

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ ΠΑΤΡΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δίκτυα έξυπνης σκόνης (SmartDust)

“Τεχνολογική μελέτη υπαρχόντων συστημάτων και σχεδιασμός και ανάλυση πρωτοκόλλων για αποδοτικό εντοπισμό και διάδοση πληροφορίας. Ανάπτυξη και λειτουργία πραγματικού πειραματικού δικτύου. Έμφαση σε πρωτόκολλα διαχείρισης του δικτύου”

Αντωνίου Β. Αθανάσιος, ΑΜ 2079

Επιβλέπων: Λέκτορας Νικολετσέας Σωτήρης

Ιούλιος 2003

Ευχαριστίες

Η διπλωματική αυτή είναι έργο συλλογικής εργασίας γι' αυτό καθ' ολη την έκταση του κειμένου της διπλωματικής δε χρησιμοποιείται πουθενά το πρώτο ενικό πρόσωπο. Αντίθετα προτιμήθηκε η χρήση του πρώτου πληθυντικού καθώς δε όχι μπορούσα σε καμία φάση της διπλωματικής να ξεχωρίσω τον εαυτό μου ως μονάδα και να διεκδικήσω προσωπικούς επαίνους για ατομική εργασία.

Θα ήθελα αρχικά να ευχαριστήσω τον επιβλέποντα λέκτορα κύριο Σωτήρη Νικολετσέα για την εμπιστοσύνη, που μου έδειξε με την ανάθεση αυτής της διπλωματικής. Πρέπει να σημειώσω ότι οι καίριες υποδείξεις και προτροπές του αποδείχτηκαν πολύτιμες για την επιτυχή ολοκλήρωση αυτής της διπλωματικής. Επιπρόσθετα με βοήθησαν να γίνω περισσότερο οργανωμένο άτομο και να διευθετώ καλύτερα τις προτεραιότητες και τους στόχους μου.

Επίσης όχι μόνο να ευχαριστήσω το συνάδελφο Χατζηιαννάκη Ιωάννη, ο οποίος είχε το ρόλο του συνεπιβλέποντα της διπλωματικής, για την σημαντική του συμβολή σε κάθε στάδιο της διπλωματικής. Όντας συνεχώς στο πλευρό μας συμβουλεύοντας, παροτρύνοντας και βοηθώντας ουσιαστικά στην επίτευξη των στόχων μας, αποδείχθηκε σημαντικός συνεργάτης και φίλος.

Οφείλω στη συνέχεια να ευχαριστήσω τον συνεργάτη, συμφοιτητή και φίλο Γιώργο Μυλωνά, στον οποίο ανατέθηκε επίσης διπλωματική με θέμα τα δίκτυα έξυπνης σκόνης και με τον οποίο συνεργαστήκαμε άρτια και αρμονικά καθ' όλη την πορεία της διπλωματικής. Επιδεικνύοντας αξιόλογες οργανωτικές και προγραμματιστικές ικανότητες και συνέπεια στα παραδοτέα της διπλωματικής, η συμβολή του στη διπλωματική αυτή ήταν καθοριστική και πολύτιμη.

Θα κλείσω αυτή την ενότητα ευχαριστώντας την οικογένεια μου για τη συμπαράσταση της σε όλες τις επιλογές που έχω κάνει στη ζωή μου.

Πρόλογος

Τα δίκτυα έξυπνης σκόνης είναι μια σχετικά νέα τεχνολογία δικτύων, η οποία παρουσιάζει μεγάλο ενδιαφέρον και προσφέρει πολλές προκλήσεις για έρευνα. Ήδη είναι πολυάριθμα τα ερευνητικά κέντρα και οι επιστημονικές κοινότητες που ασχολούνται με την έρευνα πάνω στα δίκτυα αυτά, γεγονός που υπόσχεται αλματώδεις προόδους στο άμεσο μέλλον.

Περισσότερα από τριάντα χρόνια έχουν περάσει από τη διατύπωση του νόμου του Moore, στον οποίο ο Moore προέβλεπε ότι το ελάχιστο μέγεθος κυκλώματος με επεξεργαστικές δυνατότητες θα συνεχίζει να μικραίνει. Ο νόμος αυτός ακόμα και σήμερα δεν έχει πάψει να ισχύει και είναι βέβαιο ότι θα συνεχίζει να ισχύει για πολλά ακόμα χρόνια. Άμεση εφαρμογή του νόμου του Moore βρίσκουμε στην υλοποίηση συσκευών που θα στελεχώσουν τα μελλοντικά δίκτυα έξυπνης σκόνης. Στα αμέσως επόμενα χρόνια, λοιπόν, αναμένεται να είναι υλοποιημένες και εμπορικά διαθέσιμες μικροσκοπικές συσκευές με μέγεθος ενός κόκκου άμμου, που θα ενσωματώνουν μηχανισμούς ασύρματης επικοινωνίας, ανίχνευσης του περιβάλλοντος και επεξεργασίας δεδομένων.

Ο πιο βασικός λόγος που τα δίκτυα έξυπνης σκόνης έχουν συγκεντρώσει παγκοσμίως το ερευνητικό ενδιαφέρον είναι οι πολυάριθμες εφαρμογές που μπορούν να έχουν. Για αρκετό καιρό υπήρχε η ανάγκη για ένα αποδοτικό ασύρματο δίκτυο μέσο, που θα προσέφερε δυνατότητες παρακολούθησης δυσπρόσιτων περιοχών, παρατήρησης φυσικών φαινομένων, ανίχνευσης δραστηριοτήτων σε προστατευόμενες ή στρατηγικές τοποθεσίες και αξιόπιστης διάδοσης πληροφοριών σε κέντρα ελέγχου. Τα δίκτυα έξυπνης σκόνης υπόσχονται να καλύψουν την ανάγκη αυτή προσφέροντας επιπρόσθετα πλεονεκτήματα, όπως είναι η αυτόνομη και χωρίς επίβλεψη συγκρότηση και λειτουργία τους και η ευκολία προσαρμογής τους σε οποιαδήποτε μορφολογική ή άλλη ιδιαιτερότητα περιβάλλοντος.

Η έρευνα στα δίκτυα έξυπνης σκόνης ήδη επιδεικνύει αξιοσημείωτα επιτεύγματα τόσο στον τομέα των εξειδικευμένων πρωτοκόλλων όσο και στην υλοποίηση πρότυπων συσκευών, οι οποίες μπορούν να συγκροτήσουν τα πρώτα πραγματικά ασύρματα δίκτυα έξυπνης σκόνης. Όσον αφορά τις σύγχρονα εμπορικά διαθέσιμες συσκευές, που προορίζονται για δίκτυα έξυπνης σκόνης, αυτές έχουν διαστάσεις που κυμαίνονται από μερικά κυβικά εκατοστά μέχρι αρκετά κυβικά χιλιοστά. Αυτές οι διαστάσεις θεωρούνται ακόμα σχετικά μεγάλες για κόμβους έξυπνης σκόνης και μπορεί κανείς να φανταστεί πόσο μικροσκοπικές και απαρατήρητες θα είναι οι πραγματικές συσκευές που θα ανήκουν στα μελλοντικά δίκτυα έξυπνης σκόνης.

Στη διπλωματική αυτή πραγματοποιήθηκε μελέτη ενός μεγάλου αριθμού πρωτοκόλλων για δίκτυα έξυπνης σκόνης. Τα πρωτόκολλα αυτά λόγω της εξειδικευμένης φύσης των δικτύων στα οποία αναφέρονται, παρουσιάζουν πολλές ιδιαιτερότητες και εφαρμόζουν πρωτοποριακές ιδέες, ώστε να χειρίζονται αποδοτικά τους διαθέσιμους πόρους και να επιλύουν τα διάφορα ζητήματα που προκύπτουν. Επιλέξαμε να παρουσιάσουμε ένα αντιπροσωπευτικό δείγμα από τα πρωτόκολλα αυτά, με στόχο να σχηματιστεί μια καθαρή εικόνα για τη δομή και λειτουργία που οφείλει να έχει κάθε πρωτόκολλο, το οποίο απευθύνεται σε δίκτυα έξυπνης σκόνης, και τα χαρακτηριστικά που μπορεί ή πρέπει να ενσωματώνει για να θεωρείται αποδοτικό και κατάλληλο για πρακτικές εφαρμογές.

Η δική μας συνεισφορά στην έρευνα πάνω στα δίκτυα έξυπνης σκόνης είναι κατά πρώτο λόγο η υλοποίηση ενός εξομοιωτή πρωτοκόλλων για τα δίκτυα αυτά. Ο εξομοιωτής αυ-

τός είναι γραμμένος σε C++, εκμεταλλεύεται τις αξιόπιστες δομές της βιβλιοθήκης LEDA και μπορεί να διαχειρίζεται και να εξομοιώνει αποδοτικά δίκτυα μεγάλων διαστάσεων, χωρίς μεγάλες καθυστερήσεις ή υπέρογκες απαιτήσεις σε μνήμη και υπολογιστική ισχύ. Από όσο μπορούμε να γνωρίζουμε, για πρώτη φορά είναι εφικτή η εφαρμογή πρωτοκόλλων σε εξομοιούμενα δίκτυα, τα οποία αναπτύσσονται σε εικονικούς χώρους μεγάλων διαστάσεων μέχρι και $1500 \text{ m} \times 1500 \text{ m}$. Οι υπάρχοντες εξομοιωτές, όπως ο ns2, είναι αναποτελεσματικοί στην εξομοίωση δικτύων που οι διαστάσεις τους υπερβαίνουν τα $200 \text{ m} \times 200 \text{ m}$, καθώς απαιτούν ισχυρό επεξεργαστή και αρκετή διαθέσιμη φυσική μνήμη στον Η/Υ, και παρουσιάζουν απαράδεκτα μεγάλη καθυστέρηση μέχρι να εξάγουν τα αποτελέσματα της εξομοίωσης.

Στον εξομοιωτή μας υλοποιήσαμε δύο πρωτόκολλα για τα δίκτυα έξυπνης σκόνης, ήτοι τα PFR και TEEN. Η εξομοίωση αυτών των πρωτοκόλλων μας βοήθησε να κατανοήσουμε σε βάθος τη λειτουργία τους και να μπορέσουμε να προτείνουμε για αυτά κάποιες νέες και πιο αποδοτικές εκδοχές τους. Συγκεκριμένα για το PFR, προτείναμε μια νέα εκδοχή του, το SW-PFR, όπου το δίκτυο ενδυναμώνει μηχανισμούς εξοικονόμησης ενέργειας και μηχανισμούς διαχείρισης πολλαπλών διαδιδόμενων συμβάντων. Για το TEEN, υλοποιήσαμε μια παραλλαγή του, το H-TEEN, όπου το δίκτυο οργανώνεται σε περισσότερα από ένα ιεραρχικά επίπεδα για τη διάδοση της πληροφορίας. Προχωρήσαμε στη συγχριτική αξιολόγηση των δύο αυτών πρωτοκόλλων, παρατηρώντας τις επιδόσεις τους για διάφορους σχηματισμούς δικτύων, δοκιμάζοντας διάφορες τιμές για τις διαστάσεις, την πυκνότητα σε κόμβους και άλλα χαρακτηριστικά των δικτύων. Αποφασίσαμε να περιγράψουμε τη διαδικασία και τα αποτελέσματα αυτής της αξιολόγησης σε μια ζεχωριστή ερευνητική εργασία (paper), η οποία είναι προσκελλημένη εργασία (invited paper) στο IEEE MobiWac'03 (International Mobility and Wireless Access Workshop).

Άλλη συνεισφορά της διπλωματικής εργασίας μας αποτελεί και η υλοποίηση ενός οπτικού προσομοιωτή, η οποία έγινε παράλληλα με την υλοποίηση του εξομοιωτή μας. Αυτός ο προσομοιωτής παρέχει ένα φιλικό περιβάλλον διεπαφής προς το μέσο χρήστη και αναπαριστά με ακρίβεια τη λειτουργία του εξομοιωτή μας, υποδεικνύοντας παραστατικά και με ευκολονόητο τρόπο όλα τα βήματα της εξομοίωσης. Η οπτικοποίηση των πρωτοκόλλων είναι ένα αρκετά εντυπωσιακό θέμα, που δίνει μια εποπτική αίσθηση για τον τρόπο λειτουργίας των πραγματικών δικτύων έξυπνης σκόνης. Ο οπτικός προσομοιωτής μας βοήθησε να επικυρώσουμε την ορθή λειτουργία τόσο των μηχανισμών που ενσωματώνει ο εξομοιωτής (πχ SLEEP-AWAKE) όσο και των πρωτοκόλλων που υλοποιήσαμε. Ελπίζουμε ότι το εργαλείο αυτό θα μπορέσει να χρησιμοποιηθεί για εκπαιδευτικούς σκοπούς ή γενικότερα σκοπούς επίδειξης της λειτουργίας των δικτύων έξυπνης σκόνης.

Η διπλωματική αυτή έχει να παρουσιάσει προόδους και στην ανάπτυξη ενός πραγματικού πειραματικού δικτύου έξυπνης σκόνης. Έχοντας διαθέσιμο έναν αριθμό από τις πρότυπες συσκευές έξυπνης σκόνης με την ονομασία MICA motes, ήμαστε σε θέση σε πρώτο στάδιο να μελετήσουμε τις δυνατότητες τους και έπειτα να δημιουργήσουμε με αυτές ένα πραγματικό πειραματικό δίκτυο. Για την αξιολόγηση των πρότυπων συσκευών MICA motes, εξετάσαμε τις δυνατότητες που έχει η κάθιση συσκευής για ανίχνευση περιβαλλοντικών φαινομένων, όπως επίπεδα φωτεινότητας ή αλλαγές στη θερμοκρασία, και επίσης ελέγχαμε τις δυνατότητες επικοινωνίας και συνεργασίας μεταξύ τους για τη διάδοση μιας πληροφορίας. Καθώς τα MICA motes είναι επαναπρογραμματιζόμενα, μπορέσαμε να υλοποιήσουμε διάφορες εφαρμογές και να προγραμματίσουμε με αυτές τα motes. Ως αποτέλεσμα αυτού, έγινε εφικτή η πραγμάτωση ενός πρότυπου πειραματικού δικτύου έξυπνης σκόνης, το οποίο μπορεί και αναλαμβάνει τη διάδοση μηνυμάτων από κάποιους κόμβους - πηγές σε ένα κόμβο που αποτελεί εικονικό

κέντρο ελέγχου. Ο τελευταίος κόμβος μάλιστα, προωθεί κάθε λαμβανόμενη πληροφορία σε H/Y, όπου έχουμε τη δυνατότητα παραπέρα επεξεργασίας της ή και απεικόνισης της με γραφικό τρόπο.

Η τεχνολογία των δικτύων έξυπνης σκόνης είναι ακόμα σε πρώιμο στάδιο ανάπτυξης και αυτό δημιουργεί πρόσφορο έδαφος για έρευνα και συνεισφορά από κάθε ερευνητική κοινότητα. Η συνεισφορά μας στη διπλωματική αυτή είναι ακόμα ένα μικρό βήμα για την έρευνα στα δίκτυα αυτά. Ελπίζουμε το πόνημα μας να αποτελέσει χρήσιμο βοήθημα και οδηγό για την μελλοντική έρευνα σε ένα κλάδο δικτύων, τα οποία ήδη διαφαίνεται ότι θα αποτελέσουν μια από τις κυριαρχείς τεχνολογίες του μέλλοντος.

Περιεχόμενα

I Πρωτόκολλα επικοινωνίας	1
1 Εισαγωγή στα δίκτυα έξυπνης σκόνης	2
1.1 Διαφορές Ad hoc δικτύων και Smart Dust δικτύων	4
1.2 Καθοριστικοί παράγοντες για τη σχεδίαση των δικτύων έξυπνης σκόνης	5
1.2.1 Αντοχή σε σφάλματα	5
1.2.2 Τοπολογία και κλίμακα μεγέθους	6
1.2.3 Ανάπτυξη των δικτύων (Deployment)	7
1.2.4 Τεχνικές προδιαγραφές των κόμβων Smart Dust	8
1.2.5 Κατανάλωση ενέργειας	9
1.2.6 Μέσο μεταδόσεων	10
1.2.7 Κόστος παραγωγής	13
1.3 Κίνητρα και Εφαρμογές στη χρήση των Smart Dust δικτύων	14
1.3.1 Εφαρμογές των δικτύων Smart Dust	14
1.4 Το πρόβλημα	16
1.5 Σύγχρονα επιτεύγματα και η δική μας συνεισφορά	17
1.5.1 WEBS	17
1.5.2 μAMPS	19
1.5.3 WINS	20
1.5.4 SENSONET	20
1.5.5 SCADDSS	21
1.5.6 Συνεισφορά για την έρευνα σε Smart Dust δίκτυα	21
2 Προτεινόμενα πρωτόκολλα δικτύων έξυπνης σκόνης	23
2.1 Εισαγωγή στα πρωτόκολλα	23
2.2 Το μοντέλο των δικτύων	25
2.2.1 Μοντέλο πηγής - Sink	26

2.2.2	Μοντέλο πηγής και Τείχους	27
2.2.3	Μοντέλο κατανομής LATTICE	28
2.2.4	Μοντέλο κατανάλωσης ενέργειας	29
2.3	Ανασκόπηση χαρακτηριστικών πρωτοκόλλων	30
2.3.1	Flooding	30
2.3.2	Gossiping	31
2.3.3	Greedy Directed Diffusion	31
2.3.4	LTP	37
2.3.5	MIN 2	42
3	Το πρωτόκολλο PFR και η τροποποίηση μας	44
3.1	SW-PFR	47
4	Το πρωτόκολλο TEEN και η τροποποίηση μας	49
4.1	H-TEEN	52
II	Εξομοίωση πρωτοκόλλων	54
5	Περιβάλλον εξομοίωσης	55
5.1	Βασικά χαρακτηριστικά του εξομοιωτή	55
5.1.1	Λειτουργία σε γύρους	56
5.1.2	MAC ζητήματα	56
5.1.3	Λειτουργία SLEEP - AWAKE	56
5.1.4	Κατανάλωση ενέργειας	57
5.1.5	Μέγεθος των μηνυμάτων	58
5.2	Η δομή του εξομοιωτή	58
5.3	Η λειτουργία του εξομοιωτή	59
5.4	Η λειτουργία της οπτικοποίησης	62
5.5	Μελλοντική δουλειά και προκλήσεις	64
6	Εξομοίωση δύο πρωτοκόλλων	65
6.1	Λεπτομέρειες υλοποίησης του SW-PFR	65
6.2	Λεπτομέρειες υλοποίησης του H-TEEN	67
7	Σύγκριση μεταξύ H-TEEN και SW-PFR	70
7.1	Μετρικές αξιολόγησης	70

7.2 Πειραματικές μετρήσεις και συμπεράσματα	71
III Ανάπτυξη πραγματικού πειραματικού δικτύου έξυπνης σκόνης	79
8 Εξέλιξη των συσκευών για δίκτυα αισθητήρων	80
8.1 Συσκευές Smart Dust	80
8.2 Συσκευές COTS dust	82
8.2.1 MICA motes	84
8.2.2 Κυκλώματα αισθητήρων	87
9 Προγραμματιστικό περιβάλλον και εφαρμογές	89
9.1 TinyOS	90
9.2 NesC	93
9.3 Μια πρότυπη εφαρμογή σε NesC	95
10 Μετρήσεις και εφαρμογές δικτύωσης των MICA Motes	101
10.1 Προγραμματισμός των MICA motes	101
10.1.1 Ρύθμιση ακτίνας εκπομπής	104
10.1.2 Δομή μηνυμάτων στο TinyOS και η εφαρμογή OscilloscopeRf	106
10.1.3 Διάρκεια ζωής ενός MICA mote	109
10.1.4 Δύο εφαρμογές multihop	112
10.4.1 Εφαρμογή multihop για το CntToLedsAndRfm	113
10.4.2 Εφαρμογή multihop για το OscilloscopeRf	116
10.1.5 TinyDB	120
11 Επίλογος	125

Μέρος Ι

Πρωτόχολλα επικοινωνίας

Κεφάλαιο 1

Εισαγωγή στα δίκτυα έξυπνης σκόνης

“If I have seen further than others, it is by standing upon the shoulders of giants”, Isaac Newton

Ένα από τα σπουδαιότερα επιτεύγματα της τεχνολογίας, τα ασύρματα δίκτυα, έχουν εισβάλλει δυναμικά στη σύγχρονη καθημερινότητα με ένα ευρύ πεδίο εφαρμογών. Η έλλειψη καλωδίωσης στα ασύρματα δίκτυα ευνοεί την εύκολη δημιουργία τους (deployment) και έτσι τα δίκτυα αυτά τείνουν να αντικαταστήσουν τα παροδοσιακά καλωδιωμένα δίκτυα σχεδόν σε κάθε εφαρμογή τους. Από πολύ νωρίς ωστόσο, παρατηρήθηκε ότι υπάρχουν εφαρμογές, οι οποίες απαιτούν ειδικής φύσης ασύρματα δίκτυα, τέτοια ώστε με ελάχιστη προεργασία για την εγκατάσταση τους να μπορούν να δημιουργούνται και να συντηρούνται για μεγάλο χρονικό διάστημα. Για τις εφαρμογές αυτές αναπτύχθηκε ο χλάδος των Ad hoc δικτύων, που είναι ειδικής φύσης ασύρματα δίκτυα με την πρωτοποριακή ικανότητα να οργανώνονται απουσία σταθερής υποδομής.

Και πάλι όμως η ιδιαιτερότητα κάποιων εφαρμογών δημιούργησε την ανάγκη για ένα περισσότερο εξειδικευμένο τύπο δικτύων που αν και θα είχε πολλά παρόμοια χαρακτηριστικά με τα Ad hoc δίκτυα, ουσιαστικά θα αποτελούσε ξεχωριστό είδος δικτύων προοριζόμενο για εφαρμογές ανίχνευσης και παρακολούθησης. Το νέο αυτό είδος δικτύων, το οποίο γρήγορα συγκέντρωσε το ερευνητικό ενδιαφέρον και σήμερα παρουσιάζει αλματώδεις προόδους, είναι τα δίκτυα αισθητήρων ή αλλιώς δίκτυα έξυπνης σκόνης (Smart Dust).

Τα δίκτυα Smart Dust αποτελούν ένα χλάδο δικτύων ασύρματης επικοινωνίας, όπου οι κόμβοι είναι εξοπλισμένοι με αισθητήρες, ώστε να μπορούν να παρακολουθούν φαινόμενα του περιβάλλοντος στο οποίο αναπτύσσονται. Οι τεχνολογίες της μικρο-ηλεκτρο-μηχανικής (MEMS), των ψηφιακών ηλεκτρονικών και των ασύρματων επικοινωνιών έχουν φτάσει σε τέτοια επίπεδα ωριμότητας, ώστε να είναι δυνατή η κατασκευή συσκευών με αισθητήρες που μπορούν να αποτελούν κόμβους πρότυπων πραγματικών Smart Dust δικτύων.

Στα Smart Dust δίκτυα ή δίκτυα έξυπνης σκόνης, το οξύμωρο όνομα είναι στην ουσία ενδεικτικό για το μέγεθος, την υπολογιστική ικανότητα, το κόστος και τον τρόπο ανάπτυξης στο χώρο των κόμβων που στελεχώνουν το δίκτυο.

Υπάρχουν αρκετές ιδιαιτερότητες των δικτύων έξυπνης σκόνης που δικαιολογούν τον όρο “σκόνη” στην ονομασία τους. Πρώτα, είναι το μέγεθος των κόμβων Smart Dust το οπόιο

είναι μικροσκοπικό, της τάξης μερικών κυβικών χιλιοστών, ώστε ο κάθε κόμβος να μοιάζει με κόκκινο σκόνης. Έπειτα είναι η ανάπτυξη (deployment) των κόμβων στο χώρο, που κατά κανόνα δεν είναι προκαθορισμένη αλλά τυχαία. Οι κόμβοι “διασκορπίζονται”, επικάθονται σε επιφάνειες και πρέπει κατόπιν να συνεργαστούν για να δημιουργήσουν ένα δίκτυο. Τέλος ο όρος “σκόνη”, υπό μια έννοια, είναι ενδεικτικός και για το κόστος παραγωγής αυτών των συσκευών, που πρέπει να είναι αρκετά χαμηλό, ώστε να δικαιολογεί την παραγωγή τους σε μεγάλες ποσότητες.

Ο χαρακτηρισμός των κόμβων ως “έξυπνων” δεν γίνεται χωρίς καλή αιτιολόγηση. Οι κόμβοι ενσωματώνουν υπολογιστική ικανότητα και πολλαπλά είδη αισθητήρων, ώστε να έχουν πολύ-πλευρη αντίληψη του περιβάλλοντος τους και να επεξεργάζονται οποιεσδήποτε πληροφορίες συλλέγουν. Έτσι μπορούν να προωθούν μόνο ό,τι είναι απαραίτητο να διαδοθεί. Επιπρόσθετα τα δίκτυα που συγκροτούν οι έξυπνοι κόμβοι είναι κατανεμημένης φύσης. Εκτελώντας υπολογισμούς και λειτουργίες σε συνεργασία μεταξύ τους, περιορίζεται η ενεργειακή κατανάλωση και αντισταθμίζεται ο περιορισμός των πεπερασμένων υπολογιστικών και ενεργειακών πόρων που έχει ο κάθε κόμβος ξεχωριστά. Η ιδέα της συνεργασίας τόσο μικρών συσκευών με πεπερασμένες υπολογιστικές και επικοινωνιακές δυνατότητες για αυτόνομη δημιουργία δικτύου συνιστά την κεντρική πρόκληση των δικτύων Smart Dust, πάνω στην οποία εργάζονται πολλές ερευνητικές ομάδες.

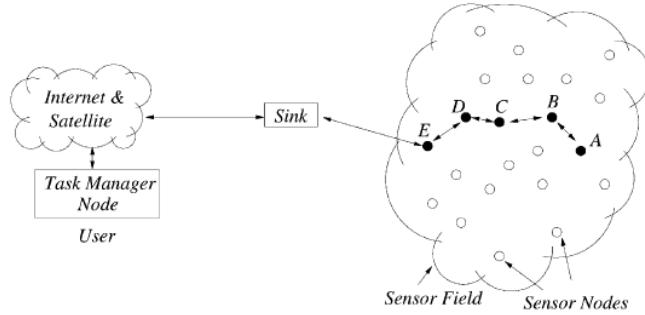
Τα Smart Dust δίκτυα αποτελούν στην ουσία ένα τεχνολογικό βήμα που έχει απώτερο σκοπό να αντικαταστήσει τη χρήση παραδοσιακών ηλεκτρονικών αισθητήρων. Οι ηλεκτρονικοί αισθητήρες συνήθως εξυπηρετούν εφαρμογές παρακολούθησης της εξέλιξης ενός φαινομένου και μπορεί να είναι τοποθετημένοι μακριά από το χώρο, όπου διαδραματίζεται το φαινόμενο. Ειδικά στην τελευταία αυτή περίπτωση της παρακολούθησης του φαινομένου εκ των μακρόθεν, χρησιμοποιούνται ειδικές τεχνικές διαχωρισμού σημάτων, τα οποία αναφέρονται στο παρατηρούμενο φαινόμενο, από τον αναπόφευκτο ύδρυβο που προκαλούν διάφοροι περιβαλλοντολογικοί παράγοντες.

Γενικά οι ηλεκτρονικοί αισθητήρες έχουν ως αποκλειστική λειτουργία τη συνεχή παρακολούθηση και μέτρηση ενός φαινομένου και τη διαρκή μετάδοση πληροφορίας σε κεντρικούς κόμβους. Στη διαδικασία αυτή δε παρεμβάλλεται επεξεργασία της πληροφορίας, ή κάποια έξυπνη δρομολόγηση της. Επιπρόσθετα οι παραδοσιακοί αισθητήρες θα πρέπει να τοποθετούνται στο χώρο διατηρώντας μια ειδικά προμελετημένη διάταξη, η οποία θα εξασφαλίζει την απρόσκοπτη διεξαγωγή των πιθανών μεταδόσεων και λήψεων πληροφορίας.

Από την άλλη πλευρά, ένα δίκτυο Smart Dust αποτελείται από ένα μεγάλο σχετικά αριθμό κόμβων - αισθητήρων, οι οποίοι συνήθως αναπτύσσονται με σχετικά μεγάλη πυκνότητα κοντά ή ακριβώς πάνω στο παρατηρούμενο φαινόμενο. Δε χρειάζεται να προκαθορίζονται οι θέσεις των κόμβων στο δίκτυο ούτε να προηγείται μια φάση μελέτης της τοπολογίας, που θα αποκτήσει το δικτύου. Κατά κανόνα οι κόμβοι έχουν τυχαία διάταξη στο χώρο και μπορούν να συνεργάζονται για τη δημιουργία αυτόνομων δικτύων, ακόμα και σε πολύ αφιλόξενα περιβάλλοντα. Τέτοια περιβάλλοντα μπορεί να είναι περιοχές δύσκολα προσπελάσιμες ή περιοχές όπου ενέχει κάποιος σοβαρός κίνδυνος και δεν είναι δυνατή η παραμονή ανθρώπινου προσωπικού, παρά ίσως για πολύ μικρό χρονικό διάστημα.

Εκτός από τη δυνατότητα να αυτό-οργανώνονται σε δίκτυα, οι κόμβοι των Smart Dust δικτύων διαθέτουν πομποδέκτες, ώστε να συνεργάζονται με τους υπόλοιπους κόμβους στη διάδοση κρίσιμων πληροφοριών στο δίκτυο. Ουσιαστικά ο κάθε κόμβος δε προορίζεται για αυτόνομη και ανεξάρτητη χρήση, καθώς διαθέτει περιορισμένες δυνατότητες ανίχνευσης, με-

τάδοσης πληροφορίας και πεπερασμένη διάρκεια ζωής. Η αξία των Smart Dust κόμβων έγκειται στην ικανότητα τους να εκτελούν κατανευμημένα πρωτόκολλα επικοινωνίας και διάδοσης πληροφορίας.



Σχήμα 1.1: Παράδειγμα ενός Smart Dust δικτύου

Στο [1, 9] παρουσιάζεται ένα ευρύ φάσμα εφαρμογών για τα δίκτυα Smart Dust, το οποίο αποτελεί μόνο μια ένδειξη της μεγάλης απήχησης που θα έχουν στο άμεσο μέλλον τα δίκτυα αυτά. Οι εφαρμογές άπτονται τομείς όπως υγεία, ασφάλεια, οικολογία και στρατιωτικές επιχειρήσεις. Μπορεί να κανείς να υποθέσει, ότι μελλοντικά τα δίκτυα έξυπνης σκόνης θα είναι αναπόσπαστο μέρος της καθημερινότητας μας.

1.1 Διαφορές Ad hoc δικτύων και Smart Dust δικτύων

Επυμολογικά Ad hoc είναι όλα τα δίκτυα που σχηματίζονται “αυτόματα”, απλά και μόνο με τη συνεργασία των κόμβων που τα απαρτίζουν, χωρίς την προϋπόθεση σταθερής δικτυακής υποδομής ή την εκ των προτέρων σχεδίαση της διάταξης των κόμβων στο χώρο. Ουσιαστικά λοιπόν, δεν απαιτούν να προγνωθεί κάποια ειδική μελέτη για τη δημιουργία, τη λειτουργία και τη συντήρηση τους. Τα παραδοσιακά Ad hoc δίκτυα αποτελούν για αρκετό καιρό αντικείμενο έρευνας και ήδη έχουν προταθεί ολοκληρωμένα πρωτόκολλα επικοινωνίας ειδικά για αυτά τα δίκτυα [17].

Θα μπορούσε κάποιος να θεωρήσει ότι τα Smart Dust αποτελούν μια υποκατηγορία των Ad hoc δικτύων. Αυτό είναι εν μέρει σωστό, γιατί τα Smart Dust επιδεικνύουν τις βασικές ιδιότητες των Ad hoc δικτύων. Όμως επιπρόσθετα παρουσιάζουν ιδιαιτερότητες και θέτουν σοβαρούς περιορισμούς, που τα διαχωρίζουν από τα παραδοσιακά Ad hoc δίκτυα. Κάποιες από τις ειδοποιούς διαφορές των δύο κλάδων δικτύων είναι :

1. Στα Smart Dust δίκτυα το πλήθος των κόμβων είναι μερικές τάξεις μεγέθους μεγαλύτερο από το αντίστοιχο πλήθος κόμβων στα Ad hoc δίκτυα.
2. Οι κόμβοι - αισθητήρες στα Smart Dust δίκτυα διατάσσονται με αρκετά μεγάλη πυκνότητα, ενώ η πυκνότητα των κόμβων στα Ad hoc δίκτυα είναι σχετικά μικρότερη.

3. Οι κόμβοι αισθητήρες είναι ευάλωτοι σε βλάβες, που μπορούν να τους καταστήσουν μόνιμα ανενεργούς. Στα Ad hoc δίκτυα υπάρχει δυνατότητα ανθρώπινης παρέμβασης και επιδιόρθωσης των ελαττωματικών κόμβων, ενώ για τα Smart Dust δίκτυα δεν προβλέπεται κάτι τέτοιο. Μάλιστα στα δίκτυα Smart Dust πρέπει να εξασφαλίζεται η κατά το δυνατόν απρόσκοπτη μακροχρόνια λειτουργία τους, έχοντας ως δεδομένο ότι κάποιοι κόμβοι θα πάψουν μόνιμα να λειτουργούν.
4. Η τοπολογία ενός δικτύου Smart Dust αλλάζει με μεγάλη συχνότητα.
5. Οι κόμβοι αισθητήρες χρησιμοποιούν κυρίως την τεχνική Broadcast για τις μεταδόσεις ενώ στα Ad hoc δίκτυα οι επικοινωνίες είναι σημείο προς σημείο (point to point).
6. Στα Smart Dust δίκτυα οι κόμβοι έχουν περιορισμένους πόρους σε ενέργεια, μέγεθος μνήμης και υπολογιστική ισχύ.
7. Οι κόμβοι ενός Smart Dust δίκτυου ενδέχεται να μην έχουν καθολικά μοναδική ταυτότητα (ID), κυρίως λόγω του μεγάλου αριθμού των κόμβων και του overhead που θα συνεπαγόταν η μοναδική ταυτοποίηση κάθε κόμβου.

Έτσι σε τελική ανάλυση τα ήδη υπάρχοντα πρωτόκολλα για τα Ad hoc δίκτυα δε μπορούν να εφαρμοστούν στα Smart Dust δίκτυα, τουλάχιστον όχι χωρίς σημαντικές προσθήκες και προσαρμογές στην ειδική αρχιτεκτονική και φιλοσοφία των Smart Dust.

1.2 Καθοριστικοί παράγοντες για τη σχεδίαση των δικτύων έξυπνης σκόνης

Κατά το σχεδιασμό των Smart Dust δικτύων, πρέπει να ληφθούν υπόψη κάποιοι κρίσιμοι παράγοντες όπως η ανοχή σε σφάλματα, η κλιμάκωση του μεγέθους του δικτύου, το κόστος παραγωγής των συσκευών - κόμβων, η τοπολογία του δικτύου, οι περιορισμοί που θέτει το υλικό των συσκευών, τα μέσα μετάδοσης και λήψης πληροφορίας και η κατανάλωση ενέργειας [1]. Ένα πρωτόκολλο που αναφέρεται σε Smart Dust δίκτυα οφείλει να δώσει ιδιαίτερη προσοχή στους παραπάνω παράγοντες για να είναι εφαρμόσιμο στην πράξη.

1.2.1 Αντοχή σε σφάλματα

Τα Smart Dust προορίζονται κατά κύριο λόγο για αυτοτελή λειτουργία χωρίς επίβλεψη μέχρι να εξαντληθούν ενεργειακά οι κόμβοι και να μην είναι εφικτή η διάδοση πληροφορίας. Η λογική αυτή συνεπάγεται ότι αν κάποιος κόμβος παρουσιάσει μηχανική βλάβη, υποστεί φυσική καταστροφή, ή γίνει ανενεργός λόγω της εξάντλησης των ενεργειακών αποθεμάτων του, τότε ο κόμβος ενδέχεται να γίνει μόνιμα “άχρηστος” για το δίκτυο. Σε κάθε περίπτωση το δίκτυο δε θα πρέπει να στηρίζεται στην απρόσκοπη λειτουργία όλων των κόμβων που το απαρτίζουν. Αντίθετα θα πρέπει να υπάρχει πρόνοια, ώστε με ευελιξία να παρακάμπτονται οι προβληματικοί κόμβοι και να αποκαθίσταται η ομαλή λειτουργία του δικτύου για όσο περισσότερο διάστημα είναι εφικτό. Στο [2] η αντοχή σε σφάλματα για τα Smart Dust δίκτυα ταυτίζεται με την αξιοπιστία $R_k(t)$ του δικτύου, δηλαδή την πιθανότητα να μην εμφανιστεί

βλάβη στο δίκτυο σε δεδομένη περίοδο t . Αν υποθέσουμε ότι οι βλάβες που παρουσιάζονται στο δίκτυο ακολουθούν μια Poisson κατανομή, τότε η πιθανότητα να μην υπάρξει βλάβη σε κόμβο του δικτύου κατα το διάστημα $(0,t)$ θα δίνεται από τη σχέση

$$R_k(t) = \exp(-\lambda_k t) \quad (1.1)$$

Όπου λ_k είναι ο ρυθμός βλαβών για ένα κόμβο k , και t είναι η χρονική περίοδος.

Καθώς οι εφαρμογές των δικτύων Smart Dust είναι πολυάριθμες και συνολικά αντιστοιχούν σε ένα άκρως ανομοιογενές φάσμα συνθηκών, είναι συχνά ευκταίο τα πρωτόκολλα να είναι εξειδικευμένα για τις συνθήκες κάθε εφαρμογής. Με τον τρόπο αυτό αποφεύγονται πλεονάζοντες περιορισμοί και αυστηρές απαιτήσεις, που πρέπει να επιβάλλει ένα πρωτόκολλο για το χειρισμό ή την πρόληψη των σφαλμάτων επικοινωνίας. Για παράδειγμα ένα πρωτόκολλο για το Smart Dust δίκτυα σε κλειστούς χώρους, μπορεί να έχει χαλαρές απαιτήσεις για αντοχή σε σφάλματα και απώλειες κόμβων, καθώς στο ελεγχόμενο περιβάλλον των κλειστών χώρων η πιθανότητα για σφάλματα είναι σχετικά μικρή. Αντίθετα, στα πεδία μαχών ή σε εφαρμογές ανίχνευσης και διάδοσης κρίσιμων γεγονότων (πυρκαγιά) πρέπει οπωσδήποτε ένα πρωτόκολλο επικοινωνίας να εξασφαλίζει ότι το δίκτυο θα διατηρεί την αξιόπιστη λειτουργία του παρά τις όποιες καταστροφές κόμβων.

1.2.2 Τοπολογία και κλίμακα μεγέθους

Όσον αφορά την κλίμακα μεγέθους των Smart Dust δικτύων υπολογίζεται ότι τυπικές εφαρμογές θα απαιτούν δίκτυα αποτελούμενα από μερικές εκατοντάδες έως και μερικές χιλιάδες κόμβων. Η πυκνότητα των δικτύων αυτών, δηλαδή ο αριθμός των κόμβων Smart Dust ανά μονάδα επιφάνειας, λαμβάνοντας υπόψη και την περιορισμένη εμβέλεια μετάδοσης των κόμβων, θα είναι σχετικά μεγάλη. Συγκεκριμένα ένας τύπος που περιγράφει την πυκνότητα των δικτύων και αναφέρεται στο [3] είναι ο παρακάτω:

$$\mu(R) = \frac{N\pi R^2}{A} \quad (1.2)$$

Όπου N ο αριθμός των κόμβων στο δίκτυο σε επιφάνεια A και R η ακτίνα μετάδοσης των εκπομπών ενός κόμβου. Εν τέλει το $\mu(R)$ είναι ο μέσος αριθμός κόμβων μέσα στην εμβέλεια μετάδοσης ενός κόμβου Smart Dust.

Τα προτεινόμενα πρωτόκολλα πρέπει να λειτουργούν αποδοτικά σε δίκτυα, τα οποία αποτελούνται από εκατοντάδες κόμβους τοποθετημένους με μεγάλη πυκνότητα. Αυτό σημαίνει ότι πρέπει να εκμεταλλεύονται επαρκώς την πυκνή διάταξη των κόμβων για τη διάδοση πληροφοριών στο δίκτυο, αλλά και να περιορίζουν ή να αποφεύγουν ανεπιθύμητες πλεονάζουσες μεταδόσεις, ατέρμονους βρόχους μεταδόσεων των ίδιων μηνυμάτων, παρεμβολές και συγκρούσεις από τη παράλληλη εκπομπή μηνυμάτων από πολλούς κόμβους σε μικρή περιοχή.

Η τοπολογία των δικτύων αυτής της κλίμακας είναι δύσκολο να διατηρηθεί, ειδικά από τη στιγμή που πολλοί κόμβοι αρχίζουν να παρουσιάζουν βλάβες και να αχρηστεύονται. Μπορούμε να διαχρίνουμε τρεις φάσεις για τη δημιουργία και συντήρηση των Smart Dust δικτύων.

α) Φάση προετοιμασίας και διασποράς των κόμβων Κατά κανόνα οι κόμβοι στα δίκτυα έχουνης σκόνης δεν τοποθετούνται σύμφωνα με κάποιο καλά μελετημένο σχέδιο,

αλλά διασκορπίζονται ή τοποθετούνται με τυχαίο τρόπο σε ένα χώρο. Κλασικές μέθοδοι διασκορπισμού των κόμβων είναι η ρίψη από αεροπλάνο ή η ρίψη μέσω πυραύλου. Ωστόσο υπάρχουν και η περιπτώσεις όπου οι κόμβοι τοποθετούνται χειρωνακτικά από τον άνθρωπο. Αν και αποκλείεται η εις βάθος μελέτη του τρόπου διάταξης των κόμβων, ωστόσο μπορεί να ληφθεί κάποια φροντίδα ώστε ο σχηματισμός των κόμβων στο δίκτυο να παρουσιάζει ευελιξία, αντοχή στα σφάλματα και να ευνοεί την αυτόματη οργάνωση τους σε δίκτυο. Σε κάθε περίπτωση το κόστος εγκατάστασης του δικτύου πρέπει να διατηρείται μικρό.

- β) **Φάση μετά τη διασπορά** Στο χρονικό διάστημα που έπεται της διασποράς, ενδέχεται να συμβούν τοπολογικές αλλαγές λόγω βλαβών σε κόμβους, εμφάνισης εμποδίων, ενεργειακής εξάντλησης κάποιων κόμβων. Επίσης αν και γενικά η φύση των Smart Dust συσκευών είναι στατική, δηλαδή δεν κινούνται αυτόβουλα, υπάρχει σημαντική πιθανότητα αλλαγής θέσης των κόμβων, καθώς λόγω του μικρού μεγέθους τους μπορούν να παρασυρθούν από τον άνεμο, τη βροχή και άλλους παράγοντες. Τα πρωτόκολλα για τα δίκτυα Smart Dust πρέπει εκτός από την αντοχή σε σφάλματα να εξασφαλίζουν την αποκατάσταση - επανοργάνωση του δικτύου ακόμα και μετά από σοβαρές αλλαγές στην τοπολογία του.
- γ) **Φάση επαναδιασποράς κόμβων** Η φάση αυτή είναι προαιρετική και συνήθως λαμβάνει χώρο όταν είναι επιθυμητό να επεκταθεί ο χρόνος ζωής ενός Smart Dust δικτύου. Αυτό που γίνεται σε αυτή τη φάση είναι η αντικατάσταση των προβληματικών κόμβων με νέους και η συμπλήρωση με κόμβους σε περιοχές μη επαρκούς πυκνότητας. Γενικά αυτό προϋποθέτει ότι το δίκτυο μπορεί ανά πάσα στιγμή να προσαρμοστεί σε τέτοιας φύσης αλλαγές, ώστε να μπορεί να εκμεταλλεύεται όλους τους κόμβους (παλιούς και νέους) που θα το απαρτίζουν. Πρέπει να σημειωθεί ότι η φάση αυτή πέρα από προαιρετική συχνά είναι και μη εφικτή, ειδικά όταν η ανάπτυξη των δικτύων γίνεται σε δύσκολα προσπελάσιμες περιοχές ή σε πεδία μαχών.

1.2.3 Ανάπτυξη των δικτύων (Deployment)

Η ανάπτυξη των δικτύων γίνεται, όπως αναφέρθηκε, χωρίς λεπτομερή μελέτη και σχεδίαση της τοποθέτησης των κόμβων σε αυτά. Οι κόμβοι πρέπει να είναι προγραμματισμένοι, ώστε να μπορούν ανταλλάσσοντας μηνύματα μεταξύ τους, να οργανωθούν σε ένα λειτουργικό δίκτυο παρά τις όποιες τυχόν μορφολογικές ή άλλου είδους ιδιαιτερότητες του χώρου ανάπτυξης του δικτύου. Επίσης πρέπει να είναι ειδικά σχεδιασμένοι για μακροχρόνια λειτουργία χωρίς επίβλεψη.

Οι περιοχές, όπου αναπτύσσονται τα δίκτυα, είναι συνήθως πολύ χοντά ή στη μέση του φαινομένου προς παρατήρηση. Η αναλώσιμη φύση των δικτύων έξυπνης σκόνης και το χαμηλό τους κόστους τα καθιστούν ιδανικά ειδικά για περιοχές, όπου ενέχει κάποιος σοβαρός κίνδυνος ή χυριαρχούν ανθυγιεινές συνθήκες ή γενικότερα συνθήκες που καθιστούν αδύνατη την παραμονή ανθρώπινου προσωπικού. Μπορεί λοιπόν να βρίσκονται στο βυθό ενός ωκεανού, σε ένα δάσος υπό πυρκαϊά, σε ένα πεδίο μάχης στο εσωτερικό ενός ανεμοστρόβιλου, ή ακόμα και στην επιφάνεια ενός άλλου πλανήτη. Επίσης τα δίκτυα λόγω του μικρού μεγέθους των κόμβων, μπορούν να έχουν τους κόμβους τους ενσωματωμένους στο εσωτερικό συσκευών και μηχανημάτων ή τοποθετημένους σε αγωγούς και μικρούς χώρους όπου είναι αδύνατη η πρόσβαση από άνθρωπο.

1.2.4 Τεχνικές προδιαγραφές των κόμβων Smart Dust

Οι κόμβοι που προορίζονται για δίκτυα έξυπνης σκόνης, έχουν στόχο να ενσωματώνουν τη μεγαλύτερη δυνατή λειτουργικότητα σε μια συσκευή μικρότερη από κυβικό εκατοστό με το μικρότερο δυνατό κόστος. Μέχρι τώρα, οι περισσότεροι κόμβοι έχουν μέγεθος μερικών κυβικών εκατοστών. Παραδείγματα τέτοιων κόμβων είναι οι Rene, MICA και οι μAMPS. Ωστόσο, έχουν αρχίσει να διατίθενται στην αγορά κόμβοι πολύ μικρότερων διαστάσεων (μικρότερων από ένα κυβικό εκατοστό). Τέτοιοι είναι οι κόμβοι τεχνολογίας MICA2Dot, ενώ σε εξέλιξη βρίσκεται η έρευνα για τους ακόμα μικρότερους κόμβους PicoNode, όπου δίνεται παράλληλα έμφαση στην ιδιαίτερα χαμηλή κατανάλωση ενέργειας της τάξης των 100 *mwatts* [31].

Στις συσκευές των κόμβων συναντάμε κατα κανόνα μια μονάδα παροχής ενέργειας, ένα μικροεπεξεργαστή, μια μονάδα μνήμης, έναν πομποδέκτη και μια μονάδα από αισθητήρες. Σε ορισμένες περιπτώσεις ανάλογα με τις απαιτήσεις συγκεκριμένων εφαρμογών οι κόμβοι μπορεί να ενσωματώνουν ένα mobilizer για να πραγματοποιούν μικρές κινήσεις στο χώρο. Επιπλέον μπορεί να διαθέτουν ένα σύστημα εύρεσης της τοποθεσίας τους με μεγάλη ακρίβεια (Global positioning system).

Τπάρχουν σοβαροί περιορισμοί όσον αφορά τους ενεργειακούς πόρους με τους οποίους μπορεί να εξοπλιστεί ένας κόμβος. Αυτό ισχύει κυρίως λόγω των μικρών διαστάσεων των κόμβων και προσδίδει μια αναλώσιμη φύση στα δίκτυα Smart Dust. Η ενέργεια συνήθως παρέχεται από δύο ή περισσότερες μπαταρίες λιθίου των 3 V και η διάρκεια ζωής υπό συνθήκες συνεχούς λειτουργίας δεν ξεπερνάει μερικές εβδομάδες. Ωστόσο με την ενδυνάμωση των κατάλληλων τεχνικών εξοικονόμησης ενέργειας, οι κόμβοι μπορούν να συνεχίσουν να λειτουργούν για αρκετούς μήνες.

Για τις προδιαγραφές των μικροεπεξεργαστών χρειάζεται να ληφθεί υπόψη το μέγεθος τους, η ενέργειακή τους κατανάλωση κατά τη διάρκεια υπολογισμών και η δυνατότητα να μεταπίπτουν σε καταστάσεις εξοικονόμησης ενέργειας. Πρέπει να σημειωθεί ότι δεν υπάρχουν μεγάλες απαιτήσεις σε υπολογιστική ισχύ λόγω του περιορισμένου επεξεργαστικού ρόλου που έχει ο κάθε κόμβος στο δίκτυο και της κατανεμημένης φύσης των εφαρμοζόμενων πρωτόκόλλων. Παράδειγμα ενός μικροελεγκτή για κόμβο Smart Dust αποτελεί ο 8-bit Atmel AT90LS85835 που ενσωματώθηκε στα Rene motes [18]. Ο μικροεπεξεργαστής αυτός λειτουργεί σε συχνότητα 4 *MHz*, και διαθέτει 512 *bytes* RAM και 512 *bytes* EEPROM.

Οι πομποδέκτες είναι απαραίτητος εξοπλισμός των κόμβων αλλά και η κύρια πηγή κατανάλωσης της ενέργειας τους. Συνηθέστερα χρησιμοποιούνται πομποδέκτες RF τεχνολογίας. Ο περιορισμένος χώρος των Smart Dust δεν αφήνει περιθώρια για μεγάλες κεραίες και μεταδόσεις σε μεγάλες αποστάσεις. Παράδειγμα ενός πομποδέκτη RF είναι ο Monolithics TR1000 που είναι ενσωματωμένος στα Rene και MICA motes και λειτουργεί στα 916.5 *MHz*. Η ταχύτητα μετάδοσης κυμαίνεται από 5 έως 20 *Kbps* [40]. Ο TR1000 διαθέτει τρεις καταστάσεις λειτουργίας, ήτοι λήψη, μετάδοση και κατάσταση sleep, οι οποίες καταναλώνουν αντίστοιχα 4.5 *mA*, 12 *mA* και 5 *µA*. Εναλλακτικά μπορεί να χρησιμοποιείται τεχνολογία IrDA(infrared) ή οπτικά μέσα (laser) για την επικοινωνία των κόμβων.

Τέλος οι αισθητήρες μπορεί να είναι ενσωματωμένοι ως αναπόσπαστό κομμάτι στη συσκευή των κόμβων ή να αποτελούν ξεχωριστό κύκλωμα που μπορεί να συνδέεται εφαπτόμενο πάνω στον κόμβο. Μερικά από τα πιο συνήθη είδη αισθητήρων είναι οι αισθητήρες φωτός, θερμοκρασίας και ήχου, ενώ υπάρχουν και αισθητήρες που μετρούν την επιτάχυνση ενός αν-

τικειμένου, την ύπαρξη μαγνητικών πεδίων, τα επίπεδα υγρασίας και άλλα. Οι λειτουργία των αισθητήρων γίνεται σε συνεργασία με μετασχηματιστή αναλογικού σε ψηφιακό σήμα, ώστε να μπορούν οι μετρήσεις που λαμβάνοται να παρέχονται στον μικροελεγκτή για παραπέρα επεξεργασία. Παράδειγμα κυκλωμάτων με αισθητήρες που διατίθενται στο εμπόριο είναι οι MTS300 και MTS310 που προορίζονται για προσάρτηση στους MICA και MICA2 κόμβους [37].

1.2.5 Κατανάλωση ενέργειας

Στα δίκτυα Smart Dust ο πιο βασικός περιορισμός που τίθεται είναι η απαίτηση για χαμηλή κατανάλωση ενέργειας τόσο σε επίπεδο κόμβου όσο και συνολικά για όλο το δίκτυο. Ο ρυθμός κατανάλωσης ενέργειας ανα κόμβο, καθορίζει τόσο τη διάρκεια ζωής όσο και ολόκληρου του Smart Dust δίκτυου, δεδομένου ότι στις περισσότερες των περιπτώσεων δεν υπάρχει δυνατότητα επαναφόρτισης ή αντικατάστασης των ενεργειακά εξαντλημένων κόμβων. Στην πιο συνήθη περίπτωση το δίκτυο παραμένει χωρίς επίβλεψη μέχρι να μην υφίσταται πια, λόγω της ενεργειακής εξάντλησης των κόμβων που το απαρτίζουν.

Λόγω του μικρού τους μεγέθους οι κόμβοι ενός δίκτυου από αισθητήρες, μπορούν να είναι εξοπλισμένοι με μπαταρίες, οι οποίες ως γνωστό έχουν πεπερασμένο χρόνο ζωής. Τυπικό μέγεθος για τα αρχικά ενεργειακά αποθέματα ενός κόμβου είναι τα 3 V. Πρέπει εδώ να σημειωθεί ότι ένας κόμβος σε Smart Dust δίκτυα επιτελεί πολλαπλές λειτουργίες, όπως ανίχνευση του περιβάλλοντος για κρίσιμα γεγονότα ή μέτρηση φαινομένων, επεξεργασία δεδομένων που προκύπτουν από μετρήσεις ή από τη λήψη μηνυμάτων, δημιουργία και αποστολή μηνυμάτων που περιέχουν πληροφορία για τα γεγονότα που ανίχνευσε, αλλά και προώθηση ή δρομολόγηση μηνυμάτων που δέχεται από άλλους κόμβους. Χονδρικά λοιπόν μπορεί να ειπωθεί ότι η κατανάλωση ενέργειας οφείλεται σε τρεις τομείς λειτουργιών :

- α) ανίχνευση του περιβάλλοντος (sensing),
- β) επικοινωνίες και
- γ) επεξεργασία δεδομένων.

Όλες αυτές οι δραστηριότητες έχουν ένα ενεργειακό κόστος και επιταχύνουν την ενέργειακή εξάντληση του κόμβου. Έτσι η διαχείριση της ενέργειας των κόμβων και η ανάπτυξη τεχνικών εξοικονόμησης της αποκτούν ειδική βαρύτητα. Τα περισσότερο αποδοτικά από τα προτεινόμενα πρωτόκολλα για δίκτυα με αισθητήρες είναι power aware, δηλαδή λαμβάνουν υπόψη τους την ενέργειακή κατανάλωση και τείνουν να περιορίσουν τους ρυθμούς της.

Αναφορικά με τις δραστηριότητες ενός κόμβου, η πιο δαπανηρή είναι μακράν η αποστολή και λήψη μηνυμάτων. Ειδικά στην αποστολή μηνυμάτων το σήμα του αποστελλόμενου μηνύματος πρέπει να ενισχυθεί επαρκώς με τη χρήση ενός amplifier, ώστε να μεταδοθεί αξιόπιστα σε δεδομένη απόσταση, χωρίς να αλλοιωθεί από παρεμβαλλόμενο ψόρυβο και ανακλάσεις. Στην περίπτωση μάλιστα μεταδόσεων που γίνονται από σημείο κοντά στο έδαφος η ενέργεια που καταναλώνεται στην ενίσχυση του σήματος είναι ανάλογη της τετάρτης δύναμης της απόστασης μετάδοσης.

Πρέπει να ειπωθεί [4], ότι στις περισσότερες πρακτικές εφαρμογές οι μεταδόσεις από κόμβους (λόγω προώθησης πληροφορίας προς το κέντρο ελέγχου) είναι σχετικά μη συχνό φαινόμενο.

Κατά κύριο λόγο στη διάρκεια της ζωής του ένας κόμβος απλά αισθάνεται - παρατηρεί το περιβάλλον του περιμένοντας να συμβεί κάποιο χρίσιμο γεγονός, οπότε τα υπόλοιπα κυκλώματα πέρα από τα αισθητήρια μπορούν να είναι σε λειτουργία χαμηλής κατανάλωσης (sleep mode). Αυτό βρίσκει εφαρμογή σε σενάρια ανίχνευσης πυρκαγιάς, σε εφαρμογές ανάπτυξης Smart Dust σε πεδία μαχών και σε περιπτώσεις όπου το δίκτυο αντιδράει (reactive) και προωθεί ένα συμβάν, μόνο όταν ικανοποιούνται προκαθορισμένες συνθήκες.

Σε περιπτώσεις, ωστόσο, πρωτοκόλλων γενικού σκοπού για Smart Dust δίκτυα μπορούν να εφαρμόζονται περισσότερο γενικευμένες πολιτικές που να ρυθμίζουν πότε ένας κόμβος λειτουργεί με χαμηλή κατανάλωση και πότε επανέρχεται σε κανονική λειτουργία ή ποιες ομάδες κυκλωμάτων μπορούν να απενεργοποιούνται και να επαναλειτουργούν όταν αυτό είναι απαραίτητο. Αυτές οι πολιτικές εξοικονόμησης ενέργειας, μπορούν σε μερικές περιπτώσεις να αποδειχθούν μη αποδοτικές. Αυτό μπορεί να συμβεί καθώς η μετάβαση μεταξύ καταστάσεων χαμηλής κατανάλωσης και κανονικής κατανάλωσης συνήθως προϋποθέτει τη δαπάνη ενός μη αμελητέου ποσού ενέργειας, Έτσι η υιοθέτηση μιας πολιτικής με μεγάλη συχνότητα μεταβάσεων μπορεί να φέρει αντίθετα από τα αναμενόμενα αποτελέσματα και να αποβεί μοιραία για το χρόνο ζωής του δικτύου.

Μια άλλη πρόταση για το πρόβλημα των πεπερασμένων ενέργειακών πόρων είναι η ενσωμάτωση μηχανισμών απορρόφησης και αποθήκευσης ενέργειας (όπως της ηλιακής με χρήση ηλιακών κυψελών). Ωστόσο, αυτό θα επιβαρύνει τους κόμβους με πρόσθιτες απαιτήσεις σε χώρο, καθώς η μικρή επιφάνεια των κόμβων έξυπνης σκόνης προσφέρει πενιχρές δυνατότητες αλίευσης ενέργειας.

1.2.6 Μέσο μεταδόσεων

Η ασύρματη φύση των δικτύων έξυπνης σκόνης, σημαίνει ότι η επικοινωνία μεταξύ των κόμβων για τη διάδοση της πληροφορίας γίνεται πάνω από ασύρματο μέσο μετάδοσης. Η ασύρματη μετάδοση μπορεί να επιτευχθεί με οπτικά μέσα, με υπεριώδεις ακτίνες (infrared) ή με μετάδοση σε RF(radio frequency). Γενικά για να είναι δυνατή η λειτουργία σε παγκόσμια κλίμακα των Smart Dust δικτύων το επιλεγόμενο μέσο μετάδοσης πρέπει να είναι διαθέσιμο προς χρήση παντού ανα τον κόσμο.

Η μετάδοση σε RF σημαίνει μετάδοση με ηλεκτρομαγνητικά κύματα και αντιστοιχεί σε ζώνη συχνοτήτων από 9 KHz μέχρι 400 GHz [1]. Για τις μεταδόσεις αυτές προσφέρεται η ζώνη συχνοτήτων που έχει δεσμευθεί για βιομηχανικούς, επιστημονικούς και ιατρικούς σκοπούς (ISM). Στις περισσότερες χώρες αυτό το φάσμα συχνοτήτων προσφέρεται για χρήση χωρίς μίσθωση ή απόκτηση άδειας χρήσης. Το κύριο πλεονέκτημα στη χρήση του ISM φάσματος είναι η ελεύθερη - χωρίς χρέωση - διάθεση του, το μεγάλο εύρος του και φυσικά η διαθεσιμότητα του σε παγκόσμια κλίμακα.

Το μέσο μετάδοσης RF, χρησιμοποιείται από τους περισσότερους πρότυπους υλοποιημένους κόμβους COTS dust. Η υλοποίηση του μΑΜΡΣ-I κόμβου - αισθητήρα χρησιμοποιεί ένα πομποδέκτη συμβατό με την τεχνολογία Bluetooth που λειτουργεί σε συχνότητες 2.4 GHz και έχει ενσωματωμένο συνθεσάζερ συχνοτήτων. Επίσης στο project WINS (Wireless Integrated Network Sensors) χρησιμοποιείται RF επικοινωνία μεταξύ των κόμβων του δικτύου. Τέλος στο project MICA Smart Dust motes, οι κόμβοι έχουν ενσωματωμένο ένα RF πομποδέκτη που λειτουργεί στα 916 MHz.

Οι RF πομποδέκτες προτιμώνται στην υλοποίηση ενσωματωμένων συστημάτων μεταδόσης/-λήψεως σε κόμβους Smart Dust, παρά τη σχετικά πάνω από τα επιθυμητά όρια κατανάλωση ενέργειας που παρουσιάζουν. Αυτό συμβαίνει κυρίως γιατί στα δίκτυα αισθητήρων τα πακέτα που μεταδίδονται είναι μικρά και οι ρυθμοί μετάδοσης αρκετά χαμηλοί, οπότε περιορίζεται η κατανάλωση ενέργειας. Επιπρόσθετα σημαντική είναι και η δυνατότητα επαναχρησιμοποίησης συχνοτήτων από ομάδες κόμβων που απέχουν ικανή απόσταση, λόγω του ότι οι μεταδόσεις γίνονται σε μικρή ακτίνα.

Οι εναλλακτικές επιλογές για την ασύρματη μετάδοση είναι μετάδοση σε infrared ή με οπτικά μέσα optical. Το πλεονέκτημα των infrared επικοινωνιών είναι ότι τα κυκλώματα των πομποδεκτών είναι σχετικά εύκολο να κατασκευαστούν και συνεπώς οι πομποδέκτες είναι αρκετά φυγηνοί. Ωστόσο, μαζί με την μετάδοση με οπτικά μέσα έχουν το μειονέκτημα ότι απαιτούν την ύπαρξη line of sight μεταξύ πομπού και δέκτη, δηλαδή την ύπαρξη νοητής ευθείας χωρίς εμπόδια που να ενώνει τον πομπό με το δέκτη. Ειδικά οι infrared μεταδόσεις έχουν επιπλέον το σοβαρό μειονέκτημα της δραματικά περιορισμένης εμβέλειας μετάδοσης. Το project Smart Dust που διεξάγεται στο πανεπιστήμιο του Berkley τείνει να υιοθετήσει την ασύρματη μετάδοση με οπτικά μέσα για τις μικροσκοπικές συσκευές που θα στελεχώνουν τα μελλοντικά δίκτυα αισθητήρων, και εκθέτει ένα σύνολο από επιχειρήματα για να υποστηρίξει την επιλογή αυτή έναντι της τεχνολογίας RF.

Συγχριτική παρουσίαση τεχνολογίας RF και optical

Κατά γενική διαπίστωση οι μέθοδοι μετάδοσης πληροφορίας από κόμβο σε κόμβο σε ένα ασύρματο δίκτυο από αισθητήρες είναι απαραίτητο να ακολουθούν μια αρχιτεκτονική που θα εξασφαλίζει κατά το δυνατό ελάχιστη κατανάλωση ενέργειας. Επίσης πρέπει να είναι χαμηλού κόστους και να μπορούν να ενσωματωθούν πάνω στις μικροσκοπικές συσκευές των δικτύων χωρίς να έχουν υπερβολικές απαιτήσεις σε χώρο. Τέλος πρέπει η αρχιτεκτονική τους να εξασφαλίζει αξιόπιστες μεταδόσεις σε επαρκώς μεγάλες αποστάσεις, ακόμα και σε πεδία με ιδιαιτερότητες στη μορφολογία ή σε πεδία με υψηλά επίπεδα θορύβου ή παρεμβολών στα ασύρματα κανάλια επικοινωνίας. Με βάση αυτά τα κριτήρια γίνεται η σύγχριση των δύο πιο διαδεδομένων τεχνολογιών μετάδοσης RF και optical [14].

Στη τεχνολογία μεταδόσης/λήψης RF το κυκλώμα του πομποδέκτη ονομάζεται ραδιοπομπόδεκτης. Οι ραδιοπομποδέκτες είναι πολύπλοκα κυκλώματα, που καθιστούν δύσκολο τον περιορισμό της ενέργειακής κατανάλωσης σε τάξη μεγέθους μερικών microwatts. Για την υλοποίηση ενός ραδιοπομποδέκτη απαιτείται ένος συνδυασμός από κυκλώματα διαμόρφωσης/αποδιαμόρφωσης σήματος (modulation/demodulation), ζωνοδιαβατά φίλτρα και επιπρόσθετα κυκλώματα αν είναι επιθυμητό οι μεταδόσεις να πολυπλέκονται ακολουθώντας τις τεχνικές TDMA ή FDMA ή CDMA. Η πολυπλοκότητα των κυκλωμάτων ενός ραδιοπομποδέκτη έχει δυσμενείς επιπτώσεις και στον απαιτούμενο χώρο στη συσκευή Smart Dust.

Κατά κανόνα για να επιτευχθεί αξιόπιστη μετάδοση με τεχνολογία RF σε μεγάλες σχετικά αποστάσεις είναι αναγκαία η ύπαρξη κεραίας η οποία είναι συνήθως απαγορευτική και λόγω των απαιτήσεων σε διαθέσιμο χώρο άλλα και γιατί έτσι οι συσκευές έξυπνης σκόνης χάνουν το πλεονέκτημα του μικρού τους μεγέθους και τη δυνατότητα να παραμένουν κρυμμένες ή τουλάχιστο δύσκολα ανιχνεύσιμες σε εχθρικά περιβάλλοντα. Η απουσία κεραίας συνεπάγεται λειτουργία του πομπού σε μικρό μήκος κύματος και συνεπώς με υψηλή συχνότητα. Ωστόσο, αυτή η τελευταία τεχνική μετάδοσης δεν έχει επιτύχει ακόμα την επιθυμητή ελάχιστη

κατανάλωση ενέργειας και συνεπώς κρίνεται μη αποδοτική.

