



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ  
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ  
«ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ»

---

**ΕΞΟΙΚΟΝΟΜΗΣΗ ΕΝΕΡΓΕΙΑΣ ΣΕ ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ  
ΑΙΣΘΗΤΗΡΩΝ ΜΕΣΩ ΕΝΕΡΓΕΙΑΚΑ ΑΠΟΔΟΤΙΚΗΣ  
ΣΥΣΤΑΔΟΠΟΙΗΣΗΣ**

---

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΑΤΡΙΒΗ

Εμμανουήλ Κρικέλης

Υπεύθυνος Καθηγητής

Χρήστος Δουληγέρης, Καθηγητής

Συνεργάτης

Στέφανος Νικολιδάκης, Υπ. Διδάκτωρ

Νοέμβριος 2014

Πανεπιστήμιο Πειραιώς

Εξοικονόμηση Ενέργειας σε Ασύρματα Δίκτυα  
Αισθητήρων Μέσω Ενεργειακά Αποδοτικής  
Συσταδοποίησης

Συγγραφέας:  
Εμμανουήλ Κρικέλης  
ΜΠΣΠ:12034

Επιβλέπων:  
Χρήστος Δουληγέρης  
Καθηγητής

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

Χρήστος Δουληγέρης  
Καθηγητής

(υπογραφή)

Δημήτριος Βέργαδος  
Επ. Καθηγητής

(υπογραφή)

Μιχαήλ Ψαράκης  
Επ. Καθηγητής

Πανεπιστήμιο Πειραιώς

Περίληψη	iii
Abstract	v
Ευχαριστίες	vii
Κατάλογος Σχημάτων	ix
<b>1. Εισαγωγή</b>	<b>1</b>
1.1. Ασύρματα Δίκτυα Αισθητήρων . . . . .	1
1.1.1. Εφαρμογές . . . . .	2
1.1.2. Χαρακτηριστικά . . . . .	2
1.2. Πρωτόκολλα Δρομολόγησης . . . . .	3
<b>2. Ιεραρχικά Πρωτόκολλα</b>	<b>7</b>
2.1. Μοντέλο Ενέργειας . . . . .	7
2.2. Πρωτόκολλο Leach . . . . .	9
2.2.1. Διαδικασία Συσταδοποίησης . . . . .	9
2.2.2. Δρομολόγηση Πληροφορίας . . . . .	10
2.3. Γνωστά Πρωτόκολλα Δρομολόγησης . . . . .	11
<b>3. Μέθοδος Εξοικονόμησης Ενέργειας</b>	<b>15</b>
3.1. Εισαγωγή . . . . .	15
3.2. Δημιουργία Συστάδων . . . . .	16
3.2.1. Έλεγχος Αποστάσεων Αισθητήρων . . . . .	17
3.2.2. Σχηματισμός Τελικών Συστάδων . . . . .	20
3.3. Διαδικασία Δρομολόγησης . . . . .	23
3.3.1. Επιλογή Cluster Head . . . . .	24
3.3.2. Συγκέντρωση Πληροφορίας . . . . .	25

## Περιεχόμενα

3.3.3. Αλγόριθμος Δρομολόγησης . . . . .	27
3.4. Διαγράμματα Ροής ΜΕΕΑΔΑ . . . . .	33
3.5. Δημιουργία TDMA Προγράμματος . . . . .	37
3.6. Συμπεράσματα . . . . .	39
<b>4. Έλεγχος Επίδοσης</b>	<b>41</b>
4.1. Υλοποίηση ΜΕΕΑΔΑ . . . . .	41
4.1.1. Παράμετροι Δικτύου . . . . .	42
4.1.2. Λειτουργία Προσομοίωσης ΜΕΕΑΔΑ . . . . .	42
4.2. Διάρκεια Ζωής του Δικτύου . . . . .	46
4.3. Μέση Κατανάλωση Ενέργειας . . . . .	51
4.4. Συμπεράσματα Επιδόσεων . . . . .	54
4.5. Βελτίωση πρωτοκόλλου ΜΕΕΑΔΑ . . . . .	55
4.5.1. Αλλαγή Δρομολόγησης Πληροφορίας . . . . .	55
4.5.2. Επιδόσεις Προτεινόμενης Βελτίωσης . . . . .	56
4.5.3. Συμπεράσματα Προτεινόμενης Βελτίωσης . . . . .	60
<b>5. Συμπεράσματα</b>	<b>61</b>
5.1. Προτεινόμενο Πρωτόκολλο Δρομολόγησης . . . . .	61
5.2. Αξιολόγηση Επιδόσεων . . . . .	62
<b>Α'.Επιπλέον Υλικό</b>	<b>Π-1</b>
Α'.1. Παραδείγματα Συσταδοποίησης ΜΕΕΑΔΑ . . . . .	Π-1
Α'.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας . . . . .	Π-5
<b>Βιβλιογραφία</b>	<b>Β-1</b>
<b>Ευρετήριο</b>	<b>Ε-1</b>

Σε αυτή την μεταπτυχιακή διατριβή θα μελετηθούν πρωτόκολλα δρομολόγησης για την αποτελεσματική διαχείριση της ενέργειας σε ασύρματα δίκτυα αισθητήρων (ΑΔΑ). Ένα ΑΔΑ αποτελείται από κόμβους οι οποίοι τροφοδοτούνται με ενέργεια μέσω μίας μπαταρίας περιορισμένης διάρκειας. Για αυτόν τον λόγο, η αποτελεσματική διαχείριση της διαθέσιμης ενέργειας μπορεί να συμβάλει στην αύξηση της συνολικής διάρκειας ζωής του δικτύου. Υπάρχουν αρκετά πρωτόκολλα δρομολόγησης που προσπαθούν να διαχειριστούν την λειτουργία των κόμβων με ενεργειακά αποτελεσματικό τρόπο. Μετά από μία σύντομη εισαγωγή στα ΑΔΑ και στα πιο γνωστά πρωτόκολλα, θα παρουσιαστεί ένα πρωτόκολλο για την εξοικονόμηση ενέργειας σε ασύρματα δίκτυα αισθητήρων. Η απόδοσή του αξιολογείται σε σχέση με το πρωτόκολλο LEACH (Low-Energy Adaptive Clustering Hierarchy). Τα αποτελέσματα των προσομοιώσεων δείχνουν ότι το προτεινόμενο πρωτόκολλο υπερτερεί του LEACH, σε ό,τι αφορά την διάρκεια ζωής του δικτύου και την μέση κατανάλωση ενέργειας.

**Λέξεις κλειδιά:** Ασύρματα Δίκτυα Αισθητήρων, Πρωτόκολλα Δρομολόγησης, Ενεργειακή Απόδοση.

Πανεπιστήμιο Πειραιώς



## ABSTRACT

In this Master's thesis, energy efficient routing protocols in wireless sensor networks (WSNs) are presented. A WSN consists of battery powered sensor nodes with limited energy resources. For this reason, energy efficiency is essential in order to increase network lifetime. Several routing protocols have been proposed to manage the operation of nodes in an energy efficient manner. After a brief introduction to WSNs and to the most known protocols, we present a protocol for energy efficient routing in wireless sensor networks. Its performance is compared with that of the well known hierarchical protocol LEACH (Low-Energy Adaptive Clustering Hierarchy). Simulation results show that the proposed protocol outperforms LEACH in terms of network lifetime and average energy consumption.

**Keywords:** Wireless Sensor Networks, Routing Protocols, Energy Efficiency.

Πανεπιστήμιο Πειραιώς

## ΕΥΧΑΡΙΣΤΙΕΣ

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντά μου Καθηγητή Χρήστο Δουληγέρη, για τις χρήσιμες συμβουλές του και για τον χρόνο που αφιέρωσε σε αυτή την εργασία.

Επίσης, θα ήθελα να ευχαριστήσω ιδιαίτερα τον υποψήφιο διδάκτορα Στέφανο Νικολιδάκη, για την καθοδήγηση και την συνεργασία του, καθ' όλη την διάρκεια της συγγραφής αυτής της εργασίας.

Επιπλέον, θα ήθελα να ευχαριστήσω τους καθηγητές του μεταπτυχιακού προγράμματος σπουδών Προηγμένα Συστήματα Πληροφορικής για την πολύτιμη γνώση που μου πρόσφεραν.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου, που με στηρίζει σε κάθε μου βήμα.

Πανεπιστήμιο Πειραιώς

## ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ

1.1. Κατηγορίες Πρωτοκόλλων Δρομολόγησης. . . . .	4
2.1. Βασικές παράμετροι του μοντέλου ενέργειας που υιοθετείται. . . . .	8
2.2. Παράδειγμα λειτουργίας του πρωτοκόλλου LEACH. . . . .	11
3.1. Παράδειγμα εκτέλεσης του αλγορίθμου ελέγχου αποστάσεων. . . . .	19
3.2. Παράδειγμα εκτέλεσης των προτύπων διάσχισης. . . . .	21
3.3. Πρόγραμμα TDMA για την συλλογή της πληροφορίας σε μία συστάδα. . . . .	26
3.4. Υπολογισμός μονοπατιού δρομολόγησης. . . . .	28
3.5. Δρομολόγηση μη αρνητικής απόστασης. . . . .	31
3.6. Διάγραμμα ροής με τις κύριες λειτουργίες της ΜΕΕΑΔΑ. . . . .	33
3.7. Διάγραμμα ροής για την επιλογή του κόμβου Primary Head. . . . .	34
3.8. Διάγραμμα ροής για την επιλογή του κόμβου Secondary Head. . . . .	35
3.9. Διάγραμμα ροής για την επιλογή του κόμβου Territorial Head. . . . .	36
3.10. Υπολογισμοί προτεραιοτήτων (P) των συστάδων αποστολέων (S) και των παραληπτών (D). . . . .	38
4.1. Πίνακας με τους κωδικούς (κλειδιά) 100 κόμβων σε δίκτυο διαστάσεων $50m \times 50m$ , μετά την εκτέλεση της συσταδοποίησης. . . . .	43
4.2. Σχέδιο δρομολόγησης για την συστάδα με κωδικό 2 του Σχήματος 4.1. . . . .	45
4.3. Διάρκεια ζωής του δικτύου μεγέθους $50m \times 50m$ και $0.25J/κόμβο$ . . . . .	48
4.4. Διάρκεια ζωής του δικτύου μεγέθους $100m \times 100m$ και $0.25J/κόμβο$ . . . . .	48
4.5. Απόδοση ΜΕΕΑΔΑ για διαφορετικά μεγέθη δικτύου και $0.5J/κόμβο$ . . . . .	49
4.6. Η απόδοση της ΜΕΕΑΔΑ για διαφορετική θέση του σταθμού βάσης και $0.5J/κόμβο$ . . . . .	49
4.7. Αποτελέσματα προσομοιώσεων για διαφορετικό μέγεθος πακέτου, σε δίκτυο διαστάσεων $100m \times 100m$ με 100 κόμβους και $1J/κόμβο$ . . . . .	50
4.8. Απόδοση ΜΕΕΑΔΑ για διαφορετική αρχική ενέργεια 500 κόμβων, σε ένα δίκτυο διαστάσεων $100m \times 100m$ . . . . .	50

## Κατάλογος Σχημάτων

4.9. Μέση κατανάλωση ενέργειας για δίκτυα μεγέθους $50m \times 50m$ , 100 κόμβους και $0.25J/\text{κόμβο}$ . . . . .	52
4.10. Μέση κατανάλωση ενέργειας για δίκτυα μεγέθους $100m \times 100m$ , 100 κόμβους και $0.25J/\text{κόμβο}$ . . . . .	52
4.11. Μέση κατανάλωση ενέργειας για διαφορετικό μέγεθος δικτύου, 100 κόμβους και αρχική ενέργεια $0.5J/\text{κόμβο}$ . . . . .	53
4.12. Μέση κατανάλωση ενεργείας για διαφορετικό μέγεθος πακέτου, $1J/\text{κόμβο}$ , σε δίκτυο διαστάσεων $100m \times 100m$ . . . . .	53
4.13. Μέση κατανάλωση ενεργείας για διαφορετικό αριθμό κόμβων και ενέργεια $0.25J/\text{κόμβο}$ Η περιοχή εφαρμογής έχει διαστάσεις $100m \times 100m$ . . . . .	54
4.14. Διάρκεια ζωής του δικτύου μεγέθους $50m \times 50m$ , 100 κόμβων και ενέργεια $0.25J/\text{κόμβο}$ (με την ΒΜΕΕΑΔΑ). . . . .	58
4.15. Διάρκεια ζωής του δικτύου μεγέθους $100m \times 100m$ , 100 κόμβων και $0.25J/\text{κόμβο}$ (με την ΒΜΕΕΑΔΑ). . . . .	58
4.16. Μέση κατανάλωση ενέργειας για δίκτυα μεγέθους $50m \times 50m$ (με την ΒΜΕΕΑΔΑ). . . . .	59
4.17. Μέση κατανάλωση ενέργειας για δίκτυα μεγέθους $100m \times 100m$ , 100 κόμβους και $0.25J/\text{κόμβο}$ (με την ΒΜΕΕΑΔΑ). . . . .	59
A'.1. Πίνακας με τους κωδικούς (κλειδιά) 50 κόμβων σε δίκτυο διαστάσεων $15m \times 50m$ , μετά την εκτέλεση της συσταδοποίησης. . . . .	Π-1
A'.2. Πίνακας με τους κωδικούς (κλειδιά) 75 κόμβων σε δίκτυο διαστάσεων $30m \times 30m$ , μετά την εκτέλεση της συσταδοποίησης. . . . .	Π-2
A'.3. Πίνακας με τους κωδικούς (κλειδιά) 100 κόμβων στο κέντρο του δικτύου διαστάσεων $50m \times 50m$ , μετά την εκτέλεση της συσταδοποίησης. . . . .	Π-3
A'.4. Πίνακας με τους κωδικούς (κλειδιά) 200 κόμβων σε δίκτυο $50m \times 50m$ . . . . .	Π-4

Στην εργασία αυτή θα παρουσιάσουμε μία μέθοδο με την οποία μπορούμε να εξοικονομήσουμε ενέργεια σε ασύρματα δίκτυα αισθητήρων – Wireless Sensor Networks (ΑΔΑ ή WSNs). Τα ασύρματα δίκτυα αισθητήρων είναι δίκτυα τα οποία αποτελούνται από συσκευές που ονομάζουμε κόμβους (nodes ή motes). Ένα δίκτυο ασύρματων αισθητήρων αποτελείται από λίγους, μερικές εκατοντάδες ή και περισσότερους κόμβους, που τοποθετούνται συνήθως με τυχαίο τρόπο σε μία περιοχή. Η υλοποίηση ενός WSN μπορεί να γίνει στο έδαφος ή την θάλασσα, πάντοτε σύμφωνα με τις εκάστοτε απαιτήσεις της εφαρμογής για την οποία προορίζεται.

## 1.1. Ασύρματα Δίκτυα Αισθητήρων

Η ανάγκη για την δημιουργία δικτύων σε περιοχές όπου τα παραδοσιακά ενσύρματα η ασύρματα δίκτυα απαιτούν υψηλό κόστος και χρόνο εγκατάστασης, οδήγησε στην υιοθέτηση των ασύρματων δικτύων αισθητήρων. Οι κόμβοι του δικτύου μας παρέχουν την δυνατότητα να ελέγχουμε διάφορες ιδιότητες του περιβάλλοντος [1] όπως θερμοκρασία, ατμοσφαιρική πίεση, ήχο, κίνηση, δονήσεις, αέρα. Η πληροφορία που συλλέγεται από τους αισθητήριους κόμβους συγκεντρώνεται σε μία συγκεκριμένη τοποθεσία<sup>1</sup> όπου και μπορούμε να την διαχειριστούμε. Τα σύγχρονα WSNs και οι εφαρμογές αυτών, αποτελούν σήμερα ένα από τα σημαντικότερα ερευνητικά θέματα στην επιστήμη της πληροφορικής [2], [3], [4], [5].

Το μέγεθος των κόμβων, δεν ξεπερνά τα μερικά εκατοστά ενώ σε ερευνητικό επίπεδο γίνονται προσπάθειες για την κατασκευή αισθητήριων κόμβων με μέγεθος που να μην μπορεί να αντιληφθεί εύκολα το ανθρώπινο μάτι. Οι ασύρματοι κόμβοι διαθέτουν κεραία για την αποστολή και λήψη δεδομένων, έναν μικροελεγκτή και ένα ηλεκτρονικό κύκλωμα για την διασύνδεση με ένα ή περισσότερους αισθητήρες. Η τροφοδοσία τους

<sup>1</sup>Οι όροι προορισμός, σταθμός βάσης ή base station θα χρησιμοποιούνται για να περιγράψουν την τοποθεσία όπου συλλέγεται η πληροφορία ενός WSN.

## 1. Εισαγωγή

με ενέργεια γίνεται μέσω μίας μπαταρίας (πεπερασμένης διάρκειας) η οποία βρίσκεται ενσωματωμένη στην συσκευή. Οι σύγχρονοι κόμβοι μπορεί να θεωρηθούν ως μικροί υπολογιστές, καθώς διαθέτουν επεξεργαστική ισχύ που τους επιτρέπει την εκτέλεση αυτοματοποιημένων διαδικασιών [6], [7].

### 1.1.1. Εφαρμογές

Η χρήση των ασύρματων δικτύων αισθητήρων ξεκίνησε κυρίως για στρατιωτικούς σκοπούς και την ανίχνευση εχθρικών στρατευμάτων μέσω αλλαγών στην θερμοκρασία του περιβάλλοντος. Στο [8] παρουσιάζεται ένα σύστημα που μπορεί να χρησιμοποιηθεί για στρατιωτικούς σκοπούς μέσω της ανίχνευσης εισβολών (intrusion detection) [9]. Κάποιες εξίσου σημαντικές εφαρμογές αφορούν την πρόληψη περιβαλλοντικών καταστροφών. Ένα παράδειγμα αποτελεί η παρακολούθηση των δασών για την άμεση ανίχνευση πυρκαγιών [10]. Αυτό επιτυγχάνεται με την χρήση κατάλληλων αισθητήρων, οι οποίοι καθιστούν δυνατή την παρακολούθηση της θερμοκρασίας και των αερίων που παράγονται από μία πυρκαγιά. Επιπλέον, υπάρχουν προσπάθειες για την εκμετάλλευση των WSNs έτσι ώστε να είναι δυνατή η έγκαιρη ειδοποίηση για επικίνδυνα θαλάσσια φαινόμενα όπως ένα τσουνάμι [11].

