήρηση αποτελεί νέο δεδομένο για τον υπολογισμό της απόστασης μεταξύ δύο κόμβων και το βασικό είναι ότι δεν εμπλέκεται η έννοια της ισχύος του σήματος.

Μοντελοποιώντας το δίκτυο αισθητήρων με ένα γράφο, μπορούμε να θεωρήσουμε ότι οι κορυφές του γράφου είναι οι sensor κόμβοι και επίσης δύο κόμβοι ενώνονται με μια ακμή αν αυτοί μπορούν να επικοινωνήσουν (ο ένας δηλαδή βρίσκεται εντός της εμβέλειας του άλλου). Μπορούμε λοιπόν να ορίσουμε ως h_{ij} τον αριθμό των hops μεταξύ των κόμβων s_i και s_j έτσι ώστε το μονοπάτι που θα οριστεί να είναι το

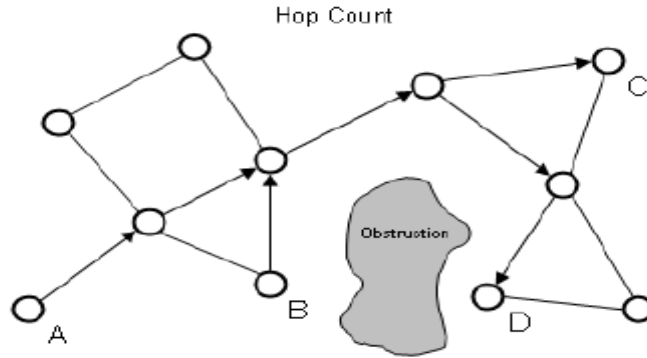
συντομότερο. Αν λοιπόν ονομάσουμε την απόσταση μεταξύ των κόμβων s_i και s_j ως d_{ij} τότε θα ισχύει $d_{ij} \leq R \cdot h_{ij}$.

Μπορεί να υπολογιστεί μία καλύτερη προσέγγιση από την παραπάνω αν είναι γνωστό το n_{local} , δηλαδή ο αναμενόμενος αριθμός των γειτόνων ανά κόμβο. Οι Kleinrock και Silvester [20] υπολόγισαν ένα καλύτερο τύπο για την απόσταση μεταξύ δύο hops (radio hop distance):

$$d_{hop} = R \left(1 + e^{-n_{local}} - \int_{-1}^1 e^{\frac{-n_{local}}{\pi} (\arccos t - t\sqrt{1-t^2})} dt \right)$$

Με βάση λοιπόν το παραπάνω τύπο ισχύει $d_{ij} \approx h_{ij} \cdot d_{hop}$.

Η μέτρηση των hops έχει και αυτή αρκετά μειονεκτήματα. Το βασικότερο είναι ότι κάποια φυσικά εμπόδια μπορεί να αποτρέψουν τις ακμές από το να εμφανιστούν στο γράφο συνεκτικότητας. Σαν αποτέλεσμα οι αποστάσεις με βάση τον αριθμό των hops μπορεί να είναι πολύ μεγάλες, ενώ στη πραγματικότητα οι δύο κόμβοι να απέχουν ελάχιστα. Αυτό γίνεται κατανοητό με τη παρατήρηση του σχήματος 7.1.



Σχήμα 7.1. : Στιγμιότυπο προβλήματος όπου η hop count μέθοδος είναι ανεπιτυχής.

7.4 ΔΙΑΦΟΡΑ ΧΡΟΝΟΥ ΚΑΤΑ ΤΗΝ ΑΦΙΞΗ (TDoA)

Η μέθοδος «Διαφορά Χρόνου Άφιξης» ή TDoA (Time Difference of Arrival) είναι ευρέως γνωστή για το προσδιορισμό της απόστασης (range). Στα συστήματα TDoA, κάθε κόμβος είναι εφοδιασμένος με ένα ηχείο (speaker) και ένα μικρόφωνο. Κάποια συστήματα χρησιμοποιούν υπερηχητικές, ενώ κάποια άλλα ακουστικές (audible) συχνότητες. Η επιλογή όμως αυτή δεν διαφοροποιεί τη βασική ιδέα αυτών των τεχνικών.

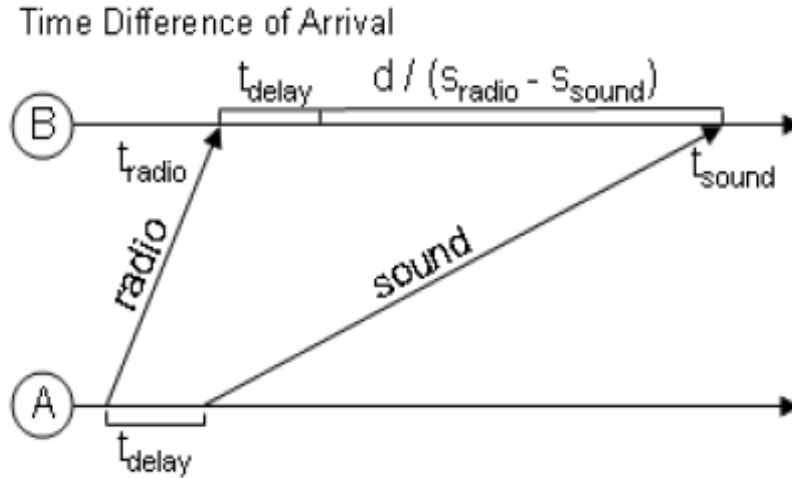
Στη TDoA μέθοδο ο πομπός στέλνει αρχικά ένα ράδιο-μήνυμα. Στη συνέχεια περιμένει για κάποιο χρονικό διάστημα t_{delay} (το οποίο μπορεί να είναι και μηδενικό) και

μετά παράγει chirp patterns (σήματα) με το ηχείο του. Όταν τώρα οι κόμβοι δέκτες ακούσουν ένα ράδιο-σήμα, τότε σημειώνουν τη τρέχουσα ώρα ως t_{radio} και στη συνέχεια ανοίγουν τα μικρόφωνα τους. Όταν τα μικρόφωνα τους ανιχνεύσουν το chirp pattern, σημειώνουν επίσης την τρέχουσα ώρα ως t_{sound} . Από τη στιγμή λοιπόν που οι κόμβοι γνωρίζουν τα t_{delay} , t_{sound} , t_{radio} μπορούν να υπολογίσουν την απόσταση τους d από το πομπό βασιζόμενοι στο γεγονός ότι τα ράδιο-κύματα ταξιδεύουν γρηγορότερα από τον ήχο στον αέρα. Έτσι η απόσταση δίνεται από παρακάτω το τύπο (s_{sound} -ταχύτητα ήχου):

$$d = (s_{radio} - s_{sound}) \cdot (t_{radio} - t_{delay} - t_{sound})$$

Για να γίνουν αντιληπτά τα παραπάνω παραθέτουμε το σχήμα 7.2. Εκεί ο κόμβος A στέλνει ένα ράδιο-παλμό, ο οποίος ακολουθείται από ένα παλμό ήχου. Υπολογίζοντας τη χρονική διαφορά ανάμεσα στις αφίξεις των δύο παλμών, ο κόμβος B μπορεί να εκτιμήσει την απόσταση του από τον A.

Η TDoA μέθοδος είναι εξαιρετικά ακριβής με Line of Sight (LOS) συνθήκες. Επίσης δουλεύει πολύ αποτελεσματικά σε περιοχές που δεν υπάρχει αντήχηση, με απαραίτητη βέβαια προϋπόθεση να είναι σωστά ρυθμισμένα τα ηχεία και τα μικρόφωνα. Αρκετές ομάδες δουλεύουν για να βελτιώσουν τη λειτουργία της TDoA ώστε να είναι αποδοτική σε περιοχές που δεν διαθέτουν τις παραπάνω ιδανικές συνθήκες.



Σχήμα 7.2. : Υπολογισμός απόστασης με TDoA

7.5 ΓΩΝΙΑ ΑΦΙΞΗΣ (AoA)

Κάποιο αλγόριθμοι βασίζονται στην γωνία άφιξης (AoA – Angle of Arrival) των δεδομένων. Αυτά τα δεδομένα μπορούν να συλλεχθούν με τη βοήθεια ράδιο ή μικροφωνικών διατάξεων, οι οποίες επιτρέπουν στο κόμβο δέκτη τα καθορίσει την κατεύθυνση εκπομπής του πομπού.

Σε αυτές τις μεθόδους αρκετά (συνήθως 3 ή 4) μικρόφωνα ανιχνεύουν ένα απλό μεταδιδόμενο σήμα. Αναλύοντας τη διαφορά φάσης ή χρόνου ανάμεσα στις διαφορετικές αφίξεις σημάτων στα διαφορετικά μικρόφωνα, είναι δυνατόν να υπολογίσουμε τη γωνία του σήματος του αφιχθέντος σήματος.

Οι AoA μέθοδοι είναι αρκετά ακριβείς και έχουν σφάλμα μόνο μερικών μοιρών. Όμως δυστυχώς το υλικό που απαιτεί το AoA είναι αρκετά ακριβότερο από αυτό των μεθόδων TDoA, αφού κάθε κόμβος πρέπει να έχει ένα ηχείο και αρκετά μικρόφωνα.

8 ΘΕΜΑΤΑ ΣΤΟ ΣΧΕΔΙΑΣΜΟ LOCALIZATION ΑΛΓΟΡΙΘΜΩΝ

8.1 ΠΕΡΙΟΡΙΣΜΟΙ ΠΟΡΩΝ

Τα δίκτυα αισθητήρων (sensor networks) λειτουργούν έχοντας στη διάθεση τους περιορισμένους πόρους. Οι κόμβοι διαθέτουν σχετικά αδύναμους επεξεργαστές, γεγονός που δεν επιτρέπει δύσκολους υπολογισμούς. Επίσης οι κόμβοι αισθητήρων χρησιμοποιούν σαν πηγή ενέργειας κάποια μπαταρία. Για αυτό το λόγο πρέπει να κάνουμε οικονομία στην ενέργεια που σπαταλάμε α) για επικοινωνία μεταξύ κόμβων και β) στην επεξεργασία δεδομένων.

Επίσης τα δίκτυα αισθητήρων προορίζονται για να χρησιμοποιηθούν σε μεγάλη κλίμακα, και να αποτελούνται από εκατοντάδες κόμβους. Αυτό το γεγονός έχει δύο σοβαρές συνέπειες: α) Το κόστος κάθε κόμβου πρέπει να είναι μικρό (στα μελλοντικά δίκτυα αισθητήρων έξυπνης σκόνης (smart dust) κάθε κόμβος υπολογίζεται να στοιχίζει το πολύ 1\$). β) Το «στήσιμο» (deployment) του δικτύου πρέπει να επίσης εύκολο.

Όπως αναφέραμε παραπάνω η διαδικασία εντοπισμού (localization) έχει πολλές εφαρμογές σε ένα sensor network. Όμως ο εντοπισμός δεν είναι ο κύριος στόχος του δικτύου αισθητήρων. Ο εντοπισμός μπορεί να χρησιμοποιείται για να τρέξει π.χ. ένας αλγόριθμος για data propagation. Για αυτό το λόγο οι αλγόριθμοι εντοπισμού πρέπει να είναι φτηνοί. Άρα λοιπόν καλός αλγόριθμος εντοπισμού δεν είναι μόνο αυτός που επιτυγχάνει να βρει με μικρό σφάλμα τις φυσικές συντεταγμένες του κόμβου, αλλά αυτός που το κάνει με μικρό κόστος ενέργειας και υλικού.

8.2 ΠΥΚΝΟΤΗΤΑ (DENSITY) ΚΟΜΒΩΝ

Πάρα πολλοί localization αλγόριθμοι είναι ευαίσθητοι στο πόσο πυκνό είναι το δίκτυο από κόμβους. Πιο συγκεκριμένα οι localization αλγόριθμοι είναι πιο αποτελεσματικοί όταν εφαρμόζονται σε ένα πυκνό δίκτυο. Για να γίνουμε πιο σαφής μπορούμε να εκφράσουμε τη πυκνότητα του δικτύου $\mu(R)$ συναρτήσει του αριθμού των κόμβων ανά περιοχή κάλυψης(εμβέλειας). Έτσι αν N κόμβοι έχουν διασκορπιστεί σε μία περιοχή εμβαδού A , και αν επίσης R είναι η εμβέλεια κάθε κόμβου τότε ισχύει:

$$\mu(R) = \frac{N \cdot \pi \cdot R^2}{A}$$

Όταν σχεδιάζουμε ή αναλύουμε ένα αλγόριθμο, είναι πολύ σημαντικό να δώσουμε ιδιαίτερη προσοχή στις υποθέσεις που κάνει για τη πυκνότητα του δικτύου, αφού η επίτευξη πολύ μεγάλης πυκνότητας στο δίκτυο είναι πολύ ακριβή και καμιά φορά ανέφικτη.

8.3 ΜΗ ΚΥΡΤΕΣ (NON CONVEX) ΤΟΠΟΛΟΓΙΕΣ

Οι localization αλγόριθμοι συναντούν συχνά προβλήματα να καθορίσουν τις συντεταγμένες κόμβων που βρίσκονται στις ακριανές πλευρές του δικτύου. Αυτό συμβαίνει επειδή δύσκολα μπορούν να μετρηθούν μετρικές όπως η απόσταση σε αυτές τις περιοχές του δικτύου, καθώς επίσης ο υπολογισμός τέτοιων μετρικών θα γίνεται από τη μία μόνο πλευρά του κόμβου. Συμπερασματικά, για τους κόμβους που είναι στα άκρα του δικτύου οι διαθέσιμες πληροφορίες είναι πάντα λιγότερες.

8.4 ΦΥΣΙΚΑ ΕΜΠΟΔΙΑ ΚΑΙ ΑΝΩΜΑΛΙΕΣ ΤΟΥ ΕΔΑΦΟΥΣ

Τα φυσικά εμπόδια και οι ανωμαλίες του εδάφους μπορούν επίσης να δημιουργήσουν προβλήματα στους Localization αλγόριθμους. Για παράδειγμα μεγάλοι βράχοι μπορούν εμποδίσουν την line of sight επικοινωνία εμποδίζοντας τον υπολογισμό απόστασης με τη τεχνική TDoA. Επίσης μπορούν να παρεμβληθούν στα ράδιο-σήματα εισάγοντας θόρυβο και σφάλμα στις αποστάσεις RSSI. Το ίδιο μπορεί να επηρεάσει και η μορφολογία του εδάφους (γρασίδι, άμμος...).

8.5 ΤΑΞΙΝΟΜΗΣΗ ΑΛΓΟΡΙΘΜΩΝ LOCALIZATION

Σε αυτή τη παράγραφο θα κάνουμε μια ταξινόμηση των αλγορίθμων localization με βάση την υπολογιστική τους δομή. Στα επόμενα κεφάλαια του μέρους αυτού, θα αναλύσουμε κάθε κατηγορία ξεχωριστά.

Αρχικά οι *κεντριοποιημένοι αλγόριθμοι (centralized)* έχουν σχεδιαστεί ώστε να τρέχουν σε μία κεντρική μηχανή όπου υπάρχει διαθέσιμη μεγάλη υπολογιστική ισχύς. Η γενική ιδέα είναι ότι οι κόμβοι αισθητήρων συγκεντρώνουν αρχικά τα δεδομένα που συλλέγουν από το περιβάλλον τους (π.χ. αποστάσεις), τα στέλνουν σε ένα σταθμό βάσης για ανάλυση, και μετά την ανάλυση οι συντεταγμένες των κόμβων έχουν υπολογιστεί και επιστρέφουν σε αυτούς μέσω του δικτύου.

Πλεονεκτήματα:

- Αντιμετωπίζουν το πρόβλημα της μικρής επεξεργαστικής ισχύς που υπάρχει στους κόμβους αισθητήρων, μιας και όλοι οι υπολογισμοί γίνονται κεντρικά.
- Μπορούμε να επιτύχουμε μεγάλη κλιμακωσιμότητα (scalability) στο δίκτυο χρησιμοποιώντας πολλούς σταθμούς βάσης.

Μειονεκτήματα:

- Απαίτηση «έξυπνων» σταθμών βάσης που είναι σε θέση να εξυπηρετούν τους κόμβους του δικτύου (καμιά φορά αυτό είναι ανέφικτο).
- Πιθανή συμφόρηση στους κόμβους κοντά στο σταθμό βάσης (θεωρώντας ότι δεν έχουμε mobile base station).

Η επόμενη κατηγορία αναφέρεται στους *κατανεμημένους localization αλγόριθμους*. Σε αυτούς τους αλγόριθμους, εξαιτίας της έλλειψης μια κεντρική υπολογιστικής δύναμης, θα πρέπει να υπάρχει παραλληλισμός και επικοινωνία μεταξύ των κόμβων.

Υπάρχουν δύο διαφορετικές προσεγγίσεις στους κατανεμημένους localization αλγόριθμους. Η πρώτη αναφέρεται στους κατανεμημένους localization αλγόριθμους που βασίζονται σε *beacon* κόμβους. Σε κάποιους αλγόριθμους οι κόμβοι εκτιμούν την απόσταση τους από κάποιους beacon κόμβους και έτσι αποφασίζουν την τοποθεσία τους. Σε κάποιους άλλους, οι κόμβοι του δικτύου μετατρέπονται οι ίδιοι σε beacon για να για να βοηθήσουν άλλους κόμβους που δεν γνωρίζουν ακόμα τη τοποθεσία τους.

Η δεύτερη κατηγορία κατανεμημένων αλγορίθμων προσπαθεί να βελτιστοποιήσει μία καθολική μετρική σε όλο το δίκτυο με ένα κατανεμημένο τρόπο. Πάλι εδώ μπορούμε να διακρίνουμε δύο προσεγγίσεις. α) Η πρώτη προσέγγιση (κεφάλαιο 11) χρησιμοποιεί αρχικά ένα όχι optimal αλγόριθμο για να προσεγγίσει περίπου τη θέση κάθε κόμβου στο δίκτυο. Στη συνέχεια ο αλγόριθμος τρέχει επαναληπτικά προσπαθώντας να κάθε φορά να υπολογίσει ακριβέστερα τη θέση κάθε κόμβου στο δίκτυο με βάση τοπική μετρική σφάλματος. β) Η δεύτερη προσέγγιση (κεφάλαιο 12) χωρίζει το δίκτυο σε μικρές επικαλυπτόμενες καμιά φορά περιοχές όπου για την καθεμία δημιουργείται ένας τοπικός optimal χάρτης. Τελικά οι υπό-περιοχές χρησιμοποιούν μία ομότιμη (peer to peer) διαδικασία για να ενώσει τους υπό-χάρτες σε ένα καθολικό χάρτη. Στα επόμενα κεφάλαια αυτού του μέρους περιγράφονται οι παραπάνω κατηγορίες.

9 ΚΕΝΤΡΙΚΟΠΟΙΗΜΕΝΟΙ ΑΛΓΟΡΙΘΜΟΙ

Αυτό το κεφάλαιο αναφέρεται στους *κεντριοποιημένους localization αλγόριθμους*. Η κεντριοποιημένη λύση δίνει τη δυνατότητα για πολύ πιο δύσκολους μαθηματικούς υπολογισμούς μιας και όπως αναφέραμε ο κεντρικός σταθμός βάσης έχει μεγάλη υπολογιστική ισχύ. Η κεντρική ιδέα των centralized localization αλγόριθμων είναι ότι κάθε κόμβος στέλνει δεδομένα όπως η συνεκτικότητα και η απόσταση του από τους άλλους κόμβους, σε ένα κεντρικό ισχυρό σταθμό βάσης, ο οποίος με τη σειρά του ξαναστέλνει πίσω (μετά από επεξεργασία) τις τοποθεσίες (συντεταγμένες) των κόμβων του δικτύου. Εκεί λοιπόν που διαφέρουν οι centralized localization αλγόριθμοι μεταξύ τους είναι ο τύπος της επεξεργασίας που γίνεται στο σταθμό βάσης. Στην επόμενη παράγραφο θα αναλύσουμε ένα τύπο επεξεργασίας στο σταθμό βάσης, που ονομάζεται πολυδιάστατη κλιμάκωση (multidimensional scaling). Ένας άλλος επίσης πολύ δημοφιλής είναι ο *semidefinite programming* [21].

9.1 MDS-MAP

Ο MDS-MAP είναι ένα κεντριοποιημένος αλγόριθμος ο οποίος βασίζεται σε μία μαθηματική τεχνική που είναι γνωστή ως πολυδιάστατη κλιμάκωση ή MDS (multidimensional scaling).

Η ιδέα πίσω από την πολυδιάστατη κλιμάκωση είναι απλή. Ας υποθέσουμε ότι υπάρχουν n σημεία σε ένα χώρο. Παρόλο που δεν γνωρίζουμε τις θέσεις των σημείων αυτών, γνωρίζουμε τις αποστάσεις μεταξύ κάθε ζεύγους σημείων. Ο MDS είναι μια $O(n^3)$ τεχνική που χρησιμοποιεί τον νόμο των συνημίτονων και τη γραμμική άλγεβρα, για να ανακατασκευάσει τις σχετικές θέσεις των σημείων χρησιμοποιώντας τις αποστάσεις ανάμεσα στα ζευγάρια. (Οι μαθηματικές λεπτομέρειες είναι έξω από το στόχο αυτής της εργασίας).

Ο αλγόριθμος MDS-MAP είναι σχεδόν μία άμεση εφαρμογή της τεχνικής MDS και αποτελείται από τέσσερα βήματα:

1. Συγκέντρωσε τα δεδομένα που περιέχουν τις αποστάσεις από το δίκτυο και κατασκεύασε ένα αραιό μητρώο R όπου r_{ij} είναι η απόσταση (range) μεταξύ των κόμβων i, j . Αν οι δύο κόμβοι είναι τόσο μακριά ώστε δεν μπορούν να επικοινωνήσουν άμεσα, η απόσταση αυτή είναι ίση με το μηδέν.
2. Τρέξε ένα αλγόριθμο συντομότερο μονοπατιού για όλα τα ζευγάρια (π.χ. Dijkstra's, Floyd's) πάνω στο μητρώο R και παράγαγε ένα πλήρες μητρώο αποστάσεων D .
3. Τρέξε τη τεχνική MDS πάνω στον D για να βρεις τις εκτιμώμενες θέσεις των κόμβων του δικτύου X .
4. Μετέτρεψε τη λύση X σε καθολικές συντεταγμένες χρησιμοποιώντας έναν αριθμό από Beacon κόμβους.

Ο αλγόριθμος MDS-MAP δουλεύει καλά με RSSI, και όπως είναι αναμενόμενο η λειτουργία του βελτιώνεται με τη βελτίωση των τεχνικών εύρεσης απόστασης που αναλύσαμε στο παραπάνω κεφάλαιο. Όμως ο αλγόριθμος MDS-MAP χρησιμοποιεί αναποδοτικά τους beacon κόμβους, μιας και εκμεταλλεύεται τη λειτουργία τους μόνο στο βήμα 4. Το κύριο παρόλα αυτό πρόβλημα του MDS-MAP είναι η κακή ασυμπτωτική του συμπεριφορά ($O(n^3)$) εξαιτίας των βημάτων 2 και 3.

10 BEACON-BASED ΚΑΤΑΝΕΜΗΜΕΝΟΙ ΑΛΓΟΡΙΘΜΟΙ

Σε αυτό το κεφάλαιο θα αναφερθούμε σε beacon-based κατανεμημένους αλγόριθμους. Αυτοί οι αλγόριθμοι υπολογίζουν τις θέσεις των κόμβων του δικτύου (*unknown nodes*) από τις θέσεις των κόμβων beacon. Έτσι λοιπόν αυτοί οι αλγόριθμοι προσδιορίζουν απευθείας τις καθολικές (global) συντεταγμένες των unknown κόμβων με τη βοήθεια των beacon. Αυτοί οι αλγόριθμοι είναι επίσης κατανεμημένοι με την έννοια ότι όλοι οι υπολογισμοί γίνονται στους ίδιους τους κόμβους αισθητήρων. Σε αυτό το κεφάλαιο θα παρουσιάσουμε αρχικά τέσσερις αλγόριθμους (diffusion, bounding box, gradient multilateration και APIT) και τέλος θα εστιάσουμε σε ένα αλγόριθμο όπου έχουμε ένα κινητό beacon κόμβο δίνοντας παραθέτοντας και πειραματικά αποτελέσματα.

10.1 DIFFUSION

Ο diffusion (αλγόριθμος διάχυσης) προκύπτει από μία πολύ απλή ιδέα. Η πιθανότερη θέση ενός κόμβου είναι στο κέντρο των θέσεων των γειτονικών του κόμβων. Οι αλγόριθμοι διάχυσης απαιτούν μόνο δεδομένα συνεκτικότητας (δηλαδή αν δύο κόμβοι μπορούν να επικοινωνήσουν με ράδιο-σήματα). Δύο παραλλαγές αυτών των αλγορίθμων είναι οι εξής:

Ο Bulusu στο paper [22] προσδιόρισε τις θέσεις των unknown κόμβων βρίσκοντας το μέσο όρο των θέσεων όλων των beacon κόμβων που είναι εντός εμβέλειας των unknown κόμβων. Έτσι λοιπόν κάθε unknown κόμβος εκτιμάει τη θέση του (X_{est}, Y_{est}) ως το μέσο όρο των θέσεων όλων των beacon κόμβων εντός εμβέλειας και έτσι ισχύει:

$$(X_{est}, Y_{est}) = \left(\frac{\sum_{i=1}^k X_i}{k}, \frac{\sum_{i=1}^k Y_i}{k} \right)$$

Αν και αυτή η μέθοδος είναι ελκυστική λόγω της απλότητας της, δεν δίνει πολύ καλά αποτελέσματα ιδιαίτερα όταν η πυκνότητα των beacon κόμβων είναι περιορισμένη ή οι unknown κόμβοι είναι στα άκρα του δικτύου.

Μία άλλη παραλλαγή της παραπάνω λογικής είναι αυτή των Fitzpatrick και Meetens [23]. Σε αυτή τη περίπτωση ο υπολογισμός του κεντροειδούς γίνεται λαμβάνοντας υπόψη και τους μη-beacon κόμβους. Εν συντομία ο αλγόριθμος λειτουργεί ως εξής:

1. Αρχικοποίησε τις θέσεις των non-beacon κόμβων στο (0,0).
2. Επανέλαβε τα παρακάτω μέχρι η θέση να συγκλίνει σε κάποια τιμή.
3. Θέσε τη θέση κάθε non-beacon κόμβου ως τη μέση τιμή των θέσεων όλων των γειτονικών κόμβων.

Το πλεονέκτημα αυτής της προσέγγισης είναι ότι απαιτεί λιγότερους beacon κόμβους. Παρόλα η ακρίβεια του αλγορίθμου είναι μικρή ιδιαίτερα όταν η πυκνότητα του δικτύου είναι μικρή, οι κόμβοι είναι εκτός της εμβέλειας των beacon κόμβων, ή όταν η πυκνότητα κόμβων ποικίλλει κατά μήκος του δικτύου. Επίσης αυτή η παραλλαγή έχει περισσότερο υπολογιστικό φόρτο, μιας και λαμβάνει υπόψη κάθε γείτονα του unknown κόμβου.

10.2 BOUNDING BOX

Ο Bounding Box αλγόριθμος είναι μια υπολογιστικά απλή μέθοδος που βρίσκει τις συντεταγμένες των κόμβων του δικτύου με βάση τις αποστάσεις αρκετών beacon κόμβων. Συγκεκριμένα κάθε κόμβος υποθέτει ότι βρίσκεται στη διατομή των κουτιών οριοθέτησης (bounding boxes) των δικών του beacon κόμβων (δηλαδή των beacon που είναι στην εμβέλεια του). Το κουτί οριοθέτησης (bounding box) ενός beacon b έχει κέντρο τη θέση του beacon (x_b, y_b) . Επίσης το bounding box έχει ύψος και πλάτος $2d_b$ όπου d_b είναι η απόσταση του κόμβου από τον beacon κόμβο.

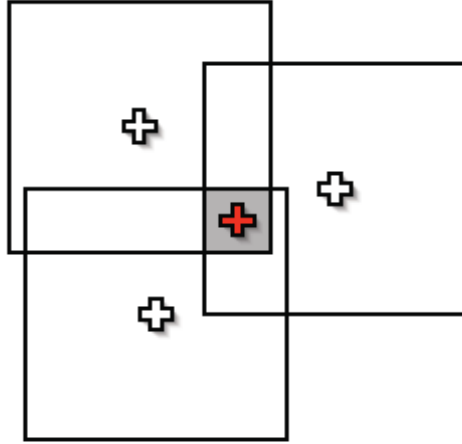
Η διατομή των bounding boxes μπορεί να υπολογιστεί χωρίς πράξεις κινητής υποδιαστολής.:

$$\left[\max(x_i - d_i), \max(y_i - d_i) \right] \times \left[\min(x_i + d_i), \min(y_i + d_i) \right]$$

για $i = 1 \dots n$

Η τελική θέση κάθε κόμβου είναι το κέντρο του τελικού bounding box όπως φαίνεται στο σχήμα 10.1.

Η ακρίβεια του Bounding Box αλγορίθμου είναι μεγαλύτερη, όταν οι πραγματικές θέσεις των unknown κόμβων είναι κοντύτερα στα κέντρα των beacon τους.



Σχήμα 10.1. : Παράδειγμα διατομής των bounding boxes. Το κέντρο της διατομής είναι η εκτιμώμενη θέση του unknown κόμβου. Το μέγεθος των κουτιών εξαρτάται από την απόσταση (hop count radio range) του beacon από τον unknown κόμβο.

10.3 GRADIENT

Η βασική μαθηματική τεχνική της μεθόδου gradient ονομάζεται multilateration. Η multilateration τεχνική μοιάζει πολύ με αυτή του τριγωνισμού (triangulation) απλά μπορεί να ενσωματώσει αποστάσεις (ranges) από περισσότερα από ένα σημεία. Τυπικά, δοθέντος m beacon κόμβους με γνωστές καρτεσιανές θέσεις $b_i, i=1...m$ και αποστάσεις (ranges) r_i (πιθανώς με προσθήκη θορύβου) από τους γνωστούς κόμβους (δηλαδή από κόμβους που γνωρίζουμε τις θέσεις τους) σε έναν unknown κόμβο s , τότε η multilateration τεχνική βρίσκει τη πιο πιθανή θέση του s . Να σημειώσουμε τέλος ότι η multilateration τεχνική χρησιμοποιείται για το localization και στην τεχνολογία GPS.

Η χρήση της μεθόδου gradient (βαθμωτή μεταβολή) για τον προσδιορισμό των αποστάσεων (ranges) που θα χρησιμοποιηθούν στη συνέχεια από την multilateration τεχνική έχει μελετηθεί από ένα μεγάλο αριθμό ερευνητών. Όλοι οι αλγόριθμοι που χρησιμοποιούν gradients για τον υπολογισμό των αποστάσεων υποθέτουν ότι υπάρχουν το λιγότερο τρεις beacon κόμβοι κάπου στο δίκτυο. Καθένας από αυτούς τους beacon κόμβους διαδίδει ένα gradient κατά μήκος του δικτύου που είναι το κατανεμημένο ισοδύναμο του υπολογισμού του συντομότερου μονοπατιού ανάμεσα σε όλα τα beacons και στους unknown κόμβους. Η μέθοδος της gradient διάδοσης λειτουργεί ως εξής:

1. Για κάθε κόμβο j και beacon k , ας υποθέσουμε ότι d_{jk} (απόσταση από τον j στον k) είναι 0 αν $j=k$ και ∞ διαφορετικά.
2. Για κάθε κόμβο j εκτέλεσε τα ακόλουθα βήματα επαναληπτικά.
3. Για κάθε beacon k και γειτονικό κόμβο i , ανάκτησε την απόσταση d_{ik} από το κόμβο i .

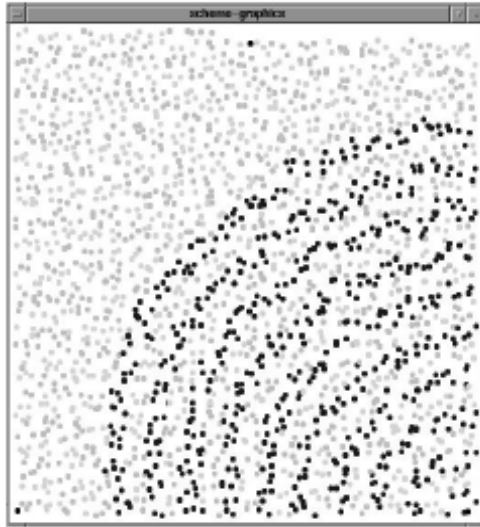
4. Για κάθε beacon k και γειτονικό κόμβο i , εφάρμοσε το παρακάτω τύπο ανανέωσης της απόστασης:

$$d_{jk} = \min(d_{ik} + \hat{r}_{ij}, d_{ik})$$

όπου $\hat{r}_{ij}$ είναι η εκτιμώμενη απόσταση μεταξύ των κόμβων i, j . Αυτές οι εκτιμώμενες αποστάσεις ανάμεσα στους κόμβους μπορεί να μην έχουν βάρος (1 εάν οι κόμβοι είναι συνδεδεμένοι, ή 0 διαφορετικά) ή να είναι αποστάσεις με βάρος (π.χ. RSSI αποστάσεις).

Μετά από κάποιο χρόνο τρεξίματος του αλγόριθμου κάθε τιμή d_{jk} θα έχει το μήκος του συντομότερου μονοπατιού ανάμεσα στο κόμβο j και στο beacon k . Μπορούμε να σημειώσουμε εδώ ότι αυτή η μέθοδος ακολουθεί μία Bellman-Ford λογική. Το σχήμα 10.2 δείχνει τα αποτελέσματα τρεξίματος του αλγόριθμου διάδοσης gradient με ένα μόνο beacon κόμβο.

Όπως οι προηγούμενοι beacon-based καταναμημένοι αλγόριθμοι και αυτός ο αλγόριθμος έχει το πλεονέκτημα της απλότητας. Επίσης είναι εύκολο να κλιμακωθεί (scalable) αν οι πυκνότητα των beacon κόμβων διατηρηθεί σταθερή. Τέλος ο αλγόριθμος αυτός είναι αρκετά αποδοτικός σε ομογενής τοπολογίες εκεί που δεν υπάρχουν φυσικά εμπόδια.