Ένα άλλο μειονέκτημα είναι ότι με την τεχνολογία RF η εξασθένιση του μεταδιδόμενου σήματος είναι ανάλογη προς μια *n*-οστή δύναμη της απόστασης μεταξύ κόμβου - πομπού και κόμβου - δέκτη (όπου το *n* μπορεί να είναι από 2 έως 4). Μάλιστα το *n* είναι συνήθως κοντά στη τιμή 4 καθώς οι κόμβοι κείτονται στο έδαφος όπου υπάρχει μεγάλη πιθανότητα “ακύρωσης” του μεταδιδόμενου σήματος από ανακλώμενα σήματα πάνω στο έδαφος.

Στα πλεονεκτήματα της χρήσης RF συγκαταλέγεται η αρκετά διαδεδομένη χρήση του σε σχέση με τις τεχνολογίες μετάδοσης με οπτικά μέσα, και η ανεξαρτησία της λειτουργίας του από την ύπαρξη ή όχι *line of sight* μεταξύ πομπού και δέκτη κόμβου. Αυτό σημαίνει ότι οι μεταδόσεις RF δεν ανακόπτονται από την τυχόν ύπαρξη εμποδίων στο χώρο του δικτύου και συνεπώς ευνοούνται οι εφαρμογές που απαιτούν ανάπτυξη του δικτύου σε εδάφη με μορφολογικές ανωμαλίες.

Όσον αφορά τις τεχνικές οπτικής μετάδοσης, τα κυκλώματα του πομποδέκτη είναι αρκετά πιο απλά από τα αντίστοιχα ενός ραδιοπομποδέκτη και πιο εύκολα να κατασκευαστούν. Αυτό συμβαίνει γιατί τα κυκλώματα για οπτική μετάδοση, δεν περιλαμβάνουν διαμορφωτές / αποδιαμορφωτές σήματος ή ενεργά ζωνοδιαβατά φίλτρα αλλά μόνο απλά βαθυπερατά αναλογικά και ψηφιακά κυκλώματα. Με την προϋπόθεση ύπαρξης *line of sight*, μια καλά σχεδιασμένη τεχνική οπτικής μετάδοσης απαιτεί σημαντικά λιγότερη ενέργεια ανά bit από την RF τεχνική. Η οπτική τεχνολογία επιτρέπει μεταδόσεις με μικρό μήκος κύματος της τάξης του μήκους κύματος του ορατού φωτός ή των infrared συχνοτήτων (γύρω στο 1 micron). Έτσι μια συσκευή μικροσκοπική της τάξης μερικών κυβικών χιλιοστών μπορεί να εκπέμψει σε επαρκή απόσταση μια στενή ακτίνα, χωρίς τη χρήση εξωτερικής κεραίας.

Γενικά ο σταθμός του κέντρου ελέγχου μπορεί να συνεργαστεί αποδοτικά με συσκευές, που εκπέμπουν με οπτικά μέσα, αν διαθέτει ένα δέκτη imaging. Αυτό σημαίνει ότι μπορεί να αντιστοιχίσει κάθε λαμβανόμενο σήμα από κόμβο σε ένα pixel της εικόνας για το χώρο εξάπλωσης του δικτύου, την οποία διατηρεί ο σταθμός ελέγχου. Αυτό το pixel υποδεικνύει τη θέση του κόμβου στο πεδίο του δικτύου. Με το δέκτη imaging ένας σταθμός ελέγχου μπορεί να αποκωδικοποιήσει ταυτόχρονες μεταδόσεις από ένα μεγάλο αριθμό Smart Dust συσκευών που βρίσκονται σε διαφορετικές τοποθεσίες του ορατού πεδίου από το κέντρο ελέγχου. Εκπέμποντας λοιπόν μια ακτίνα αρκετά μεγάλου εύρους ο σταθμός ελέγχου μπορεί να επικοινωνεί ταυτόχρονα με πολλά motes και καθώς αυτά απαντούν ταυτόχρονα ο σταθμός ελέγχου θα διαχωρίζει τις απαντήσεις τους στον πίνακα από pixels (εικόνα) που διατηρεί εφόσον βέβαια τα motes είναι επαρκώς διακριτά το ένα από το άλλο όσον αφορά την τοποθεσία τους. Έτσι υλοποιείται ένα είδος πολύπλεξης των μεταδόσεων στο χώρο (space division multiplexing).

Ένα πρόσθετο πλεονέκτημα της οπτικής τεχνολογίας για επικοινωνίες, είναι η δυνατότητα παθητικής μετάδοσης με τη χρήση των συστημάτων CCR (corner cube retroreflection). Τα συστήματα αυτά με τη χρήση καταλλήλως κεκλιμένων κατόπτρων ανακλούν την ενεργή εκπομπή φωτός που γίνεται είτε από κάποιο άλλο κόμβο του δικτύου ή από το κέντρο ελέγχου. Και εδώ απαιτείται ανεμπόδιστο *line of sight* μεταξύ ενεργού πομπού και παθητικού πομποδέκτη. Επίσης αν υποθέσουμε ότι ο σταθμός ελέγχου εκτελεί όλες τις ενεργές μεταδόσεις, τότε η διαγώνιος του συστήματος CCR πρέπει να ευθυγραμμίζεται με την κατεύθυνση του σταθμού ελέγχου με ελάχιστη επιτρεπόμενη απόκλιση μερικών μοιρών.

Η παθητική μετάδοση συνήθως λειτουργεί ως εξής: Ο σταθμός ελέγχου εκπέμπει ακτίνα

laser που φωτίζει την περιοχή που καλύπτουν τα dust motes. Αυτή η ακτίνα μπορεί να χρησιμοποιείται για να ρωτάει (query) τα motes για πληροφορίες ή να τα ξυπνάει από κατάσταση αναμονής και χαμηλής κατανάλωσης (idle modes). Οι κόμβοι χρησιμοποιώντας το CCR μπορούν να απαντήσουν στην ενεργή μετάδοση του σταθμού ελέγχου. Ένα αρκετά ενδιαφέρον παράδειγμα είναι αυτό όπου ο σταθμός ελέγχου είναι προσαρτημένος πάνω σε κυάλια που τα κατευθύνει ένας άνθρωπος ώστε να συγκεντρώνει πληροφορίες από συγκεκριμένες περιοχές που καλύπτονται από Smart Dust δίκτυα.

Ωστόσο, η δυνατότητα παθητικής μετάδοσης σε **πολλαπλά βήματα** (multihop) και η δυνατότητα δρομολόγησης της πληροφορίας αντί της απευθείας μετάδοσης της στο κέντρο ελέγχου είναι πρακτικά μη αποδοτική. Αυτό ισχύει γιατί αυτές οι δυνατότητες θα απαιτούσαν περισσότερα του ενός συστήματα CCR ενσωματωμένα στους κόμβους, από τα οποία το καθένα θα έπρεπε να ευθυγραμμίζεται με την κατεύθυνση κάποιου γειτονικού κόμβου. Κάτι τέτοιο θα είχε υπερβολικά μεγάλες απαιτήσεις σε χώρο πάνω στη συσκευή του κόμβου και δεν θα ήταν καθόλου πρακτικό.

Ως τρίτο πλεονέκτημα η τεχνολογία οπτικής μετάδοσης έχει να επιδείξει την ενεργή οπτική μετάδοση και το πιθανό συνδυασμό της με σύστημα παθητικής μετάδοσης πάνω στον ίδιο κόμβο. Η ενεργή οπτική μετάδοση επιτρέπει την επικοινωνία με άλλους κόμβους αν υπάρχει line of sight. Παράλληλα θέτει ζητήματα ζυγοστάθμισης μεταξύ της απόστασης μετάδοσης και του προσφερόμενου ρυθμού μετάδοσης (bandwidth). Ωστόσο, σε σχέση με τη παθητική μετάδοση, η ενεργή μετάδοση παρουσιάζει σχετικά μεγάλη κατανάλωση ενέργειας. Επίσης για να υλοποιηθεί το multihop με ενεργή οπτική μετάδοση απαιτείται δυνατότητα κατευθυντικότητας προς τους δέκτες των άλλων κόμβων, γεγονός που δημιουργεί την ανάγκη για ύπαρξη ενσωματωμένης λογική στους κόμβους, η οποία θα καθορίζει την κατεύθυνση της ακτίνας.

Συνοψίζοντας τα κυριότερα μειονεκτήματα της οπτικής επικοινωνίας θα αναφέραμε την ουτοπικά απαραίτητη ύπαρξη line of sight, την απαίτηση για δυνατότητα κατευθυντικότητας και την ασυμμετρία των ασύρματων συνδέσμων. Με αυτό το τελευταίο εννοούμε ότι γενικά μπορεί ένας κόμβος A να στέλνει μηνύματα που θα τα ακούσει ο κόμβος B, αλλά αυτό δε σημαίνει ότι όταν ο B θα εκπέμψει μήνυμα, ο A θα το λάβει.

Με τα σύγχρονα δεδομένα οι τεχνολογίες μετάδοσης με RF και μετάδοσης με οπτικά μέσα (πχ laser), έχουν και οι δύο να επιδείξουν θετικά σημεία που τις καθιστούν επαρκείς για την χρησιμοποίηση τους στα πρότυπα δίκτυα έξυπνης σκόνης. Μπορεί κανείς βλέποντας τις τρέχουσες προόδους στις επικοινωνίες με οπτικά μέσα να υποθέσει ότι το μέλλον στις ασύρματες επικοινωνίες ανήκει στις μεταδόσεις και λήψεις με οπτική τεχνολογία. Βέβαια δε πρέπει να παραβλέψουμε το γεγονός ότι σημαντικές πρόδοι αναμένονται και στην RF τεχνολογία, κυρίως όσον αφορά το μέγεθος και την πολυπλοκότητα των κυκλωμάτων για τους μελλοντικούς ραδιοπομποδέκτες. Έτσι δεν αποκλείεται τα μελλοντικά δίκτυα να έχουν κόμβους που να χρησιμοποιούν μία από τις δύο τεχνολογίες μετάδοσης ανάλογα με τις ιδιαίτερες απαιτήσεις της κάθε εφαρμογής, ή ακόμα να ενσωματώνουν και τις δύο τεχνολογίες και να χρησιμοποιούν όποια κρίνεται κατάλληλη κατά περίπτωση.

1.2.7 Κόστος παραγωγής

Το κόστος ανα κόμβο στα δίκτυα έξυπνης σκόνης είναι ιδαίτερης σημασίας, αν αναλογιστεί κανείς ότι κατά κανόνα τα δίκτυα αυτά αποτελούνται από εκατοντάδες έως και χιλιάδες

κόμβους. Αν το κόστος της ανάπτυξης ενός δικτύου αισθητήρων είναι μεγαλύτερο από το κόστος χρήσης μεμονωμένων αισθητήρων, τότε δεν δικαιολογείται η επένδυση στο δίκτυο των αισθητήρων.

Κύριοι παράγοντες που συνεισφέρουν στο κόστος μιας συσκευής Smart Dust είναι τα κυκλώματα του πομποδέκτη και οι διάφοροι αισθητήρες που είναι προσαρτημένοι σε αυτή. Με τη σύγχρονη τεχνολογία ένα σύστημα ραδιοπομπού Bluetooth κοστίζει περίπου 10 ευρώ. Το κόστος αυτό είναι σχετικά μεγάλο, και υπολογίζεται ότι το επιθυμητό κόστος ανα συσκευή πρέπει να κυμαίνεται σε τιμές αρκετά μικρότερες από 1 ευρώ. Ας σημειωθεί ότι ένα σύστημα Bluetooth αποτελεί ένα από τα πιο φθηνά μέσα μετάδοσης και πάλι είναι τουλάχιστο δεκαπλάσιο από το προβλεπόμενο κόστος για τις συσκευές Smart Dust.

1.3 Κίνητρα και Εφαρμογές στη χρήση των Smart Dust δικτύων

Η ανάπτυξη των δικτύων έξυπνης σκόνης πηγάζει από την ανάγκη για αποδοτικές τεχνικές παρακολούθησης και ανίχνευσης που θα εφαρμόζονται αξιόπιστα ακόμα και στα πιο επικίνδυνα ή δυσπρόσιτα περιβάλλοντα όπως είναι το εσωτερικό ενός κυκλώνα, ο κρατήρας ενός ηφαιστείου ή ένα πεδίο μάχης. Το κόστος ανάπτυξης των δικτύων που θα ενσωματώνουν αυτές τις τεχνικές πρέπει να είναι το μικρότερο δυνατό, και η διάρκεια ζωής η μεγαλύτερη δυνατή. Συχνά απαιτείται και ο περιορισμός του κινδύνου εντοπισμού του δικτύου από το αντικείμενο της παρακολούθησης, ειδικά όταν πρόκειται για εχθρικές δυνάμεις. Τέλος χρειάζεται αυτόνομη και αυτόματη οργάνωση του δικτύου και εγγυήσεις για τη λειτουργία του ακόμα και όταν αρκετοί κόμβοι καταστραφούν. Αυτές τις ανάγκες υπόσχεται να καλύψει η τεχνολογία των δικτύων Smart Dust.

1.3.1 Εφαρμογές των δικτύων Smart Dust

Οι εφαρμογές των Smart Dust δικτύων [1] είναι πολυάριθμες, καθώς οι κόμβοι στα δίκτυα αυτά έχουν πολυδιάσταση λειτουργικότητα, ενσωματώνοντας δυνατότητες ανίχνευσης, επεξεργασίας και δρομολόγησης πληροφορίας. Μερικά από τα μεγέθη που μπορεί να ανιχνεύονται από ένα κόμβο Smart Dust είναι η θερμοκρασία, η υγρασία, η ατμοσφαιρική πίεση, το επίπεδο ηχητικού θορύβου, τα μαγνητικά πεδία, το επίπεδο φωτεινότητος και η ταχύτητα κίνησης ενός αντικειμένου. Οι προτεινόμενες εφαρμογές αναφέρονται στους τομείς της υγείας, της οικιακής αυτοματοποίησης, του φυσικού περιβάλλοντος, του στρατού, του εμπορίου και σε αρκετές άλλους τομείς του ανθρώπινου πολιτισμού. Ενδεικτικά παραθέτουμε κάποιες χαρακτηριστικές εφαρμογές.

Από τις κυριότερες και ταυτόχρονα δύσκολες εφαρμογές των δικτύων Smart Dust στον τομέα των στρατιωτικών επιχειρήσεων είναι η εποπτεία του πεδίου μάχης. Τα δίκτυα αναπτύσσονται για να καλύψουν περιοχές στρατηγικής σημασίας καθώς και δρόμους που μπορούν να αποτελέσουν διόδους μετακίνησης και πιθανής επίθεσης ή εισβολής για τα εχθρικά στρατεύματα. Ο κεντρικός στόχος είναι η ανίχνευση των δραστηριοτήτων στα εχθρικά στρατεύματα και η έγκαιρη ειδοποίηση των συμμαχικών δυνάμεων. Η πρόκληση είναι η επιβίωση των δικτύων σε αφιλόξενα περιβάλλοντα, η απόκρυψη της λειτουργίας τους από τις εχθρικές δυνάμεις, και

η απρόσκοπη επικοινωνία με τα συμμαχικά κέντρα ελέγχου παρά τις όποιες προσπάθειες παρεμβολών από τις εχθρικές δυνάμεις.

Μια άλλη σημαντική και αρκετά επίκαιρη εφαρμογή είναι η έγκαιρη ανίχνευση της χρήσης χημικών, πυρηνικών ή βιολογικών όπλων από τον εχθρό και η προειδοποίηση των φίλιων δυνάμεων, ώστε να αποφευχθούν απώλειες σε ανθρώπινες ζωές. Για το σκοπό αυτό τα Smart Dust δίκτυα είναι σημαντικό να βρίσκονται κατά το δυνατό πιο κοντά στο σημείο “μηδέν” που εκδηλώνεται η επίθεση.

Άλλες στρατιωτικές εφαρμογές περικλείουν συστήματα πλοήγησης και στόχευσης για έξυπνα όπλα, καθώς και αναγνώριση των εχθρικών δυνάμεων και ανάλυση του εξοπλισμού τους.

Οι εφαρμογές στον τομέα του περιβάλλοντος περιλαμβάνουν ανάμεσα σε άλλες την παρακολούθηση της κίνησης μικρών ζώων, την παρατήρηση των περιβαλλοντολογικών συνθηκών και των αλλαγών σε αυτές, τη μελέτη των επιπέδων μόλυνσης μιας περιοχής και την ανίχνευση μιας πυρκαγιάς σε δάσος.

Η υγεία είναι επίσης ένας τομέας που μπορεί να επωφεληθεί από την ανάπτυξη των Smart Dust δικτύων. Με τη προσκόλληση των μικροσκοπικών κόμβων - αισθητήρων σε φάρμακα και ασθενείς, είναι δυνατό να γίνεται με αυτόματο και αξιόπιστο τρόπο ο προσδιορισμός των αλλεργιών ενός ασθενή και των φάρμακων που πρέπει να του χορηγηθούν. Έτσι εξαλείφονται μαζί ο κίνδυνος για τη χορήγηση λάθεμένης φαρμακευτικής αγωγής αλλά και ο κίνδυνος παρενεργειών από φάρμακα.

Επιπρόσθετα στα νοσοκομεία αλλά και σε άλλα ιδρύματα όπου δίνεται ιδιαίτερη έμφαση στη φροντίδα των φιλοξενούμενων (γηροκομεία, κέντρα αποτοξίνωσης) τα Smart Dust δίκτυα προσφέρουν μηχανισμούς ελέγχου των κινήσεων αλλά και της φυσικής κατάστασης γιατρών και νοσηλευομένων. Αν ο κάθε ασθενής διαθέτει ένα κόμβο Smart Dust προσκολλημένο στο σώμα του, ώστε να παρακολουθεί την πίεση του αίματος, τη θερμοκρασία του σώματος και τις κινήσεις του, μπορεί να αναφέρονται πολλές χρήσιμες και κρίσιμες για την υγεία του ασθενούς πληροφορίες στο κατάλληλο προσωπικό. Παράλληλα οι γιατροί θα έχουν προσκολλημένο ένα κόμβο Smart Dust στο σώμα τους, ώστε να μπορεί να διαπιστωθεί ανα πάσα στιγμή η θέση τους στο νοσοκομείο, σε περίπτωση έκτακτων περιστατικών που χρήζουν της προσοχής τους.

Από τις εφαρμογές των δικτύων Smart Dust δε πρέπει να παραλείψουμε τον οικιακό αυτοματισμό που υπόσχεται ένα πιο άνετο και εκσυγχρονισμένο οικιακό περιβάλλον. Η ιδέα έγκειται στην ενσωμάτωση Smart Dust κόμβων σχεδόν σε κάθε οικιακή συσκευή (τηλεόραση, ψυγείο, βίντεο, φούρνος, πλυντήριο). Διαμορφώνοντας ένα δίκτυο μέσα στο σπίτι το οποίο θα μπορεί να επικοινωνεί με εξωτερικά δίκτυα μέσω δορυφόρου ή από το Internet, θα γίνεται εφικτή η διαχείριση των οικιακών συσκευών είτε από πρόσωπο εντός της οικίας ή εξ' αποστάσεως με περισσότερη ευκολία.

Μερικές από τις υπόλοιπες εφαρμογές των Smart Dust δικτύων είναι η παρακολούθηση και οργανωμένη καταγραφή των αντικειμένων σε απόθεμα από εταιρίες, η ανίχνευση κλοπής οχημάτων και ο εντοπισμός τους, ο έλεγχος της κίνησης ρομποτικών μηχανισμών και η ρύθμιση περιβαλλοντολογικών συνθηκών (θερμοκρασία, υγρασία) σε κτίρια.

1.4 Το πρόβλημα

Το πρόβλημα που καλούνται να λύσουν τα προτεινόμενα πρωτόκολλα για Smart Dust είναι κατά βάση η διάδοση μιας χρίσιμης πληροφορίας από τον κόμβο, ο οποίος την ανίχνευσε σε ένα κέντρο ελέγχου [6]. Ο κόμβος που ανιχνεύει ένα χρίσιμο γεγονός ή λαμβάνει κάποια μέτρηση για την οποία κρίνει ότι αξίζει να διαδοθεί, ονομάζεται κόμβος πηγή (source). Το πρόβλημα περιπλέκεται και αποκτά περισσότερο ρεαλιστικές διαστάσεις όταν υποθέτουμε ότι υπάρχουν περισσότεροι από ένας κόμβοι - πηγές στο δίκτυο.

Το πιο σύνηθες λοιπόν σενάριο σε ένα Smart Dust δίκτυο είναι η ανίχνευση μιας σειράς από γεγονότα E1, E2, E3, E4 έως EN από κόμβους - πηγές K1, K2, K3, K4 έως KN αντίστοιχα. Για τα γεγονότα E1 έως EN υποθέτουμε ότι μπορούν να συμβαίνουν σε τυχαίους χρόνους και δεν αποκλείεται κάποια από αυτά να συμβαίνουν ταυτόχρονα. Κάθε ένας από τους κόμβους πηγές δημιουργεί ένα μήνυμα με πληροφορία που αναφέρεται στο γεγονός που ανίχνευσε, περιγράφοντας το και δηλώνοντας προαιρετικά κάποια πρόσθετα στοιχεία, όπως το χρόνο που συνέβη και τον τόπο (αν μπορεί να έχει γνώση της τοποθεσίας).

Στη συνέχεια οι κόμβοι καλούνται να συνεργαστούν για να διαδοθεί η πληροφορία που καταγράφτηκε από τους κόμβους - πηγές, μέχρι το κέντρο ελέγχου Sink. Τότε παρουσιάζονται θέματα βέλτιστης δρομολόγησης των μηνυμάτων, επεξεργασίας και συνένωσης μηνυμάτων που φέρουν διαφορετική πληροφορία καθώς και ελέγχου της κυκλοφορίας μηνυμάτων στο δίκτυο για να αποφευχθεί η συμφόρηση τμημάτων του.

Η διάδοση των μηνυμάτων μέσα στο δίκτυο πρέπει να γίνεται όσο πιο αποδοτικά γίνεται όσον αφορά στην κατανάλωση ενέργειας ανά κόμβο, την συνολική ενέργειακή κατανάλωση, και το ποσοστό άφιξης πληροφορίας στο κέντρο ελέγχου σε σχέση με τη πληροφορία που θα έπρεπε ιδανικά να φτάσει εκεί.

Το κέντρο ελέγχου έχει καθοριστικό ρόλο στη διαμόρφωση του προβλήματος. Ονομάζεται αλλιώς και Sink ή Base Station και ενδέχεται να μην είναι μόνο ένα, αλλά να υπάρχουν ομάδες από κέντρα ελέγχου. Συνήθως, ωστόσο υποθέτουμε ότι υπάρχει ένα μοναδικό κέντρο ελέγχου, το οποίο έχει ανεξάντλητους ενεργειακούς και αποθηκευτικούς πόρους και οπωσδήποτε ένα δέκτη για να λαμβάνει μηνύματα από το δίκτυο των Smart Dust. Σε κάποια πρωτόκολλα στο Sink αποδίδεται και η ικανότητα να διαδίδει μηνύματα σε ακτίνα τουλάχιστο ίση με τη διάμετρο του δικτύου, ώστε να μπορεί άμεσα να θέτει ερωτήματα (queries) σε οποιουσδήποτε επιλεγμένους κόμβους.

Η θέση του κέντρου ελέγχου έχει επίσης σημασία. Ενδέχεται, λοιπόν, να βρίσκεται αρκετά μακριά από το δίκτυο (μερικές εκατοντάδες μέτρα). Αυτό σημαίνει ότι οι κόμβοι που αναλαμβάνουν το έργο της τελικής μετάδοσης μιας πληροφορίας προς το Sink, επιφορτίζονται με μια ιδιαίτερα δαπανηρή ενέργειακά μετάδοση. Εναλλακτικά το κέντρο ελέγχου μπορεί να βρίσκεται μέσα στο δίκτυο, ώστε αυτό να αναπτύσσεται τριγύρω του ή να βρίσκεται σε ένα συνοριακό σημείο του δικτύου ώστε το δίκτυο να απλώνεται μόνο σε μια κατεύθυνση ως προς το Sink. Η τελευταία περίπτωση είναι αυτή που υιοθετούμε γενικά στην ανάλυση μας. Στην περίπτωση αυτή, η απευθείας μετάδοση προς το Sink είναι κατά μέσο όρο μικρότερης ακτίνας από την περίπτωση όπου το Sink απέχει αρκετά μέτρα από το δίκτυο. Επιπρόσθετα οι κόμβοι, που είναι “γειτονικοί” του Sink, μπορούν να μεταδώσουν πληροφορία στο Sink με δραματικά χαμηλό ενεργειακό κόστος, παρόμοιο με το κόστος μετάδοσης σε οποιοδήποτε γειτονικό κόμβο.

Τέλος για το κέντρο ελέγχου πρέπει να καθορίσουμε και τη δυνατότητα κίνησης του. Στην πλειονότητα των περιπτώσεων το κέντρο είναι ακίνητο και αντιπροσωπεύεται από ένα σημείο στο χώρο. Ωστόσο υπάρχουν και πρωτόκολλα που προϋποθέτουν ότι το Sink μπορεί να εκτελεί περιπάτους αλλάζοντας τη θέση του ως προς το δίκτυο, κυρίως για να αποσπάσει πληροφορίες από κόμβους που αδυνατούν να τις μεταδώσουν παραπέρα.

1.5 Σύγχρονα επιτεύγματα και η δική μας συνεισφορά

Αν και η τεχνολογία των δικτύων έξυπνης σκόνης είναι σχετικά καινούργια, ήδη έχουν σημειωθεί σημαντικά επιτεύγματα και οι πρόοδοι συνεχίζονται με εκθετικούς ρυθμούς. Τα τελευταία τρία χρόνια το ανεξερεύνητο πεδίο των δικτύων Smart Dust αποτέλεσε πρόκληση για εις βάθος έρευνα, σε πολλά πανεπιστήμια και ερευνητικές κοινότητες. Οι πρόοδοι στα δίκτυα Smart Dust ξεκινούν από τη συγκρότηση επιστημονικών προγραμμάτων που προτείνουν πιθανές υλοποίησεις ασύρματων δικτύων αισθητήρων και πρωτόκολλα επικοινωνίας για αυτά και φτάνουν μέχρι την παρουσίαση των πρώτων υλοποιημένων και εμπορικά διαθέσιμων συσκευών έξυπνης σκόνης. Βασικό ρόλο για τις προόδους αυτές έχει η αλματώδης εξέλιξη που παρουσιάζεται στον τομέα της μικρο-ηλεκτρο-μηχανικής (MEMS).

Η τεχνολογία MEMS τείνει να αντικαταστήσει την προϋπάρχουσα τεχνολογία της μικροηλεκτρονικής. Υπόσχεται την ολοκλήρωση μηχανικών διατάξεων, αισθητήρων και ηλεκτρονικών στοιχείων στο κοινό υπόστρωμα πυριτίου. Έτσι καθίσταται εφικτή η υλοποίηση αυτόνομων συστημάτων πολλαπλής λειτουργικότητας πάνω σε ένα ολοκληρωμένο κύκλωμα (systems on a chip). Ένα τέτοιο ολοκληρωμένο κύκλωμα, θα έχει δυνατότητες ελέγχου και μέτρησης του περιβάλλοντος, επεξεργασίας των δεδομένων από τους αισθητήρες και λήψης αποφάσεων για εκπομπή ή προώθηση πληροφορίας μέσω δικτύου. Επιπρόσθετα καθώς τα κυκλώματα MEMS κατασκευάζονται με μενόδους μαζικής παραγωγής, είναι δυνατή η διάθεση των παραγόμενων κυκλώματων σε χαμηλό κόστος σχετικά με τη λειτουργικότητα που προσφέρουν [30].

1.5.1 WEBS

Στο πανεπιστήμιο του Berkley η έρευνα πάνω σε θέματα των δικτύων έξυπνης σκόνης οργανώθηκε σε πολλές επιμέρους έρευνες, οι οποίες ανήκουν στο γενικευμένο project με το όνομα WEBS (Wireless Embedded Systems) [24]. Το WEBS αποτελεί ένα από τα σημαντικότερα project για ασύρματα δίκτυα αισθητήρων καθώς έχει παρουσιάσει αξιόλογα επιτεύγματα πάνω στον τομέα αυτό. Οι ερευνητές που εργάζονται στο project είναι οι David Culler, Eric Brewer, Kris Pister, Shankar Sastry και David Wagner, ενώ σε συνεργασία με τις Crossbow Technology, Inc έγινε εφικτή η υλοποίηση και εμπορική διάθεση πρότυπων συσκευών που μπορούν να στελεχώσουν τα πρώτα δίκτυα έξυπνης σκόνης για πειραματικούς κυρίως σκοπούς.

Στο project WEBS ανήκουν τα σημαντικά επιμέρους project με τις ονομασίες NEST, Smart-Dust, και PicoRadio.

NEST

Ουσιαστικά στο project υπό την ονομασία NEST [28], αναπτύσσεται μια πλατφόρμα λογισμικού και υλικού για ασύρματα δίκτυα με αισθητήρες, η οποία προορίζεται για να χρησιμοποιηθεί σε ερευνητικούς σκοπούς και πειράματα, ώστε να επιταχυνθεί η ανάπτυξη αλγορίθμων και πρωτοκόλλων για τα μελλοντικά δίκτυα έξυπνης σκόνης. Η βασική επιδίωξη είναι η πλατφόρμα αυτή να αποτελεί ένα πρώιμο εργαλείο που θα μεταφέρει την έρευνα από τη θεωρία των αλγορίθμων στην πρακτική εφαρμογή τους.

Στην πρώτη πειραματική πλατφόρμα που υλοποιήθηκε, οι συσκευές αν και έχουν τις δυνατότητες ανίχνευσης, επεξεργασίας και διάδοσης πληροφορίας που είναι επιθυμητές στις συσκευές έξυπνης σκόνης, είναι ωστόσο μεγάλης κλίμακας όσον αφορά το μέγεθός τους. Για την υλοποίηση σε ευσύνοπτο χρονικό διάστημα μιας λειτουργικής πλατφόρμας για πειραματικούς και μόνο λόγους, η μόνη εφικτή λύση ήταν η συνένωση κάποιων ήδη εμπορικά διαθέσιμων κυκλωμάτων για τη δημιουργία των συσκευών. Από τις υλοποιημένες συσκευές αυτές ξεχώρισαν οι συσκευές Rene και οι σύγχρονα εμπορικά διαθέσιμες MICΑ, MICΑ2 και MICΑ2Dot.

Επιπλέον, στα πλαίσια του NEST αναπτύχθηκε ένα λειτουργικό σύστημα για να διαχειρίζεται τις λειτουργίες των υλοποιημένων συσκευών. Το λειτουργικό αυτό ονομάστηκε TinyOS κυρίως λόγω των χαμηλών του απαιτήσεων σε μνήμη από το υλικό στο οποίο εγκαθίσταται. Το λειτουργικό σύστημα TinyOS συνοδεύεται από ένα σύνολο εργαλείων λογισμικού για τον προγραμματισμό των συσκευών του δικτύου με διάφορες εφαρμογές, για τη ρύθμιση και τον έλεγχο των συσκευών, και για την εξομοίωση δικτύων σε ηλεκτρονικό υπολογιστή. Τα εργαλεία αυτά αναπτύχθηκαν ως αυτόνομα επιμέρους project του NEST και ως σημαντικότερα από αυτά μπορούμε να αναφέρουμε το NesC που είναι ο compiler εφαρμογών για TinyOS, τον εξομοιωτή TOSSIM για κόμβους ασύρματου δικτύου και το εργαλείο TinyDB για την αποκόμιση πληροφορίας από κόμβους ασύρματου δικτύου με τη χρήση ερωτήσεων (queries).

Smart Dust

To project Smart Dust [32] ξεκίνησε στο τμήμα Electrical Engineering and Computer Sciences, του University of California, Berkley. Η έρευνα διεξάγεται υπό την επίβλεψή του Kris Pister σε συνεργασία με τους Joe Kahn και Bernhard Boser. Στόχος είναι οι ενσωμάτωση λειτουργιών αυτόνομης ανίχνευσης και επικοινωνίας σε συσκευές διαστάσεων της τάξης του κυβικού χιλιοστού. Στο σχεδιασμό τέτοιων συσκευών δεν υιοθετείται η RF ασύρματη επικοινωνία αλλά η μετάδοση μηνυμάτων με laser ή CCR. Ο λόγος ήταν χυρώς η έλλειψη χώρου για τα πολύπλοκα κυκλώματα ενός RF πομποδέκτη και για κεραία RF. Οι τελικές συσκευές με τις μικροσκοπικές διαστάσεις κόκκου ύμουν, οφείλουν να περικλείουν τουλάχιστο ένα αισθητήρα, τροφοδοσία ενέργειας, αναλογικά κυκλώματα, αμφίδρομη οπτική επικοινωνία και ένα προγραμματιζόμενο μικροεπεξεργαστή.

Γενικά το project οργανώθηκε σε παράλληλη έρευνα α) πάνω στις δυνατότητες οπτικής μετάδοσης από μικροσκοπικές συσκευές, β) σε πολιτικές διαχείρισης ή ανανέωσης ενεργειακών πόρων, γ) στην δημιουργία πρότυπων μικροσκοπικών συσκευών αλλά και δ) τον πειραματισμό με μάχρο-συσκευές COTS dust.

PicoNets

Η έρευνα που διεξάγεται στο Berkley Wireless Research Center σχετικά με τα PicoNets, αποβλέπει στην ασύρματη Ad hoc διαδικτύωση κόμβων έξυπνης σκόνης με άκρως ελάχιστη κατανάλωση ενέργειας [31]. Το διευθύνον στέλεχος της έρευνας είναι ο καθηγητής Jan Rabaey. Ο απώτερος στόχος είναι να επιτευχθεί η ενδυνάμωση αποδοτικών τεχνικών αποκόμισης ενέργειας από το περιβάλλον σε συσκευές με πεπερασμένους ενεργειακούς πόρους, στις οποίες η ενεργειακή κατανάλωση θα είναι μικρότερη από 100 μ Watt.

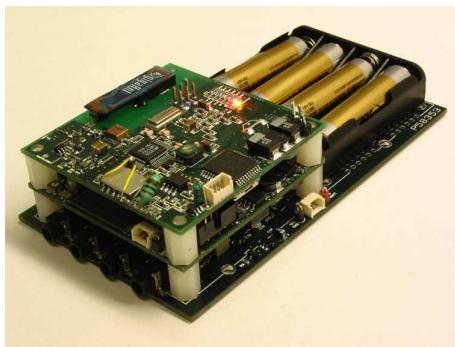
Ένας κόμβος (PicoNode) διαθέτει επαναπρογραμματίζόμενο μικροεπεξεργαστή, RF πομποδέκτη, κυκλώματα επεξεργασίας ψηφιακού σήματος και κάποιους αισθητήρες.

Στα πλαίσια του project αναπτύχθηκαν τα PicoRadio I, που είναι συσκευές από εμπορικά διαθέσιμα (COTS) συστατικά στοιχεία, ώστε να υλοποιηθεί μια πρώτη πλατφόρμα εξομοίωσης και πειραμάτων. Επίσης, το 2001 διεξήχθη έρευνα υπό την ονομασία PicoRadio II για σχεδιασμό αρχιτεκτονικών που θα ευνοούν ασύρματα πρωτόκολλα επικοινωνίας και την ενσωμάτωση των PicoNodes σε υπάρχοντα ολοκληρωμένα κυκλώματα. Το 2003 ξεκίνησε η έρευνα πάνω στο PicoRadio 3 για την δημιουργία ενός αυτόνομου ολοκληρωμένου κυκλώματος PicoNode με άκρως ελάχιστη ενεργειακή κατανάλωση.

1.5.2 μ AMPS

To project μ AMPS (μ -Adaptive Multi-domain Power Aware Sensors) ξεκίνησε στο MIT υπό την επίβλεψη του καθηγητή Anatha Chandraksan [22]. Και σε αυτή την έρευνα σκοπός είναι η δημιουργία ασύρματων δικτύων από αισθητήρες με ιδιαίτερη έμφαση στην ενεργειακή κατανάλωση και στη δυνατότητα αυτό-οργάνωσης, επανα-οργάνωσης και ευελιξίας υπό περιβαλλοντολογικές συνθήκες που δεν ελέγχονται εύκολα.

Αρχικά το ενδιαφέρον εστιάστηκε στην ανάπτυξη υλικού που θα ικανοποιούσε τις απαιτήσεις για βελτιστοποιημένη ενεργειακή διαχείριση και κατανάλωση σε όλα τα επίπεδα ιεραρχίας του συστήματος. Τα επίπεδα αυτά είναι το φυσικό επίπεδο, το επίπεδο διασύνδεσης δεδομένων, το MAC υπο-επίπεδο, το επίπεδο δικτύου και μεταφοράς πακέτων, και τα επίπεδα συνόδου, παρουσίασης και εφαρμογών.



Σχήμα 1.2: Η πρότυπη συσκευή μ AMPS-I

Σαν πρώτο επίτευγμα η έρευνα παρουσίασε τις πρότυπες συσκευές μ AMPS-1, κατασκευα-

συμένες από COTS κυκλώματα. Το 2003 η έρευνα έχει μετακινηθεί προς τη δημιουργία μ AMPS-II συσκευών, όπου η ενεργειακή κατανάλωση θα είναι της τάξης των 100 μ Watt. Επιπρόσθετα η δεύτερη αυτή γενιά συσκευών θα έχει δύο διακριτά και εξειδικευμένα ολοκληρωμένα κυκλώματα, το ένα εκ των οποίων θα αναλαμβάνει την επικοινωνία RF και το χειρισμό αναλογικού σήματος, και το άλλο όλη την ψηφιακή επεξεργασία που θα επιτελείται σε ένα κόμβο.

1.5.3 WINS

To WINS (Wireless Integrated Network Sensors (WINS)) είναι ένα project που ξεκίνησε από το 1993 στο πανεπιστήμιο UCLA, ώστε να παρέχεται η δυνατότητα προσπέλασης μέσω του διαδικτύου σε αισθητήρες, μηχανισμούς ελέγχου ή επεξεργαστές που θα είναι ενσωματωμένα σε διάφορα είδη εξοπλισμού [23]. Το όραμα που υποκινεί το WINS είναι η δημιουργία μιας γέφυρας μεταξύ του φυσικού κόσμου και του κόσμου των ηλεκτρονικών υπολογιστών, έτσι ώστε οι τελικοί χρήστες με τη βοήθεια ενός τερματικού να μπορούν να παρακολουθούν και να ελέγχουν γεγονότα και φαινόμενα σε παγκόσμια κλίμακα.

Το project αποτελεί εξέλιξη προηγούμενης έρευνας ύπο την ονομασία LWIM (Low Power Wireless Integrated Microsensors), η οποία εστίαζε στην σχεδίαση νέων συστημάτων ασύρματης επικοινωνίας και μικροαισθητήρων με χαμηλό κόστος και χαμηλή κατανάλωση ενέργειας [29].

Το 1997 έγινε μια επίδειξη της λειτουργίας ενός πρότυπου δικτύου WINS με multi-hop δρομολόγηση, ενώ λίγο αργότερα ξεκίνησε το πρόγραμμα WINS - AWAIRS. To AWAIRS (Adaptive Wireless Arrays for Interactive Reconnaissance, Surveillance and Target Acquisition in Small Unit Operations) αναπτύσσεται σε συνεργασία του UCLA με το Rockwell Science Center και βασικός στόχος του είναι να παρέχει τα δικτυακά πρωτόκολλα και τους κατανευμημένους αλγορίθμους για επεξεργασία δεδομένων στους κόμβους, ώστε να εκτελούνται με ευελιξία και αποδοτικότητα οι λειτουργίες που απαιτούνται στα δίκτυα αισθητήρων [26].

1.5.4 SENSONET

To SensoNet είναι project του Broadband and Wireless Networking laboratory στο Georgia Institute of Technology υπό την επίβλεψη του Dr. Ian Akyildiz [25]. To project βασίζεται στη διαπίστωση ότι τα υπάρχοντα πρωτόκολλα για τα γενικής φύσης Ad hoc δίκτυα είναι ανεπαρκή και μη αποδοτικά για να εφαρμοστούν στα δίκτυα αισθητήρων.

Έτσι ο κεντρικός στόχος του project είναι να προτείνει μια νέα ιεραρχία πρωτοκόλλων εξειδικευμένα για τις ασύρματες επικοινωνίες στα δίκτυα αισθητήρων καθώς και τεχνικές που θα καθιστούν περισσότερο αξιόπιστη τη λειτουργία των δικτύων αυτών. Η νέα αρχιτεκτονική ασύρματης επικοινωνίας θα εξασφαλίζει αξιόπιστη δρομολόγηση πακέτων στο δίκτυο, αποδοτική διαχείριση της ενέργειας που διαθέτει ο κάθε κόμβος και κατανεμημένη λειτουργία του δικτύου. Επίσης ερευνώνται θέματα όπως εφαρμογή ειδικού MAC πρωτοκόλλου για την αποφυγή συγκρούσεων στις ασύρματες μεταδόσεις, τεχνικές error control και συγχρονισμός των κόμβων στα δίκτυα, ώστε να επιτυγχάνεται αποδοτική συγχώνευση πληροφορίας στην πορεία προς το κέντρο ελέγχου.

1.5.5 SCADDS

Η ονομασία SCADDS είναι συντομογραφία για το Scalable coordination architectures for deeply distributed and dynamic systems. Το SCADDS [27] είναι ένα ερευνητικό πρόγραμμα που διεξάγεται στο USC/ISI υπό τη χορηγία της DARPA και αποτελεί μέρος του προγράμματος Sensor Information Technology (SenSIT). Οι κύριοι ερευνητές που ασχολούνται στο SCADDS είναι η καθ. Deborah Estrin, ο καθ. Ramesh Govindan και ο καθ. John Heidemann. Το project SCADDS ερευνά αλγορίθμους συνεργασίας με καλά χαρακτηριστικά κλιμάκωσης, για κατανεμημένα και δυναμικά συστήματα, όπως τα ασύρματα δίκτυα αισθητήρων. Σε αυτά τα δίκτυα το ενδιαφέρον δεν είναι η ανακάλυψη των πληροφοριών που διαθέτει ένας συγκεκριμένος κόμβος του δίκτυου, αλλά γενικότερα υπάρχει ενδιαφέρον για συγκεκριμένη πληροφορία που μπορούν να παράγουν οι κόμβοι του δίκτυου, ανεξάρτητα από το ποιος κόμβος παρήγαγε αρχικά την πληροφορία. Έτσι ο κάθε κόμβος δεν έχει ανάγκη από μοναδική ταυτοποίηση.

Τα κεντρικά σημεία της έρευνας είναι οι αλγόριθμοι directed diffusion και αλγόριθμοι προσαρμοζόμενης πιστότητας. Επίσης το SCADDS ασχολείται με θέματα τοπικότητας των κατανεμημένων αλγορίθμων, συγχρονισμού, αυτό-οργάνωσης, πρωτοκόλλων επικοινωνίας RF, και MAC ζητήματα.

1.5.6 Συνεισφορά για την έρευνα σε Smart Dust δίκτυα

Στα πλαίσια αυτής της διπλωματικής πέρα από μια θεωρητική επισκόπηση των επιτευγμάτων των τομέα των Smart Dust, έγινε και πρόσθετη εργασία ως συνεισφορά στην έρευνα στον τομέα αυτό. Τα επιτεύγματα της εργασίας μας, αν και υλοποιημένα μέσα σε στενά χρονικά περιθώρια, πιστεύουμε ότι είναι αρκετά αξιόλογα και σίγουρα αποτελούν ένα βήμα για την οργάνωση και την παραπέρα προώθηση της έρευνας για τα δίκτυα έξυπνης σκόνης.

Έτσι έχει υλοποιηθεί ένας εξομοιωτής (simulator), που υποστηρίζει την υλοποίηση και εξομίωση πρωτοκόλλων επικοινωνίας για τα δίκτυα έξυπνης σκόνης. Ο εξομοιωτής αυτός επιτρέπει την συγκριτική αξιολόγηση δύο ή περισσότερων πρωτοκόλλων μέσω της εφαρμογής τους σε παρόμοια μοντέλα δίκτυων και ύπο παρόμοιες συνθήκες. Ειδική έμφαση κατά το σχεδιασμό της αρχιτεκτονικής του εξομοιωτή δόθηκε στη δυνατότητα εξομοίωσης δίκτυων μεγάλης κλίμακας όσον αφορά τόσο το χώρο ανάπτυξης τους όσο και το πλήθος των κόμβων που τα στελεχώνουν. Τονίζουμε εδώ ότι ο εξομοιωτής μας μπορεί να εξομοιώνει δίκτυα διαστάσεων μέχρι και $1500 \text{ m} \times 1500 \text{ m}$ με μεγάλη ευκολία και λογικές απαιτήσεις σε υπολογιστική υποδομή, σε αντίθεση με υπάρχοντες καθιερωμένους εξομοιωτές (όπως ο ns2) που έχουν μεγάλες απαιτήσεις σε υπολογιστικούς πόρους για την εξομοίωση δίκτυων μόλις $200 \text{ m} \times 200 \text{ m}$. Επιπρόσθετα ο εξομοιωτής μας ενσωματώνει μηχανισμούς δημιουργίας πολλαπλών συμβάντων προς διάδοση σε ένα δίκτυο έξυπνης σκόνης, ώστε να προσεγγίζει κατά το δυνατό ένα πραγματικό σενάριο ανάπτυξης τέτοιου δίκτυου. Τέλος τον εξομοιωτή μας διαχρίνει και η δυνατότητα ενδυνάμωσης ενός εγγενούς μηχανισμού εξοικονόμησης ενέργειας (μηχανισμός SLEEP - AWAKE) που προσφέρεται για χρήση από τα εξομοιούμενα πρωτόκολλα.

Για τους σκοπούς της οπτικοποίησης και καλύτερης θεωρησης της εξέλιξης των πρωτοκόλλων στα δίκτυα Smart Dust, υλοποιήθηκε και ένας οπτικός προσομοιωτής (animator). Ο animator μπορεί να χρησιμοποιηθεί για την εξακρίβωση της ορθής λειτουργίας των υλο-

ποιημένων πρωτοκόλλων αλλά και για εκπαιδευτικούς σκοπούς παρουσίασης της εφαρμογής πρωτοκόλλων σε δίκτυα Smart Dust. Διαθέτει ένα φιλικό περιβάλλον διεπαφής με τον τελικό χρήστη και του παρέχει τη δυνατότητα να ρυθμίσει τις παραμέτρους εξομοίωσης που αφορούν τόσο τα χαρακτηριστικά του δικτύου όσο και του εφαρμοζόμενου πρωτοκόλλου. Έπειτα με ένα εύληπτο και καλαίσθητο οπτικά τρόπο αναπαριστάται γραφικά η πρόοδος της εξομοίωσης, παρέχοντας παράλληλα στον τελικό χρήστη διαδραστικότητα (interactivity), καθώς μπορεί αυτός άμεσα να καθορίζει ποιοι κόμβοι θα ανιχνεύσουν γεγονότα προς διάδοση.

Με τη βοήθεια του υλοποιημένου εξομοιωτή έγινε μια πρώτη συγχριτική θεώρηση και αξιολόγηση δύο αρκετά διαδεδομένων πρωτοκόλλων. Πρόκειται για τα πρωτόκολλα PFR και TEEN, υλοποιημένα όμως και επαυξημένα με νέες δυνατότητες, ώστε να είναι κατά το δυνατό πιο αποδοτικά. Η διαδικασία της υλοποίησης του εξομοιωτή μας έφερε αντιμέτωπους με πολλά πρακτικά ζητήματα υλοποίησης και εφαρμογής πρωτοκόλλων σε δίκτυα Smart Dust. Τα προβλήματα που προέκυψαν και η αναζήτηση της λύσης τους μας προσέφεραν σε βάθος γνώση των ιδιαιτεροτήτων των δικτύων αυτών και των αναγκαίων χαρακτηριστικών που πρέπει να ενσωματώνουν τα πρωτόκολλα για την αποδοτική εφαρμογή τους σε τέτοια δίκτυα. Έτσι δεδομένων των δύο ήδη γνωστών από τη θεωρία πρωτοκόλλων PFR και TEEN, αποφασίσαμε να ενσωματώσουμε σε αυτά κάποιες από τις ιδέες μας για την περισσότερο αποδοτική λειτουργία τους. Ειδικότερα στο πρωτόκολλο PFR, προσθέσαμε τη δυνατότητα χειρισμού πολλαπλών ταυτόχρονα διαδιδόμενων γεγονότων, όπως και τον μηχανισμό εξοικονόμησης ενέργειας SLEEP - AWAKE. Στο TEEN υλοποιήσαμε μια πραγματικά ιεραρχική δομή τεσσάρων επιπέδων για τη διάδοση πληροφορίας προς το κέντρο ελέγχου, καθώς στο αρχικό πρωτόκολλο η διάδοση γινόταν μεταξύ μόνο δύο επιπέδων ιεραρχίας.

Πέρα από τις εξομοιώσεις, πραγματοποιήθηκε και ένα πραγματικό πειραματικό δίκτυο από συσκευές COTSdust που αποτελούν πρότυπα για τις μελλοντικές συσκευές Smart Dust. Οι πραγματικές συσκευές που είχαμε στη διάθεση μας ήταν τα MICA motes, για τα οποία αξιολογήσαμε τις δυνατότητες τους και πειραματιστήκαμε με κάποιες εφαρμογές που ανέδειξαν τα κεντρικά χαρακτηριστικά της πλατφόρμας των MICA motes. Αναπτύξαμε έπειτα ένα πραγματικό πειραματικό δίκτυο με τα MICA motes να αποτελούν τους ασύρματους κόμβους - αισθητήρες. Στο δίκτυο αυτό μπορέσαμε να αποσπάσουμε με προγραμματιζόμενο ρυθμό πληροφορίες από τους κόμβους, για τις μετρήσεις των αισθητήρων τους, τα ενεργειακά τους αποθέματα και την τοπολογική τους διάταξη! Η επιτυχής υλοποίηση και λειτουργία του πρότυπου πραγματικού δικτύου έξυπνης, μας παρείχε την αναγκαία σύνδεση της θεωρητικής και εν μέρει εξιδανικευμένης ανάλυσης των δικτύων Smart Dust, που είχε προηγηθεί, με την τελική πραγμάτωση των όσων υποσχόταν η θεωρία. Έτσι είχαμε μια πρώτη επαφή με τα προβλήματα και τα ζητήματα που προκύπτουν σε ένα πραγματικό πειραματικό δίκτυο έξυπνης σκόνης, αλλά και με τις εντυπωσιακές και μοναδικές δυνατότητες που αυτά προσφέρουν.

Τέλος για τους σκοπούς δημοσίευσης της εργασίας μας στο διαδίκτυο, υλοποιήθηκε μια ηλεκτρονική σελίδα (web page) με πληροφορίες για τα δίκτυα έξυπνης σκόνης και κάποια από τα σημαντικότερα πρωτόκολλα γι' αυτά, για την εξέλιξη των στόχων που έθεσε η διπλωματική και για τα συμπεράσματα που έχουν εξαχθεί από τα πειράματα που έγιναν.

Κεφάλαιο 2

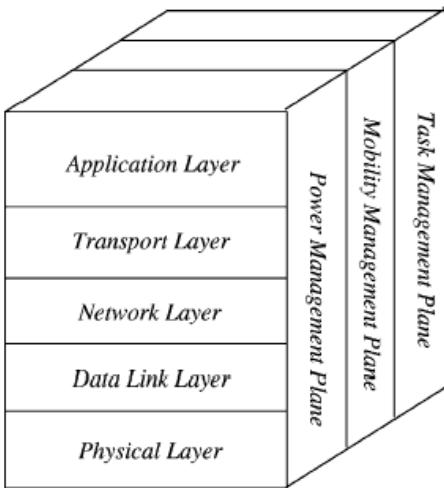
Προτεινόμενα πρωτόκολλα δικτύων έξυπνης σκόνης

Τα πρωτόκολλα που έχουν προταθεί κατα καιρούς για τη δρομολόγηση πληροφορίας στα δίκτυα Smart Dust συγκλινουν όλα σε κάποιους κοινούς στόχους. Οι σημαντικότεροι από αυτούς είναι η επίτευξη χαμηλής ενεργειακής κατανάλωσης στο δίκτυο, η αξιόπιστη διάδοση και παράδοση χρίσματος πληροφορίας στο κέντρο ελέγχου και οι κατα το δυνατό μικρές καθυστερήσεις διάδοσης. Για τα ασύρματα δίκτυα αισθητήρων έχουν ήδη προταθεί αρκετά εξειδικευμένα πρωτόκολλα επικοινωνίας. Σε αυτά τα πρωτόκολλα μπορούμε να διαχρίνουμε κοινά χαρακτηρηστικά, αλλά και σημαντικές διαφορές ικανές να τα κατατάσσουν σε διαφορετικές κατηγορίες.

2.1 Εισαγωγή στα πρωτόκολλα

Τα περισσότερα από τα πρωτόκολλα δικτύων έξυπνης σκόνης είναι εξειδικευμένα για συγκεκριμένο τύπο εφαρμογών (application specific). Αυτό συντελεί στην καλύτερη διαχείριση των πόρων του δικτύου, καθώς είναι γνωστές εκ των προτέρων οι συνθήκες, υπό τις οποίες το δίκτυο θα λειτουργεί. Έτσι μπορούν να ορίζονται στενότερα πλαίσια όπου θα κυμαίνονται οι πιο σημαντικοί παράμετροι λειτουργίας για μια συγκεκριμένη εφαρμογή, ενώ οι υπόλοιπες παράμετροι να έχουν λιγότερο αυστηρούς περιορισμούς.

Ένα άλλο χαρακτηριστικό των πρωτοκόλλων είναι ότι επικεντρώνονται στην πληροφορία (data centric) που πρέπει να φτάσει στο κέντρο ελέγχου και όχι στους ίδιους τους κόμβους που την παράγουν. Με άλλα λόγια, το κέντρο ελέγχου δε απευθύνεται στους κόμβους ένα προς ένα για τις μετρήσεις που λαμβάνουν, παρά διαδίδει το “ενδιαφέρον” (interest) του για μετρήσεις συγκεκριμένων μεγεθών - χαρακτηριστικών που υπερβαίνουν ή είναι χαμηλότερες από κάποια τιμή. Το κάθε τέτοιο “ενδιαφέρον” είναι ουσιαστικά ένα ζεύγος από ένα μέγεθος - χαρακτηριστικό και μια τιμή - όριο (threshold) που καθορίζει πότε μια μέτρηση του μεγέθους - χαρακτηριστικού είναι χρίσμη προς μετάδοση. Έτσι ουσιαστικά ο κάθε κόμβος έχοντας λάβει από το κέντρο ελέγχου τα ζεύγη αυτά, γνωρίζει πότε θα μεταδώσει πληροφορία χωρίς να χρειαστεί να ερωτηθεί ως ξεχωριστή οντότητα. Η μέθοδος αυτή λέγεται attribute based addressing, καθώς τα ζεύγη χαρακτηριστικού και τιμής που υποδεικνύουν το είδος της ανίχνευσης που θα πραγματοποιηθεί, είναι ο μόνος τρόπος διευθυνσιοδότησης



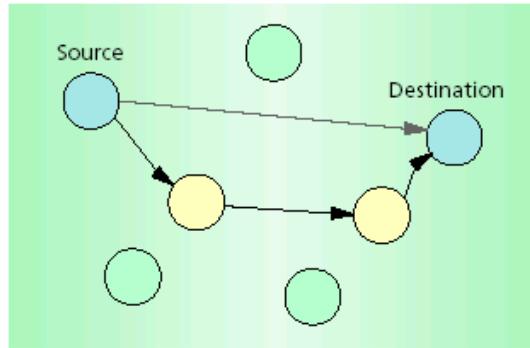
Σχήμα 2.1: Η στοιβά των πρωτοκόλλων για τα δίκτυα Smart Dust

(addressing) των κόμβων.

Προαιρετικά τα πρωτόκολλα μπορούν να ενσωματώνουν ένα μηχανισμό ελέγχου της κίνησης στο δίκτυο (traffic flow) ο οποίος ονομάζεται πολύπλεξη/συγχώνευση δεδομένων (data aggregation). Η συγχώνευση δεδομένων υπαγορεύει ότι κατά τη διάδοση πολλαπλών μηνυμάτων προς το κέντρο ελέγχου, κάποιοι ενδιάμεσοι κόμβοι δε προωθούν άμεσα τα μηνύματα που λαμβάνουν. Αντίθετα μπορούν να τα κρατούν για μικρό χρονικό διάστημα, να τα επεξεργάζονται και τα συγχωνεύουν με νέα εισερχόμενα μηνύματα και να απορρίπτουν τυχόν αντίγραφα που ξαναλαμβάνονται. Πρέπει να σημειωθεί ότι η συγχώνευση των μηνυμάτων εισάγει πρόσθετη καθυστέρηση της διάδοσης της πληροφορίας στο δίκτυο, αλλά αν γίνει σωστά μειώνει δραματικά την κίνηση μηνυμάτων στο δίκτυο, τις πιθανές συγκρούσεις και επιβραδύνει την ενεργειακή εξάντληση του. Η σωστή συγχώνευση επιτυγχάνεται μόνο όταν η συγχωνευμένη πληροφορία έχει μικρότερο μέγεθος από τις επιμέρους συνιστώσες πληροφορίες.

Όσον αφορά τη μέθοδο δρομολόγησης μηνυμάτων προς το κέντρο ελέγχου τα πρωτόκολλα γενικά ακολουθούν τη διάδοση σε πολλαπλά βήματα (**multihop**) ως τη πιο αποδοτική μέθοδο διάδοσης πληροφορίας έναντι της απευθείας μετάδοση. Η μετάδοση multihop έχει πλεονεκτήματα που την καθιστούν περισσότερη κατάλληλη για χρήση στα δίκτυα έξυπνης σκόνης. Ένα από αυτά είναι ότι δεν απαιτείται να απουσιάζουν εμπόδια από τη νοητή γραμμή που ενώνει το κόμβο - πηγή και το κέντρο ελέγχου. Η μετάδοση σε πολλά βήματα είναι πολύ ευέλικτη και μπορεί να οδηγήσει αξιόπιστα τη διαδιδόμενη πληροφορία γύρω από τυχόν εμπόδια μέχρι το κέντρο ελέγχου. Επιπλέον στη multihop μετάδοση σε δίκτυα επαρκούς πυκνότητας γίνονται κατά κανόνα μεταδόσεις σε μικρή απόσταση, οπότε περιορίζεται δραστικά η εμφάνιση φαινομένων ανάκλασης ή εξασθένισης σήματος. Πρέπει επίσης να σημειωθεί ότι στη multihop μετάδοση είναι αρχετή η τοπική γνώση του δικτύου από ένα κόμβο και η γενική κατεύθυνση του κέντρου ελέγχου. Η πληροφορία σε μια multihop δρομολόγηση, μπορεί να μεταδίδεται είτε με μεθόδους broadcast είτε με μεθόδους μετάδοσης προς γειτονικό κόμβο - σημείο. Αντίθετα στην απευθείας εκπομπή η μετάδοση πρέπει να γίνεται με οπτικά

μέσα, τα οποία πρέπει να στοχεύουν επακριβώς στη θέση του Sink, ώστε να αποφεύγεται η επιβάρυνση του υπόλοιπου δικτύου με πλεονάζουσες λήψεις μηγυμάτων.



Σχήμα 2.2: Αντιδιαστολή multihop και απευθείας μετάδοσης

Μια από τις πρωταρχικές διακρίσεις των πρωτοκόλλων για τα δίκτυα αισθητήρων είναι σε παθητικά (**proactive**) και αντιδραστικά (**reactive**) [15]. Ονομάζουμε παθητικά τα δίκτυα όπου οι κόμβοι ανιχνεύουν συνεχώς το περιβάλλον τους και στέλνουν διαρκώς πληροφορία σχετικά με κάποιο μέγεθος - χαρακτηριστικό για το οποίο το κέντρο ελέγχου δήλωσε ενδιαφέρον. Σε αυτά τα δίκτυα αγνοείται το threshold που συνοδεύει το κάθε μέγεθος - χαρακτηριστικό για το οποίο εκδίδει ένα “ενδιαφέρον” ο Sink. Περισσότερη σημασία έχει η ενημέρωση σε τακτικό χρόνο συγκεκριμένων μετρήσεων παρά ο χρόνος διάδοσης μιας μέτρησης. Η κίνηση (traffic) όμως των μηνυμάτων στα δίκτυα αυτά είναι δικαιολογημένα αυξημένη και τα επιβαρύνει ενεργειακά. Ως εφαρμογή μπορούμε να αναφέρουμε την παρακολούθηση μηχανημάτων για τυχόν παρουσίαση βλάβης, ή την καταγραφή των τιμών θερμοκρασίας σε ένα τόπο. Τα παθητικά δίκτυα λόγω των πολλών αποστελλόμενων μηνυμάτων έχουν σαρώς μικρότερο χρόνο ζωής από τα αντιδραστικά. Στα αντιδραστικά δίκτυα, η μετάδοση πληροφορίας συμβαίνει μόνο όταν η τιμή ενός μεγέθους - χαρακτηριστικού την οποία μετράει ένας κόμβος - αισθητήρας, περάσει το κατώφλι που έθεσε ο Sink σε κάποιο “ενδιαφέρον” του. Η μετάδοση της πληροφορίας σταματάει όταν η τιμή του παρατηρούμενου μεγέθους επανέλθει σε “αδιάφορο” εύρος τιμών. Στα δίκτυα αυτά είναι κρίσιμος ο χρόνος διάδοσης ενός μηνύματος προς το Sink και πρέπει να διατηρείται σε χαμηλά επίπεδα. Μερικές από τις εφαρμογές για τα αντιδραστικά δίκτυα είναι η ανίχνευση πυρκαγιάς ή κάποιας εισβολής σε παρατηρούμενο χώρο.

2.2 Το μοντέλο των δικτύων

Η μοντελοποίηση των Smart Dust δικτύων χρησιμοποιείται για τη θεωρητική ανάλυση και την ανάπτυξη πρωτοκόλλων που αναφέρονται σε αυτά. Περικλείει τη διάταξη των κόμβων στο χώρο, τη θέση του κέντρου ελέγχου (Sink) και τις δυνατότητες επικοινωνίας που έχουν διαθέσιμες οι κόμβοι και το Sink. Ως γενικευμένο μοντέλο για τα δίκτυα έξυπνης σκόνης μπορούμε να θεωρήσουμε μια διασπορά ενός μεγάλου αριθμού από συσκευές - αισθητήρες, οι οποίες έχουν πολύ μικρές διαστάσεις. Το μικρό τους μέγεθος είναι υπεύθυνο για την ονομασία grain particles (σωματίδια διαστάσεως κουκίδας).

Οι κόμβοι - αισθητήρες θεωρούνται όμοιοι μεταξύ τους και ισότιμοι όσον αφορά τις δυνατότητες και τις λειτουργίες που επιτελούν. Διαθέτουν όλοι συσκευή για λήψη και αποστολή μηνυμάτων, αισθητήρες για ανίχνευση του περιβάλλοντος χώρου και επεξεργαστή για ανάλυση των δεδομένων που λαμβάνουν ή ανιχνεύουν. Επιπρόσθετα διαθέτουν το ίδιο ποσό αρχικής ενέργειας το οποίο είναι πεπερασμένο και ξοδεύεται σταδιακά με κάθε λειτουργία που επιτελούν. Οι κόμβοι είναι ακίνητοι στην αρχική θέση που καταλαμβάνουν.

Οι βασικές λειτουργίες των κόμβων είναι η συλλογή πληροφοριών από τον περιβάλλοντα χώρο μέσω των αισθητήρων τους και η προώθηση πληροφορίας προς το κέντρο ελέγχου. Ο κάθε κόμβος μπορεί να έχει ένα σύνολο από διαφορετικούς αισθητήρες για να αντιλαμβάνεται ποικίλα φαινόμενα που διαδραματίζονται στο άμεσο περιβάλλον του. Επιπρόσθετα ο κάθε κόμβος αναλαμβάνει να δημιουργήσει πληροφορίες σχετιζόμενες με τις μετρήσεις των αισθητήρων που διαθέτει, αλλά και να δρομολόγησει κάθε εισερχόμενη πληροφορία προς το κέντρο ελέγχου. Η κύρια διαθέσιμη μέθοδος μετάδοσης στους κόμβους είναι η διάδοση μηνυμάτων με τεχνική broadcast σε συγκεκριμένη ακτίνα R . Κατά κανόνα όλοι οι ενεργοί κόμβοι μέσα στην ακτίνα R θα λάβουν το μεταδιόδευτο μήνυμα. Μια παραλλαγή αυτού του μοντέλου επιτρέπει τη διάδοση με τεχνική broadcast υπό γωνία. Αυτό σημαίνει ότι το μεταδιόδευτο μήνυμα θα λάβουν μόνο οι κόμβοι που βρίσκονται στο κυκλικό τομέα που ορίζεται από την ακτίνα R και τις επίκεντρες γωνίες $+a$ και $-a$ ως προς μια ευθεία αναφοράς.

Κάποια πρωτόκολλα προβλέπουν ότι οι κόμβοι μπορούν να χρησιμοποιούν οπτικά μέσα επικοινωνίας (πχ laser) για κατεύθυνόμενη μετάδοση προς σημείο - κόμβο. Οι κόμβοι που εξαντλούνται ενεργειακά καθίστανται ανενεργοί και δε συμμετέχουν στην ανίχνευση ή στη διάδοση πληροφορίας. Ουσιαστικά δηλαδή αποκόπτονται από το δίκτυο.