Εφαρμογές των ασύρματων δικτύων αισθητήρων συναντάμε και στον έλεγχο της ατμοσφαιρικής ρύπανσης [12] στις μεγάλες πόλεις. Λαμβάνοντας μετρήσεις ανά τακτά χρονικά διαστήματα μπορούμε να διαπιστώσουμε ποιες από τις αέριες ουσίες [13] ξεπερνούν τα επιτρεπτά όρια και θέτουν σε κίνδυνο την ζωή των πολιτών. Πέρα από την περιβαλλοντική τους χρησιμότητα [14] ιδιαίτερα σημαντική είναι και η χρήση τους στον τομέα της ιατρικής [15], [16]. Με την χρήση κατάλληλων αισθητήρων που μετρούν ένα σύνολο σωματικών λειτουργιών, είναι δυνατή η ασύρματη παρακολούθηση ανθρώπων μεγάλης ηλικίας και ασθενών με χρόνιες παθήσεις [17]. Η χρήση τους θα μπορούσε να επεκταθεί και στον αγροτικό τομέα όπου θα είναι δυνατή η παρακολούθηση κρίσιμων παραγόντων όπως η θερμοκρασία του περιβάλλοντος και η υγρασία του εδάφους [18]. Με τον τρόπο αυτό μπορούμε να λάβουμε αποφάσεις που θα οδηγήσουν στην βελτιστοποίηση της παραγωγής.

### 1.1.2. Χαρακτηριστικά

Τα ασύρματα δίκτυα αισθητήρων προσφέρουν την δυνατότητα υλοποίησης δικτύων με μεγάλο αριθμό κόμβων και χαμηλό κόστος. Οι κόμβοι του δικτύου διαθέτουν λειτουργικό σύστημα [19], [20] και μπορούν να λειτουργήσουν αυτόνομα. Η χαμηλή πολυπλοκότητα διαχείρισης των πληροφοριών που αυτά συλλέγουν, τα καθιστούν ιδανικά για εφαρμογές σε δυσπρόσιτες περιοχές. Σε σχέση με τα υπόλοιπα δίκτυα επικοινωνιών, τα WSNs διαθέτουν τα ακόλουθα χαρακτηριστικά:

- *Εξειδικευμένα για μία εφαρμογή:* Τα δίκτυα ασύρματων αισθητήρων μπορεί να χρησιμοποιηθούν για μία συγκεκριμένη εφαρμογή.
- *Κλίμακα μεγέθους και πυκνότητα:* Συνήθως ένα WSN αποτελείται από ένα μεγάλο αριθμό κόμβων. Επίσης, η πυκνότητα των κόμβων που τοποθετούνται σε



## 1.2. Πρωτόκολλα Δρομολόγησης

μία περιοχή είναι ιδιαίτερα υψηλή.

- *Λάθη και Πλεονασμός Δεδομένων*: Λόγω της ανάπτυξής τους στο ασύρματο περιβάλλον οι κόμβοι του δικτύου είναι επιρρεπείς στα λάθη. Επιπλέον, λόγω της υψηλής πυκνότητάς τους, υπάρχει πιθανότητα η πληροφορία μεταξύ γειτονικών κόμβων να μην διαφέρει σημαντικά. Το γεγονός αυτό οδηγεί στην συγκέντρωση και αποστολή πλεονάζουσας πληροφορίας. Ωστόσο, η επιπλέον πληροφορία μπορεί να χρησιμοποιηθεί για την αντιμετώπιση τυχόν σφαλμάτων.
- *Χρόνος ζωής*: Οι κόμβοι του δικτύου έχουν μικρό μέγεθος και τροφοδοτούνται μέσω μίας μπαταρίας. Αυτό σημαίνει ότι έχουν περιορισμένη υπολογιστική ικανότητα, εύρος ζώνης και διάρκεια ζωής.
- *Ποιότητα Υπηρεσιών / QoS*: Η ποιότητα των υπηρεσιών αφορά την ακρίβεια και την χωρίς καθυστερήσεις λήψη των πληροφοριών που συλλέγονται από το δίκτυο. Η ακρίβεια των λαμβανόμενων πληροφοριών εξαρτάται άμεσα από το πλήθος και από την πυκνότητα των ασύρματων κόμβων στον χώρο του δικτύου. Η χωρίς καθυστερήσεις μετάδοση, μπορεί να οδηγήσει στην γρήγορη και αποτελεσματική λήψη αποφάσεων.
- *Δημιουργία Δικτύου*: Η δημιουργία ενός WSN είναι σχετικά απλή και γίνεται με μικρό κόστος. Οι κόμβοι τοποθετούνται τυχαία σε μία περιοχή και η πυκνότητά τους μπορεί να είναι υψηλότερη σε σημεία με μεγαλύτερη κρισιμότητα.
- *Δυναμικές τοπολογίες*: Η τοπολογία ενός WSN καθορίζεται και αλλάζει σύμφωνα με τους κόμβους που το αποτελούν. Ένας κόμβος μπορεί να τεθεί εκτός δικτύου ανά πάσα στιγμή λόγω κάποια δυσλειτουργίας ή της περιορισμένης ενέργειας του. Επίσης, μπορεί να έχουμε την προσθήκη νέων κόμβων.

Η εφαρμογή των WSNs στο πραγματικό περιβάλλον θα πρέπει να γίνεται λαμβάνοντας υπόψη κάποιους παράγοντες επίδοσης/λειτουργίας. Κάποιοι από τους παράγοντες αυτούς είναι η διάρκεια ζωής των κόμβων του δικτύου [21] και το κόστος υλοποίησης σε σχέση με τα υπόλοιπα ασύρματα ή ενσύρματα δίκτυα. Επίσης, θα πρέπει να ληφθούν υπόψη και τα μειονεκτήματα των ασύρματων επικοινωνιών και οι συνθήκες του περιβάλλοντος [22] (θόρυβος, παρεμβολές), που μπορούν να μειώσουν την ποιότητα των υπηρεσιών του δικτύου.

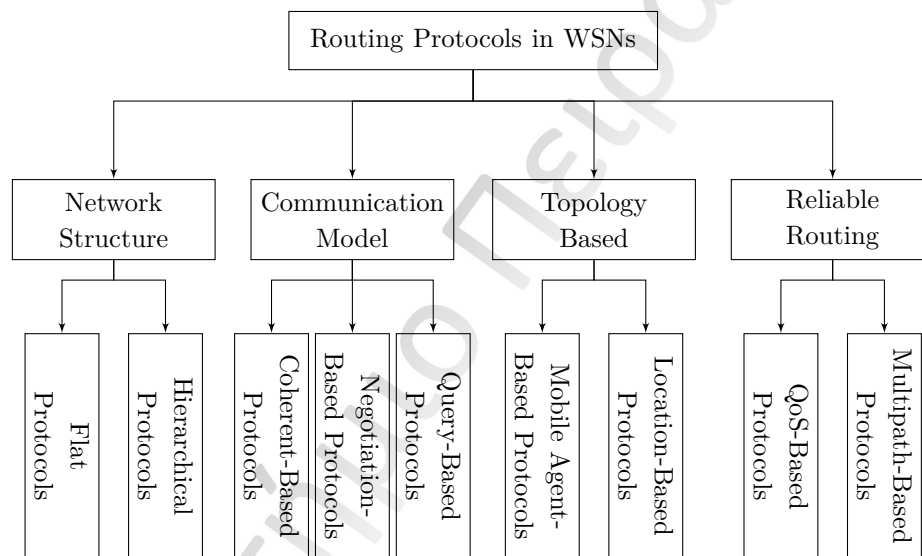
## 1.2. Πρωτόκολλα Δρομολόγησης

Η πληθώρα νέων πιθανών εφαρμογών των WSNs, καθιστά επιτακτική την ανάγκη της αποτελεσματικής διαχείρισής τους [23]. Σε γενικές γραμμές αξιολογούμε την αποδοτικότητα των δικτύων σε δύο κατηγορίες. Η πρώτη αφορά στην ποιότητα των υπηρεσιών QoS [24] και η δεύτερη στην διάρκεια ζωής του δικτύου [25]. Οι δύο αυτές κατηγορίες είναι αλληλοεξαρτώμενες και πολλές φορές θα πρέπει να γίνουν υποχωρήσεις προκειμένου να βελτιωθεί κάποια από τις δύο. Η σημαντικότητα της κάθε κατηγορίας καθορίζεται

## 1. Εισαγωγή

από την εκάστοτε εφαρμογή του δικτύου. Για κρίσιμες εφαρμογές [26], όπου είναι επιθυμητή η έγκαιρη και έγκυρη ενημέρωση, θα πρέπει ενδεχομένως να μειώσουμε την διάρκεια ζωής του δικτύου αυξάνοντας την συχνότητα αποστολής δεδομένων από τους αισθητήριους κόμβους.

Για να βελτιστοποιήσουμε την απόδοση ενός WSN παρέχοντας υψηλή ποιότητα υπηρεσιών και μεγαλύτερη διάρκεια ζωής, χρησιμοποιούμε τα πρωτόκολλα δρομολόγησης. Ένα πρωτόκολλο δρομολόγησης περιλαμβάνει τεχνικές και μεθόδους μέσα από τις οποίες αποφασίζεται ο τρόπος λειτουργίας του δικτύου. Η χρήση των πρωτοκόλλων αυτών περιλαμβάνει τον σχηματισμό του δικτύου βάση των παρεχόμενων κόμβων, την συγέντρωση της πληροφορίας από κάθε κόμβο και τον τρόπο αποστολής της πληροφορίας ολόκληρου του δικτύου στον προορισμό. Τα πρωτόκολλα μπορούμε να τα διαχωρίσουμε, ανάλογα με τις ιδιότητές τους [27], σε τέσσερις βασικές κατηγορίες: *Network Structure*, *Communication Model*, *Topology Based* και *Reliable Routing*.



Σχήμα 1.1.: Κατηγορίες Πρωτοκόλλων Δρομολόγησης [27].

Κάθε μία από τις κατηγορίες των πρωτοκόλλων (Σχήμα 1.1) χωρίζεται σε υποκατηγορίες. Στην εργασία αυτή θα μελετηθούν τα Ιεραρχικά Πρωτόκολλα, τα οποία αποτελούν μέλος της κατηγορίας *Network Structure*. Πιο συγκεκριμένα, στο Κεφάλαιο 2 θα περιγράψουμε ένα από τα διαδομένα ιεραρχικά πρωτόκολλα για την εξοικονόμηση ενέργειας στα WSNs, με την ονομασία LEACH (*Low-Energy Adaptive Clustering Hierarchy*)[28]. Η περιγραφή του πρωτοκόλλου LEACH θα συνοδεύεται και από την περιγραφή του μοντέλου ενέργειας που χρησιμοποιεί. Το μοντέλο ενέργειας, αποτελεί μία μέθοδο με την οποία μπορούμε να μετράμε την κατανάλωση ενέργειας των αισθητήριων κόμβων του δικτύου. Επίσης, θα παρουσιαστούν συνοπτικά μερικά από τα πιο γνωστά πρωτόκολλα της βιβλιογραφίας.

Επειτα, στο Κεφάλαιο 3, θα γίνει η περιγραφή ενός νέου ιεραρχικού πρωτοκόλλου με το όνομα ΜΕΕΑΔΑ (*Μέθοδος για την Εξοικονόμηση Ενέργειας σε Ασύρματα Δίκτυα*

## 1.2. Πρωτόκολλα Δρομολόγησης

Αισθητήρων). Το πρωτόκολλο<sup>2</sup> αυτό θα υλοποιηθεί στην γλώσσα προγραμματισμού C και θα αξιολογηθεί με βάση το μοντέλο ενέργειας του πρωτοκόλλου LEACH. Στο Κεφάλαιο 4, θα συγκρίνουμε τα δύο πρωτόκολλα ως προς την διάρκεια ζωής του δικτύου και την μέση κατανάλωση ενέργειας. Η σύγκριση θα περιλαμβάνει δοκιμές με διαφορετικό αριθμό αισθητήρων και διαφορετικό μέγεθος δικτύων. Για την ορθότητα της σύγκρισης μεταξύ των δύο πρωτοκόλλων θα χρησιμοποιηθούν οι ίδιες παράμετροι δικτύου. Οι παράμετροι δικτύου περιλαμβάνουν το μοντέλο ενέργειας, το μέγεθος πακέτου, τον αριθμό των αισθητήριων κόμβων και την τοποθεσία του σταθμού βάσης. Στο Κεφάλαιο 5 παρουσιάζονται τα συμπεράσματα της απόδοσης και λειτουργίας του προτεινόμενου πρωτοκόλλου. Τέλος, στο Παράρτημα Α' της εργασίας, είναι διαθέσιμος ο κώδικας που υλοποιεί την ΜΕΕΑΔΑ.

---

<sup>2</sup>Συχνά θα αναφέρεται και ως μέθοδος με την έννοια του πρωτοκόλλου.

Πανεπιστήμιο Πειραιώς

Τα ιεραρχικά πρωτόκολλα δρομολόγησης αποτελούν μέρος της κατηγορίας πρωτοκόλλων Network Structure. Πρωταρχικός τους στόχος είναι η οργάνωση των κόμβων σε συστάδες και μετέπειτα η διαδικασία συλλογής και μετάδοσης της πληροφορίας στον τελικό προορισμό. Καθένα από τα δύο βήματα, μπορεί να συμβάλει στην αύξηση της διάρκειας ζωής και της ποιότητας των υπηρεσιών του δικτύου. Για να αξιολογήσουμε την απόδοση των πρωτοκόλλων αυτών, σε επίπεδο προσομοίωσης, χρησιμοποιούμε ένα μοντέλο ενέργειας. Αφού ορίσουμε το μοντέλο ενέργειας, θα περιγράψουμε ένα από τα πιο γνωστά ιεραρχικά πρωτόκολλα, το πρωτόκολλο LEACH.

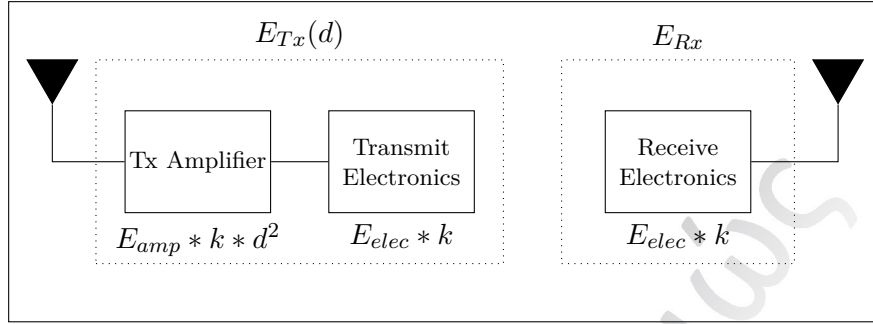
### 2.1. Μοντέλο Ενέργειας

Το μοντέλο ενέργειας που θα χρησιμοποιηθεί [28], [29], λαμβάνει υπόψιν τις λειτουργίες ενός αισθητήριου κόμβου κατά την διάρκεια της αποστολής και της λήψης δεδομένων, σε επίπεδο υλικού. Στην πραγματικότητα, εκτός από τα κυκλώματα λήψης και αποστολής, ένας κόμβος αποτελείται από περισσότερα υλικά μέρη. Τα πιο βασικά και κοινά στις περισσότερες σύγχρονες συσκευές είναι η μονάδα επεξεργασίας και τα αισθητήρια όργανα. Όλες αυτές οι διατάξεις καταναλώνουν ποσά ενέργειας που ποικίλουν ανάλογα με την εργασία που εκτελούν. Στο μοντέλο ενέργειας που θα χρησιμοποιήσουμε σε αυτή την εργασία, δεν λαμβάνουμε υπόψιν το κόστος κατανάλωσης ενέργειας των διατάξεων αυτού του είδους.

Τα πρωτόκολλα δρομολόγησης προσπαθούν να οργανώσουν τους κόμβους του δικτύου με τέτοιο τρόπο ώστε να εξοικονομήσουν σημαντικά ποσά ενέργειας. Για να πετύχουν αυτό χρησιμοποιούν επιπλέον υπολογιστικές πράξεις που επιβαρύνουν τους κόμβους. Το γεγονός αυτό οδηγεί σε ένα επιπλέον κόστος κατανάλωσης ενέργειας, που μπορεί να είναι ιδιαίτερα υψηλό. Η κατανάλωση αυτού του είδους, είναι ανεξάρτητη από αυτή της επεξεργασίας των δεδομένων που συλλέγουν τα αισθητήρια όργανα των κόμβων. Γενικότερα, σε κάθε πρωτόκολλο δρομολόγησης μπορεί να χρησιμοποιείται και

## 2. Ιεραρχικά Πρωτόκολλα

διαφορετικό μοντέλο ενέργειας. Αυτό που έχει σημασία είναι να μην υπάρχουν διαφοροποιήσεις, όταν επιδιώκεται η σύγκριση μεταξύ δύο η και περισσότερων πρωτοκόλλων.



Σχήμα 2.1.: Βασικές παράμετροι του μοντέλου ενέργειας που υιοθετείται [28].

Το μοντέλο ενέργειας περιλαμβάνει τα κυκλώματα λήψης και μετάδοσης ενός κόμβου (Σχήμα 2.1). Η ενέργεια που καταναλώνεται από το κύκλωμα μετάδοσης ή λήψης (transmitter or receiver circuitry) του κόμβου είναι 50 nJ/bit ( $E_{elec}$ ) και για τον ενισχυτή μετάδοσης (transmit amplifier) 100 pJ/bit/m<sup>2</sup> ( $E_{amp}$ ). Η αποστολή ενός μηνύματος μεγέθους  $k^1$  σε απόσταση  $d$ ,  $E_{Tx}(k, d)$ , αφορά την χρήση των κυκλωμάτων μετάδοσης  $E_{Tx-elec}(k)$  και του ενισχυτή μετάδοσης  $E_{Tx-amp}(k, d)$ . Λαμβάνοντας υπόψιν όλες τις παραμέτρους, η σχέση που περιγράφει την κατανάλωση ενέργειας ενός κόμβου κατά την αποστολή είναι:

$$E_{Tx}(k, d) = E_{Tx-elec}(k) + E_{Tx-amp}(k, d) = E_{elec} * k + E_{amp} * k * d^2 \quad (2.1)$$

Για την λήψη  $E_{Rx}(k)$  ενός μηνύματος μεγέθους  $k$ , έχουμε την χρήση των κυκλωμάτων λήψης και η κατανάλωση ενέργειας περιγράφεται από την σχέση:

$$E_{Rx}(k) = E_{Rx-elec}(k) = E_{elec} * k \quad (2.2)$$

Από την σχέση (2.1), μπορούμε να συμπεράνουμε ότι η ενέργεια που καταναλώνει ένας κόμβος κατά την αποστολή εξαρτάται από την απόσταση του κόμβου παραλήπτη. Επειδή το κόστος αποστολής αυξάνει εκθετικά σε συνάρτηση με την απόσταση, μπορούμε να εξοικονομήσουμε ενέργεια μειώνοντας τις αποστάσεις μετάδοσης. Αυτό μπορεί να επιτευχθεί υιοθετώντας μία στρατηγική πολυβηματικής δρομολόγησης (multi-hop), έναντι της μονοβηματικής (single-hop). Η multi-hop δρομολόγηση προϋποθέτει ότι η μετάδοση της πληροφορίας μεταξύ δύο κόμβων, θα υλοποιηθεί μέσα από την εκμετάλλευση ενδιάμεσων κόμβων.