Σχήμα 10.2. : Αλγόριθμος διάδοσης gradient με ένα μόνο beacon κόμβο (κάτω δεξιά γωνία). Κάθε τελεία είναι ένας sensor κόμβος. Οι κόμβοι αισθητήρων χρωματίζονται ανάλογα με την τιμή gradient

10.4 APIT

Ο APIT [24] είναι λίγο διαφορετικός από τους beacon-based καταναμημένους αλγόριθμους που περιγράψαμε μέχρι τώρα. Ο APIT χρησιμοποιεί μια προσέγγιση στην

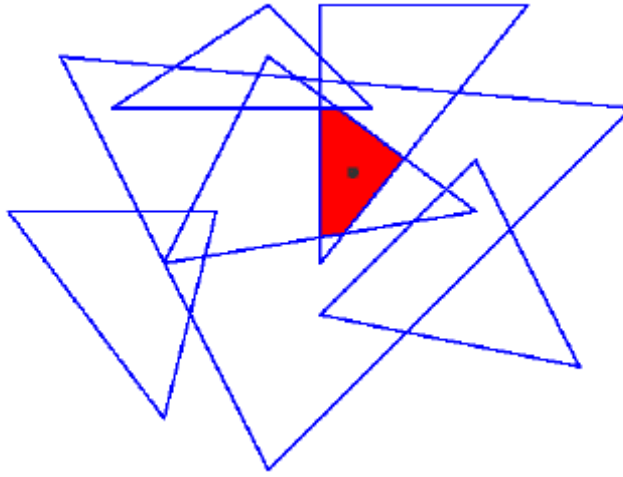
οποία υποθέτουμε ότι οι κόμβοι είναι σε θέση να ακούσουν έναν μεγάλο αριθμό από beacon κόμβους. Όμως αντίθετα οι κόμβοι δεν είναι σε θέση να εκπέμψουν (range) στους beacons. Επίσης οι unknown κόμβοι σχηματίζουν έναν αριθμό από “beacon τρίγωνα” όπου οι κορυφές των τριγώνων αποτελούνται από τρεις κόμβους beacon. Ο unknown κόμβος στη συνέχεια αποφασίζει αν βρίσκεται μέσα ή έξω από ένα δοθέν τρίγωνο συγκρίνοντας τα μεγέθη ισχύος του σήματος, με τους γειτονικούς του non-beacon κόμβους. Όταν αυτή η διαδικασία ολοκληρωθεί, ο unknown κόμβος βρίσκει τη διατομή των “beacon τριγώνων” που τον περικλείουν. Τελικά unknown κόμβος υπολογίζει το κεντροειδές αυτής της διατομής το οποίο είναι και η εκτίμηση της θέσης του. Η διαδικασία που περιγράψαμε φαίνεται στο σχήμα 10.3.

Αναλυτικά τα βήματα του αλγόριθμου που αναφέραμε παραπάνω είναι:

- **Βήμα 1:** Λάβε τις θέσεις των beacon κόμβων που μπορείς να ακούσεις (εντός εμβέλειας).
- **Βήμα 2:** Αρχικοποίησε το *inside-set* ώστε να είναι άδειο.
- **Βήμα 3:** Για κάθε τρίγωνο T_i από τα τρίγωνα που μπορούν να σχηματίζουν οι beacon κόμβοι, πρόσθεσε το T_i στο *inside-set* μόνο εάν ο unknown κόμβος βρίσκεται μέσα σε αυτό το τρίγωνο. Πήγαινε στο βήμα 4 όταν η ακρίβεια του *inside-set* είναι επαρκής.
- **Βήμα 4:** Υπολόγισε την εκτίμηση της θέσης του unknown κόμβου στο κέντρο της περιοχής της διατομής των τριγώνων που είναι στο *inside-set*.

Για να βρούμε αν ένας κόμβος είναι σε κάποιο τρίγωνο πρέπει να βασιστούμε στη γεωμετρία (PIT-point in triangle). Για ένα τρίγωνο με κορυφές A, B, C , ένα σημείο M είναι εκτός του τριγώνου ABC , εάν υπάρχει μία κατεύθυνση τέτοια ώστε ένα σημείο γειτονικό του M είναι πιο κοντά ταυτόχρονα στα σημεία A, B και C . Διαφορετικά το M είναι μέσα στο τρίγωνο ABC . Αυτή η γεωμετρική διαδικασία προτείνεται και στον API (Approximate PIT) αλγόριθμο όπου το σημείο M είναι unknown κόμβος, και τα A, B, C είναι beacon.

Αυτό που πρέπει να τονίσουμε στον αλγόριθμο αυτό είναι ότι ο APIT είναι *range-free* αλγόριθμος. Αυτό σημαίνει ότι το RSSI μέτρο απόστασης απαιτείται βαθμονομημένο και μονότονο απλά για να διεξάγονται οι συγκρίσεις μεταξύ των αποστάσεων. Δεν απαιτείται όμως το RSSI να παράγει προσεγγίσεις για το πόσο είναι κάποια απόσταση.



Σχήμα 10.3. : Εκτίμηση της θέσης του unknown κόμβου στο κέντρο της διατομής ενός αριθμού beacon τριγώνων.

10.5 LOCALIZATION ME ENA MOBILE BEACON

Σε αυτή τη παράγραφο θα αναφερθούμε σε μία ιδέα η οποία αναλύεται στο [19] και διαφέρει από τις παραπάνω μιας και χρησιμοποιεί ένα κινητό (mobile) beacon. Η βασική ιδέα λοιπόν σε αυτόν τον αλγόριθμο είναι ότι αφού το δίκτυο στηθεί, ένα κινητό beacon το οποίο διαθέτει GPS βοηθάει τους unknown κόμβους να υπολογίσουν τις θέσεις τους. Το κινητό beacon μπορεί να είναι ένας άνθρωπος, ένα όχημα με αυτόματο πιλότο, ή ακόμα στη περίπτωση που το δίκτυο παρατάσσεται ρίχνοντας τους κόμβους από ένα αεροπλάνο, να είναι το ίδιο το αεροπλάνο.

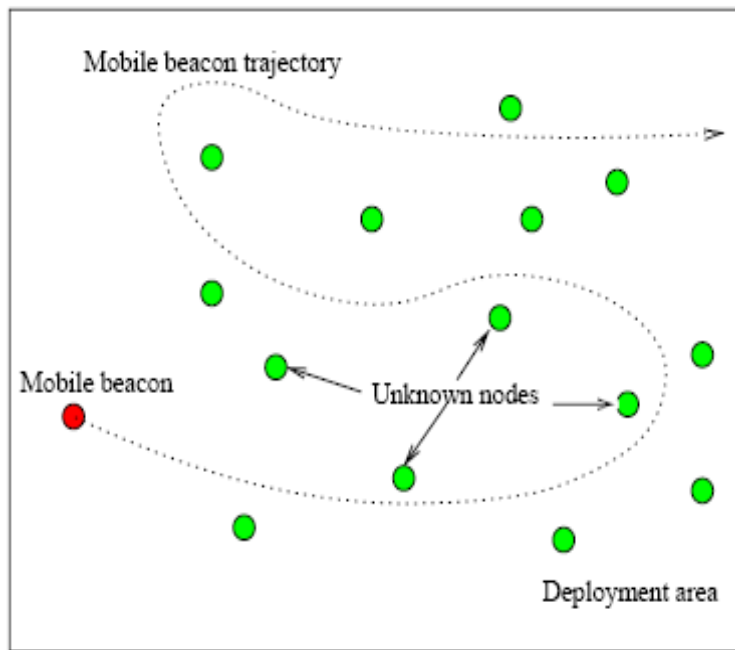
10.5.1 Ιδέα του mobile beacon

Αρχικά πρέπει να σημειώσουμε ότι όταν κάποιος localization αλγόριθμος δεν χρησιμοποιεί κάποιους beacon κόμβους οι οποίοι έχουν τη δυνατότητα GPS, δεν μπορεί να υπολογίσει τις θέσεις των beacon κόμβων με βάση κάποιο fixed σύστημα συντεταγμένων. Για αυτό λοιπόν επιλέχτηκε η χρήση beacon στον αλγόριθμο. Επίσης όπως έχουμε προαναφέρει αν και η χρήση μεγαλύτερου αριθμού από beacon κόμβους μπορεί να δώσει πιο ακριβείς υπολογισμούς θέσης, οι κόμβοι αυτοί είναι αρκετά ακριβή (λόγω του GPS εξοπλισμού τους) και έτσι η χρήση τους πρέπει να είναι περιορισμένη. Σε αυτό συμβάλλει επίσης το ότι οι κόμβοι αυτοί παύουν να είναι χρήσιμοι μετά τη βοήθεια που προσφέρουν για τον υπολογισμό της θέσης, μιας και οι unknown κόμβοι δεν είναι κινητοί. Ο αλγόριθμος που θα παρουσιάσουμε σε αυτή τη παράγραφο αν και χρησιμοποιεί ένα μόνο κόμβο beacon κάνει πολύ αποδοτικό localization.

Ο αλγόριθμος του ενός mobile κόμβου είναι επίσης εξαιρετικά scalable (κλιμακώσιμος). Ο υπολογισμός της θέσης κάθε unknown κόμβου δεν γίνεται με βάση τοπικές πληροφορίες από άλλους unknown κόμβους, αλλά με τη βοήθεια του κινητού beacon. Έτσι η προσθήκη νέων κόμβων στο δίκτυο δε δημιουργεί κάποια επιπλέον δυσκολία.

10.5.2 Αλγόριθμος του mobile beacon

Το παρακάτω σχήμα (σχήμα 10.4) μπορεί να μας δώσει μια εικόνα για το πώς είναι το δίκτυο «απλωμένο» και για το πώς μπορεί να κινηθεί ο mobile beacon κόμβος. Να σημειώσουμε εδώ ότι κατά τη κίνηση του ο beacon κόμβος εκπέμπει beacon πακέτα με τις συντεταγμένες του beacon κάθε χρονική στιγμή. Έτσι κάθε κόμβος που λαμβάνει τα beacon πακέτα θα μπορεί να συμπεράνει με μια συγκεκριμένη βεβαιότητα ότι είναι κάπου κοντά στο beacon κόμβο.



Σχήμα 10.4. : Ένας κινητός beacon κόμβος που κινείται στο πεδίο του sensor network.

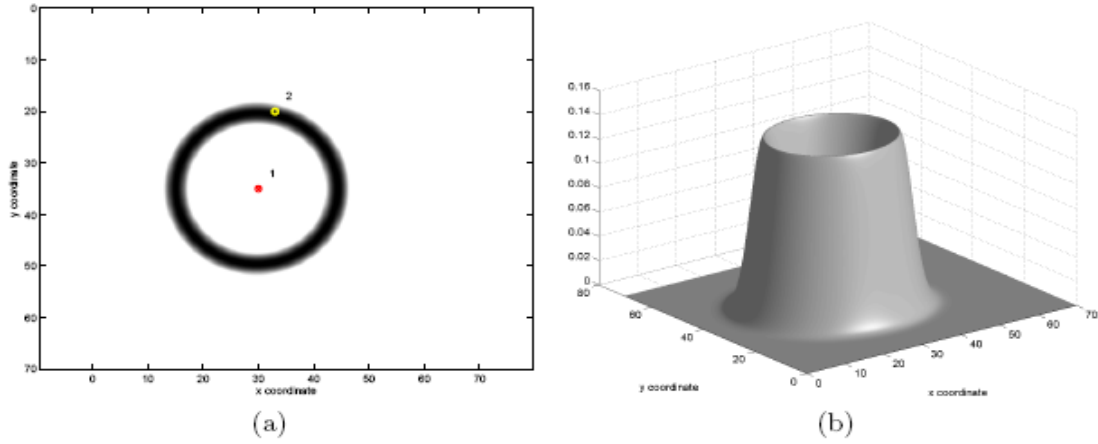
Άρα λοιπόν τα beacon πακέτα που λαμβάνονται περιορίζουν τις πιθανές τοποθεσίες που μπορεί να βρίσκεται ο unknown κόμβος. Στη συνέχεια κάθε unknown κόμβος μετράει το RSSI για κάθε beacon πακέτο που λαμβάνεται. Άρα λοιπόν γνωρίζοντας την RSSI εξασθένηση και την θέση (x_B, y_B) του beacon κόμβου (που

περιέχεται στο beacon πακέτο), κάθε κόμβος που λαμβάνει ένα beacon πακέτο κατασκευάζει μία εμβέλεια περιορισμού (constraint) για τη θέση του ως εξής:

$$\text{Constraint}(x,y) = \text{PDF}_{\text{RSSI}}(d((x,y), (x_B, y_B))) \quad \forall (x,y) \in [(x_{\min}, x_{\max}) \times (y_{\min}, y_{\max})]$$

Η PDF_{RSSI} είναι συνάρτηση πυκνότητας πιθανότητας της απόστασης με την RSSI εξασθένιση και επίσης το $d(A,B)$ συμβολίζει την ευκλείδεια απόσταση ανάμεσα στα σημεία A και B. Τέλος τα $x_{\min}, x_{\max}, y_{\min}, y_{\max}$ είναι οι συντεταγμένες που ορίζουν τα όρια «απλώματος» του δικτύου αισθητήρων.

Για να γίνουν πιο κατανοητά τα παραπάνω στο σχήμα 10.5 φαίνεται ο περιορισμός (constraint) που επιβάλλεται σε ένα unknown κόμβο 2, όταν ένα beacon πακέτο με συντεταγμένες (30,35) και RSSI 75 λαμβάνεται από το mobile beacon 1.



Σχήμα 10.5. : a) constraint 2D b) constraint 3D

Αφού υπολογιστεί ο περιορισμός (constraint) κάθε κόμβος εφαρμόζει το Bayesian πόρισμα για να υπολογίσει τη νέα εκτίμηση θέσης του *NewPostEst* από τη παλιά εκτίμησης θέσης *OldPostEst* και τον νέο constraint *Cons*.

$$\text{NewPostEst}(x,y) = \frac{\text{OldPostEst}(x,y) \times \text{Cons}(x,y)}{\int_{x_{\min}}^{x_{\max}} \int_{y_{\min}}^{y_{\max}} \text{OldPostEst}(x,y) \times \text{Cons}(x,y)}$$

$$\text{για } \forall (x,y) \in [(x_{\min}, x_{\max}) \times (y_{\min}, y_{\max})]$$

Να υπενθυμίσουμε ότι ο όρος στο παρανομαστή είναι όρος κανονικοποίησης. Επίσης να σημειώσουμε ότι η εκτίμηση θέσης αρχικοποιείτε με μία σταθερά.

Στο σχήμα 10.6 που παραθέτουμε παρακάτω, φαίνεται η εξέλιξη της εκτίμησης θέσης κάποιου unknown κόμβου που μεταβάλλεται καθώς ο κόμβος beacon εκπέμπει beacon πακέτα. Στην περίπτωση της 10.6(a) φαίνεται η θέση του unknown κόμβου και η τροχιά του beacon κόμβου. Ο beacon κόμβος στέλνει πακέτα σε κάθε μια από τις θέσεις που είναι μαρκαρισμένες με dots. Μόνο όμως εννέα beacon πακέτα λαμβάνονται από τον unknown κόμβο (αριθμημένα σημεία). Η σειρά που λαμβάνονται φαίνεται με τη σειρά αρίθμησης του σχήματος. Η εξέλιξη λοιπόν της εκτίμησης θέσης του unknown κόμβου καθώς τα beacon πακέτα 1,2,4,5 και 9 λαμβάνονται φαίνεται στα 10.6(b-f).

Από της στιγμή που θα υπολογιστεί η τελική συνάρτηση πυκνότητας πιθανότητας εκτίμησης θέσης *PostEst* (αφού ληφθούν δηλαδή όλα τα beacon πακέτα), οι βέλτιστες συντεταγμένες $(\hat{x}, \hat{y})$ του unknown κόμβου μπορούν να υπολογιστούν ως σταθμισμένος μέσος από τη σχέση:

$$(\hat{x}, \hat{y}) = \left(\int_{x_{\min}}^{x_{\max}} \int_{y_{\min}}^{y_{\max}} x \times PostEst(x, y) dx dy, \int_{x_{\min}}^{x_{\max}} \int_{y_{\min}}^{y_{\max}} y \times PostEst(x, y) dx dy \right)$$

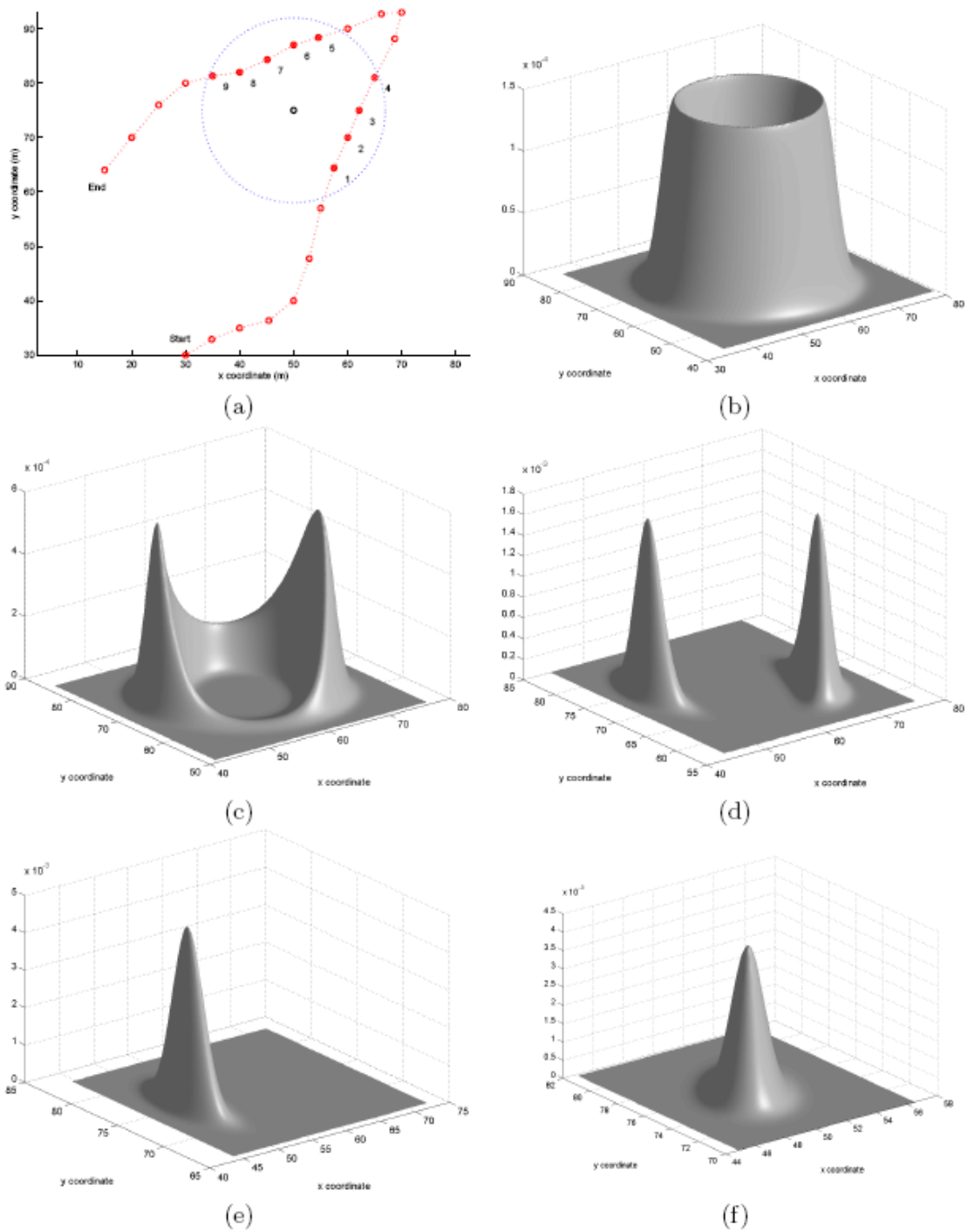
10.5.3 Τροχιά του mobile beacon

Μια ενδιαφέρουσα ερώτηση θα ήταν, «Ποια είναι η βέλτιστη πορεία (τροχιά) που πρέπει να ακολουθήσει ο Mobile Beacon» καθώς επίσης «Πότε θα πρέπει να εκπέμπονται τα Beacon πακέτα»; Τα ερωτήματα αυτά είναι δύσκολο να απαντηθούν και επίσης είναι έξω από το σκοπό της εργασίας. Αξίζει να δώσουμε όμως κάποια χαρακτηριστικά που έχει η πορεία του mobile beacon.

Αρχικά πρέπει να σημειώσουμε ότι η θέση κάθε unknown κόμβου υπολογίζεται ακριβέστερα αν η τροχιά του beacon κόμβου είναι κοντά σε αυτόν. Έτσι λοιπόν πιθανόν μια ιδανική τροχιά του beacon κόμβου θα ήταν να περνάει όσο πιο κοντά γίνεται σε ένα μεγάλο αριθμό unknown κόμβων.

Όμως πρέπει επίσης να σημειώσουμε ότι ακόμα και εάν ο mobile beacon κόμβος περάσει πολύ κοντά κατά μήκος μιας ευθείας γραμμής από έναν unknown κόμβο, το ο αλγόριθμος μας δεν θα είναι σε θέση να καθορίσει από πια πλευρά της ευθείας ο unknown κόμβος βρίσκεται. Πράγματι και οι δύο συμμετρικές θέσεις είναι εξίσου πιθανές. Στο σχήμα 10.6(d) φαίνεται η εκτίμηση θέσης μετά από τέσσερα συγγραμμικά beacon πακέτα που ελήφθησαν από το κόμβο. Για να απαλείψουμε μία από τις υποψήφιες συμμετρικές θέσεις αυτού του unknown κόμβου, πρέπει αυτός ο κόμβος να λάβει τουλάχιστον ένα μη-συγγραμμικό beacon πακέτο. Η εικόνα 10.6(e) δείχνει τη περίπτωση που ένα τέτοιο πακέτο λαμβάνεται από ένα unknown κόμβο.

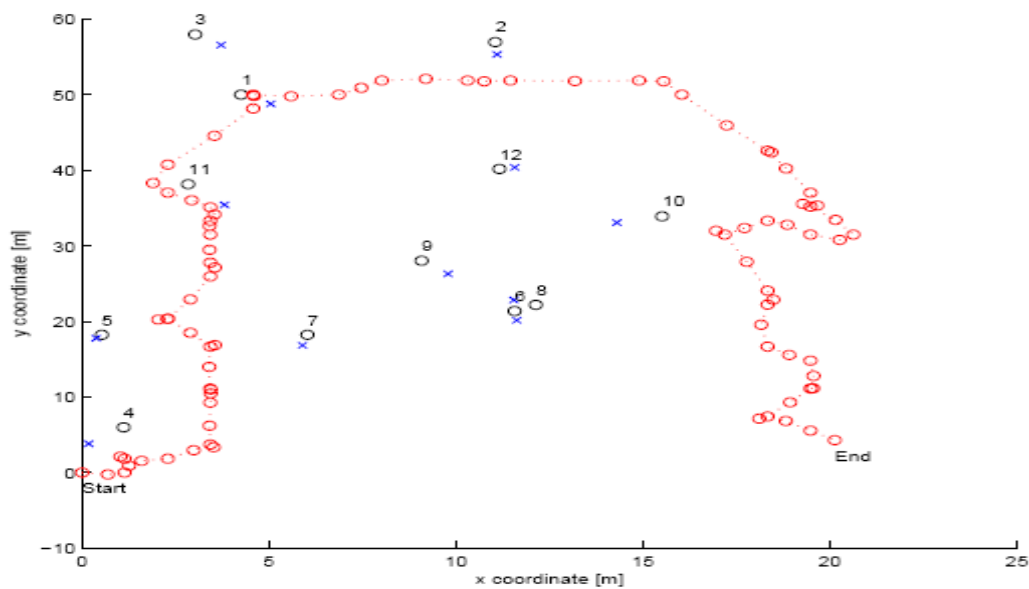
Συμπερασματικά, η τροχιά του beacon κόμβου πρέπει να σχεδιαστεί με τέτοιο τρόπο ώστε όλες οι δυνατές θέσεις να καλύπτονται πλήρως από τουλάχιστον τρία μη-συγγραμμικά beacon πακέτα και το «πλέγμα» που σχηματίζεται από τα beacon πακέτα να είναι όσο πιο «στενό» γίνεται ώστε να αυξηθεί η ακρίβεια των υπολογισμών.



Σχήμα 10.6. : *a)* Η τροχιά του mobile beacon και οι εννέα beacon μεταδώσεις που λαμβάνονται από τον unknown κόμβο. *b-f)* Η εξέλιξη της εκτίμησης θέσης του unknown κόμβου μετά από τη λήψη των 1,2,4,5 και 9 beacon πακέτων.

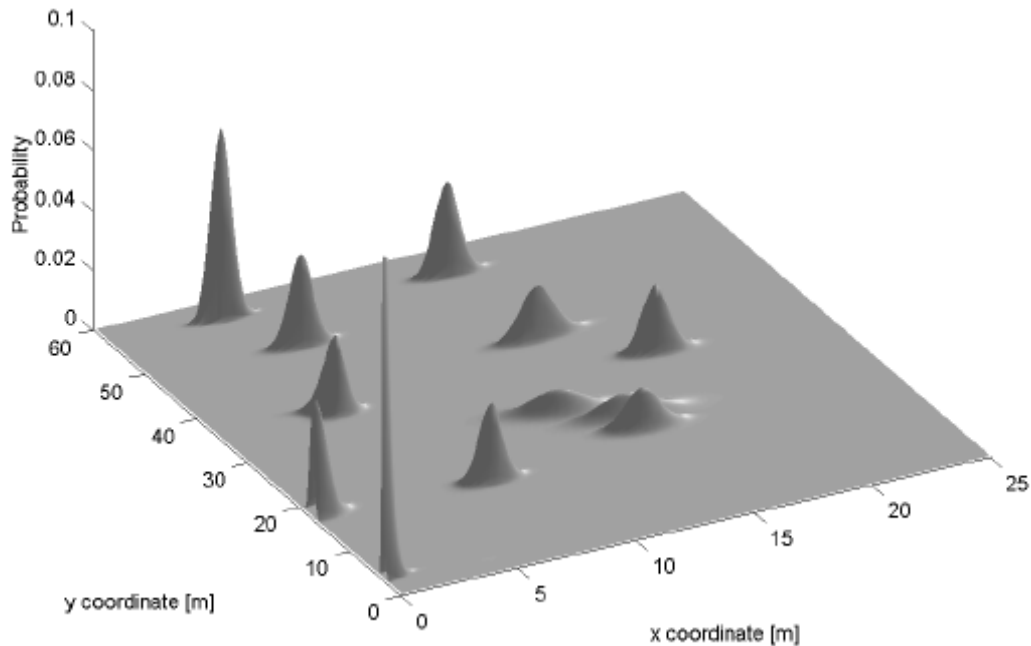
10.5.4 Πειραματικά αποτελέσματα του αλγορίθμου

Σε αυτή τη παράγραφο χωρίς να υπεισέλθουμε σε λεπτομέρειες θα δώσουμε ένα δείγμα του πόσο αποτελεσματικά δουλεύει ο αλγόριθμος. Το παρακάτω σχήμα (σχήμα 10.7) δείχνει τη πραγματική αλλά και την εκτιμώμενη θέση του unknown κόμβου και επίσης τη τροχιά του κινητού beacon κόμβου. Να πούμε ότι για το πείραμα χρησιμοποιήθηκαν ως κόμβοι τα HP iPAQ PocketPC's εφοδιασμένα με 802.11b συμβατές κάρτες Lucent Orinoco Gold. Επίσης ο mobile beacon κόμβος ήταν όμοιος με τους προηγούμενους και μεταφέρονταν από ένα φορητό γράκι.



Σχήμα 10.7. : Πραγματική ('ο') και εκτιμώμενη θέση ('x') του unknown κόμβου και η τροχιά του mobile beacon κόμβου.

Τέλος το σχήμα 10.8 απεικονίζει τη τελική εκτίμηση θέσης για 12 unknown κόμβους. Κάποιες εκτιμήσεις θέσεων που αναφέρονται σε κόμβους μακριά από τη τροχιά του beacon κόμβου, έχουν μεγαλύτερη τυπική απόκλιση πράγμα το οποίο είναι αναμενόμενο με βάση αυτά που είπαμε παραπάνω. Για περισσότερες λεπτομέρειες και αποτελέσματα μπορείτε να ανατρέξετε στο [19].



Σχήμα 10.8. : Τελική εκτίμηση θέσεων των unknown κόμβων.

11 RELAXATION-BASED ΚΑΤΑΝΕΜΗΜΕΝΟΙ ΑΛΓΟΡΙΘΜΟΙ

Αυτή η κατηγορία αλγορίθμων ξεκινά με κόμβους που κάνουν μια αρχική εκτίμηση της θέσης τους χρησιμοποιώντας κάποια από τις γνωστές localization μεθόδους όπως την gradient μέθοδο που περιγράψαμε παραπάνω. Αυτές οι αρχικές θέσεις για του κόμβους βελτιώνονται, λαμβάνοντας εκτιμήσεις θέσης από γειτονικούς τους κόμβους.

Για παράδειγμα οι Savarese, Rabaey, Beutel [25], έφτιαξαν έναν αλγόριθμο ο οποίος βελτιώνει τις θέσεις που υπολογιζόταν από την gradient μέθοδο εφαρμόζοντας multilateration σε επίπεδο γειτονικών κόμβων. Πιο συγκεκριμένα κάθε κόμβος προσαρμόζει τη θέση του χρησιμοποιώντας προσωρινά τα μηνύματα των γειτόνων του σαν beacon μηνύματα.

Τέτοιου τύπου τεχνικές αν και μπορεί να είναι αποδοτικές, είναι πολύ ευαίσθητες στις αρχικές θέσεις που υπολογίζονται (πριν γίνει δηλαδή η τοπική ανταλλαγή μηνυμάτων). Μια κακή αρχική θέση μπορεί να οδηγήσει σε μεγάλες τυπικές αποκλίσεις από την πραγματική θέση.

12 COORDINATE SYSTEM STITCHING

Στο προηγούμενο κεφάλαιο αναφερθήκαμε σε τεχνικές που συνδύαζαν την ακρίβεια των κεντριοποιημένων localization σχημάτων με τα υπολογιστικά οφέλη των κατανεμημένων συστημάτων. Οι τεχνικές *coordinate system stitching* προσεγγίζουν το ίδιο πρόβλημα από διαφορετική σκοπιά. Οι τεχνικές αυτές περιλαμβάνουν τα εξής βασικά βήματα:

1. Διαχώρισε το δίκτυο σε μικρές επικαλυπτόμενες υπό-περιοχές (sub-region). Συνήα κάθε υπό-περιοχή αποτελείται από ένα κόμβο και τους one-hop γείτονες του.
2. Για κάθε υπό-περιοχή υπολόγισε ένα «τοπικό χάρτη» (local map), ο οποίος είναι στην ουσία μία αντιστοίχιση των κόμβων της υπό-περιοχής σε ένα σχετικό σύστημα συντεταγμένων.
3. Τελικά συγχώνευσε τις υπό-περιοχές χρησιμοποιώντας μία διαδικασία εγγραφής (registration) σε ένα σύστημα συντεταγμένων. Η έννοια της «εγγραφής» (registration) σε ένα σύστημα συντεταγμένων αναφέρεται στην αντιστοίχιση ενός συστήματος συντεταγμένων σε ένα διαφορετικό. Έτσι λοιπόν το βήμα 3, τοποθετεί όλες τις υπό-περιοχές σε ένα μοναδικό καθολικό σύστημα συντεταγμένων. Υπάρχουν αρκετοί αλγόριθμοι που κάνουν αυτό το βήμα βέλτιστα, από τη στιγμή που υπάρχει μία γρήγορη, ελαχίστων τετραγώνων βέλτιστη μέθοδος για registration σε ένα σύστημα συντεταγμένων.

Τα βήματα 1, 2 της γενικής μεθόδου είναι αυτά που διαφοροποιούν κάθε αλγόριθμο αυτής της κατηγορίας. Αντίθετα το βήμα 3 είναι σχεδόν ίδιο σε κάθε αλγόριθμο. Μία μέθοδος που προτάθηκε για τα βήματα 1, 2 είναι αυτή των Meertens και Fitzpatrick [26]. Σε αυτή τη μέθοδο κάθε υπό-περιοχή αποτελείται από ένα κόμβο και τους one-hop γείτονες του. Οι «τοπικοί χάρτες» υπολογίζονται στη συνέχεια επιλέγοντας αρχικά τρεις κόμβους οι οποίοι θα ορίσουν ένα σχετικό σύστημα συντεταγμένων. Στη συνέχεια χρησιμοποιώντας multilateration η μέθοδος προσθέτει επαναληπτικά κόμβους στο χάρτη, σχηματίζοντας έτσι ένα “multilateration sub-tree”.

Το βήμα 3 ή καλύτερα το βήμα «ράψιματος» (stitching) εφαρμόζει «εγγραφή» (registration) σε ένα σύστημα συντεταγμένων με ένα peer to peer τρόπο έτσι ώστε οι τοπικοί χάρτες να συγχωνευθούν σε ένα καθολικό. Ο τρόπος που γίνεται αυτό το «ράψιμο» περιγράφεται στα παρακάτω βήματα:

1. Ας υποθέσουμε ότι ο κόμβος που είναι υπεύθυνος για κάθε τοπικό χάρτη διαλέγει ένα ακέραιο ID συστήματος συντεταγμένων τυχαία.
2. Κάθε κόμβος επικοινωνεί με τους γείτονες του. Κάθε ζευγάρι κόμβων εκτελεί τα ακόλουθα βήματα:
 - a. Αν και οι δύο κόμβοι έχουν το ίδιο ID τότε δεν εκτελείτε κάποια επιπλέον ενέργεια.
 - b. Αν όμως έχουν διαφορετικά ID's τότε «έγγραψε» (register) το χάρτη του κόμβου με το μικρότερο ID με το χάρτη του κόμβου με το μεγαλύτερο

ID. Μετά από αυτό και οι δύο κόμβοι κρατάνε το υψηλότερο ID σαν να είναι το δικό τους.

3. Επανέλαβε το βήμα 2 μέχρι όλοι οι κόμβοι να έχουν το ίδιο ID. Τώρα όλοι οι κόμβοι θα έχουν συντεταγμένες βασισμένες σε ένα καθολικό σύστημα συντεταγμένων.