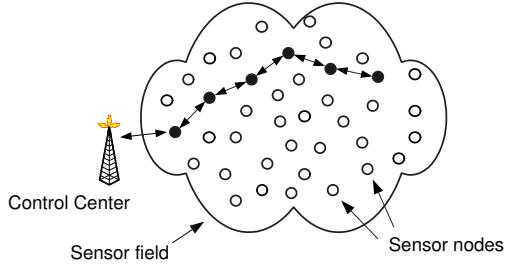
Η μετάδοση σε κάθε περίπτωση γίνεται με απώτερο σκοπό να φτάσει η πληροφορία στο κέντρο ελέγχου. Το κέντρο ελέγχου έχει απεριόριστους υπολογιστικούς και ενεργειακούς πόρους και αντιρροσωπεύει τις αρχές εκείνες, που είναι υπεύθυνες για τη ανάπτυξη του δικτύου και ενδιαφέρονται να αποκομίσουν πληροφορίες από τη λειτουργία του.

Τέλος ας σημειωθεί ότι είναι πολλές φορές απαραίτητο οι κόμβοι να γνωρίζουν την τοποθεσία όπου βρίσκονται καθώς συχνά η πληροφορία που ζητάει το κέντρο ελέγχου μπορεί να σχετίζεται με τον τόπο εκδήλωσης κάποιου φαινομένου.

2.2.1 Μοντέλο πηγής - Sink

Αυτό το μοντέλο είναι αυτό που υιοθετείται στην πλειονότητα των περιπτώσεων για την ανάπτυξη πρωτοκόλλων διάδοσης πληροφορίας σε δίκτυα έξυπνης σκόνης [12, 13, 11, 6, 15]. Οι κόμβοι του δικτύου έξυπνης σκόνης είναι διασκορπισμένοι σε μια επίπεδη επιφάνεια και αποτελούν σημεία της επιφάνειας αυτής χωρίς να λαμβάνεται υπόψη το μέγεθος τους. Η πυκνότητα του δικτύου θεωρείται αρκετά μεγάλη και ο κάθε κόμβος έχει ακτίνα μετάδοσης R συνήθως αρκετά μικρότερη από τη διάμετρο του δικτύου. Το κέντρο ελέγχου αναπαριστάται με ένα σημείο μέσα στο πεδίο, όπου αναπτύσσεται το δίκτυο.

Προκειμένου να μπορούν οι κόμβοι να εκπέμψουν μηνύματα προς το κέντρο ελέγχου, θα πρέπει να προηγηθεί της εφαρμογής οπουδήποτε πρωτοκόλλου, μια φάση όπου οι κόμβοι θα ανακαλύψουν την κατεύθυνση όπου κείται το Sink. Στη φάση αυτή ανταλλάσσονται μηνύματα ελέγχου μεταξύ των κόμβων, συνήθως με μηχανισμούς πλημμύρας (flooding) του δικτύου. Εκτός από τη κατεύθυνση του Sink σε αυτή τη φάση ο κάθε κόμβος μπορεί να

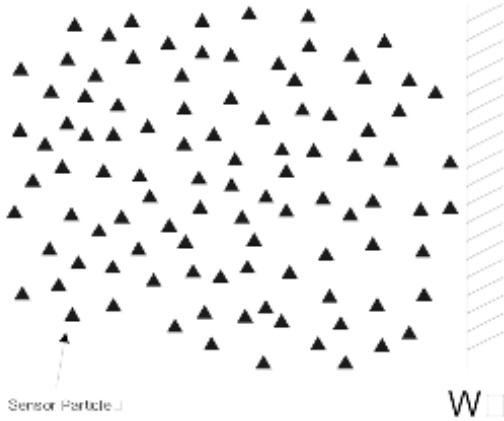


Σχήμα 2.3: Μοντέλο πηγής - Sink

μαθαίνει τη σχετική απόσταση του από τους γειτονικούς κόμβους του καθώς και τη θέση τους ως προς την ευθεία που ενώνει τον κόμβο με το Sink. Η πληροφορία αυτή μπορεί να είναι χρήσιμη για αρκετά πρωτόκολλα δρομολόγησης πληροφορίας.

2.2.2 Μοντέλο πηγής και Τείχους

Το μοντέλο αυτό το εισήγαγαν οι I. Χατζηγιαννάκης, Σ. Νικολετσέας και Π. Σπυράκης στην εργασία [5]. Η σημαντική διαφορά που παρουσιάζει σε σχέση με το προηγούμενο μοντέλο έγκειται στο ότι αντί για ένα απλό κέντρο ελέγχου - σημείο, υποθέτουμε ότι υπάρχει ένα μεγάλου μήκους τείχος το οποίο αντιπροσωπεύει το κέντρο ελέγχου. Οποιαδήποτε αποστολή μηνύματος προς το τείχος από κόμβο που είναι εντός απόστασης R από το τείχος, σημαίνει ταυτόχρονα τη λήψη της πληροφορίας από το κέντρο ελέγχου. Μια πρακτική υλοποίηση αυτού του τείχους θα ήταν μια συστοιχία από κέντρα ελέγχου που συντονισμένα θα αποτελούσαν ένα μεγάλο σε μήκος κέντρο ελέγχου. Εναλλακτικά θα μπορούσε να υλοποιηθεί από ένα κινητό κέντρο ελέγχου, δηλαδή μια σειρά από οχήματα που κινούνται με μεγάλη ταχύτητα σε ευθεία κατά μήκος του υποτιθέμενου τείχους.



Σχήμα 2.4: Μοντέλο τυχαίας κατανομής με κέντρο ελέγχου ένα Wall

Στην ανάλυση πρωτοκόλλων μπορούμε να υποθέτουμε άπειρο μήκος για το τείχος, ώστε να απλοποιηθεί η θεωρητική ανάλυση τους. Προσεγγιστικά ένα τείχος αρκετά μεγάλου μήκους

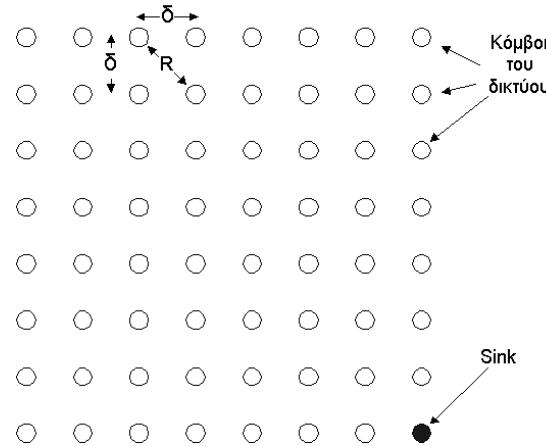
αρκεί για να ισχύουν τα αποτελέσματα της θεωρητικής ανάλυσης.

2.2.3 Μοντέλο κατανομής LATTICE

Το μοντέλο LATTICE, το οποίο εισήγαγαν οι I. Χατζηγιαννάκης, T. Δημητρίου, Σ. Νικολετσέας και Π. Συπράκης στην εργασία [6], παρουσιάζει την ιδιαιτερότητα ότι προϋποθέτει το ποιοθέτηση των κόμβων στο δίκτυο από άνθρωπο ή ρομπότ. Ουσιαστικά το μοντέλο προβλέπει ότι οι κόμβοι πρέπει να διατάσσονται σε συγκεκριμένες θέσεις πάνω σε μια δισδιάστατη επιφάνεια, ώστε να σχηματίζουν ένα είδους πλέγμα (grid).

Ο βασικός σκοπός της θεώρησης αυτού του μοντέλου είναι η δυνατότητα που παρέχει για την κατά το δυνατό ομοιόμορφη παρακολούθηση της καλυπτόμενης επιφάνειας. Το μειονέκτημα αυτού του μοντέλου είναι ότι στηρίζεται στην ύπαρξη μιας φάσης λεπτομερούς σχεδίασης και εγκατάστασης του δικτύου, η οποία κατά κανόνα απουσιάζει από τα δίκτυα Smart Dust. Ωστόσο, σε αρκετές από τις εφαρμογές των Smart Dust δικτύων, όταν είναι δυνατή η τοποθέτηση των κόμβων ένα προς ένα, ο σχηματισμός LATTICE παρουσιάζει αρκετά πλεονεκτήματα έναντι των τυχαίων διατάξεων ακριβώς λόγω της ομοιόμορφης κατανομής των κόμβων στο πεδίο. Όπως άλλωστε παρατηρείται και στο [6], ακόμα και στα δίκτυα με τυχαία διασπορά των κόμβων και με επαρκή πυκνότητα, μπορεί να βρεθεί ένας υπο-γράφος του δικτύου που ακολουθεί τη διάταξη LATTICE.

Ο σχηματισμός του πλέγματος στην κατανομή LATTICE ακολουθεί την εξής λογική: Οι κόμβοι τοποθετούνται σε μια δισδιάστατη επιφάνεια, όπου ο καθέ κόμβος ανήκει σε μια οριζόντια σειρά κόμβων με τους γειτονικούς κόμβους να είναι ισαπέχοντες, και σε μια κάθετη σειρά κόμβων, όπου πάλι οι γειτονικοί κόμβοι ισαπέχουν. Ένας $n \times n$ LATTICE σχηματισμός περιέχει n^2 κόμβους. Συμβολίζουμε με δ την οριζόντια (και κάθετη) απόσταση μεταξύ δύο κόμβων που ανήκουν σε ίδια ευθεία στη διάταξη LATTICE. Τότε συνήθως θεωρούμε ότι $R = 2\sqrt{\delta}$ είναι η ακτίνα μετάδοσης του κάθε κόμβου, και με εξαίρεση τους ακραίους κόμβους στο σχηματισμό, όλοι οι υπόλοιποι κόμβοι έχουν οκτώ γειτονικούς, που βρίσκονται εντός της ακτίνας R . Στην εικόνα 2.5 φαίνεται ένας 8×8 LATTICE σχηματισμός.



Σχήμα 2.5: Μοντέλο LATTICE

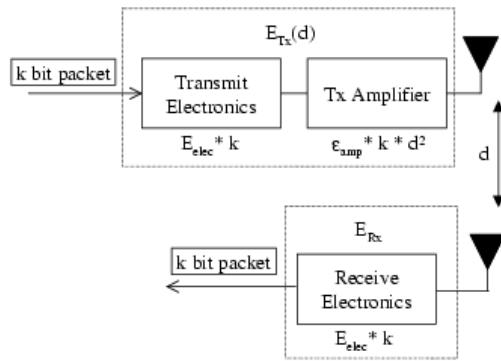
Το κέντρο ελέγχου είναι και αυτό ένα σημείο του LATTICE σχηματισμού και είναι κατά

κανόνα ακίνητο. Εναλλακτικά μπορεί να έχει την δυνατότητα να κινείται για να βρεθεί κοντά σε εξαντλημένους κόμβους του δικτύου και να συλλέξει πληροφορία που δε μπορεί να προωθηθεί παραπέρα στο δίκτυο. Στο σχηματισμό LATTICE υποθέτουμε ότι όλοι οι κόμβοι γνωρίζουν το σημείο όπου βρίσκεται ο Sink, αλλά και τη δική τους θέση στο πλέγμα.

2.2.4 Μοντέλο κατανάλωσης ενέργειας

Για την κατανάλωση ενέργειας σε κάθε κόμβο, πρέπει να λάβουμε υπόψη τις λειτουργίες ανίχνευσης του περιβάλλοντος, επεξεργασίας πληροφορίας, μετάδοσης και λήψης μηνύματος που επιτελεί ο κόμβος.

Γενικά η ενέργεια για την επεξεργασία πληροφορίας συμψηφίζεται με την ενέργεια που δαπανάται για την ανίχνευση και θεωρείται ίση με ένα ποσό E_{elec} για δεδομένο χρονικό διάστημα [11].



Σχήμα 2.6: Μοντέλο κατανάλωσης ενέργειας στον πομπόδέκτη RF

Για την αποστολή ενός μηνύματος k -bit σε απόσταση d υποθέτουμε ότι απαιτείται ενέργεια που περιγράφεται από τη σχέση

$$E_{Tx}(k, d) = E_{elec} * k + e_{amp} * k * d^2$$

ενώ η ενέργεια που δαπανάται για τη λήψη του ίδιου μηνύματος είναι

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

Το ποσό E_{elec} είναι η ενέργεια που απαιτείται για τη λειτουργία του πομπόδέκτη (transceiver), ενώ το e_{amp} είναι η ενέργεια που απαιτείται από τον πομπό για την ενίσχυση του μεταδιδόμενου σήματος, ώστε να επιτευχθεί μια επαρκής τιμή για το λόγο σήματος προς θόρυβο (SNR). Είναι φανερό ότι η ενέργεια εκπομπής θεωρείται απευθείας ανάλογη προς το τετράγωνο της απόστασης. Με τους τύπους αυτούς ακόμα, προκύπτει ότι για μικρές αποστάσεις η ενέργεια εκπομπής και λήψης συγκλίνουν στην ίδια τιμή. Το κανάλι επικοινωνίας

θεωρείται συμμετρικό υπό την έννοια ότι η ενέργεια για τη μετάδοση από τον κόμβο A στο B είναι ίδια με την ενέργεια για μετάδοση από το B στο A.

Γενικά ένας κόμβος μπορεί να λειτουργήσει και σε κατάσταση χαμηλής ενεργειακής κατανάλωσης, σταματώντας τη λειτουργία κάποιων δαπανηρών ενεργειακά κυκλωμάτων όπως ο πομποδέκτης, ή περιορίζοντας τη λειτουργία του επεξεργαστή. Στην κατάσταση αυτή υποθέτουμε ότι η δαπάνη ενέργειας είναι αμελητέα και προσεγγιστικά μηδενική. Ωστόσο όπως έχει προαναφερθεί η επαναφορά σε κατάσταση κανονικής λειτουργίας κοστίζει πάντοτε ένα μη αμελητέο ποσό ενέργειας το οποίο θεωρείται ως κάποιο πολλαπλάσιο του ποσού E_{elec} .

2.3 Ανασκόπηση χαρακτηριστικών πρωτοκόλλων

Έχοντας παρουσιάσει τα κυρίαρχα μοντέλα που ενέχονται στον σχεδιασμό πρωτοκόλλων για τα Smart Dust δίκτυα, μπορούμε να εξετάσουμε κάποια από τα πρωτόκολλα που έχουν προταθεί. Τα πρωτόκολλα που ακολουθούν δεν είναι αποδοτικά για όλα τα είδη δικτύων ή εφαρμογών. Μάλιστα κάποια αυτά φαίνονται να είναι εντελώς ακατάλληλα για δίκτυα Smart Dust. Ωστόσο πρέπει να ληφθεί υπόψη ότι μέσα στο ευρύ φάσμα των εφαρμογών για τα δίκτυα αυτά υπάρχουν πολλές εφαρμογές με ιδιαίτερες απαιτήσεις και συνθήκες ανάπτυξης. Γι' αυτές τις εφαρμογές, ένα κατά τα φαινόμενα αναποδοτικό πρωτόκολλο, μπορεί να αποδειχθεί το πιο κατάλληλο. Επίσης κάποια από τα πρωτόκολλα που ακολουθούν περιέχουν πρωτοποριακές ιδέες και μηχανισμούς που αν μη τι άλλο μπορούν να αποσπασθούν και να χρησιμοποιηθούν στη δημιουργία νέων υβριδικών πρωτοκόλλων.

2.3.1 Flooding

Το πρωτόκολλο επικοινωνίας flooding είναι το από τα πρώτα και αρκετά απλουστευμένα πρωτόκολλα επικοινωνίας για ασύρματα δίκτυα [1]. Στον αλγόριθμο αυτό, η διάδοση της πληροφορίας προς το κέντρο ελέγχου γίνεται με τη μετάδοση της σε όλους του γείτονες του κόμβου που φέρει την πληροφορία. Έτσι κατά ένα τρόπο πλημμυρίζει το δίκτυο από μηνύματα τα οποία κάθε κόμβος προωθεί ντετερμινιστικά προς όλους τους γείτονες του. Η απουσία κάποιου είδους επεξεργασίας και συνένωσης των μηνυμάτων, ώστε από πολλά μηνύματα να παράγεται ένα αντιπροσωπευτικό, συνεπάγεται πολλές πλεονάζουσες μεταδόσεις, αρκετές συγκρούσεις και μεγάλος φόρτος στο δίκτυο ακόμα και στην περίπτωση ενός ή λίγων διακριτών πληροφοριών - συμβάντων προς μετάδοση. Συγκεκριμένα είναι δύο τα πιο βασικά προβλήματα του flooding.

- 1) **Συγκέντρωση (implosion).** Έτσι ονομάζουμε την κατάσταση όπου σε ένα κόμβο στέλνονται πολλαπλά αντίγραφα του ίδιου μηνύματος. Αυτό συμβαίνει γιατί δύο γείτονες κόμβοι A και B, μπορούν να έχουν ένα σύνολο από κοινούς γείτονες. Έτσι σε περίπτωση που ο A στείλει ένα μήνυμα στον B, τότε ο B θα λάβει επίσης το ίδιο μήνυμα και από τους υπόλοιπους κοινούς γείτονες που έχει με τον A.
- 2) **Απουσία μηχανισμού εξοικονόμησης ενέργειας.** Η λογική του πρωτοκόλλου ότι κάθε κόμβος πάντα προωθεί την πληροφορία που κατέχει, σε όλους τους γείτονες του, αγνοεί εντελώς την ιδιαιτερότητα των ασύρματων δικτύων που οφείλεται στους πεπερασμένους ενεργειακούς πόρους των κόμβων.

2.3.2 Gossiping

Ο αλγόριθμος αυτός προέρχεται από τον flooding αλγόριθμο, με τη διαφορά ότι ο κάθε κόμβος με πληροφορία προς μετάδοση, αποστέλλει την πληροφορία προς ένα μόνο τυχαία επιλεγμένο γειτονικό κόμβο [1].

Με τον τρόπο αυτό αποφεύγονται πλεονάζουσες μεταδόσεις, και η λήψη πολλαπλών αντιγράφων του ίδιου μηνύματος από ένα κόμβο, αλλά ωστόσο δεν εξασφαλίζεται ότι ο τυχαίος γείτονας θα είναι η πιο κατάλληλη επιλογή για την προώθηση της πληροφορίας. Αυτό σημαίνει ότι ο κόμβος - πομπός της πληροφορίας δε λαμβάνει υπόψη του ότι ο κόμβος - δέκτης μπορεί να μην έχει ενεργειακά αποθέματα για να προωθήσει παραπέρα την πληροφορία, ή μπορεί να βρίσκεται πιο μακριά από τον κόμβο - πομπό όσον αφορά την απόσταση του από το κέντρο ελέγχου. Επιπρόσθετα η μέθοδος αυτή ενέχει και τον κίνδυνο δημιουργίας ατέρμονων βρόχων αποστολής και λήψης μηνυμάτων μεταξύ δύο κόμβων, αλλά και αδιεξόδων στη διάδοση μηνύματος προς το κέντρο ελέγχου.

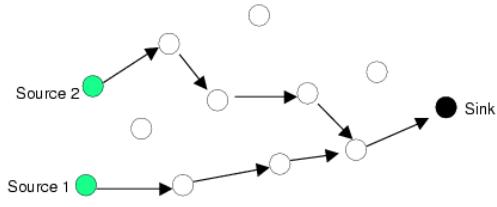
Συνοψίζοντας, με το Gossiping εξασφαλίζεται μεγαλύτερη διάρκεια ζωής για το δίκτυο από τον αλγόριθμο flooding. Ωστόσο, αυξάνονται σημαντικά τόσο ο χρόνος διάδοσης ενός μηνύματος προς το κέντρο ελέγχου όσο και το ποσοστό αποτυχίας παράδοσης μηνυμάτων στο κέντρο ελέγχου.

2.3.3 Greedy Directed Diffusion

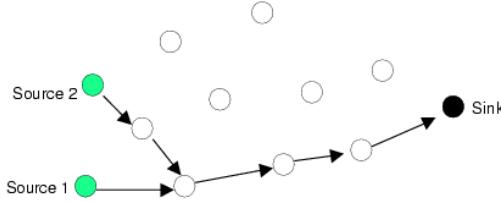
Αρκετό ενδιαφέρον παρουσιάζει η τεχνική του direct diffusion που προτάθηκε αρχικά στο [12] από τους C. Intanagonwiwat, R. Govindan και D. Estrin. Η τεχνική αυτή, δεν προτάθηκε με τη μορφή ενός γενικευμένου πρωτοκόλλου αλλά περισσότερο ως ένα παράδειγμα για διάδοση πληροφορίας σε ασύρματα δίκτυα αισθητήρων που εξυπηρετούν μια εξειδικευμένη εφαρμογή ανίχνευσης κίνησης μικρών ζώων. Ωστόσο έχει όλα τα χαρακτηριστικά ενός αυτόνομου πρωτοκόλλου επικοινωνίας, και επιπλέον είναι δυνατό πολλά από τα βασικά χαρακτηριστικά της να ενσωματωθούν σε άλλα υπάρχοντα ή υβριδικά πρωτόκολλα με θετικές επιπτώσεις στην απόδοση τους. Η άπληστη προσέγγιση (greedy approach) του directed diffusion που περιγράφεται και αξιολογείται στο [13] από τους C. Intanagonwiwat, D. Estrin, R. Govindan και J. Heidemann είναι μια παραλλαγή που υπόσχεται ακόμα καλύτερες επιδόσεις.

Το directed diffusion, εστιάζει αρκετά στο γεγονός ότι στα δίκτυα έξυπνης σκόνης οι περιορισμένοι πόροι πρέπει να χρησιμοποιούνται όσο πιο αποδοτικά γίνεται και προτείνει ενσωμάτωση της τεχνικής “ πολύπλεξης ” δεδομένων (**data aggregation**) και του έγκαιρου διαμοιρασμού μονοπατιών διάδοσης (**early path sharing**). Ειδικά στην άπληστη προσέγγιση του directed diffusion (greedy approach) η κεντρική ιδέα είναι ότι οι κόμβοι του δίκτυου συνεργάζονται για τη δημιουργία μιας δενδρικής δομής, η οποία πραγματοποιεί την πολύπλεξη δεδομένων με “ άπληστες τεχνικές ”, και έχει απώτερο στόχο να βελτιστοποιηθεί ο διαμοιρασμός των μονοπατιών διάδοσης πληροφορίας. Παράλληλα στο paper [13] γίνεται σύγκριση του greedy diffusion με μια “ οπορτουνιστική ” εκδοχή του directed diffusion όπου η πληροφορία πολυπλέκεται με “ευκαιριακό” τρόπο και διαδίδεται πάνω από δενδρική δομή αποκλειστικά δομημένη για να επιτυγχάνει χαμηλή καθυστέρηση διάδοσης γεγονότων προς την πηγή. Στο paper παρατηρείται ότι για δίκτυα υψηλής πυκνότητας η greedy approach παρουσιάζει σημαντικά ενεργειακά κέρδη.

Σύμφωνα με τη γενική λογική του directed diffusion, αρχικά το κέντρο ελέγχου οφείλει



Σχήμα 2.7: Μη αποδοτικός διαμοιρασμός μονοπατιού



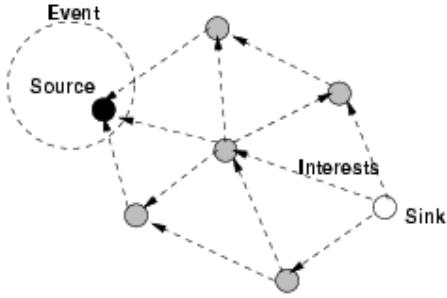
Σχήμα 2.8: Έγκαιρος διαμοιρασμός μονοπατιού

να δηλώσει το ενδιαφέρον του για μια πληροφορία. Το ενδιαφέρον αυτό είναι μια λίστα από ζεύγη χαρακτηριστικού - τιμής, που περιγράφουν ποιοι κόμβοι πρέπει να ξεκινήσουν να ανιχνεύουν το περιβάλλον τους καθώς και το τι ακριβώς ανιχνεύουν. Το ενδιαφέρον (**interest**) διαδίδεται με αλγόριθμο flooding ή επιλεκτικής προώθησης στους κόμβους του δικτύου. Αυτοί με τη σειρά τους, αποθηκεύουν τα ενδιαφέροντα που έχουν “ακούσει” και τα επεξεργάζονται.

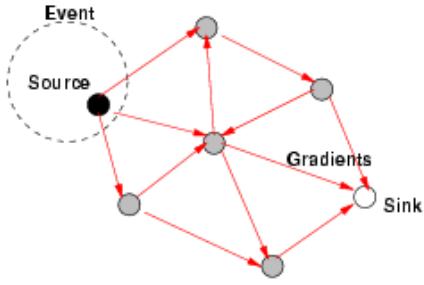
Ο κάθε κόμβος γνωρίζει ανά πάσα στιγμή από ποιους γειτονικούς κόμβους έλαβε το κάθε “interest” και αναθέτει σε αυτούς ένα χαρακτηριστικό βαθμωτό αριθμό (**gradient**). Αν από κάποιο γείτονα ο κόμβος έχει λάβει πολλαπλά interests, τότε διατηρεί ένα βαθμωτό αριθμό για κάθε ένα από αυτά τα interests. Όταν ένας κόμβος διαπιστώσει ότι μπορεί να παράγει πληροφορία ως απάντηση σε κάποιο ενδιαφέρον (interest) που έλαβε, τότε αυτός ο κόμβος γίνεται **πηγή** και ξεκινάει να ανιχνεύει το περιβάλλον του, ενεργοποιώντας τους κατάλληλους αισθητήρες. Συγκεντρώνει έτσι πληροφορία σχετική με το interest και στη συνέχεια δημιουργεί μηνύματα τα οποία στέλνει μόνο στους γείτονες για τους οποίους έχει gradient σχετικό με το συγκεκριμένο interest.

Κατά τη διάδοση της πληροφορίας, ο κάθε κόμβος διατηρεί μια μικρή μνήμη cache από όλα τα μηνύματα που έλαβε σε πρόσφατο χρόνο, για να αποφύγει τα διπλά μηνύματα και να αποτρέψει την επανάληψη αποστολής μηνυμάτων. Έτσι σε περίπτωση, όπου ένας κόμβος λάβει ένα μήνυμα, το οποίο ήδη υπάρχει στη μνήμη του, τότε το απορρίπτει. Όταν ένα μήνυμα με πληροφορία φτάσει στο κέντρο ελέγχου, τότε το κέντρο ελέγχου ακολουθώντας ένα ειδικό αλγόριθμο επιλέγει να ενισχύσει ένα γείτονα του από τον οποίο θα λάβει τα επόμενα μηνύματα - απαντήσεις για το σχετικό interest. Τον ίδιο αλγόριθμο εκτελούν αλυσιδωτά και όλοι οι ενδιάμεσοι κόμβοι στο μονοπάτι από το Sink στην πηγή, ώστε τελικά όλες οι απαντήσεις για δεδομένο ενδιαφέρον να μεταδίδονται πάνω σε ένα δυναμικά διαμορφωμένο μονοπάτι.

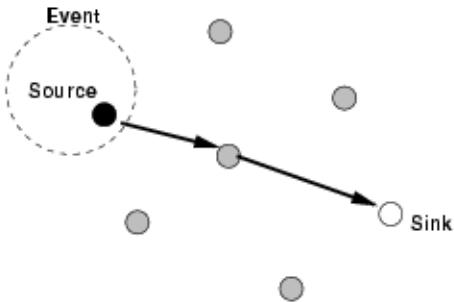
Συνοψίζοντας τα βασικά χαρακτηριστικά του directed diffusion μπορούμε να συμπεράνουμε ότι πρόκειται για πρωτόκολλο data-centric. Αυτό, όπως προαναφέρθηκε, σημαίνει ότι κάθε είδους πληροφορία προσδιορίζεται χρησιμοποιώντας ζεύγη χαρακτηριστικού - τιμής και κάθε αίτηση για ανίχνευση, προωθείται προς όλους τους κόμβους ως **ενδιαφέρον** για τέτοια



Σχήμα 2.9: Διάδοση ενός interest



Σχήμα 2.10: Αρχικοποίηση των gradients



Σχήμα 2.11: Διάδοση ενός γεγονότος στο Sink

πληροφορία. Μια άμεση συνέπεια αυτού του στοιχείου είναι ότι δεν απαιτούνται καθολικά μοναδικές ταυτότητες για τους κόμβους, καθώς το Sink δεν απευθύνεται σε κάθε ένα ξεχωριστά. Η μόνη απαίτηση είναι να μπορεί ο κάθε κόμβος να διαχωρίζει τους γείτονες του, ώστε να μπορεί να επιλέγει το σωστό γείτονα κατά την προώθηση ενός μηνύματος. Όλες οι μεταδόσεις στο πρωτόχολλο αυτό γίνονται μεταξύ γειτονικών κόμβων και δεν υπάρχουν end-to-end επικοινωνίες.

Στον ακόλουθο πίνακα συνοψίζουμε τις διαφορές μεταξύ της άπληστης εκδοχής του directed diffusion και της οπορτουνιστικής εκδοχής.

Οπορτουνιστική εκδοχή (Oportunistic)	Άπληστη Εκδοχή (Greedy)
Εγκαθίδρυση μονοπατιών μικρής καθυστέρησης, με αλγόριθμους που βασίζονται σε τοπική πληροφορία	Δημιουργία ενός ειδικού τύπου δένδρου το οποίο ονομάζεται “άπληστα αυξητικό” (greedy incremental)
Η πληροφορία από διαφορετικές πηγές μπορεί με ευκαιριακό τρόπο να συνενώνεται σε ενδιάμεσους κόμβους πάνω στα μονοπάτια διάδοσης	Η πληροφορία πολυπλέκεται χρησιμοποιώντας μια μέθοδο που βασίζεται σε μια άπληστη προσέγγιση για τη λύση του προβλήματος κάλυψης συνόλων με βάρη (weighted set covering problems)
Τα μονοπάτια που ξεκινούν από διαφορετικές πηγές δεν είναι απαραίτητο να συγχλίνουν και να διαμοιράζονται. Σε περιπτώσεις όπου δύο μονοπάτια συνενώνονται δε υπάρχει εξασφάλιση ότι αυτό γίνεται “νωρίς” σε σχέση με το μήκος των μονοπατιών	Διαμοιρασμός μονοπατιών διάδοσης πληροφορίας όσο το δυνατόν νωρίτερα. Δηλαδή δύο κόμβοι αρχίζουν να διαμοιράζονται ένα μονοπάτι διάδοσης όσο το δυνατόν πιο κοντά στην “αρετηρία” του μονοπατιού

Μπορεί κανείς να διαχρίνει ότι τα κεντρικά σημεία του πρωτοκόλλου greedy directed είναι η δημιουργία της δενδρικής δομής για τη διάδοση της πληροφορίας, ο μηχανισμός συγχώνευσης πληροφορίας και ο έγκαιρος διαμοιρασμός ων μονοπατιών διάδοσης.

Για να δομηθεί ένα άπληστα αυξητικό δένδρο, η βασική ιδέα είναι ότι εγκαθιδρύεται το συντομότερο μονοπάτι (**shortest path**) από την πρώτη πηγή, δηλαδή τον κόμβο που πρώτος θα ανιχνεύσει κάποιο γεγονός, προς το κέντρο ελέγχου (Sink). Στη συνέχεια οι υπόλοιπες πηγές, με προσθετικό τρόπο συνδέονται στο πιο κοντινό ως προς αυτές σημείο, με το υπάρχον δένδρο.

Η ιδέα της συγχώνευσης - πολύπλεξης της πληροφορίας (**data aggregation**) είναι πρωτοποριακή. Καθώς οι απαντήσεις στα interests διαδίδονται προς το Sink, οι ενδιάμεσοι κόμβοι στη διαδρομή αυτή μπορούν να συνενώνουν σε ένα περιληπτικό μήνυμα, πολλά από τα λαμβανόμενα μηνύματα. Αυτό μπορεί να μειώσει τον αριθμό των μηνυμάτων και να επεκτείνει τη ζωή του δικτύου. Ωστόσο, η πολύπλεξη της πληροφορίας είναι πρακτικά αποδοτική μόνο όταν το τελικό παραγόμενο μήνυμα έχει μειωμένο μήκος σε bit, σε σχέση με το άθροισμα του μεγέθους των μηνυμάτων από τα οποία προήλθε. Ειδάλως η συγκέντρωση κυκλοφορίας σε συγκεκριμένα σημεία του δικτύου θα επιβαρύνει τη λειτουργία του και θα έχει αρνητικές επιπτώσεις στο χρόνο ζωής του.

Η συνένωση πληροφορίας μπορεί να χωριστεί σε δύο κατηγορίες:

- 1) Χωρίς απώλειες (Lossless), όπου όλες οι λεπτομέρειες των επιμέρους μηνυμάτων διατηρούνται. Απορρίπτεται μόνο πληροφορία που χρίνεται απολύτως περιττή. Αυτό συμβαίνει στις τεχνικές timestamping και package aggregation.
- 2) Με απώλειες (Lossy), όπου κάποια τμήματα πληροφορίας μπορούν να χαθούν σαν αντάλλαγμα (tradeoff) για μεγαλύτερη εξοικονόμηση ενέργειας. Αυτό συμβαίνει στη τεχνική poutline aggregation.

Γενικά επιλέγεται πάντα εκείνη η τεχνική συγχώνευσης πληροφορίας, η οποία κρίνεται κατάλληλη ανάλογα με τις ανάγκες της κάθε εφαρμογής.

Ο διαμοιρασμός μονοπατιών διάδοσης από σημείο που είναι πιο κοντά στην αρχή του μονοπατιού παρά στην απόληξη του, αξιολογείται στον παρακάτω πίνακα :

Πλεονεκτήματα	Μειονεκτήματα
Μείωση της κατανάλωσης ενέργειας	Μεγαλύτερες καθυστερήσεις και λιγότερη ανθεκτικότητα σε τοπολογικές αλλαγές στο δίκτυο
Μείωση της συνολικής κυκλοφορίας στο δίκτυο	Η δημιουργία ενός βέλτιστου δένδρου που πληρεί τις προϋποθέσεις για “έγκαιρο διαμοιρασμό” αποτελεί πρόβλημα NP - hard
	Αν ακολουθηθεί η λύση ενός προσεγγιστικά βέλτιστου και όχι του ιδανικά βέλτιστου δένδρου, τα ενεργειακά κέρδη περιορίζονται αισθητά

Για να μπορέσει κανείς να αξιολογήσει τα χαρακτηριστικά του greedy directed diffusion πρέπει να εξετάσει αναλυτικά τη λειτουργία του. Όπως αναφέρθηκε, βασικό στόχο αποτελεί η δημιουργία ενός δένδρου greedy tree, για να ευνοηθεί η πολύπλεξη της διαδιδόμενης πληροφορίας.

Για το σκοπό αυτό ο Sink στέλνει ένα αρχικό interest, στο οποίο όμως προσδιορίζει ρητά ότι οι απαντήσεις πρέπει μεταδοθούν με χαμηλό ρυθμό (τουλάχιστον αρχικά). Μια πηγή που μπορεί να ανταποκριθεί στο ενδιαφέρον του Sink, ζεκινάει να μεταδίδει δεδομένα προς το Sink. Σε αυτή την αρχική φάση οι βαθμωτοί που τίθενται στους κόμβους και η απάντηση της πηγής ονομάζονται “διερευνητικοί” (exploratory gradients). Αφού ο Sink λάβει τις exploratory απαντήσεις, επιλέγει και ενισχύει ένα γείτονα από τον οποίο θα λαμβάνει στο μέλλον δεδομένα σχετικά με το interest με υψηλό πλέον ρυθμό. Με την ενίσχυση αυτή τίθενται αναδρομικά νέοι βαθμωτοί “ πληροφορίας ” (data gradients) για το δεδομένο interest, σε ενδιάμεσους κόμβους του δικτύου, διαμορφώνοντας ένα μονοπάτι για αποστολή δεδομένων με υψηλούς ρυθμούς.

Η δημιουργία του greedy δένδρου γίνεται ως εξής:

Κάθε πληροφορία σχετίζεται με ένα επιπλέον χαρακτηριστικό το E , που αντιπροσωπεύει το ενεργειακό κόστος για τη μετάδοση της πληροφορίας από τον κόμβο - πηγή στον τρέχοντα κόμβο. Αυτό βοηθάει στην επιλογή ενός μονοπατιού χαμηλής κατανάλωσης ενέργειας για την προώθηση πληροφορίας προς το κέντρο ελέγχου.

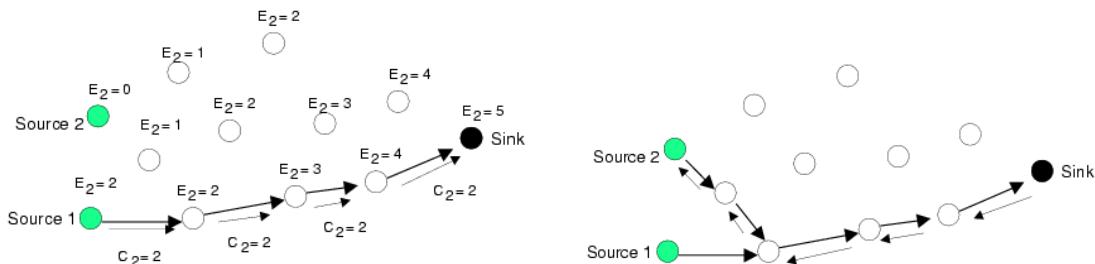
Κάθε πηγή παράγει επιπρόσθετα ένα **μήνυμα αυξανόμενου κόστους**, όταν λάβει ένα νέο exploratory μήνυμα από άλλες πηγές. Το μήνυμα αυξανόμενου κόστους έχει την ίδια ταυτότητα (ID) με το exploratory μήνυμα και επιπλέον αναφέρει το ενεργειακό κόστος C , το οποίο απαιτείται για τη διάδοση ενός μηνύματος - απάντηση μέχρι ένα κόμβο που ανήκει στο υπάρχον greedy δένδρο. Ένας κόμβος που βρίσκεται στο δένδρο όταν λάβει ένα μήνυμα αυξανόμενου κόστους, τότε θέτει το C του μηνύματος στην ελάχιστη τιμή μεταξύ του τρέχοντος C και του ενεργειακού κόστους E' του exploratory μηνύματος, το οποίο διατηρείται αποθηκευμένο στη μνήμη cache και έχει το ίδιο ID με το λαμβανόμενο μήνυμα αυξανόμενου κόστους. Έπειτα το νέο μήνυμα αυξανόμενου κόστους που προκύπτει προωθείται μόνο προς τους γείτονες για τους οποίους έχει οριστεί βαθμωτός data gradient.

Όταν το κέντρο ελέγχου Sink λάβει ένα νέο exploratory μήνυμα, αρχικοποιεί ένα χρονομετρητή T_p και όταν αυτός λήξει, τότε το Sink αποφασίζει και “ενισχύει” εκείνο το γειτονικό

του κόμβο, από τον οποίο έλαβε είτε ένα exploratory μήνυμα ή ένα μήνυμα αιχανόμενου κόστους με το μικρότερο ενεργειακό κόστος. Οι ενδιάμεσοι κόμβοι στο μονοπάτι προς την πηγή ενισχύονται με παρόμοιο τρόπο αλλά χωρίς τη χρήση χρονομετρητή. Όσοι κόμβοι ενισχύονται θα ανήκουν στο greedy δένδρο.

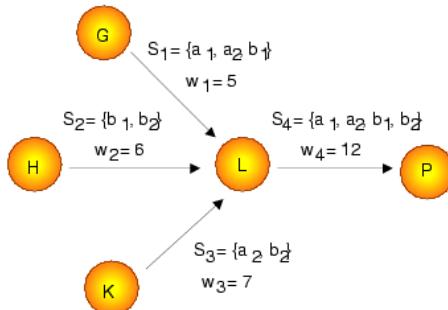
Με τον τρόπο αυτό εξασφαλίζεται ότι το μονοπάτι από τον πρώτο κόμβο - πηγή προς το Sink θα αντιστοιχεί στη χαμηλότερη δυνατή κατανάλωση ενέργειας. Οι κόμβοι που θα γίνουν πηγές σε μελλοντικό χρόνο, συνδέονται με το υπάρχον δένδρο στο πιο κοντινό σε αυτούς σημείο, αυξάνοντας έτσι τη δομή του δένδρου.

Στο greedy diffusion για τους σκοπούς μιας αποδοτικής πολύπλεξης δεδομένων, καθώς αυτά διαδίδονται προς το Sink, ο κάθε κόμβος καθορίζει σύνολα από εισερχόμενα μηνύματα (από τα οποία κάποια μπορεί να είναι ήδη συνενωμένα μηνύματα σε προηγούμενο κόμβο), τα οποία θα πρέπει να καλύπτουν όλα τα δεδομένα που πρέπει να διαδοθούν με το μικρότερο δυνατό κόστος. Στο paper [13] το πρόβλημα αυτό ανάγεται σε ένα πρόβλημα κάλυψης συνόλων με βάρη (weighted set-covering problem), το οποίο είναι NP-hard. Η αναγωγή βασίζεται στην αντιστοίχιση των εισερχόμενων μηνυμάτων με υποσύνολά που περιέχουν δεδομένα και στα οποία σύνολα ανατίθεται ένα βάρος ανάλογο προς το ενεργειακό κόστος τους. Ένας άπληστος αλγόριθμος για set-covering επιλέγει επιπλέοντα το επόμενο υποσύνολο από όσα απομένουν σε μια επαναλαμβανόμενη διαδικασία για την κάλυψη όλων των στοιχείων που δεν έχουν καλυφθεί με το μικρότερο ταυτόχρονα ενεργειακό κόστος. Ως τελικό βήμα γίνεται η αφαίρεση των όποιων πλεοναζόντων υποσυνόλων επιλέχθηκαν λόγω του άπληστου αλγορίθμου.



Σχήμα 2.12: Δημιουργία του greedy incremental tree

Σχήμα 2.13: Ενίσχυση ενός μονοπατιού διάδοσης



Σχήμα 2.14: Πολύπλεξη μηνυμάτων με χρήση weighted συνόλων

Τέλος πρέπει να αναφερθεί ότι παράλληλα με την ενίσχυση κάποιων κόμβων για τη διάδοση πληροφορίας, εφαρμόζεται και μια μέθοδος αποδυνάμωσης μονοπατιών διάδοσης. Η αποδυνάμωση μονοπατιού επιτελείται, ώστε να απαλειφθούν τα μη αποδοτικά μονοπάτια που ενώνουν τις πηγές με το Sink. Ο κεντρικός κανόνας για να γίνει αυτή η αποδυνάμωση, είναι να υπολογιστεί για κάθε κόμβο το σύνολο των γειτόνων του που καλύπτουν όλη την εισερχόμενη πληροφορία με το μικρότερο δυνατό κόστος. Οι υπόλοιποι κόμβοι αποδυναμώνονται με αποστολή ενός σχετικού μηνύματος. Τότε, ένας αποδυναμωμένος κόμβος μειώνει τον βαθμωτό που αντιστοιχεί στο γείτονα, από τον οποίο έλαβε εντολή αποδυνάμωσης. Αρχικά οι βαθμωτοί υποβαθμίζονται από data σε exploratory και όταν όλοι οι βαθμωτοί ενός κόμβου γίνουν exploratory τότε ο κόμβος με τη σειρά του αρχίζει να αποδυναμώνει δικούς του γείτονες που του στέλνουν δεδομένα, αλλά ωστόσο όχι όσους στέλνουν expository απαντήσεις.

Στο paper [13] γίνεται αξιολόγηση των επιδόσεων του greedy directed diffusion σε σχέση με τις αντίστοιχες του opportunistic diffusion. Το δυνατό σημείο της επιπόλαιης (greedy) προσέγγισης του direct diffusion είναι ότι ευνοεί τη δημιουργία διαμοιραζόμενων μονοπατιών, ενώ η οπορτουνιστική προσέγγιση επιλέγει μονοπάτια μόνο βάσει της καθυστέρησης διάδοσης προς το Sink. Γενικά παρατηρήθηκε ότι σε δίκτυα υψηλής πυκόνητας σε κόμβους, όπου το συντομότερο μονοπάτι είναι με μεγάλη πιθανότητα και διαμοιραζόμενο από πολλούς κόμβους, η greedy προσέγγιση παρουσιάζει μικρότερη συνολική κατανάλωση ενέργειας, και ταυτόχρονα συγχρινόμενα μεγέθη με την opportunistic εκδοχή για τη μέση καθυστέρηση διάδοσης και το ποσοστό επιτυχίας στη διάδοση πληροφορίας στο Sink.

Μεταβάλλοντας κάποιες παραμέτρους που ενέχονται στον αλγόριθμο της greedy προσέγγισης μπορούν να εξάγουν πρόσθετα ενδιαφέροντα συμπεράσματα όπως:

- 1) Η τυχαία τοποθέτηση των κόμβων - πηγών μειώνει την πιθανότητα διαμοιρασμού μονοπατιού ξεκινώντας από σημείο κατά το δυνατό πιο κοντά στην αφετηρία του. Κατά συνέπεια μειώνεται και η εξοικονόμηση ενέργειας.
- 2) Όταν υπάρχουν περισσότερα από ένα κέντρα ελέγχου, η εξοικονόμηση ενέργειας συγχλίνει προς την εξοικονόμηση που προσφέρει η οπορτουνιστική προσέγγιση.
- 3) Όταν αυξάνεται ο αριθμός των πηγών έχουμε παρόμοια αποτελέσματα για την εξοικονόμηση ενέργειας.

2.3.4 LTP

Το πρωτόκολλο LTP (local target protocol) προτάθηκε από τους I. Χατζηγιαννάκη, Σ. Νικολετσέα και Π. Σπυράκη στο [5]. Η αρχική παρουσίαση του έγινε για το μοντέλο του Smart Dust δικτύου, όπου το κέντρο ελέγχου αποτελεί ή μπορεί να συμβολιστεί με ένα άπειρο τείχος (Wall). Ωστόσο, το μοντέλο του Wall χρησιμοποιήθηκε για να απλοποιηθεί η θεωρητική ανάλυση της απόδοσης του πρωτοκόλλου και είναι εύκολο να διαπιστώσει κανείς ότι το LTP μπορεί να εφαρμοστεί και στο μοντέλο πηγής - Sink, όπου το κέντρο ελέγχου θεωρείται ως ένα σημείο του χώρου.

Στο Local Target Protocol, κάθε μετάδοση που αντιστοιχεί σε διάδοση πληροφορίας γίνεται πάντοτε προς τη γενική κατεύθυνση του Sink. Αυτό σημαίνει ότι οι μεταδόσεις γίνονται προς κόμβους που βρίσκονται κοντά στην ευθεία που ενώνει το κόμβο που μεταδίδει πληροφορία με

το Sink. Σύμφωνα με το πρωτόκολλο, οι υποψήφιοι κόμβοι προς τους οποίους ενδέχεται να γίνει μια μετάδοση βρίσκονται εντός προκαθορισμένης γωνίας που προσδιορίζει την απόκλιση της θέσης τους από την προαναφερθείσα ευθεία. Για την προώθηση πληροφορίας προς το κέντρο ελέγχου λαμβάνεται υπόψη τόσο η εγγύτητα του κάθε γειτονικού κόμβου στην ευθεία αυτή όσο και η απόσταση του από τον κόμβο που προωθεί την πληροφορία.

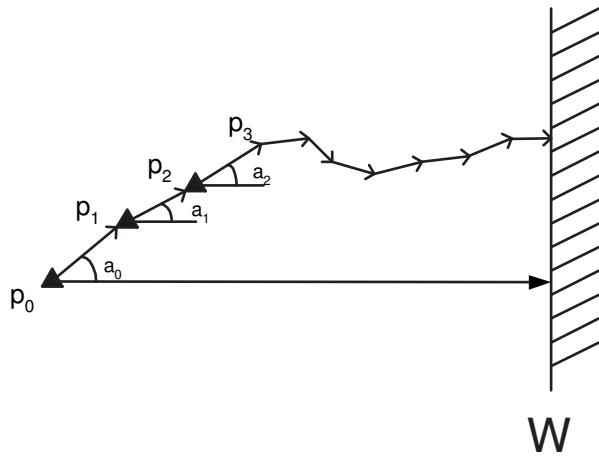
Ένα σημαντικό στοιχείο του LTP είναι ότι ο κάθε κόμβος πέρα από τη γνώση της γενικής κατεύθυνσης όπου βρίσκεται το Sink, δε χρειάζεται να έχει καθολική γνώση της τοπολογίας του δικτύου. Αντίθετα είναι αρκετό να γνωρίζει τοπικά τη θέση των γειτονικών κόμβων. Αυτή η γνώση είναι πληροφορία που μπορεί να αποκτηθεί εύκολα με ανταλλαγές μηνυμάτων σε αυστηρά τοπικό πλαίσιο. Επιπρόσθετα δε χρειάζεται οι κόμβοι να έχουν μοναδική ταυτότητα σε όλο το δίκτυο, αλλά αρκεί ο κάθε κόμβος να μπορεί να ξεχωρίζει τους γείτονές του μεταξύ τους.

Το πρωτόκολλο εξασφαλίζει με διάφορα πρόσθετα μηνύματα ελέγχου ότι ο κάθε κόμβος θα μπορεί να μάθει αν μια μετάδοση που έκανε συνεισφερεί ή όχι στη διάδοση της πληροφορίας προς το κέντρο ελέγχου. Μάλιστα σε περίπτωση που μια μετάδοση αποδειχθεί αναποτελεσματική, δίνεται η ευκαιρία στον κόμβο να επιλέξει ένα άλλο γειτονικό κόμβο για να εκπέμψει την πληροφορία ή εναλλακτικά να ειδοποιήσει τον κόμβο, από τον οποίο έλαβε αρχικά την πληροφορία, ώστε αυτός ο τελευταίος να πάρει νέα απόφαση προώθησης της.

Στην παρακάτω αναλυτική παρουσίαση του πρωτοκόλλου, υποθέτουμε ότι το κέντρο ελέγχου είναι ένα τείχος W απέριου μήκους καθώς αυτό διευκολύνει κάποιες θεωρήσεις και υποθέσεις. Επίσης συμβολίζουμε την πυκνότητα των συσκευών - smartdust στο δίκτυο με d , και ουσιαστικά εννοούμε το αριθμό συσκευών ανά m^2 . Θεωρούμε ακόμα ότι κάθε κόμβος διαθέτει δυνατότητα μεταδόσεων τόσο με τεχνική broadcast εμβέλειας R όσο και με οπτικά μέσα, δηλαδή εκπέμποντας μια ακτίνα laser εμβέλειας R προς σημείο. Μπορούμε προαιρετικά να επεκτείνουμε και την ιδέα αυτή, υποθέτοντας ότι μπορεί ένας κόμβος να εκπέμψει πληροφορία επιλεκτικά μόνο σε όσους κόμβους βρίσκονται εντός του τυμήματος του κυκλικού δίσκου που κείται μεταξύ των κύκλων ακτίνας R και $R - \delta R$, όπου δR μια τυχαία απόκλιση. Αυτό πρακτικά θα μπορούσε να πραγματοποιηθεί αν ο κάθε κόμβος γνωρίζει την απόσταση του από όλους τους γειτονικούς του κόμβους. Μια άλλη υπόθεση που κάνουμε προκειμένου να απλοποιηθεί η ανάλυση του πρωτοκόλλου είναι ότι οι πλειονότητα των κόμβων έχουν αρχικά γειτονικούς κόμβους εντός της εμβέλειας μετάδοσης, ώστε τουλάχιστο αρχικά να μπορεί κάθε πληροφορία να διαδοθεί προς το W με μεγάλη πιθανότητα.

Η γενική ιδέα του LTP που είναι κατά βάση ένα multihop πρωτόκολλο, είναι ότι για τη διάδοση της πληροφορίας ενός συμβάντος, που θα ανιχνεύσει ένας κόμβος p του δικτύου, προς το κέντρο ελέγχου θα πρέπει να συνεργαστούν οι ενδιάμεσοι κόμβοι. Στην ιδανική περίπτωση θα υπήρχε μια σειρά από κόμβους που θα σχημάτιζαν ευθεία γραμμή από το p κάθετη στο τείχος W . Οι γειτονικοί κόμβοι βέβαια στην ευθεία θα έπρεπε ανά δύο να είναι εντός εμβέλειας R . Ας σημειωθεί ότι στην περίπτωση που το κέντρο ελέγχου δεν είναι το άπειρο τείχος αλλά ένα μόνο σημείο, τότε ως ιδανική ευθεία διάδοσης θεωρούμε την ευθεία που ενώνει τον κόμβο πηγή και το κέντρο ελέγχου.

Ωστόσο, κατά κανόνα η ιδανική περίπτωση δεν ισχύει. Γενικά ονομάζουμε τον αριθμό των διωδόσεων (hops) που λαμβάνουν χώρα στην ιδανική περίπτωση $h_{opt}(p, W)$, ενώ τον αριθμό των hops που πραγματοποιούνται κατά την εφαρμογή ενός πραγματικού πρωτοκόλλου π , τον ονομάζουμε $h(\pi, p, W)$. Το πηλίκο των δύο αυτών όρων δίνει ένα μέτρο αξιολόγησης - αποδοτικότητας του πρωτοκόλλου p , που στην περίπτωση αυτή είναι το LTP.



Σχήμα 2.15: Διάδοση μηνύματος προς το Wall

Ένας άλλος όρος, με το συμβολισμό $d(p, W)$ δηλώνει την κάθετη απόσταση του κόμβου p από το W , ενώ ο όρος $d(p, p2)$ δείχνει την κάθετη απόσταση μεταξύ των δύο κόμβων p και $p2$. Επίσης, σημαντική παράμετρος αποτελεί και η γωνία α που σχηματίζεται από τη πραγματική διεύθυνση της διάδοσης του μηνύματος και την ιδανική διεύθυνση. Αυτές τις γωνίες α τις ονομάζουμε γωνίες αναζήτησης. Τέλος ως $info(e)$ συμβολίζεται η πληροφορία που πρέπει να διαδοθεί και να φτάσει στο τείχος W .

Αυτό που λαμβάνεται περισσότερο υπόψη στην αξιολόγηση της απόδοσης του πρωτοκόλλου είναι ο αριθμός των hops που γίνονται κατά τη διάδοση ενός μηνύματος. Ο αριθμός αυτός δίνει διαισθητικά πληροφορία για την κατανάλωση ενέργειας από τους κόμβους του δικτύου, και για το χρόνο διάδοσης. Αυτό συμβαίνει γιατί μπορούμε να προσεγγιστικά να υποθέσουμε ότι για τοπικές μεταδόσεις σε μικρά σχετικές αποστάσεις, κάθε εκπομπή διαρκεί σταθερό χρονικό διάστημα και της αντιστοιχεί ένα σταθερό ποσό καταναλισκόμενης ενέργειας. Αυτά τα δύο ποσά πολλαπλασιάζομενα με τον αριθμό των hops, δίνουν μια εκτίμηση της καθυστέρησης διάδοσης ενός γεγονότος προς το W και του ενεργειακού κόστους της διάδοσης αυτής.

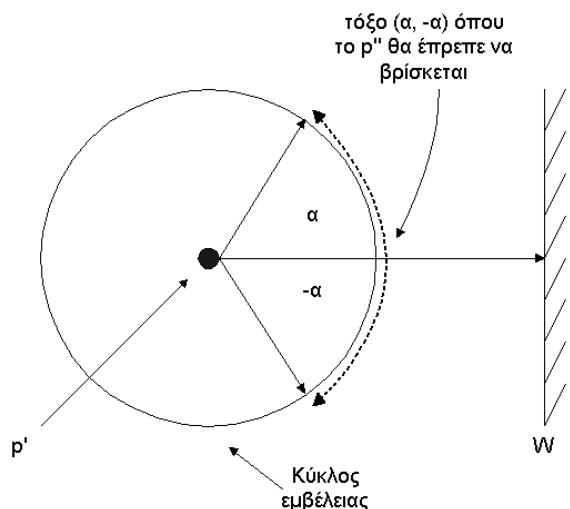
Προχωρώντας στην αναλυτική παρουσίαση του LTP πρέπει να αναφέρουμε ότι πριν την εφαρμογή των βασικών βημάτων που προτείνει το πρωτόκολλο, πρέπει να προηγηθεί μια φάση προετοιμασίας του δικτύου. Στη φάση αυτή οι κόμβοι ανταλλάσσουν μηνύματα ελέγχου ώστε ο κάθε κόμβος τελικά να γνωρίζει τη γεωγραφική κατεύθυνση, στην οποία βρίσκεται το W και να μπορεί να προωθεί μηνύματα προς εκείνη την κατεύθυνση.

Ακολουθεί η κυρίως φάση της λειτουργίας του LTP. Όπως ορίζει το πρωτόκολλο, ένας κόμβος - πηγή παράγει ένα μήνυμα $info(e)$ μόλις ανιχνεύσει ένα συμβάν που χρήζει διάδοσης. Κατόπιν το μήνυμα αυτό μεταδίδεται σε γειτονικό κόμβο p'' σύμφωνα με τα παρακάτω βήματα, τα οποία ακολουθώνται επαναληπτικά και από κάθε κόμβο του δικτύου όταν λαμβάνει ένα μήνυμα $info(e)$:

- 1) Φάση αναζήτησης (Search phase)** Στην φάση αυτή ο κόμβος p' που κατέχει το $info(e)$, αναζητεί ένα γειτονικό p'' που θα είναι τοποθετημένος πιο κοντά στο τείχος από τον εαυτό του. Αυτό γίνεται με περιοδική μετάδοση broadcast ενός σήματος.

Με το broadcast σήμα ελέγχου και τις απαντήσεις σε αυτό το σήμα που λαμβάνει ο p' , θεωρούμε ότι ο p' μαθαίνει την απόσταση του από τους γειτονικούς κόμβους που είναι εντός της εμβέλειας μετάδοσης R , αλλά και την απόκλιση αυτών των κόμβων από την ιδιαίτερη ευθεία διάδοσης. Η περιοδική εκπομπή του σήματος εξασφαλίζει ότι ο κάθε κόμβος p' μαθαίνει σχετικά εγκαίρως τυχόν τοπολογικές μεταβολές που συμβαίνουν τοπικά στο δίκτυο, όπως επίσης και ποιοι από τους γειτονικούς κόμβους έχουν εξαντληθεί ενεργειακά και αδυνατούν να προωθήσουν μήνυμα προς το W . Ο κόμβος p'' που θα αναδειχθεί ως υποψήφιος προς τον οποίο θα γίνει η μετάδοση της πληροφορίας, πρέπει να πληρεί δύο προϋποθέσεις. Η πρώτη είναι να βρίσκεται εντός ακτίνας R και προαιρετικά πέρα από μια ακτίνα $R - \delta R$. Η δεύτερη προϋπόθεση είναι ότι η γωνία αναζήτησης α , που σχηματίζεται από το ευθύγραμμο τμήμα $p'p''$ και την κάθετο από το p' στο W θα είναι κατά απόλυτη τιμή μικρότερη από μια προκαθορισμένη γωνία α_{ref} . Αυτό εξασφαλίζει ότι ο κόμβος p'' θα είναι προς την κατεύθυνση του τείχους εντός μιας επιθυμητής ή μιας αποδεκτής απόκλισης.

- 2) **Φάση απευθείας μετάδοσης (Direct transmission)** Ο κόμβος p' στέλνει το μήνυμα $info(E)$ στον κόμβο που μόλις ανακάλυψε μέσω απευθείας μετάδοσης (laser transmission). Θεωρητικά αυτό αξιολογείται ως το περισσότερο χρονοβόρο βήμα από όλα τα βήματα που ακολουθεί ο κόμβος p' .
- 3) **Φάση πισωδρόμησης (Backtrack)** Αν ο αριθμός των αποτυχημένων φάσεων αναζήτησης ξεπερνάει κάποιο συγκεκριμένο όριο, τότε το $info(e)$ ο κόμβος p' το στέλνει πίσω στον κόμβο p , από τον οποίο το είχε λάβει αρχικά. Αν ο κόμβος p' είναι η πηγή του $info(e)$, τότε αποφασίζει ότι είναι αδύνατη η διάδοση μηνύματος προς το κέντρο ελέγχου και απορρίπτει το $info(e)$.



Σχήμα 2.16: Η φάση αναζήτησης του LTP

Στη θεωρητική ανάλυση του πρωτοκόλλου γίνονται για λόγους απλοποίησης των υπολογισμών οι ακόλουθες υποθέσεις.

- 1) Η φάση αναζήτησης διαρκεί προσεγγιστικά μηδενικό χρόνο και σε αυτήν πάντα ανακαλύπτεται ένας επόμενος κόμβος, ώστε να μη χρειάζεται να υπάρξει φάση της πισωδρόμησης.
- 2) Η ενέργεια που καταναλώνεται στη φάση αναζήτησης είναι σχεδόν αμελητέα
- 3) Η θέση του κάθε επόμενου κόμβου p'' είναι ομοιόμορφα κατανεμημένη πάνω στο τόξο που αντιστοιχεί σε γωνία 2α γύρω από τη κάθετο προς το τείχος. Επίσης η επιλογή του επόμενου κόμβου γίνεται τυχαία και ανεξάρτητα από τον προηγούμενο κόμβο.

Η αποδοτικότητα, όσον αφορά τον αριθμό των hops που εμπλέκονται στη διάδοση ενός μηνύματος, συμβολίζεται με $E(C_h)$ και δίνεται από τον πηλίκο του αναμενόμενου αριθμού hops για το πρωτόκολλο προς τον ιδανικό αριθμό hops h_{opt} .

Η ανάλυση που έγινε με βάση τις προαναφερθείσες απλουστευτικές υποθέσεις κατέληξε ότι για συμβάντα που ανιχνεύονται μακριά από το τείχος το $E(C_h)$ δίνεται από τη σχέση :

$$E(C_h) \approx \frac{\alpha}{\eta \mu \alpha}$$

και επίσης

$$1 \leq E(C_h) \leq \frac{\pi}{2}, \quad 0 \leq \alpha \leq \frac{\pi}{2}$$

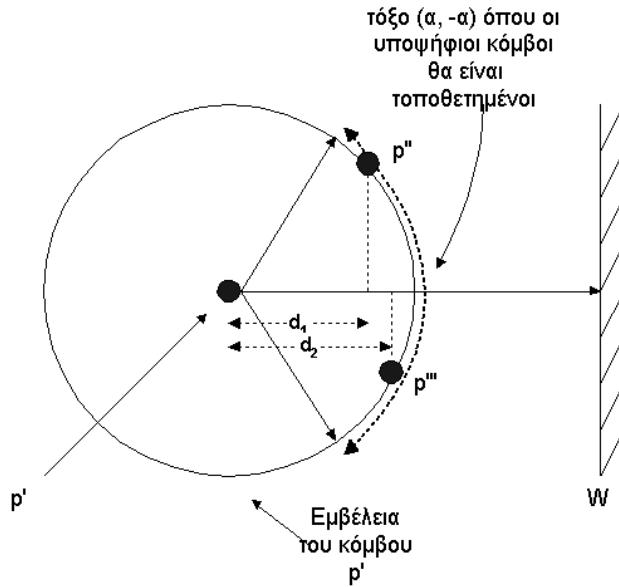
Το LTP, λοιπόν, αποτελεί ένα αποδοτικό πρωτόκολλο για δίκτυα έξυπνης σκόνης, καθώς απαιτεί μόνο τοπική γνώση για το δίκτυο από τους κόμβους και επιτυγχάνει τη διάδοση μηνυμάτων προς το κέντρο ελέγχου με αρκετά ικανοποιητικές τιμές για την ενεργειακή κατανάλωση, διατηρώντας παράλληλα χαμηλά τα επίπεδα κίνησης (traffic flow) στο δίκτυο.

Ωστόσο, παρουσιάζει και κάποια αδύνατα σημεία. Το βασικότερο από αυτά είναι ότι ένα μήνυμα μπορεί να διαδοθεί για αρκετά βήματα προς την κατεύθυνση του W , πριν ένας κόμβος ανακαλύψει ότι δε μπορεί να το προωθήσει παραπέρα. Αυτό μπορεί να συνεπάγεται δύο ενδεχόμενα. Στο πρώτο ενδεχόμενο γίνεται μια σειρά από backtracks και έτσι ενδέχεται το μήνυμα να επιστρέψει πίσω ακόμα και μέχρι τον κόμβο - πηγη πριν βρεθεί ένας νέος κόμβος που θα συνεχίσει το μονοπάτι προς το κέντρο ελέγχου. Στο δεύτερο ενδεχόμενο, το οποίο είναι και το πιο απαισιόδοξο, ο κόμβος που δε μπορεί να προωθήσει το μήνυμα, μπορεί να έχει εξαντληθεί ενεργειακά και έτσι δε θα μπορέσει να εκτελέσει ούτε τη φάση πισωδρόμησης, και συνεπώς το μήνυμα θα παραμείνει για πάντα σε αδιέξοδο. Αυτή το τελευταίο σενάριο βέβαια θα μπορούσε να προληφθεί αν το πρωτόκολλο εφάρμοζε μια πολιτική λήξης χρονομετρητών (timeouts) σε συνδυασμό με αποστελλόμενα μηνύματα acknowledge (ACK). Έτσι μέσα σε προκαθορισμένο χρονικό διάστημα ένας κόμβος θα έπρεπε να στείλει μήνυμα ACK στον κόμβο από τον οποίο έλαβε το μήνυμα αρχικά, σηματοδοτώντας ότι κατάφερε να προωθήσει παραπέρα το μήνυμα. Αν το μήνυμα ACK δεν ερχόταν πριν τη ληξη ενός χρονομετρητή, τότε θα ξεκινούσε νέα διαδικασία αναζήτησης για άλλο κόμβο, αποφεύγοντας έτσι ανεπιθύμητα αδιέξοδα.

2.3.5 MIN 2

Το πρωτόκολλο Min -Two Uniform Targets Protocol (MIN2) είναι κατ' ουσία μια βελτιστοποίηση πάνω στο LTP πρωτόκολλο και παρουσιάζεται στην εργασία [5] από τους I. Χατζηγιαννάκη, Σ. Νικολετσέα και Π. Σπυράκη. Το MIN2 προτάθηκε για να καλύψει μία αδυναμία του LTP, η οποία είναι η επιλογή στη φάση αναζήτησης, με τυχαίο τρόπο, ενός επόμενου κόμβου, χωρίς ουσιαστικά να λαμβάνεται υπόψη το κατά πόσο θα βοηθήσει αυτή η επιλογή στην διάδοση του μηνύματος προς την πηγή ή το γεγονός ότι ενδέχεται να υπήρχε πιο κατάλληλη επιλογή.