Στις περιπτώσεις όπου επιλέγεται η multi-hop στρατηγική, ένας κόμβος μπορεί να λάβει  $n$  πακέτα από  $n$  κόμβους, μεγέθους  $k$ . Στην συνέχεια, για να μην αυξηθεί  $n$  φορές το μέγεθος των δεδομένων που αποστέλλονται από τον κόμβο αυτό, χρησιμοποιούμε τεχνικές όπως είναι οι συνάθροιση δεδομένων (data aggregation) ή σύντηξη δεδομένων (data fusion) και συμπίεση δεδομένων (data compression) [30], [31], [32]. Αν και

<sup>1</sup>Το μέγεθος πακέτου μετράται σε *bit*.

οι τεχνικές αυτές, τείνουν να ελαχιστοποιούν το μέγεθος της πληροφορίας, συνήθως προσθέτουν και ένα επιπλέον υπολογιστικό κόστος. Το επιπλέον κόστος υπολογίζεται σύμφωνα με το μέγεθος του πακέτου που χρησιμοποιείται στο δίκτυο.

## 2.2. Πρωτόκολλο Leach

Ο πιο απλός τρόπος για την συλλογή της πληροφορίας που συγκεντρώνεται από ένα δίκτυο WSN είναι η απ' ευθείας δρομολόγηση. Σύμφωνα με αυτή την μέθοδο, καθένας από τους αισθητήριους κόμβους του δικτύου αποστέλλει τα δεδομένα που συνέλεξε κατευθείαν στον προορισμό τους (base station). Όπως είναι φυσικό, η δρομολόγηση αυτού του είδους είναι μη αποτελεσματική, καθώς οι κόμβοι αυτοί επικοινωνούν απ' ευθείας με το προορισμό, η απόσταση του οποίου μπορεί να είναι αρκετά μεγάλη. Καθώς αυξάνεται η απόσταση, αυξάνεται και η κατανάλωση ενέργειας, με αποτέλεσμα οι κόμβοι να χάνουν την ενέργειά τους πολύ γρήγορα. Επιπλέον, η μετάδοση σε μεγάλες αποστάσεις ενισχύει και τις αδυναμίες του ασύρματου περιβάλλοντος (θόρυβος, παρεμβολές) μειώνοντας την ποιότητα την τελικής πληροφορίας – Quality of Service (QoS).

Ένα από τα θεμελιώδη πρωτόκολλα δρομολόγησης, μέσω του οποίου προτείνονται τεχνικές βελτίωσης της διάρκειας ζωής ενός WSN, είναι το πρωτόκολλο LEACH (Low-Energy Adaptive Clustering Hierarchy). Στο πρωτόκολλο αυτό θεωρείται ότι:

- Ο προορισμός ή σταθμός βάσης (base station) βρίσκεται σε σταθερή θέση και μακριά από το δίκτυο.
- Οι αισθητήριοι κόμβοι που απαρτίζουν το δίκτυο είναι ομοιογενείς (ίδια υπολογιστική ικανότητα, εύρος ζώνης) και η ενέργειά τους πεπερασμένη.

Η λειτουργία του περιλαμβάνει τον σχηματισμό συστάδων (cluster), από ένα σύνολο κόμβων οι οποίοι τοποθετούνται τυχαία σε μία περιοχή. Σε κάθε συστάδα υπάρχει ένας κόμβος ο οποίος ονομάζεται cluster head. Ο κόμβος αυτός συλλέγει όλη την πληροφορία της συστάδας στην οποία ανήκει και στην συνέχεια την προωθεί στον σταθμό βάσης. Κάθε φορά που ολοκληρώνεται η μετάδοση όλων των cluster head προς τον σταθμό βάσης έχουμε την ολοκλήρωση ενός κύκλου δικτύου (network round ή round). Με τον τρόπο αυτό, μειώνεται η κατανάλωση ενέργειας κάθε κόμβου που δεν είναι cluster head, καθώς μικραίνουν οι αποστάσεις αποστολής. Βέβαια, οι κόμβοι που επιλέγονται ως cluster head, εμφανίζουν αυξημένη κατανάλωση ενέργειας, αλλά ο αριθμός τους είναι μικρός σε σχέση με τον αριθμό των κόμβων του δικτύου.

### 2.2.1. Διαδικασία Συσταδοποίησης

Η διαδικασία της συσταδοποίησης (clustering) ξεκινά με την επιλογή των cluster head, το πλήθος των οποίων αποφασίζεται εκ των προτέρων από το σύνολο των κόμβων του δικτύου. Ένας κόμβος  $n$  "αυτό-εκλέγεται" να αναλάβει αυτή την θέση επιλέγοντας τυχαία ένα αριθμό μεταξύ του 0 και του 1. Ο αριθμός αυτός θα πρέπει να είναι μικρότερος

## 2. Ιεραρχικά Πρωτόκολλα

από ένα κατώφλι  $T(n)$  το οποίο ορίζεται από την σχέση:

$$T(n) = \begin{cases} \frac{P}{1 - P * (r \bmod \frac{1}{P})} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

οπού  $P$  το επιθυμητό ποσοστό των cluster head (π.χ.  $P = 0.05$ ),  $r$  ο αριθμός του τρέχοντος round και  $G$  το σύνολο των κόμβων που δεν έχουν εκλεγεί στα τελευταία  $1/P$  round. Κάθε κόμβος έχει πιθανότητα  $P$  να εκλεγεί ως cluster head στην αρχή της λειτουργίας του δικτύου ( $r = 0$ ). Οι κόμβοι αποκτούν ξανά την ίδια πιθανότητα μετά από  $1/P$  round.

Αφού ολοκληρωθεί η διαδικασία εκλογής, οι cluster head ενημερώνουν μέσω μηνυμάτων τους υπόλοιπους κόμβους του δικτύου (advertisement phase) για την κατάσταση τους. Στην συνέχεια, κάθε κόμβος αποφασίζει να αποστείλει τα δεδομένα του στον cluster head που θα του αποφέρει το μικρότερο κόστος κατανάλωσης ενέργειας. Συνήθως επιλέγεται ο πιο κοντινός, εκτός και αν παρεμβάλλεται κάποιο εμπόδιο (φυσικό ή τεχνητό) οπότε μπορεί να προτιμηθεί κάποιος που βρίσκεται σε μεγαλύτερη απόσταση. Κάθε κόμβος ενημερώνει τον cluster head, ώστε να γνωρίζει τον αριθμό των κόμβων που τον έχουν επιλέξει και αποτελούν από κοινού μία συστάδα.

Το σύνολο των  $C'$  κόμβων που έχουν εκλεγεί ως cluster head την χρονική στιγμή  $t_1$  εμφανίζει μεγάλη κατανάλωση ενέργειας, καθώς επιβαρύνεται με την λήψη δεδομένων από τους κόμβους όλου του cluster και την αποστολή τους στον προορισμό. Για τον λόγο αυτό, ένα νέο σύνολο  $C'$  κόμβων εκλέγεται την χρονική στιγμή  $t_1 + d$ . Η επιλογή διαφορετικού συνόλου κόμβων ως cluster head, μπορεί να αλλάξει και τις προτιμήσεις των υπόλοιπων κόμβων του δικτύου ως προς της διαδικασία επιλογής του cluster head με το μικρότερο κόστος. Ως συνέπεια του γεγονότος αυτού, έχουμε τον σχηματισμό νέων συστάδων. Η λειτουργία αυτή ονομάζεται προσαρμοστική συσταδοποίηση (adaptive clustering) και αποτελεί το κύριο χαρακτηριστικό του πρωτοκόλλου LEACH.

Όσον αφορά τον σχηματισμό των συστάδων, ο αριθμός των κόμβων που εκλέγονται ως cluster head είναι ιδιαίτερα σημαντικός. Αν είναι πολύ μικρός, τότε μπορεί να αυξηθεί η απόσταση μετάδοσης για κάποιους από τους κόμβους του δικτύου. Αν είναι πολύ μεγάλος, τότε αυξάνεται ο αριθμός των κόμβων που συλλέγουν την πληροφορία του δικτύου και την αποστέλλουν στον προορισμό. Ο αριθμός αυτός υπολογίζεται σύμφωνα με διάφορες παραμέτρους όπως η τοπολογία του δικτύου. Βασιζόμενοι σε παραμέτρους αυτού του είδους, το πλήθος  $\tilde{N}$  των cluster head που χρησιμοποιούνται στο πρωτόκολλο αυτό, σε κάθε χρονική στιγμή  $t_1$ , ορίζεται συνήθως ως το 5% του συνόλου των κόμβων του δικτύου,  $\tilde{N} = 5\%$ .

### 2.2.2. Δρομολόγηση Πληροφορίας

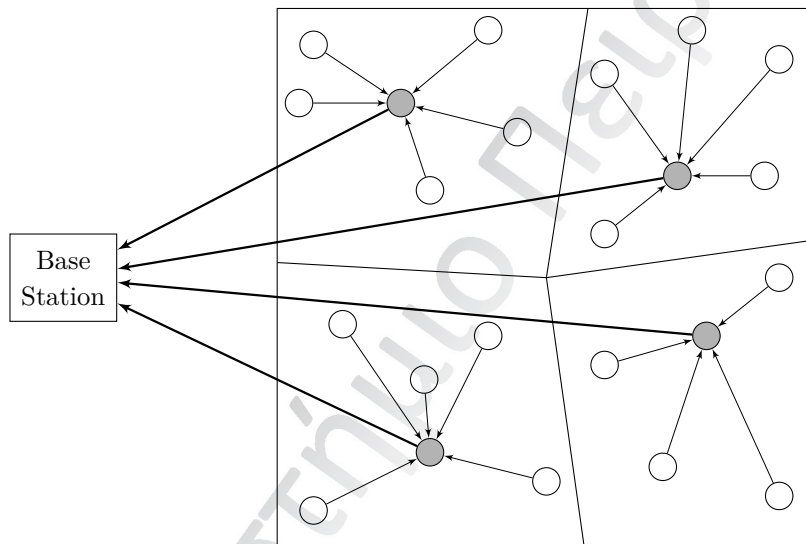
Εφόσον όλοι οι κόμβοι του δικτύου αποφασίσουν σε ποιο cluster head θα αποστείλουν την πληροφορία τους, ξεκινά η δημιουργία ενός προγράμματος TDMA (*Time Division Multiple Access*). Το πρόγραμμα αυτό καθορίζει την χρονική στιγμή που θα επικοινωνήσει κάθε κόμβος με τον cluster head που έχει επιλέξει. Ένας κόμβος συλλέγει τα δεδομένα από τα αισθητήρια όργανά του και τα μεταδίδει σύμφωνα με το πρόγραμμα.



### 2.3. Γνωστά Πρωτόκολλα Δρομολόγησης

Όταν όλοι οι κόμβοι του δικτύου προωθήσουν την πληροφορία τους στον αντίστοιχο cluster head, τότε εκείνος χρησιμοποιεί τεχνικές data aggregation και data compression ώστε να επεξεργαστεί τα δεδομένα. Με την ολοκλήρωση της επεξεργασίας κάθε cluster head δημιουργεί ένα ενιαίο πακέτο δεδομένων το οποίο και προωθεί στον σταθμό βάσης.

Η παραπάνω λειτουργία ονομάζεται μόνιμης κατάστασης (steady-state) και η ολοκλήρωση της, το τέλος ενός round. Μετά από ένα προκαθορισμένο χρονικό διάστημα ξεκινά ένα νέο round όπου έχουμε τον ορισμό νέων cluster head. Θα πρέπει να αναφερθεί ότι για την μείωση των παρεμβολών (radio interference) που δημιουργούνται κατά την διάρκεια της ασύρματης επικοινωνίας, το πρωτόκολλο LEACH χρησιμοποιεί την μέθοδο CDMA (Code Division Multiple Access). Σύμφωνα με αυτή την μέθοδο, κάθε συστάδα επικοινωνεί χρησιμοποιώντας διαφορετικούς κώδικες CDMA. Μέσω των κωδικών αυτών, είναι δυνατός ο διαχωρισμός των πληροφοριών που ανταλλάσσουν οι συστάδες. Η επιλογή του κώδικα που υιοθετείται από το κάθε cluster γίνεται από τον cluster head.



Σχήμα 2.2.: Παράδειγμα λειτουργίας του πρωτοκόλλου LEACH.

Στο Σχήμα 2.2 περιγράφεται ο γενικότερος τρόπος λειτουργίας του πρωτοκόλλου LEACH. Κάθε κόμβος (λευκό χρώμα) επιλέγει έναν από τους cluster head (γκρίζο χρώμα) για να προωθήσει τα δεδομένα που έχει συλλέξει. Στην συνέχεια, κάθε cluster head προωθεί τα δεδομένα της συστάδας του στον σταθμό βάσης.

### 2.3. Γνωστά Πρωτόκολλα Δρομολόγησης

Το πρωτόκολλο που περιγράφεται στο [33] ονομάζεται PEGASIS (Power-Efficient Gathering in Sensor Information Systems). Στόχος του είναι να δημιουργήσει μία αλυσίδα μεταξύ των κόμβων του δικτύου. Κάθε κόμβος της αλυσίδας επικοινωνεί με κάποιον

## 2. Ιεραρχικά Πρωτόκολλα

γειτονικό του μειώνοντας έτσι τις αποστάσεις μετάδοσης. Ο σχηματισμός της αλυσίδας ξεκινά από τον πιο απομακρυσμένο, σε σχέση με τον σταθμό βάσης, κόμβο. Τα δεδομένα που συλλέγονται από τους κόμβους του δικτύου, καταλήγουν στον σταθμό βάσης μόνο από έναν, που ονομάζεται leader. Ο κόμβος leader αλλάζει μετά από κάποιο χρονικό διάστημα  $t$ , ώστε να υπάρχει μία ομοιόμορφη κατανομή στην κατανάλωση της ενέργειας.

Στο [34] παρουσιάζεται ένα πρωτόκολλο με την ονομασία BCDCP (Base-Station Controlled Dynamic Clustering Protocol). Σύμφωνα με το πρωτόκολλο αυτό οι κόμβοι του δικτύου σχηματίζουν σχεδόν ισοδύναμες συστάδες για να ισορροπήσουν την κατανάλωση ενέργειας ανάμεσα στους κόμβους cluster head. Για την μετάδοση των δεδομένων του δικτύου στον σταθμό βάσης, χρησιμοποιείται μία cluster head προς cluster head στρατηγική. Η επιλογή των cluster head γίνεται από τον σταθμό βάσης ο οποίος αρχικά λαμβάνει πληροφορίες που αφορούν στην διαθέσιμη ενέργεια κάθε κόμβου. Βασισμένος σε αυτή την γνώση, υπολογίζει την μέση ενέργεια όλων των κόμβων. Κόμβοι με ενέργεια περισσότερη από την μέση, δημιουργούν ένα σύνολο  $S'$  από το οποίο επιλέγονται οι cluster head.

Το πρωτόκολλο TEEN (Threshold Sensitive Energy Efficient), που παρουσιάζεται στο [35], βασίζεται και αυτό στην ιεραρχική συσταδοποίηση. Η λειτουργία του ξεκινά όταν τα αισθητήρια όργανα των κόμβων του δικτύου ανιχνεύσουν κάποια δραστηριότητα πέρα από τα επιτρεπτά όρια. Οι συστάδες που βρίσκονται στο χαμηλότερο επίπεδο προωθούν τα δεδομένα τους σε αυτές που βρίσκονται στο υψηλότερο έως ότου καταλήξουν στον σταθμό βάσης. Στο [36], παρουσιάζεται το πρωτόκολλο SHORT (Shortest Hop Routing Tree protocol). Στόχος του είναι η ελαχιστοποίηση των καθυστερήσεων μετάδοσης (time sensitive εφαρμογές) και η αποτελεσματική διαχείριση της ενέργειας. Ο σταθμός βάσης είναι αυτός που υπολογίζει ένα κόμβο (cluster leader), που θα αναλάβει την ευθύνη της μετάδοσης των δεδομένων του δικτύου. Η επιλογή γίνεται με κριτήριο μία τιμή που προκύπτει από την υψηλότερη διαθέσιμη ενέργεια και την απόσταση της απευθείας μετάδοσης στον σταθμό βάσης.

Εκτός από τα παραπάνω πρωτόκολλα, υπάρχουν και άλλα που χρησιμοποιούν ειδικές διατάξεις προκειμένου να ελαχιστοποιήσουν την κατανάλωση ενέργειας των κόμβων του δικτύου. Ένα από αυτά παρουσιάζεται στο [37] και ονομάζεται M-GEAR (Gateway-Based Energy-Aware Multi-Hop Routing Protocol). Ξανά έχουμε την λογική του σχηματισμού συστάδων και την επιλογή ενός cluster head για καθεμιά από αυτές. Οι cluster head προωθούν τα δεδομένα των συστάδων, σε ένα κόμβο που ονομάζεται gateway. Ο κόμβος αυτός βρίσκεται σε κοντινή απόσταση από κάθε συστάδα (στο κέντρο του δικτύου) και έτσι μειώνονται οι αποστάσεις μετάδοσης. Η πληροφορία καταλήγει στον σταθμό βάσης από τον κόμβο gateway η ενέργεια του οποίου είναι μεν πεπερασμένη, αλλά μπορεί να ανανεωθεί μέσω της επαναφόρτισης. Ο σταθμός βάσης τοποθετείται εκτός δικτύου και η απόστασή του από αυτό μπορεί να είναι μεγάλη.

Η βιβλιογραφία δεν περιορίζεται μόνο στην ανεύρεση νέων πρωτοκόλλων για την εξοικονόμηση ενέργειας στα WSN. Υπάρχει επίσης αρκετή προσπάθεια για την ανάπτυξη και βελτίωση των πρωτοκόλλων που έχουν προταθεί [38], [39], [40]. Παραδείγματα τέτοιων πρωτοκόλλων αποτελεί και το πρωτόκολλο LEACH [41], [42]. Μία βελτίωση για την έκδοση που παρουσιάστηκε σε αυτή την εργασία μπορεί να βρεθεί στο [43]. Το πρωτόκολλο αυτό αποτελεί μια centralized έκδοση του LEACH (LEACH-C) και η

### 2.3. Γνωστά Πρωτόκολλα Δρομολόγησης

κύρια διαφορά του είναι η συμμετοχή του σταθμού βάσης στον σχηματισμό συστάδων. Ο σταθμός βάσης λαμβάνει πληροφορίες για την θέση και την ενέργεια των κόμβων, υπολογίζει τους κόμβους cluster head και σχηματίζει τις συστάδες. Οι υπόλοιπες λειτουργίες του πρωτοκόλλου είναι όμοιες με αυτές του LEACH.