Οι τεχνικές *coordinate system stitching* μας δίνουν τη δυνατότητα να χρησιμοποιήσουμε αποδοτικούς local map αλγόριθμους οι οποίοι είναι υπολογιστικά ακριβοί για να εφαρμοστούν σε καθολικό επίπεδο. Όμως το μειονέκτημα αυτών των τεχνικών είναι ότι έχουν τη τάση να αφήνουν «ορφανούς» (orphan) κόμβους. Αυτό συμβαίνει είτε επειδή αυτοί δεν μπορούν προστεθούν σε ένα τοπικό χάρτη, είτε επειδή οι τοπικοί χάρτες αποτυγχάνουν να επικαλυφθούν επαρκώς με τους γειτονικούς χάρτες.

13 ΠΑΡΑΤΗΡΗΣΕΙΣ ΚΡΙΤΙΚΗ

Σε αυτό το κεφάλαιο θα προσπαθήσουμε να συνοψίσουμε τα πλεονεκτήματα και τα μειονεκτήματα των διάφορων κατηγοριών των localization τεχνικών που αναλύσαμε παραπάνω. Θα προσπαθήσουμε να εστιάσουμε στα tradeoff που προκύπτουν και στο πότε κάθε τεχνική είναι καταλληλότερη.

Ο πρώτος κύριος διαχωρισμός που κάναμε ανάμεσα στους αλγόριθμους είναι αυτοί που χρησιμοποιούν beacon κόμβους και αυτούς που δεν χρησιμοποιούν. Οι Beaconless αλγόριθμοι παράγουν θέσεις σε ένα σχετικό σύστημα συντεταγμένων, το οποίο προαιρετικά μπορεί να αντιστοιχιστεί σε ένα καθολικό σύστημα συντεταγμένων. Συχνά η δομή των δικτύων αισθητήρων κάνουν τη χρήση beacon κόμβων απαγορευτική και επίσης δεν απαιτούν ένα σύστημα καθολικών συντεταγμένων. Σε αυτή λοιπόν τη περίπτωση οι beaconless αλγόριθμοι δουλεύουν ικανοποιητικά. Από την άλλη μεριά οι αλγόριθμοι που απαιτούν beacon κόμβους μπορούν να χαρακτηριστούν πιο «ακριβή αλγόριθμοι». Αυτό για οι beacon κόμβοι διαθέτοντας εξοπλισμό GPS είναι πιο ακριβοί. Καμία φορά αυτό το επιπλέον κόστος δεν εξαργυρώνεται μιας και οι beacon κόμβοι είναι παύουν να είναι χρήσιμοι μετά τη βοήθεια για Localization που προσφέρουν. Φυσικά από τους beacon αλγόριθμους υπάρχουν κάποιοι που απαιτούν πολλούς beacon κόμβους για δουλέψουν αποδοτικά (π.χ. APIT) και κάποιοι που αρκούνται σε μικρότερη αναλογία beacon και unknown κόμβων. Ο αλγόριθμος του Mobile beacon που αναφέραμε παραπάνω δουλεύει αποδοτικά με τη βοήθεια ενός μόνο beacon κόμβου.

Το επόμενο στοιχείο που θα μπορούσε να διαχωρίσει τους localization αλγόριθμους είναι το hardware που απαιτούν. Όλοι οι sensor κόμβοι διαθέτουν ράδιο πομπούς και δέκτες και οι περισσότεροι είναι σε θέση να μετρήσουν την ισχύ του σήματος. Έτσι λοιπόν οι αλγόριθμοι που βασίζονται σε τεχνικές μέτρησης hop και RSSI απαιτούν το λιγότερο δυνατό hardware. Υπάρχουν όμως και τεχνικές οι οποίες απαιτούν πιο ειδικευμένο hardware όπως μικρόφωνα και ηχεία. Ένα τέτοιο παράδειγμα είναι το TDoA που αναλύσαμε παραπάνω. Σημαντικό από άποψη hardware είναι οι απαιτήσεις σε

επεξεργαστική ισχύ αλλά και σε μνήμη για έναν sensor κόμβο. Αλγόριθμοι όπως ο bounding-box και ο APIT δεν απαιτούν μεγάλη επεξεργαστική ισχύ. Ο APIT όμως απαιτεί αρκετή μνήμη.

Μία επίσης κατηγοριοποίηση που κάναμε παραπάνω είναι ανάλογα με το εάν ένας αλγόριθμος είναι κεντριοποιημένος ή είναι κατανεμημένος. Τυπικά οι κεντριοποιημένοι αλγόριθμοι είναι σε θέση να υπολογίζουν πιο ακριβείς θέσεις, και είναι ιδιαίτερα ανταγωνιστικοί σε καταστάσεις όπου η ακρίβεια των υπολογισμών θέσης είναι ιδιαίτερα σημαντική και η διαδικασία συγκέντρωσης των τοπικών δεδομένων σε ένα κεντρικό σταθμό βάσης δεν είναι χρονοβόρα και επιρρεπής σε λάθη. Από την άλλη μεριά οι κατανεμημένοι αλγόριθμοι που είναι συχνά μια τοπική προσέγγιση των κεντριοποιημένων αλγορίθμων, έχουν το πλεονέκτημα ότι δεν εξαρτώνται από κεντρικούς υπολογιστές και είναι καλύτερα κλιμακώσιμοι (scalable). Γενικά οι κατανεμημένοι αλγόριθμοι δεν αντιμετωπίζουν προβλήματα συμφόρησης, είναι φθηνότεροι και αποτελούν τη πιο αξιόπιστη λύση σε ultra scale sensor δίκτυα.

Άλλα κριτήρια σύγκρισης μεταξύ localization τεχνικών είναι η ενέργεια που καταναλώνουν και το κόστος επικοινωνίας που εισάγουν. Το κομμάτι της αποδοτικότητας ενέργειας είναι ένα από τα πιο σημαντικά θέματα στα sensor networks μιας και αυτά δουλεύουν με μπαταρία και δεν έχουν μια μόνιμη πηγή ενέργειας.

Γενικά η επιλογή του καταλληλότερου localization αλγόριθμου εξαρτάται από πολλούς παράγοντες από τους οποίους πολύ σημαντικός είναι η εφαρμογή για την οποία προορίζεται το δίκτυο αισθητήρων.

14 ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ

Όπως γενικά το πεδίο των sensor networks έτσι και οι localization αλγόριθμοι είναι στα πρωταρχικά τους βήματα. Τόσο οι αδυναμίες του localization hardware όσο και εξωγενείς παράγοντες (εξασθενήσεις πολλαπλών μονοπατιών, παρεμβολές,...κ.α.) επιβάλλουν την ανάπτυξη στιβαρών αλγορίθμων.

Μία καινούργια κατεύθυνση των sensor δικτύων που επηρεάζει άμεσα και τους localization αλγόριθμους είναι ότι μελλοντικά οι sensor κόμβοι μπορεί να είναι κινητοί. Άρα πρέπει να αναπτυχθούν νέοι localization αλγόριθμοι που θα μπορούν να αντεπεξέλθουν στη κίνηση των κόμβων. Υπάρχουν ήδη κάποιοι αλγόριθμοι που μπορούν να ανεχθούν ένα μικρό ποσό κίνησης αλλά απαιτείται περισσότερος πειραματισμός για αυτό το πρόβλημα. Κάποιοι ερευνητές έχουν προσπαθήσει να λύσουν αυτό το πρόβλημα με προσαρμοστική τοποθέτηση beacon κόμβων.

Επίσης κανένας από τους υπάρχοντες localization αλγόριθμους δεν μπορεί να δουλέψει αποδοτικά για μεγάλης κλίμακας δίκτυα αισθητήρων (π.χ. 10000 κόμβοι). Πιθανόν τέτοια δίκτυα να απαιτούν περισσότερο ιεραρχικούς αλγορίθμους.

ΜΕΡΟΣ 5^ο : ΠΡΩΤΟΚΟΛΛΑ ΔΡΟΜΟΛΟΓΗΣΗΣ

15 ΠΡΩΤΟΚΟΛΛΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΣΤΑ WSNs

Η δρομολόγηση δεδομένων (routing) στα WSNs δίκτυα είναι ένα εξαιρετικά ενδιαφέρον θέμα εξαιτίας των ιδιαίτερων χαρακτηριστικών αυτών των δικτύων που τα κάνουν να ξεχωρίζουν από τα άλλα ασύρματα δίκτυα. Πρώτον εξαιτίας του μεγάλου αριθμού των κόμβων αισθητήρων, είναι αδύνατη η ύπαρξη ενός καθολικού σχήματος διευθυνσιοδότησης, αφού το overhead για τη διατήρηση ID στους κόμβους είναι πολύ μεγάλο. Έτσι τα παραδοσιακά βασισμένα σε IP διευθύνσεις, πρωτόκολλα δρομολόγησης δεν μπορούν να εφαρμοστούν σε ένα WSN. Επίσης επειδή τα WSNs είναι ad-hoc όσον αφορά τη δομή τους, πρέπει να είναι σε θέση να αυτο-διαμορφώνονται για να μπορούν να δρομολογήσουν τα δεδομένα αποτελεσματικά (κοιτάξτε 3^ο μέρος της εργασίας). Τέλος πρέπει να σημειώσουμε ότι σε ένα WSN η λήψη των δεδομένων είναι αυτή που έχει σημασία και όχι το ID του κόμβου που αυτά προήλθαν.

Η ιδέα της δρομολόγησης στα WSNs σε αντίθεση με άλλα δίκτυα επικοινωνίας είναι πιο συγκεκριμένη, και απαιτεί τη διάδοση δεδομένων από ένα σύνολο κόμβων που τα έχουν ανιχνεύσει, προς ένα κέντρο ελέγχου που ονομάσαμε sink. Βέβαια αυτό δεν αποκλείει μία επικοινωνία άλλου τύπου, όπως multicast ή peer to peer.

Οι κόμβοι των WSNs όπως έχουμε προαναφέρει, είναι ιδιαίτερα περιορισμένοι στα αποθέματα ενέργειας και στις δυνατότητες υπολογισμού και αποθήκευσης. Έτσι ένα αποδοτικό routing πρωτόκολλο είναι αυτό που κάνει αποδοτική διαχείριση αυτών των πόρων. Επίσης αν και οι κόμβοι είναι συνήθως στατικοί, υπάρχουν εφαρμογές που απαιτούν κόμβους με δυνατότητα κίνησης. Αυτό βέβαια έχει ως αποτέλεσμα συχνές τοπολογικές αλλαγές, και συνεπώς άμεση επίδραση στη δρομολόγηση των δεδομένων. Από τα παραπάνω λοιπόν μπορούμε να συμπεράνουμε ότι οι κόμβοι των WSNs αλλά και οι απαιτήσεις από ένα πρωτόκολλο δρομολόγησης εξαρτώνται σε μεγάλο βαθμό από την εφαρμογή.

Τις περισσότερες φορές, ένα πρωτόκολλο δρομολόγησης δεδομένων πρέπει να συνδυάζεται και από ένα localization αλγόριθμο, μιας και μια πληροφορία (π.χ. πληροφορία για ανίχνευση καπνού σε ένα δάσος) μπορεί να είναι άχρηστη εάν δεν συνοδεύεται από τις συντεταγμένες που συνέβη το γεγονός. Επειδή η χρήση εξοπλισμού GPS είναι απαγορευτική λόγω κόστους, επιβάλλεται η εκτέλεση κάποιου localization αλγόριθμου όπως αναφέραμε παραπάνω.

Εξαιτίας της διαφορετικότητας των εφαρμογών, έχουν προταθεί πάρα πολλά πρωτόκολλα δρομολόγησης. Σε αυτό το κεφάλαιο δεν θα εμβαθύνουμε σε κάποιο πρωτόκολλο συγκεκριμένα, αλλά στην επόμενη παράγραφο θα κάνουμε μια κατηγοριοποίηση των routing αλγορίθμων ανάλογα με συγκεκριμένα χαρακτηριστικά τους, με βάση το [32]. Επίσης στη τελευταία παράγραφο του κεφαλαίου αυτού θα αναφερθούμε στο routing πρόβλημα του “*local detection and propagation*” (τοπική ανίχνευση και μετάδοση της πληροφορίας), το οποίο αναφέρεται στην ανίχνευση μιας τοπικής, κρίσιμης πληροφορίας, και στην multihop διάδοση της σε κάποιο κέντρο ελέγχου. Για το πρόβλημα αυτό έχουν αναπτυχθεί μία σειρά πρωτοκόλλων τα οποία παρουσιάζουν ιδιαίτερο ενδιαφέρον, μιας και δίνουν μια γεωμετρική ανάλυση του προβλήματος, και υποστηρίζονται από μία σειρά πιθανοτικών κυρίως αποδείξεων. Κάποια από αυτά τα πρωτόκολλα θα παρουσιαστούν σε επόμενα κεφάλαια αυτού του μέρους.

15.1 ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ROUTING ΠΡΩΤΟΚΟΛΛΩΝ

Σε αυτή τη παράγραφο θα προσπαθήσουμε να κατηγοριοποιήσουμε τα πρωτόκολλα δρομολόγησης για τα WSNs. Γενικά οι αλγόριθμοι δρομολόγησης για τα WSNs μπορούν να χωριστούν σε τρεις κύριες κατηγορίες: 1) flat-based routing 2) hierarchical-based routing 3) location-based routing.

Στη *flat-based* δρομολόγηση, σε όλους τους κόμβους τυπικά ανατίθενται ίδιοι ρόλοι και λειτουργικότητες και οι κόμβοι καλούνται να συνεργαστούν μεταξύ τους για να φέρουν εις πέρας κάποια εργασία. Σε αυτή τη κατηγορία πρωτοκόλλων μπορούμε να βάλουμε το πολύ δημοφιλές “Directed Diffusion” ([12]). Το Directed Diffusion δεν προτάθηκε με τη μορφή γενικευμένου πρωτοκόλλου, αλλά περισσότερο ως ένα παράδειγμα διάδοσης της πληροφορίας στα WSNs, που εξυπηρετούν μια ειδικευμένη εφαρμογή ανίχνευσης κίνησης μικρών ζώων. Η περαιτέρω ανάλυση του πρωτοκόλλου αυτού είναι πέρα από το σκοπό αυτής της εργασίας.

Στην *ιεραρχική δρομολόγηση*, και σε αντίθεση με την *flat-based* δρομολόγηση, κάθε κόμβος μπορεί να έχει διαφορετικό ρόλο στο δίκτυο. Σε μία ιεραρχική αρχιτεκτονική οι κόμβοι που διαθέτουν μεγαλύτερα ποσά ενέργειας μπορούν να αναλάβουν το ρόλο της επεξεργασίας και αποστολής δεδομένων, ενώ οι κόμβοι χαμηλότερης ενέργειας μπορούν απλά να ανιχνεύουν τα γεγονότα από το φυσικό περιβάλλον που είναι γύρω τους. Αυτό συνεπάγεται τη δημιουργία clusters (ομάδων κόμβων) όπου ο αρχηγός κάθε ομάδας (clusterhead) αναλαμβάνει ειδικά καθήκοντα συνεισφέροντας στη κλιμακωσιμότητα (scalability) και στην αποδοτικότητα κατανάλωσης ενέργειας του δικτύου. Η ιεραρχική δρομολόγηση είναι ένας αποδοτικός τρόπος εξοικονόμησης ενέργειας αφού μειώνεται ο αριθμός των μεταδόσεων στο κέντρο ελέγχου, μιας και τα δεδομένα συσσωρεύονται και στέλνονται μαζικά στο sink. Επίσης η ιεραρχική δρομολόγηση, μπορεί να χαρακτηριστεί σαν δρομολόγηση δύο επιπέδων, μιας και το πρώτο επίπεδο της αναφέρεται στην επιλογή clusterheads, ενώ το δεύτερο αναφέρεται στη πραγματική δρομολόγηση της πληροφορίας. Ένα πολύ δημοφιλές πρωτόκολλο ιεραρχικής δρομολόγησης είναι το LEACH ([15]). Το LEACH βασίζεται στην ιδέα των clusters την οποία προχωρά ένα βήμα παραπέρα. Στο LEACH οι κόμβοι του δικτύου αυτό-οργανώνονται σε clusters, όπου όμως τα clusterheads δεν είναι σταθερά, αλλά οι κόμβοι του δικτύου αποφασίζουν με κάποια τυχαιότητα αν θα είναι clusterheads ή όχι. Ο LEACH είναι επίσης δύο επιπέδων αλγόριθμος. Στη πρώτη φάση κάθε κόμβος αποφασίζει αν θα είναι clusterhead και όλοι οι κόμβοι ειδοποιούν το υπάρχον clusterhead για την απόφασή τους, για να μπορούν να οργανωθούν. Στη δεύτερη φάση το clusterhead συλλέγει τα μηνύματα του cluster του, και μεταδίδει ένα σύνθετο μήνυμα στο sink. Ο πίνακας 15.1 παρουσιάζει κάποιες βασικές διαφορές μεταξύ ιεραρχικών και flat-based αλγόριθμων δρομολόγησης.

Στην *location-based δρομολόγηση*, η θέση των κόμβων αισθητήρων λαμβάνεται υπόψη ώστε να δρομολογηθούν τα δεδομένα στο σταθμό βάσης. Για να προσδιοριστεί η τοποθεσία μπορεί να χρησιμοποιηθεί ένας αλγόριθμος localization από αυτούς που αναφέραμε παραπάνω. Ένας location-based αλγόριθμος δρομολόγησης είναι ο SPAN ([33]). Ο αλγόριθμος αυτός επιλέγει κάποιους κόμβους αισθητήρων σαν συντονιστές ανάλογα με τη θέση τους. Οι συντονιστές κόμβοι αποτελούν ένα δίκτυο ραχοκοκαλιάς (backbone network) και είναι υπεύθυνοι να προωθούν τα δεδομένα. Ένας κόμβος μπορεί να γίνει συντονιστής εάν δύο γείτονες ενός μη-συντονιστή κόμβου δεν μπορούν να επικοινωνήσουν μεταξύ τους άμεσα, ή μέσω ενός ή δύο συντονιστών (3 hop reachability). Για περισσότερες πληροφορίες μπορείτε να δείτε το [33].

Πέρα από την κατηγοριοποίηση που αναφέραμε παραπάνω τα πρωτόκολλα δρομολόγησης μπορούν να κατηγοριοποιηθούν σε **1) Multipath-based 2) Query-based 3) Negotiation-based 4) QoS-based** ανάλογα με τη λειτουργία τους. Για παράδειγμα το SAR (Sequential Assignment Routing) είναι ένα πρωτόκολλο που εισήγαγε την ιδέα της Quality of Service στα WSNs. Η απόφαση δρομολόγησης στο SAR εξαρτάται από τρεις παράγοντες: 1) τους ενεργειακούς πόρους, 2) το QoS κάθε μονοπατιού και 3) το επίπεδο προτεραιότητας κάθε πακέτου. Για να αποφευχθεί η αποτυχία αποστολής δεδομένων μέσω ενός μονοπατιού, έχει υιοθετηθεί μια multipath προσέγγιση. Άρα το πρωτόκολλο θα μπορούσε να ήταν και multipath-based.

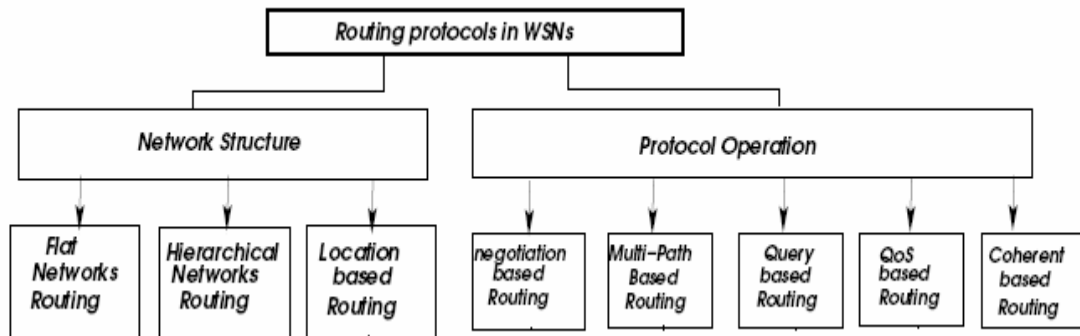
Θεωρώντας ακόμα μία διαφορετική ταξινόμηση των πρωτοκόλλων δρομολόγησης μπορούμε να θεωρήσουμε ότι χωρίζονται στις κατηγορίες: **1) proactive 2) reactive 3) υβριδικά (hybrid)**. Στα proactive πρωτόκολλα όλες οι διαδρομές υπολογίζονται εκ των προτέρων, πριν ακόμα δηλαδή τις χρειαστούμε. Αντίθετα στα reactive πρωτόκολλα έχουμε μία on-demand, κατ' απαίτηση δηλαδή δημιουργία των διαδρομών. Τα υβριδικά πρωτόκολλα αποτελούν ένα συνδυασμό των δύο παραπάνω ιδεών. Όταν οι κόμβοι αισθητήρων είναι στατικοί, είναι προτιμότερο να χρησιμοποιούμε πρωτόκολλα που βασίζονται σε πίνακες δρομολόγησης από ότι reactive. Αυτό γιατί ένα σημαντικό ποσό ενέργειας καταναλώνεται στην ανακάλυψη και εγκατάσταση των διαδρομών στα reactive πρωτόκολλα.

Ακόμα μια διαφορετική ταξινόμηση πρωτοκόλλων είναι τα *συνεργατικά ή cooperative* πρωτόκολλα. Στην cooperative δρομολόγηση οι κόμβοι στέλνουν δεδομένα σε ένα κεντρικό κόμβο όπου τα δεδομένα συσσωρεύονται και επεξεργάζονται περαιτέρω, ώστε να καταναλώνεται ενέργεια λόγω επεξεργασίας στους κόμβους.

Ανάλογα με τα ιδιαίτερα χαρακτηριστικά του, κάθε πρωτόκολλο μπορεί να ενταχθεί σε μία ή περισσότερες από τις παραπάνω κατηγορίες. Οι κατηγορίες που αναφέραμε μπορούν να συνοψιστούν στο σχήμα 15.1. Στο [32] αναφέρονται και επιπλέον πρωτόκολλα που ανήκουν στη κάθε κατηγορία. Τέλος στο σχήμα 15.2 φαίνεται μια γενικότερη κατηγοριοποίηση αλλά και σύγκριση των δημοφιλέστερων πρωτοκόλλων δρομολόγησης.

Πίνακας 15.1. Hierarchical vs Flat-based δρομολόγηση.

Hierarchical routing	Flat routing
Reservation-based scheduling	Contention-based scheduling
Collisions avoided	Collision overhead present
Reduced duty cycle due to periodic sleeping	Variable duty cycle by controlling sleep time of nodes
Data aggregation by clusterhead	node on multihop path aggregates incoming data from neighbors
Simple but non-optimal routing	Routing can be made optimal but with an added complexity.
Requires global and local synchronization	Links formed on the fly without synchronization
Overhead of cluster formation throughout the network	Routes formed only in regions that have data for transmission
Lower latency as multiple hops network formed by clusterheads always available	Latency in waking up intermediate nodes and setting up the multipath
Energy dissipation is uniform	Energy dissipation depends on traffic patterns
Energy dissipation cannot be controlled	Energy dissipation adapts to traffic pattern
Fair channel allocation	Fairness not guaranteed



Σχήμα 15.1. : Ταξινόμηση Πρωτοκόλλων Δρομολόγησης στα WSNs.

	Classification	Mobility	Position Awareness	Power Usage	Negotiation based	Data Aggregation	Localization	QoS	State Complexity	Scalability	Multipath	Query based
SPIN	Flat	Possible	No	Limited	Yes	Yes	No	No	Low	Limited	Yes	Yes
Directed Diffusion	Flat	Limited	No	Limited	Yes	Yes	Yes	No	Low	Limited	Yes	Yes
Rumor Routing	Flat	Very Limited	No	N/A	No	Yes	No	No	Low	Good	No	Yes
GBR	Flat	Limited	No	N/A	No	Yes	No	No	Low	Limited	No	Yes
MCFA	Flat	No	No	N/A	No	No	No	No	Low	Good	No	No
CADR	Flat	No	No	Limited	No	Yes	No	No	Low	Limited	No	No
COUGAR	Flat	No	No	Limited	No	Yes	No	No	Low	Limited	No	Yes
ACQUIRE	Flat	Limited	No	N/A	No	Yes	No	No	Low	Limited	No	Yes
EAR	Flat	Limited	No	N/A	No	No	No	No	Low	Limited	No	Yes
LEACH	Hierarchical	Fixed BS	No	Maximum	No	Yes	Yes	No	CHs	Good	No	No
TEEN & APTEEN	Hierarchical	Fixed BS	No	Maximum	No	Yes	Yes	No	CHs	Good	No	No
PEGASIS	Hierarchical	Fixed BS	No	Maximum	No	No	Yes	No	Low	Good	No	No
MECN & SMECN	Hierarchical	No	No	Maximum	No	No	No	No	Low	Low	No	No
SOP	Hierarchical	No	No	N/A	No	No	No	No	Low	Low	No	No
HPAR	Hierarchical	No	No	N/A	No	No	No	No	Low	Good	No	No
VGA	Hierarchical	No	No	N/A	Yes	Yes	Yes	No	CHs	Good	Yes	No
Sensor aggregate	Hierarchical	Limited	No	N/A	No	Yes	No	No	Low	Good	No	Possible
TTDD	Hierarchical	Yes	Yes	Limited	No	No	No	No	Moderate	Low	Possible	Possible
GAF	Location	Limited	No	Limited	No	No	No	No	Low	Good	No	No
GEAR	Location	Limited	No	Limited	No	No	No	No	Low	Limited	No	No
SPAN	Location	Limited	No	N/A	Yes	No	No	No	Low	Limited	No	No
MFR, GEDIR	Location	No	No	N/A	No	No	No	No	Low	Limited	No	No
GOAFR	Location	No	No	N/A	No	No	No	No	Low	Good	No	No
SAR	QoS	No	No	N/A	Yes	Yes	No	Yes	Moderate	Limited	No	Yes
SPEED	QoS	No	No	N/A	No	No	No	Yes	moderate	Limited	No	Yes

Σχήμα 15.2. : Ταξινόμηση και σύγκριση δημοφιλών πρωτοκόλλων δρομολόγησης στα WSNs.

15.2 ΠΡΟΒΛΗΜΑ ΤΟΠΙΚΗΣ ΑΝΙΧΝΕΥΣΗΣ ΚΑΙ ΔΙΑΔΟΣΗΣ

Το πρόβλημα της «τοπικής ανίχνευσης και διάδοσης της πληροφορίας» (*Problem of Local Detection and Propagation*) αναφέρεται στη τοπική ανίχνευση ενός σημαντικού γεγονότος και την αποτελεσματική τόσο χρονικά όσο και ενεργειακά μετάδοση του, σε ένα κέντρο ελέγχου. Αυτό το κέντρο ελέγχου όπως έχουμε προαναφέρει, είναι ένα σταθμός βάσης που καλείται να αναλάβει δράση μετά την

αναφορά του γεγονότος (π.χ. μπορεί να είναι ένα πυροσβεστικό τμήμα στη περίπτωση που έχουμε μια WSN εφαρμογή για ανίχνευση πυρκαγιάς). Το κέντρο ελέγχου μπορεί να υλοποιηθεί με βάση τα δικτυακά μοντέλα που αναφέραμε στη παράγραφο 3.3.

Τα περισσότερα δίκτυα WSNs αποτελούνται από ένα μεγάλο αριθμό πολύ μικρών κόμβων αισθητήρων (smart dust networks) που κατανέμονται πυκνά στο πεδίο ενδιαφέροντος. Εξαιτίας αυτής της πυκνής κατανομής, οι κόμβοι αισθητήρων είναι πολύ κοντά ο ένας στον άλλο και έτσι για τους λόγους που αναφέραμε στη παράγραφο 3.1, ευνοείται η multihop διάδοση της πληροφορίας αφού εξοικονομούμε έτσι μεγάλα ποσά ενέργειας. Επίσης η multihop διάδοση πέρα από το ότι μας απαλλάσσει από τις παρενέργειες το μεταδόσεων μεγάλης ενέργειας, είναι αποτελεσματική όταν στο πεδίο ενδιαφέροντος που είναι κατανεμημένοι οι κόμβοι, υπάρχουν πολλά φυσικά εμπόδια. Τέλος η multihop διάδοση αποδεικνύεται πιο ασφαλής σε περίπτωση που κάποιος προσπαθήσει να ανιχνεύσει το μέσο. Για όλους τους παραπάνω λόγους και επιπλέον αυτούς που αναφέραμε στην παράγραφο 3.1, στο πρόβλημα της τοπικής ανίχνευσης και διάδοσης της πληροφορίας, θα προτιμήσουμε τη multihop διάδοση και έτσι μπορούμε να χαρακτηρίσουμε το πρόβλημα ως *Problem of Local Detection and Multihop Propagation*.

Στα επόμενα κεφάλαια, θα παρουσιάσουμε πρωτόκολλα που αναφέρονται στο πρόβλημα της «τοπικής ανίχνευσης και διάδοσης της πληροφορίας», και δίνουν μια γεωμετρική και πιθανοτική διάσταση σε αυτό.

16 ΠΡΩΤΟΚΟΛΛΟ LTP

Το πρωτόκολλο LTP ή Local Target Protocol, αναφέρεται στο πρόβλημα της «τοπικής ανίχνευσης και διάδοσης της πληροφορίας» και χρησιμοποιεί το μοντέλο πηγής-τείχους που αναφέραμε στα μοντέλα δικτύων (παράγραφος 3.3). Πριν αναλύσουμε τη λειτουργία του πρωτοκόλλου, θα δούμε πως μοντελοποιείται το πρόβλημα του *local detection and propagation*, καθώς επίσης θα ορίσουμε και κάποιες μετρικές.

16.1 Το ΠΡΟΒΛΗΜΑ

Έστω ότι τυχαία επιλέγεται ένα particle p από το «σύννεφο» των smart dust κόμβων και του δίνεται η δυνατότητα να παρακολουθεί ένα τοπικό κρίσιμο γεγονός E . Γενικά το πρόβλημα διάδοσης P ορίζεται ως εξής:

«Πώς μπορεί ένα particle p , σε συνεργασία με τους άλλους κόμβους του δικτύου να διαδώσει τη πληροφορία που αναφέρεται σε ένα γεγονός E , σε ένα Wall W ;»

Ορισμός 3: Έστω $h_{opt}(p, W)$ είναι ο βέλτιστος αριθμός των hops (δηλαδή οι hops πάνω στη κάθετη ευθεία που ενώνει το p με το W) που απαιτούνται ώστε η πληροφορία να φτάσει το wall, στην ιδανική περίπτωση όπου πάντα υπάρχουν particles που ανά ζεύγη απέχουν R στη κάθετη γραμμή, από τον p στο W . Με R

συμβολίζεται η εμβέλεια κάθε particle. Επιπλέον έστω Π είναι ένα πρωτόκολλο διάδοσης σε smart dust δίκτυα, που χρησιμοποιεί ένα μονοπάτι διάδοσης μήκους $L(\Pi, p, W)$ για να στείλει πληροφορία που προέκυψε από ένα γεγονός E , σε ένα wall W . Τέλος έστω $h(\Pi, p, W)$ είναι ο αριθμός των hops (και άρα των μεταδόσεων) που απαιτούνται για να φτάσει η πληροφορία στο wall W . Τότε η *αποδοτικότητα σε hops* (ή *hops efficiency*) του πρωτοκόλλου Π δίνεται από το τύπο:

$$C_h = \frac{h(\Pi, p, W)}{h_{opt}(p, W)}.$$

Είναι σαφές ότι ο αριθμός των hops (και άρα ο αριθμός των μεταδόσεων) που απαιτούνται, χαρακτηρίζει τη κατανάλωση ενέργειας για τη διάδοση της πληροφορίας E στο wall. Αξίζει να σημειώσουμε εδώ ότι $h_{opt} = \lceil d(p, W) / R \rceil$, όπου $d(p, W)$ είναι η κάθετη απόσταση από το p στον wall W .

Στη περίπτωση που το πρωτόκολλο Π βασίζεται στην τυχαιότητα (randomized), ή στη περίπτωση που η κατανομή των particles είναι τυχαία, ο αριθμός των hops h καθώς και η αποδοτικότητα σε hops C_h , είναι τυχαίες μεταβλητές, και έτσι μελετάμε τις μέσες τους τιμές.

Ο λόγος για τους παραπάνω ορισμούς είναι ότι όταν το particle p (ή οποιοδήποτε ενδιάμεσο particle κατά τη διάρκεια μετάδοσης της πληροφορίας στο wall) ψάξει για ένα άλλο particle που θα είναι όσο δυνατόν πιο κοντά στο wall για να του παραδώσει τη πληροφορία E , μπορεί να μην βρει κάποιο που βρίσκεται στη πάνω κάθετη γραμμή που ενώνει το p με το wall W . Δηλαδή μπορεί να μη βρει κάποιο γείτονα ώστε να του μεταδώσει «βέλτιστα» τη πληροφορία. Αυτό το γεγονός μπορεί να οφείλεται σε τρεις λόγους:

1. Εξαιτίας της αρχικής κατανομής των κόμβων στο πεδίο ενδιαφέροντος και αφού θεωρούμε ότι σε αυτό το μοντέλο δικτύου ότι οι κόμβοι δεν κινούνται, μπορεί να μην υπάρχει κάποιο particle σε αυτή τη κατεύθυνση.
2. Ακόμα και να υπάρχουν κόμβοι σε αυτή τη κατεύθυνση, αυτοί μπορεί να έχουν καταστραφεί ή να μην έχουν επαρκή αποθέματα ενέργειας.
3. Οι κόμβοι μπορεί να είναι προσωρινά σε μια κατάσταση ύπνου (sleep mode) και έτσι να μην μπορούν να λάβουν κάποιο μήνυμα.

Να σημειώσουμε εδώ ότι οποιαδήποτε κατανομή των particle στο πεδίο ενδιαφέροντος μπορεί να μην επιτρέψει (είναι το πιθανότερο) μια μετάδοση με ένα βέλτιστο αριθμό hops. Πάντως το h_{opt} που αναφέραμε παραπάνω μπορεί να αποτελέσει μετρική για την αποδοτικότητα κάποιου πρωτοκόλλου ως προς τη κατανάλωση ενέργειας.