Η βασική αλλαγή που εισάγει το LTP έγκειται στο ότι στη φάση αναζήτησης, ένας κόμβος ανακαλύπτει δύο πλέον πιθανούς υποψήφιους στους οποίους ενδέχεται να διαδοθεί στο επόμενο βήμα (hop) το $info(e)$. Αν ονομάσουμε τους υποψήφιους κόμβους ως p'' και p''' , αυτοί κατά παραδοχή θα είναι ομοιόμορφα κατανεμημένοι στο τόξο $(-\alpha, \alpha)$ ακτίνας R . Μια πιο ρεαλιστική θεώρηση θα απαιτούσε οι δύο κόμβοι να είναι ομοιόμορφα κατανεμημένοι στον κυκλικό τομέα που αντιστοιχεί στο τόξο αυτό, όμως για λόγους απλοποίησης της θεωρητικής ανάλυσης απαιτούμε οι κόμβοι να ισαπέχουν από τον κόμβο που εκτελεί τη φάση αναζήτησης.



Σχήμα 2.17: Η φάση αναζήτησης του MIN2. Ο κόμβος p'' αντιστοιχεί σε μεγαλύτερη προβολή και συνεπώς είναι αυτός που τελικά θα επιλεγεί για την προώθηση της πληροφορίας

Μεταξύ των δύο υποψήφιων κόμβων επιλέγεται ο βέλτιστος με μέτρο την ουσιαστική πρόοδο που πρόκειται να επιτελεστεί κατόπιν της επιλογής του στη διάδοση προς το κέντρο ελέγχου. Η πρόοδος αυτή ουσιαστικά υποδεικνύεται από την προβολή του ευθυγράμου τμήματος που ενώνει τον κόμβο που εκτελεί τη φάση αναζήτησης με κάθε ένα από τους p'' και p''' πάνω στην ιδανική ευθεία διάδοσης. Έτσι αν α_{i1} και α_{i2} είναι οι γωνίες που αντιστοιχούν στα p'' και p''' τότε επιλέγεται ο κόμβος με την μικρότερη γωνία. Το συνημίτονο της γωνίας αυτής είναι ενδεικτικό της προόδου που γίνεται προς τη κατεύθυνση του κέντρου ελέγχου.

Αν τελικά συμβολίσουμε με α_i τη μικρότερη από τις δύο γωνίες, τότε είναι δυνατό να υπολογιστεί η αποδοτικότητα του πρωτοκόλλου βάσει του αναμενόμενου αριθμού hops που πραγματοποιούνται στη διάδοση μιας πληροφορίας προς το W . Ειδικά για την περίπτωση συμβάντων που διαδραματίζονται και ανιχνεύονται μακριά από το κέντρο ελέγχου ο αριθμός αυτός προκύπτει :

$$E(C_h) \approx \frac{\alpha^2}{2(1 - \sin \alpha)}, \quad 0 \leq \alpha \leq \frac{\pi}{2}$$

Επίσης, για τις οριακές τιμές της γωνίας α , όπου $\alpha \rightarrow 0$ και $\alpha \rightarrow \frac{\pi}{2}$, ο παραπάνω τύπος δίνει:

$$1 \leq E(C_h) \leq \frac{\pi^2}{8}$$

Οι εμπνευστές του πρωτοκόλλου αναφέρουν ότι συγκριτικά με τις επιδόσεις του LTP πρωτοκόλλου, το MIN 2 παρουσιάζει ένα κέρδος 21,5% όσον αφορά στην αποδοτικότητα βάσει του αναμενόμενου αριθμού hops.

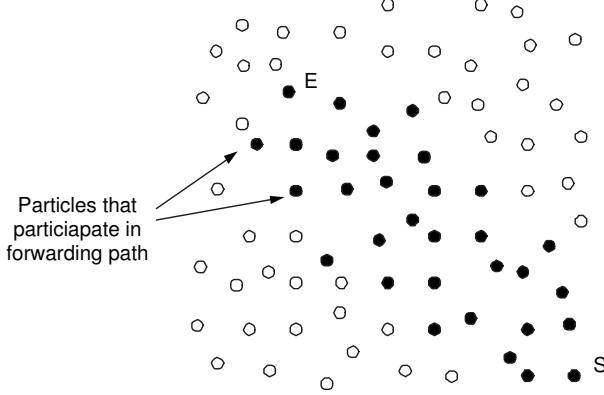
Κεφάλαιο 3

Το πρωτόκολλο PFR και η τροποποίηση μας

Το όνομα Probabilistic Forwarding Protocol (PFR) σημαίνει σε ελεύθερη μετάφραση πρωτόκολλο διάδοσης πληροφορίας με πιθανοτικές μεθόδους. Το πρωτόκολλο PFR προτάθηκε από τους I. Χατζηγιαννάκη, T. Δημητρίου, Σ. Νικολετσέα και Π. Σπυράκη στην εργασία [6]. Η σχεδίαση του έγινε με κεντρικό στόχο τη δραματική μείωση των περιττών μεταδόσεων που συμβαίνουν κατά τη διάρκεια της διάδοσης πληροφορίας σε ένα δίκτυο αισθητήρων, διατηρώντας ταυτόχρονα την καθυστέρηση διάδοσης σε αποδεκτά επίπεδα. Το πρωτόκολλο ανήκει στην κλάση των αντιδραστικών (reactive) πρωτοκόλλων, όπου οι μεταδόσεις γίνονται μόνο όταν ένα γεγονός θεωρηθεί αρκετά ενδιαφέρον για να το πληροφορηθεί και το κέντρο ελέγχου. Το μοντέλο δικτύου, όπου εφαρμόζεται το PFR είναι το μοντέλο πηγής - Sink με τυχαία κατανομή του δικτύου. Το PFR αρχικά προτάθηκε για το μοντέλο LATTICE [6]. Όπως διατυπώθηκε όμως και στο αρχικό paper και διαπιστώθηκε σε μετέπειτα χρόνο, το PFR λειτουργεί εξίσου αποδοτικά και σε τυχαίες κατανομές κόμβων στο χώρο, με επαρκή πυκνότητα κόμβων ανά μονάδα επιφανείας.

Η ιδέα που κρύβεται πίσω από τη λογική του πρωτοκόλλου είναι η τάση να ευνοούνται με πιθανοτικό τρόπο οι μεταδόσεις που γίνονται α) προς την κατεύθυνση του κέντρου ελέγχου και β) μέσα σε μια σχετικά στενή ζώνη από κόμβους γύρω από τη νοητή γραμμή που συνδέει τον κόμβο πηγής E , που ανιχνεύει ένα συμβάν, με το Sink.

Πάνω σε αυτή τη νοητή ευθεία, η διάδοση της πληροφορίας θεωρείται και αποδεικνύεται ότι είναι ιδιαίτερη λαμβάνοντας πάντοτε υπόψη το κόστος σε χρόνο και ενέργεια που συνεπάγεται. Ωστόσο η διάδοση πάνω σε αυτή τη γραμμή χωρίς αποκλίσεις δεν είναι πάντοτε εφικτή. Αυτό ισχύει καθώς ακόμα και στην περίπτωση στην οποία, αρχικά, κατά μήκος της ευθείας αυτής γραμμής υπήρχαν τοποθετημένοι κόμβοι με επαρκή πυκνότητα, είναι πολύ πιθανό κάποιοι από τους κόμβους αυτούς κατά τη διάρκεια λειτουργίας του πρωτοκόλλου να γίνουν ανενεργοί είτε μόνιμα είτε προσωρινά. Για την μόνιμη αδρανοποίηση ενός κόμβου συνήθως ευθύνεται η εξάντληση των ενεργειακών αποθεμάτων του, ενώ για την προσωρινή ενδέχεται να ευθύνεται η λειτουργία του κόμβου σε sleep mode, για μικρές χρονικές περιόδους. Ένας κόμβος μπορεί να καθίσταται ανενεργός και για άλλους λόγους όπως α) φυσικές βλάβες στους κόμβους - αισθητήρες, β) σκόπιμη απομάκρυνση τους από το πεδίο (πιθανότατα από κάποιον αντίπαλο ειδικά σε στρατιωτικές εφαρμογές) γ) αλλαγές στη θέση των κόμβων για λόγους όπως αντίξοες καιρικές συνθήκες και δ) ύπαρξη φυσικών εμποδίων που εμποδίζουν την επικοινωνία



Σχήμα 3.1: Η στενή ζώνη που συνδέει τον κόμβο-πηγή με το SINK

μεταξύ κάποιων κόμβων.

Η βασική λειτουργία που επιτελεί το πρωτόκολλο είναι η πιθανοτική μετάδοση της πληροφορίας, το οποίο εν συντομίᾳ συνεπάγεται ότι κάποιος κόμβος ενδέχεται να μην προωθήσει μια πληροφορία που έλαβε. Αυτό από μόνο του σημαίνει ότι είναι αρκετά πιθανό να υπάρχουν αποτυχίες στη μετάδοση πληροφορίας μέχρι το κέντρο ελέγχου. Συγκεκριμένα είναι δυνατό να καταπνιγεί η διάδοση μιας πληροφορίας πριν φτάσει στο κέντρο ελέγχου και μάλιστα στα αρχικά βήματα της διάδοσης. Πρακτικά είναι επιθυμητό να μην υπάρχουν αποτυχίες στη μετάδοση πληροφορίας προς το Sink, ενώ αν υπάρχουν είναι επιθυμητό η πληροφορία να έχει διαλύσει τη μεγαλύτερη δυνατή απόσταση, ώστε η τελευταία μετάδοση στη διαδικασία της διάδοσης να έγινε προς κόμβο που είναι σχετικά κοντά στο Sink και όχι στην πηγή. Στην τελευταία αυτή περίπτωση θα μπορούσε να υποτεθεί ότι ο Sink εκτελεί τυχαίους μικρούς περιπάτους, ώστε να συλλέγει αυτού του είδους την πληροφορία.

Το πρωτόκολλο λειτουργεί ως εξής: Οι κόμβοι που δεν έχουν πληροφορία για μετάδοση ή επεξεργασία, μπορούν να αισθάνονται το περιβάλλον τους για να ανιχνεύσουν κάποιο συμβάν. Μόλις ένας κόμβος ανιχνεύσει ένα συμβάν, το επεξεργάζεται και δημιουργεί ένα μήνυμα με σχετική πληροφορία που πρέπει να διαδοθεί προς το Sink. Η διάδοση της πληροφορίας εξελίσσεται σε δύο φάσεις:

Φάση 1: Η φάση αυτή φέρει την ονομασία “φάση δημιουργίας μετώπου”. Προηγείται της δεύτερης και κεντρικής φάσης του PFR καθώς αποτελεί φάση προετοιμασίας με στόχο να περιοριστούν οι πρώιμες αποτυχίες στη διάδοση πληροφορίας προς το Sink. Σε αυτή την αρχική φάση συγχροτείται ένα επαρκώς πλατύ μέτωπο από κόμβους που φέρουν την πληροφορία προς μετάδοση, ώστε τελικά αυτή η πληροφορία είτε να φτάσει στο Sink ή σε ένα σχετικά κοντινό του κόμβο. Το μέτωπο δημιουργείται με ένα ειδικά σχεδιασμένο μηχανισμό “συγχρατημένου flooding” που εφαρμόζεται για ένα προγραμματισμένο αριθμό βημάτων στη διάδοση της πληροφορίας. Η βασική λειτουργία αυτού του μηχανισμού υπαγορεύει την ύπαρξη στην επικεφαλίδα κάθε μηνύματος ενός μετρητή που ονομάζουμε beta counter. Αυτός ο μετρητής αρχικοποιείται από τον κόμβο πηγής, όταν ο κόμβος πηγής δημιουργεί το αρχικό μήνυμα που σχετίζεται με το συμβάν το οποίο ανιχνεύτηκε. Μετά από αυτή την αρχικοποίηση κάθε κόμβος όταν λάβει ένα μήνυμα που περιέχει στην επικεφαλίδα του μια θετική τιμή για το beta counter, μειώνει την τιμή του μετρητή κατά ένα και το προωθεί ντετερμινιστικά (σε αντίθεση με το πιθα-

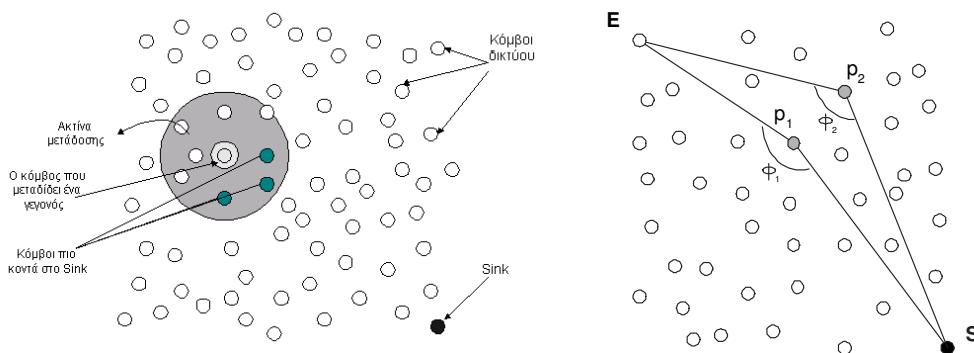
νοτικά) στην κατεύθυνση προς το Sink. Ο τρόπος μετάδοσης είναι λίγο ιδιόρρυθμος. Ο κάθε κόμβος χρησιμοποιεί μετάδοση υπό γωνία για να κάνει broadcast το μήνυμα σε όλους τους γείτονες, όσους βρίσκονται προς στην κατεύθυνση του Sink. Αυτό συνεπάγεται ότι το μήνυμα θα το λάβουν μόνο οι γείτονες που βρίσκονται εντός του κυκλικού τομέα που ορίζεται από την ακτίνα μετάδοσης του κόμβου και δυο επίκεντρες γωνίες $+ \alpha$ και $- \alpha$ με ευθεία αναφοράς την ευθεία που ενώνει τον τρέχοντα κόμβο με το Sink. Αν ο beta counter σε ένα λαμβανόμενο μήνυμα έχει την τιμή μηδέν (0), τότε ο κόμβος περνάει στη δεύτερη φάση του PFR πρωτοκόλλου. Έτσι ουσιαστικά ο beta counter καθορίζει τη διάρκεια της πρώτης φάσης του PFR.

Φάση 2: Η φάση αυτή ονομάζεται φάση πιθανοτικής προώθησης της πληροφορίας και είναι αυτή που έδωσε και το όνομα της στο πρωτόκολλο PFR. Κάθε κόμβος υπολογίζει την πιθανότητα συμμετοχής του στη διαδικασία διάδοσης μιας πληροφορίας. Όσο πιο κοντά βρίσκεται ο κόμβος στην ιδανική νοητή γραμμή μετάδοσης, που ενώνει την πηγή E με το Sink S , τόσο πιο μεγάλη είναι η πιθανότητα να προωθήσει τη πληροφορία που σχετίζεται με το ανιχνευθέν συμβόλιο. Η “πιθανότητα προώθησης πληροφορίας” (forwarding probability) επιλέγεται να είναι

$$P = \frac{\phi}{\pi}$$

όπου ϕ είναι η γωνία που ορίζεται από α) τη γραμμή η οποία ενώνει τον (τρέχοντα) κόμβο που εκτελεί τη τυχαία επιλογή με το κόμβο πηγή και β) τη γραμμή που ενώνει τον τρέχοντα κόμβο με το Sink.

Είναι προφανές ότι μεγαλύτερη γωνία συνεπάγεται ότι ο κόμβος βρίσκεται πιο κοντά στην απευθείας γραμμή μεταξύ πηγής και Sink. Όταν $\phi = \pi$, τότε ο κόμβος βρίσκεται ακριβώς πάνω στη νοητή ιδανικά γραμμή διάδοσης και η πιθανότητα προώθησης πληροφορίας είναι 1. Επίσης πρέπει να σημειωθεί ότι ο υπολογισμός του ϕ απαιτεί μόνο τοπική πληροφορία και όχι καθολική γνώση της τοπολογίας του δικτύου.



Σχήμα 3.2: Η πρώτη φάση του PFR σε Σχήμα 3.3: Οι γωνίες φ της 2ης φάσης του PFR

3.1 SW-PFR

Το SW-PFR (Sleep - Awake PFR) πρωτόκολλο, που θα παρουσιάσουμε παρακάτω και το οποίο εξομοιώθηκε στα πλαίσια αυτής της διπλωματικής, είναι στην ουσία μια παραλλαγή πάνω στο αρχικό PFR πρωτόκολλο που προτάθηκε στο [6]. Δύο σημαντικές προσθήκες έχουν γίνει σε εκείνο το αρχικό πρωτόκολλο.

Πρώτα, η νέα εκδοχή του PFR απευθύνεται σε δίκτυα αισθητήρων, όπου μπορούν να συμβούν, να ανιχνευτούν και να διαδοθούν σε παράλληλο χρόνο περισσότερα από ένα συμβάντα (multiple sense-event generations). Έτσι διευρύνονται τα πεδία εφαρμογών για το PFR πρωτόκολλο που αρχικά δεν διέθετε μηχανισμούς διαχωρισμού μεταξύ μηνυμάτων που αναφέρονται σε διαφορετικά γεγονότα. Στο νέο πρωτόκολλο SW-PFR, είναι δυνατό σε οποιαδήποτε χρονική στιγμή περισσότερα από ένα ανιχνευθέντα γεγονότα να διαδίδονται προς το κέντρο ελέγχου.

Για να αποφύγουμε ωστόσο επανάληψη μεταδόσεων και δημιουργία ατέρμονων βρόχων στη μετάδοση συμβάντων, κάθε κόμβος αναβαθμίστηκε ώστε να διαθέτει μια μνήμη cache. Στη μνήμη αυτή ένας κόμβος αποθηκεύει την ταυτότητα ID για κάθε διαχριτό συμβάν που έχει “ακούσει”. Η μοναδικότητα της ταυτότητας ID για κάθε συμβάν, είναι εγγυημένη καθώς ως ταυτότητα επιλέγεται η συνένωση της ταυτότητας του κόμβου - πηγής και της χρονοσφραγίδας που αντιστοιχεί στη χρονική στιγμή που ανιχνεύτηκε το συμβάν. Όταν πραγματοποιηθεί η λήψη ενός μηνύματος, ο κόμβος ελέγχει την μνήμη του για να διαπιστώσει αν διαθέτει καταχώρηση για το συμβάν, το οποίο περιγράφεται στο μήνυμα. Αν το συμβάν δεν υπάρχει στη μνήμη του, τότε καταγράφεται εκεί και στη συνέχεια ο κόμβος ακολουθεί τη διαδικασία που υπαγορεύει το PFR πρωτόκολλο για την προώθηση της πληροφορίας. Ωστόσο, αν το συμβάν υπήρχε ήδη στη μνήμη του κόμβου, τότε το μήνυμα απορρίπτεται και δε γίνονται παραπέρα ενέργειες για τη διάδοση του.

Γενικά, είναι ασφαλές να υποθέσει κανείς ότι ένα μέγεθος μνήμης (πχ μέχρι 2 MB) μπορεί να είναι επαρκές για τους σκοπούς του SW-PFR. Ας σημειωθεί ότι στο μέλλον μπορούν να ενσωματωθούν στο πρωτόκολλο πολιτικές περιορισμού του χρόνου ζωής μιας καταχώρησης στη μνήμη ενός κόμβου, περιορίζοντας έτσι τις απαιτήσεις για αποθηκευτικό χώρο στο ελάχιστο.

Επίσης πρόκληση για έρευνα αποτελεί και η δυνατότητα ενσωμάτωσης τεχνικών πολύπλεξης πληροφορίας που βρίσκεται στη μνήμη του κόμβου (data aggregation) πριν να προωθηθεί σε γειτονικούς κόμβους, ώστε να μειωθεί η συνολική κυκλοφορία μηνυμάτων στο δίκτυο.

Η δεύτερη προσθήκη που έγινε στο αρχικό PFR, είναι η ενσωμάτωση ενός sleep - awake μηχανισμού εξοικονόμησης ενέργειας. Σύμφωνα με το μηχανισμό αυτό, κάθε κόμβος μεταβαίνει μεταξύ εναλλασσόμενων καταστάσεων όπου είναι είτε AWAKE είτε SLEEPING. Στις περιόδους που ο κόμβος είναι σε SLEEP κατάσταση, κάθε επικοινωνία με το περιβάλλον διακόπτεται και η κατανάλωση ενέργειας είναι ελάχιστη έως πρακτικά ασήμαντη. Αντίθετα όταν ο κόμβος είναι σε AWAKE κατάσταση, κατανάλωνει το αναμενόμενο ποσό ενέργειας για κάθε λειτουργία που επιτελεί. Επιπλέον, ένα πρόσθετο ποσό ενέργειας θεωρείται ότι ξοδεύεται για τη μετάβαση από mode SLEEP σε mode AWAKE. Το ποσό αυτό ονομάζεται power-up energy.

Η τεχνική του SLEEP AWAKE στηρίζεται στα πρότυπα που τίθενται στην εργασία [7] από τους I. Χατζηγιαννάκη και Σ. Νικολετσέα. Αρχικά προκαθορίζεται η αναλογία και η διάρκεια

των περιόδων SLEEP και AWAKE. Η διάρκεια των περιόδων είναι καθολική, δηλαδή ισχύει για όλους τους κόμβους και μπορεί να είναι hardcoded στα κυκλώματα τους. Ας υποτεθεί ότι T_α είναι η διάρκεια της AWAKE περιόδου και T_s η διάρκεια της SLEEP περιόδου. Κάθε κόμβος διατηρεί ένα χρονομέτρη timer, ο οποίος αρχικά τίθεται σε μια τυχαία χρονική τιμή που επιλέγεται μέσα στο διάστημα $(0, T_s + T_\alpha]$. Μετά την αρχικοποίηση του ο timer μετράει αντίστροφα προς το 0. Όταν ο timer βρίσκεται στο διάστημα $(T_s, T_s + T_\alpha]$ ο κόμβος είναι σε AWAKE mode, ενώ όταν βρίσκεται στο διάστημα $(0, T_s]$, τότε είναι σε κατάσταση SLEEP. Ο μηδενισμός του timer συνεπάγεται μετάβαση σε AWAKE κατάσταση και επαναρχικοποίηση του σε τιμή $T_\alpha + T_s$. Η τυχαία αρχικοποίηση του timer που συμβαίνει αρχικά (σε φάση πριν την έναρξη του πρωτοκόλλου (pre deployment)) εξασφαλίζει ότι οι κόμβοι θα πέφτουν με τυχαίο τρόπο σε SLEEP κατάσταση και όχι όλοι μαζί ταυτόχρονα.

Κεφάλαιο 4

Το πρωτόκολλο TEEN και η τροποποίηση μας

Το πρωτόκολλο TEEN (Threshold sensitive Energy Efficient sensor Network protocol) προέρχεται από μια τροποποίηση του προγενέστερου πρωτοκόλλου LEACH (Low-energy Adaptive Clustering Hierarchy) που προτάθηκε στο [11] από τους W. R. Heinzelman, A. Chandrakasan και H. Balakrishnan. Και τα δύο αυτά πρωτόκολλα είναι iεραρχικής φύσης και εστιάζουν σε μεγάλο βαθμό στην δρομολόγηση πληροφορίας προς το κέντρο ελέγχου με τον πιο αποδοτικό από πλευράς κατανάλωσης ενέργειας τρόπο. Υπόσχονται τόσο χαμηλή κατανάλωση ενέργειας ανα μετάδοση όσο και ομοιόμορφη κατανομή της κατανάλωσης της ενέργειας σε όλο το δίκτυο ανεξάρτητα από την ιδιαίτερη κατεύθυνση διάδοσης πληροφορίας. Η κεντρική διαφορά των δύο πρωτοκόλλων είναι ότι το LEACH είναι παθητικό (proactive) πρωτόκολλο, ενώ το TEEN είναι αντιδραστικό (reactive).

Το TEEN παρουσιάζεται στο paper [15] από τους Arati Manjeshwar και Dharma P.Agrawal. Ως αντιδραστικό πρωτόκολλο, εστιάζει στην ανάγκη για άμεση ανταπόκριση του δικτύου σε αλλαγές ή συμβάντα που ενδιαφέρουν το κέντρο ελέγχου. Παράλληλα ενδυναμώνει τον διαχειριστή του δικτύου να μεταβάλλει κάποιες παραμέτρους του πρωτοκόλλου, ώστε με αποδοτικό τρόπο να καθορίζει την επιθυμητή ισορροπία μεταξύ χαμηλής κατανάλωσης ενέργειας, αξιόπιστης διάδοσης και σύντομων χρόνων απόκρισης.

Το μοντέλο δικτύου που υποστηρίζει το πρωτόκολλο TEEN είναι το μοντέλο πηγής - Sink. Το κέντρο ελέγχου Sink ή αλλιώς Base Station βρίσκεται τοποθετημένο σε κάποιο σημείο του χώρου ανάπτυξης του δικτύου. Όλοι οι κόμβοι ξεκινούν με το ίδιο ποσό αρχικής διαθέσιμης ενέργειας, ενώ ο Base Station έχει άπειρους διαθέσιμους ενεργειακούς πόρους.

Το δίκτυο οργανώνεται χρησιμοποιώντας iεραρχικούς clusters. Αυτό σημαίνει ότι οι κόμβοι που το απαρτίζουν διαιρούνται σε υποσύνολα τα οποία ονομάζονται clusters. Κάθε cluster έχει και ένα κόμβο αρχηγό που ονομάζεται cluster-head. Οι cluster heads μπορούν να μεταδίδουν μηνύματα στο base station, ο οποίος είναι η ρίζα της δενδρικής αυτής iεραρχίας, βρίσκεται δηλαδή στο υψηλότερο iεραρχικά επίπεδο.

Το σχήμα που περιγράφει τη γενική λειτουργία του TEEN πρωτοκόλλου είναι το παρακάτω

- 1) Όλοι οι κόμβοι που ανήκουν σε ένα cluster και πρέπει να μεταδώσουν ένα μήνυμα το στέλνουν στον cluster head,

- 2) Μόνο ο cluster head εκτελεί πρόσθετους υπολογισμούς και επεξεργασία πάνω στα μηνύματα που συγκεντρώνει.
- 3) Οι cluster heads δεν είναι μόνιμοι. Αντίθετα κάθε cluster head διατηρεί την ιδιότητα του αρχηγού μόνο για μια προκαθορισμένη περίοδο και κατόπιν επανέρχεται στην κατάσταση απλού κόμβου. Στο σημείο αυτό κάποιοι άλλοι κόμβοι αναλαμβάνουν ρόλο cluster head. Με τον τρόπο αυτό η κατανάλωση ενέργειας κατανέμεται ομοιόμορφα σε όλους τους κόμβους του δικτύου.

Πιο συγκεκριμένα, η λειτουργία του TEEN διαιρείται σε εποχές οι οποίες με τη σειρά τους απαρτίζονται από διαδοχικούς εποχικούς γύρους. Ο κάθε εποχικός γύρος χωρίζεται σε δύο φάσεις : τη φάση προετοιμασίας (set up) και τη κυρίως φάση (steady phase). Η φάση προετοιμασίας είναι σχετικά σύντομη σε σχέση με τη κυρίως φάση.

Στη φάση προετοιμασίας (αλλαγής των clusters), σχηματίζονται οι νέοι clusters. Αυτό πραγματοποιείται με κάθε κόμβο να παίρνει την απόφαση για το αν θα γίνει cluster head ή όχι. Γενικά με πειραματικές μετρήσεις μπορεί να βρεθεί ένα κατάλληλο ποσοστό κόμβων που θα πρέπει να είναι cluster heads πχ 5%. Το ποσοστό αυτό θεωρείται γνωστό εκ των προτέρων. Η απόφαση που λαμβάνει ο κάθε κόμβος στηρίζεται στην τιμή αυτού του ποσοστού και στο πόσος χρόνος έχει περάσει από την τελευταία φορά που ο ίδιος κόμβος ήταν cluster head. Ο κόμβος διαλέγει έναν αριθμό μεταξύ 0 και 1 και αν αυτός ο αριθμός είναι μικρότερος από ένα κατώφλι $T(n)$, τότε ο κόμβος γίνεται cluster head για την υπόλοιπη περίοδο του cluster. Το κατώφλι δίνεται από τη σχέση:

$$T(n) = \begin{cases} \frac{P}{1-P(r \bmod \frac{1}{P})} & \text{αν ο κόμβος } n \text{ δεν έγινε cluster heads στους} \\ & \text{τελευταίους } \frac{1}{P} \text{ εποχικούς γύρους} \\ 0 & \text{αλλιώς} \end{cases}$$

Το πηλίκο $\frac{1}{P}$ ονομάζεται και εποχή TEEN (TEEN EPOCH) και καθορίζει τη διάρκεια μιας εποχής TEEN σε μονάδες εποχικών γύρων. Ένας οποιοσδήποτε κόμβος πρέπει να γίνει οπωσδήποτε cluster head, στη διάρκεια μιας TEEN EPOCH, και οι κόμβοι που εξελέχθηκαν ως cluster heads δεν μπορούν να επανεκλεγούν cluster heads μέχρι τη λήξη μιας εποχής TEEN.

Όταν ένας κόμβος αποφασίζει ότι θα γίνει cluster head, διαφημίζει την νέα ιδιότητα του στους υπόλοιπους κόμβους του δικτύου. Οι υπόλοιποι κόμβοι που δεν είναι cluster heads λαμβάνοντας τη διαφήμιση αυτή, αποφασίζουν σε ποιο cluster ανήκουν, κυρίως βάσει της ισχύος του λαμβανόμενου σήματος. Έπειτα ανακοινώνουν αυτή την απόφαση στον κατάλληλο cluster head. Στο τέλος της σχετικά σύντομης αυτής φάσης προετοιμασίας, έχει αποκατασταθεί μια δενδρική ιεραρχία στο δίκτυο, η οποία υπαγορεύει τα μονοπάτια αποστολής μηνυμάτων. Αφού διαμορφωθούν οι clusters ο κάθε νέος cluster head μεταδίδει σε όλα τα μέλη του cluster τα χαρακτηριστικά (A) προς ανίχνευση, τα οποία κατ' ουσία προσδιορίζουν ποιοι αισθητήρες των κόμβων πρέπει να λειτουργούν και ποια γεγονότα θεωρούνται ενδιαφέροντα προς ανίχνευση. Επιπρόσθετα ο cluster head μεταδίδει δύο παραμέτρους:

- 1) **Ισχυρό κατώφλι (Hard Threshold)**, το οποίο θέτει τη τιμή κατωφλιού για το χαρακτηριστικό προς ανίχνευση, πέρα από την οποία ένας κόμβος θα πρέπει να θέσει σε λειτουργία τον πομπό του και να αποστέλλει στον cluster head την πληροφορία που συνέλεξε.

- 2) **Ασθενές κατώφλι (Soft Threshold)**, το οποίο ορίζει την ελάχιστη αλλαγή στην τιμή του χαρακτηριστικό προς ανίχνευση, ικανή ώστε να προκαλέσει νέα μετάδοση πληροφορίας.

Στην κυρίως φάση ενός εποχικού γύρου (Steady phase), οι κόμβοι μπορούν να μεταδίδουν όταν αποφασίσουν ότι έχουν συλλέξει επείγουσα πληροφορία. Όλοι οι κόμβοι έχουν διαρκώς ανοιχτούς τους αισθητήρες τους για να ανιχνεύσουν ενδιαφέροντα γεγονότα. Ωστόσο οι πομποί (transmitters) λειτουργούν μόνο όταν χρειάζεται να γίνει μια μετάδοση. Επιπρόσθετα οι μεταδόσεις δε γίνονται σε τακτικά προγραμματισμένα διαστήματα αλλά μόνο όταν πραγματοποιηθούν δραματικές αλλαγές στα φαινόμενα που κάθε κόμβος παρατηρεί. Όταν ένας κόμβος αποστέλλει δεδομένα, ως αποτέλεσμα μιας απότομης αλλαγής στην τιμή ενός χαρακτηριστικού προς ανίχνευση, τότε η τρέχουσα τιμή αυτού του χαρακτηριστικού αποθηκεύεται σε μια τοπική μεταβλητή *SV* (sensed value). Στο χρονικό διάστημα που διαρκεί η steady phase, ένας κόμβος μεταδίδει πληροφορία όταν το ισχυρό κατώφλι ξεπεραστεί για πρώτη φορά μέσα στο διάστημα αυτό. Για να γίνει νέα εκπομπή πληροφορίας από τον ίδιο κόμβο στην ίδια χρονική περίοδο όμως πρέπει να συμβαίνουν και τα δύο από τα παρακάτω:

- α) Η τρέχουσα τιμή του χαρακτηριστικού προς ανίχνευση να ξεπερνάει το ισχυρό κατώφλι και
- β) Η τρέχουσα τιμή του χαρακτηριστικού προς ανίχνευση να διαφέρει από την τιμή της μεταβλητής *SV*, κατά ποσότητα ίση ή μεγαλύτερη από την τιμή του ασθενούς κατώφλιού.

Ο ρόλος του ισχυρού κατωφλιού είναι να καθορίσει αυστηρά το εύρος τιμών που ενδιαφέρουν το κέντρο ελέγχου για ένα χαρακτηριστικό προς ανίχνευση. Όλες, λοιπόν, οι μεταδόσεις προς το κέντρο ελέγχου όμως πρέπει να μεταφέρουν πληροφορία μόνο για “ενδιαφέρουσες” τιμές των χαρακτηριστικών. Ο ρόλος του ασθενούς κατωφλιού είναι να αποτρέπει διαδοχικές μεταδόσεις που ακολουθούν την πρώτη μετάδοση πληροφορίας για ένα ενδιαφέρον συμβάν μέσα στη διάρκεια ενός εποχικού γύρου. Έτσι αποφεύγεται η μετάδοση πλεονάζουσας πληροφορίας στο δίκτυο. Τα δύο κατώφλια συνδυαζόμενα, αποσκοπούν στον αποδοτικό περιορισμό των μεταδόσεων στο δίκτυο, ώστε να συμβαίνουν μόνο οι πιο απαραίτητες και επείγουσες από αυτές.

Ο cluster head διατηρεί τον δέκτη του συνέχεια ανοιχτό κατά τη διάρκεια ενός εποχικού γύρου, ώστε να λάβει όλα τα μηνύματα από τους κόμβους στο cluster που διευθύνει. Στη συνέχεια συνενώνει όσα περισσότερα μηνύματα μπορεί και αποστέλλει ένα ενιαίο μήνυμα στον κέντρο ελέγχου. Ο χρόνος που λήγει ένας εποχικός γύρος είναι γνωστός σε όλους τους κόμβους με κάποιο τρόπο, ώστε να συντονιστούν στην επανοργάνωση του δικτύου σε clusters.

Ένα πλεονέκτημα του TEEN που αξίζει να αναφερθεί είναι ότι σε κάθε φάση αλλαγής των clusters, ένας τελικός χρήστης μπορεί να αλλάξει τις τιμές των κατωφλιών και οι καινούργιες τιμές όμως μεταδοθούν εκ νέου. Με τον τρόπο αυτό ο τελικός χρήστης μπορεί να ζυγοσταθμίσει κατά το δοκούν την ποιότητα στην παρεχόμενη πληροφορία (QoS) και την κατανάλωση ενέργειας.

Μπορούμε να συνοψίσουμε στον παρακάτω πίνακα κάποια από τα πλεονεκτήματα και μειονεκτήματα του TEEN.

Πλεονεκτήματα	Μειονεκτήματα
Οι πληροφορίες που επείγουν προς μετάδοση, φτάνουν σχεδόν άμεσα στο σταθμό βάσης.	Αν το ισχυρό κατώφλι δε ξεπεραστεί ποτέ, λόγω κακής επιλογής του, δε θα συμβεί τότε ποτέ κάποια μετάδοση και ο σταθμός βάσης δε θα γνωρίζει τίποτα για το δίκτυο, ακόμα και όταν όλοι οι κόμβοι εξαντληθούν ενεργειακά.
Οι μεταδόσεις πληροφορίες είναι λιγότερο συχνές από τις αντίστοιχες στα παυθητικά δίκτυα. Καθώς οι μεταδόσεις είναι η κύρια πηγή κατανάλωσης ενέργειας στους κόμβους, κατά συνέπεια έχουμε καλύτερη διαχείριση ενέργειας.	Όλοι οι κόμβοι πρέπει να διατηρούν ενός είδους συγχρονισμό ώστε να γνωρίζουν πότε λήγει η περίοδος cluster και να συντονίζονται για την επανοργάνωση του δικτύου.
Το ασθενές κατώφλι μπορεί να καθορίζει κατά πόσο επείγον είναι κάποιο γεγονός.	Η διαθέσιμη ενέργεια ενός κόμβου δε λαμβάνεται υπόψη στην απόφαση για να γίνει cluster head. Αυτό μπορεί να έχει αποτέλεσμα την εκλογή κόμβου ως cluster head χωρίς τη δυνατότητα προώθησης μηνυμάτων προς το κέντρο ελέγχου.
Ελέγχοντας το ασθενές κατώφλι, ένας τελικός χρήστης ελέγχει την ισορροπία μεταξύ ποιότητας της πληροφορίας και ενεργειακής κατανάλωσης.	Πρακτικά προβλήματα δημιουργούνται λόγω συγκρούσεων που μπορούν να συμβούν κατά την ασύγχρονη μετάδοση πληροφορίας σε ένα cluster. Προτεινόμενες λύσεις για το θέμα αυτό είναι η χρήση τεχνικών CDMA και TDMA αν και η χρήση τους πρέπει να εξετασθεί διεξοδικά για να διαπιστωθεί αν είναι εφικτή και αποδοτική.
Ο τελικός χρήστης μπορεί να αλλάξει τα χαρακτηριστικά που μεταδίδονται στις φάσεις αλλαγής (setup) των cluster.	

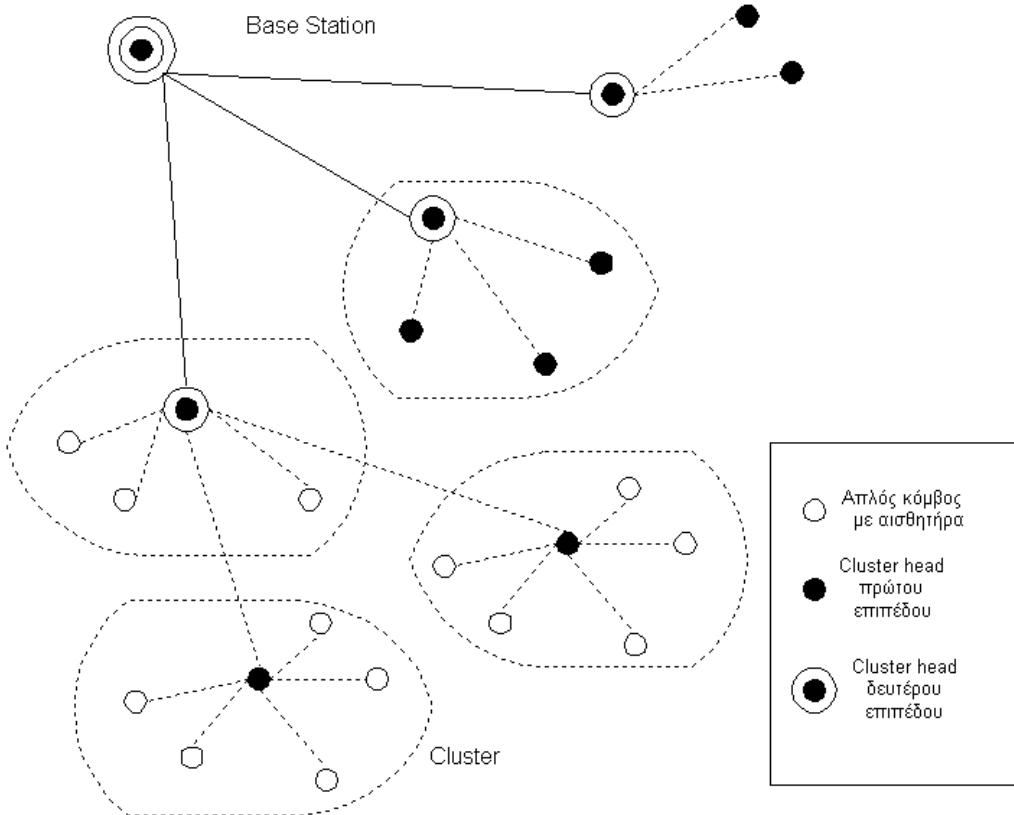
Το TEEN απαιτεί απευθείας μετάδοσεις προς σημείο, όποτε οι κόμβοι θα πρέπει να είναι εξοπλισμένοι με οπτικά μέσα μετάδοσης. Επιπρόσθετα, το TEEN προϋποθέτει ότι οι κόμβοι διαθέτουν την κατάλληλη εμβέλεια μετάδοσης, ώστε να φτάνουν το κέντρο ελέγχου από οποιοδήποτε σημείο του πεδίου, όταν είναι cluster heads.

Κάποιες από τις πιθανές εφαρμογές του TEEN είναι η ανίχνευση εχθρικής εισβολής και η ανίχνευση έκρηξης σε μηχανοστάσιο, όπου είναι υψίστης σημασίας η γρήγορη ειδοποίηση του κέντρου ελέγχου.

4.1 H-TEEN

Στις εξομοιώσεις που πραγματοποιήθηκαν στα πλαίσια της διπλωματικής υλοποιήθηκε μια εκδοχή του TEEN πρωτοκόλλου το H-TEEN, όπου υπάρχουν 4 επίπεδα ιεραρχίας. Ουσιαστικά δηλαδή η βασικές λειτουργίες του TEEN πρωτοκόλλου παραμένουν, αλλά υπάρχουν κάποιες μικρές προσθήκες στις δύο βασικές φάσεις κάθε εποχικού γύρου.

Στη φάση προετοιμασίας υπάρχουν επιπλέον βήματα όπου οι cluster-head κόμβοι μπορούν να οργανώνονται και αυτοί σε clusters σε ένα ανώτερο επίπεδο ιεραρχίας. Σε κάθε επίπεδο ιεραρχίας ο νέος cluster head επιλέγεται ανάμεσα στους cluster heads του αμέσως κατώτερου επιπέδου. Αυτό μπορεί να συμβεί για το πολύ 4 επίπεδα ιεραρχίας και για κάθε επίπεδο είναι γνωστό εκ των προτέρων το ποσοστό των κόμβων που μπορούν να γίνουν cluster heads. Οι clusters που δημιουργούνται σε επίπεδο ανώτερο του 1ου ονομάζονται superclusters και αντίστοιχα οι cluster heads ονομάζονται super cluster heads. Έτσι δημιουργείται μια δενδρική ιεραρχία για την προώθηση μηνυμάτων προς το κέντρο ελέγχου και έτσι η διάδοση των μηνυμάτων γίνεται με περισσότερο multihop τρόπο από το αυθεντικό TEEN πρωτόκολλο.



Σχήμα 4.1: Η ιεραρχική οργάνωση του δικτύου με την εφαρμογή του H-TEEN

Επιπρόσθετα στο H-TEEN πρωτόκολλο υποθέτουμε ότι ο κάθε cluster head καταρτίζει και μεταδίδει στους κόμβους του cluster που ανήκει, ένα TDMA πρόγραμμα μεταδόσεων. Ο σκοπός του TDMA προγράμματος είναι ο συντονισμός των κόμβων του cluster ώστε να ελαχιστοποιηθεί πιθανότητα συγκρούσεων στις μεταδόσεις τους. Τέλος για να αποφευχθούν οι συγκρούσεις στις μεταδόσεις που γίνονται σε γειτονικούς clusters υποθέτουμε τη χρήση της CDMA τεχνικής. Οι μεταδόσεις δηλαδή σε κάθε cluster γίνονται με κώδικες CDMA διαφορετικούς από αυτούς που χρησιμοποιούνται σε γειτονικούς clusters.

Στην κυρίως φάση του H-TEEN, ένας κόμβος που διαθέτει δεδομένα προς μετάδοση, εκπέμπει πληροφορία προς τον cluster head. Ο cluster head μεταδίδει την πληροφορία στο δικό του super cluster head και αυτό γίνεται μέχρι να φτάσει η πληροφορία στο κέντρο ελέγχου.

Μέρος II

Εξομοίωση πρωτοκόλλων

Κεφάλαιο 5

Περιβάλλον εξομοίωσης

Στα πλαίσια της παρούσας διπλωματικής επιδιώχθηκε η εξομοίωση κάποιων από τα πιο διαδεδομένα πρωτόκολλα για τα δίκτυα έξυπνης σκόνης, με τη βοήθεια ηλεκτρονικού υπολογιστή. Έτσι αναπτύχθηκε ένας πρότυπος εξομοιωτής για δίκτυα έξυπνης σκόνης στον οποίο μπορούν να εξομοιωθούν πολλά από τα γνωστά πρωτόκολλα επικοινωνίας. Παράλληλα με τον εξομοιωτή αναπτύχθηκε και ένας οπτικός εξομοιωτής (animator), ο οποίος εξυπηρετεί σκοπούς εποπτικής θεώρησης της λειτουργίας των υλοποιημένων πρωτοκόλλων αλλά και επιβεβαίωσης (verification) της σωστής λειτουργίας τους.

5.1 Βασικά χαρακτηριστικά του εξομοιωτή

Ο εξομοιωτής μας ονομάζεται simDust και αναπτύχθηκε σε περιβάλλον Linux, σε γλώσσα προγραμματισμού C++ και με τη χρησιμοποίηση των βιβλιοθηκών της LEDA 4.3.1 [16].

Ο σχεδιασμός του εξομοιωτή simDust στηρίχθηκε στον πηγαίο κώδικα που είχε αναπτυχθεί για τις ανάγκες του paper [6]. Πάνω στον κώδικα αυτό έγιναν αρκετές διορθώσεις, αλλαγές και προσθήκες, με αποτέλεσμα να αυξηθεί η λειτουργικότητα του. Γενικά στο σχεδιασμό και την οργάνωση του πηγαίου κώδικα για τον νέο εξομοιωτή δόθηκε ιδιαίτερη έμφαση στη δημιουργία μίας κατάλληλης υποδομής, ώστε να είναι δυνατή και σχετικά απλή η διεύρυνση του συνόλου των πρωτοκόλλων επικοινωνίας που αυτός εξομοιώνει. Έτσι η προσθήκη νέων πρωτοκόλλων προς εξομοίωση είναι για το simDust μια διαδικασία από τυποποιημένα βήματα.

Από τα πλεονεκτήματα τις χρήσης C++ για την υλοποίηση του εξομοιωτή είναι ότι μας δόθηκε η δυνατότητα να πειραματιστούμε με μεγάλα μεγέθη δικτύων, κάτι που ήταν ανέφικτο ή υπερβολικά χρονοβόρο με υπάρχοντες πλατφόρμες εξομοίωσης, όπως ο ns2.

Επίσης ο προγραμματισμός νέων πρωτοκόλλων που μπορούν να προστεθούν στον simDust απαιτεί τη γνώση αποκλειστικά και μόνο της C++ και όχι πολλών προγραμματιστικών γλωσσών, αποφεύγοντας έτσι να εισάγει περιττές δυσκολίες στην επέκταση του εξομοιωτή.

Τέλος, ο εξομοιωτής κατά τη διάρκεια εξομοίωσης ενός πρωτοκόλλου παράγει ένα αρχείο με στοιχεία περιγραφικά για τη λειτουργία του πρωτοκόλλου, και με στοιχεία στατιστικά για ενδιαφέροντα μεγέθη και χαρακτηριστικά.

Προς το παρόν στον εξομοιωτή έχουν υλοποιηθεί τα πρωτόκολλα LTP, SW-PFR, H-TEEN, τα οποία έχουν επωφεληθεί από τις πολλές δυνατότητες που προσφέρει ο εξομοιωτής simDust

και έχουν ενδυναμωθεί με νέες ικανότητες όπως είναι ο μηχανισμός SLEEP - AWAKE και ο χειρισμός πολλαπλών συμβάντων.

5.1.1 Λειτουργία σε γύρους

Στην υλοποίηση του simDust χρειάστηκαν να γίνουν κάποιες παραδοχές χάριν της απλότητας στην υλοποίηση και την παραγωγή ενός εκτελέσιμου προγράμματος σε εύλογο χρονικό διάστημα. Η βασικότερη παραδοχή είναι ότι η εξομοιώση κάθε πρωτοκόλλου εξελίσσεται σε διακριτούς γύρους που διαδέχονται ο ένας τον άλλο. Ορίζουμε την έννοια ενός αυτόνομου γύρου, ως το χρονικό διάστημα στο οποίο ένας κόμβος αρχικά ελέγχει αν έχει μηνύματα, κατόπιν επεξεργάζεται πληροφορίες όπως υπαγορεύει το πρωτόκολλο και αποστέλλει τον απαραίτητο αριθμό μηνυμάτων.

Η λειτουργία του εξομοιωτή σε γύρους μπορεί να ξενίζει αν λάβουμε υπόψη την εντελώς ασύγχρονη φύση των ασύρματων δικτύων με αισθητήρες. Ωστόσο η ανάπτυξη ενός εξομοιωτή για real-time ασύγχρονο δίκτυο συνεπάγεται προγραμματισμό με τεχνικές multi-threading και εισαγωγή πρόσθετων παραμέτρων στη σχεδίαση και ανάπτυξη του κώδικα, που θα το καθιστούσαν πολύ πολύπλοκο τόσο για το μέσο χρήστη όσο και για το πρόσωπο που θα κληθεί να συντηρήσει τον κώδικα και να προσθέσει νέα στοιχεία. Επιπρόσθετα, η χρήση τεχνικών πολυνηματισμού, απαιτεί συστήματα με παράλληλους επεξεργαστές ή επεξεργαστές multi-threading, και με αρκετή διαθέσιμη μνήμη, αυξάνοντας έτσι δραματικά τις απαιτήσεις σε hardware για τον εξομοιωτή.

Μπορούμε λοιπόν να υποθέτουμε ότι δε μας απασχολεί ο συγχρονισμός των κόμβων καθώς κάποιο είδος συγχρονισμού για ασύγχρονα δίκτυα, θα μπορούσε πραγματοποιείται σε κάποιο υφιστάμενο επίπεδο, τις υπηρεσίες του οποίου θα εκμεταλλεύονται τα πρωτόκολλα διάδοσης πληροφορίας.

Έπειτα η λειτουργία του simDust σε γύρους βοηθάει περισσότερο στην ευχρηστία του οπτικού εξομοιωτή, που υλοποιήσαμε, και την κατανοητή οπτικοποίηση των πρωτοκόλλων.

5.1.2 MAC ζητήματα

Πρέπει να σημειωθεί ότι για λόγους απλότητας στην υλοποίηση άλλα και για εύκολη παρακολούθηση της πορείας της εξομοιώσης, ο simDust εξομοιωτής δεν ασχολείται με θέματα συγχρούσεων στις μεταδόσεις. Υποθέτουμε την ύπαρξη ενός υφιστάμενου MAC επίπεδου που διευθετεί θέματα κοινής πρόσβασης στο ασύρματο μέσο μετάδοσης. Τις υπηρεσίες αυτού του επιπέδου χρησιμοποιεί ο εξομοιωτής, ώστε να αποφευχθεί η πολυπλοκότητα στον κώδικα του και να δοθεί έμφαση στην κατεξοχήν λειτουργία των πρωτοκόλλων.

5.1.3 Λειτουργία SLEEP - AWAKE

Αυτή η λειτουργία είναι ενσωματωμένη στη σχεδίαση του simDust, και υλοποιείται όπως προτείνεται από το [7]. Η διάρκεια των περιόδων SLEEP - AWAKE προκαθορίζεται ως παράμετρος στην κλήση του εξομοιωτή και είναι κοινή για όλα τα σωματίδια - κόμβους του δικτύου (particles). Πρέπει να σημειωθεί ότι αυτό που ορίζεται στη γραμμή εντολών είναι η

διάρκεια σε γύρους των περιόδων SLEEP - AWAKE και όχι απλά η αναλογία των περιόδων αυτών.

Αν υποθέσουμε ότι T_s είναι η διάρκεια της περιόδου SLEEP και T_α η διάρκεια της περιόδου AWAKE, τότε το άνθροισμα τους ορίζει ένα πλαίσιο χρόνου T (timeframe). Κάθε κόμβος στην φάση αρχικοποίησης του επιλέγει τυχαία ένα αριθμό x που θα πρέπει ωστόσο να ανήκει στο πλαίσιο T . Σε κάθε γύρο της εξομοίωσης ο αριθμός x μειώνεται και ανα πάσα στιγμή καθορίζει σε ποιά κατάσταση βρίσκεται ο κόμβος. Έτσι αν $x < T$ τότε το particle είναι σε κατάσταση SLEEP, ενώ όταν $T_s < x < T_\alpha$ τότε το particle είναι AWAKE. Επίσης όταν το x γίνει αρνητικό, τότε το επαναφέρουμε σε τιμή T και το particle μεταβαίνει σε κατάσταση AWAKE.

Όταν ένα particle είναι σε λειτουργία SLEEP, δεν μπορεί να λάβει μήνυμα, γιατί έχει κλειστό το δέκτη του. Ωστόσο, μπορεί να ανιχνεύσει ένα γεγονός, το οποίο θα ξεκινήσει να διαδίδει στο γύρο, όπου θα επανέλθει σε κατάσταση AWAKE.

Η λειτουργία SLEEP-AWAKE είναι διάφανη και απομονωμένη για τα πρωτόκολλά. Αν αυτό είναι απαραίτητο, μπορούν να γίνουν επεμβάσεις και να παρακαμφεί η προκαθορισμένη διαδικασία των μεταβάσεων SLEEP-AWAKE, θέτοντας ρητά την κατάσταση του particle σε μια από τις επιτρεπτές καταστάσεις. Προκειμένου να αγνοηθεί εντελώς η λειτουργία SLEEP-AWAKE, ο χρήστης του simulator μπορεί απλά να ορίσει αναλογία SLEEP-AWAKE περιόδων 0:0 στις παραμέτρους του εκτελέσιμου. Εναλλακτικά, ως πιο δραστική λύση για ένα πρωτόκολλο που δεν επιθυμεί το SLEEP-AWAKE, μπορεί στο σχεδιασμό του πρωτοκόλλου να ορίζεται ότι όλα τα particles αρχικοποιούνται σε κατάσταση ALWAYS-ON, το οποίο σημαίνει ότι όλοι οι κόμβοι θα είναι πάντοτε AWAKE.

5.1.4 Κατανάλωση ενέργειας

Για την κατανάλωση ενέργειας ο εξομοιωτής simDust χρησιμοποιεί ένα ειδικό μοντέλο παρόμοιο με το προτεινόμενο από το paper παρουσίασης του πρωτοκόλλου LEACH [11]. Σε κάθε γύρο ο κάθε κόμβος που βρίσκεται σε κατάσταση AWAKE ξοδεύει σταθερά ένα ποσό ενέργειας πχ E_{elec} το οποίο χρησιμοποιείται για τη λειτουργία των διαφόρων κυκλωμάτων ανίχνευσης και επεξεργασίας πληροφορίας.

Επιπρόσθετα σε αυτή την ενέργεια πρέπει να συνυπολογίζεται η ενέργεια που καταναλώνεται κατά τη λήψη ή αποστολή μηνυμάτων. Η ενέργεια που ξοδεύεται για την αποστολή μηνυμάτων είναι ανάλογη του τετραγώνου της απόστασης στη οποία η πληροφορία θα διαδοθεί με αξιοπιστία. Είναι επίσης ανάλογη του μεγέθους σε bits της μεταδιδόμενης πληροφορίας, ενώ στον υπολογισμό της υπεισέρχεται και η ενέργεια που απαιτείται για τη λειτουργία των ηλεκτρονικών κυκλωμάτων του πομπού.

$$E_{trans} = E_{elec} * k + e_{amp} * k * d^2$$

όπου k είναι ο αριθμός των bits του μεταδιδόμενου μηνύματος e_{amp} είναι η ενέργεια για τη λειτουργία του ενισχυτή σήματος στον πομπό, και d η απόσταση μετάδοσης του σήματος

Για τη λήψη μηνυμάτων η καταναλισκόμενη ενέργεια ανά λαμβανόμενο bit αντιστοιχεί σε ενέργεια που δαπανάται για τη λειτουργία των κυκλωμάτων του δέκτη και ψεωρείται ίση με E_{elec} . Συνεπώς η συνολική ενέργεια που δαπανάται κατά την λήψη ενός μηνύματος k bits είναι:

$$E_{rec} = E_{elec} * k$$

Στην περίπτωση που ένας κόμβος βρίσκεται σε κατάσταση SLEEP, τότε καθώς τα βασικά του ηλεκτρικά κυκλώματα δε λειτουργούν, υποθέτουμε μηδενική ενέργειακή κατανάλωση. Ωστόσο η μετάβαση από τη κατάσταση SLEEP σε κατάσταση AWAKE απαιτεί ένα σημαντικό ποσό ενέργειας που θέτουμε ίσο με $3E_{elec}$.

Αν και γενικά το αρχικό ποσό της ενέργειας που διαθέτουν οι κόμβοι είναι παραμετροποιήσιμο, αυτό συνήθως αφήνεται στην default τιμή του που είναι 1 Joule. Τυπικές τιμές που επιλέχθηκαν για τις σταθερές E_{elec} και e_{amp} , είναι $E_{elec} = 50 \text{ nJ}$ και $e_{amp} = 100 \text{ pJ}$.

5.1.5 Μέγεθος των μηνυμάτων

Τα μηνύματα που μεταδίδονται στο δίκτυο και περιέχουν χρήσιμη πληροφορία που πρέπει να φτάσει στο κέντρο ελέγχου έχουν συγκεκριμένη δομή. Συγκεκριμένα στο simDust το μέγεθος της επικεφαλίδας ενός μηνύματος είναι 64 bits, στα οποία περιγράφεται η ταυτότητα του αποστολέα, ο τύπος του μηνύματος και ίσως πρόσθιτη πληροφορία που θα χρησιμοποιηθεί από το εκάστοτε πρωτόκολλο. Ως μέγεθος του κυρίως σώματος ενός μηνύματος ορίζεται το επαρκές μήκος των 8 Kbits.

5.2 Η δομή του εξομοιωτή

Η δομή του πηγαίου κώδικα οργανώθηκε με συστηματικό τρόπο ώστε να διαχωριστούν και να απομονωθούν κατά το δυνατό οι λεπτομέρειες υλοποίησης των κόμβων (particles) του δικτύου, οι υλοποιήσεις των πρωτοκόλλων και ο σχεδιασμός των πειραμάτων της εξομοιώσης. Καθώς ο εξομοιωτής υλοποιήθηκε σε αντικειμενοστρεφή γλώσσα προγραμματισμού (C++), έχουν οριστεί διάφορες αυτόνομες τάξεις (classes) που συνεργάζονται για τη λειτουργία του simDust. Οι βασικές τάξεις που έχουν οριστεί στον εξομοιωτή είναι αυτές των Particle, Message, Event, Info, Network, Execution και Run.

Τα αντικείμενα της τάξης Particle ουσιαστικά αντιπροσωπεύουν τους κόμβους - αισθητήρες (sensor nodes) του δικτύου. Η τάξη αυτή ορίζει τις καταστάσεις στις οποίες μπορεί να βρίσκεται ένας κόμβος και οι οποίες μπορεί να είναι:

AWAKE	Όταν λειτουργεί κανονικά
SLEEP	Όταν η συσκευή είναι σε SLEEP mode και λειτουργεί με την ελάχιστη απαιτούμενη ενέργεια
ALWAYS-ON	Όταν η συσκευή θα λειτουργεί πάντα σε κατάσταση AWAKE
SENSING	Όταν ο κόμβος δειγματοληπτεί / ανιχνεύει ένα φαινόμενο / γεγονός
DEAD	Όταν έχει εξαντληθεί η διαθέσιμη ενέργεια του κόμβου.

Στην τάξη Particle ορίζονται και οι μέθοδοι που υλοποιούν τις μεταβάσεις μεταξύ των καταστάσεων αυτών.

Από τις βασικές μεθόδους της τάξης Particle είναι και η DoProtocolStep(), η οποία υπαγορεύει τις βασικές λειτουργίες που θα επιτελέσει ένας κόμβος στη διάρκεια ενός γύρου εξομοίωσης. Σύμφωνα με αυτή τη μέθοδο σε κάθε βήμα - γύρο ενός πρωτοκόλλου, ο κάθε κόμβος αρχικά ελέγχει αν πρέπει να αλλάξει κατάσταση (να μεταβεί από SLEEP σε AWAKE ή αντίστροφα). Έπειτα ελέγχεται μια ουρά από γεγονότα που αναμένουν για επεξεργασία στον κάθε κόμβο. Τα γεγονότα αυτά είναι είτε γεγονότα λήψης μηνυμάτων είτε γεγονότα ανίχνευσης ενός νέου συμβάντος. Όλα τα γεγονότα φέρουν μια χρονοσφραγίδα που καθορίζει σε ποιο γύρο θα τα επεξεργαστεί το particle. Η DoProtocolStep() φροντίζει να βρεθούν όλα τα γεγονότα που εκκρεμούν προς επεξεργασία για τον τρέχοντα γύρο και να γίνει η επεξεργασία τους όπως υπαγορεύει το εκάστοτε πρωτόκολλο που ακολουθείται. Έπειτα επιτελούνται τυχόν πρόσθετες λειτουργίες του πρωτοκόλλου που εξομοιώνεται και τέλος ξοδεύεται ένα ποσό ενέργειας για τη λειτουργία των κυκλωμάτων του κόμβου στον τρέχον γύρο.

Οι υπόλοιπες μέθοδοι της τάξης Particle περιλαμβάνουν διαδικασίες που διαχειρίζονται την μνήμη (cache) που διαθέτει ο κάθε κόμβος, εκτελώντας λειτουργίες εγγραφής, διαγραφής, ανάγνωσης σε αυτή. Επιπρόσθετα μέθοδος της τάξης αυτής είναι και η δημιουργία (triggering) των γεγονότων (events) που θα ανιχνευτούν από συγκεκριμένο αντικείμενο Particle.

Οι τάξεις των Info, Message και Event είναι στενά συνδεδεμένες. Στην τάξη των Info ορίζονται οι ιδιότητες και τα πεδία της πληροφορίας που θα διαδίδεται στο δίκτυο. Από τα πιο χαρακτηριστικά πεδία της πληροφορίας είναι το ID του κόμβου που ξεκίνησε τη διάδοση της πληροφορίας, η χρονοσφραγίδα του αρχικού συμβάντος και τα hops που έχει κάνει κατά τη διάδοση της πληροφορία. Η τάξη του Message περιλείει ένα αντικείμενο της τάξης Info, καθώς κατά κανόνα τα μηνύματα μεταφέρουν πληροφορία, και πέραν αυτού ένα αντικείμενο Message έχει και την πληροφορία (particle ID) του αποστολέα του μηνύματος. Τέλος η τάξη των Event (γεγονότα) έχει ως μέλη ένα μήνυμα που σχετίζεται με ένα συγκεκριμένο γεγονός, ένα πεδίο που υποδεικνύει το είδος του event (λήψη μηνύματος ή ανίχνευση γεγονότος) και μία χρονοσφραγίδα που υποδεικνύει πότε πρέπει να το επεξεργαστεί ο κόμβος, στην ουρά αναμονής του οποίου βρίσκεται.

Η τάξη του Network φροντίζει για τη πάροδο των γύρων στους οποίους χωρίζουμε την εκτέλεση των πρωτοκόλλων. Επίσης προσφέρει συναρτήσεις περιτυλίγματος (wrapper) για τις σημαντικότερες λειτουργίες στο δίκτυο όπως αποστολή και λήψη μηνυμάτων, λειτουργίες broadcast και δημιουργία συμβάντος από εξωτερικό παράγοντα για να το ανιχνεύσει συγκεκριμένο particle.

Βασικό ρόλο έχει η τάξη Execution που κρατάει πληροφορίες, όπως μια λίστα όλων των particles που απαρτίζουν το δίκτυο, τις διαστάσεις του χώρου στον οποίο αναπτύσσεται το δίκτυο, τον αριθμό των γεγονότων που έχουν ανιχνευθεί, το είδος (Sink ή Wall) και τη θέση του κέντρου ελέγχου.

Πληροφορίες για τα αποτελέσματα των πειραμάτων που εκτελούνται με τον εξομοιωτή διατηρεί η τάξη Run.