Πανεπιστήμιο Πειραιώς

Πανεπιστήμιο Πειραιώς

## ΚΕΦΑΛΑΙΟ 3

### ΜΕΘΟΔΟΣ ΕΞΟΙΚΟΝΟΜΗΣΗΣ ΕΝΕΡΓΕΙΑΣ

Σε αυτό το κεφάλαιο θα παρουσιαστεί μία μέθοδος για την εξοικονόμησης ενέργειας στα ασύρματα δίκτυα αισθητήρων (ΜΕΕΑΔΑ). Η μέθοδος αυτή περιλαμβάνει ένα σύνολο αλγορίθμων που καλύπτουν όλες τις λειτουργίες ενός πρωτοκόλλου δρομολόγησης, που ανήκει στην κατηγορία των ιεραρχικών πρωτοκόλλων. Πιο συγκεκριμένα, το πρωτόκολλο αυτό περιλαμβάνει αλγορίθμους, οι οποίοι αφορούν στην διαδικασία της συσταδοποίησης (clustering) και της δρομολόγησης της πληροφορίας που συγκεντρώνεται από τους κόμβους ενός WSN. Η υλοποίησή του, έχει ως στόχο την αποτελεσματική διαχείριση της ενέργειας του δικτύου και την παροχή υψηλής ποιότητας υπηρεσιών – QoS.

#### 3.1. Εισαγωγή

Η αποτελεσματική διαχείριση ενός WSN μπορεί να επιτευχθεί μέσα από την σωστή διαχείριση των πόρων του. Κατά την υλοποίηση του δικτύου, θα πρέπει να λάβουμε σοβαρά υπόψη τις ιδιότητες των κόμβων (διαθέσιμη ενέργεια, εύρος ζώνης) που το αποτελούν. Επιπλέον, το μέγεθος της περιοχής εφαρμογής του WSN σε σχέση με τον αριθμό των κόμβων, είναι παράγοντες που επηρεάζουν την βιωσιμότητα του δικτύου. Μεγάλα σε γεωγραφική κάλυψη δίκτυα και μικρός αριθμός κόμβων μπορεί να οδηγήσει στην διεύρυνση των αποστάσεων μετάδοσης και κατά συνέπεια στην αύξηση της κατανάλωσης ενέργειας. Όπως αναφέραμε και στο Κεφάλαιο 2 (Σχέση (2.1)), η κατανάλωση ενέργειας δεν είναι γραμμική σε σχέση με την απόσταση.

Ωστόσο, η αύξηση της πυκνότητας του δικτύου είναι ένα μέτρο που μπορεί να οδηγήσει στην μείωση των αποστάσεων. Κάτι τέτοιο, όμως, οδηγεί στην αύξηση της ποσότητας της πληροφορίας που συγκεντρώνεται από το δίκτυο και την πρόσθεση επιπλέον κόστους κατανάλωσης ενέργειας ανά round. Επίσης, δεν πρέπει να ξεχνάμε ότι η χρήση περισσότερων αισθητήριων κόμβων έχει ως επακόλουθο και την αύξηση του κόστους υλοποίησης του δικτύου. Η ΜΕΕΑΔΑ είναι ένα πρωτόκολλο δρομολόγησης που έχει ως πρωτεύον στόχο την μείωση των αποστάσεων μεταξύ των κόμβων, ανεξάρτητα από την σχέση μεγέθους δικτύου και αριθμού κόμβων. Αυτό επιτυγχάνε-

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

ται υιοθετώντας μία στρατηγική που περιλαμβάνει μία μέθοδο clustering δύο επιπέδων καθώς και αλγορίθμους multi-hop συλλογής και δρομολόγησης των δεδομένων του δικτύου.

Η λειτουργία της ΜΕΕΑΔΑ ξεκινά με την διαδικασία της συσταδοποίησης (clustering), η οποία μοιράζεται σε δύο επιμέρους βήματα-επίπεδα. Στο πρώτο βήμα, έχουμε την “αναγνώριση” της τοποθεσίας που βρίσκονται οι κόμβοι στο δίκτυο καθώς και των μεταξύ τους αποστάσεων. Από την πληροφορία που συγκεντρώνεται μπορούμε να σχηματίσουμε τις πρώτες συστάδες. Οι συστάδες αυτές περιέχουν κόμβους που βρίσκονται σε πολύ κοντινές αποστάσεις. Με την ολοκλήρωση αυτής της διαδικασίας, έχουμε την εκκίνηση του δεύτερου βήματος το οποίο αποσκοπεί στην ένωση των συστάδων. Τα κριτήρια με τα οποία γίνεται αυτή η ένωση, αφορούν τον επιθυμητό αριθμό των τελικών συστάδων και την περιοχή στην οποία βρίσκονται οι πρώτες συστάδες.

Έχοντας χωρίσει το δίκτυο σε συστάδες, ακολουθεί η διαδικασία της συλλογής της πληροφορίας κάθε κόμβου. Σε κάθε cluster έχουμε την εκλογή τριών cluster head. Οι κόμβοι αυτοί βρίσκονται σε στρατηγικές θέσεις και είναι υπεύθυνοι να συλλέξουν όλη την πληροφορία της συστάδας που ανήκουν. Κάθε κόμβος που δεν είναι cluster head, επιλέγει να στείλει τα δεδομένα που συνέλεξε σε έναν από τους τρεις, με κριτήριο την μικρότερη απόσταση. Έτσι έχουμε την δημιουργία τριών υποσυστάδων (subcluster) σε κάθε μια από τις κύριες συστάδες. Οι υποσυστάδες έχουν την ιδιότητα να αλλάζουν κατά την λειτουργία του δικτύου (adaptive subclustering). Η πληροφορία των cluster head καταλήγει σε έναν από τους τρεις (cluster leader) και προωθείται στον σταθμό εργασίας, είτε με απ' ευθείας μετάδοση, είτε μέσω κάποιου cluster leader ή cluster head μιας άλλης συστάδας.

Οι τεχνικές data aggregation και data compression χρησιμοποιούνται σε κάθε cluster head και cluster leader, έτσι ώστε να μειώνεται το μέγεθος των δεδομένων που μεταδίδεται. Η πληροφορία καταλήγει στον σταθμό βάσης (ολοκληρώνοντας ένα round) μόνο από έναν cluster leader<sup>1</sup>. Συνοψίζοντας, σε αντίθεση με άλλα ιεραρχικά πρωτόκολλα, στην ΜΕΕΑΔΑ έχουμε συστάδες και υποσυστάδες. Οι συστάδες δημιουργούνται μόνο μία φορά (static clustering) στην αρχή του δικτύου και δεν αλλάζουν εκτός και αν έχουμε την προσθήκη νέων κόμβων. Οι υποσυστάδες είναι προσαρμοστικές και περιέχουν κόμβους οι οποίοι αναλαμβάνουν την συλλογή της πληροφορίας. Τέλος, cluster head και cluster leader κάθε συστάδας, συνεργάζονται μεταξύ τους για να μειώσουν τις αποστάσεις μετάδοσης και κατά συνέπεια την κατανάλωση ενέργειας στο δίκτυο.

#### 3.2. Δημιουργία Συστάδων

Η διαδικασία της συσταδοποίησης γίνεται σε δύο επίπεδα και εκτελείται μόνο μία φορά στην αρχή της δημιουργίας του δικτύου. Το δίκτυο που αποτελεί το WSN περιλαμβάνει ένα αριθμό  $C'$  κόμβων οι οποίοι τοποθετούνται τυχαία σε μία περιοχή. Στόχος είναι η διαίρεση του δικτύου σε υποδίκτυα μικρότερου μεγέθους και γεωγραφικής κάλυψης. Για να επιτευχθεί κάτι τέτοιο, λαμβάνονται υπόψιν κάποιες γενικότερες παράμετροι μέσω των

<sup>1</sup> Σε εξαιρετικά ειδικές περιπτώσεις, περισσότεροι από ένας cluster leader μπορεί να μεταδώσουν στον σταθμό βάσης.

οποίων καθορίζεται η λειτουργία της ΜΕΕΑΔΑ. Έτσι λοιπόν, στο πρωτόκολλο αυτό θεωρείται ότι:

- Ο προορισμός ή σταθμός βάσης (base station) βρίσκεται σε σταθερή θέση και μακριά από το δίκτυο. Επίσης, η ενέργεια που διαθέτει είναι απεριόριστη.
- Η ελάχιστη απόσταση μεταξύ δύο κόμβων θεωρείται το ένα μέτρο ( $D_{ab} = 1m$ ).
- Οι αισθητήριοι κόμβοι που απαρτίζουν το δίκτυο είναι ομοιογενείς (ίδια υπολογιστική ικανότητα, εύρος ζώνης) και η ενέργειά τους πεπερασμένη.

Το εύρος της περιοχής κάλυψης του δικτύου και η πυκνότητά του μπορεί να είναι διαφορετικά από σημείο σε σημείο. Για τον λόγο αυτό θεωρείται προτιμότερο οι κόμβοι που βρίσκονται πολύ κοντά μεταξύ τους να ανήκουν πάντοτε στην ίδια συστάδα. Επίσης, ο αριθμός των κόμβων της κάθε συστάδας θα πρέπει να είναι περίπου ίσος. Αυτό θα βοηθήσει έτσι ώστε να επιτευχθεί μία ισορροπία σε ό,τι αφορά την κατανάλωση ενέργειας ανά round. Στο πρώτο βήμα της συσταδοποίησης έχουμε το κριτήριο της ελάχιστης απόστασης όπου ορίζεται η ομάδα που θα ανήκει ο κάθε κόμβος. Για τον υπολογισμό των αποστάσεων μεταξύ δύο κόμβων  $A(a_x, a_y)$  και  $B(b_x, b_y)$  χρησιμοποιείται η ευκλείδεια απόσταση η οποία περιγράφεται από την σχέση:

$$Euc = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2} \quad (3.1)$$

Η παραπάνω σχέση (3.1) χρησιμοποιείται οπουδήποτε απαιτείται ο υπολογισμός κάποιας απόστασης στο δίκτυο και το αποτέλεσμά της θεωρείται ότι είναι σε μέτρα. Στο δεύτερο βήμα, έχουμε την συγχώνευση των συστάδων του πρώτου βήματος. Ξεκινώντας από συγκεκριμένα “σημεία εκκίνησης” στον χώρο, θα αυξάνουμε σιγά σιγά την περιοχή κάλυψης, χρησιμοποιώντας διαφορετικά πρότυπα διάσχισης (traversal patterns). Η περιοχή θα αυξάνεται έως ότου ομαδοποιήσουμε ένα ικανοποιητικό αριθμό συστάδων. Ο αριθμός αυτός καθορίζεται εκ’ των προτέρων από τον αριθμό των κόμβων του δικτύου και από τον τελικό αριθμό των συστάδων που θέλουμε να δημιουργηθούν.

### 3.2.1. Έλεγχος Αποστάσεων Αισθητήρων

Ένα ενδιαφέρον ερώτημα κατά την διαδικασία της συσταδοποίησης είναι το ποι οι κόμβοι θα αποτελούν την κάθε συστάδα. Στην ΜΕΕΑΔΑ χρησιμοποιείται μία στρατηγική όπου δύο κόμβοι ανήκουν στην ίδια συστάδα ανάλογα με την απόσταση που τους χωρίζει. Υπεύθυνος για τον σχηματισμό των συστάδων στο πρώτο επίπεδο είναι ο σταθμός βάσης, η ενέργεια του οποίου θεωρείται απεριόριστη. Για την αποτελεσματικότητα της μεθόδου, είναι απαραίτητο να υπάρχει η γνώση της τοποθεσίας του κάθε κόμβου στο δίκτυο. Αυτή η γνώση μπορεί να αποκτηθεί από τον σταθμό βάσης, μέσα από τους ίδιους τους κόμβους. Οι κόμβοι μπορούν να μεταδώσουν σε συγκεκριμένες χρονικές στιγμές την τοποθεσία και την κατάσταση τους, μέσω ενός προγράμματος TDMA.

Εφόσον ολοκληρωθεί η διαδικασία και ο σταθμός βάσης γνωρίζει την τοποθεσία κάθε κόμβου, ως “κόμβος εκκίνησης” ξεκινά με την επιλογή του κοντινότερου κόμβου. Ο

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

κόμβος αυτός, επιλέγεται ως ο νέος κόμβος εκκίνησης και δημιουργεί την πρώτη συστάδα. Έπειτα ακολουθείται μία παρόμοια στρατηγική προκειμένου να αποφασιστεί αν η απόσταση του επόμενου κοντινότερου κόμβου είναι ικανή να δημιουργήσει μία νέα συστάδα ή όχι. Πιο συγκεκριμένα, για τον σχηματισμό των συστάδων στο πρώτο βήμα χρησιμοποιείται ο αλγόριθμος ελέγχου αποστάσεων (Αλγόριθμος 1).

---

#### Αλγόριθμος 1: ΕΛΕΓΧΟΣ ΑΠΟΣΤΑΣΕΩΝ ( $D$ )

---

**Είσοδος:** Ένα πεπερασμένο σύνολο  $C' = \{nid_1, nid_2, \dots, nid_n\}$  όπου  $nid_i$  ο κωδικός κόμβου (αρχικά όλοι 0 και  $n = 0$ ) και  $nid_{x,y}$  οι συντεταγμένες  $x$  και  $y$  του  $nid_i$  στον χώρο.

**Έξοδος:** Κάθε  $nid_i \in C'$  με ένα κωδικό συστάδας  $m$ .

1.  $z \leftarrow C'$
  2.  $s_{x,y} \leftarrow BS_{x,y}$
  3.  $d_1, d_2 \leftarrow r$
  4.  $m \leftarrow k$
  5.  $threshold \leftarrow maxdist$
  6.  $x \leftarrow maxX$
  7.  $y \leftarrow maxY$
  8. **ενόσω** ( $n \neq z$ )
    9. **για**  $i \leftarrow 1$  **έως**  $x$
    10. **για**  $j \leftarrow 1$  **έως**  $y$
    11. **Αν** ( $nid = 0$ )
    12.  $distA \leftarrow$  ΕΥΚΛΕΙΔΕΙΑ ΑΠΟΣΤΑΣΗ ( $nid_{x,y}, s_{x,y}$ )
    13. **Αν** ( $distA < d_1$ )
    14.  $t_{x,y} \leftarrow nid_{x,y}$
    15.  $d_1 \leftarrow distA$
    16.  $distB \leftarrow distA$
    17. **Αν** ( $distB > d_2$  **ή**  $distB > threshold$ )
    18.  $t \leftarrow m + 1$
    19. **Άλλως**
    20.  $t \leftarrow m$
    21.  $s_{x,y} \leftarrow t_{x,y}$
    22.  $d_2 \leftarrow distB$
    23.  $d_1 \leftarrow r$
    24.  $n \leftarrow n + 1$
  25. **επιστροφή** συστάδων
- 

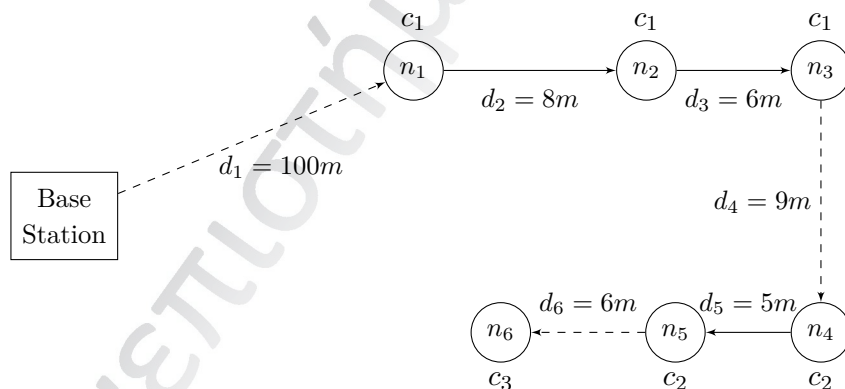
Ο αλγόριθμος 1 έχει ως είσοδο ένα σύνολο κόμβων  $c \in C'$  οι οποίοι τοποθετούνται τυχαία σε μία περιοχή. Κάθε κόμβος διαθέτει ένα κωδικό ο οποίος χρησιμοποιείται για να είναι δυνατός ο διαχωρισμός μεταξύ των συστάδων (αρχικά όλοι διαθέτουν τον κωδικό 0). Η λειτουργία ξεκινά (γραμμές 1-4) με τον ορισμό της μεταβλητής  $z$  που συμβολίζει το πλήθος των κόμβων στο δίκτυο, τον κόμβο εκκίνησης  $s_{x,y}$ , ο οποίος αρχικά είναι ο σταθμός βάσης  $BS_{x,y}$ , και των αποστάσεων  $d_1$  και  $d_2$ . Η απόσταση  $r$  είναι μία τυχαία απόσταση πολύ μεγαλύτερη από κάθε δυνατή απόσταση μετάδοσης



οποιοδήποτε κόμβου στον σταθμό βάσης. Στην συνέχεια ορίζουμε τον κωδικό  $m$  που θα αποκτήσει κάθε κόμβος  $c$  του δικτύου. Ο κωδικός αυτός  $k$  είναι ένας αριθμός ο οποίος θα πρέπει να είναι τουλάχιστον ίσος με  $2 * C' + 2$ . Αυτό γίνεται για να διατηρηθεί η ομαλή λειτουργία, όταν είναι επιθυμητό να μην υπάρχουν περιορισμοί σε ό,τι αφορά την θέση του σταθμού βάσης.

Στην συνέχεια (γραμμές 5-7) έχουμε τον ορισμό των μεγίστων  $maxX$  και  $maxY$  του δικτύου προκειμένου να είναι γνωστά τα όρια της περιοχής όπου περιέχονται οι κόμβοι. Επίσης, ορίζουμε και ένα όριο  $threshold$ , όπου δύο κόμβοι δεν μπορούν να ανήκουν στην ίδια συστάδα αν η απόστασή τους είναι μεγαλύτερη από  $maxdist$ . Έπειτα, (γραμμές 8-24) έχουμε το κύριο μέρος του αλγορίθμου το οποίο θα εκτελεστεί για  $n = z = C'$  φορές (αρχικά  $n = 0$ ). Η διαδικασία αυτή περιλαμβάνει (γραμμές 9-16) μία αναζήτηση για τον κόμβο  $t_{x,y}$  με την μικρότερη απόσταση  $distA$  από τον κόμβο  $s_{x,y}$ . Αρχικά, (γραμμή 11) επιλέγεται ένας κόμβος  $n_{id} = c$ , ο κωδικός του οποίου θα πρέπει να είναι ίσος με το 0 (χωρίς συστάδα). Η απόσταση του κοντινότερου κόμβου (γραμμές 12-16)  $t_{x,y}$  ορίζεται ως  $distB$ .