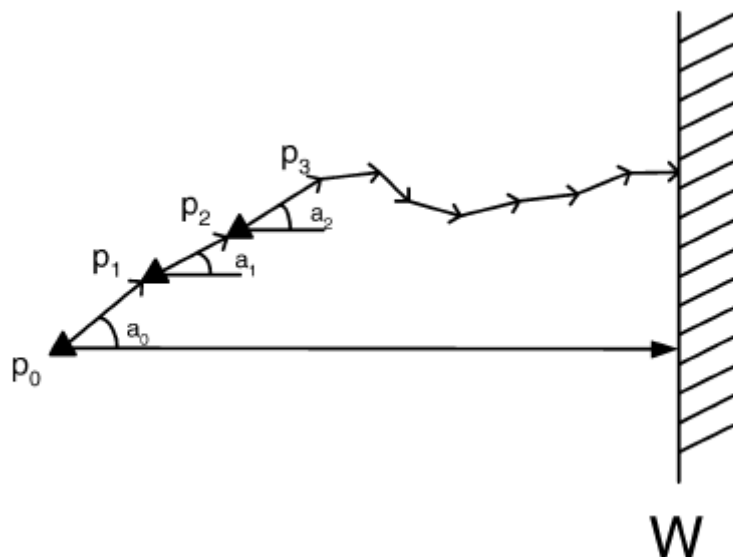
16.2 LOCAL TARGET PROTOCOL

Ας υποθέσουμε ότι $d(p_i, p_j)$ είναι η απόσταση (κατά μήκος της αντίστοιχης κάθετης γραμμής προς τον W) ανάμεσα στα particles p_i, p_j και επίσης $d(p_i, W)$ είναι η κάθετη απόσταση μεταξύ p_i και wall W. Ας υποθέσουμε επίσης ότι $info(E)$ είναι η πληροφορία που προκύπτει μετά την ανίχνευση κάποιου σημαντικού γεγονότος E, το οποίο και πρέπει να διαδοθεί στο σταθμό βάσης. Στο LTP πρωτόκολλο, κάθε particle p' που έχει λάβει την $info(E)$ από ένα p (ο p μπορεί να έχει λάβει τη πληροφορία πιθανότητα μέσω άλλων particles), πρέπει να κάνει τα ακόλουθα βήματα:

- *Search phase*: Αυτή η φάση κάνει μια περιοδική εκπομπή χαμηλής ενέργειας beacon μηνυμάτων, με σκοπό να ανακαλύψει ένα particle πιο κοντά στον W από ότι είναι το ίδιο (το p' δηλαδή). Ουσιαστικά αναζητείται ένα particle p'' όπου θα ισχύει $d(p'', W) < d(p', W)$.
- *Direct transmission phase*: Στη συνέχεια ο p' στέλνει την $info(E)$ στον p'' κατά μήκος της ευθεία νοητής γραμμής που τους ενώνει (π.χ. μέσω laser), καταναλώνοντας βέβαια χρόνο και ενέργεια.
- *Backtrack phase*: Εάν διαδοχικές επαναλήψεις της search phase αποτύχουν να ανακαλύψουν ένα κόμβο πιο κοντά στο wall, τότε ο p' στέλνει την $info(E)$ στο p (στο κόμβο δηλαδή από τον οποίο την έλαβε).

Να σημειώσουμε εδώ ότι εκ των προτέρων μπορούμε να υπολογίσουμε τον αριθμό των επαναλήψεων που απαιτεί η search phase, χρησιμοποιώντας τη πιθανότητα επιτυχίας κάθε search φάσης. Αυτό το όριο μπορεί να χρησιμοποιηθεί για να αποφασιστεί τότε θα γίνει οπισθοχώρηση (backtrack).

Να σημειώσουμε επίσης ότι η μέγιστη απόσταση $d(p', p'')$ είναι R (εμβέλεια μετάδοσης). Για να γίνει κατανοητό πως γίνεται η διάδοση πληροφορίας με βάση τις φάσεις που αναφέραμε παραπάνω, παραθέτουμε το σχήμα 16.1.



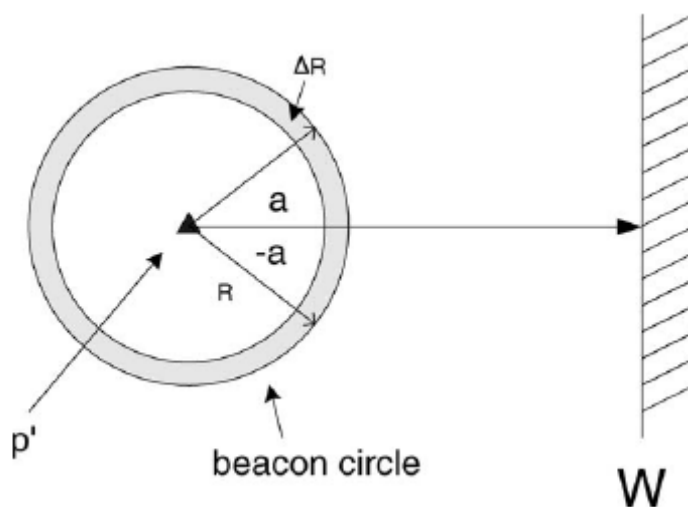
Σχήμα 16.1. : Παράδειγμα μετάδοσης στο LTP.

Για να μπορέσει να γίνει μια ακριβής ανάλυση τέτοιου τύπου smart dust πρωτοκόλλων, θεωρείται η εξής απλουστευτική υπόθεση. Η search φάση χρειάζεται μηδενικό χρόνο για να εκτελεστεί, και επίσης πάντα ανακαλύπτεται ένα particle p'' για να γίνει η περαιτέρω διάδοση της πληροφορίας μέσα σε ένα ημικύκλιο με κέντρο p προς την κατεύθυνση του W . Η υπόθεση όμως αυτή πάντα μπορεί να γίνει πιο ρεαλιστική αν θεωρήσουμε τα παρακάτω:

- Η φάση ανακάλυψης (search phase) τρέχει για πολλές επαναλήψεις μέχρι να βρεθεί particle p'' . Αυτό έχει νόημα μόνο αν όντως υπάρχει ένα particle p'' αλλά μπορεί να βρισκόταν σε κατάσταση sleep όταν διεξάγονταν οι search φάσεις.
- Μπορούμε να θεωρήσουμε αντί για ένα ημικύκλιο, ένα κυκλικό τομέα που ορίζεται από του κύκλους με ακτίνα $R - \Delta R$ και R και βέβαια να λάβουμε υπόψη μας πόσο πυκνό είναι το σύννεφο smart dust (σχήμα 16.2).
- Αν το πρωτόκολλο κατά τη διάρκεια της φάσης αναζήτησης αποτύχει να βρει κάποιο particle προς το wall μπορεί να οπισθοχωρήσει (backtrack).

Στην ανάλυση του πρωτοκόλλου δεν θεωρείται η ενέργεια που ξοδεύεται κατά τη φάση αναζήτησης. Αλλά ακόμα και αν η ενέργεια αυτή, είναι συγκρίσιμη με αυτή που καταναλώνεται για τη πραγματική μετάδοση της πληροφορίας, ο αριθμός των hops μπορεί να μετρήσει διπλά (η συνολική ενέργεια που ξοδεύεται στο σύστημα φράσσεται από ένα πολλαπλάσιο της ενέργειας των πραγματικών μεταδόσεων που γίνονται σε αυτό).

Στην ανάλυση θεωρείται ότι η θέση του p'' είναι ομοιόμορφη στο τόξο γωνίας $2a$ γύρω από τη κάθετη γραμμή που ενώνει το p' με τον W . Επίσης κάθε μετάδοση δεδομένων (one hop) χρειάζεται σταθερό χρόνο t (έτσι οι «hops» και η αποδοτικότητα σε χρόνο του πρωτοκόλλου συμπίπτουν). Τέλος θεωρείται ότι η επιλογή του στόχου για διάδοση της πληροφορίας είναι τυχαία και ανεξάρτητη από τις προηγούμενες, με την έννοια ότι η επιλογή κατανέμεται ομοιόμορφα στους κόμβους στο τόξο $(-a, a)$.



Σχήμα 16.2. : Παράδειγμα της φάσης αναζήτησης.

Μπορούμε εύκολα να αντιληφθούμε ότι οι παραπάνω υποθέσεις μπορεί είναι μην είναι απόλυτα ρεαλιστικές, όμως επιτρέπουν να γίνει μία ακριβής ανάλυση. Από την ανάλυση αυτή στο [34] αποδεικνύεται το εξής λήμμα για το LTP.

Λήμμα 1: Η μέση αποδοτικότητα σε hops του LTP πρωτοκόλλου στη περίπτωση της α -ομοιόμορφης κατανομής είναι $E(C_h) \approx a/\sin \alpha$ για μεγάλο h_{opt} . Επίσης ισχύει $1 \leq E(C_h) \leq \pi/2 \approx 1.57$ για $0 \leq \alpha \leq \pi/2$.

Για το LTP υπάρχουν αρκετές παραλλαγές που μπορούμε να θεωρήσουμε. Μία από αυτές αναλύεται στο [34] και ονομάζεται m2TP (“min two uniform targets”). Στο m2TP η φάση αναζήτησης επιστρέφει δύο particles p'' , p''' μέσα στο τόξο $(-a, a)$ και το πρωτόκολλο επιστρέφει το καλύτερο από τα δύο σημεία το οποίο είναι το particle που είναι κοντύτερα στο wall. Θα μπορούσε βέβαια το πρωτόκολλο να μην επιστρέφει μόνο δύο particles, αλλά όλα που βρίσκονται στο τόξο $(-a, a)$ και να επιλέγεται το καλύτερο σύμφωνα με το κριτήριο της ελάχιστης απόστασης από το wall. Ένα άλλο κριτήριο θα ήταν να θεωρούσαμε ως το καλύτερο particle αυτό που βρίσκεται πιο κοντά στην ευθεία που ενώνει το p με το W . Έτσι θα στέλναμε τη πληροφορία όσο γίνεται πιο κοντά στο βέλτιστο μονοπάτι.

17 ΠΡΩΤΟΚΟΛΛΟ PFR

Το πρωτόκολλο PFR (ή Probabilistic Forwarding Protocol) αναφέρεται στο πρόβλημα της «τοπικής ανίχνευσης και διάδοσης πληροφορίας» και προσπαθεί να ελαχιστοποιήσει τη κατανάλωση ενέργειας ευνοώντας πιθανοτικά συγκεκριμένα μονοπάτια για τοπική διάδοση της πληροφορίας προς τον sink. Για αυτό το λόγο ονομάζεται και Probabilistic Forwarding Protocol.

Το πρωτόκολλο PFR υιοθετεί μία δισδιάστατη lattice κατανομή (αναλύεται στη παράγραφο 3.3, μαζί με τα υπόλοιπα δικτυακά μοντέλα), η οποία φαίνεται στο σχήμα 3.8. Το PFR υποθέτει λοιπόν μια $n \times n$ lattice κατανομή με n^2 particles, κάθε ένα από τα οποία έχει εύρος μετάδοσης R και έχει τοποθετηθεί σε οριζόντια και κάθετη απόσταση δ από κάποιο άλλο. Να σημειωθεί εδώ, ότι το δ εξαρτάται από το R και επίσης χαρακτηρίζει τη πυκνότητα του δικτύου. Συγκεκριμένα για το πρωτόκολλο επιλέχθηκε $R = \delta\sqrt{2}$. Στην ανάλυση του PFR θεωρείτε για λόγους απλότητας ότι $\delta=1$. Επίσης υπενθυμίζουμε ότι κάθε particle (εκτός από αυτά που βρίσκονται στα όρια του lattice) έχει οκτώ άμεσους γείτονες μέσα στο εύρος μετάδοσης του. Τέλος υπενθυμίζουμε ακόμα ότι στο μοντέλο ο sink S είναι ένα σημείο στη περιοχή του δικτύου και αντιπροσωπεύει το κέντρο ελέγχου.

Στο πρωτόκολλο PFR θεωρείτε ότι κάθε particle έχει τις εξής ιδιότητες:

1. Μπορεί να εκτιμήσει την κατεύθυνση μια λαμβανόμενης μετάδοσης (αυτό μπορεί να γίνει με τη τεχνολογία κατευθυντικών κεραιών).
2. Μπορεί να εκτιμήσει την απόσταση από ένα γειτονικό κόμβο που του μετέδωσε κάποια πληροφορία (βασίζόμενος στην εξασθένηση του λαμβανόμενου σήματος).

3. Γνωρίζει την κατεύθυνση προς τον sink (δεν είναι απαραίτητο να είναι γνωστές οι ακριβείς συνταγμένες). Αυτό μπορεί να γίνει με τη βοήθεια μια φάσης εγκατάστασης, όπου ο πολύ ισχυρός ενεργειακά sink, εκπέμπει τη πληροφορία για τη κατεύθυνση σε όλους τους κόμβους του δικτύου.
4. Όλοι οι κόμβοι του δικτύου έχουν ένα κοινό σύστημα συντεταγμένων.

Αν και οι παραπάνω υποθέσεις φαίνονται μη-ρεαλιστικές, όπως αναφέρεται και στο [11] υπάρχουν εναλλακτικές λύσεις και διαφορετικοί τρόποι υλοποίησης τους. Το βασικό πλεονέκτημα του PFR είναι ότι δεν απαιτεί οι κόμβοι του να έχουν GPS εξοπλισμό. Επίσης δεν είναι απαραίτητο κάθε κόμβος να γνωρίζει τη καθολική δομή του δικτύου (π.χ. το μέγεθος του, τις θέσεις των κόμβων κ.α.).

17.1 Το ΠΡΟΒΛΗΜΑ

Η διατύπωση του προβλήματος δεν αλλάζει από αυτή του LTP αν θεωρήσουμε ότι στη περίπτωση του PFR έχουμε sink και όχι wall. Ας υποθέσουμε λοιπόν ότι ένα particle P ανιχνεύει ένα σημαντικό γεγονός E . Τότε το πρόβλημα διάδοσης είναι το εξής:

«Πώς μπορεί ένα particle P , σε συνεργασία με τους άλλους κόμβους του δικτύου να διαδώσει αποδοτικά τη πληροφορία που αναφέρεται σε ένα γεγονός E , σε ένα Sink S ;»

Για να μειωθεί όσο γίνεται περισσότερο η κατανάλωση ενέργειας σε ένα WSN, είναι επιθυμητό να αποφύγουμε να πλημμυρίσουμε το δίκτυο με μηνύματα (flooding) και να μειώσουμε τον αριθμό των hops που λαμβάνουν μέρος στη διαδικασία διάδοσης της πληροφορίας. Βέβαια ένα αποδοτικό πρωτόκολλο θα πρέπει να επιτυγχάνει να στείλει τη σημαντική πληροφορία στο κέντρο ελέγχου. Στο [11] το πρόβλημα του local detection and propagation ορίζεται ως εξής:

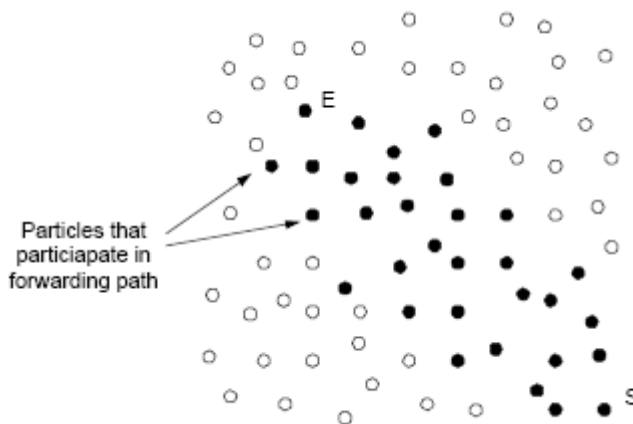
Έστω E η θέση του particle που ανίχνευσε ένα σημαντικό γεγονός. Επίσης έστω S η θέση του sink. Κάθε πρωτόκολλο Π που λύνει το πρόβλημα διάδοσης της πληροφορίας, πρέπει να ικανοποιεί τα εξής:

- **Correctness** (Ορθότητα): Το Π πρέπει να μπορεί να εγγυηθεί ότι τα δεδομένα φτάνουν στη θέση S , δοθέντος ότι ολόκληρο το δίκτυο υπάρχει και είναι λειτουργικό.
- **Robustness** (Στιβαρότητα): Το Π πρέπει να μπορεί να εγγυηθεί ότι τα δεδομένα φτάνουν σε αρκετά σημεία κοντά στο S , σε περιπτώσεις όπου κάποια κομμάτια του δικτύου δεν είναι λειτουργικά, αλλά βέβαια δεν έχει απομονωθεί ο sink.
- **Efficiency** (Αποδοτικότητα): Αν το Π ενεργοποιεί k particles κατά τη διάρκεια της λειτουργίας του, τότε είναι επιθυμητό ο λόγος $r = \frac{k}{N}$ για το Π να

είναι μικρός ώστε το πρωτόκολλο να είναι ενεργειακά αποδοτικό. Υπενθυμίζουμε ότι N είναι ο αριθμός των particles στο δίκτυο.

17.2 ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΠΡΩΤΟΚΟΛΛΟΥ

Η βασική ιδέα στο PFR, είναι ότι το πρωτόκολλο πιθανοτικά ευνοεί τη μετάδοση προς τον sink, μέσω μια στενής ζώνης από particles που βρίσκονται γύρω από τη γραμμή που συνδέει το particle που ανίχνευσε το γεγονός με το sink (σχήμα 17.1). Υπενθυμίζουμε ότι η μετάδοση κατά μήκος αυτής της γραμμής είναι ενεργειακά βέλτιστη. Βέβαια δεν είναι δυνατόν πάντα να επιτύχουμε αυτή τη βέλτιστη μετάδοση, κυρίως επειδή κάποιοι κόμβοι κατά μήκος αυτής της γραμμής μπορεί να είναι ανενεργοί είτε μόνιμα (μπορεί να έχουν εξαντληθεί να ενεργειακά αποθέματα τους) είτε παροδικά (μπορεί οι κόμβοι να έχουν μπει σε κατάσταση sleep για να εξοικονομήσουν ενέργεια.). Επιπλέον λόγοι μπορεί να είναι α) κάποιες φυσικές καταστροφές των κόμβων, β) κάποια σκόπιμη μετακίνηση τους από το διαχειριστή του δικτύου, γ) γενικότερα κάποια αλλαγή στη θέση των κόμβων εξαιτίας διάφορων παραγόντων (ανθρώπινη παρέμβαση, καιρικές συνθήκες), δ) φυσικά εμπόδια που μπλοκάρουν την επικοινωνία.



Σχήμα 17.1. : Στενή ζώνη από particles.

Το πρωτόκολλο PFR έχει δύο κύριες φάσεις:

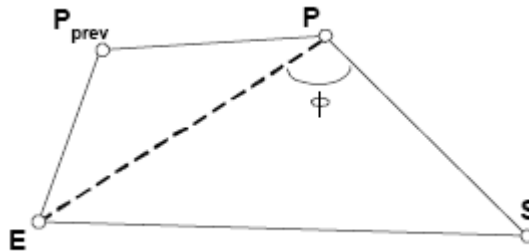
1. **The Front Creation Phase** (Φάση δημιουργίας «πρόσοψης»): Στη 1^η φάση, αρχικά χτίζετε (με τη χρήση περιορισμένου από άποψη γύρων flooding) μία επαρκώς μεγάλη «πρόσοψη» από particles για να εγγυηθεί την επιβίωση της διαδικασίας διάδοσης της πληροφορίας. Κατά τη διάρκεια αυτής της φάσης, κάθε particle που έχει λάβει κάποια δεδομένα για να τα προωθήσει, πρέπει με ένα ντετερμινιστικό τρόπο να τα προωθήσει προς το sink. Πιο συγκεκριμένα, για ένα μεγάλο αριθμό βημάτων $s = 180\sqrt{2}$, κάθε particle πρέπει να εκπέμπει τη πληροφορία σε όλους του τους γείτονες, προς την κατεύθυνση του sink (selective

flooding). Ο αριθμός αυτών των βημάτων s , αναλύεται στο [11], στο κομμάτι της απόδειξης της ορθότητας (correctness) του πρωτοκόλλου. Στην υλοποίηση του πρωτοκόλλου, για να μετρηθεί αυτός ο αριθμός βημάτων, χρειάζεται ένας counter σε κάθε μήνυμα, ο οποίος πρέπει να είναι $\lceil \log 180\sqrt{2} \rceil$ bits.

2. **The Probabilistic Forwarding Phase** (Φάση πιθανοτικής διάδοσης): Στη διάρκεια αυτής της φάσης, κάθε particle P που έχει τη πληροφορία προς διάδοση, υπολογίζει μια γωνία ϕ καλώντας ένα υπό-πρωτόκολλο « ϕ -υπολογισμός» (η περιγραφή του ακολουθεί παρακάτω) και εκπέμπει την $info(E)$ σε όλους του γείτονες του με πιθανότητα P_{fwd} (ή διαφορετικά δεν διαδίδει τη πληροφορία με πιθανότητα $1 - P_{fwd}$) η οποία ορίζεται ως εξής:

$$P_{fwd} = \begin{cases} 1, & \text{εάν } \phi \geq \phi_{threshold} \\ \frac{\phi}{\pi}, & \text{διαφορετικά} \end{cases}$$

όπου ϕ είναι η γωνία που ορίζεται από τη γραμμή EP και τη γραμμή PS (σχήμα 17.2), και $\phi_{threshold} = 134^\circ$ (η τιμή της $\phi_{threshold}$ αναλύεται στο [11], στο κομμάτι της απόδειξης της ορθότητας (correctness)).



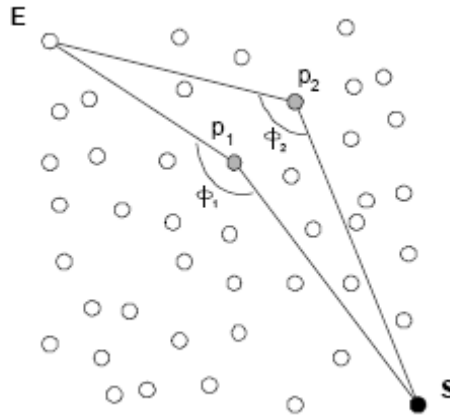
Σχήμα 17.2. : Παράδειγμα υπολογισμού γωνίας ϕ .

Στις δύο φάσεις, αν ένα particle έχει ήδη εκπέμψει την $info(E)$ και την λάβει ξανά, τότε τη παραβλέπει. Έτσι δεν δημιουργούνται πολλαπλά μονοπάτια και επίσης τα μηνύματα δεν μεγαλώνουν κατά την διάδοση τους στο δίκτυο. Επίσης το PFR μπορεί να εφαρμοστεί και στη περίπτωση πολλαπλών γεγονότων, υποθέτοντας ένα κατάλληλο MAC επίπεδο και ένα σχήμα κωδικοποίησης με βάση το οποίο θα μπορούν να διακρίνονται μοναδικά τα γεγονότα. Τέλος αν το ίδιο γεγονός ανιχνευτεί από περισσότερους του ενός κόμβους αισθητήρων, μπορεί να εφαρμοστεί κάποια τεχνική συσσώρευσης δεδομένων κατά τη διάρκεια της 1^η φάσης έτσι ώστε να αποφευχθούν οι επαναλήψεις μηνυμάτων.

Μία ακόμα λεπτομέρεια του πρωτοκόλλου είναι ότι αν $\phi = \pi$ τότε το P κείται στη γραμμή ES και έτσι πάντα μεταδίδει.

Εάν η πυκνότητα των particles είναι κατάλληλα μεγάλη, τότε για μια γραμμή ES υπάρχει με μεγάλη πιθανότητα μία σειρά σημείων (τα οποία είναι διαδοχικά και

το ένα βρίσκεται εντός το εύρος R του άλλου), τα οποία «περιβάλλουν» την ES , και των οποίων οι γωνίες φ είναι μεγαλύτερες από την $\phi_{threshold}$. Έτσι όλα τα σημεία εκπέμπουν και επιπλέον ακολουθούν τη γραμμή ES (σχήμα 17.3).



Σχήμα 17.3. : Γωνία φ κοντά στην βέλτιστη γραμμή.

17.2.1 Το υπό-πρωτόκολλο « φ -υπολογισμού»

Το υπό-πρωτόκολλο που θα περιγράψουμε σε αυτή την παράγραφο, αναφέρεται στο πως γεωμετρικά κάθε φορά υπολογίζεται η γωνία φ στο PFR. Έστω λοιπόν ότι P_{prev} είναι το particle που μετάδωσε την $info(E)$ στο P (σχήμα 17.2).

1. Όταν το P_{prev} εκπέμπει την $info(E)$, επισυνάπτει επίσης και τις πληροφορίες $|EP_{prev}|$ και την κατεύθυνση $\overrightarrow{P_{prev}E}$.
2. Το P εκτιμάει τη κατεύθυνση και το μήκος του τμήματος της γραμμής $P_{prev}P$ όπως αναφέρθηκε και στις ιδιότητες που έχει κάθε particle στην αρχή του κεφαλαίου.
3. Το P υπολογίζει στη συνέχεια τη γωνία $(E\hat{P}_{prev}P)$ καθώς επίσης και το $|EP|$ και τη κατεύθυνση $\overrightarrow{PE}$ (αυτή θα χρησιμοποιηθεί για τη περαιτέρω διάδοση από το P).
4. Το P υπολογίζει επίσης τη γωνία $(P_{prev}\hat{P}E)$ και αφαιρώντας την από την $(P_{prev}\hat{P}S)$ υπολογίζει τελικά τη φ .

Αξίζει να παρατηρήσουμε τα εξής:

- Η κατεύθυνση και η απόσταση από τους ενεργούς κόμβους αισθητήρων στο E επαναλαμβάνεται επαγωγικά (δηλαδή το P θα γίνει P_{prev} στην επόμενη φάση).
- Το PFR απαιτεί μηνύματα όπου το μήκος τους φράσσεται από το $\log A$, όπου A είναι κάποιο μέτρο του μήκους της περιοχής του δικτύου, και επίσης όπως αναφέραμε και παραπάνω δεν υπάρχει αύξηση του μεγέθους του μηνύματος κατά την μετάδοση της πληροφορίας.

Περισσότερα για το PFR και τις ιδιότητες του, μπορείτε να διαβάσετε στο [11].

18 ΠΡΩΤΟΚΟΛΛΟ MP - DFR

Το πρωτόκολλο MP-DFR (Multipath Diffusion Forwarding Protocol) προτάθηκε από τους Ι. Πευκιανάκη, Ι. Χατζηγιαννάκη και Σ. Νικολετσέα στο [35], και αναφέρεται στο πρόβλημα της «τοπικής ανίχνευσης και διάδοσης πληροφορίας». Το MP-DFR βασίζεται στο μοντέλο των πολλαπλών sinks που αναλύσαμε στη παράγραφο 3.3, και επιτυγχάνει όπως θα δούμε στη συνέχεια μια στιβαρή (robust) και ανεκτική στα λάθη (fault tolerant) διάδοση της πληροφορίας, διαχέοντας την πληροφορία στο δίκτυο, μέσω πολλαπλών μονοπατιών.

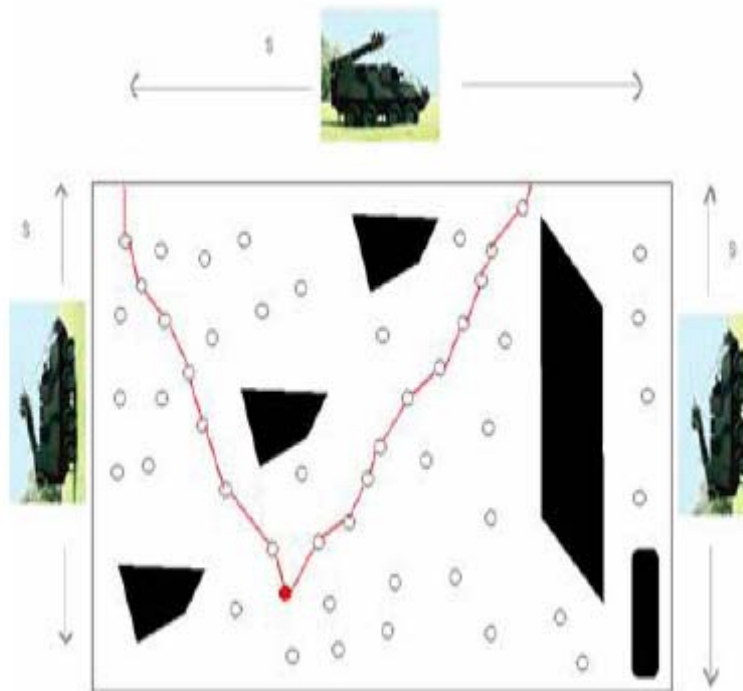
Στις επόμενες παραγράφους του κεφαλαίου θα αναλύσουμε το πρωτόκολλο, την ορθότητα του, την αποδοτικότητα του σε ενέργεια αλλά και σε τι εφαρμογές θα ήταν ιδιαίτερα χρήσιμο. Επίσης στο επόμενο μέρος της εργασίας μας, θα παραθέσουμε τα αποτελέσματα από την εξομοίωση του, και θα το συγκρίνουμε με τα πρωτόκολλα LTP και PFR που αναλύσαμε παραπάνω.

18.1 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΠΑΡΑΔΕΙΓΜΑΤΑ ΕΦΑΡΜΟΓΗΣ

Η διαθεσιμότητα των μικρό-αισθητήρων και η δυνατότητα για χαμηλής ισχύος ασύρματης επικοινωνίας, θα επιτρέψουν την δημιουργία πολύ πυκνά κατανομημένων WSNs που θα αναφέρονται σε περιβαλλοντολογικές εφαρμογές (ανίχνευση πυρκαγιάς), στρατιωτικές εφαρμογές (παρακολούθηση εχθρικών δυνάμεων), εφαρμογές ασφάλειας κ.α. Τα κύρια χαρακτηριστικά όλων αυτών των εφαρμογών είναι το αυξημένο επίπεδο *dynamics*, παραγόντων δηλαδή που δεν είναι σταθεροί και προβλέψιμοι. Όπως έχουμε αναφέρει και σε προηγούμενες παραγράφους, το γεγονός ότι υπάρχει ένας μεγάλος αριθμός κόμβων αισθητήρων που είναι περιορισμένης ενέργειας και επιρρεπής σε βλάβες ή αποτυχίες, εισάγει ένα αυξημένο επίπεδο από *dynamics* συστήματος (*system dynamics*). Επίσης το γεγονός ότι το πεδίο ενδιαφέροντος μπορεί να είναι ιδιαίτερα «αφιλόξενο», και ότι οι περιβαλλοντολογικές συνθήκες μπορεί να επηρεάζουν την συνολική λειτουργία του δικτύου, εισάγει ένα αυξημένο επίπεδο περιβαλλοντολογικών *dynamics* (*environmental dynamics*).

Για παράδειγμα ας θεωρήσουμε μια στρατιωτική εφαρμογή, όπου η παρακολούθηση των εχθρικών δυνάμεων, ανατίθεται σε ένα δίκτυο αισθητήρων, του

οποίου οι κόμβοι είναι κατανεμημένοι στο πεδίο μάχης. Ο σκοπός λοιπόν αυτού του WSN, είναι να ανιχνεύει στοιχεία τα οποία συνιστούν «απειλή», και να στέλνει σήματα συναγερμού σε οποιοδήποτε κινητό sink, στη περίμετρο του επιπέδου του δικτύου. Η διάδοση όμως μια τέτοιας πολύ σημαντικής πληροφορίας σε ένα τέτοιο δίκτυο, μπορεί να είναι εξαιρετικά δύσκολη, όχι μόνο εξαιτίας της αχρήστευσης κάποιων κόμβων του δικτύου (π.χ. από έλλειψη ενέργειας), αλλά επειδή ίσως κάποιο μεγάλο κομμάτι του δικτύου έχει πάψει να είναι λειτουργικό. Για παράδειγμα αν θεωρήσουμε ότι κάποια έκρηξη βόμβας αχρηστεύει ένα κομμάτι του δικτύου, ή κάποια φυσικά εμπόδια δεν επιτρέπουν LOS επικοινωνία ή κάποιες άλλες περιβαλλοντικές συνθήκες κάνουν αδύνατη τη μετάδοση δεδομένων προς κάποιες κατευθύνσεις, τότε είναι κατανοητό ότι στο δίκτυο μας υπάρχουν περιοχές με μη-ομοιόμορφης κατανομής κόμβων, η καλύτερα περιοχές λειτουργικές και λιγότερο λειτουργικές. Άρα λοιπόν ένα στιβαρό πρωτόκολλο για το πρόβλημα του “local detection and propagation”, θα πρέπει να είναι προσαρμοστικό στις διαφορετικές συνθήκες που δημιουργούνται στο δίκτυο, και να μπορεί να εκμεταλλευτεί το λειτουργικό κομμάτι του δικτύου, για να διαδώσει τη πληροφορία στο κέντρο ελέγχου. Για παράδειγμα το σχήμα 18.1 απεικονίζει ένα σενάριο όπου φαίνεται ένα πεδίο μάχης το οποίο καλύπτεται από κόμβους αισθητήρων, και επίσης έναν κόμβο που έχει ανιχνεύσει ένα σημαντικό γεγονός και το διαδίδει μέσω πολλαπλών μονοπατιών (δύο στη προκειμένη περίπτωση) στη περίμετρο του δικτύου όπου οι κινητοί sink μπορούν να παραλάβουν.



Σχήμα 18.1. : Στρατιωτική εφαρμογή. Τα μαύρα πολύγωνα είναι οι μη λειτουργικές περιοχές του δικτύου. Παρατηρείστε ότι υπάρχουν sinks σε τρεις πλευρές του παραλληλογράμμου.

Η μετάδοση της πληροφορίας μέσω πολλαπλών μονοπατιών διαφορετικών κατευθύνσεων (disjoint paths), μπορεί να εισάγει ένα πλεονασμό και συνεπώς μεγαλύτερη σπατάλη ενέργειας, όμως δίνει τη δυνατότητα μιας πιο στιβαρής διάδοσης της πληροφορίας, αφού αν ένα μονοπάτι «σκοντάψει σε μία μη-λειτουργική

περιοχή» κάποιο άλλο μπορεί να καταφέρει να αποφύγει ένα μη-λειτουργικό κομμάτι του δικτύου. Αν αυτή η λογική συνδυαστεί με το μοντέλο των πολλαπλών sink που έχουμε αναλύσει και το οποίο θεωρεί ότι υπάρχουν κινητοί sinks περιμετρικά του δικτύου που είναι σε θέση να συλλέγουν τις πληροφορίες από τις boundary particles, τότε μπορεί να μας δώσει ένα πολύ στιβαρό πρωτόκολλο.