5.3 Η λειτουργία του εξομοιωτή

Το εκτελέσιμο του εξομοιωτή δέχεται μία σειρά από υποχρεωτικές παραμέτρους όπως φαίνεται παρακάτω:

Παράμετρος	Σημασία	Τυπικές τιμές
Protocol	Το είδος του πρωτοκόλλου που θα εξομοιωθεί	LTP, PFR, TEEN
Setup	Η κατανομή των κόμβων στο δίκτυο	R: Random, L: Lattice, P: Poisson
Control Center	Ο τύπος και η θέση του κέντρου ελέγχου	S : σημείο στη θέση (X, Y) , W : τείχος στην ευθεία $y = Y$, C : σημείο στη θέση $(\frac{3}{4}X, \frac{1}{2}Y)$
Particles	Αριθμός των σωματιδίων που θα αποτελούν τους κόμβους του δικτύου	Ακέραιος θετικός αριθμός
Simtime	Ο χρόνος εκφρασμένος σε αριθμό γύρων, που επιθυμούμε να διαρκέσει η εξομοίωση	Ακέραιος θετικός αριθμός
Runs	Ο αριθμός των επαναλήψεων που θέλουμε για το πείραμα μας	Ακέραιος θετικός αριθμός
X	Η διάσταση X του πεδίου ανάπτυξης του δικτύου	Θετικός αριθμός κινητής υποδιαστολής
Y	Η διάσταση Y του πεδίου ανάπτυξης του δικτύου	Θετικός αριθμός κινητής υποδιαστολής
R	Η ακτίνα μετάδοσης των σωματιδίων	Θετικός αριθμός κινητής υποδιαστολής
deltaR	Καθορίζεται όταν θέλουμε να περιορίσουμε τις μεταδόσεις μεταξύ των κύκλων ακτίνας R και R - deltaR	Θετικός αριθμός κινητής υποδιαστολής (συνήθως deltaR = R)
Alpha	Γωνία μετάδοσης, για τις υπό γωνία broadcast μεταδόσεις	Τιμή σε μοίρες από 0 έως 180
W	Η διάρκεια του χρόνου παραμονής σε κατάσταση AWAKE	Θετικός ακέραιος. ($W = 0$ αν θέλουμε τα σωματίδια να είναι πάντα AWAKE)
S	Η διάρκεια του χρόνου παραμονής σε κατάσταση SLEEP	Θετικός ακέραιος. ($S = 0$ αν θέλουμε τα σωματίδια να είναι πάντα AWAKE)

Επίσης τις παραμέτρους αυτές μπορεί να ακολουθεί μια σειρά από προαιρετικές παραμέτρους που περιγράφονται παρακάτω:

Παράμετρος	Σημασία	Τυπικές τιμές
Multiple Events X	Είναι ο αριθμός των γεγονότων ή των ομάδων από γεγονότα που θα συμβούν κατά τη διάρκεια της εξομοίωσης	Θετικός ακέραιος. (αν δεν οριστεί τότε έχει την τιμή 1)
Multiple Events Y	Αν $Y \leq 1$, τότε εκφράζει την πιθανότητα να συμβεί ένα γεγονός σε κάποιο γύρο. Αν $Y > 1$, τότε τα γεγονότα θα συμβαίνουν διαδοχικά ανά ομάδες των Y γεγονότων (burst event mode)	Θετικός ακέραιος. (αν δεν οριστεί τότε έχει την τιμή 1, δηλαδή ανά γύρο θα συμβαίνει και ένα γεγονός)
Lambda	Για την LATTICE κατανομή καθορίζει την απόσταση (κάθετη και οριζόντια) μεταξύ των σωματιδίων	Θετικός ακέραιος. (αν δεν οριστεί τότε έχει την τιμή 1)
P (Print)	Καθορίζει ότι θέλουμε να βλέπουμε στην έξοδο του προγράμματος πληροφορίες για τους τελευταίους P γύρους	Θετικός ακέραιος. (αν δεν οριστεί τότε έχει την τιμή 10)

Η πρώτη εργασία του εξομοιωτή, είναι να γίνει η επεξεργασία των παραμέτρων αυτών και έπειτα να καταρτισθεί ένας πίνακας χρονοπρογραμματισμού, όπου θα δηλώνονται τα γεγονότα ανίχνευσης που πρέπει να δημιουργηθούν (triggered) κατά τη διάρκεια ενός πειράματος εξομοίωσης. Αυτό σημαίνει ότι προσδιορίζονται επακριβώς και εκ των προτέρων τα particles που θα τα ανιχνεύσουν και ο χρόνος που θα συμβούν.

Έπειτα ξεκινάει η εκτέλεση ενός πειράματος, με τη “ρίψη” των κόμβων - smartdust σε κατάλληλη κατανομή στο δεδομένο πεδίο, και την τοποθέτηση του κέντρου ελέγχου, το οποίο για παράδειγμα αν είναι σημείο - Sink, θα έχει συντεταγμένες (X, Y), που αντιστοιχούν στο ακρότατο σημείο πάνω και δεξιά, αν υποτεθεί ότι X και Y είναι οι διαστάσεις του πεδίου. Έπειτα, πριν την έναρξη του χυρίως πρωτοκόλλου προηγείται μια φάση, όπου το κάθε particle αναγνωρίζει και διαχωρίζει τους γείτονές του, ανάλογα με την απόστασή τους από αυτό και τη γωνία που σχηματίζουν η ευθεία που συνδέει το particle με το γείτονα και η ευθεία που συνδέει το particle με το Sink. Στην πράξη η φάση αυτή θα μπορούσε να υλοποιηθεί με broadcast από κάθε particle ενός μηνύματος που θα περιέχει το ID ή και τη θέση του, και κατόπιν μετρώντας τη δύναμη του λαμβανόμενου σήματος από τα γειτονικά particles να εξάγονται συμπεράσματα για τη σχετική απόσταση τους. Σε αυτή τη φάση δεν αναθέτουμε κατανάλωση ενέργειας καθώς θεωρούμε ότι γίνεται πάντα, ανεξαρτήτως πρωτοκόλλου. Συνεπώς όλα τα particles καταναλώνουν το ίδιο ακριβώς ποσό ενέργειας στη φάση αυτή, οπότε αυτό μπορεί να αγνοηθεί με ασφάλεια.

Στην εξομοίωση ακολουθεί ένα στάδιο, όπου πραγματοποιούνται τυχόν αρχικοποιήσεις στα particles, οι οποίες υπαγορεύονται από το εκάστοτε εφαρμοζόμενο πρωτόκολλο.

Η χυρίως φάση του πειράματος είναι ένας βρόχος, ο οποίος σε κάθε γύρο καλεί για κάθε particle την κεντρική συνάρτηση DoProtocolStep() που περιγράφηκε παραπάνω.

Στο τέλος του πειράματος παράγονται αρχεία καταγραφής trace, τα οποία είναι αρχεία που περιέχουν σημαντικές πληροφορίες για την εκτέλεση του πειράματος και από τα οποία μπορούν να εξαχθούν στατιστικές και χρήσιμα συμπεράσματα. Σε ένα trace file μπορούν να βρίσκονται πληροφορίες για το ποσοστό επιτυχών παραδόσεων μηνυμάτων στο κέντρο ελέγ-

χου, τη μέση κατανάλωση ενέργειας και τη διάρκεια ζωής του δικτύου. Στην πιο πρόσφατη έκδοση του o simulator παράγει δύο trace files, εκ των οποίων το ένα (result.out) είναι περιληπτικό της όλης εκτέλεσης ενός πειράματος, ενώ το δεύτερο (trace.out) δίνει στοιχεία για την κατάσταση του δικτύου ανά 10 γύρους. Παρακάτω (εικόνα 5.1) φαίνεται ένα απόσπασμα από το trace.out αρχείο.

PROT	Setup	X	Y	Part	W/S	R/angle	E-Rate	Round	RecvEvent	TotalEnv	SuckRate	TotalEng	MinEng	AvgEng	MaxEng	Alive	Awake	TotTR	TotRCV	
PFR	RAND/S	200	200	400	6/2	18/45	0.6	0	0	0	399	1	1	1	399	0	0	0	0	
PFR	RAND/S	200	200	400	6/2	18/45	0.6	10	0	6	398.9	0.99	0.9999	399	315	49	64			
PFR	RAND/S	200	200	400	6/2	18/45	0.6	20	0	12	398.8	0.99	0.9999	399	314	43	191			
PFR	RAND/S	200	200	400	6/2	18/45	0.6	30	0	17	398.7	0.99	0.9999	399	312	199	522			
PFR	RAND/S	200	200	400	6/2	18/45	0.6	40	1	23	398.6	0.99	0.9999	399	302	265	428			
PFR	RAND/S	200	200	400	6/2	18/45	0.6	50	3	30	398.5	0.99	0.9999	399	286	373	627			
PFR	RAND/S	200	200	400	6/2	18/45	0.6	60	3	35	398.4	0.99	0.9999	399	297	432	717			
PFR	RAND/S	200	200	400	6/2	18/45	0.6	70	3	41	398.4	0.99	0.9999	399	324	471	768			
PFR	RAND/S	200	200	400	6/2	18/45	0.6	80	4	49	398.3	0.98	0.99	0.9999	399	325	540	876		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	90	5	55	398.2	0.98	0.99	0.9999	399	313	647	1044		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	100	5	61	398.1	0.98	0.99	0.9999	399	316	710	1141		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	110	5	66	398.0	0.98	0.99	0.9999	399	313	771	1222		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	120	5	70	397.9	0.98	0.99	0.9999	399	320	818	1288		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	130	5	78	397.9	0.98	0.99	0.9999	399	306	879	1392		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	140	5	84	397.8	0.97	0.99	0.9999	399	189	993	1556		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	150	8	90	397.6	0.97	0.99	0.9999	399	291	1096	1738		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	160	9	97	397.5	0.97	0.99	0.9999	399	318	1163	1854		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	170	10	100	397.4	0.97	0.99	0.9999	399	326	1253	2004		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	180	10	100	397.4	0.97	0.99	0.9999	399	315	1303	2080		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	190	10	100	397.3	0.97	0.99	0.9999	399	318	1314	2095		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	200	10	100	397.3	0.97	0.99	0.9999	399	313	1314	2095		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	210	10	100	397.3	0.97	0.99	0.9999	399	321	1314	2095		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	220	10	100	397.3	0.97	0.99	0.9999	399	309	1314	2095		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	230	10	100	397.3	0.97	0.99	0.9999	399	285	1314	2095		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	240	10	100	397.3	0.97	0.99	0.9999	399	288	1314	2095		
PFR	RAND/S	200	200	400	6/2	18/45	0.6	250	10	100	397.3	0.97	0.99	0.9999	399	318	1314	2095		

Σχήμα 5.1: Αρχείο καταγραφής της λειτουργίας του εξομοιωτή

5.4 Η λειτουργία της οπτικοποίησης

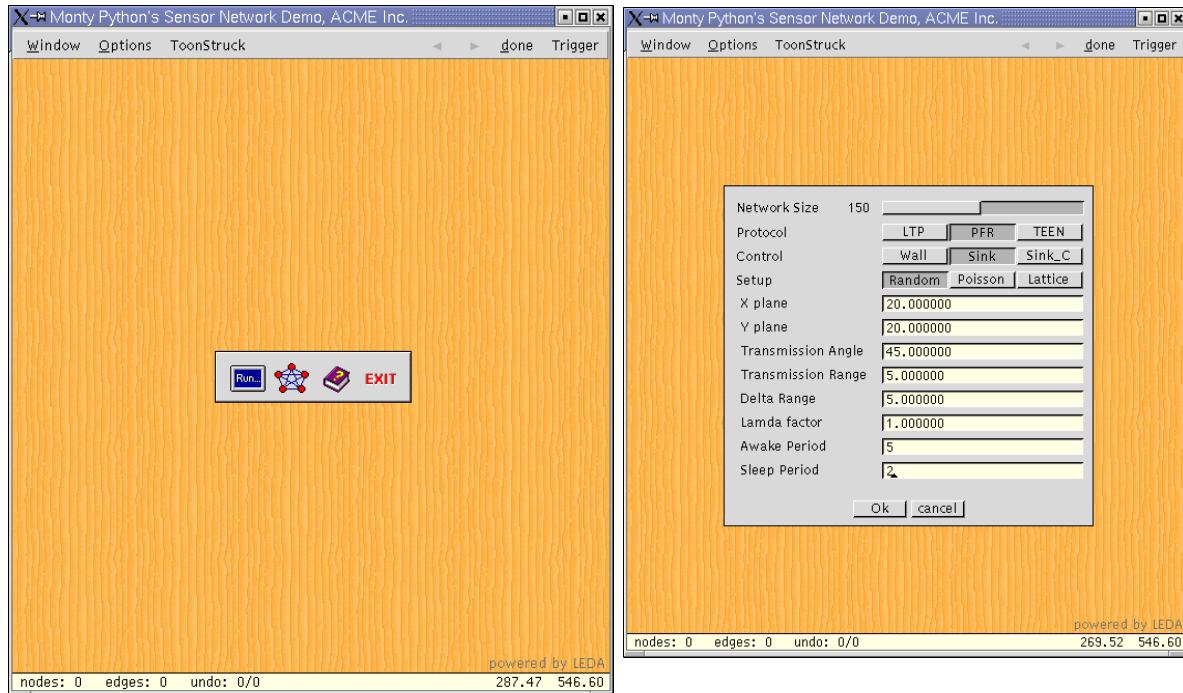
Η δημιουργία του οπτικού εξομοιωτή animDust έγινε με πρωταρχικό σκοπό την εποπτική παρακολούθηση της εφαρμογής των υλοποιημένων πρωτοκόλλων, και την εξακρίβωση της επιθυμητής λειτουργίας τους. Επιπρόσθετα, έγινε για λόγους επίδειξης της λειτουργίας των πρωτοκόλλων σε Smart Dust δίκτυα, μέσα από ένα φιλικό περιβάλλον διεπαφής (interface) με το χρήστη. Η υλοποίηση της οπτικοποίησης έγινε με τη βοήθεια της graphwin βιβλιοθήκης που ανήκει στο γενικότερο σύνολο βιβλιοθηκών της LEDA.

Ο χρήστης του οπτικού εξομοιωτή έχει πρόσβαση σε μια σειρά από παραμέτρους που είναι αντίστοιχες με τις παραμέτρους του command line εξομοιωτή (εικόνα 5.3). Μερικές από αυτές είναι το είδος του εφαρμοζόμενου πρωτοκόλλου, το είδος του control center, το μοντέλο κατανομής των κόμβων στο δίκτυο, οι διαστάσεις του χώρου όπου θα αναπτυχθεί το δίκτυο, ο αριθμός των κόμβων που θα το αποτελούν, η ακτίνα μετάδοσης του κάθε κόμβου και η διάρκεια των περιόδων SLEEP - AWAKE (προαιρετικά). Μεταβάλλοντας τις τιμές των παραμέτρων αυτών ο χρήστης μπορεί να παρακολουθεί άμεσα τις επιπτώσεις των μεταβολών στην λειτουργία των πρωτοκόλλων.

Αφού θέσει ένας χρήστης τις επιθυμητές τιμές για τις παραπάνω παραμέτρους, μπορεί να εκκινήσει τη διαδικασία οπτικής εξομοίωσης ενός πρωτοκόλλου. Αφού ξεκινήσει η οπτική παρουσίαση, γίνονται ενεργές καινούργιες επιλογές που επιτρέπουν στο χρήστη να ρυθμίσει την ταχύτητα του animation, το zooming (εστίαση και μεγέθυνση συγκεκριμένων τμημάτων του δικτύου), το “πάγωμα” του πρωτοκόλλου ανά καθορισμένο αριθμό γύρων και την αποθήκευση των παρουσιάζομενων εικόνων σε αρχείο postscript.

Η λειτουργία των πρωτοκόλλων, όπως παρουσιάζεται από τον οπτικό εξομοιωτή, αντιστοιχίζεται απόλυτα με την λειτουργία του command line εξομοιωτή simDust, καθώς ουσιαστικά επεκτάθηκαν οι συναρτήσεις του command line εξομοιωτή, με εντολές και συναρτήσεις, που

έχαναν εφικτή την οπτική απεικόνιση των λειτουργιών που επιτελούν τα πρωτοκόλλα. Ωστόσο, από τον οπτικό εξομοιωτή λείπει η λειτουργία εξαγωγής trace file με τα αποτελέσματα της εξομοίωσης και η δυνατότητα προκαθορισμού της κατανομής για τη δημιουργία γεγονότων (event triggering) στο δίκτυο. Αν και ο οπτικός εξομοιωτής δέχεται ορίσματα από τη γραμμή εντολών (command line) όπως ο command line εξομοιωτής, ωστόσο είναι πολύ πιο πρακτικό και φιλικό προς το χρήστη να γίνεται η χρήση του οπτικού περιβάλλοντος για να ρυθμίζονται οι διάφορες παράμετροι.



Σχήμα 5.2: Το αρχικό menu του animator

Σχήμα 5.3: Το menu ρυθμίσεων για την οπτική προσωμοίωση

Κατά τη λειτουργία των πρωτοκόλλων παρουσιάζονται οι κόμβοι και η διάταξη τους στο χώρο, ενώ το κέντρο ελέγχου ξεχωρίζει καθώς αναπαριστάται με διαφορετικό σχήμα από τους Smart Dust κόμβους. Η λειτουργία είναι και εδώ διακριτή ως προς το χρόνο, δηλαδή η οπτική εξομοίωση διεξάγεται σε γύρους. Σε κάθε γύρο φαίνονται εποπτικά με μια ματιά ποιοι κόμβοι είναι σε SLEEP mode και ποιοι σε AWAKE. Επιπρόσθετα φαίνονται με παραστατικό τρόπο οι εκπομπές μηνυμάτων από κάθε κόμβο (με μεθόδους broadcast, broadcast υπό γωνία, ή απευθείας μετάδοση), οι λήψεις τους, η ανίχνευση (sensing) γεγονότων και στην περίπτωση ιεραρχικών πρωτοκόλλων παρουσιάζονται τα clusters και το δένδρο ιεραρχίας για τη μετάδοση προς το Sink.

Καθ' όλη τη λειτουργία ενός πρωτοκόλλου, υπάρχει αγοικτό ένα “παράθυρο καταγραφής και επεξηγήσεων” με αυτόματη ολίσθηση (auto - scroll), όπου διατηρείται ένα υπόμνημα για το κάθε πρωτόκολλο σχετικά με τι αντιπροσωπεύουν τα χρησιμοποιούμενα σχήματα και οι χρωματισμοί στην οπτικοποίηση. Επίσης καταγράφονται τα σημαντικότερα από τα γεγονότα που διαδραματίζονται στο παράθυρο της οπτικής εξομοίωσης. Μερικά από αυτά είναι η δήλωση των κόμβων που έγιναν πηγές ανιχνεύοντας γεγονότα, η άφιξη μιας πληροφορίας στο κέντρο ελέγχου και η έναρξη περιόδων ανακατατάξεων στην περίπτωση ιεραρχικών πρωτοκόλλων.

Η δημιουργία γεγονότων (triggering) γίνεται με ένα απλό δεξί κλικ του ποντικιού πάνω σε ένα κόμβο. Κατόπιν στον αμέσως επόμενο γύρο ο επιλεγμένος ή οι επιλεγμένοι κόμβοι θα ανιχνεύσουν ένα συμβάν, το οποίο σε επόμενους γύρους θα μεταδώσουν προς το Sink.

5.5 Μελλοντική δουλειά και προκλήσεις

Ως μελλοντική εργασία πάνω στον εξομοιωτή simDust μπορούμε να προτείνουμε τα παρακάτω:

- 1) Ακόμα καλύτερη οργάνωση του κώδικα, με τυποποίηση ή και αυτοματοποίηση της διαδικασίας εισαγωγής - προσθήκης νέου πρωτοκόλλου στον κώδικα.
- 2) Ενσωμάτωση ενός κατάλληλου MAC μηχανισμού για ασύρματα δίκτυα που να λαμβάνει υπόψη του απώλειες μηνυμάτων και συγκρούσεις κατά τις μεταδόσεις.
- 3) Η δυνατότητα real-time λειτουργίας του πρωτοκόλλου αποτελεί μια πρόκληση για περαιτέρω ανάπτυξη στον εξομοιωτή, καθώς υπάρχουν νέες παράμετροι που πρέπει να ληφθούν υπόψη και περισσότερο πολύπλοκοι μηχανισμοί να ενσωματωθούν στον κώδικα. Ωστόσο καθώς η απομόνωση και η αφαιρεση στον κώδικα έχει γίνει αρκετά προσεγμένα, οι αλλαγές θα γίνουν σε λίγα κεντροποιημένα σημεία χωρίς να επηρεάσουν την οριθή λειτουργία του κώδικα.
- 4) Η τεκμηρίωση του κώδικα με κείμενο (documentation), που θα βοηθήσει στην εύκολη καταγόηση, συντηρισμότητα και παραπέρα ανάπτυξη του project simDust.
- 5) Η δημοσίευση του κώδικα στο διαδίκτυο, σε κοινότητες ανάπτυξης ανοιχτού λογισμικού (open source development), ώστε να ωφεληθεί ο εξομοιωτής από τη συνεισφορά άλλων προγραμματιστών, τις προτάσεις (suggestions) και την αναφορά προβλημάτων (bug reporting)

Κεφάλαιο 6

Εξομοίωση δύο πρωτοκόλλων

Χρησιμοποιώντας τον υλοποιημένο εξομοιωτή, εξομοίωσαμε δύο πρωτόκολλα διάδοσης πληροφορίας για δίκτυα Smart Dust. Αυτά ήταν το SW-PFR και το H-TEEN. Παρακάτω ακολουθούν κάποιες διευκρινήσεις ως προς τον τρόπο που υλοποιήθηκαν τα δύο πρωτόκολλα.

6.1 Λεπτομέρειες υλοποίησης του SW-PFR

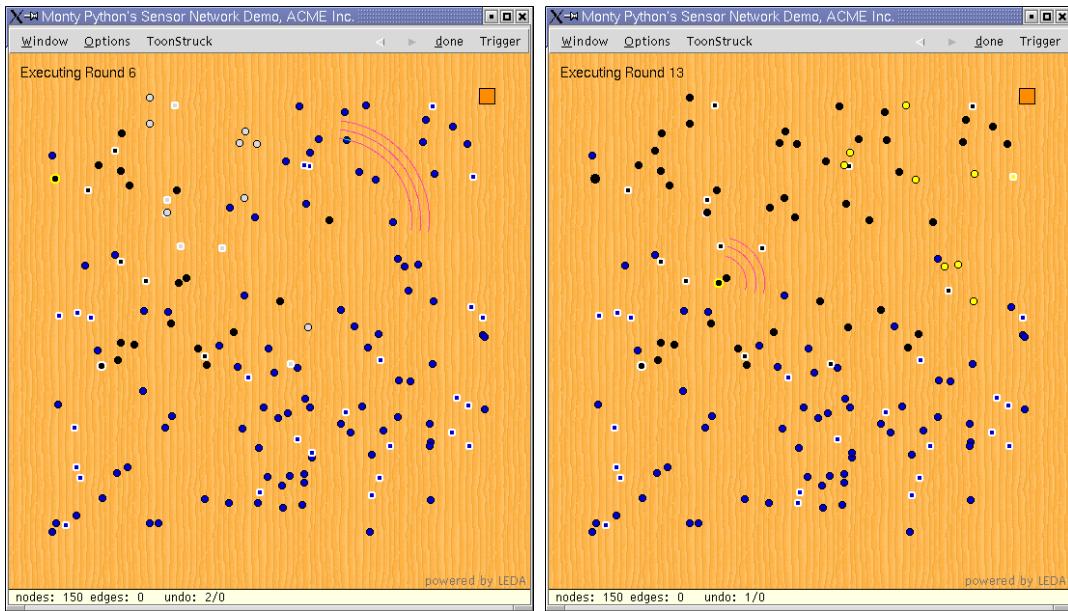
Η υλοποίηση του SW-PFR έγινε με τέτοιο τρόπο, ώστε να μην αλλοιωθεί η λογική του αρχικού PFR πρωτοκόλλου [6] από την ενσωμάτωση των νέων δυνατοτήτων για χειρισμό πολλαπλών γεγονότων (Multiple event triggers) και τον μηχανισμό SLEEP - AWAKE.

Το SW-PFR εκμεταλλεύεται τον εγγενή μηχανισμό SLEEP - AWAKE, που προσφέρει ο εξομοιωτής, στον οποίο ωστόσο γίνεται η εξής παρέμβαση: Όλα τα particles έχουν διαρκώς ανοιχτούς τους αισθητήρες τους, είτε είναι σε κατάσταση SLEEP είτε σε κατάσταση AWAKE και σε περίπτωση ανίχνευσης ενός γεγονότος ενόσω ένας κόμβος είναι σε κατάσταση SLEEP, τότε στον αμέσως επόμενο γύρο, ο κόμβος επανέρχεται αναγκαστικά (forced) σε κατάσταση AWAKE για να διαδώσει το μήνυμα όσο το δυνατό γρηγορότερα.

Κατά τα άλλα, η ανίχνευση ενός γεγονότος συνεπάγεται την εγγραφή του στη μνήμη του particle - πηγή. Οι εγγραφές στη μνήμη έχουν απεριόριστη διάρκεια ζωής και στη δομή τους περιλαμβάνουν ένα πεδίο με την πληροφορία του γεγονότος, ένα πεδίο με την χρονοσφραγίδα δημιουργίας του γεγονότος ή άφιξης του μηνύματος που περιγράφει το γεγονός και τέλος ένα πεδίο που υποδεικνύει αν ο κόμβος έχει ήδη προωθήσει επιτυχώς ή όχι την πληροφορία για το γεγονός.

Επιπρόσθετα τη στιγμή της ανίχνευσης ενός γεγονότος, ο κόμβος - πηγή υπολογίζει δυναμικά τον beta μετρητή του πρωτοκόλλου, που θα αντιστοιχισθεί στη διάδοση του συγκεκριμένου γεγονότος και θα καθορίσει τη διάρκεια της πρώτης φάσης του SW-PFR. Ο beta μετρητής τίθεται, ώστε να παίρνει την μέγιστη από τις τιμές 3 και το δεκαδικό λογάριθμο της απόστασης του κόμβου - πηγή από το Sink.

Κατόπιν στον αμέσως επόμενο γύρο από την εμφάνιση - ανίχνευση του γεγονότος, γίνεται η εκπομπή του μηνύματος με την πληροφορία για το συγκεκριμένο γεγονός. Η εκπομπή γίνεται με την τεχνική broadcast υπό γωνία α και ακτίνα R , όπου τα α και R είναι παράμετροι και



Σχήμα 6.1: Ενα στιγμούτυπο από την Σχήμα 6.2: Ενα δεύτερο στιγμούτυπο από την οπτικοποίηση του PFR

```

PFR PROTOCOL

RECEIVED      :: GREY
TRANSMITTED   :: BLACK
DID NOT TRANSMIT :: YELLOW
#####
NODE 117 TRIGGERED
AT ROUND 2
NODE 143 TRIGGERED
AT ROUND 2
EVENT FROM 117 (2.272,7.6117) CREATED AT 2
REACHED SINK AT 8.99638
EVENT FROM 143 (0.127,16.201) CREATED AT 2
REACHED SINK AT 8.9964

```

Σχήμα 6.3: Το παράθυρο καταγραφής για το PFR

μπορούν να καθοριστούν από τη γραμμή εντολών (command line). Όσοι κόμβοι λέβουν το μήνυμα όταν ξεκινήσουν τη διαδικασία χειρισμού λήψης ενός μηνύματος. Σύμφωνα με αυτή ένας κόμβος ελέγχει αν έχει ήδη κάποια καταχώρηση στη μνήμη του για το συγκεκριμένο γεγονός, το οποίο περιγράφεται στο λαμβανόμενο μήνυμα, και αν έχει τότε απορρίπτει το μήνυμα. Άλλιως, ενθέτει την νέα πληροφορία στη μνήμη του και αν έχει τότε απορρίπτει το μήνυμα. Αλλιώς, ενθέτει την νέα πληροφορία στη μνήμη του και αν ο beta counter του μηνύματος είναι θετικός, μειώνεται κατά ένα και το μήνυμα με τον ενημερωμένο beta counter διαδίδεται και πάλι γιατερμηνιστικά με broadcast υπό γωνία.

Αν ο beta counter είναι μηδέν (0), τότε ο κόμβος επιλέγει με τη χρήση μιας γεννήτριας τυχαίων αριθμών, ένα αριθμό x , όπου $x \in [0, 1]$. Έπειτα υπολογίζεται δυναμικά σε rad η

γωνία φ που σχηματίζουν η ευθεία που ενώνει την πηγή του συγκεκριμένου γεγονότος με το Sink και η ευθεία που ενώνει τον τρέχοντα κόμβο με την πηγή. Οι συντεταγμένες της πηγής αποσπώνται από την πληροφορία info του μηνύματος που λήφθηκε. Η γωνία αυτή (ϕ) διαιρείται με τη σταθερά π , και συγχρίνεται με τον τυχαίο αριθμό x , ώστε σύμφωνα με το πρωτόκολλο να αποφασιστεί αν ο κόμβος θα συμμετέχει στη διάδοση της πληροφορίας ή όχι. Αυτό σημαίνει ότι αν $x \leq \frac{\phi}{\pi}$ τότε ο κόμβος θα συμμετέχει, ενώ αν $x > \frac{\phi}{\pi}$, τότε ο κόμβος θα απέχει από την προώθηση της πληροφορίας.

Πρέπει να αναφερθεί ότι στην υλοποίηση του SW-PFR εξασφαλίζεται ότι προηγείται μια φάση (setup phase), όπου ο κάθε κόμβος αποκτά μια μοναδική ταυτότητα και επιπλέον μαθαίνει ποιοι από τους γειτονικούς του κόμβους είναι μέσα στην εμβέλεια του broadcast υπό γωνία (angle broadcast), ποιες είναι οι συντεταγμένες του στο πεδίο του δικτύου, ποιες είναι οι συντεταγμένες του Sink και του ανατίθεται μια μοναδική ταυτότητα.

Η εμβέλεια των μεταδόσεων από ένα κόμβο είναι σταθερή και συνήθως επιλέγεται πολύ μικρότερη από τη διάμετρο του δικτύου, αν και αυτό εν τέλει αφήνεται στην χρίση του χρήστη που εκτελεί μια εξομοίωση του πρωτοκόλλου.

Η αναλογία των περιόδων SLEEP - AWAKE καθορίζεται, όπως αναφέρθηκε, από τη γραμμή εντολών.

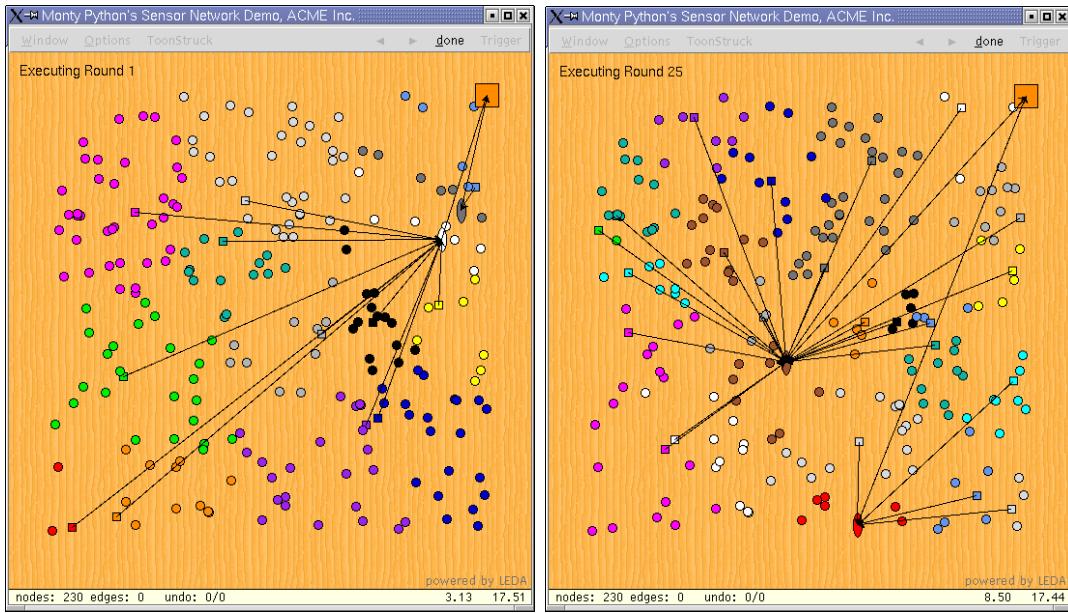
6.2 Λεπτομέρειες υλοποίησης του H-TEEN

Το πρωτόκολλο H-TEEN, υλοποιήθηκε με αρκετές παραδοχές και προσθήκες, καθώς το paper παρουσίασης του αυθεντικού πρωτοκόλλου TEEN [15] περιείχε μια αρκετά γενική παρουσίαση του πρωτοκόλλου, αφήνοντας κάποια λογικά κενά και ασάφειες.

Στην υλοποίηση του H-TEEN δεν γίνεται χρήση του μηχανισμού SLEEP-AWAKE του εξομοιωτή. Αντίθετα όλοι οι κόμβοι αρχικοποιούνται στην κατάσταση ALWAYS-ON και παραμένουν AWAKE καθ' όλη τη διάρκεια ζωής τους. Επιπρόσθετα οι κόμβοι δεν χρησιμοποιούν τη μνήμη τους για να αποθηκεύσουν πληροφορίες, καθώς το πρωτόκολλο προβλέπει απευθείας μεταδόσεις από σημείο προς σημείο και έτσι δεν υπάρχει η πιθανότητα εμφανίσης ατέρμονων βρόχων και επαναληπτικών μεταδόσεων. Αυτό που χρησιμοποιούν κυρίως οι cluster heads και οι super cluster heads, είναι μια εγγενής τεχνική buffering, η οποία υλοποιείται από τη FIFO ουρά αναμονής γεγονότων προς επεξεργασία, που όπως προαναφέρθηκε διαθέτει ο κάθε κόμβος. Έτσι όταν συγκεντρώνονται πολλά μηνύματα προς διάδοση στους κόμβους αυτούς, αυτοί ελέγχοντας ανά γύρο την ουρά αναμονής τους μπορούν να στέλνουν όσα μηνύματα εκκρεμούν προς διάδοση και κατόπιν να τα σβήνουν από την ουρά.

Το H-TEEN, όντας ιεραρχικό κατά βάση, υλοποιεί μέχρι και 4 επίπεδα ιεραρχίας εξασφαλίζοντας ένα διάστημα 24 γύρων πριν επαναληφθεί η set-up φάση που θα ανανεώσει την ιεραρχική δενδρική δομή. Στο διάστημα αυτών των 24 γύρων εφαρμόζεται η κυρίως φάση (steady phase) του TEEN σύμφωνα με την οποία ένας κόμβος που ανιχνεύει ένα συμβάν το προωθεί μόνο στον cluster head. Έπειτα ο cluster head μεταδίδει όσα μηνύματα έχει “ακούσει” στον τυχόν super cluster head του αμέσως ανώτερου επίπεδου ιεραρχίας και ούτω καθ' εξής μέχρι να φτάσουν τα μηνύματα στη ρίζα της ιεραρχίας που είναι το κέντρο ελέγχου Sink.

Λόγω της φύσης του πρωτοκόλλου και της ανάγκης να μπορεί ο κάθε κόμβος να μεταδίδει



Σχήμα 6.4: Ενα στιγμούτυπο από την Σχήμα 6.5: Ενα δεύτερο στιγμούτυπο από την οπτικοποίηση του TEEN

```
X-W IT'S ALIVE
TEEN PROTOCOL
EACH CLUSTER HAS DISTINCT COLOR
SIMPLE CLUSTER HEAD :: SQUARE NODE
UPPER LEVEL CLUSTER HEAD :: ELLIPSE NODE
#####
NODE 18 DECIDED TO BECOME A CLUSTER-HEAD
NODE 157 DECIDED TO BECOME A CLUSTER-HEAD
NODE 89 DECIDED TO BECOME A CLUSTER-HEAD
NODE 186 DECIDED TO BECOME A CLUSTER-HEAD
NODE 220 DECIDED TO BECOME A CLUSTER-HEAD
NODE 223 DECIDED TO BECOME A CLUSTER-HEAD
NODE 105 DECIDED TO BECOME A CLUSTER-HEAD
NODE 74 DECIDED TO BECOME A CLUSTER-HEAD
NODE 171 DECIDED TO BECOME A CLUSTER-HEAD
NODE 103 DECIDED TO BECOME A CLUSTER-HEAD
NODE 128 DECIDED TO BECOME A CLUSTER-HEAD
NODE 129 DECIDED TO BECOME A CLUSTER-HEAD
NODE 220 DECIDED TO BECOME A CLUSTER-HEAD
NODE 69 DECIDED TO BECOME A CLUSTER-HEAD
NOW SHOWING THE TREE OF TRANSMISSIONS
FROM CLUSTER-HEADS TO THE SINK
```

Σχήμα 6.6: Το παράθυρο καταγραφής για το TEEN

μέχρι το κέντρο ελέγχου σε περίπτωση που εκλεγεί ως cluster head, η μέγιστη εμβέλεια μετάδοσης για τον κάθε κόμβο είναι ίση με τη διάμετρο του δικτύου.

Όταν εκτελείται ένας γύρος set-up συμβαίνουν τα ακόλουθα. Ο κάθε κόμβος αποφασίζει αν πρέπει να γίνει ή όχι cluster head, επιλέγοντας ένα τυχαίο αριθμό στο $[0,1]$ και συγχρίνοντας τον με το κατώφλι $T(n)$ του πρωτοχόλου. Αν είναι μικρότερος του $T(n)$ τότε ο κόμβος γίνεται cluster head. Υπενθυμίζεται ότι το κατώφλι αυτό δίνεται από τη σχέση:

$$T(n) = \begin{cases} \frac{P}{1-P(r \bmod \frac{1}{P})} & \text{αν ο κόμβος } n \text{ δεν έγινε cluster heads στους} \\ & \text{τελευταίους } \frac{1}{P} \text{ εποχικούς γύρους} \\ 0 & \text{αλλιώς} \end{cases}$$

όπου r είναι ο τρέχον γύρος και P η πιθανότητα να γίνει cluster head ένας κόμβος σε ένα εποχικό γύρο. Η πιθανότητα P αντιστραμμένη ($\frac{1}{P}$) καθορίζει και τη διάρκεια μιας εποχής TEEN. Στον εξομοιωτή έχουμε ορίσει σταθερή $P = 0.05$ και συνεπώς TEEN Epoch = 20 εποχικοί γύροι. Πρέπει να διευχρινίσουμε εδώ ότι ένας εποχικός γύρος είναι γνήσιο υπερσύνολο των γύρων στους οποίους εξελίσσεται η λειτουργία του εξομοιωτή. Μάλιστα ένας εποχικός γύρος ορίζεται όπως αναφέραμε ώστε να αποτελείται από 24 γύρους, από τους οποίους ο πρώτος περικλείει και τη φάση set-up του πρωτοκόλλου. Ο έλεγχος για τη λήξη μιας εποχής γίνεται στην αρχή κάθε εποχικού γύρου. Η λήξη μιας εποχής TEEN σηματοδοτεί την αρχικοποίηση σε τιμή 1 της μεταβλητής r της εξίσωσης.

Ο κόμβος που αποφασίζει να γίνει cluster head, εκπέμπει - διαφημίζει την απόφαση του αυτή σε όλο το δίκτυο επιλέγοντας για ακτίνα εκπομπής τη διάμετρο του δικτύου. Το μήκος αυτού του μηνύματος - διαφήμιση επιλέγεται να είναι σταθερό και ίσο με 128 bits. Ο κάθε κόμβος διατηρεί μια ουρά, όπου καταχωρεί όλους τους cluster heads που γνωρίζει, διατεταγμένους με αύξουσα σειρά μικρότερης απόστασης τους από τον εαυτό του. Έτσι οι απλοί κόμβοι (μη cluster heads) επιλέγουν τελικά να ανήκουν στον cluster head που είναι ο πιο κοντινός σε αυτούς. Την επιλογή τους αυτή τη δηλώνουν στον κατάλληλο cluster head, αποστέλλοντας ένα μήνυμα των 40 bits.

Ένας cluster head, μεταδίδει επίσης και ένα TDMA πρόγραμμα για το συντονισμό των εκπομπών από τους κόμβους που ανήκουν στο cluster που διευθύνει. Στην υλοποίηση μας αυτό το μήνυμα εξυπηρετεί μόνο το σκοπό ενεργειακής χρέωσης του cluster head, ενώ οι μεταδόσεις των κόμβων δεν ακολουθούν κάποιο TDMA πρόγραμμα, όντας διευθετημένες, από εγγενείς μηχανισμούς του εξομοιωτή ώστε να αποφεύγονται οι μεταξύ τους συγκρούσεις. Τα bits που αντιστοιχίζονται στο μήνυμα με το TDMA πρόγραμμα είναι 32.

Μετά από τις παραπάνω ενέργειες έχει αποκατασταθεί το πρώτο επίπεδο ιεραρχίας. Ακολουθούν δύο παρόμοιες φάσεις ώστε να αποκατασταθούν το πολύ δύο ακόμα ανώτερα επίπεδα ιεραρχίας. Έτσι το ένα επίπεδο εκλέγει super cluster heads από τους υπάρχοντες cluster heads του αμέσως κατώτερου επιπέδου. Το επόμενο επίπεδο εκλέγει hyper cluster heads από τους υπάρχοντες super cluster heads του αμέσως κατώτερου επιπέδου. Η πιθανότητα P που συμμετέχει στο τύπο του κατωφλιού $T(n)$ έχει την ίδια τιμή (0.05) και για τη διαμόρφωση των δύο αυτών επιπέδων ιεραρχίας. Επίσης πρέπει να σημειωθεί ότι μόνο στην περίπτωση που οι cluster heads του αμέσως κατώτερου επιπέδου είναι παραπάνω από έναν ικανό αριθμό (πχ 8) προχωρούμε στην αποκατάσταση νέου ιεραρχικού επιπέδου. Τέλος εξυπακούεται ότι μόνο οι cluster heads του ανώτερου επιπέδου μπορούν να επικοινωνούν απευθείας με το Sink.

Κεφάλαιο 7

Σύγκριση μεταξύ H-TEEN και SW-PFR

Η υλοποίηση των δύο πρωτοκόλλων H-TEEN και SW-PFR δεν έγινε μόνο για σκοπούς επισκόπησης της λειτουργίας τους σε ασύρματα δίκτυα έξυπνης σκόνης. Εξυπηρέτησε έπισης και το βασικό στόχο της συγκριτικής θεώρησης τους στα πλαίσια αυτής της διπλωματικής. Μάλιστα η συγκριτική αυτή θεώρηση αποτέλεσε και το θέμα ξεχωριστής εργασίας [8], η οποία συντάχθηκε από τους Σ. Νικολετσέα, Ι. Χατζηγιαννάκη, Χ. Ευθυμίου, Α. Κινάλη, Γ. Μυλωνά και τον υπογράφοντα αυτής της διπλωματικής Α. Αντωνίου.

7.1 Μετρικές αξιολόγησης

Με τη βοήθεια του εξομοιωτή, έγινε δυνατή η σύγκριση των δύο πρωτοκόλλων SW-PFR και H-TEEN κάτω από παρόμοιες συνθήκες ανάπτυξης του δικτύου και κατανομής των γεγονότων ανίχνευσης. Η σύγκριση των πρωτοκόλλων έγινε με βάση συγκεκριμένα κριτήρια και μετρικές που παρουσιάζονται παρακάτω.

Κατ' αρχήν αναφέρουμε ότι στο μοντέλο δικτύου που χρησιμοποιήθηκε, το Sink είναι ένα ακόμα σημείο στο πεδίο του δικτύου τοποθετημένο στο ακρότατο σημείο (X, Y) όπου X και Y είναι οι διαστάσεις του πεδίου. Επίσης η κατανομή των κόμβων στο πεδίο ήταν τυχαία και ομοιόμορφη. Υποθέτουμε ότι K είναι ο συνολικός αριθμός των γεγονότων που πρέπει να ανιχνευτούν και να διαδοθούν προς το κέντρο ελέγχου. Επίσης ορίζουμε ως k τον αριθμό των γεγονότων για τα οποία έμαθε τελικά το κέντρο ελέγχου μετά την εκτέλεση του πειράματος. Αυτό σημαίνει ότι γενικά είναι $k \leq K$.

Το ποσοστό επιτυχίας P_s ενός πρωτοκόλλου δίνεται από το πηλίκο του αριθμού των γεγονότων που μεταδόθηκαν επιτυχώς προς το κέντρο ελέγχου προς το συνολικό αριθμό γεγονότων που ανιχνεύτηκαν. Αυτό το ποσοστό παρέχει και ένα μέτρο αξιολόγησης της ποιότητας των υπηρεσιών (QoS) που προσφέρουν τα δύο πρωτόκολλα. Είναι προφανές ότι θα ισχύει:

$$P_s = \frac{k}{K}$$

Μια άλλη χρήσιμη μετρική αξιολόγησης των πρωτοκόλλων είναι η μέση διαθέσιμη ενέργεια

E_{avg} ανά κόμβο του δικτύου. Έχει ενδιαφέρον να παρατηρούμε τις μεταβολές στις τιμές αυτής της μετρικής με την πάροδο του χρόνου εξομοίωσης, και να εξάγουμε χρήσιμα συμπεράσματα για την ενεργειακή κατανάλωση των πρωτοκόλλων και τη διάρκεια ζωής των δικτύων. Η μέση διαθέσιμη ενέργεια E_{avg} ανά κόμβο ορίζεται ως

$$E_{avg} = \sum_{i=1}^n \frac{E_i}{n}$$

Όπου E_i είναι η διαθέσιμη ενέργεια ανά κόμβο σε δεδομένη χρονική στιγμή και n το πλήθος των κόμβων στο δίκτυο. Γενικά από ένα αποδοτικό πρωτόκολλο προσδοκούμε μικρή κατανάλωση ενέργειας συναρτήσει του χρόνου. Πρέπει να αναφέρουμε επίσης ότι η σύγκριση των πρωτοκόλλων βάσει αυτής της μετρικής, έγινε διατηρώντας σε παρόμοια ποσοστά τις υπόλοιπες παραμέτρους αξιολόγησης. Αυτό σημαίνει ότι για παράδειγμα απαιτήσαμε η σύγκριση των δύο πρωτοκόλλων όσον αφορά την ενεργειακή κατανάλωση να γίνει δεδομένου ότι και τα δύο ικανοποιούν παρόμοια κριτήρια ποιότητας υπηρεσιών (QoS). Άλλιώς η σύγκριση θα ήταν άνιση και δε θα επέτρεπε να εξαχθούν παραγωγικά συμπεράσματα.

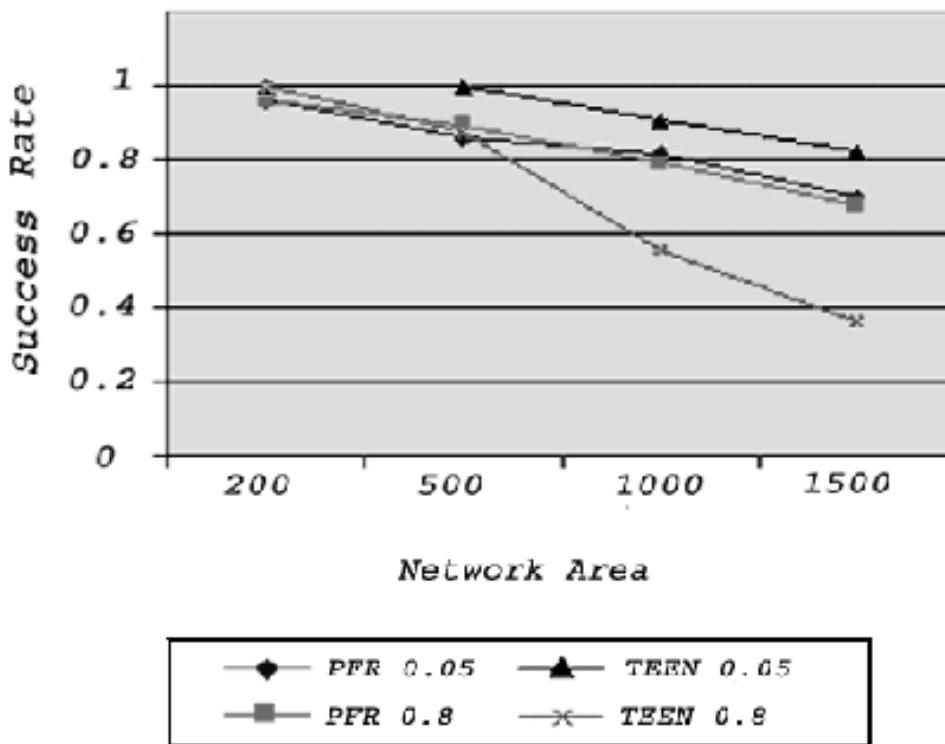
Ο αριθμός των κόμβων h_a που παραμένουν “ενεργοί”, ήτοι δεν έχουν εξαντληθεί τα ενέργειακά αποθέματα τους με την πάροδο του χρόνου εξομοίωσης, αποτελεί ένα τρίτο κριτήριο αξιολόγησης. Όσο περνάει ο χρόνος εξομοίωσης, ένα πρωτόκολλο αποδεικνύεται αποτελεσματικό αν διατηρεί σε υψηλά επίπεδα τον αριθμό των ενεργών κόμβων, οπότε επιμηκύνεται ο χρόνος ζωής του δικτύου. Βέβαια πρέπει κανείς να λάβει επιπρόσθετα υπόψη του τον τρόπο με τον οποίο οι κόμβοι καθίστανται ανενεργοί με την πάροδο του χρόνου και συγκεκριμένα την κατανομή της διαδικασίας “νέκρωσης” του δικτύου σε σχέση με την επιφάνεια του πεδίου που αυτό καλύπτει.

Τέλος πρέπει να αναφέρουμε την παράμετρο I_s που συμβολίζει το ρυθμό δημιουργίας γεγονότων στο δίκτυο (injection rate). Αυτή η παράμετρος έχει καθοριστικό ρόλο στη συγκριτική θεώρηση των δύο πρωτοκόλλων, καθώς μας επιτρέπει να συμπεράνουμε την καταλληλότητα των πρωτοκόλλων τόσο για εφαρμογές με μικρό ρυθμό εμφάνισης κρίσιμων γεγονότων όσο και για εφαρμογές με κατακλυσμό από κρίσιμα γεγονότα.

7.2 Πειραματικές μετρήσεις και συμπεράσματα

Για τη σύγκριση των δύο πρωτοκόλλων, διεξήχθησαν αρκετά πειράματα. Είχαμε την ευκαιρία να παρατηρήσουμε τη συμπεριφορά των πρωτοκόλλων σε μια ποικιλία από πεδία δικτύων όσον αφορά τις διαστάσεις τους και τον αριθμό των σωματιδίων - κόμβων που “διασκορπίστηκαν” σε αυτά. Οι τυπικές διαστάσεις για τα πεδία επιλέχθηκαν να κυμαίνονται από $200 m \times 200 m$ μέχρι και $1500 m \times 1500 m$. Ας σημειωθεί ότι αυτές οι διαστάσεις είναι σημαντικά μεγαλύτερες από τις διαστάσεις πεδίων που χρησιμοποιήθηκαν σε παρόμοιες έρευνες από άλλους ερευνητές και έτσι μας επιτρέπουν να εξάγουμε πολύ χρήσιμα συμπεράσματα για τη συμπεριφορά των πρωτοκόλλων και την προσαρμοστικότητα τους σε μικρής και μεγάλης έκτασης δίκτυα. Ο αριθμός των σωματιδίων που “διασκορπίζουμε” στα πεδία αυτά κυμαίνεται από 500 έως 3000 σωματίδια που κατανέμονται τυχαία και ομοιόμορφα στα πεδία. Γενικά διασκορπίζουμε περισσότερα σωματίδια στα δίκτυα μεγάλων διαστάσεων και λιγότερα στα δίκτυα μικρών διαστάσεων, ώστε προσεγγιστικά να διατηρείται σταθερή η πυκνότητα των δικτύων.

Αρχικά παρουσιάζουμε τα αποτελέσματα σύγκρισης του SW-PFR με το H-TEEN, βάσει των ποσοστών επιτυχίας τους. Επιλέξαμε να αναπαραστήσουμε τα ποσοστά επιτυχίας των πρωτοκόλλων συναρτήσει του μεγέθους του δικτυακού πεδίου για δύο διαφορετικές τιμές της παραμέτρου I_s (0.05 και 0.8). Επιλέγοντας ακραίες τιμές για την I_s παράμετρο είμαστε σε θέση να παρατηρήσουμε πως ανταποκρίνονται τα πρωτόκολλα σε ακραίες συνθήκες. Η εικόνα 7.1 δείχνει τις τιμές του ποσοστού επιτυχίας για τα πρωτόκολλα καθώς το μέγεθος του πεδίου του δικτύου μεγαλώνει. Όπως είναι εμφανές, για μικρές σχετικά διαστάσεις πεδίων ($200 m \times 200 m$ έως $500 m \times 500 m$), και τα δύο πρωτόκολλα πετυχαίνουν καλές επιδόσεις για το ποσοστό επιτυχίας (> 0.85). Ωστόσο οι επιδόσεις πέφτουν καθώς το μέγεθος του πεδίου αυξάνει. Το SW-PFR παρουσιάζει ποσοστά επιτυχίας που παραμένουν αρκετά υψηλά ακόμα και για πολύ μεγάλα δίκτυα όπως $1500 m \times 1500 m$ και φαίνεται ότι αποδίδει αρκετά καλά και για τις δύο ακραίες τιμές του ρυθμού εμφάνισης γεγονότων I_s . Αντίθετα το H-TEEN παρουσιάζει σημαντική πτώση στις επιδόσεις του στην περίπτωση του υψηλού I_s , καθώς ο ρυθμός επιτυχίας πέφτει κάτω από 0.3 για μεγάλα δίκτυα της τάξης του $1500 m \times 1500 m$.



Σχήμα 7.1: Ρυθμός επιτυχίας συναρτήσει των διαστάσεων του δικτύου, για ρυθμός εμφάνισης γεγονότων 0.05 και 0.08

Στην εικόνα 7.2 παρουσιάζουμε το ποσοστό επιτυχίας των δύο πρωτοκόλλων για δύο τιμές μεγέθους δικτυακού πεδίου, ήτοι $500 m \times 500 m$ και $1500 m \times 1500 m$, και συναρτήσει της μεταβλητής I_s . Στην περίπτωση του $500 m \times 500 m$ πεδίου, και τα δύο πρωτόκολλα συμπεριφέρονται σχεδόν άριστα πετυχαίνοντας ποσοστά επιτυχίας κοντά στη μονάδα. Στην περίπτωση όμως του $1500 m \times 1500 m$, ο ρυθμός I_s επηρεάζει αρκετά το ποσοστό

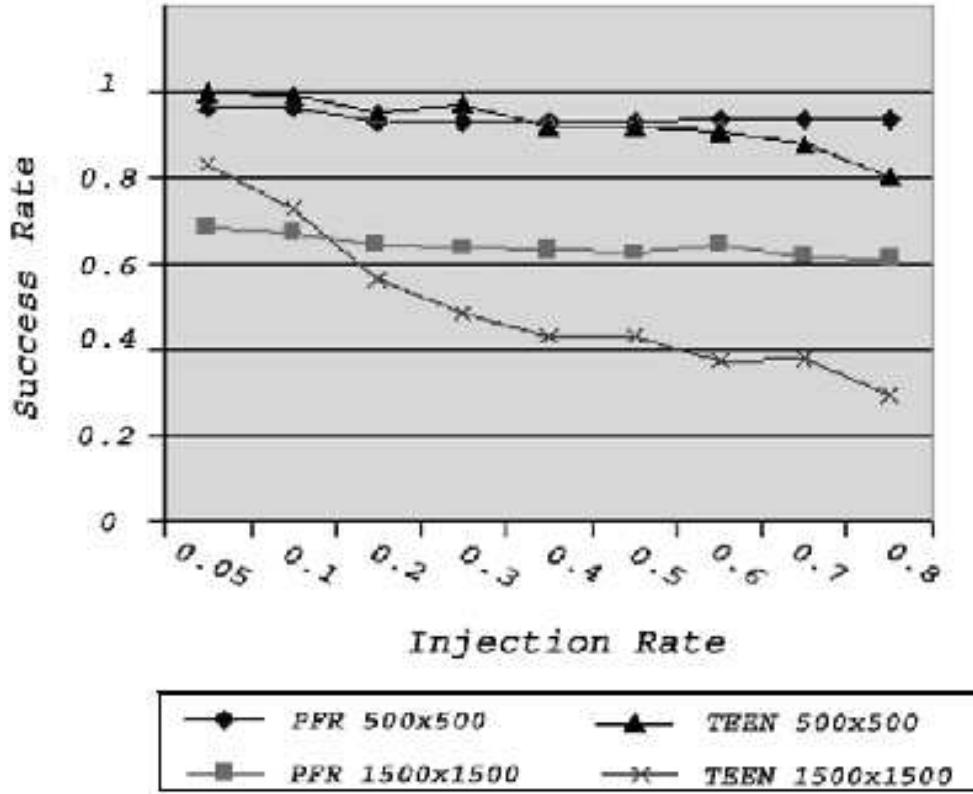
επιτυχίας των δικτύων. Ειδικά για το H-TEEN, ο ρυθμός επιτυχίας πέφτει από 85% στο 30%, ενώ στο SW-PFR ο ρυθμός επιτυχίας ζεκινάει από το 70% αλλά παρουσιάζει σαφώς πιο συγκρατημένη πτώση καθώς το I_s αυξάνει.

Η έντονη εξάρτηση των ποσοστών επιτυχίας του H-TEEN από το ρυθμό δημιουργίας κρίσιμων γεγονότων στο δίκτυο, αποδίδεται στο διαχωρισμό του δικτύου σε clusters από το πρωτόκολλο. Καθώς αυξάνει ο ρυθμός I_s , ένας cluster head επιφορτίζεται με την ευθύνη να μεταδίδει όλο και περισσότερα μηνύματα και έτσι καταναλώνει σημαντικά ποσά ενέργειας κατά τη διάρκεια της αρχηγίας του. Μάλιστα αν το I_s είναι αρκετά μεγάλο ενδέχεται ένας cluster head να κατασταθεί ανενεργός πριν τελειώσει η αρχηγία του. Τότε ουσιαστικά όλο το αντίστοιχο cluster καθίσταται τουλάχιστο προσωρινά ανενεργό, καθώς όλες οι μεταδόσεις για νέα γεγονότα δε θα μπορούν να διαδοθούν προς το Sink. Σε μεγάλα δίκτυα της τάξης των $1500 \text{ m} \times 1500 \text{ m}$, το φαινόμενο γίνεται πιο έντονο καθώς οι cluster heads είναι αναγκασμένοι να πραγματοποιούν μεταδόσεις σε μεγάλες αποστάσεις διαδίδοντας ένα γεγονός. Καθώς η δαπάνη ενέργειας για μια μετάδοση είναι ανάλογη προς το τετράγωνο της απόστασης, οι ενέργειακή κατανάλωση που επιβαρύνει τους cluster heads είναι σημαντικά μεγάλη.

Το SW-PFR, αν και δεν επηρεάζεται τόσο από τις μεγάλες τιμές του ρυθμού εμφάνισης γεγονότων I_s , παρουσιάζει σημαντικά μειωμένες επιδόσεις με την αύξηση των διαστάσεων του δικτύου. Αυτό μπορεί να αποδοθεί στη φύση του πρωτοκόλλου που υπαγορεύει multihop διάδοση για κάθε κάρτη πληροφορία. Σε μεγάλα δίκτυα, είναι προφανές ότι απαιτούνται περισσότερα βήματα (hops) για να φτάσει ένα γεγονός στο Sink, και συνεπώς μεγαλύτερη συνολική κατανάλωση ενέργειας και γρηγορότερη εξάντληση των κόμβων που συμμετέχουν στην προώθηση πληροφορίας.

Ως επόμενο αποτέλεσμα της συγκριτικής θεώρησης των δύο πρωτοκόλλων, παρουσιάζουμε τις εικόνες 7.3 και 7.4 που δείχνουν τη μέση τιμή για τα ενέργειακά αποθέματα ανά κόμβο, συναρτήσει του χρόνου εξομοίωσης. Αρχικά όπως έχει αναφερθεί κάθε σωματίδιο διαθέτει 1 Joule ενέργειακό απόθεμα. Στη εικόνα 7.3 φαίνεται η διαθέσιμη ενέργεια ανά κόμβο για δίκτυο διαστάσεων $500 \text{ m} \times 500 \text{ m}$, και η εικόνα 7.4 αναφέρεται σε δίκτυο διαστάσεων $1000 \text{ m} \times 1000 \text{ m}$. Ο ρυθμός εμφάνισης γεγονότων και στα δύο πειράματα επιλέχθηκε να είναι ίσος με 0.05, ενώ εξασφαλίσαμε ότι παράλληλα ο ρυθμός επιτυχίας των πρωτοκόλλων δεν έπεφτε κάτω από το 80%. Για το μικρό δίκτυο το H-TEEN φαίνεται να καταναλώνει με πιο αργούς ρυθμούς τη διαθέσιμη ενέργεια των κόμβων, από ότι το SW-PFR. Από την άλλη μεριά όμως στο μεγάλο δίκτυο το SW-PFR έχει καλύτερη συμπεριφορά όσο αφορά την ενέργειακή κατανάλωση.

Η εξήγηση για τις παραπάνω επιδόσεις των δύο πρωτοκόλλων πρέπει να αναζητηθεί στους μηχανισμούς διάδοσης που διαθέτουν αυτά. Στο H-TEEN, ο αριθμός των βημάτων hops για τη διάδοση πληροφορίας προς το κέντρο ελέγχου είναι το πολύ 4, καθώς το πολύ τέσσερα επίπεδα ιεραρχίας μπορούν να αποκατασταθούν και ο κάθε κόμβος μεταδίδει σε κόμβο (cluster head) που βρίσκεται στο αμέσως επόμενο επίπεδο ιεραρχίας. Αντίθετα στο SW-PFR οι μεταδόσεις έχουν μικρή εμβέλεια και ο αριθμός των βημάτων διάδοσης ενός μηνύματος προς το Sink είναι σχετικά πολύ μεγαλύτερος. Έτσι για δίκτυα μικρών διαστάσεων, για τη διάδοση ενός γεγονότος οι μεταδόσεις των cluster heads κοστίζουν συνολικά αρκετά λιγότερο από τις πολλαπλές μεταδόσεις multihop του SW-PFR. Σε δίκτυα μεγάλων διαστάσεων ωστόσο, για το H-TEEN, οι cluster heads χρειάζεται να αυξήσουν την καταναλισκόμενη ενέργεια για τη διάδοση μηνυμάτων σε μακρύτερες αποστάσεις. Η ενέργεια αυτή μάλιστα

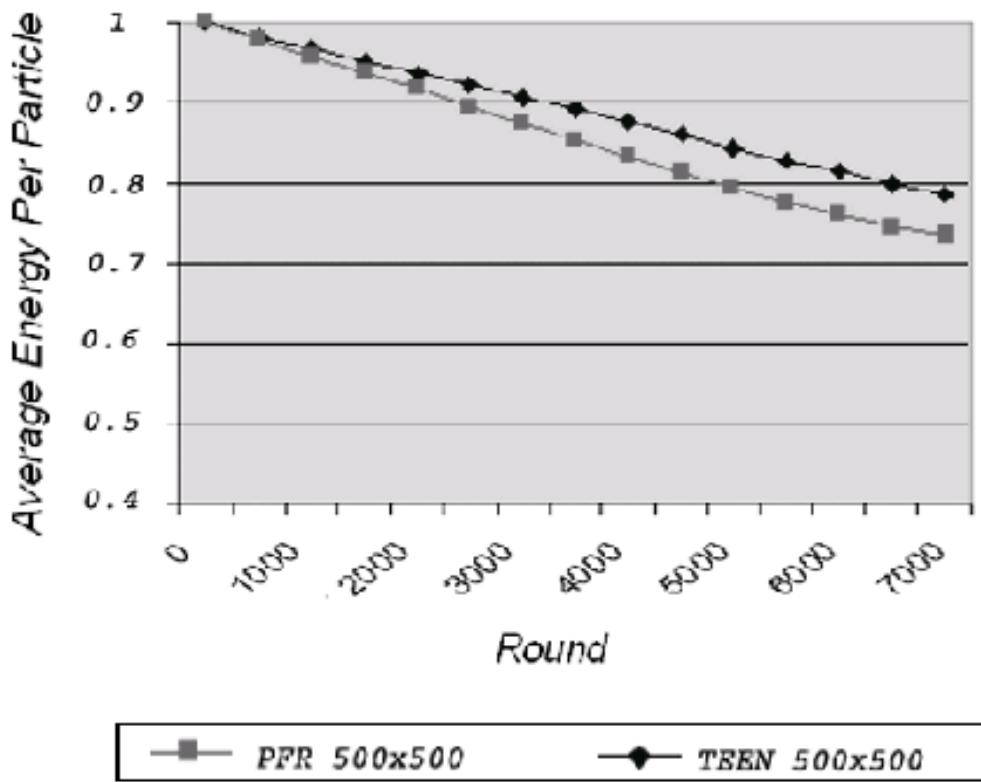


Σχήμα 7.2: Ρυθμός επιτυχίας συναρτήσει του ρυθμού εμφάνισης γεγονότων, για δίκτυα διαστάσεων 500×500 και 1500×1500

αυξάνεται ανάλογα προς το τετράγωνο της απόστασης από τον cluster head του επόμενου επιπέδου ή το Sink. Από την άλλη μεριά στο SW-PFR ο αριθμός των multihop μεταδόσεων για ένα μήνυμα αυξάνεται γραμμικά με το μέγευθος του πεδίου του δικτύου. Συνεπώς το SW-PFR αναδεικνύεται ως πιο κατάλληλο από το H-TEEN πρωτόκολλο για δίκτυα μεγάλων διαστάσεων.

Για τη συγκριτική παρουσίαση των δύο πρωτοκόλλων εκτελέσαμε και κάποια πειράματα για να παρατηρήσουμε τη μεταβολή του αριθμού των ενεργών σωματίδων με τη πάροδο του χρόνου εξομοίωσης. Στις εικόνες 7.5 και 7.6 φαίνεται ο αριθμός των ενεργών κόμβων συναρτήσει του χρόνου για δίκτυα διαστάσεων $500 m \times 500 m$ και $1000 m \times 1000 m$ αντίστοιχα και για κοινό ρυθμό $I_s = 0.05$.