Αν η απόσταση  $distB$  του κόμβου  $t_{x,y}$  (γραμμές 17-20) είναι μεγαλύτερη από την  $d_2$  ή μεγαλύτερη από το όριο που έχουμε θέσει, τότε ο κωδικός του κόμβου  $t$  θα είναι ίσος με τον τρέχον  $m + 1$ . Σε αντίθετη περίπτωση, ο κωδικός του  $t$  θα είναι ίσος με  $m$ . Η ολοκλήρωση του βήματος (γραμμές 21-24) γίνεται με τον ορισμό του κόμβου  $t_{x,y}$  ως νέο κόμβο εκκίνησης  $s_{x,y}$ . Η απόσταση  $d_2$  γίνεται ίση με την  $distB$  και αρχικοποιείται η  $d_1$ . Μετά από αυτό, έχουμε την ολοκλήρωση ενός βήματος, πράγμα που σημαίνει ότι ένας κόμβος έχει αποκτήσει ένα κωδικό συστάδας. Αφού αυξήσουμε το  $n$  κατά ένα, επαναλαμβάνουμε την ίδια διαδικασία για όλους τους κόμβους. Ο αλγόριθμος ολοκληρώνεται (γραμμή 25) επιστρέφοντας τους κόμβους του δικτύου, με κάθε κόμβο να διαθέτει ένα κωδικό συστάδας.



Σχήμα 3.1.: Παράδειγμα εκτέλεσης του αλγορίθμου ελέγχου αποστάσεων.

Στο παράδειγμα του Σχήματος 3.1, μπορούμε να παρατηρήσουμε την εκτέλεση του αλγορίθμου ελέγχου αποστάσεων για έξι αισθητήριους κόμβους. Από ένα σύνολο  $C' = \{n_1, n_2, n_3, n_4, n_5, n_6\}$  κόμβων δημιουργείται ένα σύνολο  $S' = \{c_1, c_2, c_3\}$  συστάδων. Οι διακεκομμένες ακμές συμβολίζουν την έναρξη μιας νέας συστάδας και οι συμπαγείς

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

τους κόμβους που ανήκουν στην ίδια. Ξεκινώντας από τον σταθμό βάσης, η απόστασή του είναι πάντοτε μεγαλύτερη από το όριο που θέτουμε (π.χ. 10 μέτρα) γιατί θεωρείται ότι βρίσκεται μακριά από το δίκτυο. Ο κόμβος που βρίσκεται πιο κοντά στον σταθμό βάσης  $n_1$  αποκτά τον κωδικό της πρώτης συστάδας  $c_1$ . Ο επόμενος κοντινότερος, αυτή την φορά για τον κόμβο  $n_1$ , βρίσκεται σε απόσταση  $d_2$ , η οποία είναι μικρότερη από την  $d_1$  και μικρότερη από το όριο. Για τον λόγο αυτό, οι κόμβοι  $n_1$  και  $n_2$  θα ανήκουν στην ίδια συστάδα.

Το ίδιο ισχύει και για τον κοντινότερο στον  $n_2$ , κόμβο  $n_3$  όπου  $d_3 < d_2$  και  $d_3 < threshold$ . Ωστόσο, ο κόμβος  $n_4$  κοντινότερος του  $n_3$  δημιουργεί μία νέα συστάδα διότι  $d_4 > d_3$  ή  $d_4 > threshold$ . Με τον ίδιο τρόπο συνεχίζεται η κατηγοριοποίηση και των υπολοίπων κόμβων. Οι έξι κόμβοι κατηγοριοποιήθηκαν σε τρεις “πρώτες συστάδες”, που στο δεύτερο επίπεδο μπορούν να δημιουργήσουν 1 ως  $n$  κύριες συστάδες (όπου  $n$  το πλήθος των πρώτων συστάδων). Για λόγους χώρου, το παράδειγμα του Σχήματος 3.1 δεν είναι αντιπροσωπευτικό ενός πραγματικού δικτύου, καθώς σε ένα WSN μπορεί να έχουμε εκατοντάδες κόμβους.

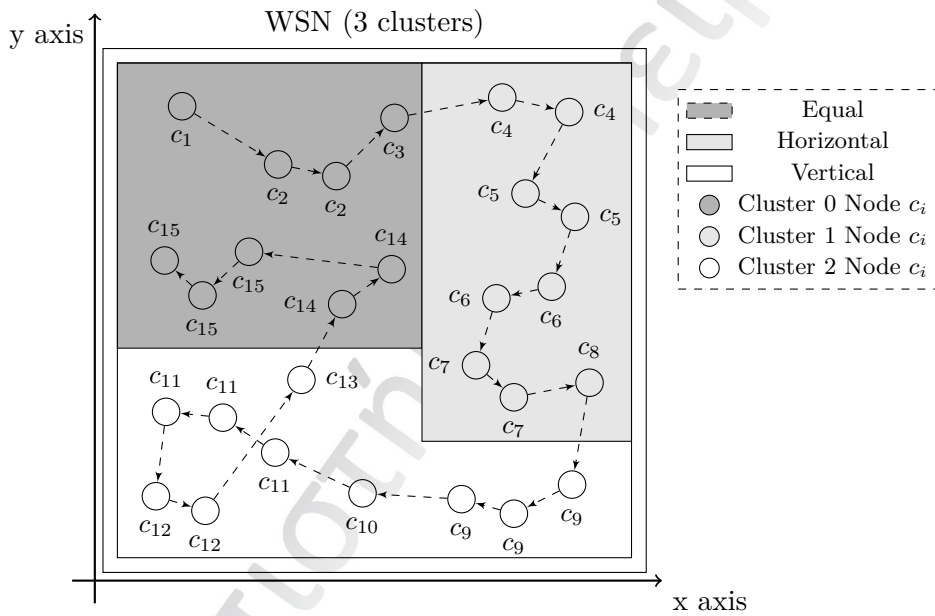
#### 3.2.2. Σχηματισμός Τελικών Συστάδων

Ο αριθμός των συστάδων που δημιουργούνται από το πρώτο βήμα του αλγορίθμου της συσταδοποίησης, μπορεί να είναι αρκετά μεγάλος σε σχέση με τον αριθμό των κόμβων του δικτύου. Στο δεύτερο βήμα θα προχωρήσουμε σε περαιτέρω συσταδοποίηση, σύμφωνα με ένα ποσοστό που καθορίζεται από το αρχικό πλήθος των κόμβων. Οι νέες συστάδες θα μπορούν λειτουργήσουν αυτόνομα ως ξεχωριστά υποδίκτυα. Οι τελικές συστάδες ορίζονται ως  $N = 5\%$ , όπως και στο πρωτόκολλο LEACH, επειδή το ποσοστό αυτό προσφέρει πολύ καλές επιδόσεις σε δίκτυα με διαφορετικό μέγεθος και πλήθος κόμβων.

Κάθε κόμβος θα έχει την εποπτεία μόνο μίας περιορισμένης περιοχής του δικτύου. Αυτό μπορεί να αποδειχθεί ιδιαίτερα σημαντικό, στην περίπτωση όπου οι κόμβοι κληθούν να ορίσουν μόνοι τους τον τρόπο που θα γίνει η συλλογή της πληροφορίας στην συστάδα. Για την εκπλήρωση αυτού του στόχου και την δημιουργία υποδικτύων μέσα στο δίκτυο, είναι αναγκαίο να εφευρεθεί ένας τρόπος ο οποίος θα λαμβάνει υπόψιν του, τον αριθμό των συστάδων του πρώτου επιπέδου και την ευρύτερη περιοχή εφαρμογής του WSN. Επίσης, ο διαχωρισμός θα πρέπει να γίνει με τέτοιο τρόπο ώστε να τηρείται μια ικανοποιητική αναλογία μεταξύ του αριθμού κόμβων και του μεγέθους της περιοχής.

Αυτό επιτυγχάνεται μέσα από πρότυπα διάσχισης. Μέσω των προτύπων αυτών γίνεται η προσπάθεια ώστε η κάθε τελική συστάδα-υποδίκτυο να καταλαμβάνει περίπου ίσο χώρο στο δίκτυο, αναλογικά με τις άλλες. Επίσης, οποιαδήποτε από τις συστάδες θα πρέπει να μην βρίσκεται σε μεγάλη απόσταση από τον σταθμό βάσης, ο οποίος μπορεί να βρίσκεται σε οποιαδήποτε θέση. Αυτό το μετράμε με τον αριθμό των hop που θα πρέπει να γίνουν από την συστάδα προκειμένου η πληροφορία να καταλήξει στον προορισμό. Για παράδειγμα αν μεταξύ μιας συστάδας  $A = \{a_1, a_2, \dots, a_n\}$  και του σταθμού βάσης παρεμβάλλεται μόνο μία συστάδα  $B = \{b_1, b_2, \dots, b_n\}$  τότε η απόσταση της  $A$  από τον σταθμό βάσης είναι ίση με ένα hop. Η επίτευξη των προαναφερθέντων γίνεται μέσα από τα ακόλουθα πρότυπα διάσχισης:

1. *Equal Traversal*: Το πρότυπο αυτό εκτελείται μόνο μία φορά στην αρχή και μετά έχουμε την εναλλαγή μεταξύ των horizontal και vertical. Από ένα σημείο εκκίνησης  $b_{x,y}$  και τερματισμού  $e_{x,y}$ , αναζητείται ο επιθυμητός αριθμός, *number* των συστάδων. Αν ο αριθμός αυτός δεν βρεθεί τότε αυξάνουμε<sup>2</sup> τα σημεία  $x, y$  του  $e_{x,y}$  κατά μία σταθερή ποσότητα  $r$  ( $r = 1$ ).
2. *Horizontal Traversal*: Από ένα σημείο σημείο εκκίνησης  $b_{x,y}$  και τερματισμού  $e_{x,y}$  αν ο αριθμός, *number* των συστάδων δεν είναι επαρκής, τότε αυξάνουμε το σημείο  $y$  του  $e_{x,y}$  κατά μία σταθερή ποσότητα  $r$  ενώ το σημείο  $x$  αυξάνεται κατά  $\lceil \text{max}X/10 \rceil$ , όπου  $\text{max}X$  η μέγιστη τιμή  $x$  της περιοχής του WSN.
3. *Vertical Traversal*: Σε αντίθεση με την οριζόντια διάσχιση, αν ο αριθμός *number*, μεταξύ  $b_{x,y}$  και  $e_{x,y}$  δε είναι επαρκής τότε το σημείο  $y$  του  $e_{x,y}$  αυξάνεται κατά  $\lceil \text{max}Y/10 \rceil$ , όπου  $\text{max}Y$  η μέγιστη τιμή  $y$  της περιοχής εφαρμογής του WSN. Το σημείο  $x$  του  $e_{x,y}$  αυξάνεται κατά μία σταθερή ποσότητα  $r$ .



Σχήμα 3.2.: Παράδειγμα εκτέλεσης των προτύπων διάσχισης.

Στο Σχήμα 3.2 έχουμε ένα δίκτυο WSN με τρεις τελικές συστάδες που δημιουργούνται από δεκαπέντε πρώτες  $c_1, c_2, \dots, c_{15}$  (βλ. Αλγόριθμος 1 και Σχήμα 3.1). Για κάθε τελική συτάδα αντιστοιχούν πέντε πρώτες  $c_i$ . Με σκούρο γκρι χρώμα παρουσιάζεται η περιοχή που καταλαμβάνει η πρώτη συστάδα. Η συστάδα αυτή δημιουργήθηκε με την χρήση της ίσης διάσχισης και αρχικό σημείο  $b_{x,y} = [1][\text{max}Y]$  και τελικό  $e_{x,y} = [1][\text{max}Y]$ .

<sup>2</sup>Ο όρος “αυξάνουμε” χρησιμοποιείται με την έννοια της επιλογής ενός σημείου που αυξάνει την περιοχή αναζήτησης. Μπορεί να υποδηλώνει την πράξη της αφαίρεσης και της πρόσθεσης ταυτόχρονα.

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

Η περιοχή αναζήτησης αυξάνεται έως ότου βρεθεί το ζητούμενο πλήθος συστάδων. Αφού ολοκληρωθεί, επιστρέφει μία μεταβλητή *trigger* που συμβολίζει ότι η μετέπειτα αναζήτηση θα ξεκινήσει με την χρήση της οριζόντιας διάσχισης. Με ανοιχτόχρωμο γκρι χρώμα εμφανίζεται η περιοχή που επιλέχθηκε από την οριζόντια διάσχιση. Το σημείο έναρξής της είναι το  $b_{x,y} = [j][maxY]$ , όπου  $j$  το σημείο  $x$  τερματισμού της ίσης διάσχισης.

Επειδή στην οριζόντια διάσχιση η περιοχή αναζήτησης αυξάνεται ταχύτερα στον άξονα των  $x$ , αυτό μπορεί να γίνει σύντομα ίσο με  $maxX$ . Όταν συμβαίνει αυτό και δεν έχει βρεθεί ο επιθυμητός αριθμός συστάδων, τότε η περιοχή αυξάνει μόνο ως προς  $y$ . Η οριζόντια διάσχιση σταματά την λειτουργία της επιστρέφοντας μία μεταβλητή *trigger*, που ορίζει την κάθετη διάσχιση ως την επόμενη μέθοδο αναζήτησης. Η κάθετη ξεκινά για σημείο έναρξης και τερματισμού  $b_{x,y} = [1][maxY]$  και  $e_{x,y} = [1][maxY]$  αντίστοιχα. Η αναζήτηση του Σχήματος 3.2 καταλαμβάνει όλη την περιοχή του δικτύου προκειμένου να βρεθεί (λευκό χρώμα) ο απαραίτητος αριθμός συστάδων.

---

#### Αλγόριθμος 2: ΣΧΗΜΑΤΙΣΜΟΣ ΤΕΛΙΚΩΝ ΣΥΣΤΑΔΩΝ ( $E$ )

---

**Είσοδος:** Ένα πεπερασμένο σύνολο  $S' = \{c_1, c_2, \dots, c_n\}$  όπου  $c_i$  ο κωδικός συστάδας,  $b_{x,y}$  το σημείο έναρξης και  $e_{x,y}$  (αρχικά  $x = 1, y = maxY$ ) το σημείο τερματισμού.

**Έξοδος:** Κάθε  $c_i \in S'$  με ένα κωδικό τελικής συστάδας.

1.  $k \leftarrow S'$
  2.  $l \leftarrow 0$
  3.  $d \leftarrow \lceil \hat{N} \rceil$
  4.  $trigger \leftarrow 0$
  5. **ενόσω** ( $d \neq 0$ )
    6.  $number \leftarrow \lceil k/d \rceil$
    7. **για**  $i \leftarrow b_x$  **έως**  $e_x$
    8. **για**  $j \leftarrow b_y$  **έως**  $e_y$
    9.  $F \leftarrow \text{ΜΕΤΡΗΣΗ ΣΥΣΤΑΔΩΝ}(b_{x,y}, e_{x,y})$
    10. **Αν** ( $F < number$ )
    11. **Αν** ( $trigger = 0$ )
    12. EQUAL TRAVERSAL ( $b_{x,y}, e_{x,y}$ )
    13. **Αν** ( $trigger = 1$ )
    14. HORIZONTAL TRAVERSAL ( $b_{x,y}, e_{x,y}$ )
    15. **Αν** ( $trigger = 2$ )
    16. VERTICAL TRAVERSAL ( $b_{x,y}, e_{x,y}$ )
  17. **Άλλως**
  18. ΚΛΕΙΔΙ  $l$  ΓΙΑ ΚΑΘΕ ΚΟΜΒΟ ( $b_{x,y}, e_{x,y}$ ) ΜΕ  $c > l$
  19.  $d \leftarrow d - 1$
  20.  $k \leftarrow k - F$
  21.  $l \leftarrow l + 1$
  22. **επιστροφή** συστάδων
- 

Η γενικότερη λειτουργία του δευτέρου επιπέδου της συσταδοποίησης, η οποία χρη-

σιμοποιείται για τον σχηματισμό των τελικών συστάδων του δικτύου περιγράφεται από τον αλγόριθμο 2. Ο αλγόριθμος αυτός δέχεται ως είσοδο το σύνολο των συστάδων που δημιουργήθηκαν από τον αλγόριθμο 1 και επιστρέφει στην έξοδο κάθε κόμβο με ένα κωδικό τελικής συστάδας. Η διαδικασία ξεκινά (γραμμές 1-4) με τον ορισμό του αριθμού των συστάδων του πρώτου βήματος, ένα κλειδί τελικής συστάδας, μία μεταβλητή  $d$  που συμβολίζει το 5% των αρχικών κόμβων  $C'$  του δικτύου και έναν *trigger*. Η βασική λειτουργία (γραμμές 5-21) περιλαμβάνει τον υπολογισμό του απαραίτητου πλήθους συστάδων, *number* (γραμμή 6), προς συγχώνευση. Έπειτα (γραμμές 7-8), από ένα σημείο έναρξης και ένα τερματισμού  $b_{x,y}$  και  $e_{x,y}$  μετράμε τον αριθμό των συστάδων που περιέχονται σε αυτή την περιοχή. Αν ο αριθμός τους  $F$  είναι μικρότερος από αυτόν που θέλουμε (γραμμή 9) τότε εκτελείται ένα από τα πρότυπα διάσχισης.

Για τον σχηματισμό της πρώτης συστάδας χρησιμοποιείται πάντοτε η ίση διάσχιση (γραμμές 11-12). Μετά την ολοκλήρωση της πρώτης συστάδας, θα έχουμε την εναλλαγή (γραμμές 13-16) της οριζόντιας και της κάθετης διάσχισης, κάθε φορά που θέλουμε να δημιουργηθεί μία νέα συστάδα. Η εκτέλεση κάθε διάσχισης, περιλαμβάνει τον ορισμό νέων σημείων έναρξης και τερματισμού,  $b_{x,y}$  και  $e_{x,y}$  και νέου *trigger*. Όταν ολοκληρωθεί η αναζήτηση (γραμμή 18), ορίζουμε τον κωδικό κάθε συστάδας που περιλαμβάνεται μεταξύ του  $b_{x,y}$  και  $e_{x,y}$  ίσο με  $l$ . Ο ορισμός αφορά κόμβους με κωδικό μεγαλύτερο από  $l$ , ώστε να μην επηρεάζονται ήδη κατηγοριοποιημένες συστάδες. Τέλος (γραμμές 19-21), μειώνουμε την μεταβλητή  $d$ , ορίζουμε τον νέο αριθμό συστάδων  $k$  που δεν έχουν αποκτήσει κωδικό και αυξάνουμε το κλειδί  $l$  κατά 1. Τερματίζοντας (γραμμή 22), ο αλγόριθμος επιστρέφει τις συστάδες που δημιούργησε.