Το MP-DFR που θα αναλυθεί παρακάτω, βασίζεται σε αυτές τις ιδέες και είναι κατάλληλο για τέτοιου τύπου εφαρμογές. Τέτοιες ιδέες βέβαια δεν αναφέρονται μόνο σε στρατιωτικές εφαρμογές, αλλά μπορούν αποδοτικά να χρησιμοποιηθούν σε περιπτώσεις όπου το δίκτυο δεν είναι αρκετά πυκνό, ή σε ένα δίκτυο που υπάρχουν πολλές αποτυχίες κόμβων (π.χ. εξαντλούν την ενέργεια τους) ή ακόμα για περιπτώσεις όπου το μέσο μετάδοσης είναι ιδιαίτερα προβληματικό (π.χ. εξαιτίας πολλών παρεμβολών).

18.2 SERVICE DIFFERENTIATION

Όπως έχουμε ήδη αναφέρει, το πρόβλημα της «τοπικής ανίχνευσης και διάδοσης πληροφορίας», αναφέρεται στην ανίχνευση ενός σημαντικού γεγονότος από το φυσικό περιβάλλον, και την αποδοτική τόσο ενεργειακά όσο και χρονικά διάδοση του, σε ένα κέντρο ελέγχου. Σε ένα τέτοιο πρόβλημα, θα μπορούσαμε επίσης να εισάγουμε την ιδέα τις «διαφοροποίησης υπηρεσιών» (ή *Service Differentiation*), σύμφωνα με την οποία σε κάθε πακέτο δεδομένων ανατίθεται και ένα επίπεδο προτεραιότητας (priority level) ανάλογα με τη πληροφορία που περιέχει και τη σημασία αυτής της πληροφορίας. Έτσι λοιπόν μπορούμε να εισάγουμε με αυτό τον τρόπο την ιδέα της ποιότητας υπηρεσιών (QoS) σε ένα WSN.

Η ιδέα της «διαφοροποίησης υπηρεσιών» και της ανάθεσης προτεραιότητας στα δεδομένα στα δίκτυα αισθητήρων, αναφέρθηκε στο [36]. Σύμφωνα λοιπόν με το [36], οι πληροφορίες, έχουν διαφορετικά επίπεδα κρισιμότητας, και ένα δίκτυο αισθητήρων θα πρέπει να είναι πρόθυμο να ξοδέψει περισσότερους ενεργειακούς πόρους, για να φτάσουν στο προορισμό τους τα πακέτα τα οποία έχουν μεγαλύτερο βαθμό κρισιμότητας. Για παράδειγμα, ένα WSN που ανιχνεύει πυρκαγιά σε ένα δάσος, θα πρέπει να καταναλώνει περισσότερους πόρους για να μεταφέρει ένα μήνυμα το οποίο θέλει να αναφέρει μια θερμοκρασία 100° C (σίγουρη πυρκαγιά), παρά ένα μήνυμα που θέλει να αναφέρει θερμοκρασία 50° C (υποψία πυρκαγιάς).

Άρα λοιπόν στη «διαφοροποίησης υπηρεσιών», κάθε πακέτο έχει ένα επίπεδο προτεραιότητας l , όπου $l \in l_1, l_2, \dots, l_n$. Ο αριθμός των επιπέδων προτεραιότητας n , μπορεί να εξαρτάται από την εφαρμογή και τα επίπεδα διαφοροποίησης της πληροφορίας μπορεί να εξαρτώνται από το διαχειριστή του δικτύου. Επίσης κάθε επίπεδο προτεραιότητας l_i σχετίζεται με μία πιθανότητα επιτυχούς μετάδοσης (reaching probability) P_i , έτσι ώστε κάθε πακέτο με επίπεδο προτεραιότητας l_i , αναμένεται να φτάσει στο κέντρο ελέγχου με πιθανότητα P_i .

Στο MP-DFR, για να μπορέσουμε να εγγυηθούμε ότι η παράδοση του πακέτου θα γίνει με πιθανότητα P_i , υιοθετούμε την ιδέα της εισαγωγής πλεονασμού στην αποστολή πακέτων. Με βάση αυτή τη προσέγγιση, για να μπορέσουμε να επιτύχουμε την απαραίτητη αξιοπιστία στη διάδοση της πληροφορίας, μεταδίδουμε πολλαπλά αντίγραφα του πακέτου, μέσω πολλαπλών μονοπατιών τα οποία έχουν διαφορετικές κατευθύνσεις, και διαχέονται σε όλο το δίκτυο. Έτσι ανάλογα με το P_i ,

η προσέγγιση αυτή επιβάλλει βέβαια ένα επιπλέον overhead στη διάδοση της πληροφορίας.

18.3 ΣΧΕΤΙΚΗ ΈΡΕΥΝΑ

Σε αυτή τη παράγραφο θα κάνουμε μια ανασκόπηση στη έρευνα που είναι σχετική με τις ιδέες που αναπτύσσονται στο MP-DFR. Δηλαδή θα μιλήσουμε και για άλλα πρωτόκολλα που αναφέρονται στο πρόβλημα της «τοπικής ανίχνευσης και διάδοσης πληροφορίας» (έχουμε ήδη αναλύσει τα LTP, PFR), σε πρωτόκολλα που βασίζονται σε multipath διάδοση της πληροφορίας και σε πρωτόκολλα τα οποία χρησιμοποιούν πολλαπλούς sinks. Επίσης θα προσπαθήσουμε να εντοπίσουμε που αυτά τα πρωτόκολλα συνδέονται και που διαφέρουν.

Όπως έχουμε προαναφέρει, υπάρχουν αρκετά smart dust πρωτόκολλα για το πρόβλημα της «τοπικής ανίχνευσης και διάδοσης πληροφορίας». Για παράδειγμα το LTP στο [13], χρησιμοποιεί μια φτηνή φάση αναζήτησης για να βρει το κατάλληλο γείτονα και να του παραδώσει τη πληροφορία. Το Sleep-Awake πρωτόκολλο (SWP) ([38]), θεωρεί search φάση, φάση οπισθοχώρησης και φάση μετάδοσης όπως το LTP, όμως υποθέτει επίσης ξεχωριστές περιόδους όπου ένα particle μπορεί να είναι σε φάση sleep (ύπνου) ή awake (ενεργό particle). Έτσι λοιπόν το SWP θεωρεί μία εναλλαγή μεταξύ των φάσεων sleep-awake, με απώτερο σκοπό την εξοικονόμηση ενέργειας. Τέλος στο [11] προτείνεται το PFR, το οποίο προσπαθεί να ελαχιστοποιήσει τη κατανάλωση ενέργειας ευνοώντας πιθανοτικά συγκεκριμένα μονοπάτια για τοπική διάδοση της πληροφορίας προς τον sink.

Μια διαφορετική προσέγγιση στη multihop μετάδοση δεδομένων, είναι η multihop μετάδοση μέσω πολλαπλών μονοπατιών όπως αυτή εξετάζεται στο [39]. Με βάση αυτή τη προσέγγιση, ορίζονται αλγόριθμοι εύρεσης θέσης, για τη κατασκευή των εναλλακτικών μονοπατιών. Ο πρώτος μηχανισμός που προτείνεται στο [39] κατασκευάζει ένα μικρό αριθμό εναλλακτικών μονοπατιών τα οποία είναι “*node-disjoint*” (δηλαδή τα μονοπάτια δεν έχουν κοινούς κόμβους) με το πρωτεύον μονοπάτι, ενώ ο δεύτερος μηχανισμός κατασκευάζει “*braided*” μονοπάτια τα οποία διαχωρίζονται μερικώς από το πρωτεύον μονοπάτι (μπορεί να έχουν λίγους κοινούς κόμβους). Όμως η κύρια ιδέα και στις δύο περιπτώσεις, είναι η εξασφάλιση αξιοπιστίας στη μετάδοση δεδομένων, με την εισαγωγή κάποιου πλεονασμού, ο οποίος έχει τη μορφή πολλαπλών αντιγράφων κάθε πακέτου, τα οποία και μεταδίδονται μέσω ενός αριθμού από disjoint μονοπάτια από τη πηγή στο sink. Όμως η προσέγγιση αυτή όπως εξετάζεται στο [39], απαιτεί ένα πολύ μεγάλο κόστος για τη διατήρηση των μονοπατιών.

Μια άλλη διαφορετική multipath προσέγγιση διάδοσης δεδομένων, είναι το πρωτόκολλο ReInForm ([40]), σύμφωνα με το οποίο τα δεδομένα μεταδίδονται σε επιθυμητά επίπεδα αξιοπιστίας, με επιπλέον κόστος φυσικά, ανεξάρτητα όμως από πιθανά σφάλματα τα οποία μπορεί να συμβαίνουν στο κανάλι. Η προσέγγιση αυτή χρησιμοποιεί την ιδέα της δυναμικής κατάστασης πακέτου (dynamic packet state) για να υπολογίσει τον απαραίτητο αριθμό πακέτων που απαιτούνται για να επιτευχθεί η επιθυμητή αξιοπιστία, χρησιμοποιώντας τοπική γνώση για το ρυθμό που συμβαίνουν σφάλματα στο κανάλι, χωρίς να απαιτεί εκ των προτέρων υπολογισμούς, ή μηχανισμούς διατήρησης των καναλιών.

Όλα τα προαναφερθέντα πρωτόκολλα (ακόμα και τα [39], [40] που υλοποιούν multipath διάδοση πληροφορίας) δεν λαμβάνουν υπόψη το γεγονός ότι το δίκτυο

μπορεί να μην είναι πυκνό ή ότι μπορεί να έχει κάποιες περιοχές, οι οποίες δεν είναι λειτουργικές (παράγραφος 18.1). Έτσι λοιπόν, η προώθηση των δεδομένων μέσω μίας ζώνης με particles προς κάποια κατεύθυνση, μπορεί να αποτύχει εξαιτίας της δυσλειτουργίας του δικτύου σε κάποιο σημείο προς αυτή τη κατεύθυνση. Το πρωτόκολλο MP-DFR χρησιμοποιώντας το μοντέλο των πολλαπλών sinks που αναφέραμε, «διαχέει» τη πληροφορία μέσω πολλαπλών μονοπατιών σε ολόκληρο το δίκτυο. Η διάδοση της πληροφορίας θεωρείται επιτυχημένη όταν ακόμα και ένας μόνο sink καταφέρει να τη συλλέξει.

Η ιδέα των πολλαπλών sinks, εισάγει ένα καινούργιο δικτυακό μοντέλο όπου υπάρχουν πολλοί σταθμοί βάσης που μπορούν να γίνουν παραλήπτες των πληροφοριών για όλους τους κόμβους του WSN. Κυρίως για θέμα στιβαρότητας (robustness), κλιμακωσιμότητας (scaling), και ισορροπίας του φόρτου του δικτύου (load balancing), είναι επιθυμητό να εισάγουμε πολλαπλούς sinks στο δίκτυο μας, και να προτρέπουμε του κόμβους να στέλνουν τα δεδομένα στο πιο κοντινό τους sink (με βάση κάποια μετρική όπως ο αριθμός των hops), παρά όλοι οι κόμβοι να στέλνουν σε ένα μοναδικό sink. Επίσης με τη χρήση πολλαπλών sink, δεν θα είναι απαραίτητο κάποιος από αυτούς να πλημμυρίζει το δίκτυο με ερωτήματα προς του κόμβους και έτσι να υπέρ-φορτώνει το δίκτυο. Οι ιδέες αυτές έχουν ήδη μελετηθεί στα [41], [42], και [43]. Για παράδειγμα το [41] προτείνει έναν αλγόριθμο (Voronoi Scoring) όπου διαχέονται μηνύματα από έναν sink μόνο σε ένα υποσύνολο κόμβων που κείται μέσα στο Voronoi cluster του sink, χωρίς να εμπλέκονται κόμβοι που είναι εκτός αυτού του cluster. Επίσης η ιδέα του [42] σχετίζεται με τη παραπάνω προσέγγιση, μιας και κάθε κόμβος αισθητήρων μπορεί να επιλέξει να προωθήσει τα δεδομένα στο πιο κοντινό του sink (με βάση κάποια μετρική) ώστε να μειώσει το κόστος επικοινωνίας. Τέλος στο [43] προτείνεται ένα αποδοτικό multisink σχήμα διαχείρισης θέσης του κάθε κόμβου, για παρακολούθηση αντικειμένων που κινούνται, όπου ο χρήστης μπορεί να ενημερωθεί για τις τοποθεσίες των αντικειμένων μέσω οποιουδήποτε sink του δικτύου.

Στο [44], ο σχεδιασμός του multisink δικτύου, θεωρείται ότι είναι ένα πρόβλημα clustering, όπου ο αριθμός των κόμβων sink είναι ισοδύναμος με τον αριθμό των cluster. Στο [44] λοιπόν, το πρόβλημα της αποδοτικής ομαδοποίησης (clustering) των κόμβων του δικτύου, δοθέντος ενός συγκεκριμένου αριθμού sinks, ονομάζεται “Best Sinks Locations” πρόβλημα (BSL). Όμως το multisink μοντέλο που εξετάζεται στα [41], [42], [43] και [44], και που ορίζει άμεσα ή έμμεσα clusters, δεν είναι το ίδιο με τη σχετική έρευνα στην υλοποίηση clustering για Ad Hoc και WSN δίκτυα ([45], [15]). Συγκεκριμένα το πρόβλημα clustering που εξετάζεται σε αυτά τα papers θεωρεί μια διαδικασία εκλογής αρχηγού cluster (clusterhead), ενώ αντίθετα στα [41], [42], [43] και [44], οι sinks αποφασίζονται εκ των προτέρων, εξαιτίας των ιδιαίτερων χαρακτηριστικών τους (μόνιμη τροφοδοσία ενέργειας, μεγάλη υπολογιστική ισχύς κ.α.).

Τέλος στο [29], παρουσιάζεται μια τελείως διαφορετική προσέγγιση από τις παραπάνω. Πιο συγκεκριμένα, μελετάτε μια προσέγγιση, υπάρχουν πολλοί κινητοί sinks (data mules) που μπορούν να κινούνται στο πεδίο ενδιαφέροντος και να συλλέγουν δεδομένα. Η ιδέα της χρησιμοποίησης πολλαπλών sinks σε αυτή τη περίπτωση, είναι η αποφυγή «μποτιλιάρισματος» (bottleneck) στο δίκτυο και η μη-ομοιόμορφη κατανάλωση πόρων των κόμβων του δικτύου.

Στις προσεγγίσεις που αναφέραμε παραπάνω, ο σχεδιασμός του δικτύου με τη χρήση πολλαπλών sinks, έχει ως σκοπό να διευκολύνει γενικότερα τη διαχείριση του δικτύου, και να εξοικονομηθεί ενέργεια αποφεύγοντας τις μεγάλες διαδρομές κατά τη διάδοση ενός μηνύματος. Επίσης οι πολλαπλοί sinks επιτυγχάνουν γρηγορότερες

απαντήσεις σε ερωτήματα που θέτουν στο δίκτυο, καθώς επίσης αποτρέπουν προβλήματα συμφόρησης που θα δημιουργούσε η ύπαρξη ενός μόνο sink. Όμως ένα τέτοιο μοντέλο μπορεί να είναι ωφέλιμο και με άλλους τρόπους. Ας υποθέσουμε ότι οι πολλαπλοί sinks είναι διασκορπισμένοι σε διαφορετικές περιοχές του πεδίου ενδιαφέροντος. Όπως έχουμε αναφέρει, αν ένας κόμβος ανιχνεύσει ένα σημαντικό γεγονός και προσπαθήσει να το προωθήσει σε ένα μοναδικό sink, μπορεί να αποτύχει εξαιτίας μη λειτουργικών περιοχών του δικτύου. Αν όμως έσπευτε τα δεδομένα προς διαφορετικές κατευθύνσεις προς τους πολλαπλούς sink, τότε θα απέφευγε πιθανόν τέτοιες καταστάσεις. Άρα λοιπόν ένα μοντέλο πολλαπλών sink, μπορεί να οδηγήσει στο σχεδιασμό ενός στιβαρού πρωτοκόλλου διάδοσης δεδομένων, το οποίο θα εκμεταλλεύεται τη δομή του δικτύου και θα στέλνει τη πληροφορία μέσω διαφορετικών μονοπατιών στους διαφορετικούς sinks. Σε αυτή την ιδέα βασίζεται και το πρωτόκολλο MP-DFR.

18.4 ΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ

Σε αυτή τη παράγραφο, θα προσπαθήσουμε να δώσουμε ένα πιο τυπικό ορισμό του προβλήματος «τοπικής ανίχνευσης και διάδοσης πληροφορίας», όσον αφορά τη προσέγγιση του από το MP-DFR. Δηλαδή θα δούμε πώς μοντελοποιείται το πρόβλημα, στη περίπτωση που έχουμε το μοντέλο πολλαπλών sinks.

Έστω λοιπόν ότι ένα particle P , ανιχνεύει ένα σημαντικό γεγονός E . Το πρόβλημα διάδοσης για το P είναι το εξής:

«Πώς μπορεί ένα particle P , σε συνεργασία με τους άλλους κόμβους του δικτύου να διαδώσει αποδοτικά τη πληροφορία $info(E)$ που αναφέρεται σε ένα γεγονός E , σε κάποιον (οποιονδήποτε) από τους walls W_i ;»

Για να μειώσουμε τη κατανάλωση ενέργειας στο δίκτυο, είναι επιθυμητό να αποφύγουμε να πλημμυρίσουμε το δίκτυο με μηνύματα (flooding) και να μειώσουμε τον αριθμό των hops που λαμβάνουν μέρος στη διαδικασία διάδοσης της πληροφορίας. Βέβαια ένα αποδοτικό πρωτόκολλο θα πρέπει να επιτυγχάνει να στείλει τη σημαντική πληροφορία στο κέντρο ελέγχου.

Ορισμός 4: Ας υποθέσουμε ότι E είναι η θέση του particle P που ανιχνεύει το σημαντικό γεγονός. Επίσης έστω W είναι το κομμάτι εκείνο της περιμέτρου του δικτύου που καλύπτεται από το κινητό sink S . Κάθε πρωτόκολλο Π που λύνει το πρόβλημα διάδοσης του P πρέπει να ικανοποιεί τα εξής:

- **Correctness** (Ορθότητα): Το Π πρέπει να μπορεί να εγγυηθεί ότι τα δεδομένα πάντα φτάνουν στο W (δηλαδή σε κάποιο σημείο της περιμέτρου του δικτύου), δοθέντος ότι ολόκληρο το δίκτυο υπάρχει και είναι λειτουργικό.
- **Robustness** (Στιβαρότητα): Το Π πρέπει να μπορεί να εγγυηθεί ότι τα δεδομένα φτάνουν στο W (δηλαδή σε κάποιο σημείο της περιμέτρου του δικτύου) με μεγάλη πιθανότητα (η οποία εξαρτάται από το πόσο σημαντικά είναι τα δεδομένα), ακόμα και σε περιπτώσεις όπου κάποια κομμάτια του

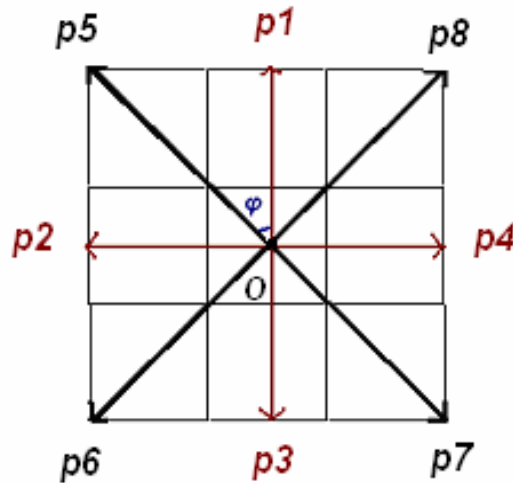
δικτύου δεν είναι λειτουργικά, αλλά βέβαια δεν έχουν απομονωθεί οι κινητοί sinks.

- **Efficiency** (Αποδοτικότητα): Αν το Π ενεργοποιεί k particles κατά τη διάρκεια της λειτουργίας του, τότε είναι επιθυμητό ο λόγος $r = \frac{k}{N}$ για το Π να είναι μικρός ώστε το πρωτόκολλο να είναι ενεργειακά αποδοτικό. Υπενθυμίζουμε ότι N είναι ο αριθμός των particles στο δίκτυο.

18.5 MULTIPATH DIFFUSION FORWARDING PROTOCOL (MP-DFR)

Όπως έχουμε αναφέρει, η βασική ιδέα του MP-DFR είναι η διάδοση της πληροφορίας μέσω πολλαπλών μονοπατιών κατά μήκος όλου του δικτύου, για την αποφυγή τοπικών μη-λειτουργικών περιοχών του, ή γενικότερα περιοχών όπου δεν υπάρχει πυκνή κατανομή κόμβων. Καθένα από αυτά τα μονοπάτια, σχετίζεται με μία διαφορετική κατεύθυνση προς κάποιο τείχος W . Σε ιδανική περίπτωση κάθε μονοπάτι είναι μια ευθεία γραμμή προς κάποια κατεύθυνση, και μια κατεύθυνση (μονοπάτι δηλαδή) διαφέρει από τη γειτονική της κατά μια γωνία φ . Ένα παράδειγμα δικτύου που θεωρεί ότι υπάρχουν κινητοί sinks, κατά μήκος όλης της περιμέτρου, φαίνεται στο σχήμα 18.2. Να σημειώσουμε σε αυτό το σημείο ότι όλα τα μονοπάτια που δημιουργούνται από το πρωτόκολλο, μπορούν να θεωρηθούν ότι δεν συναντιούνται (disjoint) εξαιτίας του τρόπου που δημιουργούνται. Ο τρόπος αυτός περιγράφεται στη 2^η φάση του πρωτοκόλλου και θα παρουσιαστεί παρακάτω. Να σημειώσουμε επίσης ότι τα μονοπάτια που δημιουργούνται, με μεγάλη πιθανότητα δεν είναι ιδανικά, με την έννοια ότι δεν ακολουθούν την ευθεία γραμμή προς το Wall όπως φαίνεται στο σχήμα 18.2. Αυτό συμβαίνει επειδή μπορεί να μην υπάρχουν κόμβοι πάνω στη ευθεία γραμμή που ενώνει την πηγή με το σημείο του Wall που ορίζεται από τη κατεύθυνση του μονοπατιού, ή ακόμα και να υπάρχουν, μπορεί να είναι ανενεργοί ή σε κατάσταση «ύπνου» (sleep mode).

Ο αριθμός των μονοπατιών που δημιουργεί το MP-DFR μεταβάλλεται δυναμικά ανάλογα με: **a)** με το πόσο σημαντικά είναι τα γεγονότα που ανιχνεύονται (service differentiation), **b)** με κάποια χαρακτηριστικά του δικτύου (πυκνότητα, κατανομή κόμβων), **c)** με τη κάθετη απόσταση ενός particle P (που ανίχνευσε το γεγονός) από κάποιο Wall W ($d(P, W)$), **d)** με βάση πληροφορίες από γειτονικούς κόμβους. Έτσι λοιπόν το MP-DFR διαχέει (diffuse) τη πληροφορία στο δίκτυο με ένα προσαρμοστικό τρόπο (adaptive), για να επιτύχει τη διάδοση της πληροφορίας, ανεξαρτήτως των συνθηκών και των χαρακτηριστικών του δικτύου.



Σχήμα 18.2. : Παράδειγμα απεικόνισης των πιθανών κατευθύνσεων των πολλαπλών μονοπατιών.

Όπως αναφέραμε στο μοντέλο των πολλαπλών Walls (παράγραφος 3.3.4) σύμφωνα με τα χαρακτηριστικά και τις δυνατότητες που έχει κάθε κόμβος, μπορεί να γνωρίζει τις κατευθύνσεις $direct_i$ για τους υπάρχοντες διαφορετικούς Walls W_i . Υπενθυμίσουμε ότι οι W_i αναφέρονται στις διαφορετικές πλευρές του τετραγώνου και αντιπροσωπεύουν τα διαφορετικά «τείχη» του δικτύου. Αν θεωρήσουμε λοιπόν ότι υπάρχουν Walls σε όλη τη περίμετρο του τετραγωνικού επιπέδου του δικτύου, υπάρχουν τέσσερις βασικές κατευθύνσεις $direct_1, direct_2, direct_3, direct_4$ οι οποίες αντιστοιχούν στα τέσσερα βασικά μονοπάτια $p1, p2, p3, p4$ (σχήμα 18.2). Ένα καινούργιο μονοπάτι μπορεί να οριστεί με τον υπολογισμό μιας νέας κατεύθυνσης, όπως για παράδειγμα ισχύει $direct_5 = direct_1 - \phi$ (σχήμα 18.2). Το πρωτόκολλο MP-DFR ταξινομεί τα μονοπάτια σε επίπεδα προτεραιότητας (priority levels) $plevel_i$. Στο πρώτο επίπεδο προτεραιότητας $plevel_1$, ανήκουν τα κύρια μονοπάτια $p1, p2, p3, p4$ που ορίζουν τις κατευθύνσεις $direct_1, direct_2, direct_3, direct_4$ (σχήμα 18.2). Τα μονοπάτια του $plevel_2$, προκύπτουν από τη διχοτόμηση των γωνιών που σχηματίζονται από τις «κατευθύνσεις» των μονοπατιών του $plevel_1$. Για παράδειγμα στο σχήμα 18.2 τα $p5, p6, p7, p8$ είναι μονοπάτια του $plevel_2$. Γενικότερα λοιπόν, τα μονοπάτια του $plevel_i$, προκύπτουν από τη διχοτόμηση των γωνιών σχηματίζονται από τις «κατευθύνσεις» των μονοπατιών των $plevel_1, \dots, plevel_{i-2}, plevel_{i-1}$. Ένα επίπεδο προτεραιότητας μονοπατιών $plevel_i$, ορίζει τη προτεραιότητα με την οποία θα κατασκευαστούν τα μονοπάτια (π.χ. το πρωτόκολλο δημιουργεί πρώτα τα μονοπάτια του $plevel_i$ και στην συνέχεια αν απαιτούνται περισσότερα, δημιουργεί τα μονοπάτια του $plevel_{i+1}$). Η ιδέα πίσω από τα επίπεδα των μονοπατιών, είναι η προσπάθεια για «διάχυση» της πληροφορίας κατά μήκος όλου του δικτύου.

Για το πρωτόκολλο MP-DFR, υποθέσαμε επίσης ότι όταν η $info(E)$ διαδοθεί με επιτυχία σε κάποιο Wall, ο κινητός sink που θα λάβει τη πληροφορία, θα ενημερώσει το particle-πηγή ότι η πληροφορία παραλήφθηκε με επιτυχία, είτε με μία υψηλής ενέργειας εκπομπή, είτε με επιλεκτικό flooding success μηνυμάτων. Για να

εξοικονομηθεί γενικά ενέργεια στο δίκτυο, ο sink μπορεί να στέλνει αυτά τα μηνύματα περιοδικά. Όπως έχουμε ήδη αναφέρει όλη η περιγραφή του πρωτοκόλλου που ακολουθεί βασίζεται στο μοντέλο των πολλαπλών sinks.

Ορισμός 5: Η πιθανότητα αποτυχίας διάδοσης της πληροφορίας μέσω ενός μονοπατιού, ή καλύτερα *Path Forwarding Failure Probability* ($P_{pfailure}^k$), είναι η πιθανότητα ένα particle που είναι υπεύθυνο να διαδώσει παραπέρα την σημαντική πληροφορία $info(E)$, να μην βρει κάποιο γειτονικό κόμβο (δηλαδή κάποιο κόμβο εντός της εμβέλειας του, που να είναι υποψήφιος παραλήπτης της πληροφορίας), σε κάποιο βήμα της διαδικασίας διάδοσης της πληροφορίας κατά μήκος ενός μονοπατιού k . Εάν $P_{psuccess}^k$ είναι η πιθανότητα επιτυχούς διάδοσης της πληροφορίας κατά μήκος ενός μονοπατιού k , τότε ισχύει $P_{pfailure}^k = 1 - P_{psuccess}^k$.

Ορισμός 6: Αξιοπιστία από άκρη σε άκρη ή *End-to-end reliability* ($P_{reliability}$), είναι η πιθανότητα να έχουμε τουλάχιστον μια επιτυχή μετάδοση μέσω ενός μονοπατιού k (οποιοδήποτε από αυτά που έχουμε κατασκευάσει) προς κάποιο, οποιοδήποτε Wall W . Η $P_{reliability}$ εκφράζει ουσιαστικά τη πιθανότητα επιτυχούς μετάδοσης.

Ορισμός 7: Δοθέντος ότι K είναι ένα σύνολο από disjoint μονοπάτια που κατασκευάζονται από το πρωτόκολλο, τότε η *End-to-end reliability* δίνεται από το τύπο:

$$P_{reliability} = 1 - \prod_{k \in K} P_{pfailure}^k = 1 - \prod_{k \in K} (1 - P_{psuccess}^k)$$

Το πρωτόκολλο MP-DFR έχει δύο κύριες φάσεις:

Φάση 1^η: Υπολογισμός των Πολλαπλών Μονοπατιών. Όταν ένα particle P ανιχνεύσει ένα σημαντικό γεγονός E , εκτιμάει το επίπεδο προτεραιότητας του l_i και υπολογίζει μία πιθανότητα επιτυχούς μετάδοσης $P_i = P_{reliability}$ (σύμφωνα με το service differentiation που αναφέραμε στη παράγραφο 18.2). Στη συνέχεια το πρωτόκολλο διεξάγει μια φάση «ανακάλυψης» (search phase) χρησιμοποιώντας χαμηλής ενέργειας μηνύματα, για να «ρωτήσει» τα γειτονικά particles για τα στατιστικά αποτυχημένης μετάδοσης της πληροφορίας (*propagation failure statistics*) προς κάθε κατεύθυνση που είναι διαθέσιμη.

Ορισμός 8: Στατιστικά αποτυχημένης μετάδοσης (*propagation failure statistics*) A_i^n από έναν γείτονα n προς μια κατεύθυνση i , ορίζεται ως $A_i^n = \frac{failp_i^n}{totalp_i^n}$, όπου $totalp_i^n$ είναι οι συνολικές προσπάθειες για μετάδοση της πληροφορίας προς μια κατεύθυνση i , και $failp_i^n$ είναι οι αποτυχημένες προσπάθειες για μετάδοση της πληροφορίας από τον n προς μια κατεύθυνση i μέχρι εκείνη τη στιγμή.

Αν δεν υπάρχει διαθέσιμο κάποιο A_i^n από ένα γείτονα, τότε αυτός επιστρέφει $A_i^n = 0$. Τελικά, μετά που θα παραληφθούν A_i^n για $\forall n \in P_N$, το particle P υπολογίζει $A_i = \frac{1}{|P_N|} \sum_{n \in P_N} A_i^n$ για $\forall i \in D$ όπου P_N είναι το σύνολο των γειτονικών κόμβων του P , και D είναι το σύνολο των διαθέσιμων κατευθύνσεων. Αν και η κατάσταση του δικτύου μεταβάλλεται γρήγορα, η μετρική A_i περιγράφει τη σταθερότητα ενός μονοπατιού διάδοσης της πληροφορίας προς μια κατεύθυνση i , και χρησιμοποιείται σαν βάρος για την $P_{pfailure}^k$.

Όταν η A_i για $\forall i \in D$ έχει υπολογιστεί, το particle P αρχίζει τη διαδικασία του υπολογισμού των μονοπατιών. Αρχικά το πρωτόκολλο επιλέγει μονοπάτια που βρίσκονται στο πρώτο επίπεδο προτεραιότητας $plevel_1$. Υπάρχουν τρεις διαφορετικές παραλλαγές για τον τρόπο, τη σειρά δηλαδή που θα επιλεγθούν τα μονοπάτια.

1. **Ελάχιστη Απόσταση.** Θεωρώντας ότι $d(P, W(d_i))$ είναι η κάθετη απόσταση ενός particle P , και του Wall W προς μια κατεύθυνση i , αυτή η προσέγγιση επιλέγει το μονοπάτι του $plevel_i$ που είναι πιο κοντά στο wall ($\min_{\forall i \in D} \{d(P, W(d_i))\}$). Να σημειώσουμε ότι οι κάθετες αποστάσεις προς τις κατευθύνσεις $direct_1, direct_2, direct_3, direct_4$ είναι γνωστές αν θεωρήσουμε τα χαρακτηριστικά των particles που αναφέραμε στη παράγραφο 3.3.4. Έτσι μπορούμε εύκολα να υπολογίσουμε την $d(P, W(d_i))$ για κάθε κατεύθυνση $direct_i$ (για παράδειγμα στο σχήμα 18.2 $\cos \phi = \frac{Op1}{Op5}$ όπου ϕ και $Op1$ είναι γνωστές).
2. **Βάρος A_i .** Σύμφωνα με αυτή τη προσέγγιση το πρωτόκολλο επιλέγει το μονοπάτι προς τη κατεύθυνση i με το μικρότερο A_i ($\min_{\forall i \in D} \{A_i\}$). Αυτό σημαίνει ότι το πρωτόκολλο επιλέγει το πιο σταθερό μονοπάτι.
3. **Τυχαία.** Σύμφωνα με αυτή τη προσέγγιση, τα μονοπάτια του $plevel_i$ επιλέγονται με την ίδια πιθανότητα. Η ιδέα πίσω από αυτή τη προσέγγιση είναι η μεταβλητή φύση του δικτύου. Τα χαρακτηριστικά του δικτύου μπορεί αν αλλάξουν τόσο γρήγορα, έτσι ώστε να μην έχει νόημα να ψάξουμε τη καλύτερη κατεύθυνση και το καλύτερο μονοπάτι.

Χρησιμοποιώντας κάποια από τις προσεγγίσεις που αναφέραμε, το πρωτόκολλο επιλέγει από το $plevel_1$ το κατάλληλο μονοπάτι k και υπολογίζει την *path forwarding failure probability* $P_{pfailure}^k$ για το k , σύμφωνα με τον τύπο:

$$P_{pfailure}^k = 1 - P_{psuccess}^k =$$

$$= 1 - \left(\frac{3}{q} \right)^{\left\lceil \frac{d(P, W(d_k))}{R} \right\rceil} \frac{N^N e^{-N} \sqrt{2\pi N}}{\left(N - \left\lceil \frac{d(P, W(d_k))}{R} \right\rceil \right) \left(N - \left\lceil \frac{d(P, W(d_k))}{R} \right\rceil \right) e^{\left\lceil \frac{d(P, W(d_k))}{R} \right\rceil} \sqrt{2\pi \left(N - \left\lceil \frac{d(P, W(d_k))}{R} \right\rceil \right)}}$$

Να σημειώσουμε ότι N είναι ο αριθμός των particles στο δίκτυο, και $d(P, W(d_k))$ είναι η κάθετη απόσταση ανάμεσα στο particle P και στο τείχος W προς τη κατεύθυνση του μονοπατιού k (να σημειώσουμε ότι στη 2^η προσέγγιση (ελάχιστης απόστασης) αναφέραμε ότι η $d(P, W(d_k))$ για κάθε μονοπάτι k μπορεί εύκολα να υπολογιστεί).