Για το δίκτυο διαστάσεων $500 m \times 500 m$, φαίνεται ότι το H-TEEN έχει ελάχιστες απώλειες ενεργών κόμβων. Αυτό δείχνει ότι το H-TEEN πετυχαίνει το βασικό στόχο της ομοιόμορφης κατανομής της ενεργειακής κατανάλωσης σε όλους τους κόμβους του δικτύου. Αντίθετα το SW-PFR παρουσιάζει απότομα μεγάλες απώλειες σε κόμβους δικτύου. Αυτό αναδεικνύει ένα αδύνατο σημείο του SW-PFR, το οποίο είναι η γρήγορη εξάντληση των κόμβων που είναι γύρω από το Sink καθώς με μεγάλη πιθανότητα όλα τα γεγονότα προς διάδοση θα περάσουν από αυτούς πριν φτάσουν στο Sink. Έτσι λοιπόν το SW-PFR παρουσιάζει συμπεριφορά κατωφλιού καθώς υπάρχει κάποιο χρονικό σημείο πέρα από το οποίο οι συγκεκριμένοι κόμβοι



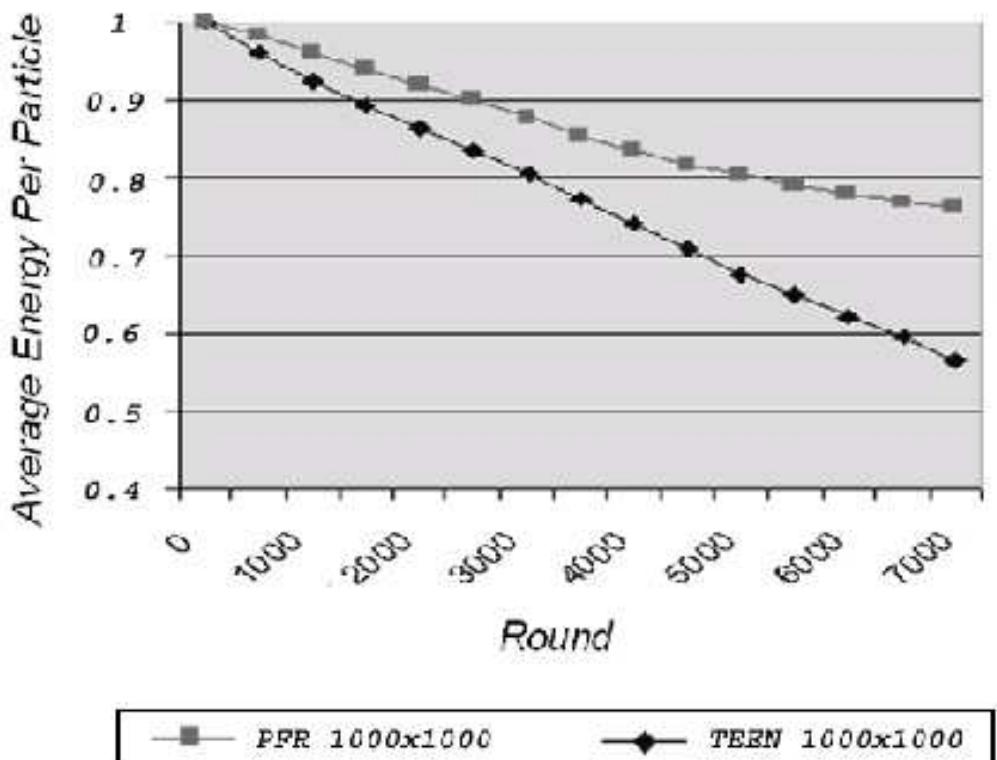
Σχήμα 7.3: Μέση κατανάλωση ενέργειας ανα κόμβο για $I_s = 0.05$ για δίκτυα διαστάσεων 500×500

αρχίζουν να απενεργοποιούνται με γοργούς ρυθμούς.

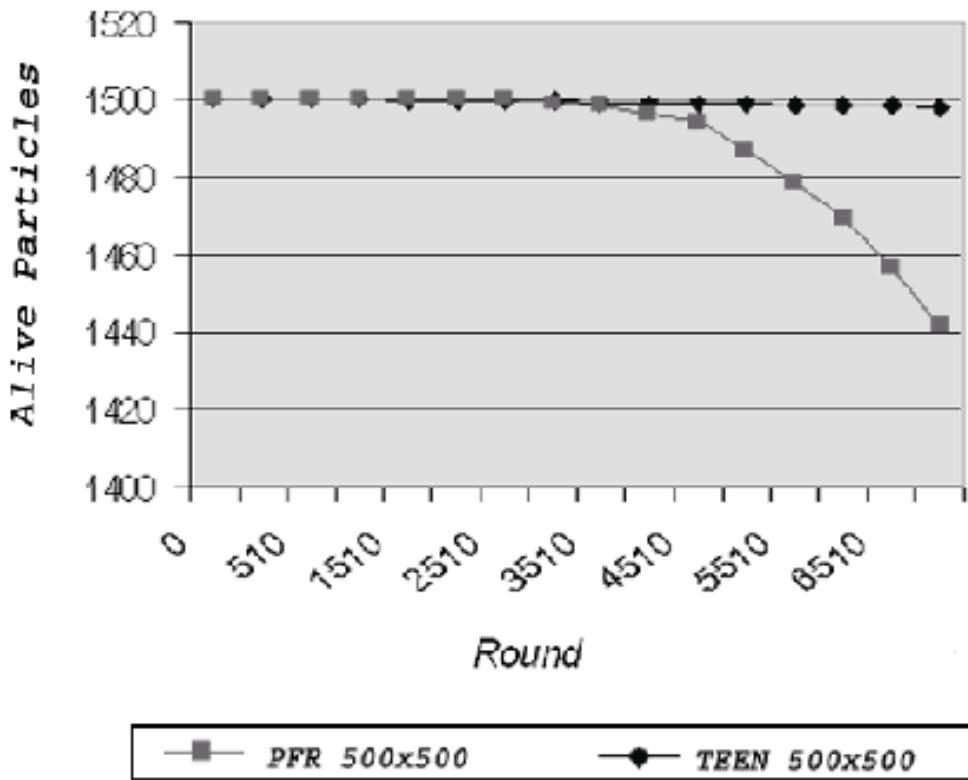
Για το δίκτυο $1500 \text{ m} \times 1500 \text{ m}$, τα αποτελέσματα ήταν διαφορετικά. Συγκεκριμένα στο H-TEEN πρωτόκολλο οι απώλειες των ενεργών κόμβων ξεκίνησαν νωρίτερα από τις αντίστοιχες του SW-PFR και επιπρόσθετα ο ρυθμός τους ήταν μεγαλύτερος. Και πάλι αυτή η συμπεριφορά του H-TEEN σε δίκτυα μεγάλων διαστάσεων αποδίδεται στην υποχρέωση κάποιων κόμβων να εκτελούν μεταδόσεις σε μεγάλες αποστάσεις, εξαντλώντας με γοργούς ρυθμούς τα αποθέματα τους σε ενέργεια. Για το SW-PFR παρατηρούμε και πάλι μια “κατωφλιακή” συμπεριφορά η οποία έχει την εξήγηση της στη γρήγορη εξάντληση των κόμβων που είναι γειτονικοί του Sink.

Ειδικά για το SW-PFR πρωτόκολλο μας δόθηκε η ευκαιρία να μεταβάλλουμε κάποιες από τις παραμέτρους που καθορίζουν τη συμπεριφορά του πρωτοκόλλου και να παρατηρήσουμε τις επιδόσεις του για δίκτυα σταθερής πυκνότητας. Κάποιες από τις παραμέτρους αυτές ήταν η εμβέλεια των μεταδόσεων και η αναλογία των περιόδων SLEEP και AWAKE. Σκοπός ήταν να καθορίσουμε ένα σύνολο τιμών για τις παραμέτρους για τις οποίες το SW-PFR θα παρουσίαζε τις καλύτερες επιδόσεις. Το SW-PFR φαίνεται να επηρεάζεται αρκετά από την επιλογή των τιμών στις παραμέτρους του.

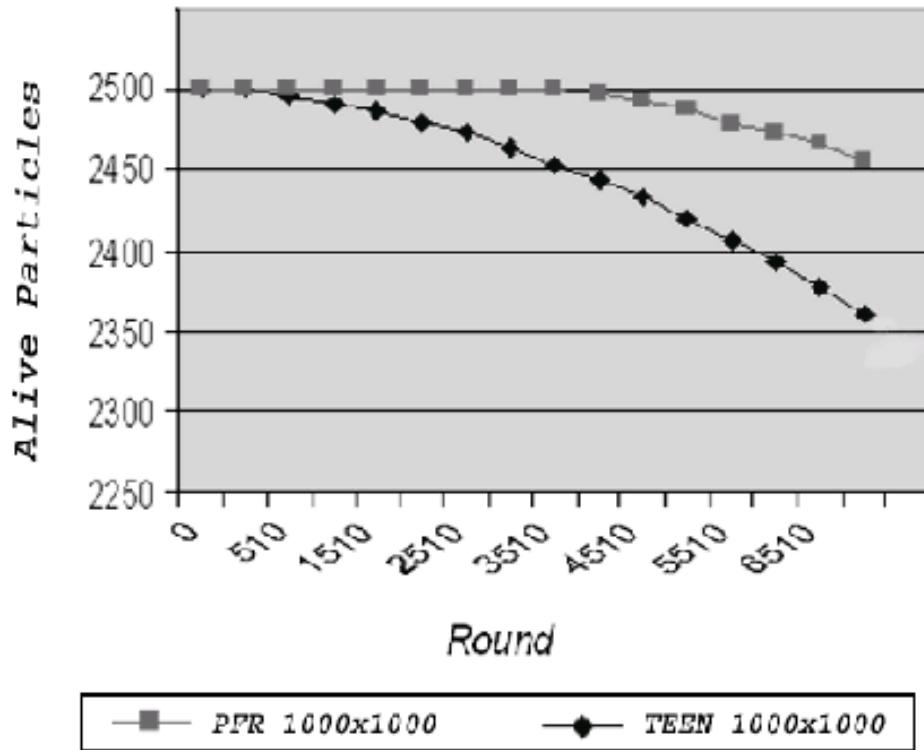
Στην εικόνα 7.7 φαίνεται ο ρυθμός επιτυχίας του SW-PFR για διάφορες τιμές της εμβέλειας μεταδόσεων. Μπορεί κανείς να διαπιστώσει ότι υπάρχει μια περιορισμένη περιοχή τιμών για την ακτίνα μετάδοσης (μεταξύ 35 m και 60 m), όπου οι αντίστοιχες τιμές του ρυθμού



Σχήμα 7.4: Μέση κατανάλωση ενέργειας ανα κόμβο για $I_s = 0.05$ για δίκτυα διαστάσεων 1000×1000



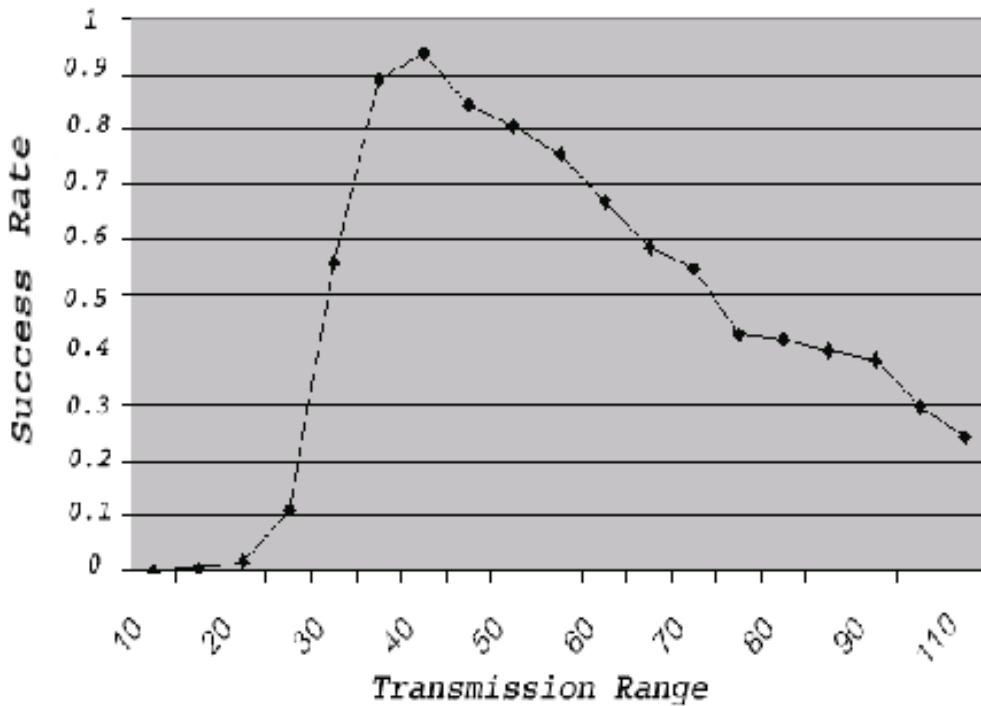
Σχήμα 7.5: Αριθμός ενεργών κόμβων για $I_s = 0.05$ και δίκτυο 500×500



Σχήμα 7.6: Αριθμός ενεργών κόμβων για $I_s = 0.05$ και δίκτυο 1000×1000

επιτυχίας είναι επαρκώς υψηλές (> 0.8).

Αντίθετα για το H-TEEN χρησιμοποιήθηκαν οι παράμετροι που αναφέρονται στις λεπτομέρειες υλοποίησης του πρωτοκόλλου. Σίγουρα αποτελεί μελλοντικό στόχο η παραμετροποίηση των επιπέδων ιεραρχίας που μπορεί να υπάρχουν στο H-TEEN, όπως επίσης και η παραμετροποίηση ανά επίπεδο ιεραρχίας της πιθανότητας εκλογής ενός κόμβου ως αρχηγού. Έτσι όμως μπορούμε να αποκομίσουμε σημαντικές πληροφορίες για τις επιδόσεις του H-TEEN μεταβάλλοντας την τιμή των παραμέτρων του.



Σχήμα 7.7: Πυθμός επιτυχίας του SW-PFR συναρτήσει της ακτίνας εκπομπής

Μέρος III

Ανάπτυξη πραγματικού πειραματικού
δικτύου έξυπνης σκόνης

Κεφάλαιο 8

Εξέλιξη των συσκευών για δίκτυα αισθητήρων

Το μεγάλο ερευνητικό ενδιαφέρον που συγκέντρωσαν τα ασύρματα δίκτυα αισθητήρων, απέδωσε γρήγορα τις πρώτες πρότυπες συσκευές για τη στελέχωση τέτοιων δικτύων. Αν και είναι πολλές οι ερευνητικές κοινότητες που ήδη έχουν υλοποιήσει πρότυπους κόμβους, εμείς θα εστιάσουμε στα ερευνητικά επιτεύγματα του project WEBS που διεξάγεται στο Berkley University. Ο λόγος γι' αυτή την επιλογή είναι ότι από το project WEBS προέρχονται οι πρότυπες συσκευές που είχαμε στη διάθεση μας για την ανάπτυξη πραγματικών ασύρματων δικτύων με αισθητήρες στα πλαίσια της παρούσης διπλωματικής.

Στο ερευνητικό project WEBS [24], διεξάγονται δύο σημαντικές και παράλληλες έρευνες για την υλοποίηση συσκευών που μπορούν να στελεχώσουν τα ασύρματα δίκτυα αισθητήρων και τα μελλοντικά δίκτυα έξυπνης σκόνης. Η μία έρευνα φέρει την ονομασία Smart Dust, ενώ η δεύτερη την ονομασία COTS Dust. Και στις δύο μπορούμε να διακρίνουμε μια σειρά από υλοποιήσεις πρότυπων συσκευών, κάθε μια από τις οποίες έχει να προτείνει κάτι νέο και διαφορετικό από τις προηγούμενες, ενώ πολύ συχνά μια συσκευή διαθέτει πολύ πιο εξελιγμένα χαρακτηριστικά από τις προηγούμενες της σε σημείο να τις καθιστά ξεπερασμένες.

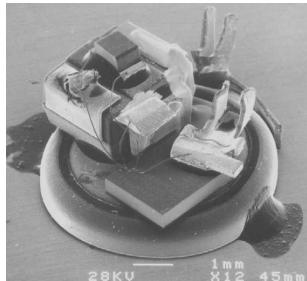
8.1 Συσκευές Smart Dust

Η πρώτη έρευνα υπό την ονομασία Smart Dust αφορά στον απευθείας σχεδιασμό και υλοποίηση μικροσκοπικών συσκευών της τάξης μερικών κυβικών χιλιοστών και με απότερο στόχο την παραπέρα μείωση του μεγέθους τους σε μεγέθη μικρότερα από ένα κυβικό χιλιοστό [32]. Στις πρότυπες συσκευές Smart Dust δεν χρησιμοποιούνται έτοιμα εμπορικά διαθέσιμα υποκυκλώματα (COTS). Αυτό γίνεται γιατί η μεγαλύτερη προτεραιότητα της έρευνας είναι η επίτευξη του μικρότερου δυνατού μεγέθους για τις συσκευές που θα στελεχώσουν τα μελλοντικά και πραγματικά δίκτυα έξυπνης σκόνης. Η αμέσως επόμενη προτεραιότητα είναι η ενσωμάτωση στις μικροσκοπικές αυτές συσκευές, όσο το δυνατό περισσότερων λειτουργιών ανίχνευσης, επεξεργασίας πληροφορίας και μετάδοσης χαμηλής ενεργειακής κατανάλωσης.

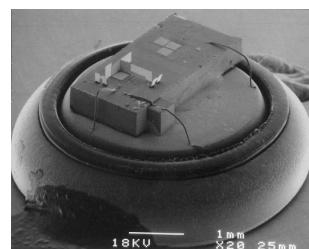
Οι πρόοδοι στο project Smart Dust έχουν ήδη να επιδείξουν κάποιες πρότυπες συσκευές [33]. Αυτές είναι τα Flashy Dust με διαστάσεις 138 mm^3 και τα Daft Dust με διαστάσεις 63 mm^3 . Επίσης υλοποιήθηκαν τα Golem Dust που μπορούν και αποθηκεύουν ηλιακή

ενέργεια, έχουν αισθητήρες φωτός και επιτάχυνσης και διαστάσεις γύρω στα 11 mm^3 .

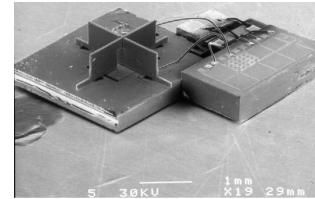
Γενικά αυτές οι συσκευές έχουν κατασκευαστεί, ώστε να περιέχουν τουλάχιστο έναν αισθητήρα και επιπλέον να έχουν δυνατότητες επικοινωνίας δυνατότητες με άλλες συσκευές. Ως μέθοδος επικοινωνίας προτιμάται η μετάδοση και λήψη με οπτικά μέσα. Πρέπει να σημειωθεί ότι εξακολουθούν να παραμένουν ζητήματα χαμηλής ενεργειακής κατανάλωσης από τις μεθόδους επικοινωνίας με οπτική τεχνολογία που πρέπει να μελετηθούν παραπέρα και να βρεθούν νέες υλοποιήσεις που θα παρέχουν ικανοποιητικές επιδόσεις όσο αφορά την καταναλισκόμενη ενέργεια. Επιπρόσθετα οι μικρές συσκευές που ήδη έχουν υλοποιηθεί δεν έχουν λύσει το πρόβλημα των σημαντικά περιορισμένων διαθέσιμων ενεργειακών πόρων που οφείλεται κατά κύριο λόγο στο μικρό τους μέγεθος. Ως πρότυπες συσκευές οι υλοποιήσεις αυτές έχουν αρκετές ατέλειες και προβλήματα, και δεν είναι εμπορικά διαθέσιμες.



Σχήμα 8.1: Flashy Dust

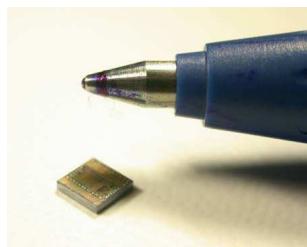


Σχήμα 8.2: Duft Dust



Σχήμα 8.3: Golem Dust

Τέλος πρόσφατα (2003) έχει υλοποιηθεί ακόμα μια πρότυπη μικροσκοπική συσκευή υπό την ονομασία Specs που έχει τη δυνατότητα RF επικοινωνίας με άλλες παρόμοιες συσκευές [36]. Η συσκευή έχει διαστάσεις $2\text{ mm} \times 2.5\text{ mm}$ και ανάμεσα στα άλλα χαρακτηριστικά της περικλείει ένα μικροελεγκτή AVR-like, 3 KBytes μνήμη, 8-bit ADC, FSK ραδιο-πομπό και δυνατότητα άμεσης επικοινωνίας με σειριακή θύρα UART. Ωστόσο, στα αδύνατα σημεία του σχεδιασμού της συσκευής συγκαταλέγονται τόσο η ανάγκη για εξωτερική πηγή ενέργειας και όσο και η ανάγκη για εξωτερική κεραία.



Σχήμα 8.4: Specs Dust



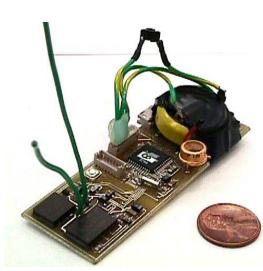
Σχήμα 8.5: Ένα Specs Dust mote πάνω στον επεξεργαστή ενός MICA mote

8.2 Συσκευές COTS dust

Εξίσου σημαντικά επιτεύγματα παρουσιάστηκαν και στη παράλληλη έρευνα για δημιουργία συσκευών COTS dust. Η υλοποίηση των κόμβων Smart Dust με τις αυστηρές προδιαγραφές του μικροσκοπικού μεγέθους, μικρού κόστους, επαρκών ενεργειακών πόρων και πολλαπλής λειτουργικότητας, απαιτούσε κάποια χρόνια έρευνας για να γίνει εφικτή, πρακτικά εφαρμόσιμη σε πραγματικά δίκτυα και οικονομικά βιώσιμη. Η συνειδητοποίηση αυτή οδήγησε στην ανάπτυξη του ερευνητικού προγράμματος COTS Dust [10]. Στο πρόγραμμα αυτό επιδιώχθηκε να δημιουργηθούν όσο το δυνατό νωρίτερα οι πρώτες πρότυπες συσκευές δικτύων για αισθητήρες. Για το σκοπό αυτό οι συσκευές αυτές χρησιμοποιούν επιμέρους ηλεκτρονικά κυκλώματα που είναι ήδη εμπορικά διαθέσιμα (COTS = Commercial Off The Self components). Όλες οι υλοποιήσεις COTSdust είναι σχεδιασμένες ώστε να αποτελούν αυτόνομους κόμβους ενός δικτύου αισθητήρων, με χαμηλό κόστος υλοποίησης, με δυνατότητες δυναμικού επαναπρογραμματισμού και λειτουργία χωρίς να είναι απαραίτητη η διαρκή επιτήρηση ή διαχείριση τους από ανθρώπινο προσωπικό.

Οι διαστάσεις αυτών των συσκευών είναι γύρω στη μία κυβική ίντσα, πολύ μακριά από το στόχο το κυβικού χιλιοστού που θέτουν οι προδιαγραφές των Smart Dust. Ωστόσο περικλείουν όλες τις επιθυμητές λειτουργίες ενός κόμβου έξυπνης σκόνης και αποτελούν τη πρώτη πλατφόρμα για την εφαρμογή προτεινόμενων πρωτοκόλλων και την πειραματική αξιολόγηση τους, παρέχοντας σημαντική πληροφορία για τον τρόπο λειτουργίας των μελλοντικών πραγματικών δικτύων έξυπνης σκόνης. Συνεπώς το παράλληλο project των Smart Dust συσκευών επωφελήθηκε σημαντικά τόσο από τις εξελίξεις στο τομέα των COTS dust όσο και από τα συμπεράσματα που εξήχθηκαν κατά τη χρήση τους στην ανάπτυξη πρότυπων πραγματικών δικτύων.

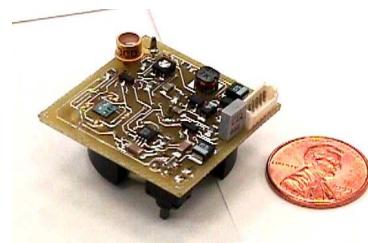
Οι περισσότερες από τις πρότυπες συσκευές που έχουν ήδη υλοποιηθεί περιέχουν έναν Atmel AVR επεξεργαστή, ένα πομποδέκτη RF ή οπτικής τεχνολογίας και ένα σύνολο από αισθητήρες. Επιπρόσθετα τροφοδοτούνται από μπαταρίες λιθίου 3 V.



Σχήμα 8.6: RF mote



Σχήμα 8.7: Laser mote



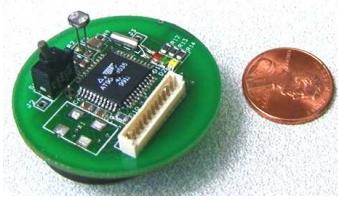
Σχήμα 8.8: CCR mote

Τυπικά χαρακτηριστικά των μακρό-συσκευών COTS Dust είναι η περιορισμένη ακτίνα μετάδοσης, που κυμαίνεται γύρω στα 20 m, και ο πεπερασμένος χρόνος ζωής που είναι κοντά στη μια εβδομάδα αν μια συσκευή είναι σε συνεχή λειτουργία. Ωστόσο, ο χρόνος ζωής τους μπορεί να φτάσει και τα δύο χρόνια αν υποθέσουμε λειτουργία με αποδοτικές τεχνικές διαχείρισης της ενεργειακής κατανάλωσης.

Ο Seth Edward-Austin Hollar στο MIT σχεδίασε και υλοποίησε αρχετές συσκευές CO-

TSdust, όπως τα RF COTSdust, laser COTSdust και CCR COTSdust. Τα ονόματα των συσκευών αυτών είναι ενδεικτικά της τεχνολογίας που χρησιμοποιούν για την επικοινωνία τους με ομώνυμες συσκευές, Έτσι τα RF COTSdust ενσωματώνουν ραδιοπομποδέκτες, τα laser COTSdust διαθέτουν κυκλώματα για ενεργή οπτική επικοινωνία και τα CCR COTSdust έχουν τη δυνατότητα παθητικής οπτικής μετάδοσης με τη χρήση της τεχνολογίας CCR (Corner Cube retroreflection).

Ενώ αρχικά το μέγεθος των συσκευών COTS dust ήταν μερικά κυβικά εκατοστά, η νεότερη γενιά τους παρουσιάζει μεγέθη μικρότερα από ένα κυβικό εκατοστό (μέγεθος ενός νομίσματος). Τα weC motes ανήκουν στη νεότερη αυτή γενιά και υλοποιήθηκαν το Σεπτέμβριο του 1999. Ανάμεσα στα άλλα ενσωματώνουν ένα επεξεργαστή AT90LS8535 4 MHz, αισθητήρες φωτός και θερμοκρασίας και μία κεραία πάνω στο ίδιο ολοκληρωμένο. Επιπρόσθετα για πρώτη φορά στα motes αυτά εξασφαλίστηκε η δυνατότητα να μπορούν να προγραμματιστούν πάνω από ασύρματη σύνδεση, σε αντίθεση με τα προηγούμενα COTSdust που προγραμματίζονται μόνο πάνω από σειριακή ή παράλληλη σύνδεση με τερματικό.



Σχήμα 8.9: WeC COTS mote

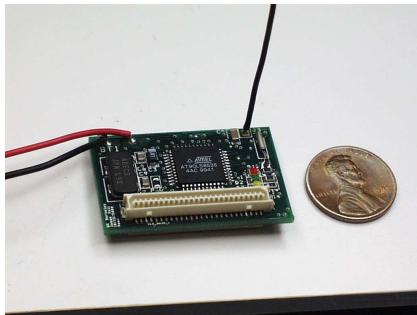
Στα COTS dust υπάγονται και οι συσκευές Rene motes [18]. Τα Rene motes υλοποιήθηκαν τον Οκτώβρη του 2000 και αποτέλεσαν την πρώτη εμπορικά διαθέσιμη πλατφόρμα για την ανάπτυξη πραγματικών δικτύων αισθητήρων από συσκευές COTS dust με δυνατότητες ανίχνευσης, επεξεργασίας δεδομένων και μετάδοσης στο ασύρματο δίκτυο. Σύμφωνα με το [34] ήδη αρκετές χιλιάδες από τα Rene motes έχουν αγοραστεί από ερευνητικά ιδρύματα προωθώντας την έρευνα και τις εφαρμογές στο τομέα των ασυρμάτων δικτύων. Στα motes αυτά ενσωματώθηκε ο ίδιος επεξεργαστής με τα weC, δηλαδή ένας AT90LS8535 4 MHz, με προγραμματιζόμενη μνήμη των 8 KByte και διαθέσιμη RAM μόλις 0.5 KB. Το σύστημα πομποδέκτη που περικλείουν είναι το TR1000 της Monolithic που επιτρέπει επικοινωνία RF σε συχνότητες 916 MHz. Οι ταχύτητες μεταδόσεων που πετυχαίνουν τα Rene motes φτάνουν τα 10 Kbps.

Τον Ιούνιο του 2001 παρουσιάστηκε στην αγορά η δεύτερη έκδοση των Rene motes, τα Rene2 motes που είχαν το ίδιο σύστημα πομποδέκτη αλλά ενσωμάτωναν τον νέο μικροελεγκτή ATMega163. Με τον τρόπο αυτό διπλασιάστηκε τόσο η προγραμματιζόμενη μνήμη των motes όσο και η διαθέσιμη RAM αποκτώντας πλέον τιμές 16 KB και 1 KB αντίστοιχα.

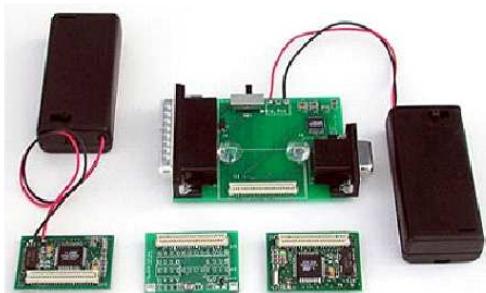
Ας σημειωθεί ότι τα Rene motes δε διαθέτουν ενσωματωμένη κεραία και απαιτούν πρόσθετη εξωτερική για μεταδόσεις σε ικανές αποστάσεις. Επιπρόσθετα δεν διαθέτουν ενσωματωμένη δυνατότητα μοναδικής ταυτοποίησης τους μέσα σε ένα δίκτυο από motes.

Το ενεργειακό κόστος μεταδόσεων στα Rene motes κυμαίνεται κοντά στο $1 \mu J/bit$. Σε κατάσταση αδράνειας (inactive state) απαιτείται ρεύμα μόλις $5 \mu A$, ενώ σε κατάσταση πλήρους λειτουργίας η μέγιστη ροή ρεύματος φτάνει τα $20 mA$.

Τα Rene motes μπορούν να συνδεθούν με μια ειδικά σχεδιασμένη πλακέτα, όπου συνδυάζονται κυκλώματα αισθητήρων, ώστε να αποκτήσουν δυνατότητες ανίχνευσης του περιβάλλοντος. Η πλακέτα αυτή ονομάζεται basic sensorboard και διαθέτει δυνατότητες μετρήσεως της θερμοκρασίας και των επιπέδων φωτεινότητας του περιβάλλοντος χώρου. Ουσιαστικά η πλακέτα “κουμπώνει” πάνω σε ειδική θύρα που διαθέτουν τα motes και έτσι μόνο όσα motes είναι απαραίτητο διαθέτουν δυνατότητες ανίχνευσης, ενώ τα υπόλοιπα μπορούν να έχουν τις βασικές δυνατότητες επεξεργασίας και διάδοσης πληροφορίας.



Σχήμα 8.10: Rene mote



Σχήμα 8.11: Η πλατφόρμα των Rene motes

Επιπρόσθετα τα Rene motes μπορούν να προγραμματιστούν με διάφορες εφαρμογές, αν συνδεθούν πάνω σε μια πλακέτα προγραμματισμού. Η πλακέτα αυτή είναι ειδικά σχεδιασμένη για σύνδεση με τα Rene motes και διαθέτει δυνατότητες επικοινωνίας με την παράλληλη και τη σειριακή θύρα ενός PC. Έτσι με τη χρήση ενός H/Y και με το κατάλληλο λογισμικό, το οποίο είναι η 0.6.1 έκδοση του TinyOS, γίνεται εφικτός ο ακριβής καθορισμός της λειτουργίας που επιθυμούμε να επιτελεί κάθε Rene mote σε ένα πρότυπο πραγματικό πειραματικό δίκτυο αισθητήρων.

Για την ανάπτυξη της πλατφόρμας Rene συγκροτήθηκε μια on line κοινότητα. Ανάμεσα σε αυτούς που αποτελούν μέλη της κοινότητας είναι οι U.C. Berkeley, U.C. Los Angeles, Intel Research Labs, Robert Bosch Corp., U.S. Air Force Research Labs, Crossbow Technology. Η ανάπτυξη γίνεται σχεδόν αποκλειστικά σε μοντέλο open-source. Αυτό σημαίνει ότι η σχεδίαση του hardware λαμβάνει χώρα σε δημόσιο domain (public domain) και επιπρόσθετα γίνονται πολλές συνεισφορές από μέλη σε πηγαίο κώδικα προγραμμάτων. Η οργάνωση αυτή οδηγεί σε ραγδαίες προόδους και καλή εκμετάλλευση του feedback που προέρχεται από τους αγοραστές των motes.

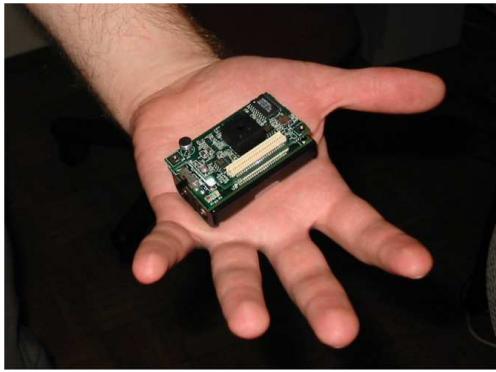
8.2.1 MICA motes

Η πλατφόρμα δεύτερης γενιάς για πρότυπους Smart Dust κόμβους κατασκευασμένους από COTS κυκλώματα είναι τα MICA motes [41]. Η πλατφόρμα αυτή αναπτύχθηκε βασισμένη στο feedback πάνω σε θέματα που προέκυψαν από τη διάθεση στην αγορά των Rene motes και σε προτάσεις - σχόλια των πελατών. Η υλοποίηση της έγινε το Φεβρουάριο του 2002 στο UC Berkley σε συνεργασία με άλλους ερευνητές που ανήκουν στην on-line κοινότητα ανάπτυξης συσκευών COTS dust. Τα MICA motes αποτέλεσαν μια σημαντική αναβάθμιση έναντι των Rene motes και ουσιαστικά τα εκτόπισαν από την αγορά.

Όπως και οι προκάτοχοι τους, τα MICA motes είναι κόμβοι COTS Dust με δυνατότητες ασύρματης επικοινωνίας και επεξεργασίας δεδομένων. Το μέγεθός τους είναι περίπου $2.0 \times 1.5 \times 0.5$ ίντσες. Τα MICA motes διαθέτουν προγραμματιζόμενο μικροεπεξεργαστή και δυνατότητα διασύνδεσης με πλακέτα που διαθέτει αισθητήρες, ώστε να λειτουργήσουν ως πραγματικοί κόμβοι ενός δίκτυου από αισθητήρες. Τέλος η τροφοδοσία τους γίνεται από οποιαδήποτε $3 V$ πηγή ρεύματος, όπως δύο AA μπαταρίες ή ένας μετασχηματιστής. Το βασικό υλικό των MICA motes καταναλώνει ισχύ μικρότερη από $1 Watt$.



Σχήμα 8.12: MICA mote



Σχήμα 8.13: Οι διαστάσεις του MICA mote σε σχέση με την ανθρώπινη παλάμη

Τα MICA motes διατίθενται στην αγορά από την Crossbow Technologies [37]. Τα MICA motes ήταν οι συσκευές που είχαμε στη διάθεση μας για να αναπτύξουμε πρότυπο πραγματικό πειραματικό δίκτυο έξυπνης σκόνης στα πλαίσια της παρούσης διπλωματικής. Αυτός είναι και ο λόγος που θα εμβαθύνουμε περισσότερο στις προδιαγραφές και τα τεχνικά χαρακτηριστικά τους.

Η πλατφόρμα των MICA motes απαρτίζεται από τρία βασικά στοιχεία. Το πρώτο στοιχείο είναι η πλακέτα MPR300 με τα βασικά ηλεκτρονικά στοιχεία του MICA mote (mote processor/radio, MPR board), η οποία κατά βάση περικλείει ένα μικροελεγκτή και ένα ραδιοπομπό RF. Το δεύτερο στοιχείο είναι μια πλακέτα με αισθητήρες, η οποία μπορεί να προσαρτηθεί πάνω στη βασική πλακέτα του MICA mote (plug-in) διευρύνοντας έτσι τη λειτουργικότητα του mote. Μια σειρά από πλακέτες αισθητήρων είναι διαθέσιμες εμπορικά, ώστε να ανταποκρίνονται σε διαφορετικούς τύπους εφαρμογών και ποικίλες ανάγκες ανίχνευσης και παρατήρησης. Τέλος, στην πλατφόρμα ανήκει και η πλακέτα προγραμματισμού των motes, (programming board) η οποία έχει το διττό ρόλο του προγραμματισμού των motes με ειδικά προγράμματα που καθορίζουν τη λειτουργία τους, αλλά και της επικοινωνίας των motes με ηλεκτρονικό υπολογιστή υπηρετώντας κατά κύριο λόγο σκοπούς καταγραφής πληροφοριών σε κέντρο ελέγχου.

Ο επεξεργαστής των MICA motes στην υλοποίηση της Crossbow είναι ένας Atmel AVR ATmega128L με συχνότητα λειτουργίας $4 MHz$. Σύμφωνα με τα specs που παρέχει η Atmel για τον επεξεργαστή αυτό [35], πρόκειται για ένα χαμηλής κατανάλωσης $8-bit$ μικροελεγκτή που βασίζεται στην ενισχυμένη RISC αρχιτεκτονική του AVR. Ο ATmega128 μπορεί να επιτυγχάνει throughputs που φτάνουν το $1 MIPS$ ανά Mhz και επιτρέπει στο σχεδιαστή ενός συστήματος να βελτιστοποιήσει το συνδυασμό επεξεργαστικής ταχύτητας και κατανάλωσης ισχύος. Η τάση λειτουργίας κυμαίνεται στα 2.7 με $5.5 Volts$ και η συχνότητα



Σχήμα 8.14: Programming board



Σχήμα 8.15: Ένα MICA mote προσαρτημένο στο programming board

λειτουργίας μπορεί να φτάνει τα 16 Mhz . Ανάμεσα στα άλλα χαρακτηριστικά ο μικροελεγκτής περιέχει μια μνήμη Flash 128 Kbytes , 4 Kbytes EEPROM, ένα μετρητή πραγματικού χρόνου, ένα 10 bit ADC 8 καναλιών και ένα προγραμματιζόμενο χρονόμετρο Watchdog. Πέρα από τη κανονική λειτουργία του, ο επεξεργαστής υποστηρίζει τη μετάβαση μεταξύ έξι καταστάσεων λειτουργίας χαμηλής κατανάλωσης (sleep modes) με απώτερο στόχο τη καλύτερη διαχείριση της καταναλισκόμενης ενέργειας. Τα 6 παρεχόμενα sleep modes είναι τα idle, power down, power save, ADC noise reduction, standby και extended standby. Σε κατάσταση χαμηλής κατανάλωσης κάποια από τα ηλεκτρονικά στοιχεία σταματούν να λειτουργούν εξοικονομώντας ενέργεια, ενώ παράλληλα εξασφαλίζεται η λειτουργία με λιγότερο παρεμβαλλόμενο θόρυβο για τα υπόλοιπα στοιχεία.

Στο βασικό υλικό των motes υπάρχει επίσης ένας χαμηλής ισχύος πομποδέκτης RF, μια εσωτερική (on-board) κεραία και διακριτά κυκλώματα για να ρυθμίζουν χαρακτηριστικά του φυσικού επιπέδου επικοινωνίας (physical layer), όπως είναι η ευαισθησία στη λήψη σήματος και η ισχύς του αποστελλόμενου σήματος. Ο ραδιο πομποδέκτης RFM TR1000 είναι από τα βασικά στοιχεία του MPR κυκλώματος (πλακέτας) καθώς ευθύνεται για τη διάδοση και λήψη πληροφοριών και την πραγματοποίηση της συνεργασίας των κόμβων στο Smart Dust δίκτυο. Σύμφωνα με τα τεχνικά χαρακτηριστικά του [40] είναι ειδικά σχεδιασμένος για εφαρμογές που απαιτούν μικρής εμβέλειας ασύρματη επικοινωνία. Το μικρό του μέγεθος, η χαμηλή ενεργειακή κατανάλωση και το πολύ μικρό κόστος υλοποίησης είναι κάποια πρόσθετα στοιχεία που τον καθιστούν ιδανικό για τα MICA motes. Η συχνότητα λειτουργίας του πομποδέκτη είναι τα 916 MHz (ISM band transceiver) και μπορεί να επιτυγχάνει ταχύτητες μετάδοσης / λήψης που φτάνουν τα 50 Kbps χρησιμοποιώντας το μοντέλο των on/off keyed μεταδόσεων. Ο ραδιο πομποδέκτης μπορεί και αυτός να υπακούει σε πολιτικές διαχείρισης ενέργειας καθώς με τη βοήθεια κατάλληλων σημάτων ελέγχου μπορεί να λειτουργεί ως πομπός ή δέκτης ή να είναι κλειστός. Τέλος το κύκλωμα TR1000 δεν διαθέτει λειτουργία αποθήκευσης των λαμβανόμενων bit σε ουρά αναμονής (buffering), οπότε το κάθε bit πρέπει να το διαχειρίζεται εγκαίρως ο επεξεργαστής του MPR.

Η εμβέλεια της εσωτερικής κεραίας που διαθέτουν τα motes είναι προγραμματιζόμενη, το οποίο σημαίνει ότι μπορεί το κάθε mote να προγραμματιστεί, ώστε να επιτευχθεί η επιυψημητή ακτίνα μετάδοσης. Αυτό γίνεται στην ουσία καθορίζοντας την τιμή της αντίστασης ενός δεδομένου ποτενσιόμετρου που επηρεάζει τη ροή του ρεύματος από το κύκλωμα της κεραίας. Είναι αναμενόμενο ότι μεγαλύτερη τιμή για την αντίσταση αυτή αντιστοιχεί σε μικρότερη ροή ρεύματος και κατά συνέπεια σε μικρότερη εμβέλεια. Γενικά η ακτίνα μετάδοσης, χρησιμο-

ποιώντας την εσωτερική κεραία των motes, κυμαίνεται μεταξύ ενός εκατοστού και περίπου 2 μέτρων.

Προαιρετικά στα MICA motes μπορεί να συνδεθεί εξωτερική κεραία αντικαθιστώντας την εσωτερική. Στο δικτυακό τόπο [39] περιγράφεται η διαδικασία προσάρτησης μια τέτοιας κεραίας σε ένα mote. Η διαδικασία αυτή απαιτεί την αχρήστευση της εσωτερικής κεραίας του mote και εξασφαλίζει ακτίνα μετάδοσης μέχρι και μερικές εκατοντάδες μέτρα.

Το MRP κύκλωμα έχει μοναδική διεύθυνση 64-bit που είναι μόνιμα εγγεγραμμένη στο υλικό τμήμα του (hard coded). Αυτή η διεύθυνση είναι η ψηφιακή ταυτότητα του κάθε κόμβου δοσμένη από την εταιρεία Dallas Semiconductors και είναι μοναδική ανάμεσα σε όλα τα MICA motes. Τέλος πάνω στο MPR, βρίσκονται 3 LEDS τα οποία μπορούν να χρησιμοποιηθούν από το προγραμματιστή του mote για να υποδεικνύουν την πορεία εκτέλεσης μιας εφαρμογής και να επιβεβαιώνουν τη σωστή λειτουργία του mote.

Η διάρκεια ζωής του κάθε κόμβου εξαρτάται από τη διάρκεια ζωής της μπαταρίας του σε συνδυασμό με την κατανάλωση ισχύος από τις δραστηριότητες του κόμβου. Σε μια τυπική εφαρμογή που αναθέτει ένα κανονικό φόρτο εργασίας στον επεξεργαστή, στο ραδιοπομπό και στους αισθητήρες αντιστοιχεί ενεργειακή κατανάλωση περίπου 100 mW. Όταν όλα τα στοιχεία είναι σε κατάσταση χαμηλής κατανάλωσης (sleep mode), τότε η κατανάλωση πέφτει στα 30 mW. Γενικά λοιπόν είναι επιθυμητό να υπάρχουν μεγάλοι περίοδοι λειτουργίας με χαμηλή ενεργειακή κατανάλωση και μικρά διαλείμματα, όπου θα λαμβάνει χώρα η κυρίως λειτουργία του mote.

Πρέπει εδώ να αναφερθεί ότι ήδη η Crossbow Technology έχει προχωρήσει στην εμπορική διάθεση της επόμενης γενιάς COTS dust συσκευών που έχουν την ονομασία MICA2 (MPR400) και MICA2Dot (MPR500). Η νέα αυτή γενιά συσκευών έχει βελτιωμένα χαρακτηριστικά όσον αφορά τον πομποδέκτη, καθώς αυτός είναι πολυκάναλος και έχει μεγαλύτερη εμβέλεια μεταδόσεων. Επιπρόσθετα τόσο τα MICA2 motes όσο και τα MICA2Dot διαθέτουν ενσωματωμένο αισθητήρα θερμοκρασίας και μετρητή των διαθέσιμων ενεργειακών αποθεμάτων. Τέλος τα MICA2 motes μπορούν να επικοινωνούν και να συνεργάζονται με τα MICA2Dot motes, και ο προγραμματισμός και των δύο μπορεί να γίνει από ασύρματο μέσο.

8.2.2 Κυκλώματα αισθητήρων

Υπάρχουν περισσότερα από ένα board κυκλωμάτων με αισθητήρες (MTS modules) που υπό τη μορφή μικρής πλακέτας μπορούν να προσαρμόζονται πάνω στην πλακέτα MPR. Η διεπαφή και σύνδεση των δύο κυκλωμάτων γίνεται από μια 51 pin υποδοχή. Πάνω στο MPR κύκλωμα υπάρχει ένας A/D μετατροπέας 8 καναλιών επιτρέποντας τη σύνδεση με πλακέτες MTS που διαθέτουν αναλογικούς ή ψηφιακούς σένσορες. Η δειγματοληψία όλων των αισθητήρων γίνεται από το μικροεπεξεργαστή του MICA mote και για κάθε δείγμα παράγεται μια πληροφορία μεγέθους 10-bit.

Τα MTS κυκλώματα είναι εύκολα στη σχεδίαση και στη ρύθμιση. Μοναδική απαίτηση είναι η χρήση του 51-pin συνδέσμου για τη μετάδοση δεδομένων στον επεξεργαστή της MPR πλακέτας. Ο λόγος που υπάρχουν κυκλώματα MTS που διαθέτουν κάποιους αισθητήρες αλλά δε διαθέτουν άλλους είναι κατά βάση το ότι τα κυκλώματα αυτά απευθύνονται σε διαφορετικούς τύπους εφαρμογών. Επιπρόσθετα πρέπει κανείς να λάβει υπόψη του ότι με τον τρόπο αυτό γίνονται περικοπές από το κόστος υλοποίησης των πλακετών MTS.

Στο δικτυακό τόπο της Crossbow technology [37] υπάρχουν τα τεχνικά χαρακτηριστικά των MTS κυκλωμάτων με αισθητήρες που διατίθενται εμπορικά. Τα MTS κυκλώματα είναι συμβατά με τα MICa motes και τα καινούργια MICa2 motes, ενώ για τα MICa2Dot motes διατίθεται ένα ειδικό κύκλωμα (MDA500) για συλλογή πληροφορίας, λόγω της διαφορετικής κατασκευής και του ιδιόρρυθμου σχήματος των τελευταίων.

Το πιο απλό μοντέλο κυκλώματος με αισθητήρες είναι το MTS101, που περικλείει μόνο ένα ανιχνευτή φωτός και ένα ανιχνευτή θερμότητας. Για περισσότερο πολύπλοκες εφαρμογές υπάρχει το κύκλωμα MTS300 το οποίο διαθέτει επιπρόσθετα ένα μονοτονικό ηχείο Sirius PS14T40A 4.3 KHz, και ένα Panasonic WM-62A μικρόφωνο. Το ζωνοδιαβατό σήμα εξόδου από το μικρόφωνο κατευθύνεται σε ένα ανιχνευτή ηχητικού τόνου National Semiconductor LMC567CM που λειτουργεί σε συχνότητα 4.5 KHz. Έτσι γίνεται εφικτή η ανίχνευση της παρουσίας ήχου που μπορεί να υπάρχει στο περιβάλλον του mote.

Στο paper [38] αναφέρεται και η χρήση ενός κυκλώματος με αισθητήρες που ονομάζεται weather board και μπορεί να παρακολουθεί τις αλλαγές στις περιβαλλοντολογικές συνθήκες ενός χώρου υποκαθιστώντας τηρουμένων των αναλογιών ένα σύγχρονο μετεωρολογικό σταθμό. Οι αισθητήρες που περιέχει είναι ανιχνευτές θερμοκρασίας, βαρομετρικής πίεσης, ανιχνευτής φωτός και ανιχνευτής υγρασίας.



Σχήμα 8.16: Το MTS310 sensor board



Σχήμα 8.17: Ένα MICa mote με το MTS310 sensor board

Η πλακέτα με αισθητήρες που είχαμε εμείς στη διάθεση μας ήταν το μοντέλο MTS310 που περικλείει όλους τους αισθητήρες του MTS300 και διαθέτει και κάποιους πρόσθετους. Συγκεκριμένα το MTS310 περιέχει τον ανιχνευτή θερμοκρασίας Panasonic ERT-J1VR103J, τον ανιχνευτή φωτός Clairex CL9P4L, το ίδιο μικρόφωνο και ηχείο με το MTS300 κύκλωμα και επιπλέον ένα μαγνητόμετρο HMC1002 της Honeywell και ένα μετρητή επιτάχυνσης σε δύο άξονες ADXL202JE της Analog Devices.

Κεφάλαιο 9

Προγραμματιστικό περιβάλλον και εφαρμογές

Η απαίτηση για έξυπνους κόμβους στα δίκτυα αισθητήρων, που θα μπορούν να επεξεργάζονται πληροφορίες, να δρομολογούν μηνύματα και να συνεργάζονται στη διάδοση κρίσιμων γεγονότων προς ένα κέντρο ελέγχου, συνεπάγεται την ανάγκη για ένα πακέτο λογισμικού που θα εξυπηρετεί στον προγραμματισμό των συσκευών. Ο προγραμματισμός αυτός εμπειρίζει τη ρύθμιση μιας συσκευής, ώστε να ακολουθεί τα βήματα ενός πρωτοκόλλου ή ενός αλγορίθμου αλλά και τον αποδοτικό και αποτελεσματικό συντονισμό των επιμέρους υποσυστημάτων του κάθε mote, ώστε αυτά να συνεργάζονται με τον επιμυητό κάθε φορά τρόπο. Βέβαια η κάθε λογισμική υποστήριξη υποχρεούται να συμβιβαστεί και να προσαρμοστεί σε αυστηρούς περιορισμούς που ισχύουν στα δίκτυα Smart Dust. Οι σοβαρότεροι από τους περιορισμούς αυτούς είναι η μικρού μεγέθους διαθέσιμη μνήμη στις συσκευές, οι περιορισμένες υπολογιστικές ικανότητες και τα πεπερασμένα ενεργειακά αποθέματα.

Ως πρώτη προτεραιότητα τέθηκε η υλοποίηση ενός λειτουργικού συστήματος που μπορεί να “εμφυτεύεται” στα MICA motes και να παρέχει δυνατότητες διαχείρισης των υπολογιστικών, επικοινωνιακών, ενεργειακών πόρων και των ανιχνευτικών δυνατοτήτων ενός mote. Αυτό το λειτουργικό σύστημα (Operating System) οφείλει να είναι ειδικά σχεδιασμένο για εφαρμογή σε ασύρματους κόμβους που ενσωματώνουν λειτουργίες αίσθησης και ανίχνευσης του περιβάλλοντος.

Ειδικότερα πρέπει να παρέχει συναρτήσεις για την παραμετροποίηση του διαθέσιμου υλικού των motes και την ενεργοποίηση του κατά τις ανάγκες εφαρμογών. Πρέπει να δίνει επιπρόσθετα δυνατότητες καθορισμού του ρυθμού δειγματοληψίας από τους αισθητήρες, αποθήκευσης και διαχείρισης των μετρήσεων που λαμβάνονται, να παρέχει ρουτίνες για την αποστολή και λήψη δεδομένων και γενικότερα τη δρομολόγηση πληροφορίας, διαδικασίες ελέγχου της λειτουργίας των υποσυστημάτων του mote και διαχείρισης της κατανάλωσης της ενέργειας. Παράλληλα θα πρέπει να εισάγει μια αφαιρετική θεώρηση του υλικού αποκρύπτοντας τις τεχνικές λεπτομέρειες της δικτυακής επικοινωνίας, της διαχείρισης ενέργειας και της αποκόμισης μετρήσεων από τους αισθητήρες, παρέχοντας με τον τρόπο αυτό ένα φιλικό περιβάλλον για την ανάπτυξη εφαρμογών (API). Τέλος πρέπει να είναι σε θέση να ανταποκρίνεται στις απαιτήσεις ενός μεγάλου εύρους εφαρμογών που μπορούν να έχουν τα Smart Dust δίκτυα και να εξασφαλίζει αξιόπιστη λειτουργία των motes σε κάθε περίπτωση, ενώ είναι σημαντικό να μην εξαρτάται από την αρχιτεκτονική ενός είδους κόμβου Smart Dust αλλά να εγγυά-

ται συμβατότητα με όλες τις συσκευές που μπορούν να χρησιμοποιηθούν σε δίκτυα έξυπνης σκόνης.

9.1 TinyOS

Μετά από αρκετή δουλειά και έρευνα, η οποία ξεκίνησε από τον Jason Hill του U.C. Berkley EECS Dpt έγινε πραγματικότητα το πρώτο λειτουργικό σύστημα για smart dust motes το οποίο φέρει την ονομασία TinyOS (Tiny Microthreading Operating System). Σύμφωνα με το [19, 20] πρόκειται για ένα λειτουργικό περιβάλλον που απαρτίζεται από ρουτίνες χειρισμού γεγονότων (event based) και προορίζεται για δικτυωμένους αισθητήρες. Είναι ειδικά σχεδιασμένο για να υποστηρίζει ταυτόχρονη εκτέλεση πολλών και εντατικών λειτουργιών που απαιτούν οι δικτυωμένοι αισθητήρες έχοντας, ωστόσο, ελάχιστες απαιτήσεις σε υλικό (μνήμη, επεξεργαστική ισχύ).

Καθώς το TinyOS έχει σχεδιαστεί αποκλειστικά για δίκτυα ασύρματων κόμβων με αισθητήρες, πρέπει να τονίσουμε τα ιδιαίτερα χαρακτηριστικά του που το καθιστούν ιδανικό για αυτού του είδους τα δίκτυα. Πρώτα όμως αναφέρουμε ότι στο TinyOS ως άμεση προτεραιότητα τίθεται η αποδοτική χρήση των διαθέσιμων ενεργειακών αποθεμάτων. Η προσέγγιση, που ακολουθεί το λογισμικό, υπαγορεύει την ενεργοποίηση υπολογιστικών μηχανισμών και διαδικασιών μόνο όταν συμβεί κάποιο γεγονός (πχ hardware interrupt) και αυτό εξασφαλίζει ότι το σύστημα χρησιμοποιεί τους υπολογιστικούς του πόρους αποδοτικά. Σε κάθε γεγονός αντιστοιχίζεται ένα σύνολο από διεργασίες, οι οποίες εκτελούνται άμεσα μόλις συμβεί το γεγονός, χωρίς να μπαίνουν σε ουρά αναμονής και έτσι εξαλείφεται η ανάγκη να ελέγχει ο μικροεπεξεργαστής με τεχνικές polling τα διάφορα υποσυστήματα του κόμβου. Την περίοδο όπου η κεντρική μονάδα επεξεργασίας δεν χειρίζεται ένα γεγονός, το TinyOS φροντίζει να θέτει την CPU σε κατάσταση Sleep.

Έπειτα πρέπει να αναφερθεί η δυνατότητα που παρέχει το TinyOS για δημιουργία αυτόνομου δικτύου (self configuration) από ασύρματους κόμβους. Είναι ειδικά σχεδιασμένο, ώστε να επιτρέπει στους κόμβους ενός Smart Dust δικτύου να οργανωθούν σε δίκτυο χωρίς προσχεδιασμένες διαδικασίες εγκατάστασης και τοποθέτησης των κόμβων.

Το TinyOS επίσης ανταποκρίνεται ικανοποιητικά στις απαιτήσεις για πραγματικού χρόνου (real time) ανταπόκριση του δικτύου σε διάφορα συμβάντα, όπως αυτό απαιτείται από πολλές εφαρμογές. Με τη χρήση πολυνηματισμού κατορθώνει να παραληλοποιεί τη διαχείριση των πολλαπλών μηνυμάτων με δεδομένα που λαμβάνονται από ένα κόμβο, τα οποία πρέπει να υποβληθούν σε επεξεργασία και να αποσταλούν σε μικρό χρονικό διάστημα καθώς δεν υπάρχει αποθηκευτικός χώρος για buffering και επιπλέον δεν επιθυμούμε αισθητές καθυστερήσεις στη ροή της πληροφορίας.

Τέλος πρέπει να αναφέρουμε ότι το TinyOS μπορεί να εξασφαλίσει την αξιόπιστη λειτουργία του δικτύου παρέχοντας μηχανισμούς για τη συνέχιση λειτουργίας ακόμα και όταν πολλές συσκευές του δικτύου πεθάνουν (εξαντληθούν ενεργειακά). Και ακόμα είναι άξιο αναφοράς το μεγάλο εύρος των εφαρμογών που υποστηρίζονται από το TinyOS. Παρέχοντας τη βασική υποδομή και το αναπτυξιακό περιβάλλον για το σχεδιασμό εξειδικευμένων εφαρμογών το TinyOS αποτελεί τελικά μια γενικευμένη (generic) πλατφόρμα, η οποία υποστηρίζει ανάπτυξη λογισμικού για συσκευές έξυπνης σκόνης χωρίς πολλές απαιτήσεις σε μνήμη και υπολογιστική ισχύ.

Το TinyOS διαφέρει από την κλασική θεώρηση των λειτουργικών συστημάτων, με την έννοια ότι δεν έχει την σύνθετη δομή ή το επίπεδο λειτουργικότητας που προσφέρει ένα σύγχρονο λειτουργικό σύστημα. Αυτό συμβαίνει κατά βάση γιατί ένας κόμβος του Smart Dust δικτύου δε απαιτεί τέτοια επίπεδα λειτουργικότητας και κατά δεύτερο λόγο επειδή δε διατίθενται οι πόροι για να υποστηρίζουν ένα “βαρύ” λειτουργικό. Βέβαια παρά την ονομασία του, το TinyOS δε θεωρείται αυτόνομα ως λειτουργικό. Είναι ένα συσσωμάτωμα λειτουργικού συστήματος και εφαρμογής το οποίο μετά από την απαιτούμενη μεταγλώττιση και την επιτυχή παραγωγή δυαδικού κώδικα μπορεί να εκτελεστεί σε ένα ATmega επεξεργαστή. Η αρχιτεκτονική του στηρίζεται στην εξυπηρέτηση γεγονότων, δεν υπάρχει επεξεργαστικός πυρήνας (kernel) ούτε περίπλοκη διαχείριση φυσικής μνήμης και δεν χρησιμοποιείται η τεχνική της ιδεατής μνήμης (virtual memory). Η μοναδική δομή μνήμης είναι μια στοίβα (stack) η οποία είναι διαμοιραζόμενη.

Η αρχιτεκτονική του TinyOS προβλέπει για κάθε εφαρμογή την ύπαρξη ενός απλοποιημένου χρονοδρομολογητή εργασιών (scheduler) και μια ιεραρχική δομή γράφου του οποίου οι κόμβοι ονομάζονται αρθρώματα (components). Κάθε άρθρωμα έχει 4 αλληλεπιδραστικά τμήματα : ένα σύνολο συναρτήσεων που διαχειρίζονται εντολές (command handlers), ένα σύνολο συναρτήσεων που διαχειρίζονται συμβάντα (event handlers), ένα σύνολο από απλές διαδικασίες που επιτελούν βασικές λειτουργίες (tasks), και μια δομή (frame) που υποδεικνύει την κατάσταση (status) του αρθρώματος. Όλες οι εντολές, οι συναρτήσεις χειρισμού και οι διεργασίες, εκτελούνται στα πλαίσια του περιβάλλοντος (context), το οποίο ορίζεται από τη δομή frame, και επιδρούν στην κατάσταση του component. Κάθε άρθρωμα, κατά τη διοδικασία υλοποίησης του, δηλώνει τις εντολές που χρησιμοποιεί, οι οποίες θα πρέπει να είναι διαθέσιμες και υλοποιημένες από κάποιο άρθρωμα κατώτερου ιεραρχικά επιπέδου. Επίσης δηλώνει και τα συμβάντα (events) που μπορεί να σηματοδοτήσει, ώστε να ενεργοποιήσει κάποια διαδικασία χειρισμού συμβάντος σε ένα ανώτερο ιεραρχικά component. Ως κατώτερο ιεραρχικά component θεωρείται το φυσικό υλικό του κόμβου.

Ο χωρισμός σε αρθρώματα, βοηθάει στη δημιουργία αυτόνομων λειτουργικών μονάδων που μπορεί να αντιπροσωπεύουν ένα αυτοδύναμο υποσύστημα του κόμβου (πχ. πομποδέκτης, αισθητήρας φωτός, αισθητήρας θερμοκρασίας). Τα περισσότερα από αυτά τα αρθρώματα μπορούν να επαναχρησιμοποιούνται σε διάφορες εφαρμογές, διευκολύνοντας το έργο του προγραμματισμού της λειτουργίας των κόμβων. Πρέπει να σημειωθεί ότι η εκάστοτε εφαρμογή υλοποιείται και αυτή ως άρθρωμα που βρίσκεται στο πιο υψηλό ιεραρχικά επίπεδο του γράφου. Για κάθε άρθρωμα η δομή frame, η οποία διατηρεί το context είναι πάντα σταθερού μεγέθους και έτσι μπορούμε να γνωρίζουμε με ακρίβεια, κατά το χρόνο μεταγλώττισης του κώδικα σε δυαδικό, ποιες είναι οι απαιτήσεις μιας εφαρμογής σε μνήμη. Συνεπώς αναιρείται κάθε ανάγκη για δυναμική ανάθεση μνήμης κατά το χρόνο εκτέλεσης μιας εφαρμογής, η οποία θα ήταν χρονοβόρα και θα συνεπαγόταν τη χρήση δεικτών σε θέσης μνήμης και πρόσθιτο κώδικα για αρχικοποίηση, δέσμευση και απελευθέρωση της μνήμης.

Ως εντολές (commands) ορίζονται οι απερίσπαστες (non blocking) αιτήσεις προς αρθρώματα κατώτερου επιπέδου ώστε αυτά να εκτελέσουν συγκεκριμένες λειτουργίες για τις οποίες είναι υπεύθυνα. Αυτό σημαίνει ότι κάθε άρθρωμα “διαφημίζει” τις λειτουργίες που είναι σε θέση να παρέχει κατ’ απαίτηση σε αρθρώματα του ανώτερου επιπέδου. Η ιεραρχική δομή του TinyOS υπαγορεύει ότι σε κάθε επίπεδο ένα άρθρωμα μπορεί να χρησιμοποιεί τη λειτουργικότητα που παρέχεται από τα κατώτερα επίπεδα αρθρωμάτων, για να υλοποιήσει τις δικές του ρουτίνες εξυπηρέτησης. Έτσι το κάθε επίπεδο εισάγει ένα νέο φίλτρο αφαίρεσης στην παρεχομένη λειτουργικότητα από το υλικό, διευκολύνοντας το μέσο προγραμματιστή. Όταν ένα άρθρωμα

εκτελέσει τη ρουτίνα που αντιστοιχεί σε μια εντολή, την οποία έλαβε από άρθρωμα ανώτερου επιπέδου, οφείλει να στείλει και μια απάντηση στο άρθρωμα αυτό για την επιτυχή ή όχι έκβαση της εντολής.

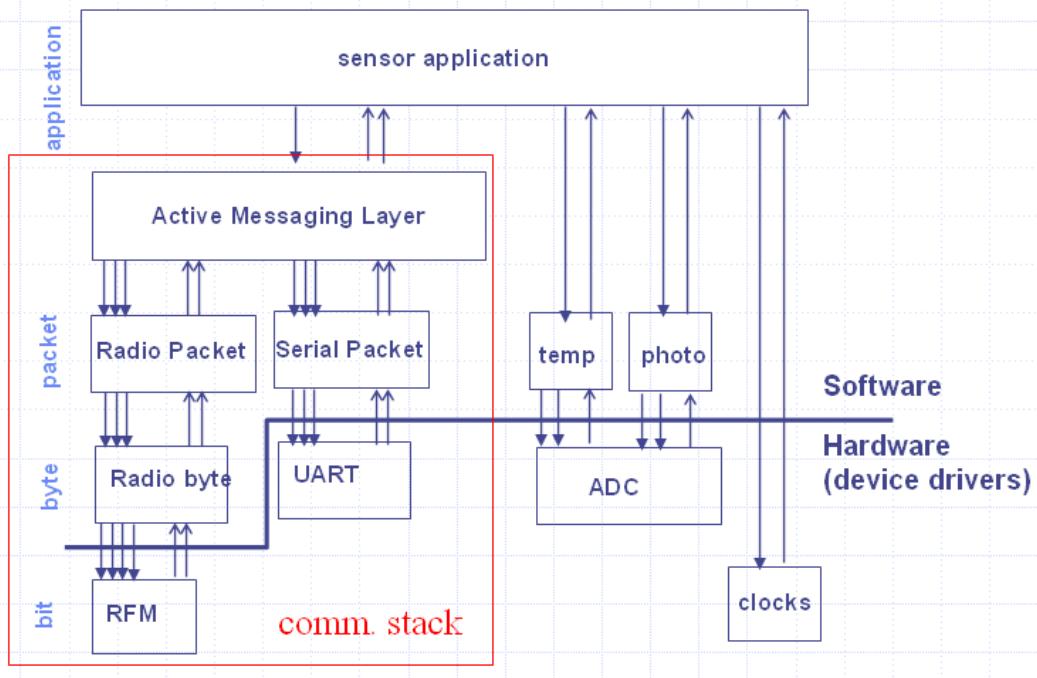
Οι διαχειριστές των συμβάντων (event handlers) καλούνται για να χειριστούν συμβάντα που προκαλούνται άμεσα ή έμμεσα από το υλικό. Τα άρθρωμα των κατώτερων επιπέδων ιεραρχίας διαθέτουν ρουτίνες άμεσου χειρισμού των διακοπών (interrupts) που προκαλούνται από το υλικό. Μια διαδικασία χειρισμού ενός συμβάντος, μπορεί να αποθηκεύει πληροφορίες στο context του άρθρωματος όπου ανήκει, να απαιτήσει την εκκίνηση απλών διαδικασιών (tasks), να αποστείλει εντολές (commands) σε άρθρωματα των κατώτερων επιπέδων ή ακόμα και να δημιουργήσει νέα συμβάντα που θα θέσουν σε λειτουργία διαδικασίες χειρισμού σε άρθρωματα των ανώτερων επιπέδων. Έτσι ένα συμβάν (interrupt) του υλικού είναι η αρετηρία για την εκτέλεση μιας αλυσίδας από ρουτίνες επεξεργασίας. Κατά την εκτέλεση της αλυσίδας αυτής μια δημιουργία νέου συμβάντος από ένα άρθρωμα οδηγεί σε άρθρωμα ανώτερου επιπέδου, για να το χειριστεί, ενώ μια εντολή οδηγεί σε άρθρωμα κατώτερου επιπέδου για να την εκτελέσει. Για να μη δημιουργηθούν ατέρμονοι βρόχοι υπαγορεύεται ρητά από το λογισμικό ότι οι εντολές δεν μπορούν κατά την εκτέλεση τους να δημιουργήσουν νέο συμβάν. Όπως και οι εντολές, έτσι και οι διαδικασίες χειρισμού γεγονότων εκτελούνται απερίσπαστες (non blocking).