Η διαδικασία του σχηματισμού των τελικών συστάδων είναι δυναμική και εκτελείται ανάλογα με την κατανομή των συστάδων του πρώτου βήματος στον χώρο. Αυτό σημαίνει ότι η περιοχή που καλύπτει μία τελική συστάδα (βλ. Σχήμα 3.2) μπορεί να διαφέρει σε μέγεθος. Επιπλέον, μπορεί να υπάρχει κάποιος κόμβος εκτός της περιοχής, σε πολύ κοντινή απόσταση, από αυτή που ορίζει ένα πρότυπο διάσχισης. Κάτι τέτοιο μπορεί να συμβεί, διότι κατά την αναζήτηση για ένα αριθμό συστάδων πρώτου βήματος, θεωρείται αρκετό να υπάρχει μόνο ένας κόμβος της συστάδας εντός εμβέλειας. Οι υπόλοιποι που έχουν τον ίδιο κωδικό, αλλάζουν ακόμα και αν βρίσκονται οριακά εκτός περιοχής κάλυψης.

### 3.3. Διαδικασία Δρομολόγησης

Έχοντας χωρίσει το δίκτυο σε συστάδες, η αμέσως επόμενη επιλογή είναι η συγκέντρωση της πληροφορίας κάθε κόμβου. Για τον σκοπό αυτό, θεωρείται ότι κάθε κόμβος έχει συλλέξει δεδομένα σε κάθε round. Η διαδικασία της συσταδοποίησης δημιουργήσε  $n$  υποδίκτυα τα οποία μπορούν να λειτουργήσουν ανεξάρτητα μεταξύ τους. Κάθε κόμβος μπορεί να περιορίσει την εμβέλεια μετάδοσής του στην συγκεκριμένη περιοχή, μειώνοντας έτσι την κατανάλωση ενέργειας. Στην αρχή, κάθε συστάδα θα πρέπει να εκλέγει τον cluster leader και τους cluster head με τέτοιο τρόπο, ώστε να ελαχιστοποιούνται οι αποστάσεις μετάδοσης. Στο δεύτερο και τελευταίο βήμα, για να μειώσουμε τις αποστάσεις μετάδοσης των απομακρυσμένων συστάδων προς τον σταθμό βάσης,

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

ακολουθείται μία στρατηγική multi-hop δρομολόγησης.

Για την αποτελεσματική συλλογή της πληροφορίας σε κάθε συστάδα, η ΜΕΕΑΔΑ χρησιμοποιεί αλγόριθμους επιλογής ειδικών κόμβων. Οι κόμβοι αυτοί λειτουργούν σε συνεργασία μεταξύ τους και καθορίζουν την απόσταση που μεταδίδει κάθε κόμβος. Σε κάθε κύκλο δικτύου (round) έχουμε την εκλογή διαφορετικών cluster leader και cluster head, με συνέπεια να αλλάζουν οι προτιμήσεις μετάδοσης των κόμβων. Το γεγονός αυτό οδηγεί σε ένα είδος προσαρμοστικής συσταδοποίησης σε κάθε συστάδα. Κάτι τέτοιο είναι ιδιαίτερα σημαντικό, καθώς μπορεί να έχουμε μία πιο ομοιόμορφη κατανομή στην κατανάλωση ενέργειας των κόμβων. Επίσης, σημαντικό είναι και το γεγονός ότι οι cluster leader από ένα σημείο και μετά μπορούν να εκλέγονται μέσα από προσαρμοστικά κριτήρια, έτσι ώστε να επιτυγχάνουμε ακόμα καλύτερη απόδοση.

Μετά το βήμα της συλλογής, έχουμε αυτό της δρομολόγησης της πληροφορίας. Κάθε συστάδα μεταδίδει τα δεδομένα της μέσω του κόμβου cluster leader. Ο προορισμός, εκτός από τον σταθμό βάσης, μπορεί να είναι κάποιος άλλος cluster leader ή (μετά από κάποιο χρονικό διάστημα) cluster head μιας άλλης συστάδας. Η απόφαση γίνεται ανάλογα με την θέση σταθμού βάσης στο δίκτυο και τον αριθμό των hop (συστάδων που παρεμβάλλονται). Επιπρόσθετα κριτήρια αποτελούν οι αποστάσεις μεταξύ των cluster leader και cluster head, όπως επίσης και οι αποστάσεις αυτών από τον σταθμό βάσης. Ο αλγόριθμος μέσα από τον οποίο γίνεται η δρομολόγηση της πληροφορίας ονομάζεται αλγόριθμος μη αρνητικής απόστασης (*non-negative distance routing*).

#### 3.3.1. Επιλογή Cluster Head

Η επιλογή των cluster head και του cluster leader μπορεί να γίνει με δύο τρόπους. Ο πρώτος και πιο εύκολος είναι μέσω του σταθμού βάσης. Αυτός θα τους επιλέξει βασιζόμενος σε πληροφορίες όπως η θέση και η διαθέσιμη ενέργεια κάθε κόμβου. Εφόσον ολοκληρωθεί η διαδικασία, κάθε κόμβος ενημερώνεται για τον ρόλο που θα αναλάβει. Ο δεύτερος τρόπος είναι μέσα από τους ίδιους τους κόμβους. Μετά την διαδικασία της συσταδοποίησης κάθε κόμβος έχει λάβει ένα πίνακα με την θέση των κόμβων που σχηματίζουν την ίδια συστάδα. Σε κάθε round, η επιπλέον πληροφορία που θα πρέπει να μεταδίδει ένας κόμβος προς τους υπόλοιπους της συστάδας είναι η κατάσταση της διαθέσιμης ενέργειάς του. Αφού ολοκληρωθεί η εκλογή των cluster head, ο κόμβος cluster leader είναι ο μόνος που θα πρέπει να ενημερώσει τους cluster leader των άλλων συστάδων για την θέση του.

Κάθε κόμβος ή ο σταθμός βάσης εκλέγει με τα ίδια κριτήρια τρεις διαφορετικούς κόμβους. Οι κόμβοι κατηγοριοποιούνται σύμφωνα με την σειρά επιλογής τους και είναι:

1. *Primary*: Ως πρωτεύων κόμβος, εκλέγεται αυτός με την περισσότερη ενέργεια στην συστάδα. Ο κόμβος αυτός αναλαμβάνει εξ' ορισμού, την θέση του cluster leader ενόσω όλοι οι κόμβοι διαθέτουν αρκετή ενέργεια.
2. *Secondary*: Ο κόμβος αυτός επιλέγεται με κριτήριο την απόσταση από τον primary καθώς και την διαθέσιμη ενέργεια. Μπορεί να αναλάβει την θέση του cluster leader όταν η απόστασή του από τον σταθμό βάσης, είναι μικρότερη από αυτή του primary.

3. *Territorial*: Αυτός ο cluster head είναι προαιρετικός και παρεμβάλλεται μεταξύ του primary και του secondary. Ο ρόλος του είναι υποστηρικτικός και δεν μπορεί να αναλάβει την θέση του cluster leader.

Στην αρχή έχουμε την εκλογή του primary head, ο οποίος είναι ο κόμβος με την μεγαλύτερη ενέργεια. Η επιλογή αυτή μπορεί να οδηγήσει κάποιους από τους πιο απομακρυσμένους κόμβους, στην κατανάλωση περισσότερης ενέργειας. Για να το αποφύγουμε αυτό, εκλέγουμε ένα δεύτερο cluster head, τον secondary, ο οποίος τοποθετείται σε μεγάλη απόσταση από τον primary. Έτσι δίνουμε την επιλογή σε κάποιους από τους κόμβους να επιλέξουν ένα κοντινότερο κόμβο. Ένας κόμβος μπορεί να αναλάβει την θέση του secondary ενόσω η απόστασή του (από τον primary) αυξάνεται και η διαθέσιμη ενέργειά του δεν μειώνεται σε σχέση με την απόσταση και την ενέργεια κάποιου άλλου κόμβου της ίδιας συστάδας.

Από τους κόμβους που επιλέγουν τον primary και από αυτούς που επιλέγουν τον secondary, μπορούμε να ξεχωρίσουμε δύο υποσυστάδες. Ένας από τους κόμβους που ανήκει στην υποσυστάδα του primary και βρίσκεται στην κοντινότερη απόσταση από τον secondary, ορίζεται ως territorial. Το κριτήριο της διαθέσιμης ενέργειας, χρησιμοποιείται επίσης για την εκλογή αυτού του κόμβου. Μπορούμε να πούμε ότι ο territorial επιλέγεται ενόσω ανήκει στην υποσυστάδα του primary, η απόσταση από τον secondary μειώνεται ενώ η ενέργεια του όχι. Αυτό υπολογίζεται πάντοτε σε σχέση με την απόσταση και την ενέργεια κάποιου άλλου κόμβου της ίδιας συστάδας.

#### 3.3.2. Συγκέντρωση Πληροφορίας

Αφού ολοκληρωθεί η διαδικασία της εκλογής των cluster head και του cluster leader, ξεκινά η διαδικασία της συλλογής της πληροφορίας. Η λειτουργία αυτή, περιλαμβάνει την συγκέντρωση της πληροφορίας των κόμβων μιας συστάδας στους cluster head και μετέπειτα στον cluster leader. Η διαδικασία της συσταδοποίησης ξεκινά και τελειώνει από τον σταθμό βάσης. Όταν χωρίσει το δίκτυο σε συστάδες, ενημερώνει κάθε κόμβο αποστέλλοντας του ένα πίνακα με την τοποθεσία και την διαθέσιμη ενέργεια των κόμβων με τους οποίους βρίσκεται στην ίδια συστάδα. Έχοντας αυτόν τον πίνακα, κάθε κόμβος μπορεί να υπολογίσει την θέση των τριών cluster head, του cluster leader και μετέπειτα να επιλέξει αυτόν που βρίσκεται στην κοντινότερη απόσταση από τον ίδιο.

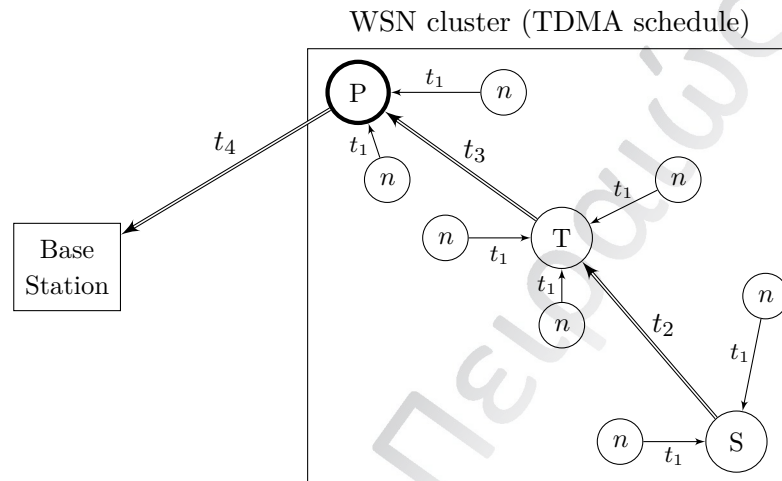
Η παραπάνω διαδικασία μπορεί να εκτελεστεί τοπικά σε κάθε κόμβο αποφεύγοντας έτσι το κόστος αποστολής και λήψης δεδομένων. Όλοι οι κόμβοι λειτουργούν επιλέγοντας με βάση τα ίδια κριτήρια επιλογής, οπότε όλοι θα εκλέξουν τους ίδιους κόμβους. Στην συνέχεια, οι κόμβοι δημιουργούν ένα πρόγραμμα TDMA και ξεκινούν την μετάδοση των πληροφοριών τους. Όταν η μετάδοση ολοκληρωθεί κάθε κόμβος έχει προωθήσει τα δεδομένα που συνέλεξε σε έναν από τους cluster head ή στον cluster leader. Το μόνο που απομένει είναι να προωθηθεί από τους cluster head στον cluster leader. Η διαδικασία αυτή είναι ανάλογη με τον κόμβο που επιλέγεται ως cluster leader και γίνεται με δύο τρόπους χρησιμοποιώντας πάντοτε ένα TDMA πρόγραμμα:

1. Στην περίπτωση που ο primary επιλεγεί ως cluster leader, τότε η μετάδοση αρχίζει από τον secondary στον territorial και μετέπειτα από τον territorial στον primary.

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

Αν ο κόμβος territorial δεν υπάρχει, τότε η μετάδοση γίνεται απευθείας από τον secondary στον primary.

2. Στην περίπτωση που ο secondary επιλεγεί ως cluster leader, τότε η μετάδοση αρχίζει από τον primary στον territorial (αν υπάρχει) και μετέπειτα από τον territorial στον secondary.



Σχήμα 3.3.: Πρόγραμμα TDMA για την συλλογή της πληροφορίας σε μία συστάδα.

Στο Σχήμα 3.3 έχουμε την διαδικασία συλλογής της πληροφορίας για μία συστάδα η οποία αποτελείται από  $C' = 10$  κόμβους. Ο κόμβος με σκούρο χρώμα  $P$  (Primary) συμβολίζει τον κόμβο cluster leader. Οι επόμενοι κόμβοι  $S, T$  συμβολίζουν τους κόμβους Secondary και Territorial αντίστοιχα. Οι κόμβοι  $n$  συμβολίζουν τους υπόλοιπους κόμβους της συστάδας. Η διαδικασία ξεκινά την χρονική στιγμή  $t_1$ , όπου οι κόμβοι  $n$  προωθούν τα δεδομένα τους στον αντίστοιχο προορισμό. Στην χρονική στιγμή  $t_2$ , ο cluster head secondary προωθεί τα δεδομένα του στον territorial. Ο territorial, αφού αναμένει για χρονικό διάστημα  $t_2$ , μεταδίδει στο επόμενο διάστημα  $t_3$  στον κόμβο primary. Ο κόμβος primary, αφού λάβει την χρονική στιγμή  $t_3$ , μεταδίδει στην  $t_4$  στον προορισμό. Θα πρέπει να επισημάνουμε ότι σε καθένα από τους τρεις κόμβους ( $P, T, S$ ) χρησιμοποιούνται τεχνικές data aggregation και data compression προκειμένου να μειωθεί το μέγεθος της πληροφορίας που μεταδίδεται.

Μετά την ολοκλήρωση ενός round, οι κόμβοι αναμένουν για ένα μικρό χρονικό διάστημα (διάστημα αναμονής) όπου υπάρχει η πιθανότητα να λάβουν ενημερώσεις από τον σταθμό βάσης. Οι ενημερώσεις αφορούν μεταβολές στον τρόπο λειτουργίας των κόμβων του δικτύου. Πριν την έναρξη του πρώτου round, οι κόμβοι έχουν λάβει πίνακες με τους κόμβους που βρίσκονται στην ίδια συστάδα. Σε αυτό το σημείο θεωρείται ότι όλοι οι κόμβοι είναι λειτουργικοί. Ωστόσο, κατά την διάρκεια της συλλογής της πληροφορίας κάποιος από τους κόμβους μπορεί να εμφανίσει κάποια δυσλειτουργία ή κάποιο αδιάπεραστο εμπόδιο να εμποδίσει την μετάδοσή του. Αν ένας cluster head ή cluster leader



δεν λάβει πληροφορία (μήνυμα) από αυτούς που έχει υπολογίσει, προσθέτει πληροφορία στο πακέτο του με στόχο να ενημερώσει τον σταθμό βάσης.

Έπειτα, κατά το διάστημα της αναμονής, ο σταθμός βάσης ενημερώνει τους υπόλοιπους κόμβους της συστάδας ώστε να αφαιρέσουν τον κόμβο από τον πίνακά τους. Από την πλευρά του κόμβου, αν είναι λειτουργικός και η αποστολή του απέτυχε λόγω κάποιου προσωρινού προβλήματος, τότε θα διαπιστώσει ότι έχει αφαιρεθεί από τον πίνακα της συστάδας. Σε αυτή την περίπτωση θα λειτουργήσει μόνος του, χωρίς να επηρεάζει την λειτουργία του δικτύου, προωθώντας τα δεδομένα του και την κατάσταση της ενέργειάς του στον προορισμό. Αν ο σταθμός βάσης λάβει πληροφορία από ένα κόμβο με κωδικό που ανήκει σε μία συστάδα αλλά δεν υπάρχει στους πίνακες των κόμβων, τότε επαναλαμβάνει την διαδικασία ενημέρωσης των πινάκων στο διάστημα της αναμονής.

Ένα χαρακτηριστικό της ΜΕΕΑΔΑ είναι ότι η διαθέσιμη ενέργεια κάθε κόμβου της συστάδας θα μπορούσε να υπολογίζεται με βάση το μοντέλο ενέργειας που έχει υιοθετηθεί από το δίκτυο. Αυτό σημαίνει ότι θα υπολογίζεται τοπικά, χωρίς να είναι απαραίτητη η συνεννόηση των κόμβων μέσα από ενεργοβόρες αποστολές και λήψεις δεδομένων. Στις περιπτώσεις που κάποιος κόμβος διαγραφεί από τους πίνακες ενώ διαθέτει ακόμη ενέργεια, μπορεί να προωθήσει τα δεδομένα του ως cluster leader, με τον τρόπο που περιγράψαμε στην προηγούμενη παράγραφο. Βέβαια, ένας τέτοιος τρόπος υπολογισμού μπορεί να μην είναι ακριβής και θα πρέπει να δοκιμαστεί σε μία πραγματική υλοποίηση.

#### 3.3.3. Αλγόριθμος Δρομολόγησης

Έως τώρα περιγράψαμε την διαδικασία της συλλογής της πληροφορίας σε μία συστάδα, όπου όλα τα δεδομένα των κόμβων καταλήγουν στον cluster leader. Στο επόμενο και τελικό βήμα, έχουμε την διαδικασία της δρομολόγησης. Ένα WSN μπορεί να περιλαμβάνει περισσότερους από ένα κόμβους cluster leader. Οι κόμβοι αυτοί συνεργάζονται μεταξύ τους, προκειμένου να συγκεντρώσουν την πληροφορία ολόκληρου του δικτύου και να την προωθήσουν στον σταθμό βάσης, με το ελάχιστο δυνατό κόστος ενέργειας. Για τον σκοπό αυτό χρησιμοποιείται μία μέθοδος η οποία ονομάζεται δρομολόγηση μη αρνητικής απόστασης. Η μέθοδος αυτή βασίζεται στις αποστάσεις μεταξύ των κόμβων και σε αυτές από τον σταθμό βάσης.

Για να είναι ακριβής η διαδικασία της δρομολόγησης, οι κόμβοι που έχουν οριστεί ως cluster leader θα πρέπει να γνωρίζουν την θέση των αντίστοιχων κόμβων των άλλων συστάδων. Στην πιο απλή της μορφή, η γνωστοποίηση της θέσης μπορεί να γίνει από τον σταθμό βάσης κατά την διάρκεια της διαδικασίας αναμονής. Ο σταθμός βάσης, αφού υπολογίσει το σχέδιο δρομολόγησης (Routing Plan) ενημερώνει τους cluster leader για την χρονική στιγμή που θα μεταδώσουν σε μία γειτονική συστάδα ή στον ίδιο. Η διαδικασία μπορεί να πραγματοποιηθεί και από τους κόμβους, αν οι πίνακες που διαθέτουν τους παρέχουν καθολική γνώση του δικτύου. Ωστόσο, οι μεγαλύτεροι πίνακες μπορεί να οδηγήσουν στην αύξηση την κατανάλωσης ενέργειας, σε δίκτυα με χιλιάδες κόμβους, καθώς κάθε κόμβος θα πρέπει να υπολογίσει τους cluster head όλων των συστάδων.