Δοθέντος μιας πιθανότητας επιτυχούς μετάδοσης $P_i = P_{reliability}$, το πρωτόκολλο υπολογίζει αν $P_{reliability} \leq 1 - \prod_{k \in K} A_k \cdot P_{pfailure}^k$ όπου $K = \{k\}$ (το σύνολο των μονοπατιών για τη διάδοση της πληροφορίας) και A_k είναι τα στατιστικά αποτυχημένης μετάδοσης για ένα μονοπάτι k . Αν δεν ισχύει η ανισότητα, τότε ένα άλλο μονοπάτι $k_1 \in plevel_i$ υπολογίζεται με την ίδια προσέγγιση, και η ανισότητα $P_{reliability} \leq 1 - \prod_{k \in K} A_k \cdot P_{pfailure}^k$ ξανά-υπολογίζεται όπου όμως $K = \{k, k_1\}$. Αυτή διαδικασία συνεχίζεται μέχρι να ισχύσει η ανισότητα $P_{reliability} \leq 1 - \prod_{k \in K} A_k \cdot P_{pfailure}^k$ για κάποιο $K = \{k, k_1, \dots, k_i\}$. Να σημειώσουμε επιπλέον, ότι όταν δεν υπάρχουν άλλα μονοπάτια στο $plevel_i$, τότε τα επόμενα υπονήφια μονοπάτια υπολογίζονται από το $plevel_{i+1}$ με την ίδια προσέγγιση.

Όταν η πρώτη φάση του πρωτοκόλλου έχει ολοκληρωθεί, το particle P έχει υπολογίσει ένα σύνολο μονοπατιών $K = \{k, k_1, \dots, k_i\}$ που εξασφαλίζουν την πιθανότητα επιτυχούς μετάδοσης που έχει ζητηθεί. Στη διάρκεια της επόμενης φάσης η πληροφορία $info(E)$ θα διαδοθεί στις κατευθύνσεις $K = \{k, k_1, \dots, k_i\}$.

Για μια καλύτερη κατανόηση της 1^{ης} φάσης του MP-DFR, στο πίνακα 18.1 παρουσιάζουμε με ψευδοκώδικα τη διαδικασία υπολογισμού των πολλαπλών μονοπατιών.

Φάση 2^η: Διάδοση της Πληροφορίας. Όταν ο αριθμός των μονοπατιών και οι κατευθύνσεις τους έχουν υπολογιστεί, η φάση 2 έρχεται να υπαγορεύσει πώς θα γίνει η μετάδοση της πληροφορίας κατά μήκος ενός μονοπατιού. Αυτή η φάση είναι η ίδια για κάθε μονοπάτι k και βασίζεται στην LTP διάδοση της πληροφορίας που αναλύσαμε στο κεφάλαιο 16. Ας θεωρήσουμε λοιπόν τους διαφορετικούς walls W_i του μοντέλου των πολλαπλών sinks, οι οποίοι ορίζουν και τις κύριες κατευθύνσεις $direct_i$. Ας υποθέσουμε επίσης ότι $d(P, W_i)$ είναι η κάθετη απόσταση ανάμεσα στο particle P και στο W_i (π.χ. στο σχήμα 18.2 $d(P, W_1) = |Op1|$). Κατά τη διαδικασία διάδοσης της πληροφορίας, όλα τα particles P_i κατά μήκος ενός μονοπατιού k θα θεωρήσουν την απόσταση τους από το wall W ως $d(P_i, W_k)$ όπου W_k είναι μία από τις κύριες κατευθύνσεις W_i που είναι πιο κοντά στη πραγματική κατεύθυνση του

μονοπατιού k . Στο σχήμα 18.2 η $d(P_i, W_5)$ ($d(P_i, W_5) = |Op5|$) για $\forall P_i \in p5$ μπορεί να είναι είτε $d(P_i, W_1)$ ή $d(P_i, W_2)$.

Στη φάση 2 ένα particle P_{i+1} κατά μήκος ενός μονοπατιού k που έχει λάβει τη πληροφορία $info(E)$ από ένα particle P_i εκτελεί ένα βήμα διάδοσης της πληροφορίας που αποτελείται από δύο υπό-φάσεις.

1. **Φάσης Αναζήτησης (search phase).** Αυτή η φάση χρησιμοποιεί χαμηλής ενέργειας μηνύματα (τα οποία περιέχουν τη κατεύθυνση του μονοπατιού k) για να ανακαλύψει ένα particle που είναι πιο κοντά στο W προς την κατεύθυνση του μονοπατιού k . Το σύνολο λοιπόν C των υποψήφιων προς μετάδοση γειτόνων, αποτελείται από τα γειτονικά particles P_{i+2} όπου $d(P_{i+2}, W_k) < d(P_{i+1}, W_k)$. Να σημειώσουμε εδώ ότι για να ευνοήσει τη μετάδοση της πληροφορίας προς μια συγκεκριμένη κατεύθυνση, το MP-DFR θέτει ένα επιπλέον περιορισμό για τους υποψήφιους γείτονες μιας, και αυτοί θα πρέπει να είναι μέσα σε ένα συγκεκριμένο εύρος μετάδοσης όπως φαίνεται στο σχήμα 16.2. Ο καλύτερος υποψήφιος γείτονας μπορεί να είναι **α)** είτε αυτός που απέχει λιγότερο από το W_k ($\min\{d(P_{i+2}, W_k)\}$) **β)** είτε αυτός που $\forall P_{i+2} \in C$

βρίσκεται πάνω στην ευθεία γραμμή προς την κατεύθυνση k (π.χ. στο σχήμα 18.4 ο καλύτερος υποψήφιος γείτονας είναι ο P'). Η παραλλαγή (β) ονομάζεται MP-DFRa. Έτσι εκτός από την απόσταση, το μήνυμα απάντησης του γειτονικού particle, περιέχει και τη γωνία που σχηματίζει αυτή με την ευθεία γραμμή προς την κατεύθυνση που ορίζει το k (π.χ. στο σχήμα 18.4 το particle P' επιστρέφει και την $d(P', W_k)$ και το ω).

2. **Φάση Μετάδοσης (direct transmission phase).** Αν το $C \neq \emptyset$ τότε το P_{i+1} στέλνει την $info(E)$ στο P_{i+2} με άμεση μετάδοση (π.χ. με laser). Όμως αν $C = \emptyset$ τότε η μετάδοση κατά μήκος του μονοπατιού k τερματίζεται.

Συμπερασματικά η φάση διάδοση της πληροφορίας κατά μήκος ενός μονοπατιού k τερματίζεται είτε όταν $C = \emptyset$ είτε όταν σε κάποιο βήμα του πρωτοκόλλου, η πληροφορία φτάνει σε ένα boundary particle.

Αφού το MP-DFR ευνοεί τη διάδοση της πληροφορίας κατά μήκος ενός μονοπατιού k το οποίο είναι όσο το δυνατό πιο κοντά στην ευθεία γραμμή που ορίζει η κατεύθυνση k , μπορούμε τετριμμένα να αποδείξουμε ότι στην ιδανική περίπτωση όπου θα υπάρχουν πάντα υποψήφιοι κόμβοι πάνω σε αυτή την ευθεία γραμμή, τα μονοπάτια που θα δημιουργούνται θα είναι disjoint. Πρακτικά βέβαια, το WSN μπορεί να μην είναι τόσο πυκνό ώστε να εγγυηθεί ότι τα μονοπάτια που δημιουργούνται δεν τέμνονται. Όμως όπως φαίνεται και στα αποτελέσματα εξομοίωσης του πρωτοκόλλου, αυτή η λεπτομέρεια δεν έχει ουσιαστική σημασία, και δεν επηρεάζει την αποδοτικότητα του πρωτοκόλλου.

Πίνακας 18.1. Συνάρτηση Αλγόριθμος 1^{ης} φάσης.

Algorithm Calculation of multiple paths

Input: 1) Number of particles N ,
2) Array of vertical distances for the main directions
 $d(P, W) = (d(P, W(d_1)), d(P, W(d_2)), \dots, d(P, W(d_i)))$
3) The number of squares areas q of our network plane.

Output: Array of path directions ($path_vector$).

BEGIN

$P_{failure} = 1$; /* Initializations */

$path_vector = \emptyset$;

$path_level = plevel_1$;

$P_{reliability} = detects(E)$; /* Detection of crucial event E */

$send_toall(beacon_msg)$; /* search phase */

$receive_fromall(buffer)$; /* $buffer[Num_directions][Num_neighbors]$ */

for ($i=0$; $i < |D|$; $i++$) { /* D = set of directions, P_N = set of neighbors */

 for ($j=0$; $j < |P_N|$; $j++$)

$A[i] = A[i] + buffer[i][j]$;

$A[i] = A[i] / |P_N|$ /* propagation failure statistics */

}

do {

$path = shortest_dist(d(P, W), path_level)$; /* Approach of the shortest distance */

$add(path, path_vector)$; /* Add path to the array of path directions */

 /* $P_{pfailure} =$ path forwarding failure probability */

$P_{pfailure} = failure_formula(N, d(P, W(d_{path})), q)$;

$P_{failure} = P_{failure} \cdot P_{pfailure} \cdot A[path]$;

$path_level = remove_from(path_level, path)$; /* remove $path$ from $path_level$ set */

 if($path_level == \emptyset$)

$path_level = next_level(path_level)$; /* from $plevel_i$ to $plevel_{i+1}$ */

}while($P_{reliability} > 1 - P_{failure}$)

return $path_vector$;

END

18.5.1 Πιθανότητα αποτυχίας διάδοσης της πληροφορίας

Σε αυτή τη παράγραφο θα αποδείξουμε το τύπο της πιθανότητας αποτυχίας διάδοσης της πληροφορίας μέσω ενός μονοπατιού (*path forwarding failure probability* $P_{pfailure}^k$), τον οποίο χρησιμοποιήσαμε στον υπολογισμό των πολλαπλών μονοπατιών.

Όπως αναφέραμε και παραπάνω, η $P_{pfailure}^k$ είναι η πιθανότητα κατά τη διάρκεια της μετάδοσης της πληροφορίας κατά μήκος ενός μονοπατιού προς μια κατεύθυνση k , να μην βρεθεί κάποιος υποψήφιος γείτονας (με βάση τα χαρακτηριστικά που αναφέραμε παραπάνω) ώστε να συνεχιστεί η multihop μετάδοση προς κάποιο sink, και έτσι η διάδοση κατά μήκος του k να τερματιστεί. Στην παρακάτω ανάλυση θεωρούμε $P_{pfailure}$ αντί για $P_{pfailure}^k$ αφού ο τύπος για την *path forwarding failure probability* είναι ο ίδιος για όλα τα μονοπάτια.

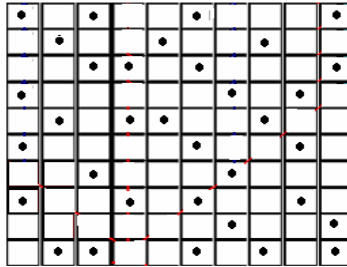
Η πιθανότητα αποτυχίας διάδοσης της πληροφορίας για να «αντανακλά» τις συνθήκες του δικτύου λαμβάνει υπόψη τις εξής παραμέτρους:

1. *Πυκνότητα του δικτύου*. Λαμβάνοντας υπόψη την πυκνότητα του δικτύου το πρωτόκολλο μπορεί να προσαρμοστεί στις καταστροφές και αποτυχίες των particles, που είναι πολύ σημαντικές για την επιτυχή διάδοση της πληροφορίας.
2. *Κατανομή των particles στο επίπεδο*. Η συνάρτηση κατανομής, αντικατοπτρίζει τη γεωγραφική κατανομή των κόμβων.
3. *Κάθετη απόσταση μεταξύ πηγής και τείχους*. Όσο πιο κοντά είναι ένα particle σε κάποιο Wall, τόσο αυξάνει η πιθανότητα για επιτυχή μετάδοση της πληροφορίας.

Απόδειξη *path forwarding failure probability*: Για την απόδειξη της $P_{pfailure}$ θα αναφερθούμε βέβαια στο μοντέλο των πολλαπλών sinks και θα εξετάσουμε τη πιθανότητα να αποτύχει η διάδοση της πληροφορίας προς ένα συγκεκριμένο wall W . Επίσης για να κάνουμε μία γεωμετρική ανάλυση θα θεωρήσουμε ότι το δισδιάστατο επίπεδο του σχήματος 3.10 αποτελείται από q τετράγωνα με πλευρά l . Έτσι η συνολική περιοχή που καλύπτει το δίκτυο ql^2 . Θεωρώντας επίσης ότι κάθε particle έχει εύρος μετάδοσης R , υποθέτουμε ότι τα particles βρίσκονται σε απόσταση l (οριζόντια και κάθετα) μεταξύ τους. Να σημειώσουμε ότι το l εξαρτάται από το R και χαρακτηρίζει τη πυκνότητα του δικτύου. Συγκεκριμένα επιλέγουμε $R = l\sqrt{2}$. Στην ανάλυση του πρωτοκόλλου θεωρούμε ότι ο αριθμός κόμβων που είναι κατανομημένοι στο δίκτυο είναι N , με $N < q$. Συμπερασματικά, κάποια από τα q τετράγωνα δεν θα περιέχουν κάποιο particle. Τέλος θεωρούμε ότι η κατανομή των κόμβων είναι ομοιόμορφη. Στο σχήμα 18.3 υπάρχει ένα παράδειγμα του δικτύου σύμφωνα με τη παραπάνω περιγραφή.

Σύμφωνα με τη διαδικασία διάδοσης της πληροφορίας όπως περιγράφηκε παραπάνω, το particle που είναι υπεύθυνο να συνεχίσει τη διάδοση της πληροφορίας, ανακαλύπτει ένα γείτονα μέσα στο εύρος μετάδοσης του, που είναι πιο κοντά στο wall από ότι αυτό, και του στέλνει την *info(E)*. Σε κάθε βήμα διάδοσης της πληροφορίας, υπάρχουν το πολύ τρεις υποψήφιοι γείτονες για να λάβουν την *info(E)* με βάση το μοντέλο που περιγράψαμε παραπάνω. Στο σχήμα 18.4 φαίνεται αυτή η ιδέα, όπου ένα particle P έχει μέσα στο εύρος μετάδοσης του τα particles P' , P'' και

P''' . Τέλος να αναφέρουμε ότι αναλύουμε τη διαδικασία διάδοσης τη πληροφορίας προς ένα wall W . Η ίδια ανάλυση ισχύει για κάθε κατεύθυνση του δικτύου.



Σχήμα 18.3. : Παράδειγμα μοντέλου δικτύου.

Θεωρώντας ομοιόμορφη κατανομή των particles στο δίκτυο, η πιθανότητα ένα συγκεκριμένο τετράγωνο να περιέχει ένα particle είναι:

$$\underbrace{\frac{1}{q} + \frac{1}{q} \dots + \frac{1}{q}}_N = \frac{N}{q}$$

Επίσης θεωρώντας ότι σε κάθε βήμα διάδοσης της πληροφορίας ένα particle P έχει τρεις υποψήφιους γείτονες για να συνεχίσουν τη διάδοση της πληροφορίας, μπορούμε να συμπεράνουμε ότι $P(success_i) = \frac{3N}{q}$.

Ορισμός 9: $P(success_i)$ ορίζεται ως η πιθανότητα επιτυχούς μετάδοσης στο βήμα i . Πιο συγκεκριμένα, $success_i$ είναι το γεγονός να υπάρχει ένα particle σε ένα τουλάχιστον από τα «υποψήφια» τετράγωνα στο i_{th} βήμα διάδοσης.

Δοθέντος ότι η διαδικασία διάδοσης της πληροφορίας ήταν σωστή στο βήμα 1, μπορούμε να θεωρήσουμε ότι ισχύει $P(success_2) = \frac{3(N-1)}{q}$. Ομοίως η πιθανότητα επιτυχούς μετάδοσης στο βήμα i είναι $P(success_i) = \frac{3(N-i+1)}{q}$.

Ας υποθέσουμε τώρα ότι $h_{opt}(P, W)$ είναι ο βέλτιστος αριθμός των hops που απαιτούνται για να φτάσουμε το wall W , στην ιδανική περίπτωση όπου πάντα υπάρχουν particles όπου ανά ζευγάρια απέχουν R στην ευθεία γραμμή από το P στο W . Επίσης ας υποθέσουμε ότι $d(P, W)$ είναι η απόσταση από P στο W . Τότε λαμβάνοντας υπόψη ότι κάθε particle έχει μέγιστο εύρος μετάδοσης R , ισχύει:

$$h_{opt} = \left\lceil \frac{d(P, W)}{R} \right\rceil$$

Με βάση τη παραπάνω ανάλυση μπορούμε να θεωρήσουμε ότι ο πραγματικός αριθμός των hops που απαιτούνται για να φτάσει ο P τον W είναι $h = \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil$ όπου $\varepsilon > 1$ μια σταθερά. Η σταθερά ε είναι η αποδοτικότητα σε hops για κάποιο πρωτόκολλο Π , και δίνεται από το λόγο $C_h = \frac{h(\pi, p, W)}{h_{opt}(p, W)}$. Εδώ πρέπει να παρατηρήσουμε επίσης ότι $h = \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil$ είναι ο αριθμός των βημάτων που τρέχει το πρωτόκολλο για τη διάδοση της πληροφορίας σε μια κατεύθυνση k .

Σύμφωνα με τις παραπάνω παρατηρήσεις η πιθανότητα επιτυχούς μετάδοσης στο τελευταίο βήμα είναι:

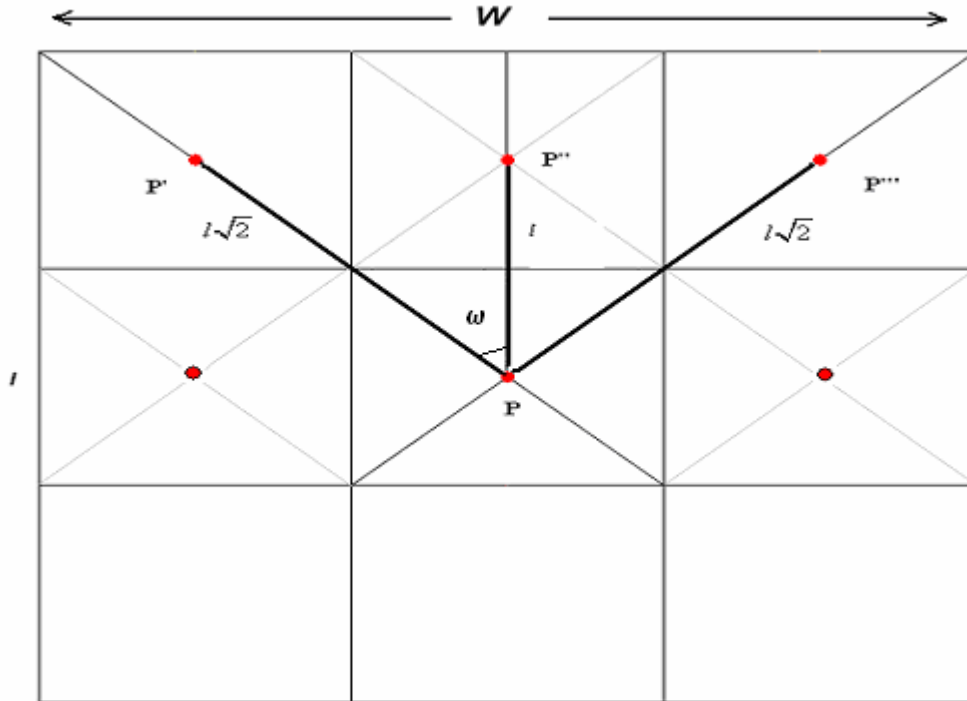
$$P(success_h) = \frac{3 \left(N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil + 1 \right)}{q}$$

Συνολικά η πιθανότητα επιτυχούς μετάδοσης κατά μήκους ενός μονοπατιού $P(psucces)$ είναι:

$$\begin{aligned} P(psucces) &= P(success_1) \cdot P(success_2 | success_1) \cdot \dots \cdot P(success_h | success_{h-1} \cap \dots \cap success_1) \\ \Rightarrow P(psucces) &= \frac{3N}{q} \cdot \frac{3(N-1)}{q} \cdot \dots \cdot \frac{3 \left(N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil + 1 \right)}{q} = \left(\frac{3}{q} \right)^{\left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil} \frac{N!}{\left(N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil \right)!} \\ \Rightarrow P(psucces) &= \left(\frac{3}{q} \right)^{\left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil} \frac{N^N e^{-N} \sqrt{2\pi N}}{\left(N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil \right)^{\left(N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil \right)} e^{N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil} \sqrt{2\pi \left(N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil \right)}} \end{aligned}$$

Σύμφωνα με το ορισμό 5, η path forwarding failure probability είναι αντίθετη της πιθανότητας $P(psucces)$. Έτσι με βάση τη παραπάνω ανάλυση:

$$\begin{aligned} P_{pfailure} &= 1 - P(psucces) = \\ &= 1 - \left(\frac{3}{q} \right)^{\left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil} \frac{N^N e^{-N} \sqrt{2\pi N}}{\left(N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil \right)^{\left(N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil \right)} e^{N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil} \sqrt{2\pi \left(N - \left\lceil \varepsilon \frac{d(P,W)}{R} \right\rceil \right)}} \end{aligned}$$



Σχήμα 18.4. : Βήμα για μετάδοση της πληροφορίας.

18.6 ΙΔΙΟΤΗΤΕΣ ΤΟΥ MP-DFR

Σε αυτή τη παράγραφο θα αποδείξουμε κατά πόσο το MP-DFR ικανοποιεί τις ιδιότητες της *ορθότητας* και της *αποδοτικότητας της ενέργειας* που είχαμε ορίσει στη παράγραφο 18.4. Στο επόμενο μέρος οι θεωρητικές αυτές αποδείξεις θα συμπληρωθούν με τα αποτελέσματα του simulation, και επίσης θα τεκμηριωθεί πειραματικά η στιβαρότητα (robustness) του πρωτοκόλλου.

18.6.1 Ορθότητα του MP-DFR

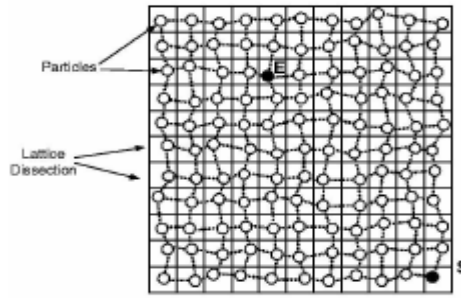
Ας θεωρήσουμε το διαχωρισμό της περιοχής του δικτύου σε τετράγωνα, και τα particles μέσα σε αυτά να σχηματίζουν ένα πλέγμα (σχήμα 18.5). Ας θεωρήσουμε επίσης όπως και προηγουμένως ότι η πλευρά κάθε τετραγώνου είναι l , και ότι ο αριθμός τους είναι q . Άρα η περιοχή που καλύπτεται είναι και πάλι ql^2 . Αν υποθέσουμε ότι ρίχνουμε στη περιοχή ενδιαφέροντος $N = aq \log q$ particles (όπου $a > 0$ κατάλληλη σταθερά), τότε η πιθανότητα ένα τετράγωνο να μη περιέχει κάποιο particles είναι:

$$\left(1 - \frac{1}{q}\right)^{aq \log q} \leq e^{-a \log q} = q^{-a}$$

Βασιζόμενοι στη παραπάνω σχέση, η πιθανότητα όλα τα τετράγωνα να περιέχουν particles είναι τουλάχιστον:

$$1 - q \cdot q^{-a} = 1 - q^{-(a-1)} = 1 - \left(\Theta \left(\frac{N}{\log N} \right) \right)^{-(a-1)}$$

Αν θεωρήσουμε ένα στιγμιότυπο F όπου κάθε τετράγωνο περιέχει particles, τότε το πεδίο ενδιαφέροντος μοντελοποιείται σαν lattice, όπως φαίνεται και στο σχήμα 18.5. Αν υποθέσουμε λοιπόν ότι ισχύει το στιγμιότυπο F και ότι ολόκληρο το δίκτυο είναι λειτουργικό, είναι τετριμμένο να αποδείξουμε την ορθότητα του πρωτοκόλλου με βάση την διαδικασία διάδοσης της πληροφορίας που έχουμε περιγράψει, αφού κάθε μετάδοση θα είναι επιτυχής (θα υπάρχουν δηλαδή πάντα particles στο σύνολο των υποψήφιων γειτόνων C).



Σχήμα 18.4. : Lattice G

18.6.2 Αποδοτικότητα ενέργειας του MP-DFR

Για την ανάλυση της αποδοτικότητας ενέργειας του MP-DFR, θα θεωρήσουμε ότι το πεδίο ενδιαφέροντος καλύπτει μια τετραγωνική περιοχή στην οποία είναι ομοιόμορφα κατανεμημένοι N κόμβοι. Να σημειώσουμε ότι δεν υποθέτουμε την ύπαρξη lattice. Έστω P το particle που ανιχνεύει το σημαντικό γεγονός, και έστω $d(P, W)$ η κάθετη απόσταση του P και κάποιου wall W . Επίσης ας υποθέσουμε ότι $d(P, W)_{\max} = \max \{d(P, W)\}$ είναι η κάθετη απόσταση του P και κάποιου wall W που αντιστοιχεί στο μεγαλύτερο μονοπάτι που μπορεί να κατασκευαστεί από το πρωτόκολλο. Ο βέλτιστος αριθμός hops για το πιο μεγάλο μονοπάτι είναι $h_{opt_max} = \left\lceil \frac{d(P, W)_{\max}}{R} \right\rceil$ όπου R είναι το εύρος μετάδοσης του particle. Τέλος ας υποθέσουμε ότι $K = \{k, k_1, \dots, k_i\}$ είναι το σύνολο μονοπατιών που υπολογίζονται από το πρωτόκολλο.

Θεώρημα 1: Η αποδοτικότητα ενέργειας του MP-DFR είναι $O\left(\frac{h_{opt_max} \cdot |K|}{N}\right)$

Απόδειξη: Η απόδειξη του παραπάνω θεωρήματος βασίζεται στη μέση αποδοτικότητα σε hops του LTP που αποδεικνύεται στο [13]. Ας υποθέσουμε ότι η

ακολουθία σημείων (particles) του μονοπατιού που δημιουργείται από το πρωτόκολλο μετά την ανίχνευση κάποιου σημαντικού γεγονότος E από ένα particle P , είναι $P_0 = P, P_1, P_2, \dots, P_{h-1}, P_h$. Το P_{h-1} είναι ένα boundary particle και βρίσκεται εντός της εμβέλειας κάποιου wall, ενώ το P_h είναι κομμάτι του wall (μπορεί να είναι ένας κινητός sink που καλύπτει το wall). Ας υποθέσουμε επίσης ότι a_i είναι η γωνία (θετική ή αρνητική) που σχηματίζει το P_i με την ευθεία κάθετη γραμμή του P_{i-1} με το wall. Τα παραπάνω μπορούν να γίνουν αντιληπτά με τη παρατήρηση του σχήματος 16.1. Ισχύει:

$$\sum_{i=1}^{h-1} d(P_{i-1}, P_i) \leq d(P, W) \leq \sum_{i=1}^h d(P_{i-1}, P_i)$$

Από τη στιγμή που κατά τη διάδοση της πληροφορίας στο W ισχύει $d(P_{i-1}, P_i) = R \cdot \cos a_i$ τότε:

$$\sum_{i=1}^{h-1} \cos a_i \leq h_{opt} \leq \sum_{i=1}^h \cos a_i$$

Από την ανισότητα του Wald για τη μέση τιμή του αθροίσματος τυχαίων αριθμών, ανεξάρτητων τυχαίων μεταβλητών, ισχύει:

$$E(h-1) \cdot E(\cos a_i) \leq E(h_{opt}) = h_{opt} \leq E(h) \cdot E(\cos a_i)$$

Επίσης $\forall i$ ισχύει $E(\cos a_i) = \int_{-a}^a \cos x \left(\frac{1}{2a} \right) dx = \frac{\sin a}{a}$. Έτσι

$$\frac{a}{\sin a} \leq \frac{E(h)}{h_{opt}} \leq \frac{a}{\sin a} + \frac{1}{h_{opt}} \Rightarrow E(C_h) \approx \frac{a}{\sin a}$$

Για $0 \leq a \leq \frac{\pi}{2}$ ισχύει $1 \leq \frac{a}{\sin a} \leq \frac{\pi}{2}$. Έτσι προκύπτει ότι $1 \leq E(C_h) \leq \frac{\pi}{2} \approx 1.57$. Έτσι

$$\text{ισχύει } \frac{E(h)}{E(h_{opt})} \approx 1.57 \Rightarrow E(h) \leq 1.57 \cdot \left\lceil \frac{d(P, W)_{\max}}{R} \right\rceil.$$

Η αποδοτικότητα του πρωτοκόλλου (r) είναι ο αριθμός των particles που ενεργοποιούνται κατά την εκτέλεση του πρωτοκόλλου προς το συνολικό αριθμό των particles. Θεωρώντας ότι $E(h)$ είναι ο μέσος αριθμός των particles που ενεργοποιούνται για κάθε μονοπάτι που δημιουργείται, ισχύει:

$$r = \frac{E(h) \cdot |K|}{N} \Rightarrow r \leq \frac{1.57 \cdot \left\lceil \frac{d(P, W)_{\max}}{R} \right\rceil \cdot |K|}{N} = \frac{1.57 \cdot h_{opt_max} \cdot |K|}{N}$$

Έτσι αποδείξαμε το θεώρημα.

ΜΕΡΟΣ 6^ο : ΕΞΟΜΟΙΩΣΗ & ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

19 ΕΞΟΜΟΙΟΤΗΣ SIMDUST

Στα πλαίσια αυτής της διπλωματικής εργασίας εξομοιώσαμε τη λειτουργία του MP-DFR πρωτοκόλλου και του μοντέλου των πολλαπλών sinks, χρησιμοποιώντας τον *SimDust*. Ο εξομοιωτής *SimDust* άρχισε να αναπτύσσεται από τον Ι. Χατζηγιαννάκη (υποψήφιο διδάκτορα του τμήματος), και εξελίχθηκε περαιτέρω στα πλαίσια των διπλωματικών εργασιών των Γ. Μυλωνά και Α. Αντωνίου. Στον *SimDust* υπάρχουν ήδη υλοποιημένα τα πρωτόκολλα PFR και LTP τα οποία παρουσιάσαμε παραπάνω, γεγονός που δίνει την δυνατότητα λήψης συγκριτικών αποτελεσμάτων όπως θα δούμε και παρακάτω.

Οι λόγοι για τη δημιουργία του *SimDust*, είναι η ανάγκη για την ύπαρξη κάποιου εξομοιωτή που θα μοντελοποιεί και θα δίνει τη δυνατότητα ανάπτυξης πρωτοκόλλων για smart dust δίκτυα. Για την ανάπτυξη τέτοιου τύπου πρωτοκόλλων είναι βέβαια διαθέσιμος και ο ns-2 (network simulator). Μάλιστα στον ns-2 έχει υλοποιηθεί ένα πρότυπο ασύρματου δικτύου που έχει τα χαρακτηριστικά ενός WSN, καθώς επίσης έχουν υλοποιηθεί και τα πρωτόκολλα LEACH και Directed Diffusion. Όμως ο ns-2 δεν είναι κατάλληλος για εξομοίωση πρωτοκόλλων όπως το PFR, LTP και το MP-DFR, μιας και αυτά τα πρωτόκολλα δίνουν μια γεωμετρική διάσταση στο πρόβλημα του *local detection and propagation*, η οποία δύσκολα μπορεί να αποτυπωθεί στον ns-2. Στον ns-2 υπάρχουν διαθέσιμα πολλά δικτυακά πρωτόκολλα (TCP, UDP...), καθώς και υλοποιήσεις σε επίπεδο MAC ενσύρματων (Ethernet) και ασύρματων (IEEE 802.11) δικτύων καθώς και διαφορετικές τοπολογίες. Αν λοιπόν σε μια υλοποίηση των πρωτοκόλλων PFR, LTP, MP-DFR εστιάζαμε για παράδειγμα στις συγκρούσεις που συμβαίνουν σε επίπεδο MAC, θα χάναμε την ουσία των πρωτοκόλλων που είναι ο τρόπος με τον οποίο γίνεται το propagation. Αν λόγω της πολυπλοκότητας του ns-2, δεν μπορούσαμε να μοντελοποιήσουμε ένα δίκτυο με διαβαθμίσεις στον αριθμό των κόμβων (από μερικές δεκάδες κόμβους μέχρι χιλιάδες), δεν θα μπορούσαμε να δούμε με σαφήνεια το ρόλο της πυκνότητας των κόμβων στο τρόπο που γίνεται η διάδοση της πληροφορίας και να έχουμε σαφή αποτελέσματα για τη λειτουργία των πρωτοκόλλων. Η απλότητα λοιπόν του *SimDust* μας έδωσε τη δυνατότητα να έχουμε μια πιο σαφή εικόνα για τη λειτουργία κάθε πρωτοκόλλου.

19.1 ΠΛΑΤΦΟΡΜΑ ΥΛΟΠΟΙΗΣΗΣ

Ως πλατφόρμα υλοποίησης, επιλέχθηκε το λειτουργικό σύστημα Linux και αλγοριθμική βιβλιοθήκη LEDA. Πιο συγκεκριμένα επιλέχθηκε η έκδοση 4.3.1 της LEDA καθώς και ο compiler gcc 3.2.