Οι απλές διαδικασίες (tasks) επιτελούν στοιχειώδη εργασία για μια εφαρμογή και μπορούν να τεθούν σε αναμονή (preempted), αν συμβεί κάποιο κρίσιμο (critical) συμβάν. Μια απλή διαδικασία μπορεί να δίνει εντολές σε άρθρωματα κατώτερου επιπέδου, να δημιουργεί γεγονότα, τα οποία θα χειριστούν τα ανώτερα άρθρωματα και να καλεί άλλες απλές διαδικασίες του ίδιου άρθρωματος. Γενικά δεν πρέπει ένα άρθρωμα να προγραμματίζεται κάνοντας υποθέσεις για το χρόνο εκτέλεσης των tasks καθώς αυτά δεν εκτελούνται άμεσα, αλλά όποτε έρθει η σειρά τους σύμφωνα με τον χρονοπρογραμματιστή και η εκτέλεση τους αναστέλλεται με την ανίχνευση ενός γεγονότος. Έτσι τα tasks αντιπροσωπεύουν εργασίες που δεν επείγουν να πραγματοποιηθούν και δεν προϋποθέτουν κάποιον συγχρονισμό.

Ο χρονοπρογραμματιστής ακολουθεί πολιτική FIFO χρονοδρομολόγησης διεργασιών, με μια ουρά αναμονής πεπερασμένου μεγέθους. Ωστόσο, ανάλογα με τις απαιτήσεις της κάθε εφαρμογής, ο χρονοπρογραμματιστής μπορεί να κάνει χρήση περισσότερο πολύπλοκων δομών με πολιτικές προτεραιοτήτων ή λήξης καταχωρήσεων. Το TinyOS εξασφαλίζει ότι όταν αδειάσει η ουρά εξυπηρέτησης του χρονοπρογραμματιστή, ο επεξεργαστής θα μεταπηδήσει σε κατάσταση sleep (χαμηλής κατανάλωσης). Τα περιφερειακά υποκυλώματα θα συνεχίζουν τη λειτουργία τους, ώστε να μπορούν να αφυπνίσουν ανά πάσα στιγμή τον επεξεργαστή. Ας σημειωθεί ότι όταν αδειάσει η ουρά εξυπηρέτησης, τότε ένα νέο task μπορεί να προστεθεί σε αυτή μόνο μετά την ανίχνευση ενός γεγονότος. Συνεπώς ο χρονοδρομολογητής μπορεί να είναι ανενεργός μέχρι την εμφάνιση ενός νέου συμβάντος.

Ως ενδεικτικό μέγεθος μιας εφαρμογής που εκτελεί τόσο λειτουργίες ανίχνευσης (sensing) όσο και επικοινωνίας με άλλους κόμβους, δίνονται τα 3.4 KBytes. Αυτό είναι αξιοσημείωτο για εφαρμογή με τόση λειτουργικότητα και αποτελεί ένα από τα κύρια πλεονεκτήματα του TinyOS.

Το TinyOS διατίθεται σε δύο εκδόσεις, ήτοι την 0.6.1, που ήταν έτοιμη το Μάιο 2002 για τις πλατφόρμες MICA, RENE και RENE2 motes, και την έκδοση 1.0.0 που εκδόθηκε τον Οκτώβριο 14, 2002 για τα MICA motes. Η τελευταία αυτή έκδοση είναι και αυτή που είχαμε εγκατεστημένη σε H/Y για την υλοποίηση εφαρμογών του TinyOS στα πλαίσια αυτής της



Σχήμα 9.1: Ο γενικευμένος γράφος ιεραρχίας των αρθρωμάτων του TinyOS

διπλωματικής.

Η κάθε έκδοση του TinyOS, πέρα από τις βιβλιοθήκες του κυρίως λειτουργικού συστήματος και τον πηγαίο κώδικα για κάποια βασικά αρθρώματα, περιλαμβάνει μια συλλογή από εργαλεία προγραμματισμού και εργαλεία offline εξομοίωσης εφαρμογών σε εικονικά motes, και από παραδείγματα εφαρμογών με γραφική ή όχι διεπαφή. Τα λειτουργικά συστήματα που υποστηρίζουν την εγκατάσταση της αναπτυξιακής πλατφόρμας του TinyOS είναι τόσο το Linux όσο και τα Microsoft Windows (μέσω του unix - like περιβάλλοντος Cygwin). Ο πηγαίος κώδικας για την πιο πρόσφατη έκδοση του TinyOS, το μεταγλωττιστή NesC για τις εφαρμογές του TinyOS καθώς και μερικές ενδεικτικές εφαρμογές και κάποια on line βοηθητικά εγχειρίδια είναι διαθέσιμα στην on line κοινότητα ανάπτυξης open-source κώδικα, η οποία είναι γνωστή ως Sourceforge.

Για το εγγύς μέλλον η κοινότητα που ασχολείται με την ανάπτυξη του TinyOS κατευθύνεται σε επέκταση του λειτουργικού αυτού συστήματος, ώστε να είναι συμβατό με πολλές σύγχρονες πλατφόρμες ασύρματων δικτύων με απώτερο στόχο να καθιερωθεί η χρήση του σε μελλοντικές υλοποιήσεις πραγματικών δικτύων Smart Dust. Επίσης είναι επιθυμητό να ενισχυθούν τα πρωτόκολλά επικοινωνίας που υλοποιεί το TinyOS, με περισσότερο περίπλοκα επίπεδα MAC και επίπεδα (protocol layers) για τη διάδοση μηνυμάτων.

9.2 NesC

Η γλώσσα προγραμματισμού του TinyOS ονομάζεται NesC. Η γλώσσα αυτή είναι μια επέκταση της C και χρησιμοποιεί ένα εξειδικευμένο μεταφραστή (compiler), ο οποίος έχει δώσει το όνομα του και στη γλώσσα. Η NesC σχεδιάστηκε με στόχο την αποδοτική υλοποίηση

της αρχιτεκτονικής του TinyOS. Είναι μια χαμηλού επιπέδου, αντικειμενοστραφής γλώσσα που χρησιμοποιείται για τον προγραμματισμό των motes και ίσως μελλοντικά να χρησιμοποιηθεί ως ενδιάμεση γλώσσα προγραμματισμού για μελλοντικές υψηλού επιπέδου γλώσσες προγραμματισμού [21].

Ως επέκταση της C, η NesC μπορεί να χρησιμοποιηθεί για την παραγωγή αποδοτικού κώδικα για όλους τους μικροεπεξεργαστές που μπορούν να χρησιμοποιηθούν σε δίκτυα έξυπνης σκόνης. Η C ήδη παρέχει μεθόδους και συναρτήσεις χαμηλού επιπέδου για την προσπέλαση υποσυστημάτων του υλικού και επιπρόσθετα πολλοί προγραμματιστές είναι ήδη εξοικειωμένοι με τη χρήση της C. Η NesC εισάγει επιπρόσθετα το μοντέλο της δομημένης εφαρμογής, που χρησιμοποιεί ως λειτουργικές μονάδες κάποια αυτοτελή αρθρώματα.

Από τη NesC λείπουν εντελώς οι συναρτήσεις για δυναμική ανάθεση και χειρισμό μνήμης και πάντοτε κατά το χρόνο μεταγλωττισμού είναι πλήρως γνωστός ο γράφος με τις κλήσεις μιας εφαρμογής. Έτσι τα προγράμματα μπορούν να αναλυθούν και να βελτιωθούν ευχολότερα και με περισσότερη ακρίβεια.

Το πιο βασικό στοιχείο της NesC είναι ότι υποστηρίζει πλήρως το προγραμματιστικό μοντέλο του TinyOS. Αυτό συμπεριλαμβάνει την ιδέα του αρθρωτού προγραμματισμού και την λογική του άμεσου χειρισμού συμβάντων. Επιπρόσθετα η NesC παρέχει μεθόδους για την επίλυση συγκρούσεων και ασυνεπειών που προκύπτουν από τη χρήση διαμοιραζόμενης μνήμης σε ένα mote. Γενικά η NesC έλυσε πολλά από τα προβλήματα που είχαν προκύψει από την πρώτη υλοποίηση του TinyOS σε γλώσσα C, και αυτό οδήγησε στην επαναϋλοποίηση του TinyOS στη γλώσσα NesC.

Στη NesC οι εφαρμογές δημιουργούνται από συλλογή αρθρωμάτων, κάθε ένα από τα οποία έχει καλά δηλωμένες και ορισμένες διεπαφές (interfaces). Όπως θα γίνει φανερό παρακάτω μια εφαρμογή δημιουργείται ουσιαστικά με τη δικτύωση αρθρωμάτων (wiring components) και τη συναρμολόγηση ενός κεντρικού αρθρώματος από πολλά επιμέρους αρθρώματα. Κάθε άρθρωμα έχει δύο “όψεις” (scopes). Η μία όψη περιέχει τις δηλώσεις των συναρτήσεων του αρθρώματος και είναι ενδεικτική της λειτουργικότητας και της χρησιμότητας του αρθρώματος. Η δεύτερη όψη είναι εξίσου απαραίτητη καθώς περιέχει τις υλοποιήσεις των συναρτήσεων αυτών.

Ένα άρθρωμα στη NesC **παρέχει** (provides) και **χρησιμοποιεί** (uses) διεπαφές. Αυτές οι διεπαφές είναι ο μόνος τρόπος για την προσπέλαση ενός αρθρώματος από κάποιο άλλο. Η προσπέλαση αυτή σε ένα άρθρωμα μπορεί να γίνει είτε με μία διακοπή του συστήματος (hardware interrupt) ή από μια απλή διαδικασία task. Κάθε διεπαφή μοντελοποιεί μια παρεχόμενη υπηρεσία (όπως για παράδειγμα η λήψη ενός μηνύματος). Στο TinyOS προσφέρεται μια συλλογή από διεπαφές οι οποίες περιγράφουν ένα μεγάλο σύνολο από παρεχόμενες υπηρεσίες και λειτουργίες που μπορεί να υποστηρίζει ένα mote. Οι διεπαφές αυτές δηλώνονται σε ξεχωριστά αρχεία, τυήμα της βιβλιοθήκης του TinyOS και είναι ανεξάρτητες από τα αρθρώματα και τις υλοποιήσεις τους. Κάθε διεπαφή είναι απλά μια περιγραφή υπηρεσιών και δεν παρέχει καμία υλοποίηση αυτών των υπηρεσιών αφήνοντας περιθώρια στον προγραμματιστή να αποφασίσει πως θέλει να υλοποιηθούν.

Οι διεπαφές είναι αμφίδρομες. Αυτό σημαίνει ότι κάθε διεπαφή μπορεί να παρέχει ένα σύνολο συναρτήσεων που ονομάζονται **εντολές** (commands) και να χρησιμοποιεί ένα άλλο σύνολο από συναρτήσεις που ονομάζονται **γεγονότα** (events). Οι εντολές πρέπει να υλοποιούνται από το άρθρωμα που **παρέχει** (provides) τη διεπαφή, ενώ τα γεγονότα ή πιο σωστά οι

συναρτήσεις χειρισμού των γεγονότων πρέπει να υλοποιούνται από τα αρθρώματα που **χρησιμοποιούν** τη διεπαφή. Είναι σημαντικό να διαπιστώσει κανείς τη χρησιμότητα αυτού του μοντέλου. Καθώς το TinyOS υπαγορεύει ότι οι εντολές και οι συναρτήσεις χειρισμού συμβάντων είναι απερίσπαστες (non-blocking), ο τερματισμός τους μπορεί να ανιχνευτεί μόνο με τη σηματοδότηση ενός νέου γεγονότος (event) (πχ. commandDone). Έτσι για πρακτικούς σκοπούς ένα άρθρωμα που **χρησιμοποιεί** μια διεπαφή, δε μπορεί να χρησιμοποιήσει μια παρεχόμενη εντολή της διεπαφής, αν δεν υλοποιήσει πρώτα μια συνάρτηση χειρισμού του γεγονότος τερματισμού της εντολής.

Γενικά οι κλήσεις εντολών οδηγούν τον έλεγχο του προγράμματος σε αρθρώματα που ανήκουν σε χαμηλότερα επίπεδα ιεραρχίας ενώ τα γεγονότα οδηγούν τον έλεγχο ροής σε ανώτερα επίπεδα. Η σύνδεση των αρθρωμάτων μεταξύ τους είναι στατική και δεν αλλάζει με τη ροή του προγράμματος. Όπως αναφέρθηκε η σύνδεση αυτή γίνεται μέσω των διεπαφών που αυτά **παρέχουν** και των διεπαφών που **χρησιμοποιούν**. Αυτό επεξηγηματικά σημαίνει ότι ένα άρθρωμα που **χρησιμοποιεί** μια διεπαφή μπορεί να συνδεθεί μόνο με οποιοδήποτε άρθρωμα **παρέχει** αυτή την διεπαφή. Ιεραρχικά τότε το δεύτερο άρθρωμα θα είναι κατώτερο από το πρώτο.

9.3 Μια πρότυπη εφαρμογή σε NesC

Ο καλύτερος τρόπος για να κατανοήσει κανείς πλήρως τη λειτουργία της NesC είναι μέσα από μια απλή εφαρμογή προγραμματισμού ενός mote. Επιλέγουμε για το παράδειγμα αυτό να περιγράψουμε μια εφαρμογή η οποία διατηρεί ένα μετρητή, ο οποίος μετράει κυκλικά μεταξύ 0 και 7 και εμφανίζει τη δυαδική αναπαράσταση της τιμής του στα LEDS του mote, ενώ παράλληλα το mote αποστέλλει μέσω του πομπού του ένα μήνυμα με την τιμή του μετρητή. Η εφαρμογή αυτή λέγεται CntToLedsAndRfm και βρίσκεται ανάμεσα στις πρότυπες εφαρμογές που συνοδεύουν το λογισμικό του TinyOS.

Ήδη έχει αναφερθεί ότι μια εφαρμογή υλοποιημένη σε NesC αποτελείται από ένα σύνολο αρθρωμάτων (components) κατάλληλα συνδεδεμένων, ώστε να αποτελούν ένα λειτουργικό εκτελέσιμο αρχείο. Προχωρώντας σε λεπτομέρειες θα πρέπει να σημειώσουμε ότι στη NesC τα αρθρώματα διαχωρίζονται σε modules και configurations. Ένα module είναι ένα άρθρωμα, που περιέχει την υλοποίηση μιας λειτουργικής μονάδας, η οποία θα συνεισφέρει στη δόμηση της εφαρμογής. Από την άλλη, ένα configuration είναι ένα άρθρωμα που έχει ως κύρια εργασία τη διασύνδεση πολλών αρθρωμάτων με λογική “καλωδίωσης” (wiring), την οποία συναντάμε σε χαμηλού επιπέδου γλώσσες προγραμματισμού (όπως Verilog, VHDL). Κάθε εφαρμογή της NesC προσδιορίζεται από ένα μοναδικό άρθρωμα configuration, το οποίο βρίσκεται στην κορυφή της ιεραρχικής δομής της. Αυτό το configuration υποδεικνύει το σύνολο των επιμέρους αρθρωμάτων και τον τρόπο συνεργασία τους για τη δημιουργία της εφαρμογής.

Ο διαχωρισμός των αρθρωμάτων σε configurations και modules έγινε για να διευκολυνθεί γενικά ο μέσος προγραμματιστής εφαρμογών TinyOS. Έτσι παρέχεται η δυνατότητα σε ένα προγραμματιστή να δημιουργήσει μια εφαρμογή απλά και μόνο παρέχοντας ένα configuration άρθρωμα στο οποίο θα συνδέει υπάρχοντα modules που βρίσκονται στη βιβλιοθήκη του TinyOS. Επιπρόσθετα όμως μπορεί ένας προγραμματιστής να υλοποιεί αυτόνομα modules χωρίς να σχεδιάζει ολοκληρωμένες εφαρμογές, με σκοπό αυτά να ενσωματωθούν και να

επεκτείνουν τη βιβλιοθήκη του TinyOS και να χρησιμοποιηθούν μελλοντικά στη συγκρότηση νέων εφαρμογών.

Η εφαρμογή CntToLedsAndRfm εντάσσεται στην περίπτωση, όπου ο προγραμματιστής παρέχει μόνο ένα configuration και κανένα νέο module στον κώδικα του. Έτσι στον υποκατάλογο της εφαρμογής (/apps/CntToLedsAndRfm) συναντάμε μόνο το αρχείο CntToLedsAndRfm.nc, το οποίο περιέχει τον ορισμό του configuration της εφαρμογής. Τα περιεχόμενα του αρχείου φαίνονται παρακάτω:

```
configuration CntToLedsAndRfm { }

implementation {
    components Main, Counter, IntToLeds, IntToRfm, TimerC;
5
    Main.StdControl -> Counter.StdControl;
    Main.StdControl -> IntToLeds.StdControl;
    Main.StdControl -> IntToRfm.StdControl;
    Counter.Timer -> TimerC.Timer[unique("Timer")];
10   IntToLeds <- Counter.IntOutput;
    Counter.IntOutput -> IntToRfm;
}
```

Έχουμε ήδη κάνει μια νύξη ότι κάθε άρθρωμα έχει δύο “όψεις”, την όψη με τις δηλώσεις και την όψη με την υλοποίηση. Στο αρχείο αυτό βλέπουμε και τις δύο όψεις του configuration CntToLedsAndRfm. Το πρώτο τμήμα κώδικα είναι η δήλωση του configuration CntToLedsAndRfm και συνοψίζεται στην πρώτη γραμμή. Μπορεί κανείς εύκολα να διακρίνει ότι η δήλωση αυτή δεν περιέχει κώδικα. Απλά υπάρχει ώστε να πληροί την προϋπόθεση ύπαρξης δήλωσης της ονομασίας του αρθρώματος. Ας σημειωθεί εδώ ότι μια δήλωση ενός configuration μπορεί να περιέχει κώδικα που ενδέχεται να προσδιορίζει διεπαφές (interfaces) που χρησιμοποιεί ή παρέχει το συνολικό configuration. Η περίπτωση αυτή δε θα μας απασχολήσει εδώ, αλλά παρακάτω θα δούμε τι γίνεται όταν στη δήλωση ενός αρθρώματος προσδιορίζονται χρησιμοποιούμενα και παρεχόμενα interfaces. Στο δεύτερο τμήμα κώδικα που αποτελεί το υπόλοιπο κομμάτι του κώδικα στο αρχείο, παρέχεται η υλοποίηση του configuration, όπως δηλώνει και το keyword **implementation**. Στην υλοποίηση αυτή δηλώνονται αρχικά τα αρθρώματα που θα χρησιμοποιεί άμεσα η εφαρμογή. Πρόκειται για τα αρθρώματα Main, Counter, IntToLeds, IntToRfm και TimerC.

Το άρθρωμα Main είναι ένα άρθρωμα, το οποίο έχουν υποχρέωση να χρησιμοποιούν όλες οι NesC εφαρμογές. Αυτό το άρθρωμα αναλαμβάνει την αρχικοποίηση κάθιε εφαρμογής, μέσω της διεπαφής StdControl που χρησιμοποιεί. Στις αμέσως επόμενες τρεις γραμμές φαίνεται η διασύνδεση με λογική wiring του αρθρώματος Main με κάποια από τα υπόλοιπα αρθρώματα της εφαρμογής. Θυμίζουμε ότι τα αρθρώματα είναι προσπελάσιμα και μπορούν να συνδεθούν μεταξύ τους μόνο μέσα από τις διεπαφές που παρέχουν ή χρησιμοποιούν. Έτσι η γραμμή

```
Main.StdControl -> Counter.StdControl;
```

υποδηλώνει ότι η διεπαφή StdControl που χρησιμοποιείται από το άρθρωμα Main, συνδέεται με την διεπαφή StdControl που παρέχεται από το άρθρωμα Counter. Η φορά του διανύσματος (\rightarrow) υποδεικνύει ποιο άρθρωμα χρησιμοποιεί ένα interface και ποιο άρθρωμα παρέχει το interface. Η διασύνδεση εδώ σημαίνει ότι το άρθρωμα Counter υλοποιεί όλες τις εντολές που παρέχει η διεπαφή StdControl, ώστε να μπορεί να τις χρησιμοποιεί το άρθρωμα Main. Ας σημειωθεί ότι πρέπει πάντα να συνδέονται δύο ομώνυμες διεπαφές των επιμέρους αρθρωμάτων. Ακριβώς επειδή συμβαίνει αυτό μπορούμε χάριν συντομίας να παραλείψουμε το όνομα της διεπαφής στο δεύτερο άρθρωμα που μετέχει στη διασύνδεση, καθώς αυτό έμεσα εννοείται. Έτσι η παραπάνω γραμμή θα μπορούσε να γραφτεί ορθά και ως

Main . StdControl -> Counter ;

Σε αυτές τις τρεις γραμμές γίνεται και μια δεύτερη παρατήρηση που αφορά τη δυνατότητα “οδήγησης” (“fan out”) μιας διεπαφής. Η διεπαφή StdControl του Main, μπορεί να συνδέεται με περισσότερες από μία διεπαφές StdControl που παρέχονται από διαφορετικά ανά δύο αρθρώματα. Αυτό έχει ως αποτέλεσμα τελικά το άρθρωμα Main να προσφέρει μέσω της διεπαφής του StdControl, τις διαδικασίες αρχικοποίησης και ελέγχου πολλαπλών αρθρωμάτων.

Η επόμενη γραμμή που συναντάμε είναι η

Counter . Timer -> TimerC . Timer [unique(“Timer”)] ;

στην οποία διασυνδέεται το άρθρωμα Counter με το άρθρωμα TimerC μέσω της διεπαφής Timer. Το άρθρωμα TimerC προσφέρει την υλοποίηση των εντολών που παρέχει η διεπαφή Timer και έτσι μπορεί να τις χρησιμοποιεί το άρθρωμα Counter. Το τελικό C στην ονομασία του TimerC αποτελεί σύμβαση ονομασίας που γενικά διευκολύνει την ανάγνωση του κώδικα NesC και υποδηλώνει ότι το TimerC είναι configuration άρθρωμα.

Επίσης μπορούμε να δούμε ότι η διεπαφή Timer που υλοποιείται από το άρθρωμα TimerC είναι παραμετροποιήσιμη και επιδέχεται την παράμετρο unique(“Timer”). Αυτό με λίγα λόγια σημαίνει ότι επιθυμούμε ένα συγκεκριμένο στιγμιότυπο της διεπαφής Timer από τα πολλά (συγκεκριμένα 256) που αυτή προσφέρει. Ένας λόγος για να γίνει αυτό είναι γιατί μπορεί να θέλουμε να χρησιμοποιήσουμε ένα μετρητή αποκλειστικά για ένα συγκεκριμένο σκοπό της εφαρμογής μας και ίσως έναν άλλο μετρητή με διαφορετικά χαρακτηριστικά και χρόνους λήξης για ένα άλλο σκοπό. Για να μην έχουμε συγκρούσεις λοιπόν στη χρήση μετρητών από διαφορετικά αρθρώματα της εφαρμογής μας χρησιμοποιούμε διαφορετικά στιγμιότυπα της ίδιας διεπαφής Timer, τα οποία προσδιορίζονται μοναδικά από το όρισμα - συνάρτηση unique(identifier). Αυτή η συνάρτηση εγγυάται την παραγωγή ενός μοναδικού αριθμού που αντιστοιχίζεται στο αλφαριθμητικό identifier κάθε φορά που χρησιμοποιείται. Έτσι συνεχόμενες κλήσεις της συνάρτησης αυτή για το ίδιο όρισμα θα παράγουν διαφορετικούς μεταξύ τους αριθμούς. Γενικά λοιπόν κάθε φορά που θέλουμε να χρησιμοποιήσουμε ένα νέο μετρητή χρησιμοποιούμε το όρισμα unique(“Timer”), ώστε να διασφαλίσουμε την αυτονομία και τη μοναδικότητα του.

Προχωρώντας στις τελευταίες γραμμές

IntToLeds <- Counter . IntOutput ;
Counter . IntOutput -> IntToRfm ;

βλέπουμε ότι το άρθρωμα IntToLeds, όπως και το άρθρωμα IntToRfm, παρέχει τη διεπαφή IntOutput, την οποία χρησιμοποιεί το άρθρωμα Counter. Αυτό το μοτίβο “οδήγησης” είναι ακριβώς αντίστοιχο αυτού που συναντήσαμε προηγούμενα με τη διεπαφή StdControl του αρθρώματος Main.

Μέχρι εδώ, έχουμε ολοκληρώσει την παρουσίαση του κεντρικού αρχείου της εφαρμογής Cnt-ToLedsAndRfm. Αν και ο προγραμματιστής της εφαρμογής αυτής δεν ασχολήθηκε καθόλου με υλοποίηση των επιμέρους αρθρωμάτων που συνέδεσε, θα ακολουθήσει παρακάτω μια συνοπτική παρουσίαση των αρθρωμάτων της βιβλιοθήκης του TinyOS που χρησιμοποιήθηκαν.

Το πρώτο άρθρωμα είναι το Counter που βρίσκεται στο αρχείο /tos/lib/Counter.nc.

```

5    module Counter {
        provides {
            interface StdControl;
        }
5     uses {
            interface Timer;
            interface IntOutput;
        }
} implementation {
10    int state;
15
16    command result_t StdControl.init()
17    {
18        state = 0;
19        return SUCCESS;
20    }
21
22    command result_t StdControl.start()
23    {
24        return call Timer.start(TIMER_REPEAT, 250);
25    }
26
27    command result_t StdControl.stop()
28    {
29        return call Timer.stop();
30    }
31
32    event result_t Timer.fired()
33    {
34        state++;
35        return call IntOutput.output(state);
36    }
37
38    event result_t IntOutput.outputComplete(result_t success)
39    {
40        if(success == 0) state--;
41        return SUCCESS;
42    }
}

```

Στον παραπάνω κώδικα ορίζεται ένα module - άρθρωμα με την ονομασία Counter. Στη δήλωση του module περιέχεται ο προσδιορισμός των διεπαφών που θα χρησιμοποιεί και των διεπαφών που παρέχει το module. Συγκεκριμένα το module **παρέχει** τη διεπαφή StdControl και **χρησιμοποιεί** τις διεπαφές Timer και IntOutput. Αυτό συνεπάγεται ότι στην παρακάτω υλοποίηση του module, πρέπει να υλοποιούνται όλες οι εντολές (commands) που προσφέρει η διεπαφή StdControl, και όλες οι συναρτήσεις χειρισμού συμβάντων (events) των διεπαφών Timer και IntOutput. Ο κώδικας υλοποίησης του module βρίσκεται μέσα στην εμβέλεια των αγκίστρων που ακολουθούν το keyword **implementation**. Ο κώδικας μοιάζει πολύ με C και είναι εύκολα κατανοητός. Είναι φανερό ότι εδώ βρίσκεται η υλοποίηση των εντολών της διεπαφής StdControl, οι οποίες είναι οι init(), start(), stop(). Το πρόθεμα StdControl που ακολουθείται με τελεία (.), είναι ενδεικτικό της διεπαφής στην οποία ανήκει μια εντολή ή ένα event. Τα events που υλοποιούνται στο module είναι τα Timer.fired() και IntOutput.outputComplete().

Η πιο σημαντική λειτουργία του module Counter είναι η αρχικοποίηση του module Timer, που αποτελεί τον χρονομετρητή της εφαρμογής καθώς και ο χειρισμός των συμβάντων λήξης του Timer. Συγκεκριμένα όταν λήγει ο Timer αυξάνεται μια ακέραια μεταβλητή state η οποία κατόπιν χρησιμοποιείται στην κλήση της εντολής IntOutput.output(state). Αυτή η εντολή είναι υπεύθυνη για την απεικόνιση της δυαδικής αναπαράστασης της τιμής του state (modulo 8) στα LEDS ενός mote.

Το δεύτερο άρθρωμα που θα μελετήσουμε είναι το TimerC και βρίσκεται στο /tos/lib/TimerC.nc

```

5 configuration TimerC {
    provides interface Timer[uint8_t id];
    provides interface StdControl;
}

```

```

implementation {
    components TimerM, ClockC, NoLeds;
10   TimerM.Leds -> NoLeds;
    TimerM.Clock -> ClockC;

    StdControl = TimerM;
    Timer = TimerM;
15 }

```

Το TimerC αποτελεί configuration άρθρωμα και στη δήλωση του περικλείει δύο διεπαφές τις οποίες **παρέχει** στα υπόλοιπα αρθρώματα που το χρησιμοποιούν. Αυτές οι διεπαφές είναι η StdControl και η παραμετροποιήσιμη διεπαφή Timer[]|. Στην υλοποίηση του TimerC μπορούμε να διαχρίνουμε τις διασυνδέσεις των επιμέρους αρθρωμάτων και να παρατηρήσουμε ότι οι γραμμές

```

StdControl = TimerM;
Timer = TimerM;

```

σημαίνουν ότι οι παρεχόμενες διεπαφές StdControl και Timer, είναι οι ίδιες με τις παρεχόμενες διεπαφές από το module TimerM και υλοποιούνται σε αυτό. Δεν θα προχωρήσουμε πιο κάτω στην ιεραρχία των αρθρωμάτων για την υλοποίηση του Timer, γιατί δεν είναι αυτός ο σκοπός μας. Ας αναφερθεί, ωστόσο, ότι το module TimerM που παρέχει την κεντρική λειτουργία του Timer, παρέχει στην ουσία την υλοποίηση ενός χρονομετρητή που μπορεί να λήγει και να επαναρχικοποιείται σε τακτικά χρονικά διαστήματα, τα οποία προσδιορίζονται σε milliseconds. Συνεπώς η γραμμή του Counter.nc

```

Timer.start(TIMER_REPEAT, 250);

```

σημαίνει ότι ο χρονομετρητής θα λήγει κάθε 250 *milliseconds* δηλαδή κάθε τέταρτο του δευτερολέπτου.

Ένα άλλο άρθρωμα που χρησιμοποιείται από την εφαρμογή IntToLedsAndRfm είναι το IntToLeds.

```

configuration IntToLeds {
    provides interface IntOutput;
    provides interface StdControl;
}
implementation {
    components IntToLedsM, LedsC;

    IntOutput = IntToLedsM.IntOutput;
    StdControl = IntToLedsM.StdControl;
    IntToLedsM.Leds -> LedsC.Leds;
}

```

Αυτό το άρθρωμα είναι επίσης configuration και **παρέχει** δύο διεπαφές, οι οποίες είναι οι StdControl και IntOutput. Η υλοποίηση των διεπαφών αυτών βρίσκεται στο module IntToLedsM αλλά δε θα μπούμε στις λεπτομέρειές της. Αντίθετα, θα αναφέρουμε μόνο ότι στο module αυτό η υλοποιημένη διεπαφή IntOutput χρησιμεύει στην απεικόνιση της δυαδικής αναπαράστασης των τριών LSB (less significant bits) ενός ακεραίου στα LEDS του mote.

Το τελευταίο άρθρωμα της εφαρμογής IntToLedsAndRfm είναι το IntToRfm και βρίσκεται στο /tos/lib/IntToRfm.nc

```

includes IntMsg;

configuration IntToRfm {
    provides {
        interface IntOutput;
        interface StdControl;
    }
}

```

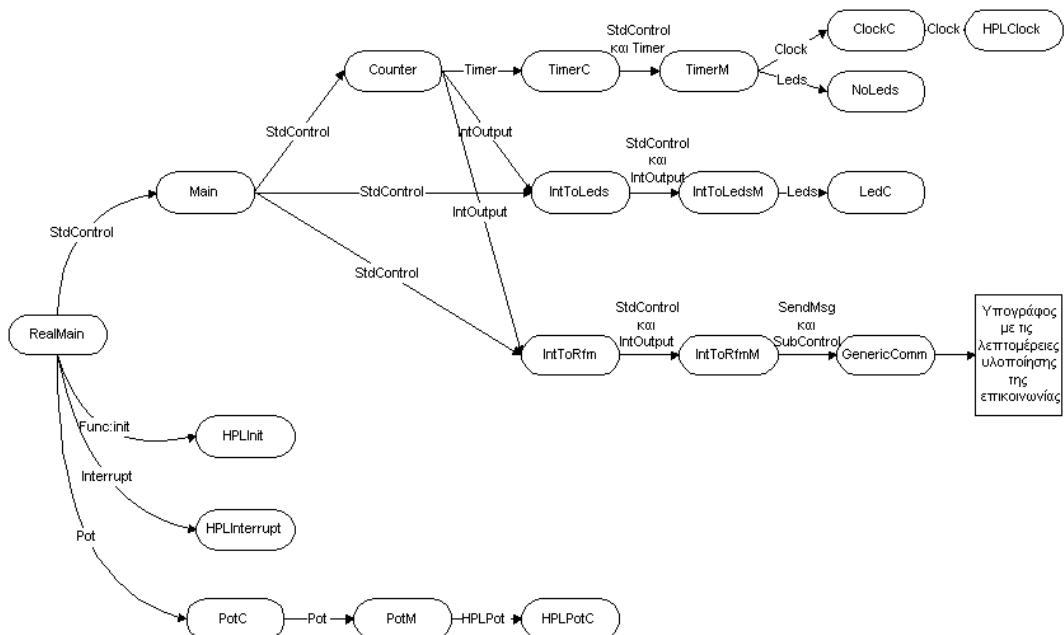
```

10 } implementation {
11     components IntToRfmM, GenericComm as Comm;
12     IntOutput = IntToRfmM;
13     StdControl = IntToRfmM;
14
15     IntToRfmM . Send -> Comm . SendMsg [AM::INTMSG];
16     IntToRfmM . SubControl -> Comm;
17 }
```

Το άρθρωμα αυτό είναι configuration, που **παρέχει** τις διεπαφές IntOutput και StdControl. Αυτές οι διεπαφές υλοποιούνται στο module IntToRfmM.

Το module IntToRfmM χρησιμοποιεί τις διεπαφές SendMsg και SubControl που παρέχονται από το module GenericComm. Χωρίς να παρουσιάσουμε το module IntToRfmM, αναφέρουμε ότι στην υλοποίηση της εντολής IntOutput(int value) η τιμή του ακέραιου ορίσματος ενσωματώνεται σε μια δομή IntMsg, που αποτελεί ειδική δομή για αποστελλόμενα μηνύματα με ακέραιο αριθμό για περιεχόμενο. Για την αποστολή ενός μηνύματος με μέθοδο broadcast σε όλους τους γειτονικούς κόμβους, καλείται η εντολή send που προσφέρει το module GenericComm. Τέλος το module IntToRfmM διαχειρίζεται το συμβάν SendDone της διεπαφής SendMsg, ειδοποιώντας έτσι την εφαρμογή ότι το τρέχον μήνυμα εστάλει και ότι μπορεί να ξεκινήσει η αποστολή νέου μηνύματος.

Στην εικόνα 9.2 φαίνεται η δομή και οι διασυνδέσεις της εφαρμογής CntToLedsAndRfm.



Σχήμα 9.2: Τα αρθρώματα της CntToLedsAndRfm

Κεφάλαιο 10

Μετρήσεις και εφαρμογές δικτύωσης των MICA Motes

Έπειτα από τη θεωρητική παρουσίαση του προγραμματιστικού περιβάλλοντος TinyOS και της γλώσσας NesC, θα πρέπει να παρουσιάσουμε τις εφαρμογές που εκτελέσαμε με τα MICA motes που είχαμε στη διάθεση μας. Πρώτα θα παρουσιάσουμε τον τρόπο προγραμματισμού ενός mote με τη βοήθεια του programming board και ενός H/Y που διαθέτει ελεύθερη παράλληλη θύρα. Στη συνέχεια θα ασχοληθούμε με τη δομή των μηνυμάτων που αποστέλλονται και λαμβάνονται σε ένα δίκτυο από motes, τα οποία έχουν προγραμματιστεί με TinyOS εφαρμογές. Θα περιγράψουμε ένα πείραμα που εκτελέσαμε με ένα MICA mote για να εξάγουμε συμπεράσματα σχετικά με τη διάρκεια ζωής του, καθώς και δύο εφαρμογές που υλοποιήσαμε για να πετύχουμε multihop διάδοση μηνυμάτων σε πραγματικό πειραματικό δίκτυο από MICA motes. Τέλος θα αναφερθούμε στο πολύ χρήσιμο λογισμικό εργαλείο TinyDB, που συνοδεύει την πλατφόρμα του TinyOS, προσφέροντας μια πλειάδα διευκολύνσεων και αξιόπιστων λειτουργιών στον διαχειριστή ενός δικτύου από MICA motes.

10.1 Προγραμματισμός των MICA motes

Στην ενότητα αυτή θα παρουσιάσουμε τον τρόπο προγραμματισμού ενός MICA mote με μία εφαρμογή. Η εφαρμογή που θα χρησιμοποιήσουμε σε αυτό το παράδειγμα είναι μια απλοποιημένη εκδοχή του CntToLedsAndRfm, η οποία ονομάζεται CntToLeds και εκτελεί τις ίδιες λειτουργίες με την CntToLedsAndRfm εκτός από την αποστολή μηνυμάτων σε γειτονικούς κόμβους.

Για να προγραμματίσουμε ένα MICA mote, θα πρέπει να μεταγλωττίσουμε την εφαρμογή CntToLeds, ώστε να παράγουμε δυαδικό κώδικα, ο οποίος έπειτα θα μπορεί να εγγραφεί στον Atmel AVR μικροεπεξεργαστή του mote.

Στον κατάλογο, όπου βρίσκεται μια εφαρμογή περιέχεται πάντοτε ένα Makefile, ένα αρχείο δηλαδή με οδηγίες για τον τρόπο μεταγλώτισης της εφαρμογής. Το αρχείο Makefile για το CntToLeds έχει τη μορφή

```
COMPONENT = CntToLeds
include .. / Makerules
```

Το προσαρτημένο αρχείο Makerules είναι ένα κεντρικό αρχείο που χρησιμοποιείται για τη μετάφραση όλων των εφαρμογών TinyOS και περιέχει παραμέτρους και κανόνες μεταγλώτισης.

Η εντολή μεταγλώτισης δίνεται από τον υποκατάλογο, όπου βρίσκεται η εφαρμογή και είναι η

```
make mica
```

Αυτή η εντολή χρησιμοποιεί όλες τις οδηγίες που περιέχονται στο Makefile και Makerules και αποκρύπτει τις λεπτομέρειες κλήσεως του μεταγλωτιστή NesC (ncc). Βέβαια η έξοδος της εντολής (output) υποδεικνύει βήμα προς βήμα ποιες εντολές εκτελούνται πραγματικά. Έτσι εναλλακτικά μπορεί κανείς να εκτελέσει

```
ncc -o main.exe -target=mica CntToLeds.nc
```

ώστε να μεταγλωτίσει τον κώδικα ASCII της εφαρμογής CntToLeds, ξεκινώντας από το βασικό configuration που βρίσκεται στο αρχείο CntToLeds.nc. Πρέπει να σημειωθεί εδώ ότι εκτελώντας την εντολή μεταγλώτισης για το αρχείο της εφαρμογής που είναι στο υψηλότερο ιεραρχικά επίπεδο του γράφου εκτέλεσης και περιέχει το configuration της εφαρμογής, η NesC αναλαμβάνει τον εντοπισμό και τη σύνδεση όλων των διαφόρων αρθρωμάτων που απαιτεί η εφαρμογή, εξασφαλίζοντας ότι στο τελικό εκτελέσιμο όλα τα αρθρώματα θα συνεργάζονται όπως υπαγορεύει η εφαρμογή. Το αρχείο main.exe που παράγεται είναι δυαδικό και εκτελέσιμο από το Mica mote.

Πριν όμως μεταφέρουμε τον κώδικα στο Mica mote, θα πρέπει να εκτελέσουμε και την εντολή

```
avr-objcopy --output-target=srec main.exe main.srec
```

για να δημιουργηθεί ένα αρχείο main.srec που αποτελεί την αναπαράσταση του δυαδικού main.exe σε μορφή κειμένου (text), το οποίο μπορεί έπειτα να χρησιμοποιηθεί στη μεταφόρτωση του κώδικα στο mote. Ωστόσο, δε θα χρειαστεί σχεδόν ποτέ να εκτελούμε μια προς μία τις παραπάνω εντολές χειρωνακτικά, παρά μόνο σε ελάχιστες περιπτώσεις αποσφαλμάτωσης (debugging) της διαδικασίας προγραμματισμού. Για τη συνήθη περίπτωση αρκεί όπως προαναφέρθηκε η εντολή make mica.

Αφού έχουμε κάνει τη μεταγλώτιση του κώδικα και έχουμε διαπιστώσει ότι δεν προέκυψαν σφάλματα στον κώδικα μπορούμε να προχωρήσουμε στη φάση της μεταφόρτωσης του κώδικα της εφαρμογής στο mica mote. Στη φάση αυτή πρέπει πρώτα να συνδέσουμε το mica mote πάνω στην υποδοχή του programming board. Επειτα χρειάζεται να τροφοδοτήσουμε το programming board με ρεύμα και αυτό μπορεί να γίνει είτε από μια πηγή ρεύματος 3 V συνδεδεμένη σε ειδική υποδοχή του programming board ή μέσω των μπαταριών του συνδεδεμένου mica mote, δεδομένου ότι το mica mote είναι ενεργό (power switch on). Το κόκκινο Led του programming board υποδεικνύει την ύπαρξη ή όχι ικανής τροφοδοσίας για τη λειτουργία του board.

Στη συνέχεια μπορούμε να συνδέσουμε το programming board μέσω καλωδίου ή απευθείας, με την παράλληλη θύρα του H/Y, όπου βρίσκεται ο μεταγλωτισμένος κώδικας της εφαρμογής μας και είναι εγκατεστημένο το περιβάλλον ανάπτυξης εφαρμογών TinyOS. Ενεργοποιούμε επίσης αν δε τέχνεις ήδη κάνει, το mica mote, θέτοντας το διακόπτη λειτουργίας

του στη θέση ON. Ας σημειωθεί εδώ ότι το μήκος του καλωδίου διασύνδεσης με την παράλληλη θύρα του H/Y πρέπει να είναι μικρότερο από 1.5 m, γιατί στην αντίθετη περίπτωση προκύπτουν λάθη κατά τη μεταφορά του εκτελέσιμου κώδικα στο mote.

Από τον υποκατάλογο όπου βρίσκεται η εφαρμογή μας εκτελούμε την εντολή

```
make mica install
```

η οποία για κάποιο λόγο εμπειρέχει και ξαναεκτελεί τη διαδικασία μεταγλώττισης της εφαρμογής αλλά επιπρόσθετα εκτελεί αυτόματα τη διαδικασία μεταφόρτωσης του κώδικα στο mica mote. Εναλλακτικά μπορούμε να εκτελέσουμε την εντολή

```
make mica install.xx
```

Όπου xx μπορεί να είναι ένας ακέραιος αριθμός. Η εντολή αυτή έχει τα ίδια αποτελέσματα με την προηγούμενη, αλλά επιπρόσθετα προγραμματίζει στο mote, ώστε να έχει ταυτότητα (ID) στο δίκτυο ίση με την τιμή xx. Αυτό εξυπηρετεί σκοπούς ταυτοποίησης του αποστολέα ενός μηνύματος σε εφαρμογές που ενέχουν επικοινωνία RF.

Αν κατά τη διαδικασία μεταφόρτωσης του εκτελέσιμου κώδικα στο mote προκύψει κάποιο σφάλμα, αυτό εμφανίζεται στην οθόνη του H/Y. Συνήθως ένα σφάλμα στην διαδικασία μεταφόρτωσης σημαίνει ότι το mica mote είναι ανενεργό, ή το programming board δεν δέχεται επαρκή τροφοδοσία ή ότι υπάρχει πρόβλημα με το καλώδιο διασύνδεσης με την παράλληλη θύρα του H/Y. Επίσης μπορεί να σημαίνει ότι το εργαλείο uisp που χρησιμοποιείται για την μεταφόρτωση του εκτελέσιμου στο mote, δεν είναι στη σωστή έκδοση και θα πρέπει να ενημερωθεί. Ο επιτυχής προγραμματισμός του mote συνοδεύεται συνήθως από ένα μήνυμα στην οθόνη του H/Y παρόμοιο με το ακόλουθο :

```
compiling CntToLeds to a mica binary ncc -board=micasb -o
build/mica/main.exe -Os -target=mica -Wall -Wshadow
-DDEF_TOS_AM_GROUP=0x7d -finline-limit=200
-fnesc-cfile=build/mica/app.c CntToLeds.nc -lm
avr-objcopy --output-target=srec build/mica/main.exe
build/mica/main.srec
compiled CntToLeds to build/mica/main.srec
installing mica binary
uisp -dprog=dapa --erase
pulse
Atmel AVR ATmega128 is found.
Erasing device ...
pulse
Reinitializing device
Atmel AVR ATmega128 is found.
sleep 1
uisp -dprog=dapa --upload if=build/mica/main.srec
pulse
Atmel AVR ATmega128 is found.
Uploading: flash
sleep 1
uisp -dprog=dapa --verify if=build/mica/main.srec
```

```
pulse
Atmel AVR ATmega128 is found.
Verifying: flash
```

Βέβαια ο πιο πρακτικός τρόπος για να επιβεβαιώσουμε ότι το mote έχει προγραμματιστεί σωστά με την εφαρμογή CntToLeds είναι να αποσυνδέσουμε το mote από το programming board και, αφού βεβαιωθούμε ότι είναι ενεργοποιημένο, να διαπιστώσουμε ότι λειτουργεί με τον επιθυμητό τρόπο. Στην προκειμένη περίπτωση θα πρέπει οι ενδείξεις των LEDs του mote να σχηματίζουν διαδοχικά τη διαδική αναπαράσταση των αριθμών από 0 έως 7 και έπειτα πάλι από την αρχή, με συχνότητα αλλαγής 4 Hz. Ας σημειωθεί τέλος ότι για να διαγράψουμε όλα τα αρχεία που δημιουργήθηκαν από τη μεταγλώττιση της εφαρμογής μπορούμε από τον υποκατάλογο του CntToLeds να εκτελέσουμε την εντολή

```
make clean
```

Επίσης για τον επαναπρογραμματισμό του mote δε χρειάζεται να διαγράψουμε “χειρωνακτικά” το προηγούμενο εκτελέσιμο από τον μικροεπεξεργαστή καιώς αυτό αποτελεί μέρος της διαδικασίας μεταφόρτωσης, που εκτελείται με την εντολή make mica install.

10.1.1 Ρύθμιση ακτίνας εκπομπής

Οι δυνατότητες επικοινωνίας των mote, μέσω της ενσωματωμένης κεραίας που διαθέτουν είναι αρκετά περιορισμένες. Ωστόσο η ακτίνα αξιόπιστης μετάδοσης μηνυμάτων μπορεί να προγραμματιστεί και να τεθεί ανάλογα με τις ανάγκες της κάθισε εφαρμογής. Η ακτίνα διάδοσης ενός μηνύματος με τη βοήθεια της ενσωματωμένης κεραίας μπορεί να κυμαίνεται από ένα εκατοστό μέχρι περίπου δύο μέτρα, ενώ καλό θα είναι να απουσιάζουν τυχόν εμπόδια μεταξύ των επικοινωνούντων κόμβων.

Στο κύκλωμα του πομποδέκτη υπάρχει ένα ποτενσιόμετρο του οποίου η εμπέδηση μπορεί και ελέγχεται από το λογισμικό. Η τιμή της εμπέδησης αυτής καθορίζει την ισχύ της εκπομπής μηνυμάτων από το κύκλωμα RFM TR1000. Οι επιτρεπτές τιμές για την εμπέδηση του ποτενσιόμετρου δίνονται κανονικοποιημένες σε κλίμακα από 0 έως 100, ενώ επίσης στην τεκμηρίωση του TinyOS αναφέρεται ότι η εμβέλεια των μεταδόσεων δεν μεταβάλλεται απαραιτητά γραμμικά με την αντίσταση του ποτενσιόμετρου. Σε κάθε περίπτωση η μικρότερη τιμή της αντίστασης (0) αντιστοιχεί στη μέγιστη δυνατή ακτίνα μετάδοσης, ενώ η μέγιστη τιμή (100) αντιστοιχεί σε ελάχιστη ακτίνα μετάδοσης, η οποία δεν υπερβαίνει το ένα εκατοστό.

Προκειμένου να προγραμματίσουμε το ποτενσιόμετρο για τις ανάγκες μιας συγκεκριμένης εφαρμογής, θα πρέπει να ενσωματώσουμε στην εκάστοτε εφαρμογή τη διαδικασία αρχικοποίησης του ποτενσιόμετρου σε συγκεκριμένη τιμή.

Αυτό γίνεται ακολουθώντας τα παρακάτω απλά βήματα, τα οποία για να γίνουν περισσότερο κατανοητά τα εφαρμόζουμε στη γνωστή μας εφαρμογή CntToLedsAndRfm.

Αρχικά στο βασικό αρχείο, που περιέχει το ανώτερο iεραρχικά configuration της εφαρμογής, συμπεριλαμβάνουμε ανάμεσα στα αρθρώματα που συνδέονται, το άρθρωμα βιβλιοθήκης PotC. Έπειτα εντοπίζουμε ένα (οποιοδήποτε) άρθρωμα που συμμετέχει στο configuration και παρέχει μια υλοποίηση της διεπαφής StdControl στο module main. Γι' αυτό το άρθρωμα ορίζουμε ότι θα χρησιμοποιεί τη διεπαφή βιβλιοθήκης Pot της οποίας η υλοποίηση

βρίσκεται στο module PotC. Έτσι το νέο αρχείο με το configuration για την εφαρμογή CntToLedsAndRfm θα είναι το παρακάτω :

```

configuration CntToLedsAndRfm {
    implementation {
        components Main, Counter, IntToLeds, IntToRfm, TimerC, PotC;

        5      Main.StdControl -> Counter.StdControl;
        Main.StdControl -> IntToLeds.StdControl;
        Main.StdControl -> IntToRfm.StdControl;
        Counter.Timer -> TimerC.Timer[unique("Timer")];
        IntToLeds <- Counter.IntOutput;
        Counter.IntOutput -> IntToRfm;
        IntToRfm.Pot -> PotC;
    }
}

```

Από τον παραπάνω κώδικα είναι φανερό ότι έχουμε επιλέξει το configuration IntToRfm ως άρθρωμα που χρησιμοποιεί τη διεπαφή Pot. Γενικά καλό θα είναι να επιλέγουμε ένα module και όχι ένα configuration για να χρησιμοποιεί τη διεπαφή Pot, γιατί με την επιλογή ενός configuration θα πρέπει να “κατέβουμε” τα επίπεδα ιεραρχίας στη δόμηση της εφαρμογής μέχρι να συναντήσουμε ένα module. Ωστόσο, ελλείψει modules, θα πρέπει υποχρεωτικά να διαλέξουμε ένα configuration. Έτσι στη συνέχεια πρέπει να αλλάξουμε το αρχείο που περιέχει τον ορισμό του επιλεγμένου configuration. Ανοίγουμε λοιπόν το αρχείο /tos/lib/IntToRfm.nc και το τροποποιούμε, ώστε

- α) το configuration IntToRfm να μπορεί να χρησιμοποιεί τη διεπαφή Pot, και
- β) να ανακατευθύνουμε τη χρήση της διεπαφής Pot στο module IntToRfmM, το οποίο - είναι σημαντικό να αναφερθεί ότι - παρέχει την υλοποίηση της διεπαφής StdControl.

```

includes IntMsg;

configuration IntToRfm {
    provides {
        interface IntOutput;
        interface StdControl;
    }
    uses {
        interface Pot;
    }
}
implementation {
    components IntToRfmM, GenericComm as Comm;

    5      IntOutput = IntToRfmM;
    StdControl = IntToRfmM;
    Pot = IntToRfmM;

    IntToRfmM.Send -> Comm.SendMsg[AM_INTMMSG];
    10   IntToRfmM.SubControl -> Comm;
}

```

Στο τελευταίο βήμα θα τροποποιήσουμε το αρχείο /tos/lib/IntToRfmM.nc που περιέχει τον ορισμό και την υλοποίηση του module IntToRfmM. Η διαδικασία που ακολουθεί, είναι επίσης και αυτή που θα εφαρμόζαμε αν από την αρχή έχαμε επιλέξει ένα module για να χρησιμοποιεί τη διεπαφή Pot. Οι προσθήκες που κάνουμε στο αρχείο IntToRfmM.nc περιλαμβάνουν

- α) μία δήλωση ότι το module χρησιμοποιεί τη διεπαφή Pot και
- β) μια κλήση της συνάρτησης Pot.init(0) μέσα στην εμβέλεια της υλοποίησης της StdControl.init(). Στη θέση της τιμής μηδέν (0) μπορούμε να έχουμε οποιαδήποτε τιμή από 0 έως 100, αν και προτιμήσαμε το 0 για να διαπιστώσουμε στην πράξη το μήκος της μέγιστης ακτίνας μετάδοσης του mote.

Μετά από το τελευταίο βήμα προγραμματίζουμε ένα MICA mote με την νέα εφαρμογή Cnt-ToLedsAndRfm και ένα δεύτερο MICA mote με την εφαρμογή RfmToLeds, η οποία επιτελεί τη λειτουργία λήψης μηνυμάτων και της δυαδικής αναπαράστασης των δεδομένων τους στα Leds του MICA mote. Αν ο προγραμματισμός έχει γίνει σωστά, τότε όταν τα δύο motes είναι σε λειτουργία και σε απόσταση μεταξύ τους που δεν υπερβαίνει τα δύο μέτρα, το mote δέκτης θα έχει τις ίδιες ενδείξεις ταυτόχρονα με το mote πομπό.

10.2 Δομή μηνυμάτων στο TinyOS και η εφαρμογή OscilloscopeRf

Πρέπει εδώ να αναφέρουμε κάποια στοιχεία για τη δομή ενός μηνύματος που αποστέλλεται με τη μέθοδο RF στις εφαρμογές του TinyOS. Γενικά το TinyOS υπαγορεύει ότι όλα τα αποστελλόμενα μηνύματα θα ακολουθούν μια γενικευμένη δομή (struct), η οποία ονομάζεται TOS_Msg. Η δομή αυτή περιγράφεται στο αρχείο /tos/system/AM.h και ο τύπος μηνύματος, του οποίου τη δομή καθορίζει, ονομάζεται συμβατικά active message (AM). Η δομή αυτή ανάμεσα σε άλλα περιέχει τα παρακάτω βασικά πεδία

Πεδίο	Τύπος	Μέγεθος
Διεύθυνση παραλήπτη	int16_t	2 bytes
Τύπος μηνύματος	int8_t	1 byte
Ταυτότητα group	int8_t	1 byte
Μήκος δεδομένων	int8_t	1 byte
Buffer δεδομένων	Πίνακας από 29 θέσεις int8_t	29 bytes

Τα υπόλοιπα πεδία της δομής απλά ενημερώνονται και κρατούν πληροφορίες εσωτερικών ελέγχων και δε θα μας απασχολήσουν. Τα παραπάνω πεδία, ωστόσο, είναι σημαντικά, γιατί τα τέσσερα πρώτα αποστέλλονται σε κάθε μήνυμα ως επικεφαλίδα, ενώ από το πέμπτο το οποίο είναι ο buffer με τα δεδομένα του μηνύματος, μεταδίδονται μόνο τόσα bytes όσα ορίζει το τέταρτο πεδίο που είναι το “μήκος των δεδομένων”.

Το πρώτο πεδίο της δομής TOS_Msg είναι μια τιμή των 16 bit και αναπαριστά τη διεύθυνση του προορισμού για το μήνυμα. Αυτή η διεύθυνση μπορεί να είναι μία από τις προκαθορισμένες τιμές TOS_BCAST_ADDR ή TOS_UART_ADDR, καθορίζοντας έτσι ότι το μήνυμα θα διαδοθεί με broadcast από το ραδιοπομπό ή θα μεταδοθεί στη σειριακή θύρα ενός H/Y αντίστοιχα. Επίσης μπορεί να είναι απλά η διεύθυνση ενός μόνο mote. Έχουμε ήδη αναφέρει και θα το επαναλάβουμε εδώ ότι ένα mote μπορεί να αποκτήσει μοναδική ταυτότητα η οποία λειτουργεί ως τοπική διεύθυνση (local address) αν εκτελέσουμε την εντολή

```
Make mica install.xx
```

κατα τη διαδικασία μεταφόρτωσης του κώδικα μιας εφαρμογής στο mote. Τότε ο κόμβος θα αποκτήσει τη διεύθυνση xx, όπου xx μπορεί να είναι ένας ακέραιος 16 bit. Για παράδειγμα η εντολή

```
make mica install.01
```

θα προγραμματίσει ένα mote με μία εφαρμογή, από τον κατάλογο της οποίας εκτελέσαμε την εντολή, και επιπρόσθετα θα αποδώσει τη διεύθυνση 01 στο mote. Έτσι θα μπορούν

όλοι οι γείτονες του να επικοινωνούν μόνο με αυτό δηλώνοντας τη διεύθυνση 01 στο πεδίο “διεύθυνση παραλήπτη” της δομής TOS_Msg.

Το δεύτερο πεδίο της δομής TOS_Msg, είναι ο τύπος του μηνύματος και καθορίζεται από την κάθε εφαρμογή όπως θα δούμε παρακάτω. Η χρησιμότητα του έγκειται στην ορθή αναγνώριση του τύπου ενός μηνύματος από τον κόμβο παραλήπτη, ώστε να μπορεί να εκτελέσει τις κατάλληλες ενέργειες χειρισμού και επεξεργασίας του.

Το τρίτο πεδίο, είναι η ταυτότητα του group (group ID) στο οποίο απευθύνεται το μήνυμα. Στο TinyOS επιτρέπεται να υπάρχουν πολυάριθμα σύνολα (groups) από motes, και σε κάθε group οι κόμβοι του μοιράζονται το ίδιο κανάλι RF επικοινωνίας. Ο διαχωρισμός των motes σε groups γίνεται για να απομονωθεί η επικοινωνία των motes μεταξύ μόνο των συσκευών που ανήκουν στο ίδιο group, ενώ τα motes που ανήκουν σε διαφορετικά groups δε θα μπορούν να προκαλέσουν παρεμβολές στην επικοινωνία. Ένα mote λοιπόν που ανήκει σε ένα group δε μπορεί να λάβει μηνύματα από motes ενός άλλου group παρά μόνο από το δικό του group. Η ταυτότητα ενός group είναι ένας αριθμός 8-bit και πρέπει να προγραμματιστεί σε κάθε mote που θα ανήκει στο group αυτό. Αυτό γίνεται με την προσθήκη της γραμμής

`DEFAULT\LOCAL\GROUP = 0x42`

στο αρχείο Makefile, που βρίσκεται στον κατάλογο της εκάστοτε εφαρμογής με την οποία σκοπεύουμε να προγραμματίσουμε το mote. Ο αριθμός 0x42 είναι η ταυτότητα του group και φυσικά είναι ενδεικτικός, καθώς θα μπορούσε να είναι οποιοσδήποτε αριθμός μεταξύ 0x00 και 0xFF. Το πρόθεμα 0x υποδεικνύει ότι ακολουθεί δεκαεξαδικός αριθμός. Η παραπάνω γραμμή πρέπει να προηγείται της γραμμής που υπαγορεύει τη σύναψη του αρχείου Makerules. Όπως είναι λογικό, στο αρχείο Makefile η παραπάνω γραμμή πρέπει να διατηρηθεί απείραχτη κατά τον προγραμματισμό όλων των motes ενός δεδομένου group, ενώ θα αλλάξει αν θέλουμε να προγραμματίσουμε με την ίδια εφαρμογή τα motes ενός άλλου group.

Η κάθε εφαρμογή στο TinyOS που απαιτεί αποστολή δεδομένων, συνήθως προτείνει και χρησιμοποιεί μια ειδική δομή για τα μηνύματα της. Ως παράδειγμα παραθέτουμε τα μηνύματα της εφαρμογής OscilloscopeRf. Η εφαρμογή αυτή, η οποία παρέχεται μαζί με το περιβάλλον ανάπτυξης εφαρμογών του TinyOS, λαμβάνει δείγματα ανά σταύρωση χρονικά διαστήματα από ένα αισθητήρα και παράλληλα αποστέλλει ένα συλλογικό μήνυμα για κάθε 10 δείγματα που έλαβε.

Η ειδική δομή μηνύματος που φαίνεται να χρησιμοποιεί το OscilloscopeRf ονομάζεται OscilloscopeMsg και ορίζεται στο αρχείο /tos/lib/OscilloscopeMsg.h, το οποίο είναι αρχείο επικεφαλίδας (header) και επισυνάπτεται με εντολή include στο αρχείο υλοποίησης του module OscilloscopeM. Τα πεδία λοιπόν ενός OscilloscopeMsg είναι τα παρακάτω:

Πεδίο	Τύπος	Μέγεθος
Ταυτότητα mote	uint16_t	2 bytes
Αύξων αριθμός του πιο πρόσφατου δείγματος	uint16_t	2 bytes
Κανάλι ADC λήψης δειγμάτων group	uint16_t	2 bytes
Buffer με 10 δείγματα	Πίνακας 10 στοιχείων από uint16_t	20 bytes

Αυτή η δομή είναι εμφανές ότι έχει συνολικό μέγεθος 26 bytes. Προτού η εφαρμογή Oscillo-

scopeRf καλέσει την εντολή αποστολής ενός μηνύματος, φροντίζει να ενθυλαχώσει την ειδική δομή του μηνύματος της στο buffer δεδομένων του γενικευμένου τύπου TOS_Msg. Με άλλα λόγια το πέμπτο πεδίο του TOS_Msg που έχει μήκος 29 bytes, ενημερώνεται με τα περιεχόμενα των πεδίων της δομής OscopeMsg. Η NesC διευκολύνει αρκετά αυτή τη διαδικασία ενθυλάκωσης, χρησιμοποιώντας τους γνωστούς από τη C μηχανισμούς υπονοούμενης μετατροπής τύπων(casting). Παρακάτω φαίνονται χαρακτηριστικά αποσπάσματα από τον κώδικα για το module OscilloscopeM, που υλοποιεί τη βασική λειτουργικότητα της εφαρμογής.

```

TOS_Msg msg[2]; //orism'os d'uodeikt'wnsem'hnuma      TOS_Msg•
event result_t ADC.dataReady(uint16_t data) {
    struct OscopeMsg *pack = (struct OscopeMsg *)msg[currentMsg].data;
    pack->data[packetReadingNumber++] = data;
    readingNumber++;

10   /*Met'a ap'oimetr'hseis 10 */
    if (packetReadingNumber == BUFFER_SIZE)
    {
        packetReadingNumber = 0;
        pack->channel = 1;
15       pack->lastSampleNumber = readingNumber;
        pack->sourceMoteID = TOS_LOCAL_ADDRESS;

20   //Apostol'h en'ospak'etou
    if (call DataMsg.send(TOS_BCAST_ADDR, sizeof(struct OscopeMsg),
                          &msg[currentMsg]))
    {
        .
        .
    }
25   .
}

```

Στον παραπάνω κώδικα φαίνεται ότι το πέμπτο πεδίο (data) της δομής TOS_Msg, η οποία εδώ έχει το όνομα msg[currentMsg], μετατρέπεται με casting σε δείκτη σε μια δομή OscopeMsg, ο οποίος φέρει το όνομα pack. Στη συνέχεια το πέμπτο πεδίο (data) της δομής TOS_Msg ενημερώνεται με ευανάγνωστο τρόπο, απλά ενημερώνοντας τα πεδία της δομής OscopeMsg, που δεικτούνται ο δείκτης pack.

Κάθε εφαρμογή που θέλει να στείλει και να λάβει μηνύματα συνήθως περιέχει στο κεντρικό configuration της μια διασύνδεση με το module GenericComm. Το module αυτό παρέχει, ανάμεσα σε άλλες, τη διεπαφή SendMsg για την αποστολή μηνυμάτων και τη διεπαφή ReceiveMsg για τη λήψη μηνυμάτων. Και οι δύο αυτές διεπαφές είναι παραμετροποιήσιμες και δέχονται ως όρισμα τον τύπο του κάθε μηνύματος ο οποίος διαφέρει ανάλογα με την εφαρμογή. Στις τελευταίες γραμμές του κώδικα για το module OscilloscopeM, που επισυνάψαμε, φαίνεται η κλήση

Call DataMsg.send(TOS_BCAST_ADDR, sizeof(struct OscopeMsg),
&msg[currentMsg]))

όπου το DataMsg είναι ψευδώνυμο (alias) για τη διεπαφή SendMsg και send() είναι μια από τις παρεχόμενες εντολές αυτής της διεπαφής. Τα ορίσματα της send δίνουν, με τη σειρά που τα διαβάζουμε, πρώτα τη διεύθυνση του παραλήπτη, ο οποίος εδώ είναι όλα τα γειτονικά motes (broadcast address) και έπειτα το μέγεθος της ειδικής δομής OscopeMsg, το οποίο όπως είδαμε όταν αποθηκευτεί στην επικεφαλίδα του TOS_Msg ως τέταρτο πεδίο και όταν καθορίσει πόσα bytes από το πέμπτο πεδίο όταν αποσταλούν. Το τελευταίο όρισμα της send() είναι ένας δείκτης στη βασική δομή μηνύματος TOS_Msg. Με αυτό τον απλό τρόπο ένα μήνυμα που μεταφέρει πληροφορία ειδικά δομημένη για τις ανάγκες μίας εφαρμογής, αποστέλλεται στο δίκτυο.