Ένα σχέδιο διομολόγησης περιλαμβάνει αρχικά μόνο τους cluster leader κάθε συστάδας. Σε αυτή την χρονική στιγμή οι κόμβοι αυτοί είναι οι κόμβοι primary, οι οποίοι διαθέτουν την περισσότερη ενέργεια. Όταν οι κόμβοι primary χάσουν αρκετή από την

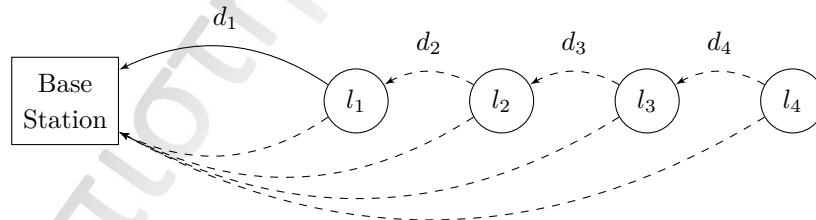
### 3. Μέθοδος Εξοικονόμησης Ενέργειας

ενέργειά τους, τότε ως cluster leader (*Exhead*) εκλέγεται και ο κόμβος secondary. Η επιλογή μεταξύ των δύο εξαρτάται από την απόσταση των κόμβων αυτών από τον σταθμό βάσης, με τον κοντινότερο να έχει την προτεραιότητα. Με αυτό τον τρόπο, κάθε κόμβος cluster leader του δικτύου μπορεί να προωθεί την πληροφορία του από  $\hat{N} - 1$  σε  $2 * \hat{N} - 2$  κόμβους. Ο πρώτος λόγος για τον οποίο γίνεται αυτή η εναλλαγή, είναι διότι οι κόμβοι μπορεί να μην διαθέτουν την ίδια ενέργεια αρχικά.

Ο δεύτερος λόγος για τον οποίο υιοθετείται μία προσαρμοστική επιλογή του cluster leader, είναι η συχνότητα με την οποία ένας κόμβος αναλαμβάνει την θέση του secondary. Η συχνότητα αυτή, είναι μεγαλύτερη από αυτή του primary. Αν ο κόμβος secondary, είχε από την αρχή την δυνατότητα να εκλεγεί ως cluster leader, τότε θα έχουμε μία ανισορροπία σε ό,τι αφορά την κατανάλωση της ενέργειας στο δίκτυο. Το γεγονός αυτό μπορεί να οδηγήσει κάποιους από τους κόμβους στο να χάσουν την ενέργεια τους πολύ γρήγορα. Η “πρόωρη” απώλεια κόμβων στο δίκτυο, δημιουργεί κενά που αυξάνουν τις αποστάσεις μεταξύ των κόμβων.

Καθορίζοντας τον cluster leader κάθε συστάδας, ξεκινάμε με την δημιουργία του σχεδίου δρομολόγησης. Το σχέδιο είναι ανάλογο με την θέση του σταθμού βάσης και υλοποιείται μέσα από τον αλγόριθμο μη αρνητικής απόστασης. Αυτό που προσπαθούμε να δημιουργήσουμε, είναι ένα κατευθυνόμενο άκυκλο μονοπάτι, το οποίο θα περιλαμβάνει όλους τους κόμβους του δικτύου και θα καταλήγει στον προορισμό. Η συνολική απόσταση του μονοπατιού θα πρέπει να είναι η μικρότερη από όλους τους πιθανούς συνδυασμούς. Ο υπολογισμός της χαμηλού κόστους διαδρομής βασίζεται σε μία απόσταση που ονομάζεται μη αρνητική.

**Ορισμός 1.** Η απόσταση ενός cluster leader  $[j] \in \text{cluster } [j]$  από τον base station που είναι μικρότερη από αυτή ενός cluster leader  $[i] \in \text{cluster } [i]$  ονομάζεται “μη αρνητική απόσταση”. Ισχύει ότι  $i \neq j$ .



Σχήμα 3.4.: Υπολογισμός μονοπατιού δρομολόγησης.

Στο Σχήμα 3.4 παρατηρούμε ένα σχέδιο δρομολόγησης για ένα απλό δίκτυο, το οποίο περιλαμβάνει ένα σύνολο cluster leader  $L' = \{l_1, l_2, l_3, l_4\}$ . Από το σύνολο αυτό και τις αποστάσεις από τον σταθμό βάσης (διακεκομμένες ακμές), προκύπτει μία διαδρομή (συμπαγείς ακμές) όπου κάθε κόμβος εκμεταλλεύεται τους γειτονικούς του, προκειμένου να μεταδώσει τα δεδομένα του. Το κόστος της μετάδοσης της πληροφορίας του δικτύου ισούται με τις αποστάσεις  $d_1, d_2, d_3, d_4$  που θα πρέπει να μεταδώσει καθένας από τους

κόμβους. Οι υπολογισμοί γίνονται σύμφωνα με τον παρακάτω αλγόριθμο δρομολόγησης μη αρνητικής απόστασης (Αλγόριθμος 3).

---

**Αλγόριθμος 3: ΔΡΟΜΟΛΟΓΗΣΗ ΜΗ ΑΡΝΗΤΙΚΗΣ ΑΠΟΣΤΑΣΗΣ ( $R$ )**


---

**Είσοδος:** Ένα πεπερασμένο σύνολο  $L' = \{l_1, l_2, \dots, l_n\}$  όπου  $l_i$  ο κωδικός συστάδας ενός cluster leader στην θέση  $l_{x,y}$  (όπου  $x$  και  $y$  οι συντεταγμένες του  $l_i$  στον χώρο).

**Έξοδος:** Ένας προορισμός μετάδοσης για κάθε  $l_i \in L'$ .

1.  $tl_{x,y} \leftarrow BS_{x,y}$
  2.  $adist, bdist, cdist, ddist, edist \leftarrow 0$
  3.  $maxdist1, maxdist2 \leftarrow r$
  4.  $c \leftarrow 0$
  5.  $v \leftarrow \max l \in L'$
  6. **ενόσω** ( $c \neq v + 1$ )
  7.     **για**  $i \leftarrow 1$  **έως**  $maxX$
  8.         **για**  $j \leftarrow 1$  **έως**  $maxY$
  9.             **Αν** ( $l = c$  **και** ( $l_{x,y} = Primary_{x,y}$  **ή**  $l_{x,y} = Exhead_{x,y}$ ))
  10.                  $adist \leftarrow$  ΕΥΚΛΕΙΔΕΙΑ ΑΠΟΣΤΑΣΗ ( $l_{x,y}, BS_{x,y}$ )
  11.                 **Αν** ( $adist < maxdist1$ )
  12.                      $maxdist1 \leftarrow adist$
  13.                      $cr_{x,y} \leftarrow l_{x,y}$
  14.     **Αν** ( $cr_{x,y} =$  Μη ορισμένη για  $l_i = c$ )
  15.         Μετάβαση στην γραμμή 30
  16.     **για**  $i \leftarrow 1$  **έως**  $maxX$
  17.         **για**  $j \leftarrow 1$  **έως**  $maxY$
  18.             **Αν** ( $l \neq c$  **και** ( $l_{x,y} = Primary_{x,y}$  **ή**  $l_{x,y} = Exhead_{x,y}$ ))
  19.                  $bdist \leftarrow$  ΕΥΚΛΕΙΔΕΙΑ ΑΠΟΣΤΑΣΗ ( $l_{x,y}, BS_{x,y}$ )
  20.                  $cdist \leftarrow$  ΕΥΚΛΕΙΔΕΙΑ ΑΠΟΣΤΑΣΗ ( $l_{x,y}, cr_{x,y}$ )
  21.                 **Αν** ( $bdist < maxdist1$  **και**  $cdist < maxdist2$ )
  22.                      $maxdist2 \leftarrow cdist$
  23.                      $tl_{x,y} \leftarrow l_{x,y}$
  24.      $ddist \leftarrow$  ΕΥΚΛΕΙΔΕΙΑ ΑΠΟΣΤΑΣΗ ( $cr_{x,y}, BS_{x,y}$ )
  25.      $edist \leftarrow$  ΕΥΚΛΕΙΔΕΙΑ ΑΠΟΣΤΑΣΗ ( $cr_{x,y}, tl_{x,y}$ )
  26.     **Αν** ( $ddist > edist$ )
  27.         ΚΟΜΒΟΣ ΠΡΟΟΡΙΣΜΟΣ  $cr_{x,y} \leftarrow tl_{x,y}$
  28.     **Άλλως**
  29.         ΚΟΜΒΟΣ ΠΡΟΟΡΙΣΜΟΣ  $cr_{x,y} \leftarrow BS_{x,y}$
  30.      $tl_{x,y} \leftarrow BS_{x,y}$
  31.      $maxdist1, maxdist2 \leftarrow maxdist$
  32.      $adist, bdist, cdist, ddist, edist \leftarrow 0$
  33.      $c \leftarrow c + 1$
  34. Ολοκλήρωση υπολογισμού προορισμών για ένα round
- 

Ο αλγόριθμος ξεκινά έχοντας ως είσοδο ένα σύνολο  $L'$  από cluster leader. Στόχος

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

του είναι να υπολογίσει τον προορισμό μετάδοσης όλων των κόμβων του συνόλου  $L'$ . Στην αρχή (γραμμική 1), ορίζουμε μία μεταβλητή  $tl$  με σημεία  $x, y$  στον χώρο τα οποία αντιστοιχούν στην θέση  $x, y$  του σταθμού βάσης  $BS$ . Έπειτα (γραμμικές 2-3), ορίζουμε ένα σύνολο αποστάσεων οι οποίες θα χρειαστούν για τον υπολογισμό των προορισμών. Οι αποστάσεις  $adist, bdist, cdist, ddist$  έχουν οριστεί ως 0 ενώ οι  $maxdist1, maxdist2$  έχουν την τιμή μίας μεγάλης απόστασης  $r$  (βλ. Αλγόριθμος 1). Στην συνέχεια (γραμμικές 4-5), ορίζουμε τον αρχικό κωδικό συστάδας  $c$ , για τον οποίο αναζητούμε τον προορισμό και μία μεταβλητή  $v$  που ισούται με τον μέγιστο κωδικό  $l \in L'$ . Η τιμή του  $c$  είναι ίση με 0 καθώς κατά την διαδικασία της συσταδοποίησης οι κωδικοί των τελικών συστάδων (βλ. Αλγόριθμος 2) ξεκινούν από το 0.

Η κύρια λειτουργία του αλγορίθμου (γραμμικές 6-33) εκτελείται για κάθε κωδικό κάθε κόμβου του συνόλου  $L'$ . Η διαδικασία περιλαμβάνει (γραμμικές 7-13) την αναζήτηση του κοντινότερου στον σταθμό βάσης κόμβου  $cr$  με κωδικό  $c$ . Η επιλογή (γραμμική 9) γίνεται μέσα από τους κόμβους *Primary* και *Exhead* με την χρήση της ευκλείδειας απόστασης. Ο κόμβος *Exhead* διαθέτει την τιμή του *Primary* ενόσω όλοι οι κόμβοι του δικτύου διαθέτουν ενέργεια. Αν κάποιος από τους κόμβους χάσει όλη την ενέργειά του και τεθεί εκτός δικτύου, τότε η τιμή του *Exhead* είναι ίση με αυτή του *Secondary*. Βέβαια, θα πρέπει να προσέξουμε ότι μετά από ένα διάστημα  $t$ , μία συστάδα μπορεί να μην διαθέτει κανένα κόμβο. Αν ο κόμβος  $cr$ , δεν οριστεί για μία συστάδα με κωδικό  $c$  (γραμμικές 14-15), ο αλγόριθμος αγνοεί την μετέπειτα διαδικασία και ξεκινά μία νέα αναζήτηση με διαφορετικό κωδικό.

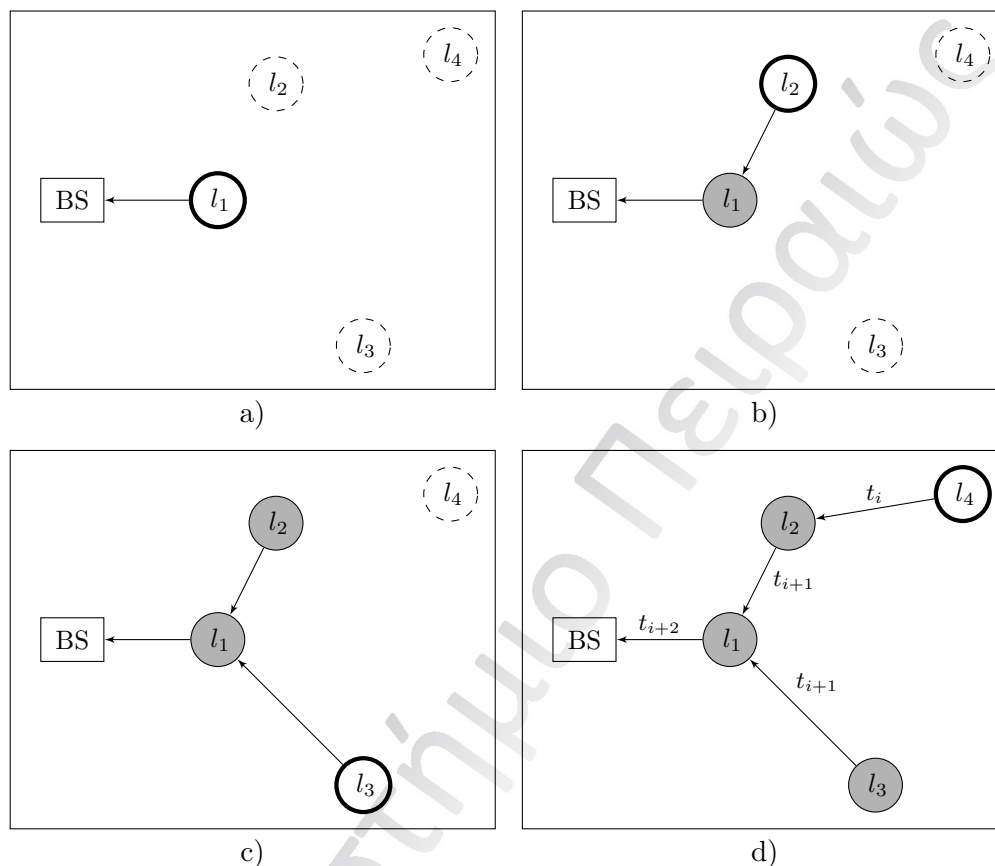
Αφού ορίσουμε την τιμή του κόμβου  $cr$  τότε αναζητούμε (γραμμικές 16-23) τον κόμβο στον οποίο θα προωθήσει τα δεδομένα του. Αυτή την φορά, η επιλογή αφορά τους κόμβους *Primary* και *Exhead* με διαφορετικό κωδικό από αυτόν του  $cr$ . Οι δύο απαραίτητες αποστάσεις (γραμμικές 19-20) που θα μας βοηθήσουν στην εύρεση του κατάλληλου κόμβου προορισμού, είναι αυτή του κόμβου  $l$  από τον  $BS$  και αυτή του  $l$  από τον  $cr$ . Στην πράξη, ο αλγόριθμος αναζητά (γραμμικές 21-23) τον κοντινότερο κόμβο  $tl$ , η απόσταση του οποίου από τον  $BS$ , είναι μικρότερη από αυτή του  $cr$  από τον  $BS$ . Όταν βρεθεί και ο υποψήφιος κόμβος προορισμός  $tl$ , τότε συγκρίνουμε (γραμμικές 24-29) την απόσταση του από τον  $cr$ , με αυτή του  $cr$  από τον  $BS$ . Στην περίπτωση (γραμμική 26) που η απόσταση μετάδοσης του  $cr$  προς τον  $BS$  είναι μεγαλύτερη από την απόσταση μετάδοσης προς τον  $tl$ , τότε ως κόμβος προορισμός (γραμμική 27) επιλέγεται ο κόμβος  $tl$ . Σε αντίθετη περίπτωση (γραμμική 29), επιλέγεται ο  $BS$ .

Όταν βρεθεί ο κόμβος προορισμός (γραμμικές 30-33), έχουμε την αρχικοποίηση των μεταβλητών και την αύξηση του κωδικού της συστάδας  $c$ . Η διαδικασία επαναλαμβάνεται για όλους τους κωδικούς συστάδων. Όταν ολοκληρώνεται (γραμμική 34), μπορούμε να σχηματίσουμε μία διαδρομή η οποία περιέχει όλους τους κόμβους και καταλήγει στον σταθμό βάσης. Στον αλγόριθμο αυτό, δύο κόμβοι μπορούν να έχουν τον ίδιο προορισμό, όταν οι αποστάσεις τους από τον σταθμό βάσης ή από ένα κόμβο είναι ίσες. Αυτό σημαίνει ότι σε ειδικές περιπτώσεις, είναι δυνατό να προωθήσουν την πληροφορία στον σταθμό βάσης περισσότεροι από ένας κόμβοι.

Η δρομολόγηση μη αρνητικής απόστασης δημιουργεί μονοπάτια χαμηλού κόστους, για οποιαδήποτε θέση του σταθμού βάσης στο δίκτυο. Αν και χρησιμοποιώντας την μέθοδο αυτή, είναι δυνατή η εξοικονόμηση σημαντικής ενέργειας, αυξάνονται οι καθυστερήσεις

### 3.3. Διαδικασία Δρομολόγησης

μετάδοσης. Ένας κόμβος δεν μπορεί να προωθήσει τα δεδομένα του, αν πρώτα δεν λάβει την πληροφορία από τους κόμβους που τον έχουν επιλέξει ως κόμβο προορισμό. Για τον σκοπό αυτό δημιουργείται ένα πρόγραμμα TDMA το οποίο επηρεάζει ακόμα και την διαδικασία της συλλογής σε κάθε συστάδα. Ο κόμβος που μεταδίδει στον σταθμό βάσης θα περιμένει όλους τους υπόλοιπους. Εκείνος που δεν λαμβάνει από κανένα ξεκινά συνήθως την διαδικασία της αποστολής.



Σχήμα 3.5.: Δρομολόγηση μη αρνητικής απόστασης.