Η χρήση της βιβλιοθήκης LEDA ήταν ένας προφανής τρόπος για να υλοποιηθεί το «γεωμετρικό μοντέλο» του simulator. Αυτό γιατί η LEDA δίνει τη δυνατότητα δημιουργίας σημείων στο δισδιάστατο επίπεδο, μέτρησης γωνιών και δυνατότητα υπολογισμού ευκλείδειων αποστάσεων. Έτσι οι κόμβοι μπορούσαν να υλοποιηθούν σαν σημεία στο επίπεδο, το οποίο ορίζεται να έχει συγκεκριμένες διαστάσεις. Επίσης η LEDA παρέχει ένα πλήθος δομών δεδομένων και πολλές συναρτήσεις για τη διαχείρισή τους, διευκολύνοντας έτσι την υλοποίηση. Οι υλοποιήσεις της LEDA είναι πάρα πολύ αποδοτικές, και επίσης παρέχουν ένα απλό

και ενοποιημένο interface για της συναρτήσεις της, κάνοντας τη χρήση της LEDA ιδιαίτερα απλή.

Το Linux από την άλλη πλευρά, είναι ένα λειτουργικό σύστημα το οποίο είναι αξιόπιστο και διευκολύνει τη χρήση της LEDA. Επίσης, αν και η LEDA είναι διαθέσιμη και σε Microsoft Windows, η αρχική της υλοποίηση είναι για Linux, γεγονός που του δίνει ένα σημαντικό πλεονέκτημα.

Αφού η LEDA είναι υλοποιημένη σε C++, αυτή η γλώσσα προγραμματισμού ήταν η επόμενη προφανής επιλογή για την υλοποίηση του simulator. Εκτός από αυτό η C++ είναι μια γλώσσα object-oriented και άρα δίνει τη δυνατότητα για μία αντικειμενοστραφή σχεδίαση που είναι ιδιαίτερα βολική στη περίπτωση του simulator. Επίσης η C++ είναι αρκετά γρήγορα (λίγο πιο αργή από τη C), πράγμα που της δίνει πλεονέκτημα σε σχέση με τις άλλες object-oriented γλώσσες προγραμματισμού.

Τέλος για την ομαλή συνεργασία κατά την ανάπτυξη του κώδικα, χρησιμοποιήθηκε το πρόγραμμα CVS, το οποίο επιτρέπει εύκολα και γρήγορα την ταυτόχρονη εργασία μια ομάδας προγραμματιστών στα αρχεία ενός κοινού project παρέχοντας ευκολίες, όπως έλεγχο έκδοσης (version control), τήρηση αρχείων αλλαγών και άλλες ευκολίες.

19.2 ΟΡΓΑΝΩΣΗ ΚΩΔΙΚΑ ΣΤΟΝ SIMDUST

Ο κώδικας του SimDust είναι οργανωμένος σε αρχεία και καταλόγους που τονίζουν τη λογική του δομή. Η δομή αυτή παρουσιάζεται στο σχήμα 19.1 και όπως φαίνεται αποτελείται από πέντε βασικούς καταλόγους.

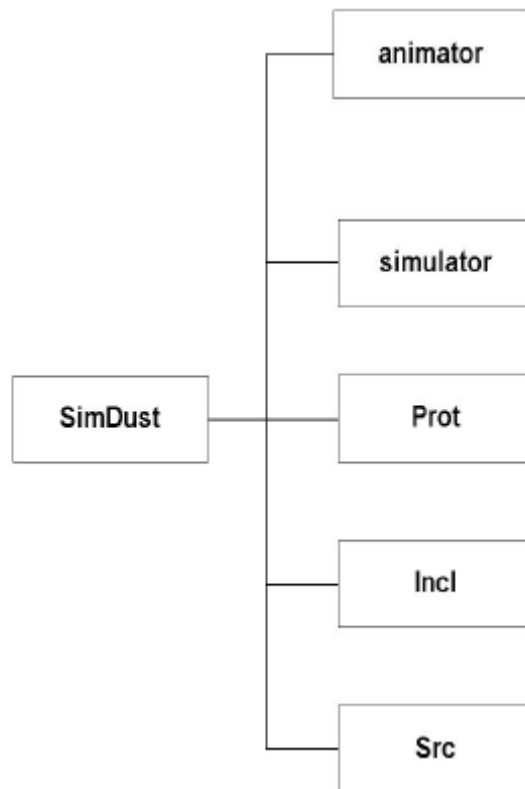
Οι πρώτοι δύο κατάλογοι περιέχουν το κώδικα για να παραχθούν τελικά τα δύο εκτελέσιμα αρχεία που αποτελούν τον εξομοιωτή, και είναι τα simulator και animator. Ο simulator είναι ο εξομοιωτής του οποίου η λειτουργία γίνεται μέσω της γραμμής εντολών, και ο animator είναι το εκτελέσιμο αρχείο που αναφέρεται στην οπτικοποίηση των πειραμάτων που εκτελούνται από το simulator. Ο animator είναι μια παραθυρική εφαρμογή που εκμεταλλεύεται τις βιβλιοθήκες της LEDA για animation δικτύων, και τρέχει σε περιβάλλον X-Windows.

Ο διαχωρισμός μεταξύ simulator και animator (σε χωριστούς καταλόγους) έγινε για λόγους απλότητας της υλοποίησης. Από κει και πέρα και τα δύο modules χρησιμοποιούν τα ίδια αρχεία που υπάρχουν και στους υπόλοιπους καταλόγους για την εκτέλεση των πρωτοκόλλων.

Στη συνέχεια στο κατάλογο *Incl* υπάρχουν τα header αρχεία των βασικών αρχείων υλοποίησης του εξομοιωτή (δηλαδή των .cpp αρχείων) που βρίσκονται στο κατάλογο *Src*. Επίσης πέρα από τα header αρχεία κάποιων βασικών .cpp του καταλόγου *Src*, στο *Incl* υπάρχουν ένα header (*constants.h*) που περιέχει τα βασικά definitions (#define) του κώδικα καθώς και το *include.h* με τις βασικές βιβλιοθήκες που απαιτούνται, καθώς και με τους νέους τύπους δεδομένων που ορίζονται.

Ο κατάλογος *Src* όπως αναφέραμε και παραπάνω, περιέχει τις υλοποιήσεις κάποιων βασικών κλάσεων (τα headers τους βρίσκονται στον *Incl*) που υλοποιούν κάποιες βασικές λειτουργίες του δικτύου, όπως κατασκευή μηνυμάτων, πληροφοριών, cache μνήμης, αλλά και κατασκευή του ίδιου του δικτύου, όπως κατασκευή particles, sinks κ.α. Παρακάτω θα αναφερθούμε λίγο αναλυτικότερα σε αυτές τις κλάσεις.

Τέλος, στο κατάλογο *Prot* υλοποιούνται τα ξεχωριστά πρωτόκολλα του SimDust. Κάποια πρωτόκολλα τα οποία έχουν υλοποιηθεί είναι το LTP (και κάποιες παραλλαγές του), το PFR, το TEEN ([37]) κ.α. Σε αυτό το κατάλογο τοποθετήσαμε το κώδικα υλοποίησης για το MP-DFR και τις παραλλαγές του όπως θα αναφέρουμε και παρακάτω.



Σχήμα 19.1. : Οργάνωση SimDust σε καταλόγους.

19.3 ΣΗΜΑΝΤΙΚΕΣ ΚΛΑΣΕΙΣ ΓΙΑ ΤΗ ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ SIMDUST

Σε αυτή τη παράγραφο θα αναφέρουμε τις πιο βασικές κλάσεις του SimDust και θα αναλύσουμε τη λειτουργικότητα της καθεμίας από αυτές. Η λεπτομερής ανάλυση του κώδικα κάθε κλάσης, είναι έξω από το σκοπό αυτής της διπλωματικής εργασίας. Περισσότερες λεπτομέρειες για την υλοποίηση του SimDust υπάρχουν στις διπλωματικές εργασίες των Γ. Μυλωνά και Α. Αντωνίου. Κάποιες λεπτομέρειες θα αναφέρουμε μόνο για την υλοποίηση του MP-DFR στις επόμενες παραγράφους.

19.3.1 Η κλάση Execution

Η υλοποίηση της κλάσης αυτής γίνεται σε δύο αρχεία, τα *Execution.h* και *Execution.cpp* τα οποία περιέχονται στους καταλόγους *Incl* και *Src* αντίστοιχα. Η κλάση αυτή αποτελεί μια αρκετά αφαιρετική αναπαράσταση του πεδίου που βρίσκεται το WSN. Περιέχει πληροφορίες όπως οι διαστάσεις του πεδίου

ενδιαφέροντος, δείκτες στους κόμβους του δικτύου, το τρέχον γύρο, μια λίστα από γεγονότα που συμβαίνουν στο κάθε γύρο και θα εκτελεστούν στο τρέχοντα γύρο.

Η κλάση αυτή είναι βασική για τη λειτουργία του δικτύου αφού όταν φτιάχνεται ένα στιγμιότυπο αυτής, τότε αρχικοποιείται το δίκτυο. Λέγοντας αρχικοποίηση, εννοούμε ότι κατασκευάζονται τα particles ως σημεία στο επίπεδο, ανάλογα με την κατανομή που θα δώσει ο χρήστης (random, ή lattice) και τα particles αυτά αρχικοποιούνται με το να βρουν τα γειτονικά τους particles καθώς και άλλα στοιχεία.

Από τη κλάση αυτή τέλος, κληρονομούν οι κλάσεις Experiment και Animation, οι οποίες είναι ουσιαστικά ο animator και ο simulator αντίστοιχα.

19.3.2 Η κλάση Particle

Η κλάση αυτή υλοποιεί ουσιαστικά το κόμβο του δικτύου έξυπνης σκόνης. Η κλάση αυτή κληρονομεί από τη τάξη point της LEDA, η οποία συμβολίζει ένα σημείο στο επίπεδο με συντεταγμένες (x, y) . Επιπλέον σε αυτή τη κλάση προστίθεται πληροφορία όπως το ID του particle, η λίστα με τους γείτονες, η διαθέσιμη ενέργεια, μεταβλητές που δείχνουν τη κατάσταση του κόμβου (αν είναι ενεργός, σε κατάσταση sleep..κ.α.) καθώς και άλλες πληροφορίες. Ο κώδικας για τη κλάση αυτή βρίσκεται στα αρχεία Incl/Particle.h και Src/Particle.cpp.

Η τάξη αυτή είναι πολύ γενική και χρησιμοποιείται για τη παραγωγή κλάσεων κόμβων που έχουν ειδικό ρόλο, όπως ο κόμβος-sink (SINKParticle) και οι κόμβοι για καθένα από τα πρωτόκολλα που εξομοιώνουμε (LTPParticle, PFRParticle, MP_DFRParticle) οι οποίοι έχουν τα εξειδικευμένα χαρακτηριστικά που απαιτεί κάθε πρωτόκολλο.

Τέλος η κλάση παρέχει και ένα interface για να γίνεται εναλλαγή καταστάσεων μεταξύ sleep και awake για του κόμβους του δικτύου, καθώς και ένα δείκτη, που δείχνει σε αντικείμενο τύπου Network. Η κλάση Network θα αναλυθεί παρακάτω.

19.3.3 Η κλάση Network

Η κλάση Network αναφέρεται στο επικοινωνιακό κομμάτι του δικτύου, και όχι στα «μορφολογικά» χαρακτηριστικά του, όπως οι διαστάσεις, η κατανομή των particles κ.τ.λ. Αυτά τα «μορφολογικά» χαρακτηριστικά είπαμε ότι βρίσκονται κυρίως στις κλάσεις Particle και Execution.

Έτσι λοιπόν στη κλάση αυτή υπάρχουν συναρτήσεις που διεκπεραιώνουν την επικοινωνία μεταξύ κόμβων, όπως η αποστολή μηνύματος σε γείτονα, broadcast σε όλους τους γείτονες, broadcast σε γείτονες που βρίσκονται σε μια ημικυκλική περιοχή (π.χ. στο LTP) κ.τ.λ. Μάλιστα υπάρχουν και δύο διαφορετικές υλοποιήσεις της τάξης αυτής, μια για τον simulator (αρχείο simulator/Network_exp.cpp) και μια για τον animator (αρχείο animator/Network_anim.cpp), ενώ ο ορισμός του βρίσκεται στο αρχείο Incl/Network.h.

Ο λόγος πίσω από το σχεδιασμό αυτό είναι ότι τα δύο εκτελέσιμα καλούν τις ίδιες συναρτήσεις. Στη περίπτωση του animator όμως, υπάρχει επιπλέον η οπτικοποίηση της λειτουργίας των πρωτοκόλλων, οπότε απαιτείται να χρησιμοποιηθούν επιπλέον βιβλιοθήκες της LEDA.

19.3.4 Η κλάση Info

Η κλάση αυτή χρησιμοποιείται στα αντικείμενα της κλάσης Message, την οποία θα αναλύσουμε παρακάτω. Ουσιαστικά αναπαριστά τη πληροφορία που μεταφέρει ένα μήνυμα και είναι απαραίτητη για τη λειτουργία του πρωτοκόλλου. Ο κώδικας της κλάσης αυτής βρίσκεται στα Incl/Info.h και Src/Info.cpp.

Πέρα από το ότι η κλάση αυτή περιλαμβάνει κάποια ιδιαίτερα χαρακτηριστικά για κάθε πρωτόκολλο, αποθηκεύει και το particle initiator της πληροφορίας (δηλαδή το particle που ανίχνευσε τη πληροφορία), καθώς και τη χρόνο-σφραγίδα της, χαρακτηριστικά που περιγράφουν μοναδικά τη πληροφορία.

19.3.5 Η κλάση Message

Η κλάση αυτή αναπαριστά τα μηνύματα που ανταλλάσσονται μεταξύ των κόμβων. Όταν δημιουργείται ένα μήνυμα, του αναθέτουμε κάποια πληροφορία (*Info*) και από κει και πέρα, το μήνυμα μπορεί να είναι διαφόρων τύπων και να εξυπηρετεί διαφορετικούς σκοπούς ανάλογα με το πρωτόκολλο. Για παράδειγμα μπορεί ένας κόμβος με ένα μήνυμα να ενημερώνει του γείτονες του ότι θέλει να στείλει μήνυμα προς το κέντρο ελέγχου (sink), και με την σειρά τους να απαντούν αν είναι πρόθυμοι να το μεταφέρουν παραπέρα. Ο κώδικας της κλάσης αυτής βρίσκεται στα αρχεία Incl/Message.h και Src/Message.cpp.

19.3.6 Η κλάση Event

Η κλάση αυτή χρησιμοποιείται για να αναπαραστήσει ένα γεγονός το οποίο συμβαίνει σε ένα particle κάποια χρονική στιγμή και επηρεάζει τη λειτουργία του. Αυτό το γεγονός μπορεί να περιλαμβάνει το triggering ενός κόμβου (δηλαδή τη περίπτωση όπου ένα κόμβος ανιχνεύει ένα γεγονός και πρέπει να διαδώσει στο κέντρο ελέγχου), ή τη λήψη κάποιου μηνύματος από ένα γειτονικό κόμβο. Ο κώδικας της κλάσης αυτής βρίσκεται στα αρχεία Incl/Event.h και Src/Event.cpp.

19.3.7 Η κλάση Cache_info_Obj

Η κλάση αυτή υλοποιεί τη cache μνήμη για κάθε particle. Συγκεκριμένα κάθε particle (κλάση Particle) διαθέτει μία λίστα σε αντικείμενα τύπου *Cache_info_Obj* (τη λίστα *infoCache*) που ουσιαστικά αντιπροσωπεύει τη cache για κάθε particle. Η κλάση *Cache_info_Obj* κρατάει κάποια βασικά στοιχεία, όπως είναι βέβαια η πληροφορία (*Info*) η οποία αποθηκεύεται στη cache, ο χρόνος ζωής της πληροφορίας στη cache, τη χρόνο-σφραγίδα άφιξης της πληροφορίας κ.α. Ο κώδικας της κλάσης αυτής βρίσκεται στα αρχεία Incl/Cache_info_Obj.h και Src/Cache_info_Obj.cpp.

19.3.8 Η κλάση Experiment

Η κλάση αυτή κληρονομεί από την Execution και περιγράφει ουσιαστικά το πώς τρέχει ένα πείραμα στο simulator. Με τη δημιουργία ενός αντικειμένου τύπου Experiment, καλείται λοιπόν και ο constructor της κλάσης Execution και έτσι αρχικοποιείται το δίκτυο. Στη συνέχεια η συνάρτηση execute αυτής της κλάσης, τρέχει ένα for loop ανάλογα με τον αριθμό των γύρων που έχουμε ορίσει να τρέξει το πείραμα, και σε κάθε ένα από αυτούς τους γύρους τρέχει και ένα βήμα του εκάστοτε πρωτοκόλλου. Ο κώδικας για τη κλάση Experiment βρίσκεται στα αρχεία Simulator/Experiment.h και Simulator/Experiment.cpp.

19.3.9 Η κλάση Animation

Η κλάση αυτή είναι το αντίστοιχο Experiment για τον animator. Ουσιαστικά έχει επιπρόσθετα ένα αντικείμενο GraphWin της LEDA, και διαφορετική υλοποίηση για το αντικείμενο Network. Έτσι εκτελείται ουσιαστικά το loop που εκτελείται και στο simulator, αλλά παράλληλα γίνεται και ένα animation στο δίκτυο WSN που εξομοιώνεται. Ο κώδικας της κλάσης αυτής βρίσκεται στα αρχεία Animator/Animation.h και Animator/Animation.cpp.

19.3.10 Τα αρχεία misc.h και misc.cpp

Στο αρχείο misc.h ορίζονται και στο αρχείο misc.cpp υλοποιούνται κάποιες πολύ βασικές συναρτήσεις που χρησιμοποιούνται στη συνέχεια από τη κλάση Execution για να γίνει η αρχικοποίηση του δικτύου. Για παράδειγμα η συνάρτηση *makeSetupRANDOM* δημιουργεί particles με βάση τυχαία και ομοιόμορφη κατανομή, ενώ η συνάρτηση *makeNeighbourhoodsWall* χρησιμοποιείται στο μοντέλο πηγής-τείχους ώστε κάθε particle να μπορεί να υπολογίσει τους γείτονες του σε σχέση με το wall. Αναφέρουμε την λειτουργία που παρέχουν αυτά τα αρχεία, μιας και μέσα σε αυτά υλοποιήσαμε επιπλέον συναρτήσεις για το μοντέλο των πολλαπλών sinks που χρησιμοποιείται από το MP-DFR.

19.4 ΥΛΟΠΟΙΗΣΗ ΤΟΥ MULTI-SINK ΜΟΝΤΕΛΟΥ ΚΑΙ ΤΟΥ MP-DFR

Στη παράγραφο αυτή θα αναφερθούμε στην υλοποίηση του multi-sink μοντέλου και του πρωτοκόλλου MP-DFR στον SimDust. Συγκεκριμένα θα αναφερθούμε στους βασικούς άξονες της υλοποίησης (στις κύριες συναρτήσεις) και δεν θα αναλωθούμε σε λεπτομέρειες στο κώδικα.

19.4.1 Υλοποίηση multi-sink μοντέλου

Κατ' αρχήν στην υλοποίηση και στα πειράματά μας θεωρήσαμε ότι είναι δυνατόν να υπάρχουν κινητοί sinks σε ολόκληρη τη περίμετρο του πεδίου ενδιαφέροντος.

Επίσης, όπως αναφέραμε και στην ανάλυση του MP-DFR, τα μονοπάτια που δημιουργούνται κατά την εφαρμογή του πρωτοκόλλου ουσιαστικά είναι διαδρομές προς διαφορετικές κατευθύνσεις. Για να μπορέσουμε λοιπόν να «υλοποιήσουμε» αυτές τις διαφορετικές κατευθύνσεις, δημιουργήσαμε sinks περιμετρικά στο δίκτυο, δηλαδή δημιουργήσαμε SINKParticles που έχουν όλες τις ιδιότητες του sink και που είναι σημεία στο επίπεδο και άρα μπορούν να ορίσουν διαφορετικές κατευθύνσεις. Ο αριθμός αυτών των particles καθορίζεται από το πόσα επίπεδα προτεραιότητας (priority levels) επιθυμεί ο διαχειριστής του δικτύου να ορίσει. Στο αρχείο Incl/constants.h δηλώσαμε **#define PATH_LEVELS 3** που σημαίνει ότι ορίζουμε τρία επίπεδα μονοπατιών. Ο τρόπος με τον οποίο ορίζονται οι συντεταγμένες των sink κόμβων που δημιουργούμε, είναι αυτός που αναφέραμε στην ανάλυση του MP-DFR, δηλαδή για παράδειγμα για να προκύψουν οι κατευθύνσεις του $plevel_2$ πρέπει να διχοτομηθούν οι γωνίες που έχουν προκύψει από τις κατευθύνσεις του $plevel_1$. Συμπερασματικά οι sinks που δημιουργούνται όταν έχουμε ορίσει τρία επίπεδα μονοπατιών είναι 16, και άρα μπορούν να δημιουργηθούν το πολύ 16 κατευθύνσεις και άρα 16 μονοπάτια. Η συνάρτηση που κατασκευάζει τους sinks είναι η **makeSetupSINKS** και ορίζεται στο αρχείο Incl/misc.h και υλοποιείται στο Src/misc.cpp.

Στα ίδια αρχεία (misc.h και misc.cpp) ορίζεται και υλοποιείται και η συνάρτηση **setupNeighbourhoodsSink**. Κάθε particle διατηρεί μία λίστα με δείκτες σε γειτονικά particles (δηλαδή σε particles που βρίσκονται εντός του εύρους μετάδοσης του R), αλλά διατηρεί επίσης και μια λίστα σε γειτονικά particles τα οποία έχουν την ιδιότητα να βρίσκονται μέσα σε μια συγκεκριμένη γωνία μετάδοσης (σχήμα 16.2). Επειδή λοιπόν κάθε sink ορίζει μια διαφορετική κατεύθυνση, οι γείτονες αυτοί για ένα particle p είναι διαφορετικοί για κάθε sink S . Η συνάρτηση λοιπόν **setupNeighbourhoodsSink** υπολογίζει και αποθηκεύει αυτούς τους γείτονες για κάθε sink.

19.4.2 Υλοποίηση του MP-DFR

Η υλοποίηση του MP-DFR βρίσκεται στα αρχεία prot/MP_DFR/MP_DFRParticle.h και prot/MP_DFR/MP_DFR.cpp. Στο header MP_DFRParticle.h υπάρχουν οι ορισμοί συναρτήσεων και μεταβλητών που δίνουν τις επιπλέον ιδιότητες στα particles και έτσι διαφοροποιούν το κάθε πρωτόκολλο. Αυτό που αξίζει να σχολιάσουμε είναι μια λίστα με δείκτες σε particles (*m_neighbours_sorted*), όπου εκεί αποθηκεύονται τα particles ταξινομημένα με βάση την απόστασή τους από κάποιο sink.

Στο MP_DFR.cpp γίνονται οι κύριες υλοποιήσεις των συναρτήσεων. Η μόνη συνάρτηση στην οποία θα σταθούμε είναι η **doExecute()** η οποία είναι αυτή που υλοποιεί βασικά το πρωτόκολλο. Ο κώδικας αυτής φαίνεται στο πίνακα 19.1.

Πίνακας 19.1. Συνάρτηση doExecute.

```
void
MP_DFRParticle::doExecute()
```



```

{

    int sinkID=0;
    int tmp_sinkID=0;
    list_item item;
    list<int> ProtSinks; //This list stores sinks calculated by the protocol

    // If we do not hold any event-info
    if (!exist_NewInfos())
        return;

    /*check if particle is next to some Wall and if yes, set it as Wall */
    isNextToAnyWall(m_MPDFR_R, m_MPDFRparticles, m_MPDFRsinks,
    this->getID());
    if (this->isNextToWall())
    {

        // LOOP ON ALL NEW UNSENT INFOS.
        Cache_Info_Obj *newInfoObj;
        while(exist_NewInfos())
        {
            newInfoObj = getFirstNewInfoObj();
            doForward(nil,newInfoObj);

        }
        return;
    }

    /*Search Phase */
    if (m_searched != true) {

        // Search for Neighbours
        Message* ms = new Message(MESSAGE_LTP_LS1, this);
        broadcast(ms); // Broadcast to all neighbours because we may
                       // construct many paths
        delete(ms);

        #ifdef DEBUG_MPDFR
        cout <<"\nSEARCH\n";
        #endif

        m_searched = true;
        m_waited = false;
        return;
    }

    if (m_waited != true) {
        // Wait an extra round to get all replies
        m_waited = true;
    }
}

```

```

#ifdef DEBUG_MPDFR
cout << "\nWAIT\n";
#endif

return;
}

// LOOP ON ALL NEW UNSENT INFOS.
Cache_Info_Obj *newInfoObj;
Info *strdinfo;
while(exist_NewInfos())
{
    newInfoObj = getFirstNewInfoObj();
    strdinfo = newInfoObj->getInfo();

    if(this->getID() == strdinfo->getInitiator()) //If particle is initiator it
                                                    //has to create paths
    {
#ifdef DEBUG_MPDFR
cout << "\n Initiator Particle";
fflush(stdout);
#endif

ProtSinks=RetSinks(this->getID(), 0); //Get the sink id's
                                     //calculated by the protocol

forall_items(item, ProtSinks){

    tmp_sinkID=ProtSinks.inf(item);
    strdinfo->setDestSink(tmp_sinkID); //Store sink ID to
                                       //Info

//We have to calculate new neighbours according to the specific Wall
makeNeighbourhoodsSink(m_MPDFR_R,
m_MPDFR_X, m_MPDFR_Y,
m_MPDFRparticles,m_MPDFRsinks, this->getID(),
tmp_sinkID);
// clear m_neighbours_sorted for the new set up
m_neighbours_sorted->clear();

//Sort neighbours according to distance
point dummy_sink = point(this-
>getWallPoint().xcoord(),this-
>getWallPoint().ycoord());

doSetup(&dummy_sink);

// Select next target
Particle *next = SenseNeighbours(newInfoObj);
// Check if we have found any!

```

```

        if (next == nil)
            // Failure ! - Backtrack (if we have any)
            {
                bool Backtrack_Possible = false; //Protocol does
                                                //not allow backtrack phase
                //doBacktrack(newInfoObj,
                //    &Backtrack_Possible);
            }
        else
        {
            // Success ! - Pass message to next
            doForward(next,newInfoObj);
        }
    }
    ProtSinks.clear();
}
else // If it is just a RECEIVE Info, just propagate it further
{
    sinkID=strdinfo->getDestSink();

#ifdef DEBUG_MPDFR
    cout << "\n Forwarding to Particle";
    cout << "\n Info of Sink: (" << sinkID << ")";
    fflush(stdout);
#endif

    //We have to calculate new neighbours according to the specific Wall

    makeNeighbourhoodsSink(m_MPDFR_R, m_MPDFR_X,
        m_MPDFR_Y, m_MPDFRparticles,m_MPDFRsinks, this-
        >getID(), sinkID);

    m_neighbours_sorted->clear(); //clear m_neighbours_sorted for
                                //the new set up
    //Sort neighbours according to distance
    point dummy_sink = point(this->getWallPoint().xcoord(),this-
        >getWallPoint().ycoord());
    doSetup(&dummy_sink);

    // Select next target
    Particle *next = SenseNeighbours(newInfoObj);
    // Check if we have found any!
    if (next == nil)
        // Failure ! - Backtrack (if we have any)
        {
            bool Backtrack_Possible = false; //Protocol does not
                                                //allow backtrack phase

```

```

        //doBacktrack(newInfoObj, &Backtrack_Possible);
    }
    else
    {
        // Success ! - Pass message to next
        doForward(next,newInfoObj);
    }
}
/* Even when the protocol does not manage to send info, it must be
   considered as sent */
newInfoObj->setSend(true);
}
}

```

Η συνάρτηση `doExecute` τρέχει αν υπάρχει κάποιο γεγονός προς μετάδοση. Τότε το `particle` ελέγχει αν βρίσκεται δίπλα (εντός εμβέλειας) σε κάποιο σημείο της περιμέτρου του δικτύου (καλείτε η συνάρτηση `isNextToAnyWall`). Αν πράγματι βρίσκεται, τότε διαδίδει τη πληροφορία στον αντίστοιχο sink. Αν το `particle` δεν βρίσκεται κοντά σε κάποιο wall, κάνει τα εξής. Αν το `particle` ανιχνεύσει κάποιο γεγονός (`if(this->getID() == strdinfo->getInitiator())`) τότε δημιουργεί μονοπάτια όπως προστάζει το πρωτόκολλο MP-DFR καλώντας τη συνάρτηση ***RetSinks***. Αν το γεγονός που συμβαίνει σε ένα `particle` δεν είναι triggered (ανίχνευση γεγονότος) αλλά μια πληροφορία που πρέπει να μεταδοθεί παραπέρα, τότε μεταδίδεται στο γείτονα που είναι πρώτος στη λίστα `m_neighbours_sorted` (αυτό γίνεται στη συνάρτηση ***SenseNeighbours***).

Πέρα από το MP-DFR, υλοποιήσαμε και μία παραλλαγή αυτού που είναι το MP-DFRa. Σε αυτό, καλύτερος υπονήγιος γείτονας δεν είναι αυτός που είναι πιο κοντά στο wall (όπως συμβαίνει στο MP-DFR), αλλά αυτός που είναι πιο κοντά στην ευθεία που ενώνει το `particle`, με τον sink στον οποίο στοχεύετε να διαδοθεί η πληροφορία. Δηλαδή επιθυμούμε να διαδίδουμε τη πληροφορία πιο κοντά στη κατεύθυνση του μονοπατιού. Η υλοποίηση του MP-DFRa βρίσκεται στα αρχεία `prot/MP_DFRa/MP_DFRaParticle.h` και `prot/MP_DFRa/MP_DFRa.cpp`. Η βασική διαφορά των δύο υλοποιήσεων είναι η συνάρτηση ***insertionSort*** που ταξινομεί τους γείτονες στη λίστα `m_neighbours_sorted`.

19.5 ΧΡΗΣΗ ΤΟΥ SIMULATOR

Δίνοντας στη γραμμή εντολών `./simulator` εμφανίζονται στην οθόνη μας οδηγίες με το πώς τρέχουμε ένα πείραμα. Μετά και την προσθήκη του πρωτοκόλλου MP-DFR, αλλά και του μοντέλου των πολλαπλών sinks οι παράμετροι του simulator γίνονται όπως φαίνονται στο πίνακα 19.2.

Πίνακας 19.2. Παράμετροι simulator.

Πρωτόκολλο	Πρωτόκολλο που τρέχουν οι κόμβοι του
------------	--------------------------------------

	δικτύου (LTP, PFR, MP_DFR)
Κατανομή	Κατανομή κόμβων στο δίκτυο. Τυχαία ομοιόμορφη κατανομή (R) και lattice (L)
Κέντρο Ελέγχου	Τύπος κέντρου ελέγχου. Απλός κόμβος sink (S), wall (W), πολλαπλοί sinks/walls (MW)
Πλήθος Κόμβων	Πλήθος των κόμβων του δικτύου
Διάρκεια Εξομοίωσης	Πλήθος γύρων εξομοίωσης
Επαναλήψεις	Πλήθος επαναλήψεων πειράματος
Διάσταση X	Διάσταση X του πεδίου
Διάσταση Y	Διάσταση Y του πεδίου
Ακτίνα Μετάδοσης	Εμβέλεια κόμβου R
ΔR	Χρησιμοποιείται στα PFR,LTP,MP-DFR και το θέτουμε συνήθως ίσο με την ακτίνα
Γωνία α	Γωνία μετάδοσης
Awake Period	Περίοδος κανονικής λειτουργίας
Sleep Period	Περίοδος sleep λειτουργίας
Παράμετρος Γεγονότων X	Προαιρετική παράμετρος. Αν τη θέσουμε είναι το πλήθος των γεγονότων στη διάρκεια της εξομοίωσης
Παράμετρος Γεγονότων Y	Προαιρετική παράμετρος. Αν έχει τεθεί η προηγούμενη παράμετρος, αν δώσουμε τιμή <1 είναι η πιθανότητα να συμβεί ένα γεγονός σε οποιονδήποτε γύρο, αλλιώς είναι το πλήθος των γεγονότων στη σειρά

Για παράδειγμα αν τρέξουμε στη γραμμή εντολών:

```
>./simulator MP_DFR R MW 600 120 10 51 51 5 5 90 0 0 100 0.3
```

σημαίνει ότι τρέχουμε το MP-DFR πρωτόκολλο, σε ένα δίκτυο με 600 κόμβους, για 120 γύρους και για 10 πειράματα σε ένα δίκτυο πολλαπλών sinks. Επίσης το δίκτυο μας έχει διαστάσεις 51×51 και κάθε κόμβος έχει εμβέλεια 5. Η γωνία μετάδοσης είναι 90 μοίρες, δεν υπάρχει sleep-awake και συνολικά συμβαίνουν 100 γεγονότα με πιθανότητα 0.3 σε οποιονδήποτε γύρο.

Περισσότερες λεπτομέρειες για την έξοδο του simulator αλλά και για την λειτουργία του animator υπάρχουν στις διπλωματικές εργασίες των Μυλωνά και Α. Αντωνίου.

20 ΠΕΙΡΑΜΑΤΙΚΑ ΑΠΟΤΕΛΕΣΜΑΤΑ

Σε αυτό το τελευταίο κεφάλαιο, θα παρουσιάσουμε κάποια πειραματικά αποτελέσματα που προέκυψαν από την παραπάνω υλοποίηση. Ουσιαστικά θα συγκρίνουμε το MP-DRR με τα πρωτόκολλα LTP και PFR, και θα τεκμηριώσουμε

έτσι τον ισχυρισμό μας ότι το πρωτόκολλο MP-DFR μπορεί να εγγυηθεί επιτυχημένη διάδοση της πληροφορίας, ακόμα και όταν το δίκτυο είναι αραιό και μη λειτουργικό. Να σημειώσουμε ότι η έκδοση του LTP που χρησιμοποιούμε, επιλέγει το βέλτιστο γείτονα όσον αφορά την απόσταση του από το wall, και διαθέτει και backtrack φάση.

Στα πειράματα που θα παρουσιάσουμε στη συνέχεια, δεν χρησιμοποιούμε κάποιον από τους μηχανισμούς για υπολογισμό των μονοπατιών που αναλύσαμε στη πρώτη φάση του πρωτοκόλλου του MP-DFR, και έτσι σε κάθε πείραμα δημιουργούμε ένα συγκεκριμένο αριθμό μονοπατιών. Επιλέξαμε αυτό τον τρόπο για τα πειράματα, θέλοντας περισσότερο να εστιάσουμε στο μοντέλο των multiple-sinks και στο τρόπο που διαδίδεται η πληροφορία, παρά στο αν τηρείται η πιθανότητα αξιοπιστίας, ή στο ποιος είναι ο βέλτιστος τρόπος για τον υπολογισμό μονοπατιών. Αυτά τα πειράματα αφήνονται ως μελλοντική έρευνα.