Ας σημειωθεί τέλος, αν και αυτό δεν αποτελεί τμήμα από το παράδειγμα της εφαρμογής OscilloscopeRf, ότι μια εφαρμογή που θα πραγματοποιεί τη λήψη μηνυμάτων θα πρέπει πάλι στο κεντρικό configuration της να περιέχει μια διασύνδεση με το module GenericComm. Η λήψη ενός μηνύματος πραγματοποιείται με τη αλήση της εντολής receive(TOS_MsgPtr) που παρέχει η διεπαφή ReceiveMsg. Η συνάρτηση receive() είναι συνάρτηση χειρισμού του γεγονότος λήψης δεδομένων από τον πομποδέκτη. Το όρισμα που δέχεται είναι ένας δείκτης σε δομή TOS_Msg, ο οποίος δεικτοδοτεί τη δομή του λαμβανόμενου μηνύματος. Έτσι είναι δυνατόν από αυτή τη δομή με τεχνική casting πάνω στο πέμπτο πεδίο της (data) να αποσπαστεί η χρήσιμη για την εφαρμογή πληροφορία του μηνύματος.

10.3 Διάρκεια ζωής ενός MICa mote

Στα πλαίσια του ελέγχου των δυνατοτήτων και των αντοχών ενός MICa mote, πραγματοποιήσαμε ένα πείραμα για να προσδιορίσουμε το χρόνο ζωής ενός mote. Ως χρόνο ζωής ενός mote εννοούμε το χρονικό διάστημα που ορίζεται από την έναρξη της λειτουργίας ενός mote, στο οποίο εκτελείται μια δεδομένη εφαρμογή, μέχρι την εξάντληση των ενεργειακών αποθεμάτων του mote με άμεσο αποτέλεσμα τη δυσλειτουργία της εφαρμογής που αυτό εκτελεί.

Για τον προγραμματισμό του mote, αποφασίσαμε να επιλέξουμε μια εφαρμογή η οποία θα απαιτούσε τόσο τη λειτουργία των ανιχνευτών του MICa mote, όσο και τη τακτική χρήση του πομποδέκτη για αποστολή μηνυμάτων. Αποφύγαμε τις εφαρμογές που εκτελούν απλές λειτουργίες, γιατί θέλαμε να διαπιστώσουμε την αντοχή του mote σε μια ενεργειακά, υπολογιστικά και επικοινωνιακά απαιτητική εφαρμογή. Αυτή η εφαρμογή προέκυψε από μια τροποποίηση της OscilloscopeRf, η οποία προσφέρεται από την πλατφόρμα ανάπτυξης εφαρμογών του TinyOS.

Η τροποποιημένη εφαρμογή OscilloscopeRf λαμβάνει τακτικά μετρήσεις από τον αισθητήρα φωτός, για τα επίπεδα φωτεινότητας στο περιβάλλον ενός mote. Οι μετρήσεις αυτές ομαδοποιούνται ανά 10 σε μηνύματα, τα οποία στη συνέχεια μεταδίδονται με τεχνική broadcast σε όλους τους γειτονικούς κόμβους του δικτύου. Κάθε μέτρηση του αισθητήρα αναπαριστάται με ένα αριθμό των 10 bit, ο οποίος αποθηκεύεται σε δύο bytes ($2 \times 8 = 16$ bit, με 6 πλεονάζοντα bit). Ο ρυθμός που λαμβάνονται τα δείγματα από τον αισθητήρα φωτός τέθηκε στα 32 Hz (ενώ στην αρχική υλοποίηση του OscilloscopeRf ήταν στα 8 Hz), δηλαδή είχαμε λήψη 32 δειγμάτων ανά δευτερόλεπτο. Αυτό σημαίνει επίσης ότι, καθώς ανά 10 δείγματα έχουμε ένα νέο μήνυμα, σε διάρκεια 5 δευτερόλεπτων αντιστοιχούν 16 νέες μεταδόσεις μηνυμάτων.

Επιπρόσθετα με κάθε αποστολή ενός μηνύματος το κίτρινο LED του mote αλλάζει κατάσταση, δηλαδή αν είναι σβηστό τότε θα ανάψει και το αντίθετο. Τέλος, όταν μια μέτρηση έχει τιμή μεγαλύτερη από 0x300 (δεκαεξαδικό), αυτό σημαίνει ότι το περιβάλλον του mote είναι αρκετά φωτεινό και ως ένδειξη αυτής της διαπίστωσης το mote ανάβει το κόκκινο LED που διαθέτει. Αν η τιμή ενός δείγματος πέσει κάτω από 0x300, τότε το κόκκινο LED θα σβήσει.

Για το πείραμα χρησιμοποιήσαμε ένα mote, εφοδιασμένο με δύο καινούργιες και φορτισμένες μπαταρίες AA των 1.5 Volt. Προγραμματίσαμε το mote με την τροποποιημένη εφαρμογή OscilloscopeRf, στην οποία αρχικοποιήσαμε το ποτενσιόμετρο του πομποδέκτη με τον τρόπο που προαναφέρθηκε, ώστε να αποστέλλει μηνύματα στη μέγιστη δυνατή απόσταση. Έπειτα σε αυτό προσαρτήθηκε μια πλακέτα MTS310 με αισθητήρες για να λαμβάνει μετρήσεις

από τον αισθητήρα φωτός και το αφήσαμε ανοικτό, σημειώνοντας το χρόνο έναρξης της λειτουργίας του.

Για να υλοποιήσουμε ένα εικονικό κέντρο ελέγχου, με σκοπό αυτό να λαμβάνει τα μηνύματα του mote και να υποδεικνύει την ορθή λειτουργία της εφαρμογής OscilloscopeRF χρειάστηκε να εκτελέσουμε τα ακόλουθα: Προγραμματίσαμε ένα δεύτερο mote με την εφαρμογή GenericBase του TinyOS η οποία υπαγορεύει στο mote την προώθηση των μηνυμάτων που δέχεται από το δέκτη του, στη σειριακή θύρα του programming board. Το mote με την GenericBase εφαρμογή παρέμεινε προσκολλημένο στο programming board, ενώ με ένα σειριακό καλώδιο συνδέσαμε τη σειριακή έξοδο του board με τη σειριακή θύρα COM1 ενός H/Y. Και αυτό το δεύτερο mote διέθετε δύο καινούργιες AA μπαταρίες.



Σχήμα 10.1: Το mote με το OscilloscopeRF και το εικονικό κέντρο ελέγχου (mote πάνω στο programming board)

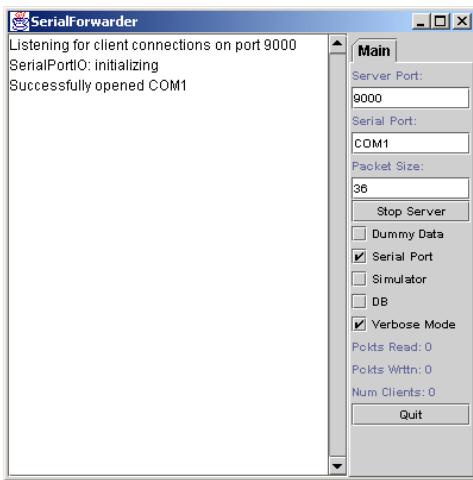
Στον H/Y με τον οποίο συνδέσαμε το programming board, χρειάστηκε να εκτελέσουμε δύο εφαρμογές γραμμένες σε java. Η πρώτη ονομάζεται SerialForward και η βασική εργασία της είναι να λαμβάνει πακέτα μηνυμάτων από τη σειριακή θύρα και να τα προωθεί σε κάποιο port του H/Y, ώστε αυτά να μπορούν να ληφθούν από οποιαδήποτε κατάλληλη εφαρμογή “ακούει” σε αυτό το port, η οποία μπορεί να εκτελείται είτε στο ίδιο υπολογιστή ή να εκτελείται σε διαφορετικό υπολογιστή και να συνδέεται στο συγκεκριμένο port μέσω δικτύου. Μάλιστα είναι δυνατόν πολλαπλές εφαρμογές να “ακούν” τα ίδια πακέτα μηνυμάτων από το ίδιο προκαθορισμένο port καθώς δεν υπάρχει κανένας περιορισμός στον αριθμό συνδέσεων εφαρμογών σε ένα port.

Για να εκτελέσουμε το SerialForward πρόγραμμα βρεθήκαμε στον υποκατάλογο tools/java/

και εκτελέσαμε την εντολή

java net.tinyos.sf.SerialForward &

όπου το τελευταίο & σημαίνει ότι θέσαμε την εφαρμογή να τρέχει “αυθόρυβα” στο background. Η επιτυχής εκτέλεση του SerialForward θα εμφανίσει ένα γραφικό περιβάλλον (GUI) που φαίνεται στην εικόνα 10.2. Στο γραφικό αυτό περιβάλλον φαίνεται η σειριακή θύρα (COM1), στην οποία το πρόγραμμα περιμένει εισερχόμενα πακέτα μηνυμάτων καθώς και το port (9000) στο οποίο θα προωθήσει τα πακέτα που θα λάβει.



Σχήμα 10.2: SerialForwarder



Σχήμα 10.3: Oscilloscope

Παράλληλα με το SerialForward, εκτελέσαμε και την εφαρμογή Oscilloscope. Η εφαρμογή αυτή συνεργάζεται με το SerialForward με την έννοια ότι “ακούει” στο port 9000 του ίδιου H/Y για να λάβει τα πακέτα που προωθεί το SerialForward. Στη συνέχεια επεξεργάζεται την πληροφορία τους και αναπαριστά σε γραφική παράσταση τις τιμές των μετρήσεων, οι οποίες ουσιαστικά προέρχονται από τον αισθητήρα του φωτός που βρίσκεται στο mote - πομπό. Η εφαρμογή Oscilloscope εκτελείται από τον υποκατάλογο /tools/java με την εντολή

java net.tinyos.oscope.oscilloscope

Στην εικόνα 10.3 φαίνεται το γραφικό περιβάλλον του Oscilloscope.

Με τον τρόπο αυτό είδαμε μεταβολές στη γραφική παράσταση του Oscilloscope, όταν προσπαθήσαμε να σκιάσουμε τον αισθητήρα φωτός του mote - πομπού, διαπιστώνοντας έτσι την ορθή λειτουργία του.

Η απρόσκοπη λειτουργία του mote κράτησε περίπου 5 ημέρες, κατά τις οποίες έστελνε σωστά και ταχτικά πληροφορίες στο εικονικό κέντρο ελέγχου, ενώ το κόκκινο LED του, το οποίο υποδείχνει την ύπαρξη επαρκούς φωτισμένου περιβάλλοντος ήταν ανοικτό κατά τη διάρκεια της κάθε ημέρας από τις 7:00 το πρωί έως τις 8:00 το απόγευμα και το κίτρινο LED αναβόσβηνε 8 φορές ανά 5 δευτερόλεπτα. Μετά την πέμπτη μέρα τα ενεργειακά αποθέματα του mote εξαντλήθηκαν και άρχισε να δυσλειτουργεί, σε σημείο να μη στέλνει πια μηνύματα και τα LEDs του τελικά να σβήσουν οριστικά.

Δεδομένου λοιπόν της παραπάνω προσεγγίσεως για το χρόνο ζωής ενός mote που εκτελεί τη τροποποιημένη εφαρμογή OscilloscopeRf, μπορούμε να εξάγουμε κάποια παραπέρα συμπεράσματα. Στην προηγούμενη ενότητα είδαμε ότι συνολικά η δομή του ειδικού τύπου μηνύματος OscopeMsg καταλαμβάνει σταθερά 26 bytes. Καθώς όμως αυτή η δομή ενθυλακώνεται στη γενικευμένη δομή του TOS_Msg, τελικά το μήνυμα που θα αποστέλλεται ανά 10 δείγματα μετρήσεων θα έχει μέγεθος

$$2 + 1 + 1 + 1 + 26 = 31 \text{ bytes} \text{ ανά μήνυμα}$$

όπου τα 5 επιπλέον bytes “χρεώνονται” στα τέσσερα πρώτα πεδία της δομής TOS_Msg.

Αν σκεφτούμε ότι ανά 5 δευτερόλεπτα έχουμε την αποστολή 16 μηνυμάτων, τότε στις πέντε ημέρες ($5 \times 24 \times 3600$ δευτερόλεπτα) αντιστοιχεί αποστολή 1382400 μηνυμάτων ή αλλιώς 40,86 Mbytes πληροφορίας !

10.4 Δύο εφαρμογές multihop

Στα ήδη υπάρχοντα παραδείγματα εφαρμογών, που περικλείονται στο περιβάλλον ανάπτυξης εφαρμογών του TinyOS, η επικοινωνία των κόμβων περιορίζόταν σε μεταδόσεις **ενός βήματος**, από τον κόμβο πομπό προς τον κόμβο που αποτελούσε εικονικό κέντρο ελέγχου και ήταν συνδεδεμένος με H/Y. Ο κόμβος - κέντρο ελέγχου, εκτελούσε πάντα την εφαρμογή GenericBase, η οποία αναλάμβανε τη λήψη μηνυμάτων από το δέκτη και την προώθηση τους στη σειριακή θύρα του board. Αυτή η θύρα με τη σειρά της ήταν συνδεδεμένη στη σειριακή θύρα του H/Y μεταδίδοντας έτσι τη λαμβανόμενη πληροφορία σε εφαρμογές εκτελούμενες στο H/Y(πχ SerialForward).

Ως επόμενος στόχος της διπλωματικής ήταν η δημιουργία νέων εφαρμογών που θα μπορούσαν να κάνουν εφικτή τη μετάδοση σε πολλαπλά βήματα ενός μηνύματος από ένα mote - πομπό σε ένα εικονικό κέντρο ελέγχου ή γενικότερα σε ένα οποιοδήποτε mote - δέκτη. Έτσι θα ήταν εφικτή η υλοποίηση ενός πιο ρεαλιστικού πραγματικού πειραματικού δικτύου έξυπνης σκόνης καθώς η δυνατότητα multihop μεταδόσεων είναι σχεδόν απαραίτητη στα περισσότερα πρωτόκολλα διάδοσης πληροφορίας στα δίκτυα αυτά.

Η εφαρμογή GenericBase αποτέλεσε πρότυπο για την εργασία που αναλάβαμε, καθώς ήδη υλοποιούσε ένα μηχανισμό λήψης και προώθησης, αν και σε αυτή η προώθηση των μηνυμάτων γινόταν προς τη σειριακή θύρα του programming board και όχι στο δίκτυο. Ο βασικός μας στόχος ήταν η δημιουργία μιας εφαρμογής που εκτελούμενη σε ένα mote θα προωθούσε απευθείας τα μηνύματα που λάμβανε στο δίκτυο με μέθοδο broadcast.

Σύντομα όμως διαπιστώσαμε ότι αυτό δεν ήταν όσο εύκολο, όσο ακούγεται καθώς εμφανίστηκε ένα σημαντικό πρόβλημα. Η αποστολή ενός μηνύματος με τη μέθοδο broadcast συνεπάγεται ότι όλοι οι κόμβοι εντός της εμβέλειας μετάδοσης θα λάβουν το μήνυμα. Αν υποθέσουμε ότι οι κόμβοι αυτοί εκτελούν τη multihop εφαρμογή μας, τότε όλοι θα προωθήσουν το μήνυμα μόλις το λάβουν. Καθώς όμως αναπόφευκτα κάποιοι από τους κόμβους αυτούς είναι γειτονικοί θα λάβουν και πάλι το ίδιο μήνυμα και θα το επαναπροωθήσουν στο δίκτυο δημιουργώντας έτσι ατέρμονους βρόχους και πλεονάζουσες μεταδόσεις μηνυμάτων.

Για το λόγο αυτό προέκυψε η ανάγκη “ενσωμάτωσης” μέσα από την εφαρμογή μας μιας μνήμης (cache) σε κάθε κόμβο, η οποία θα διατηρούσε το ιστορικό των πρόσφατων μηνυμάτων που απέστειλε ο κόμβος, ώστε να αποφευχθεί η περιττή επαναπροωθησή τους.

Τελικά υλοποιήσαμε δύο εφαρμογές multihop, η πρώτη από τις οποίες προορίζεται για χρήση μαζί με την εφαρμογή CntToLedsAndRfm, ενώ η δεύτερη συνεργάζεται με την εφαρμογή OscilloscopeRf.

10.4.1 Εφαρμογή multihop για το CntToLedsAndRfm

Η πρώτη εφαρμογή έχει το όνομα ForwardCast και συνεργάζεται με τη γνωστή μας εφαρμογή CntToLedsAndRfm. Αυτό σημαίνει ότι πρέπει να προγραμματίσουμε απαραίτητα ένα mote με την εφαρμογή CntToLedsAndRfm, το οποίο θα λειτουργεί ως πομπός στο δίκτυο μας, ενώ τα υπόλοιπα motes μπορούν να εκτελούν την εφαρμογή ForwardCast και να διαδίδουν με multihop τρόπο τα μηνύματα του πομπού. Υπενθυμίζουμε ότι η CntToLedsAndRfm διατηρεί ένα μετρητή που σε κάθε τέταρτο του δευτερολέπτου αλλάζει τιμή, παίρνοντας κυκλικά τιμές από 0 έως 7. Ταυτόχρονα η νέα τιμή αποστέλλεται στο δίκτυο και η δυαδική της αναπαράσταση φαίνεται στα LEDs του πομπού.

Η ForwardCast αναλαμβάνει τη λήψη των μηνυμάτων αυτών και την προώθηση τους και πάλι στο δίκτυο. Ωστόσο, πριν προωθήσει ένα εισερχόμενο μήνυμα, φροντίζει να ελέγξει τη cache μηνύμη του mote, ώστε να εξαχριβώσει ότι το μήνυμα αυτό δεν έχει ξανααποσταλλεί πρόσφατα, και μόνο στην περίπτωση αυτή το προωθεί. Επίσης για λόγους επιβεβαίωσης της ορθής λειτουργίας του ForwardCast, η ακέραια τιμή, η οποία περιέχεται σε ένα μήνυμα, που προωθείται, αναπαριστάται στη δυαδική της μορφή από τα LEDs του mote. Έτσι εν τέλει τα LEDs του mote με την εφαρμογή ForwardCast επιδεικνύουν τις ίδιες ακριβώς ενδείξεις με αυτές των LEDs του mote που εκτελεί την εφαρμογή CntToLedsAndRfm, σημάδι ότι η εφαρμογή μας λειτουργεί σωστά.

Παρακάτω παραθέτουμε αποσπάσματα από τον κώδικα που γράψαμε για την εφαρμογή ForwardCast. Το βασικό configuration αρχείο είναι το FCast.nc και έχει τη παρακάτω μορφή

```

5    includes IntMsg;
6
7    configuration Fcast {
8      implementation {
9        components Main, FcastM, GenericComm as Comm, IntToLeds, PotC;
10
11      Main.StdControl -> FcastM;
12      Main.StdControl -> IntToLeds.StdControl;
13
14      FcastM.IntOutput -> IntToLeds;
15      FcastM.CommControl -> Comm;
      FcastM.SendTheMsg -> Comm. SendMsg [AM_INTMMSG];
      FcastM.ReceiveTheMsg -> Comm. ReceiveMsg [AM_INTMMSG];
      FcastM.Pot -> PotC;
    }
}

```

όπως είναι φανερό στη διασύνδεση του configuration, περικλείουμε το module GenericComm που παρέχει τις διεπαφές SendMsg και ReceiveMsg, για την αποστολή και λήψη μηνυμάτων. Οι διεπαφές αυτές είναι παραμετροποιήσιμες και παίρνουν ως όρισμα τον τύπο του μηνύματος AM_INTMMSG που στέλνει η εφαρμογή CntToLedsAndRfm. Το module FcastM, υλοποιεί τη βασική λειτουργία της εφαρμογής μας καθώς όπως υποδεικνύει το wiring αναλαμβάνει το χειρισμό των λαμβανόμενων μηνυμάτων, την αποστολή τους, την αρχικοποίηση του ποτενσιομέτρου για τον πομποδέκτη και την αναπαράσταση των ακεραίων αριθμών στα LEDs του mote. Ας σημειωθεί ότι είναι το άρθρωμα IntToLeds, αυτό το οποίο παρέχει τις εντολές αναπαράστασης ενός ακεραίου στα LEDs, και το άρθρωμα PotC, αυτό που παρέχει τις εντολές ρύθμισης του ποτενσιομέτρου του πομποδέκτη. Το αρχείο FCastM.nc περιέχει τη δήλωση και υλοποίηση του module FCastM και ακολουθεί παρακάτω

```

includes IntMsg;

module FcastM {
    provides interface StdControl;
    uses {
        interface Pot;
        interface ReceiveMsg as ReceiveTheMsg;
        interface SendMsg as SendTheMsg;
        interface StdControl as CommControl;
        interface IntOutput;
    }
}
// IMPLEMENTATION OF MODULE
implementation {
    TOS_MsgPtr msg;
    bool bcast_pending;
    TOS_Msg buf;
    int8_t head;
    enum{maxdata=4, maskdata=0x3 };// to mask data einai gia ylopoikhsh prakhs modulo ston kykliko
    buffer
    int16_t rdata[maxdata]; // o kyklikos buffer me ta maxdata pio recent packets
}

// Xeirismos tou SEND DONE event gia reset tou pending flag
//
event result_t SendTheMsg.sendDone(TOS_MsgPtr pmsg, result_t
status) {
    if (status == SUCCESS) bcast_pending = FALSE;
    return status;
}

// Apo8hkeoume to INT perioxomeno twn mhnymatwn sto cyclic buffer
//
inline void putdata(int16_t val)
{
    int16_t p = (int16_t)head;
    head = (p+1) & maskdata;
    rdata[p] = val;
}

// Arxikopoikhsh ths application
//
command result_t StdControl.init() {
    int16_t my_i;
    msg = &buf;
    bcast_pending = FALSE;

    head = 0;
    for(my_i = 0 ; my_i < maxdata ; my_i=my_i+1)
    {
        putdata(0xff); // arxikopoikhsh ths kyklikhs cache.
    }
    call Pot.init(30);

    return (call CommControl.init());
}
// h Start function
//
command result_t StdControl.start(){
    return (call CommControl.start());
}

// H stop function
//
command result_t StdControl.stop(){
    return (call CommControl.stop());
}

// A module-scoped inline function.
//
// Decide whether a received message is new. To val tou DEN prepei na yparxei sta prhgoumena
// apo8hkeumena
// sthn kyklikh cache
//
inline bool is_new_msg(struct IntMsg *bmsg) {
    bool ich_dont_habe;
    int16_t my_other_i;
    if (bcast_pending) return FALSE;
    ich_dont_habe = TRUE;
    for(my_other_i = 0 ; my_other_i < maxdata ; my_other_i = my_other_i+1)
    {
        if(((int16_t)bmsg->val) == rdata[my_other_i])
        {
            ich_dont_habe = FALSE;
            return ich_dont_habe;
        }
    }
}

```

```

95     return ich_dont_habe;
}

// Handle function gia tha periptwsh pou ginei ena RECEIVE!!!!!
100    event TOS_MsgPtr ReceiveTheMsg.receive(TOS_MsgPtr pmsg)
{
    TOS_MsgPtr ret = msg;
    result_t retval;
    IntMsg *message;
    msg = pmsg;
    //
    // Edw kanoume casting se Intmsg 'h de kserw egw ti kai pairnoume tha plhroforia pou 8eloume na
    // deixnoume sta LEDs kai FYSIKA tha plhroforia pou 8a sygkrinoume an ypsarxei hdh sthn cache
    . (mporei
110    // autes oi dyo plhrofories na einai idies (RFTOLEDSTORF)
    message = (IntMsg *)msg->data;
    //
    // Check if this is a new broadcast message
115    if (is_new_msg(message))
    {
        // apouhkeoume th nea plhroforia sthn kyklikh cache
        putdata((int16_t)message->val);
        //
        // edw stelnoume th sygkekrimenh plhroforia
        //
        if (!bcast_pending /*&& (received->group == (TOS_AM_GROUP & 0xff))*/)
        {
            //
            // Kanoume standard forward apo Rf se RF kai deixnoume ti metadidoume.
            //
            // h message->val paramenei kai allazw to message->src pedio
            message->src = TOS_LOCAL_ADDRESS;
            retval = call SendTheMsg.send(TOS_BCAST_ADDR, sizeof(IntMsg), msg);
            call IntOutput.output(message->val);
            if (retval != FAIL)
            {
                bcast_pending = TRUE;
            }
            else
            {
                call IntOutput.output(0xFF);      // ola ta fwta anoixta xwris logo ypodeiknyoun
                kapoio la8os
            }
        } // kanoume send MONO otan DEN yparoun PENDING SENDS... (to pending epanerxetai me to
          event sendDone)
    }
    return ret;
}
event result_t IntOutput.outputComplete(result_t success)
{
    return 0;
}
} // end of implementation

```

Γενικά η λειτουργία του module FCastM μπορεί να συνοψιστεί στα ακόλουθα. Στο mote διατηρείται ένας κυκλικός buffer rdata των τεσσάρων θέσεων. Αρχικά το ποτενσιόμετρο του πομποδέκτη αρχικοποιείται στην τιμή 30, η οποία αντιστοιχεί χονδρικά σε εμβέλεια μετάδοσης γύρω στο 1 μέτρο. Επίσης ο κυκλικός buffer που αντιπροσωπεύει την cache μνήμη του mote αρχικοποιείται σε τιμή 0xFF (δεκαεξαδική) για όλες τις καταχωρήσεις του, καθώς αυτή η τιμή δε πρόκειται να εμφανιστεί ποτέ στα λαμβανόμενα μηνύματα από το CntToLedAndRfm. Όταν συμβεί ένα γεγονός λήψης μηνύματος από τον πομποδέκτη καλείται η συνάρτηση γειρισμού λαμβανόμενων μηνύμάτων

event TOS_MsgPtr ReceiveTheMsg.receive(TOS_MsgPtr pmsg)

Εκεί βρίσκουμε το πέμπτο πεδίο (data) της δομής TOS_Msg του λαμβανόμενου μηνύματος και κάνουμε casting σε δείκτη δομής IntMsg. Αυτή η δομή είναι η ειδική δομή που χρησιμοποιεί η εφαρμογή CntToLedsAndRfm για την αποστολή μηνυμάτων και έχει δύο πεδία: α) την ακέραια τιμή του μετρητή του πομπού και β) την ταυτότητα του πομπού (η οποία τίθεται πάντα στη προκαθορισμένη τιμή TOS_LOCAL_ADDRESS). Ελέγχουμε έπειτα ότι η ακέραια τιμή του μετρητή δεν έχει αποσταλεί στις τέσσερις πιο πρόσφατες μεταδόσεις και αν όντως

αυτό ισχύει, τότε αυτή η τιμή προστίθεται στη cache rdata του mote και αποστέλλεται στο δίκτυο με μέθοδο Broadcast. Εφόσον η αποστολή στεφθεί με επιτυχία, τότε η τιμή που περιείχε το μήνυμα αναπαριστάται και στα LEDs του mote. Σε περίπτωση, ωστόσο, που υπάρχει ήδη η τιμή αυτή στην κυκλική cache του mote, τότε απορρίπτουμε το λαμβανόμενο μήνυμα.

Για την εξακρίβωση της ορθής λειτουργίας της εφαρμογής μας προγραμματίσαμε ένα mote με την εφαρμογή CntToLedsAndRfm, ώστε να αποτελεί τον πομπό στο δίκτυο μας και δύο άλλα motes με την εφαρμογή ForwardCast. Τοποθετήσαμε έπειτα τα motes έτσι ώστε το ένα μόνο από τα δύο ForwardCast motes να βρίσκεται εντός της εμβέλειας μεταδόσεων του mote - πομπού CntToLedsAndRfm. Το δεύτερο ForwardCast mote τοποθετήθηκε εκτός της εμβέλειας του mote - πομπού, αλλά εντός της εμβέλειας του πρώτου ForwardCast mote. Ενεργοποιώντας τα mote, διαπιστώσαμε από τις ενδείξεις των Leds τους ότι τα μηνύματα του πομπού διαδίδονταν με επιτυχία στο πρώτο ForwardCast mote και έπειτα από αυτό το mote προωθούνταν στο δεύτερο ForwardCast mote, κάνοντας πραγματικότητα μια multihop διάδοση μηνυμάτων.

10.4.2 Εφαρμογή multihop για το OscilloscopeRf

Η δεύτερη εφαρμογή multihop έχει την ονομασία ForwardOsc και υλοποιήθηκε με σκοπό να συνεργάζεται με την εφαρμογή OscilloscopeRf. Υπενθυμίζουμε ότι η εφαρμογή OscilloscopeRf όταν εκτελείται σε ένα MICA mote, το οποίο διαθέτει προσαρτημένο κύκλωμα με αισθητήρες (sensor board), τότε συλλέγει πληροφορίες από τον αισθητήρα φωτός, λαμβάνοντας δείγματα μετρήσεων με καθορισμένο ρυθμό και κατόπιν στέλνει στο ασύρματο μέσο μηνύματα που περιέχουν σύνολα από 10 μετρήσεις το κάθε ένα.

Στο βασικό configuration άρθρωμα της εφαρμογής ForwardOsc οι διαφορές είναι ελάχιστες συγκριτικά με το βασικό configuration της ForwardCast. Όπως φαίνεται παρακάτω αυτό που έχει αλλάξει είναι η επισύναψη (include) του αρχείου επικεφαλίδας που ορίζει την ειδική δομή μηνύματος (OsceopeMsg) της εφαρμογής OscilloscopeRf και οι παράμετροι στις διεπαφές SendMsg και ReceiveMsg, που πλέον αντιστοιχούν στον τύπο των μηνυμάτων της εφαρμογής OscilloscopeRf.

```

1    includes OsceopeMsg;
2
3    configuration Fosc {
4      implementation {
5        components Main , FoscM , GenericComm as Comm, IntToLeds , PotC;
6
7        Main .StdControl -> FoscM;
8        Main .StdControl -> IntToLeds .StdControl ;
9
10       FoscM .IntOutput -> IntToLeds ;
11       FoscM .CommControl -> Comm;
12       FoscM .SendTheMsg -> Comm. SendMsg [AM_OSCOPEMSG] ;
13       FoscM .ReceiveTheMsg -> Comm. ReceiveMsg [AM_OSCOPEMSG] ;
14       FoscM .Pot -> PotC;
15     }
}

```

Για το module FoscM, του οποίου ο κώδικας ακολουθεί, η υλοποίηση είναι παρόμοια με του FCastM

```

1    includes OsceopeMsg;
2
3    module FoscM {
4      provides   interface StdControl;
5      uses {
6        interface Pot;
7        interface ReceiveMsg as ReceiveTheMsg;
8        interface SendMsg as SendTheMsg;
}

```

```

10 //      interface ReceiveMsg as ResetCounterMsg;
11 //      interface StdControl as CommControl;
12 //      interface IntOutput;
13 }
14 /**
15 /**
16 /**
17 /**
18 /**
19 /**
20 /**
21 /**
22 /**
23 /**
24 /**
25 /**
26 /**
27 /**
28 /**
29 /**
30 /**
31 /**
32 /**
33 /**
34 /**
35 /**
36 /**
37 /**
38 /**
39 /**
40 /**
41 /**
42 /**
43 /**
44 /**
45 /**
46 /**
47 /**
48 /**
49 /**
50 /**
51 /**
52 /**
53 /**
54 /**
55 /**
56 /**
57 /**
58 /**
59 /**
60 /**
61 /**
62 /**
63 /**
64 /**
65 /**
66 /**
67 /**
68 /**
69 /**
70 /**
71 /**
72 /**
73 /**
74 /**
75 /**
76 /**
77 /**
78 /**
79 /**
80 /**
81 /**
82 /**
83 /**
84 /**
85 /**
86 /**
87 /**
88 /**
89 /**
90 /**
91 /**
92 /**
93 /**
94 /**
95 /**
96 /**
97 /**
98 /**
99 /**
100 /**

```

```

105     if (((int16_t)bmmsg->lastSampleNumber) == rdata[my_other_i])
106     {
107         ich_dont_habe = FALSE;
108         return ich_dont_habe;
109     }
110 }

115 /**
116 // Handle function gia thn periptwsh pou ginei ena RECEIVE!!!!
117 /**
118 /**
119 event TOS_MsgPtr ReceiveTheMsg.receive(TOS_MsgPtr pmsg)
120 {
121     TOS_MsgPtr ret = msg;
122     result_t retval;
123     struct OscopeMsg * message;
124     msg = pmsg;
125     /**
126     // Edw kanoume casting se Intmsg 'h de kserw egw ti kai pairnoume thn plhroforia pou 8eloume na
127     // deixnoume sta LEDA kai FYSIKA thn plhroforia pou 8a sygkrinoyme an ypsarxei hdh sthn cache
128     .(mporei
129     // autes oi dyo plhrofories na einai idies (RFTOLEDSTORF)
130
131     message = (struct OscopeMsg *)msg->data;
132
133     // Check if this is a new broadcast message
134
135     if (is_new_msg(message))
136     {
137         // apouhkeuoume th nea plhroforia sthn kyklikh cache
138         putdata((int16_t)message->lastSampleNumber);
139         /**
140         // edw stelnoume th sygkekrimenh plhroforia
141         /**
142         if (!bcast_pending /*&& (received->group == (TOS_AM_GROUP & 0xff))*/)
143         {
144             /**
145             // Kanoume standard forward apo Rf se RF kai deixnoume ti metadidoume.
146             /**
147             // h message->val paramenei kai allazw to message->src pedio
148             message->sourceMoteID = TOS_LOCAL_ADDRESS;
149             retval = call SendTheMsg.send(TOS_BCAST_ADDR, sizeof(struct OscopeMsg), msg);
150             call IntOutput.output(message->lastSampleNumber);
151             if (retval != FAIL)
152             {
153                 bcast_pending = TRUE;
154             }
155             else
156             {
157                 call IntOutput.output(0xFF);      // ola ta fwta anoixta xwrис logo ypodeiknyoun
158                 kapoio la8os
159             }
160         } // kanoume send MONO otan DEN yparoun PENDING SENDS... (to pending epanerxetai me to
161           // event sendDone)
162     }
163     return ret;
164 }
165 event result_t IntOutput.outputComplete(result_t success)
166 {
167     return 0;
168 }
169 } // end of implementation

```

Και πάλι εδώ έχουμε μια κυκλική cache των 4 θέσεων, και το ποτενσιόμετρο του πομποδέκτη αρχικοποιείται στην τιμή 0, ώστε να είναι μέγιστη η ακτίνα μετάδοσης. Όταν συμβεί ένα γεγονός λήψης μηνύματος, τότε καλείται η συνάρτηση

event TOS_MsgPtr ReceiveTheMsg.receive(TOS_MsgPtr pmsg)

η οποία αποσπάει με κατάλληλο casting τη χρήσιμη δομή OscopeMsg από τα δεδομένα του λαμβανόμενου μηνύματος. Έπειτα εντοπίζουμε την τιμή του πεδίου της OscopeMsg που δείχνει τον αύξοντα αριθμό του πιο πρόσφατου δείγματος μέτρησης (lastSampleNumber) από τα δέκα που περιέχονται στο μήνυμα. Αν αυτός ο αριθμός βρίσκεται ήδη στην cache του mote, τότε το μήνυμα απορρίπτεται, ενώ στην αντίθετη περίπτωση προωθείται με τεχνική broadcast στο δίκτυο και τα Leds του mote απεικονίζουν τη δυαδική αναπαράσταση του lastSampleNumber.

Το εντυπωσιακό τμήμα της εφαρμογής ForwardOsc έγκειται στην υλοποίηση ενός πρότυπου δικτύου από τρεις κόμβους. Ο πρώτος κόμβος προγραμματίστηκε με την εφαρμογή OscilloscopeRf και σε αυτόν προσαρτήθηκε μια πλακέτα MTS310 με αισθητήρες για να λαμβάνει μετρήσεις από τον αισθητήρα φωτός. Ο δεύτερος κόμβος προγραμματίστηκε με το ForwardOsc για να αποτελεί τον ενδιάμεσο κόμβο στις μεταδόσεις του κόμβου OscilloscopeRf. Τέλος ο τρίτος κόμβος προγραμματίστηκε με την εφαρμογή GenericBase, ώστε να λαμβάνει εισερχόμενα μηνύματα και να τα προωθεί στη σειριακή θύρα του programming board. Ο κόμβος αυτός αποτέλεσε το εικονικό κέντρο ελέγχου, όπου καταλήγουν όλα τα μηνύματα του πρώτου κόμβου.



Σχήμα 10.4: Εφαρμογή της multihop διάδοσης για το OscilloscopeRf

Ακολουθώντας τη διαδικασία που περιγράφαμε νωρίτερα, αφήσαμε τον κόμβο με το GenericBase πάνω στο programming board και συνδέσαμε αυτό το board με τη σειριακή θύρα COM1 ενός H/Y. Στον υπολογιστή αυτό εκτελέσαμε τις java εφαρμογές SerialForward και Oscilloscope. Τοποθετήσαμε έπειτα τον πρώτο κόμβο με το OscilloscopeRf σε τέτοια απόσταση από το κέντρο ελέγχου, ώστε το τελευταίο να είναι εκτός της εμβέλειας των μεταδόσεων. Τότε η java εφαρμογή Oscilloscope δεν έδειχνε στη γραφική παράσταση της καμία μέτρηση. Όταν, ωστόσο, ανάμεσα στο OscilloscopeRf mote και στο GenericBase mote παρεμβάλλαμε το mote με το ForwardOsc όπως φαίνεται στην εικόνα 10.4, είδαμε στη γραφική παράσταση του οπτικού περιβάλλοντος (GUI) Oscilloscope, ότι το κέντρο ελέγχου μπορούσε να λαμβάνει πλέον όλες τις μετρήσεις από τον αισθητήρα φωτός του πρώτου mote. Έτσι επιβεβαιώσαμε ότι το ForwardOsc mote, μπορεί και αποτελεί όντως ενδιάμεσο κόμβο για τη μετάδοση μηνυμάτων της εφαρμογής OscilloscopeRf.

10.5 TinyDB

Ως μια τελευταία αλλά καθόλου ευκαταφρόνητη εφαρμογή υλοποίησης πραγματικού πειραματικού δικτύου έξυπνης σκόνης θα πρέπει να αναφέρουμε το TinyDB [42]. Το TinyDB αποτελεί ένα πολυδύναμο εργαλείο για τη διαχείριση και την οργάνωση ενός πραγματικού πειραματικού δικτύου έξυπνης σκόνης. Είναι ανάμεσα στα εργαλεία που συνοδεύουν τις εκδόσεις του TinyOS και έχει αρκετά εντυπωσιακές εφαρμογές.

Η ανάπτυξη του TinyDB ξεκίνησε από τους Sam Madden, Wei Hong, Joe Hellerstein και Kyle Stankel στη διάρκεια της άνοιξης του 2002 στο ίνστιτούτο Intel-Research Berkeley και το πανεπιστήμιο UC Berkeley.

Ο σκοπός του εργαλείου TinyDB είναι να αποτελέσει ένα σύστημα υποβολής ερωτήσεων (queries) σε ένα δίκτυο από αισθητήρες, ώστε να μπορεί να αποσπάει πληροφορίες από τα motes του δικτύου. Το TinyDB χρησιμοποιεί μια διεπαφή για την υποβολή queries, η οποία είναι παρόμοια με τη διεπαφή του δημοφιλούς συστήματος διαχείρισης βάσεων δεδομένων (RDBMS) SQL. Το δίκτυο με τους αισθητήρες αντιμετωπίζεται ως μια κατανεμημένη βάση δεδομένων, από την οποία μπορούμε να λαμβάνουμε τις επιθυμητές πληροφορίες με προγραμματιζόμενο ρυθμό. Έχοντας ετοιμάσει μια ερώτηση (query), η οποία αντικατοπτρίζει το ενδιαφέρον μας για συγκεκριμένες πληροφορίες του δικτύου, το TinyDB αναλαμβάνει τη διαδικασία διάδοσης της ερώτησης στους κόμβους του δικτύου, της συλλογής της πληροφορίας που δίνουν αυτοί ως απάντηση, το φίλτραρισμα ή την συγχώνευση των επιμέρους πληροφοριών και την προώθηση των δεδομένων στη σειριακή ύδρα ενός H/Y.

Ας σημειωθεί ότι το TinyDB μπορεί να χρησιμοποιηθεί ως αυτόνομη εφαρμογή προσφέροντας ακόμα και φιλικό γραφικό περιβάλλον διεπαφής με τον τελικό χρήστη, για την υποβολή ερωτήσεων στο δίκτυο και την παρακολούθηση της κατάστασης των κόμβων του δικτύου. Μπορεί ακόμα εναλλακτικά να ενσωματωθεί σε νέες εφαρμογές που επιθυμούν να το χρησιμοποιήσουν ως μηχανισμό ανάκτησης πληροφορίας από δίκτυα έξυπνης σκόνης, παρέχοντας παράλληλα στον προγραμματιστή ένα περιβάλλον ανάπτυξης εφαρμογών Java (API), για τη συγγραφή προγραμμάτων που θα εκτελούνται σε H/Y και θα μπορούν να αποσπούν πληροφορίες από το δίκτυο. Μάλιστα το TinyDB έχει δημιουργηθεί ακριβώς για να απαλλάξει το μέσο προγραμματιστή εφαρμογών για δίκτυα έξυπνης σκόνης, από τη διαδικασία συγγραφής κώδικα σε NesC προκειμένου να υλοποιηθούν εφαρμογές ανάκτησης πληροφορίας από το δίκτυο. Το TinyDB υπόσχεται ότι χρησιμοποιεί αποδοτικούς υπολογιστικά και ενεργειακά αλγορίθμους για να επιτύχει το σκοπό του και μπορεί να θεωρηθεί ως ένα “μαύρο κουτί” για το μέσο προγραμματιστή που μπορεί να δομήσει περίπλοκες εφαρμογές “βασιζόμενος” στις αξιόπιστες υπηρεσίες του TinyDB.

Στο σημείο αυτό μπορούμε να συνοψίσουμε μερικές από τις σημαντικότερες υπηρεσίες που προσφέρει το TinyDB. Αυτές είναι

- 1) Ερωτήσεις με δομή υψηλού επιπέδου. Αυτό σημαίνει ότι η σύνταξη των ερωτήσεων (queries) που υποβάλλονται στο δίκτυο είναι πολύ κοντά στη φυσική γλώσσα, καθιστώντας εύκολη και εύληπτη τη λειτουργία τους.
- 2) Διαχείριση τοπολογίας του δικτύου. Το TinyDB υπόσχεται να διαχειρίζεται το δίκτυο των motes, εξασφαλίζοντας τη μέγιστη δυνατή συνεκτικότητα του, διατηρώντας καταχωρήσεις για τους γειτονικούς κόμβους του κάθε mote και πίνακες δρομολόγησης για την αποδοτική και αξιόπιστη μετάδοση μιας πληροφορίας στο κέντρο ελέγχου.

- 3) Πολλαπλές ερωτήσεις. Οι μηχανισμοί που χρησιμοποιεί το TinyDB, εγγυώνται ότι ένας κόμβος ή ένα σύνολο κόμβων μπορεί να υποβληθεί ταυτόχρονα σε περισσότερες από μία ερωτήσεις, οι οποίες μπορεί να αφορούν διαφορετικά χαρακτηριστικά ή αισθητήρες του κόμβου και ενδέχεται να αντιστοιχούν σε διαφορετικούς ρυθμούς δειγματοληψίας.
- 4) Υποστήριξη επεκτασιμότητας του δίκτυου. Ανά πάσα στιγμή στο δίκτυο μπορούν να προστεθούν νέα motes τα οποία εκτελούν την εφαρμογή TinyDB. Το TinyDB αναλαμβάνει να εντάξει τους νέους κόμβους στην τοπολογία του δίκτυου που διαχειρίζεται, εξασφαλίζοντας την απρόσκοπη λειτουργία του δίκτυου μετά από τοπολογικές αλλαγές.
- 5) Επεκτασιμότητα της διεπαφής. Στο TinyDB διατηρείται ένας κατάλογος με όλες τις διαθέσιμες εντολές για ανάκτηση πληροφορίας από το δίκτυο, και τα διαθέσιμα χαρακτηριστικά, που μπορεί ο τελικός χρήστης να ελέγξει σε κάθε mote του δίκτυου. Αυτός ο κατάλογος μπορεί να επεκταθεί με νέες εντολές και χαρακτηριστικά, που μπορεί να υλοποιήσει κατά τις απαιτήσεις του ο κάθε προγραμματιστής.

Παρακάτω περιγράφεται η διαδικασία που ακολουθήσαμε για την εγκατάσταση του TinyDB σε ένα πραγματικό πειραματικό δίκτυο από MICA motes.

Για τη χρησιμοποίηση του TinyDB, θα πρέπει κάποιος να εγκαταστήσει τη NesC εφαρμογή TinyDBApp σε κάθε κόμβο-mote του δίκτυου από αισθητήρες. Το TinyDBApp βρίσκεται στον υποκατάλογο /apps/TinyDBApp/ και πρέπει ταυτόχρονα με τον προγραμματισμό του κάθε κόμβου να προσδώσουμε σε αυτόν και μία μοναδική ταυτότητα. Τουτέστι για κάθε κόμβο του δίκτυου εκτελούμε την εντολή

```
make mica install.x
```

Όπου *x* είναι ένας ακέραιος αριθμός, ο οποίος καθορίζει την ταυτότητα του κάθε κόμβου. Φροντίζουμε επίσης να αναθέσουμε την ταυτότητα “0” στον κόμβο, ο οποίος θα αποτελέσει το “κέντρο ελέγχου” παραμένοντας συνδεδεμένος πάνω στο programming board, ώστε να προωθεί δεδομένα στη σειριακή του έξοδο. Η ταυτότητα αυτού του κόμβου πρέπει υποχρεωτικά να έχει την τιμή “0” σύμφωνα με το documentation του TinyDB. Το programming board με τον κόμβο “0” συνδέεται στη σειριακή θύρα του H/Y, όπου βρίσκεται εγκατεστημένη η πλατφόρμα του TinyOS.

Στη συνέχεια μπορούμε να εκτελέσουμε τη java εφαρμογή TinyDBMain, η οποία επιδεικνύει τις δυνατότητες του TinyDB. Για να το κάνουμε αυτό θα πρέπει πρώτα να θέσουμε μια μεταβλητή περιβάλλοντος (environment variable) που έχει το όνομα CLASSPATH με τις τιμές που περιέχονται στο αρχείο /tools/java/javapath. Αυτό μπορεί να γίνει απλά αν από ένα csh shell εκτελέσουμε την εντολή

```
setenv CLASSPATH '/pathtotinyos/tinyos-1.x/tools/java/javapath'
```

όπου το /pathtotinyos/ θα αντικατασταθεί με το πραγματικό μονοπάτι προς την εγκατάσταση του TinyOS, το οποίο πιθανόν να διαφέρει σε κάθε H/Y. Η διαδικασία αυτή γίνεται, γιατί το εκτελέσιμο της java εφαρμογής TinyDBMain απαιτεί για τη λειτουργία του τη σύνδεση με κάποια αρχεία JLex.jar, cup.jar και plot.jar. Το μονοπάτι προς αυτά τα αρχεία ορίζεται στη μεταβλητή CLASSPATH. Αφού θέσουμε την παραπάνω μεταβλητή πηγαίνουμε στον υποκατάλογο της java εφαρμογής TinyDBMain με

```
cd tinyos-1.x/tools/java/net/tinyos/tinydb
```

και εκτελούμε

```
make
```

Έπειτα από την επιτυχή μεταγλώττιση του κώδικα της εφαρμογής επιστρέφουμε στο κατάλογο/tinyos-1.x/tools/java και εκτελούμε την εφαρμογή με

```
java net/tinyos.tinydb.TinyDBMain
```

Στο σημείο αυτό θα εμφανιστούν δύο “παράθυρα”, εκ των οποίων το ένα είναι το παράθυρο εντολών, μέσω του οποίου μπορούμε να επιλέξουμε και να στείλουμε κάποιες εντολές στα motes του δικτύου και το άλλο είναι το παράθυρο των ερωτήσεων, όπου μπορεί κανείς να συντάξει queries και να τις υποβάλει στο δίκτυο.



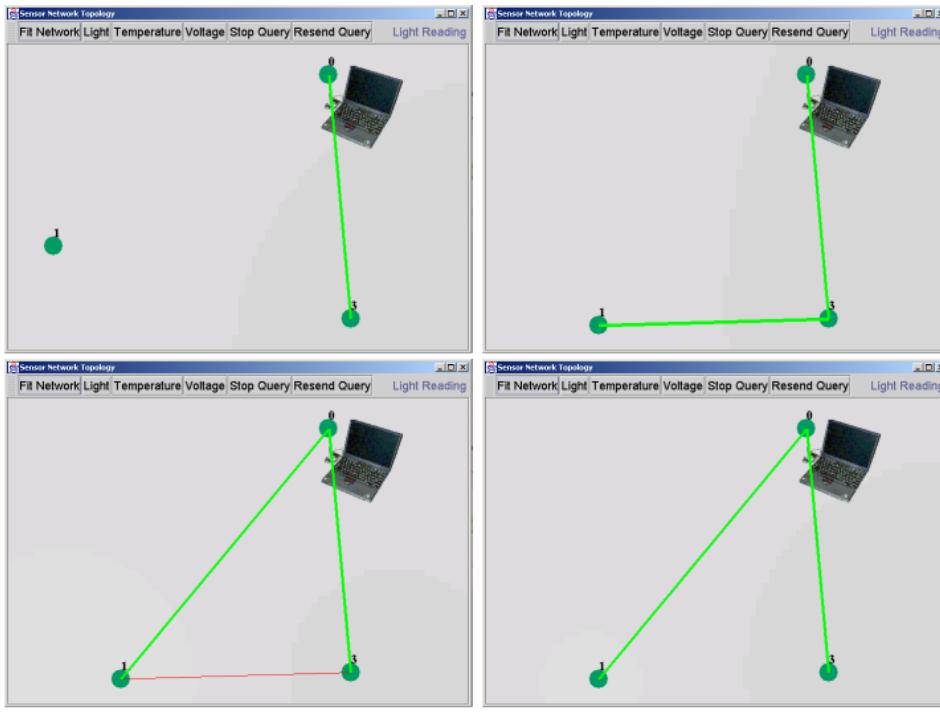
Σχήμα 10.5: Το παράθυρο ερωτήσεων

Σχήμα 10.6: Το παράθυρο εντολών του TinyDBMain

Το πραγματικό πειραματικό δίκτυο, που δημιουργήσαμε εμείς, περιέχει τρεις κόμβους εκ των οποίων ο ένας είχε ταυτότητα “0” και αποτέλεσε το κέντρο ελέγχου, ενώ οι δύο άλλοι είχαν ταυτότητες “1” και “3”.

Αφού εκτελέσαμε την εφαρμογή TinyDBMain, ασχοληθήκαμε περισσότερο με τις υπηρεσίες, που προσφέρει το παράθυρο των ερωτήσεων, οι οποίες είναι αρκετά ενδιαφέρουσες.

Επιλέγοντας το “Network Topology” άνοιξε ένα νέο παράθυρο, όπου μπορούσαμε να δούμε την τοπολογία του δικτύου μας, δηλαδή τη σχετική θέση των κόμβων καθώς και τα μονοπάτια δρομολόγησης πληροφορίας προς το κέντρο ελέγχου. Τα μονοπάτια σχηματίζονται από τις ακμές που ζωγραφίζει η εφαρμογή ανάμεσα στους κόμβους. Ουσιαστικά πατώντας το κουμπί “Network Topology” υποβλήθηκε μια ειδική query στο δίκτυο, ώστε κάθε κόμβος να ανακοινώσει στο κέντρο ελέγχου πληροφορίες σχετικά με τους γείτονες - κόμβους με τους οποίους μπορεί να επικοινωνεί. Το κέντρο ελέγχου διαχειρίζεται αυτή την πληροφορία και αναπαριστά μια πιθανή τοπολογία του δικτύου. Αυτή η λειτουργία γενικά αποδεικνύεται χρήσιμη καθώς μπορούμε να διαπιστώσουμε αν όλοι οι κόμβοι του πραγματικού πειραματικού μας δικτύου λειτουργούν και επικοινωνούν σωστά.

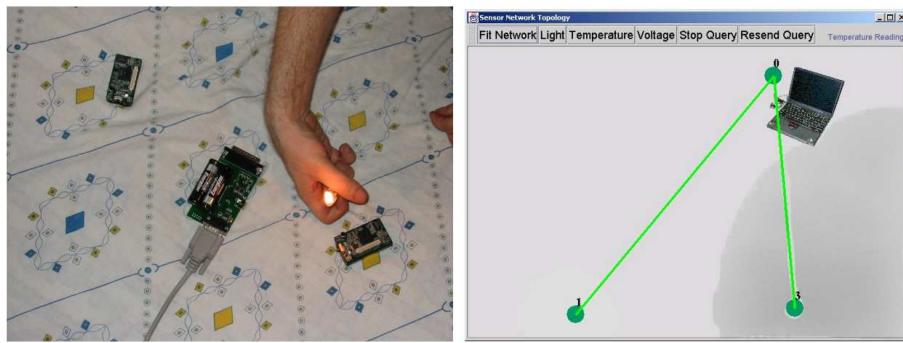


Σχήμα 10.7: Η ανακάλυψη ενός νέου κόμβου στο TinyDBMain

Επιπρόσθετα, στο νέο αυτό παράθυρο (Network Topology) υπάρχουν κάποια “κουμπιά”, που αποτελούν στην ουσία συντομεύσεις για την αυτόματη υποβολή κάποιων βασικών ερωτήσεων στο δίκτυο με τα MICA motes. Οι συντομεύσεις αυτές είναι για τον έλεγχο των επιπέδων φωτεινότητας που αντιλαμβάνεται ο κάθε κόμβος, τον έλεγχο των ενεργειακών αποθεμάτων κάθε κόμβου και της θερμοκρασίας κάθε κόμβου. Πατώντας το κουμπί για τον έλεγχο των επιπέδων του φωτός που αντιλαμβάνεται ο κάθε κόμβος παρατηρήσαμε ότι σκιάζοντας τον ένα κόμβο, τότε στο παράθυρο με την τοπολογία του δικτύου η περιοχή γύρω από τον αντίστοιχο κόμβο σκοτείνιασε. Έτσι με ένα εντυπωσιακά εύληπτο τρόπο μπορούσαμε να παρατηρούμε άμεσα τις απότομες αλλαγές στις μετρήσεις που έστελνε ο κάθε κόμβος. Πατώντας το κουμπί για τον έλεγχο της θερμοκρασίας παρατηρήσαμε ανάλογη αντίδραση στο δίκτυο, όταν ανεβάσαμε με τη βοήθεια ενός αναπτήρα τη θερμοκρασία του περιβάλλοντος κοντά σε ένα mote του δικτύου μας. Αυτό φαίνεται παραστατικά στην εικόνα 10.8. Τέλος το κουμπί για τον έλεγχο των ενεργειακών αποθεμάτων δεν έδειξε διαφοροποίηση στο δίκτυο, γεγονός αναμενόμενο καθώς όλοι οι κόμβοι ήταν εφοδιασμένοι με καινούργιες AA μπαταρίες.

Τέλος επιστρέφοντας στο παράθυρο των ερωτήσεων, δοκιμάσαμε τη φιλική διεπαφή που προσφέρεται για τη σύνταξη μιας ερώτησης και την υποβολή της προς το δίκτυο. Ο χρήστης μπορεί να επιλέξει από μια λίστα ποια χαρακτηριστικά και ποιους αισθητήρες των motes επιθυμεί να ελέγξει. Επίσης μπορεί να ρυθμίσει το ρυθμό δειγματοληψίας για τα χαρακτηριστικά αυτά ή τους αισθητήρες (sample period) και κατόπιν να παρακολουθήσει με γραφικό τρόπο σε ένα νέο παράθυρο τις απαντήσεις που στέλνει ο κάθε κόμβος.

Οι δυνατότητες της εφαρμογής TinyDB δε σταματούνε εδώ και αποτελεί πρόκληση η πα-



Σχήμα 10.8: Αύξηση θερμοκρασίας ενός mote στο πραγματικό πειραματικό δίκτυο - αντίδραση στο παράθυρο του TinyDB

ραπέρα εξερεύνηση των υπηρεσιών του TinyDB, αλλά και η σύνταξη εφαρμογών που θα στηρίζονται στην υποδομή που προσφέρει αυτό.

Κεφάλαιο 11

Επίλογος

Ο στόχος της διπλωματικής αυτής ήταν να παρουσιάσει την τεχνολογία των δικτύων έξυπνης σκόνης και τα σύγχρονα προτεινόμενα πρωτόκολλα διάδοσης πληροφορίας στα δίκτυα αυτά. Επίσης βασικός στόχος μας ήταν η δημιουργία μιας σταθερής και αξιόπιστης υποδομής για την απαρχή και τη συνέχιση της έρευνας πάνω στα δίκτυα αυτά, σε τοπικό επίπεδο. Η υποδομή αυτή συνιστάται από τον υλοποιημένο εξομοιωτή και τον οπτικό προσομοιωτή, οι οποίοι ενσωματώνουν μια ποικιλία από χαρακτηριστικά και προσφέρουν πολλαπλές υπηρεσίες. Αποτελούν έτσι ισχυρά εργαλεία, που μπορούν να προάγουν την έρευνα πάνω σε πρωτόκολλα δικτύων έξυπνης σκόνης. Επιπρόσθετα αναπτύξαμε ένα πρότυπο πραγματικό πειραματικό δίκτυο έξυπνης σκόνης από MICΑ motes, έχοντας παράλληλα τη δυνατότητα να αξιολογήσουμε τις επιδόσεις ενός σύγχρονου κόμβου έξυπνης σκόνης και να εξοικειωθούμε με το TinyOS και τις εφαρμογές του. Η συνολική εμπειρία που αποκομίσαμε έχει καταγραφεί στο παρόν κείμενο και ελπίζουμε να αποτελέσει σημαντική βοήθεια σε όποιο ερευνητή επιθυμεί να ασχοληθεί μελλοντικά με το δυναμικά εξελισσόμενο τομέα των Smart Dust.

Τα τελευταία λόγια ενός κειμένου έρευνας πάνω στη σχετικά νέα τεχνολογία των δικτύων έξυπνης σκόνης θα είναι η δική μας ευχή για τη μελλοντική τους εξέλιξη. Διαβλέποντας λοιπόν τη μελλοντική τους καθιέρωση στην καθημερινότητα μας, μια και πρόκειται για ένα κλάδο δικτύων με πολλαπλές εφαρμογές, ευχόμαστε τα δίκτυα έξυπνης σκόνης να χρησιμοποιηθούν στη συντριπτική τους πλειοψηφία στην προαγωγή των συνιθηκών διαβίωσης του ανθρώπου και γενικότερα να ταχθούν σε σκοπούς που δε θα αντιστρατεύονται την ανθρώπινη ζωή και αξιοπρέπεια. Γιατί τελικά μια τεχνολογία με τις δυνατότητες των δικτύων έξυπνης σκόνης, αν υπηρετήσει ειρηνικούς και αναπτυξιακούς σκοπούς, τα οφέλη θα είναι εντυπωσιακά για την πρόοδο της ανθρώπινης κοινωνίας.

“We shall not cease from exploration, and the end of all our exploring will be to arrive where we started and know the place for the first time” , T.S. Elliot

Βιβλιογραφία

- [1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci: Wireless sensor networks: a survey. In the Journal of Computer Networks, Volume 38, pp. 393-422, 2002.
- [2] G. Hoblos, M. Staroswiecki, A. Aitouche: Optimal design of fault tolerant sensor networks, IEEE International Conference on Control Applications, Anchorage, AK. September 2000, pp. 467-472
- [3] N. Bulusu, D. Estrin, L. Girod, J. Heidemann: Scalable coordination for wireless sensor networks: self-configuring localization systems, International Symposium on Communication Theory and Applications (ISCTA 2001), Ambleside, UK, July 2001.
- [4] C. Schurgers, V. Tsiatsis, S. Ganeriwal and M Srivastava: Topology Management for Sensor Networks: Exploiting Latency and Density MOBIHOC'02, EPFL, Lausanne, Switzerland, June 2002
- [5] I. Chatzigiannakis, S. Nikoletseas and P. Spirakis: Smart Dust Protocols for Local Detection and Propagation. In *Proc. 2nd ACM Workshop on Principles of Mobile Computing – POMC'2002*.
- [6] I. Chatzigiannakis, T. Dimitriou, S. Nikoletseas and P. Spirakis: A Probabilistic Forwarding Protocol for Efficient Data Propagation in Sensor Networks. FLAGS Technical Report, FLAGS-TR-14, 2003.
- [7] I. Chatzigiannakis and S. Nikoletseas: A Sleep-Awake Protocol for Information Propagation in Smart Dust Networks. In Proc. 3rd Workshop on Mobile and Ad-Hoc Networks (WMAN), IPDPS Workshops, IEEE Press, 2003.
- [8] S. Nikoletseas, I. Chatzigiannakis, H. Euthimiou, A. Kinalis, A. Antoniou, G. Mylonas: Energy Efficient Protocols for Sensing Multiple Events in Smart Dust Networks, Invited Paper to the IEEE International Workshop on Mobility Management and Wireless Access (MobiWac 2003) to appear in the proceedings, Orlando, 2003.
- [9] D. Estrin, R. Govindan, J. Heidemann and S. Kumar: Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proc. 5th ACM/IEEE International Conference on Mobile Computing – MOBICOM'1999*.
- [10] S.E.A. Hollar: COTS Dust. Msc. Thesis in Engineering-Mechanical Engineering, University of California, Berkeley, USA, 2000.
- [11] 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*.
- [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] C. Intanagonwiwat, D. Estrin, R. Govindan and J. Heidemann: Impact of Network Density on Data Aggregation in Wireless Sensor Networks. Technical Report 01-750, University of Southern California Computer Science Department, November, 2001.
- [14] J.M. Kahn, R.H. Katz and K.S.J. Pister: Next Century Challenges: Mobile Networking for Smart Dust. In *Proc. 5th ACM/IEEE International Conference on Mobile Computing*, pp. 271-278, September 1999.

- [15] 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*.
- [16] K. Mehlhorn and S. Näher: LEDA: A Platform for Combinatorial and Geometric Computing. *Cambridge University Press*, 1999.
- [17] C.E. Perkins: Ad Hoc Networking. *Addison-Wesley*, Boston, USA, January, 2001.
- [18] W. Ye, J. Heidemann and D. Estrin: An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proc. 12th IEEE International Conference on Computer Networks – INFOCOM’2002*.
- [19] TinyOS: A Component-based OS for the Network Sensor Regime. <http://webs.cs.berkeley.edu/tos/>, October, 2002.
- [20] R. Szewczyk, A. Woo, S. Hollar, D. Culler, K. Pister: System Architecture Directions for Networked Sensors Jason Hill, Dpt of Electrical Engineering and Computer Sciences University of California, Berkeley, USA, 2000.
- [21] D. Gay, D. Culler, P. Levis: nesC Language Reference Manual. September 2002.
- [22] μ -Adaptive Multi-domain Power aware Sensors: <http://www-mtl.mit.edu/research/icsystems/uamps>,
- [23] Wireless Integrated Sensor Networks: <http://www.janet.ucla.edu/WINS/>, October, 1999.
- [24] Wireless Embedded Systems: <http://webs.cs.berkeley.edu/index.html>, June, 2000.
- [25] SenoNet Project: Protocols for Sensor Networks <http://users.ece.gatech.edu/%7Eweilian/Sensor/>, June, 2000.
- [26] AWAIRS: Reconnaissance, Surveillance, and Target Acquisition in Small Unit Operations <http://www.janet.ucla.edu/awairs/>
- [27] SCADDS: Scalable Coordination Architectures for Deeply Distributed Systems USC/ISI <http://www.isi.edu/scadds/>, October, 2002.
- [28] NEST: An Open Experimental software/hardware Platform for Network Embedded Systems Technology research <http://webs.cs.berkeley.edu/nest-index.html>
- [29] Low Power Wireless Integrated Microsensors: <http://www.janet.ucla.edu/WINS/lwim-innovative.htm#the%201994%20LWIM%20Proposal> 1994.
- [30] MEMS: Microelectromechanical Systems <http://www.darpa.mil/mto/mems/>, June, 2000.
- [31] J. Rabaey, M. Ammer, Julio da Silva Jr., D. Shad: PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. Roundy University of California, Berkeley. July, 2000.
- [32] Smart Dust, Autonomous sensing and communication in a cubic millimeter: <http://robotics.eecs.berkeley.edu/pister/SmartDust/>, July, 2001.
- [33] Smart Dust Prototypes: <http://www-bsac.eecs.berkeley.edu/warneke/SmartDust/>, November, 2001.
- [34] M. Horton, Crossbow Technology, Inc. D. Culler, K. Pister, J. Hill, R. Szewczyk, and Alec Woo, U.C. Berkeley: MICA: The Commercialization of Microsensor Motes Sensors Online Magazine April, 2002.
- [35] ATMEL ATmega128(L): Technical Specifications by Atmel February, 2003
- [36] Specs COTS motes <http://www.cs.berkeley.edu/jhill/spec/index.htm>
- [37] Crossbow Electronics: Smarter Sensors in Silicon <http://www.xbow.com/>
- [38] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson: Wireless Sensor Networks for Habitat Monitoring, ACM Int. Workshop on Wireless Sensor Networks and Applications, Atlanta GA, September 2002.

- [39] External Antenna on Mica boards
http://webs.cs.berkeley.edu/tos/hardware/design/ORCAD_FILES/MICA/antenna.html
- [40] RF Monolithics TR1000 916,50 *MHz* Hybrid Transceiver: Technical Specifications January, 2000.
- [41] MICA WIRELESS MEASUREMENT SYSTEM: Technical Specifications by Crossbow Electronics
- [42] S Madden, J Hellerstein, W Hong: TinyDB: In-Network Query Processing in TinyOS (Version 2.0). September 26, 2002