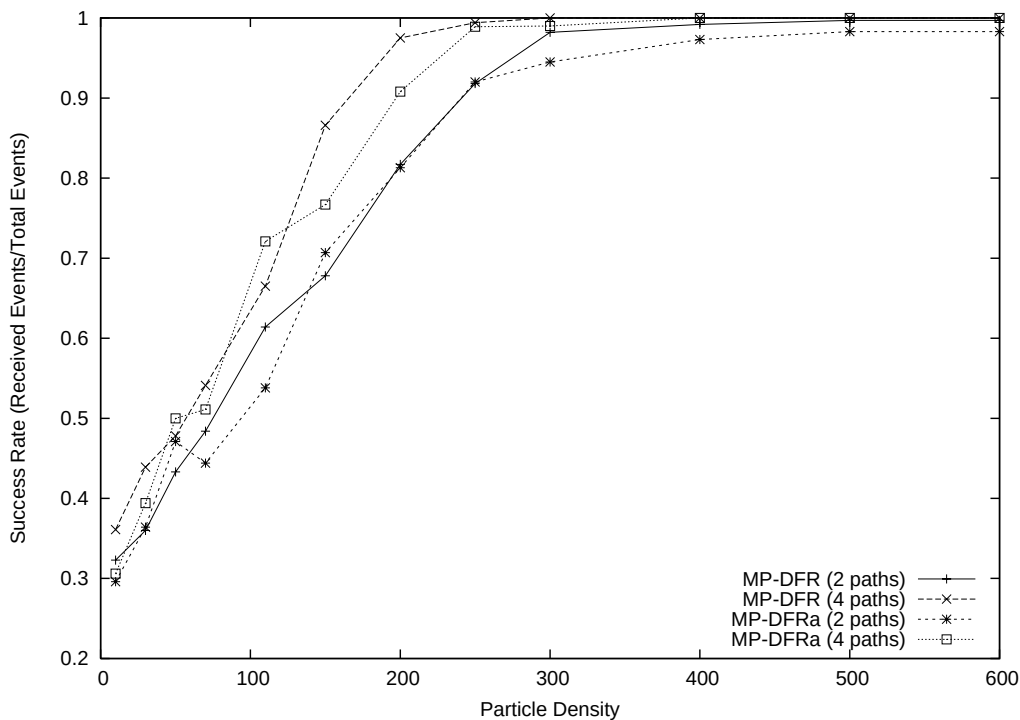
Επίσης τα πειράματα που ακολουθούν λήφθηκαν με τις εξής παραμέτρους:

- *Επαναλήψεις* = 10
- *Διάσταση X* = *Διάσταση Y* = 51
- *Ακτίνα Μετάδοσης* = 5
- $\Delta R = 5$
- *Γωνία* $\alpha = 90$
- *Awake Period* = 0
- *Awake Period* = 0
- *Παράμετρος Γεγονότων X* = 100
- *Παράμετρος Γεγονότων Y* = 1

Ο αριθμός των κόμβων που παρατάξαμε στο δίκτυο, καθώς και ο αριθμός των μονοπατιών που δημιουργούσαμε κάθε φορά, διαφέρει σε κάθε πείραμα.

20.1 MP-DFR vs MP-DFRa

Σε αυτή παράγραφο θα παραθέσουμε τα συγκριτικά αποτελέσματα δύο παραλλαγών του πρωτοκόλλου. Στο MP-DFRa όπως έχουμε αναφέρει, καλύτερος υποψήφιος γείτονας δεν είναι αυτός που είναι πιο κοντά στο wall (όπως συμβαίνει στο MP-DFR), αλλά αυτός που είναι πιο κοντά στην ευθεία που ενώνει το particle, με τον sink στον οποίο στοχεύετε να διαδοθεί η πληροφορία. Στο σχήμα 20.1 παρουσιάζεται το success rate των δύο παραλλαγών ως προς το συνολικό αριθμό των particles του δικτύου (Particle Density), όταν δημιουργούνται 2 και 4 μονοπάτια. Να σημειώσουμε ότι το success rate είναι το πλήθος των γεγονότων που παραλήφθηκαν από τους sinks, ως προς το συνολικό αριθμό γεγονότων που στάλθηκαν (*Received Events/Total Events*). Συμπερασματικά 100% επιτυχία έχουμε όταν *success rate* = 1.



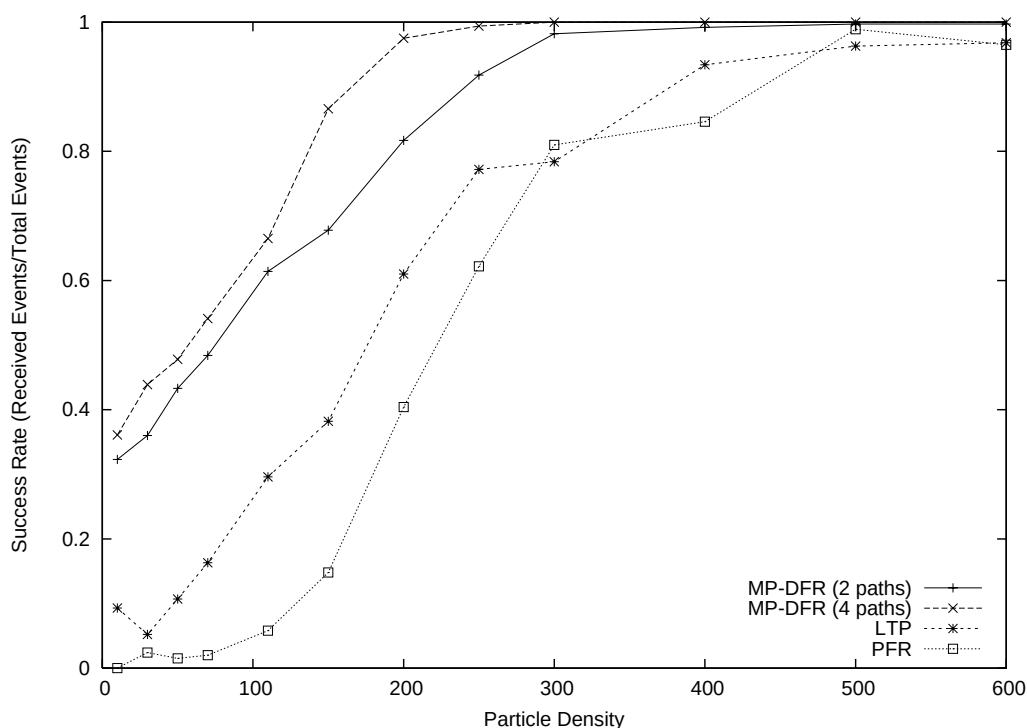
Σχήμα 20.1. : Success rates των MP-DFR και MP-DFRa σε ιδανικό δίκτυο.

Το πρώτο συμπέρασμα που προκύπτει από τη παραπάνω γραφική παράσταση (και δεν αφορά τη σύγκριση των παραλλαγών), είναι ότι τα 4 ανεξάρτητα μονοπάτια που δημιουργούνται δίνουν σαφώς καλύτερα αποτελέσματα από τα 2, και φτάνουν στο μέγιστο δυνατό success rate (που είναι η μονάδα) όταν η πυκνότητα του δικτύου είναι μόλις 300 particles. Να σημειώσουμε εδώ ότι το πειραματικό δίκτυο είναι απόλυτα λειτουργικό (δεν έχουμε καταστροφή κόμβων), και επίσης ότι τα μονοπάτια επιλέγονται με τη σειρά που προστάζει το πρωτόκολλο (δηλαδή όλα καλύπτονται από το p_{level_1} , αφού είναι το πολύ τέσσερα).

Όσον αφορά τη σύγκριση των παραλλαγών, το MP-DFR και για τα 2, αλλά και για τα 4 μονοπάτια φαίνεται να είναι λίγο καλύτερο. Για αυτό το λόγο, στα επόμενα πειράματα θα χρησιμοποιήσουμε την MP-DFR παραλλαγή.

20.2 MP-DFR vs LTP vs PFR ΣΕ ΙΔΑΝΙΚΟ ΔΙΚΤΥΟ

Σε αυτή παράγραφο θα συγκρίνουμε τα success rates ως προς τον αριθμό των κόμβων, των πρωτοκόλλων MP-DFR (για 2 και 4 μονοπάτια), LTP και PFR. Να σημειώσουμε επίσης ότι το πειραματικό δίκτυο είναι απόλυτα λειτουργικό (δεν έχουμε καταστροφή κόμβων). Το σχήμα 20.2 δείχνει το αποτέλεσμα αυτής της σύγκρισης.



Σχήμα 20.2. : Success rates των MP-DFR (2 και 4 μονοπάτια), LTP και PFR σε ιδανικό δίκτυο.

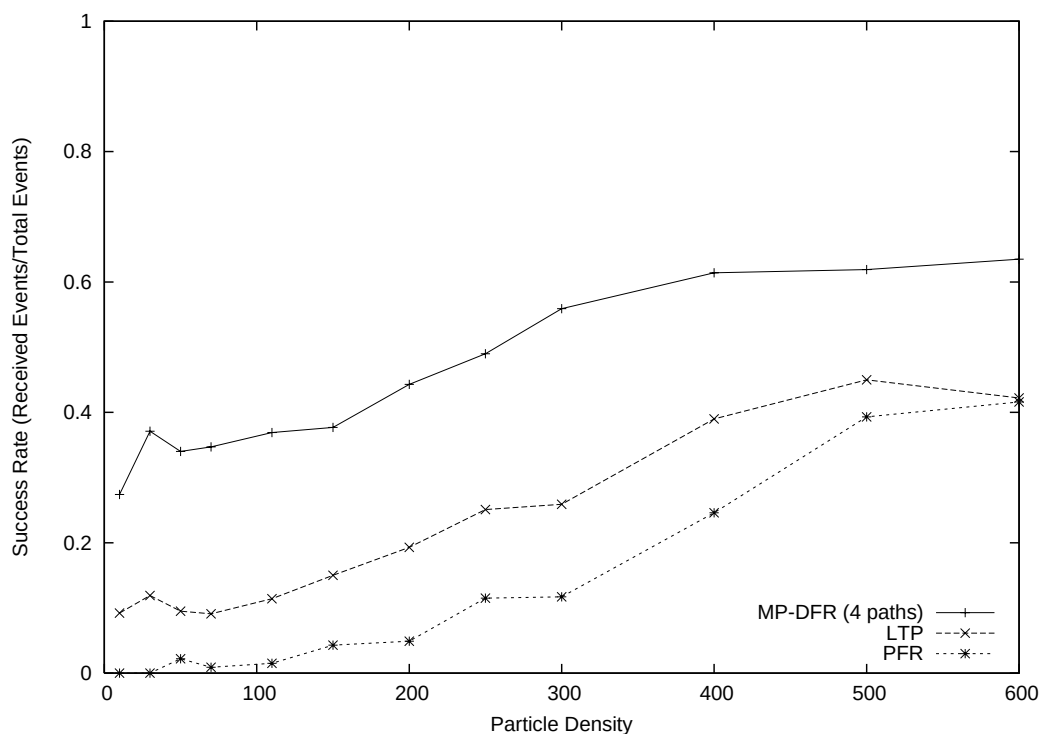
Η παραπάνω γραφική παράσταση είναι η πρώτη απόδειξη ότι το MP-DFR βασιζόμενο στο μοντέλο των πολλαπλών sinks, δίνει πολύ μεγαλύτερο success rate, ακόμα και όταν δημιουργεί μόνο δύο μονοπάτια. Ακόμα και όταν το δίκτυο είναι πάρα πολύ αραιό, δηλαδή όταν έχουμε 10 κόμβους, το MP-DFR με 4 μονοπάτια δίνει success rate περίπου 0.38, ενώ το LTP δίνει μόλις 0.1 και το PFR 0. Επίσης το MP-DFR με 4 μονοπάτια έχει φτάσει σε success rate ίσο με τη μονάδα, όταν η πυκνότητα του δικτύου είναι 300 κόμβοι, ενώ τα LTP, PFR δεν καταφέρνουν αυτήν την απόδοση, ακόμα και όταν υπάρχουν 600 κόμβοι στο δίκτυο, δηλαδή αυτό είναι εξαιρετικά πυκνό. Όσον αφορά την απόδοση LTP και PFR, είναι προφανές ότι το PFR είναι αποδοτικό μόνο σε πυκνό δίκτυο, και σε αυτή τη περίπτωση μόνο είναι καλύτερο από το LTP.

Συμπερασματικά λοιπόν, ακόμα και το MP-DFR 2 μονοπατιών είναι σαφώς αποδοτικότερο από τα LTP και PFR ακόμα και σε ένα αραιό δίκτυο, αποδεικνύοντας έτσι την στιβαρότητα του.

20.3 FAULT TOLERANCE ΓΙΑ ΤΑ MP-DFR, LTP ΚΑΙ PFR

Σε αυτή τη παράγραφο θα συγκρίνουμε τα success rates των MP-DFR (4 μονοπάτια), LTP, PFR, στη περίπτωση όμως που το δίκτυο δεν είναι ιδανικό, αλλά τα particles μπορούν να βγουν εκτός λειτουργίας (π.χ. να εξαντλήσουν την ενέργεια τους, να πάθουν βλάβη κ.α). Δημιουργήσαμε λοιπόν ένα «κακό» δίκτυο όπου κάθε particle μπορεί να βγει εκτός λειτουργίας σε κάθε γύρο του πρωτοκόλλου με πιθανότητα 0.01.

Με αυτό τον τρόπο δημιουργούνται κενά στο δίκτυο. Κάνοντας αυτό το πείραμα, βγάλαμε κάποια σημαντικά συμπεράσματα για το πόσο τα παραπάνω πρωτόκολλα είναι ανεκτικά στα σφάλματα των particles (fault tolerance). Στο σχήμα 20.3 φαίνονται τα αποτελέσματα του πειράματος.

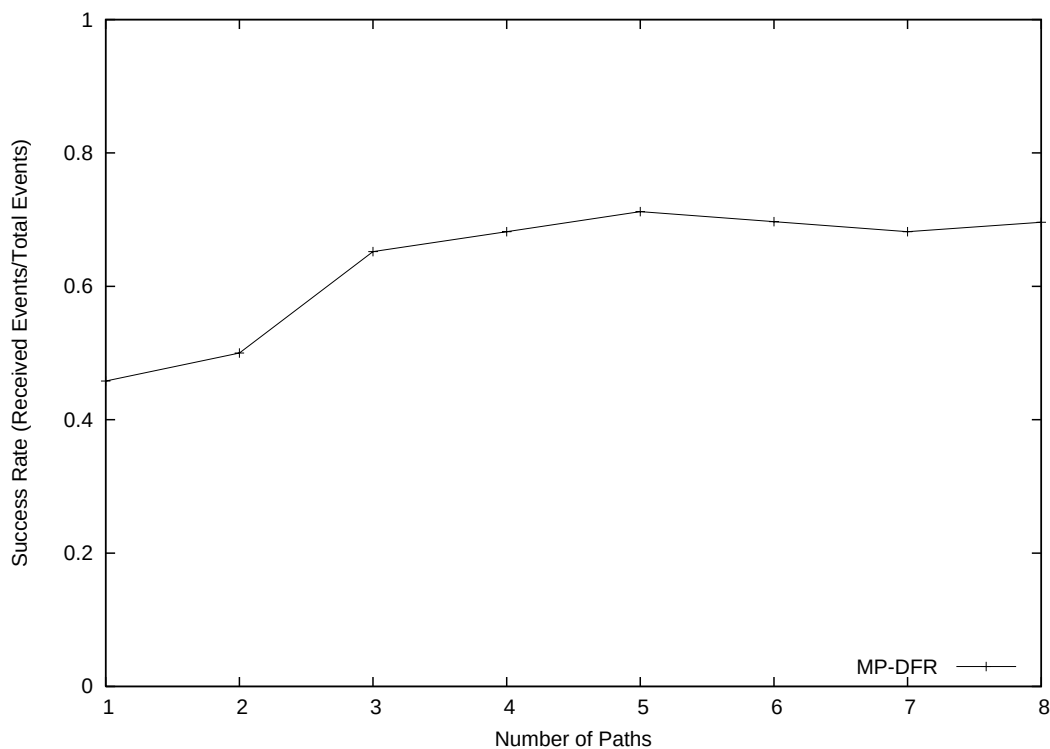


Σχήμα 20.3. : Success rates των MP-DFR (4 μονοπάτια), LTP και PFR σε μη ιδανικό δίκτυο.

Η πρώτη παρατήρηση μας από τη παραπάνω γραφική παράσταση, είναι ότι για όλα τα πρωτόκολλα τα success rates είναι σαφώς μικρότερα, όπως άλλωστε αναμενόταν. Πάλι όμως το MP-DFR φαίνεται να είναι σαφώς αποδοτικότερο και για όλες της διαφορετικές πυκνότητες των particles είναι καλύτερο *τουλάχιστον* κατά 0.2 από το LTP, και πολύ περισσότερο από το PFR. Επίσης, και σε αυτή τη γραφική παράσταση αποδεικνύεται ότι το PFR είναι αποδοτικό μόνο σε πυκνά δίκτυα.

20.4 ΔΗΜΙΟΥΡΓΙΑ ΜΟΝΟΠΑΤΙΩΝ ΣΤΟ MP-DFR

Σε αυτή τη παράγραφο θα μελετήσουμε πώς μεταβάλλεται το success rate του MP-DFR σε σχέση με τον αριθμό των μονοπατιών που κατασκευάζονται. Στο σχήμα λοιπόν 20.4 φαίνεται πως μεταβάλλεται το success rate του MP-DFR σε ιδανικό δίκτυο, για ένα έως οκτώ μονοπάτια (δηλαδή για μονοπάτια που βρίσκονται στα πρώτα δύο path levels). Η πυκνότητα του δικτύου για τα παραπάνω πειράματα είναι 100 particles.



Σχήμα 20.4. : Success rate του MP-DFR σε σχέση με τον αριθμό μονοπατιών.

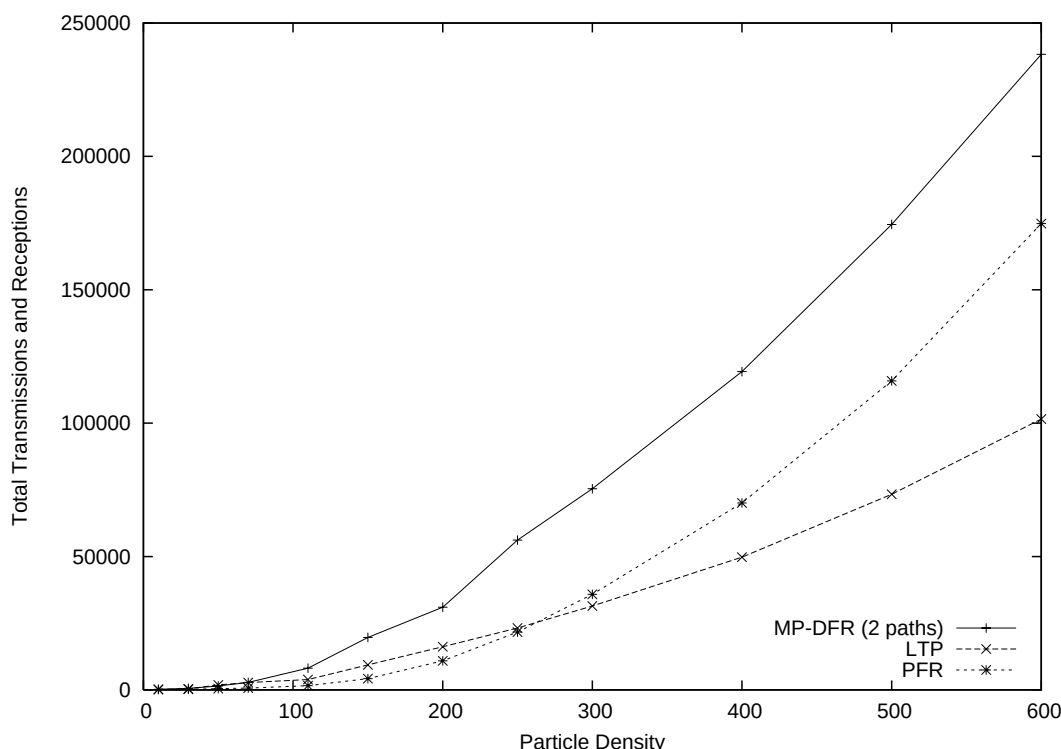
Κατ' αρχήν αν παρατηρήσουμε στο σχήμα 20.2 το LTP για πυκνότητα ίση με 100 particles, θα διαπιστώσουμε ότι παρόλο που διαθέτει backtrack φάση δεν έχει τόσο καλό success rate, όσο το MP-DFR του ενός μονοπατιού. Αυτό οφείλεται στο μοντέλο των πολλαπλών sink που καλύπτει περιμετρικά όλο το δίκτυο.

Όσον αφορά τον αριθμό των μονοπατιών, το success rate φαίνεται να αυξάνει μέχρι τα πέντε μονοπάτια. Από εκεί και έπειτα παρουσιάζει μια μάλλον σταθερή συμπεριφορά. Αυτό είναι λογικό, μιας και το δίκτυο δεν είναι πυκνό (μόνο 100 particles) και τα μονοπάτια του p_{level_2} (από το πέμπτο μονοπάτι μέχρι και το όγδοο) είναι κοντά σε αυτά του p_{level_1} , όπως έχουμε δείξει στην ανάλυση του MP-DFR. Έτσι ουσιαστικά τα γειτονικά μονοπάτια βρίσκουν το ίδιο σύνολο γειτόνων, με αποτέλεσμα να αναπτύσσονται με τον ίδιο τρόπο και έτσι να μην διαφέρει το success rate. Αν αυξήσουμε τη πυκνότητα του δικτύου, η σταθεροποίηση αυτή του success rate γίνεται για λίγο μεγαλύτερο αριθμό μονοπατιών.

20.5 ΦΟΡΤΟΣ ΔΙΚΤΥΟΥ ΣΤΑ MP-DFR, LTP ΚΑΙ PFR

Σε αυτή τη παράγραφο θα αναλύσουμε το πώς κάθε πρωτόκολλο επιβαρύνει το δίκτυο κατά τη διάδοση γεγονότων στον sink. Πιο συγκεκριμένα θα υπολογίσουμε τις συνολικές μεταδόσεις και παραλήψεις μηνυμάτων ως προς την πυκνότητα του δικτύου. Το μέγεθος αυτό είναι ενδεικτικό τόσο επειδή φανερώνει την κατανάλωση ενέργειας στο δίκτυο, αλλά και γενικότερα τον υπολογιστικό φόρτο που έχει κάθε particle. Στο σχήμα 20.5, φαίνεται η γραφική παράσταση που στον άξονα Y έχει τις

συνολικές μεταδόσεις και παραλήψεις μηνυμάτων, στο άξονα X τη πυκνότητα του δικτύου, για τα πρωτόκολλα MP-DFR (δύο μονοπάτια), LTP και PFR.



Σχήμα 20.5. : Φόρτος δικτύου για τα MP-DFR (2 μονοπάτια), LTP, PFR.

Το MP-DFR όπως έχουμε αναφέρει και παραπάνω, εισάγει ένα πλεονασμό για να εξασφαλίσει την επιτυχή μετάδοση των δεδομένων. Αυτό φαίνεται και από τη παραπάνω γραφική παράσταση όπου το MP-DFR φαίνεται να εισάγει μεγαλύτερο φόρτο στο δίκτυο ειδικότερα όταν η πυκνότητα του δικτύου ξεπερνά τους 100 κόμβους. Να σημειώσουμε όμως, ότι στις παραλήψεις μηνυμάτων έχουμε υπολογίσει και αυτές που γίνονται από τους sinks, οι οποίες είναι αρκετές μιας και ένα γεγονός μπορεί να παραληφθεί από περισσότερους του ενός sinks. Οι sinks όμως μιας και είναι κόμβοι θεωρητικά άπειρης ενέργειας και υπολογιστικής ισχύος, δεν επιβαρύνονται από αυτές τις παραλήψεις.

Όσον αφορά τα άλλα δύο πρωτόκολλα, το PFR φαίνεται να επιβαρύνει λιγότερο το δίκτυο, γεγονός το οποίο έχει μελετηθεί και αποδειχτεί στο [11]. Αυτό είναι περισσότερο προφανές όταν η πυκνότητα του δικτύου ξεπερνά τους 300 κόμβους.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel and S. Roundy. *PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking*. IEEE Computer, 33(7): 42-48, 2000.
- [2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. *Wireless sensor networks: a survey*. Journal of Computer Networks, 38:393-422, 2002.
- [3] C. E. Perkins, editor. *Ad Hoc Networking*. Addison-Wesley, Upper Saddle River, NJ, 2001
- [4] R. Amirtharajah and A. P. Chandrakasan, "Self-powered Signal Processing Using Vibration-based Power Generation," IEEE JSSC, pp. 687-695, May 1998.
- [5] S. Meninger, J.O. Mur-Miranda, *et al.*, "Vibration-to-Electric Energy Conversion" IEEE Trans. on VLSI Systems, pp. 64-76, February 2001.
- [6] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, A. Chandrakasan, *Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks*, Proceedings of ACM Mobicom'01, Rome, Italy, July 2001, pp. 272-286.
- [7] Holger Karl, Andreas Willig, *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons Ltd, January 2006.
- [8] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao. *Habitat Monitoring: Application Driver for Wireless Communications Technology*. In Proceedings of the ACM SIGCOM Workshop on Data Communications in Latin America and the Caribbean, San Jose, Costa Rica, 2001.
- [9] "State of the art report for Communication and Computing in Ad-hoc Mobile Networks", Foundational Aspects of Global Computing (FLAGS), IST/FET Project (IST-2001-33116), Deliverable D3.1, September 2001.
- [10] A. Woo and D. Culler: *A Transmission Control Scheme for Media Access in Sensor Networks*. In Proc. 7th ACM/IEEE International Conference on Mobile Computing, pp 221-235, July 2001.
- [11] I. Chatzigiannakis, T. Dimitriou, S. Nikolettseas, and P. Spirakis, "A Probabilistic Algorithm for Efficient and Robust Data Propagation in Smart Dust Networks", in the Proceedings of the 5th European Wireless Conference on Mobile and Wireless Systems beyond 3G (EW 2004), pp. 344-350, 2004.
- [12] C. Intanagonwiwat, R. Govindan and D. Estrin: *Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks*. In Proc.6th ACM/IEEE International Conference on Mobile Computing – MOBICOM'2000.

- [13] I. Chatzigiannakis, S. Nikolettseas, and P. Spirakis. *Smart dust protocols for local detection and propagation*. In 2nd ACM International Annual Workshop on Principles of Mobile Computing (POMC 2002), pages 9.16, 2002.
- [14] J. Diaz, J. Petit and M. Serna: *Random Scaled Sector Graphs*. FLAGS Technical Report FLAGS-TR-22.
- [15] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan: *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*. In Proc. 33rd Hawaii International Conference on System Sciences – HICSS’2000.
- [16] Budhaditya Deb, Sudeept Bhatnagar and Badri Nath Department of Computer Science, Rutgers University, *A Topology Discovery Algorithm for Sensor Networks with Applications to Network Management*, Technical Report, DCS-TR-441
- [17] Alberto Cerpa and Deborah Estrin, UCLA Computer Science Department, Ascent: *Adaptive Self-Configuring Sensor Network Topologies*, Technical Report UCLA/CSD-TR 01-0009, May 2001.
- [18] S. Singh and C. Raghavendra; PAMAS: *Power Aware Multi-Access protocol with signaling for Ad Hoc Networks*, ACM Computer Communications Review, 1999."
- [19] Mihail L. Sichitiu and Vaidyanathan Ramadurai, "*Localization of Wireless Sensor Networks with a Mobile Beacon*" in Proceedings of the First IEEE Conference on Mobile Ad-hoc and Sensor Systems\ (MASS 2004), (Fort Lauderdale, FL), October 2004.
- [20] L. Kleinrock and J.A. Silvester. *Optimum transmission radii for packet radio networks or why six is a magic number*. In IEEE National Telecommunications Conference, December 1978.
- [21] L. Doherty, L. El Ghaoui, and K. S. J. Pister. *Convex position estimation in wireless sensor networks*. In Proceedings of Infocom 2001, April 2001.
- [22] Nirupama Bulusu, Vladimir Bychkovskiy, Deborah Estrin, and John Heidemann. *Scalable, ad hoc deployable rf-based localization*. In Grace Hopper Celebration of Women in Computing Conference 2002, Vancouver, British Columbia, Canada, October 2002.
- [23] Stephen Fitzpatrick and Lambert Meertens. *Diffusion based localization private communication*, 2004.
- [24] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. *Range-Free Localization Schemes in Large-Scale Sensor Networks*. In Proc. of the Intl. Conference on Mobile Computing and Networking (MOBICOM), September 2003.
- [25] C. Savarese, J. Rabaey, and J. Beutel. *Locationing in distributed ad-hoc wireless sensor networks*. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 2037--2040, Salt Lake City, UT, May 2001.

- [26] Lambert Meertens and Stephen Fitzpatrick. “*The distributed construction of a global coordinate system in a network of static computational nodes from inter-node distances*”, Kestrel Institute Technical Report KES.U.04.04, Kestrel Institute, Palo Alto, 2004.
- [27] Nirupama Bulusu, Deborah Estrin, Lewis Girod and John Heidemann. *Scalable Coordination for wireless sensor networks: Self-Configuring Localization Systems* International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, Lake District, UK, July 2001
- [28] Jonathan Bachrach and Christopher Taylor. “*Localization in Sensor Networks*”, Massachusetts Institute of Technology, Cambridge, MA., 2004
- [29] David Jea, Arun A. Somasundara, Mani B. Srivastava: *Multiple Controlled Mobile Elements (Data Mules) for Data Collection in Sensor Networks*. DCOSS 2005: 244-257
- [30] T. Dimitriou and A. Kalis. *Efficient delivery of information in sensor networks using smart antennas*. In 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS 2004), pages 109.122. Springer-Verlag, 2004. Lecture Notes in Computer Science, LNCS 3121.
- [31] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. *Geographic routing without location information*. In 9th ACM/IEEE Annual International Conference on Mobile Computing (MOBICOM 2003), pages 96.108, San Diego, CA, 2003.
- [32] J. N. Al-Karaki and A. E. Kamal, "Routing techniques in wireless sensor networks: a survey," IEEE Wireless Comm., vol. 11, pp. 6-28, 2004.
- [33] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, “*SPAN: an energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks*”, Wireless Networks, Vol. 8, No. 5, Page(s): 481-494, September 2002.
- [34] I. Chatzigiannakis, P. Spirakis, S. Nikolettseas, “*Efficient and Robust Protocols for Local Detection and Propagation in Smart Dust Networks*”, accepted in the ACM Mobile Networks (MONET) journal Special Issue on Algorithmic Solutions for Wireless, Mobile, Ad Hoc and Sensor Networks, February 2005.
- [35] I. Pefkianakis, I. Chatzigiannakis, S. Nikolettseas. *A Probabilistic Multipath Algorithm for Fault-Tolerant Data Propagation in Multisink Wireless Sensor Networks*. Technical Report RA-CTI No.TR2005/12/01, May 2006.
- [36] S. Bhatnagar, B .Deb, and B. Nath: *Service Differentiation in Sensor Networks*. In Proc. of Wireless Personal Multimedia Communications, (2001)
- [37] A. Manjeshwar and D.P. Agrawal: *TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks*. In Proc. 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile

Computing, satellite workshop of 16th Annual International Parallel & Distributed Processing Symposium -- IPDPS'02.

[38] I. Chatzigiannakis and S. Nikolettseas. *A sleep-awake protocol for information propagation in smart dust networks*. In 3rd International Workshop on Mobile, Ad-hoc and Sensor Networks (WMAN 2003), page 225, 2003. IPDPS Workshops.

[39] D. Ganesan, R. Govindan, S. Shenker, D. Estrin. *Highly-resilient, energy-efficient multipath routing in wireless sensor networks*. POSTER SESSION: Best poster papers from MOBIHOC 2001 Pages: 11 - 25 Year of Publication: 2001

[40] B. Deb, S. Bhatnagar and B. Nath, "*ReInForm: Reliable Information Forwarding using Multiple Paths in Sensor Networks*", appear in Proc. of IEEE LCN, 2003.

[41] H. Dubois-Ferrière, D. Estrin, and T. Stathopoulos. *Efficient and Practical Query Scoping in Sensor Networks*. In IEEE International Conference on Mobile Ad-hoc and Sensor Systems, pages 564-566, 2004.

[42] A. Das and D. Dutta, "*Data Acquisition in Multiple-sink Sensor Networks*", Proceedings of the 2nd international conference on embedded networked sensor systems, pages 271 – 272, 2004

[43] Chih-Yu Lin, Yu-Chee Tseng and Ten H. Lai. "*Message-Efficient In-Network Location Management in a Multi-sink Wireless Sensor Network*", IEEE SUTC2006.

[44] E. Ilker Oyman and Cem Ersoy, "*Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks*", ICC June 2004

[45] Bandyopadhyay, S., Coyle, E.J.: *An energy efficient hierarchical clustering algorithm for wireless sensor networks*. In: Proceedings of the IEEE Conference on Computer Communications (INFOCOM), San Francisco (2003)

ΛΕΞΙΛΟΓΙΟ ΟΡΩΝ – ΑΡΚΤΙΚΟΛΕΞΟ

ΑΓΓΛΙΚΟΙ ΌΡΟΙ

Ακρόνυμο	Επεξήγηση
ADC	Analog/Digital Converter
ARQ	Automatic Repeat ReQuest
CMOS	Complementary Metal Oxide Semiconductor
CPU	Central Processing Unit
DAC	Digital/Analog Converter
DSP	Digital Signal Processor
DVS	Dynamic Voltage Scaling
EEPROM	Electrically Erasable Programmable Read Only Memory
FIFO	First In First Out
HVAC	Humidity Ventilation Air Conditioning
LOS	Line of Sight
LTP	Local Target Protocol
MAC	Medium Access Control
MANET	Mobile Ad Hoc Network
PFR	Probabilistic Forwarding Protocol
PLL	Phase Lock Loop
QoS	Quality of Service
RF	Radio Frequency
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
ToS	Type of Service
WSN	Wireless Sensor Networks