Ένα παράδειγμα για την βήμα προς βήμα εκτέλεση του αλγορίθμου δρομολόγησης (βλ. Αλγόριθμος 3) είναι αυτό του Σχήματος 3.5. Για ένα σύνολο τεσσάρων cluster leader  $L' = \{l_1, l_2, l_3, l_4\}$ , δημιουργούμε μία διαδρομή χαμηλού κόστους, ώστε να προωθήσουμε την πληροφορία των συστάδων του δικτύου. Στο παράδειγμά μας θεωρούμε ότι κάθε συστάδα εκπροσωπείται μόνο από τον κόμβο *Primary* και ο κόμβος *Exhead* είναι ανενεργός. Ο κόμβος με σκούρο περίγραμμα, συμβολίζει ένα κόμβο  $l_i$  για τον οποίο αναζητείται ένας κόμβος προορισμός. Οι κόμβοι με διακεκομμένο περίγραμμα, συμβολίζουν εκείνους που η απόστασή τους δεν είναι μη αρνητική για τον επιλεγμένο  $l_i$ . Με γκριζό χρώμα συμβολίζονται οι κόμβοι με μη αρνητικές αποστάσεις.

Ο επιλεγμένος κόμβος  $l_i$  αναζητά ως κόμβο προορισμό τον κοντινότερο από τους

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

γκρίζους κόμβους. Όλοι οι υπόλοιποι cluster leader (διακεκομμένο περίγραμμα) τίθενται εκτός επιλογής, ακόμη και αν βρίσκονται σε πολύ κοντινή απόσταση. Αν για ένα κόμβο δεν υπάρχουν διαθέσιμες επιλογές, τότε ο προορισμός μετάδοσής του θα είναι ο σταθμός βάσης. Όταν ολοκληρωθεί η διαδικασία της δρομολόγησης, ακολουθεί ο υπολογισμός του προγράμματος TDMA, όπου καθορίζονται οι χρόνοι μετάδοσης του κάθε cluster leader. Το πρόγραμμα που δημιουργείται επηρεάζει και την διαδικασία της συλλογής της πληροφορίας (βλ. Σχήμα 3.3).

Η διαδικασία της δρομολόγησης λειτουργεί για οποιαδήποτε θέση του σταθμού βάσης στον χώρο. Επίσης, είναι δυνατή και η τοποθέτησή του εντός του δικτύου. Σε μία τέτοια περίπτωση είναι πολύ πιθανό οι κόμβοι να προωθήσουν τα δεδομένα τους απευθείας στον προορισμό, καθώς η επιλογή ενός άλλου κόμβου μπορεί να είναι λιγότερο συμφέρουσα. Έχοντας λοιπόν δημιουργήσει ένα αλγόριθμο για την δημιουργία χαμηλού κόστους μονοπατιών, είναι σημαντικό να βεβαιωνούμε ότι λειτουργεί χωρίς προβλήματα. Το σημαντικότερο από αυτά, είναι να μην υπάρχουν κύκλοι δρομολόγησης που περιορίζουν την μετάδοση της πληροφορίας μεταξύ  $n$  κόμβων. Το σενάριο που θα εξετάσουμε, αφορά την δρομολόγηση που περιλαμβάνει μόνο τους κόμβους *Primary*.

Για παράδειγμα, αν ένας κόμβος  $l_a$  επιλέξει ως κόμβο προορισμό τον  $l_b$  και ο  $l_b$  τον  $l_a$ , τότε η πληροφορία θα απομονωθεί μεταξύ των δύο και δεν θα φτάσει ποτέ στον προορισμό. Για να συμβεί αυτό, θα πρέπει να ισχύει η παρακάτω σχέση:

$$dist_{a,bs} < dist_{b,bs} \text{ και } dist_{a,bs} > dist_{b,bs} \quad (3.2)$$

όπου  $dist_{a,bs}$  η απόσταση του  $l_a$  από τον *BS* και  $dist_{b,bs}$  η απόσταση του  $l_b$  από τον *BS*. Σύμφωνα με την σχέση (3.2) ένας κόμβος  $l_a$  θα επιλέξει τον  $l_b$  όταν η απόστασή του από τον *BS*, είναι μικρότερη από την αντίστοιχη του  $l_b$ . Ακολουθώντας μία παρόμοια στρατηγική, ο  $l_b$  θα επιλέξει ως προορισμό τον  $l_a$ . Ωστόσο, κάτι τέτοιο δεν θα συμβεί ποτέ, καθώς μία απόσταση δεν μπορεί να είναι μικρότερη και παράλληλα μεγαλύτερη, για σταθερή θέση του σταθμού βάσης.

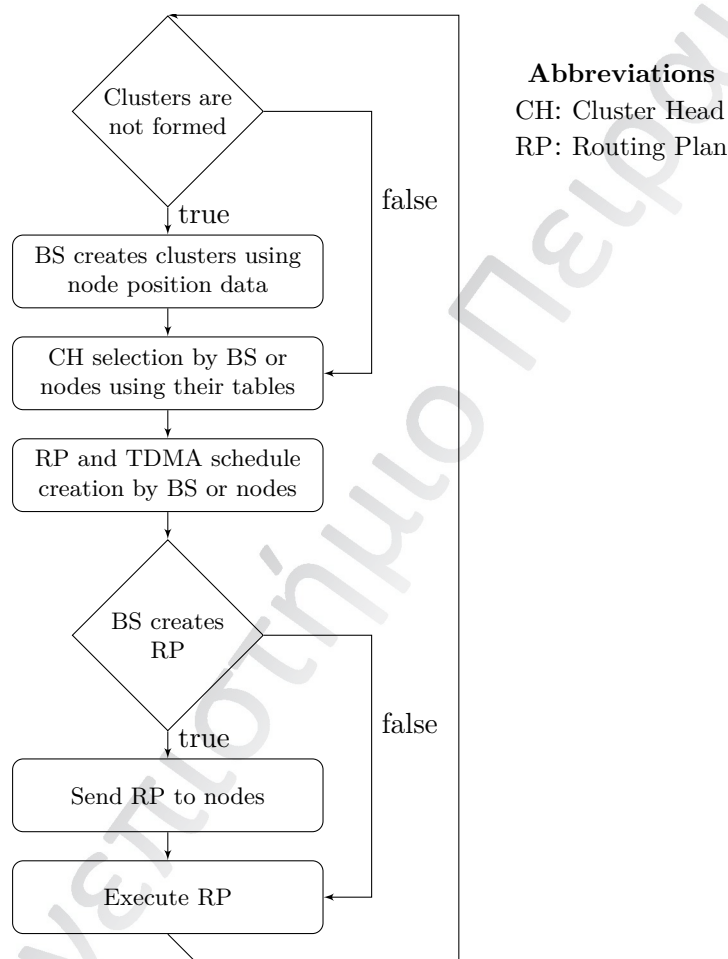
Οι αποστάσεις  $dist_{a,bs}$  και  $dist_{b,bs}$  μπορούν συσχετιστούν μεταξύ τους, μόνο όταν είναι ίσες. Στην περίπτωση αυτή, οι δυο κόμβοι  $l_a, l_b$  θα επιλέξουν ως προορισμό τον *BS* ή κάποιο άλλο κόμβο, χωρίς να δημιουργηθούν προβλήματα. Η παραπάνω σχέση (3.2), μπορεί να γενικευτεί και για περιπτώσεις με περισσότερους από δύο κόμβους. Αυτό γιατί ένας cluster leader  $l_n$ , δεν μπορεί να επιλέξει ένα κόμβο προορισμό με μεγαλύτερη απόσταση από τον *BS* (δεν υπάρχει μη αρνητική απόσταση). Με την ίδια λογική μπορούμε να δείξουμε ότι δεν θα υπάρχουν προβλήματα κύκλων δρομολόγησης όταν αρχίσει και η εκμετάλλευση των κόμβων *Secondary* (*Exthead*). Αυτή την φορά, ως cluster leader κάθε συστάδας εκλέγεται αυτός που βρίσκεται πιο κοντά στον σταθμό βάσης. Η επιλογή αυτή είναι που μας εξασφαλίζει την ορθότητα της δρομολόγησης.

Ένας κόμβος  $l_{ai}$  μπορεί να προωθήσει τα δεδομένα του σε έναν από τους  $l_{b1}$  ή  $l_{b2}$ . Η επιλογή γίνεται ανάλογα με το ποιος είναι ο κοντινότερος. Εάν ο κοντινότερος είναι ο  $l_{b2}$  τότε ισχύει ότι η απόσταση του  $dist_{l_{b2},bs} < dist_{l_{ai},bs}$ . Από την άλλη, η απόσταση του κόμβου  $dist_{l_{b1},bs}$  σε σχέση με την  $dist_{l_{ai},bs}$  μπορεί να είναι είτε μικρότερη, μεγαλύτερη ή ίση. Στο επόμενο βήμα, θα αναζητηθεί ο κόμβος προορισμός για τον κοντινότερο από τον *BS*,  $l_{b1}$  ή  $l_{b2}$ . Όποιος και αν είναι ο κοντινότερος  $l_{bi}$ , δεν θα μπορεί να επιλέξει ως

κόμβο προορισμό έναν από τους  $l_{a1}, l_{a2}$ . Αν ο κόμβος  $l_{ai}$  επέλεξε τον κόμβο  $l_{b2}$ , ο  $l_{b2}$  δεν μπορεί να τον επιλέξει ξανά λόγω του ότι  $dist_{l_{ai},bs} > dist_{l_{b2},bs}$ . Το ίδιο ισχύει και για τον  $l_{b1}$ , από την στιγμή που αυτός κάνει την επιλογή του κόμβου προορισμού, ως ο πιο κοντινός (από τους  $l_{b1}, l_{b2}$ ) στον σταθμό βάσης.

### 3.4. Διαγράμματα Ροής MEEAΔA

Από την περιγραφή της λειτουργίας του προτεινόμενου πρωτοκόλλου MEEAΔA, μπορούμε να δημιουργήσουμε ένα διάγραμμα ροής. Μέσα από το διάγραμμα αυτό περιγράφονται συνοπτικά οι πιο σημαντικές λειτουργίες του πρωτοκόλλου.



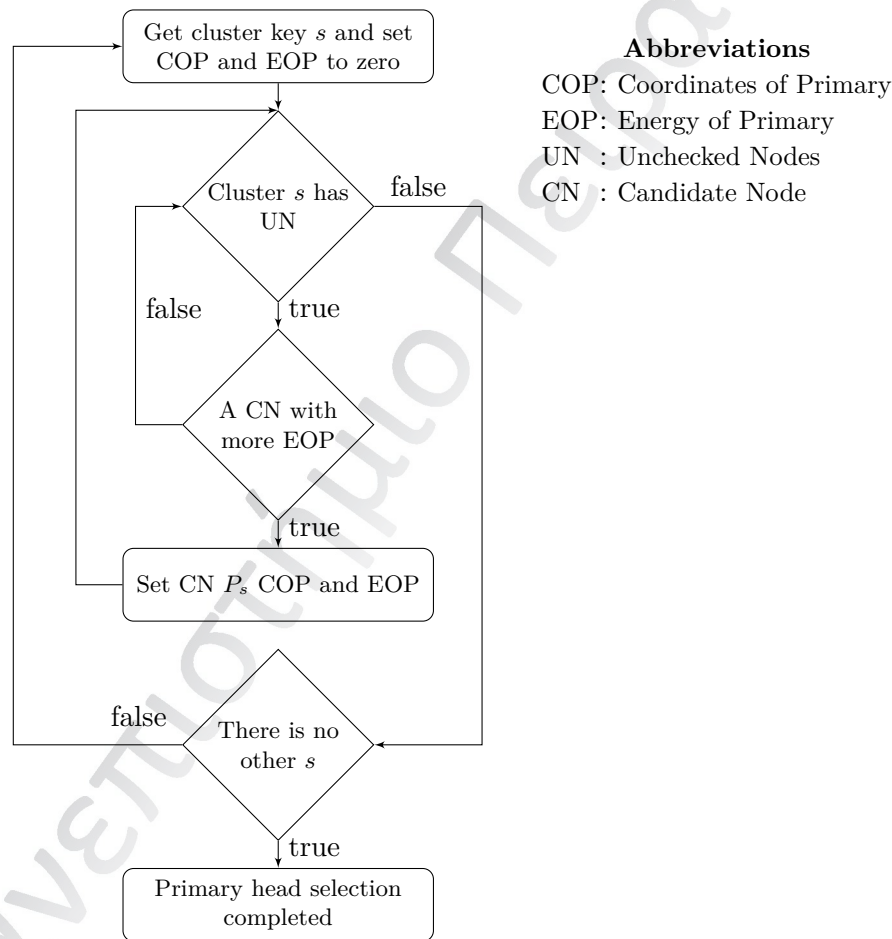
Σχήμα 3.6.: Διάγραμμα ροής με τις κύριες λειτουργίες της MEEAΔA.

Στο διάγραμμα ροής του Σχήματος 3.6 χωρίζουμε την λειτουργία της MEEAΔA σε κύριες διαδικασίες. Η πρώτη λειτουργία του πρωτοκόλλου είναι ο σχηματισμός των συστάδων. Όταν ολοκληρωθεί αυτό το βήμα, τότε ο σταθμός βάσης στέλνει τους ανάλο-

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

γους πίνακες δρομολόγησης σε κάθε κόμβο. Οι κόμβοι διαθέτουν πλέον ένα πίνακα με τους κόμβους που ανήκουν στην ίδια συστάδα. Έπειτα, ακολουθεί η διαδικασία εκλογής των cluster head και του cluster leader σε κάθε συστάδα. Στην συνέχεια, υπολογίζεται το σχέδιο δρομολόγησης, το οποίο περιλαμβάνει τις διαδικασίες της συλλογής και δρομολόγησης της πληροφορίας.

Το σχέδιο αυτό δημιουργείται από τον σταθμό βάσης ή από τους κόμβους και συνοδεύεται από ένα πρόγραμμα TDMA. Αν το σχέδιο δρομολόγησης δημιουργηθεί από τον σταθμό βάσης, τότε θα πρέπει να σταλεί στους κόμβους. Σε αντίθετη περίπτωση, οι κόμβοι εκτελούν το σχέδιο σε μία χρονική στιγμή έναρξης  $t$ . Η επιτυχής εκτέλεση του σχεδίου, συμβολίζει την ολοκλήρωση ενός round. Οι διαδικασίες του διαγράμματος ροής (εξαιρώντας αυτή της συσταδοποίησης) επαναλαμβάνονται έως ότου υπάρχουν κόμβοι με διαθέσιμη ενέργεια.

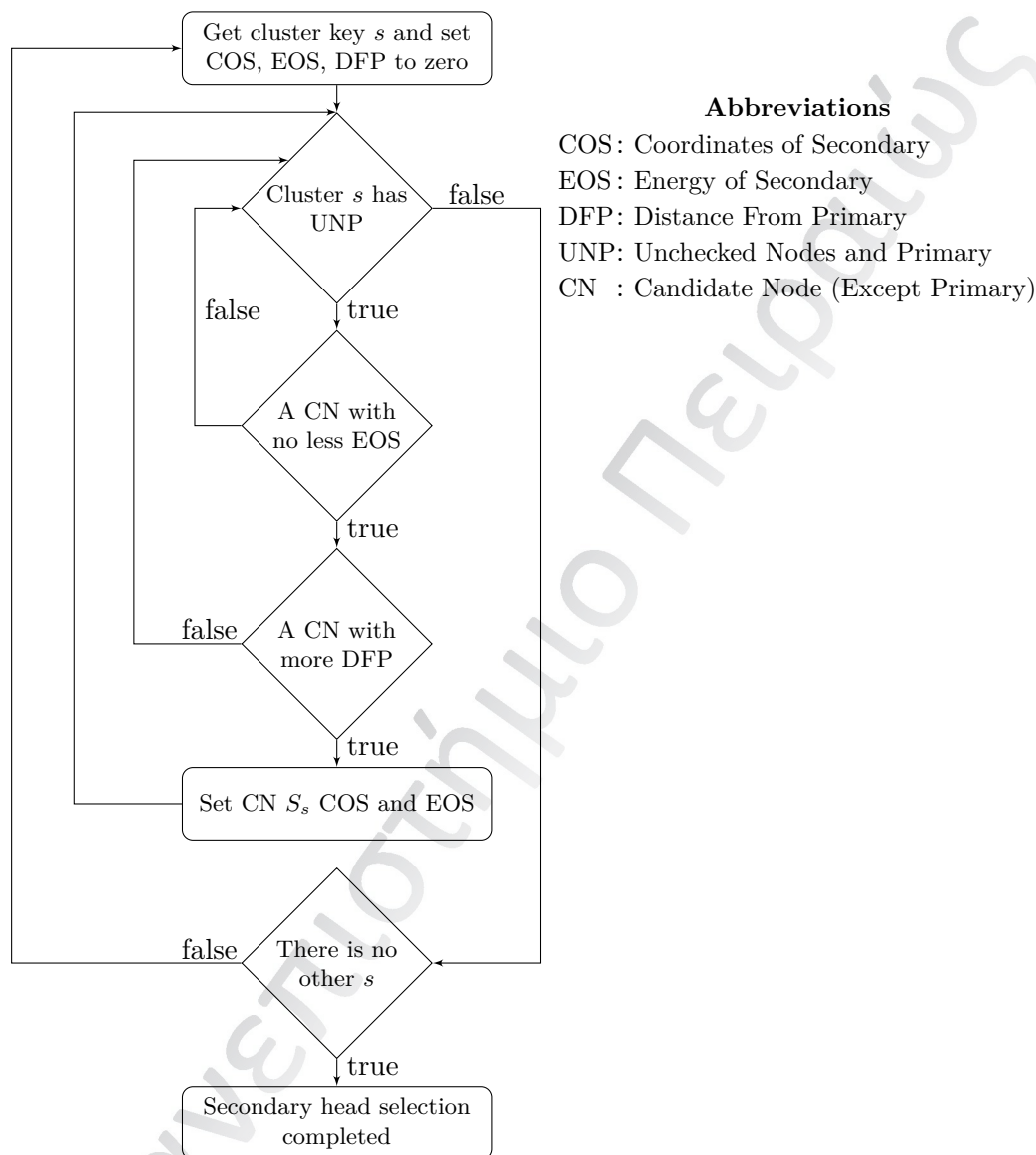


Σχήμα 3.7.: Διάγραμμα ροής για την επιλογή του κόμβου Primary Head.

Στα επόμενα διαγράμματα παρουσιάζεται ο τρόπος με τον οποίο επιλέγονται οι κόμβοι primary, secondary και territorial. Στο διάγραμμα του Σχήματος 3.7, περιγράφεται



η διαδικασία επιλογής του κόμβου primary. Αρχικά, ορίζονται ως μηδενικές (zero) οι συντεταγμένες  $x$ ,  $y$  (Coordinates of Primary - COP) και η ενέργεια του primary (Energy of Primary - EOP). Έπειτα ακολουθεί ο έλεγχος των κόμβων της συστάδας  $s$ . Κάθε κόμβος ελέγχεται μόνο μία φορά. Αν κάποιος από τους υποψήφιους κόμβους (Candidate Node - CN) διαθέτει περισσότερη ενέργεια από την EOP τότε ορίζεται ως primary και ενημερώνουμε τις COP και EOP.

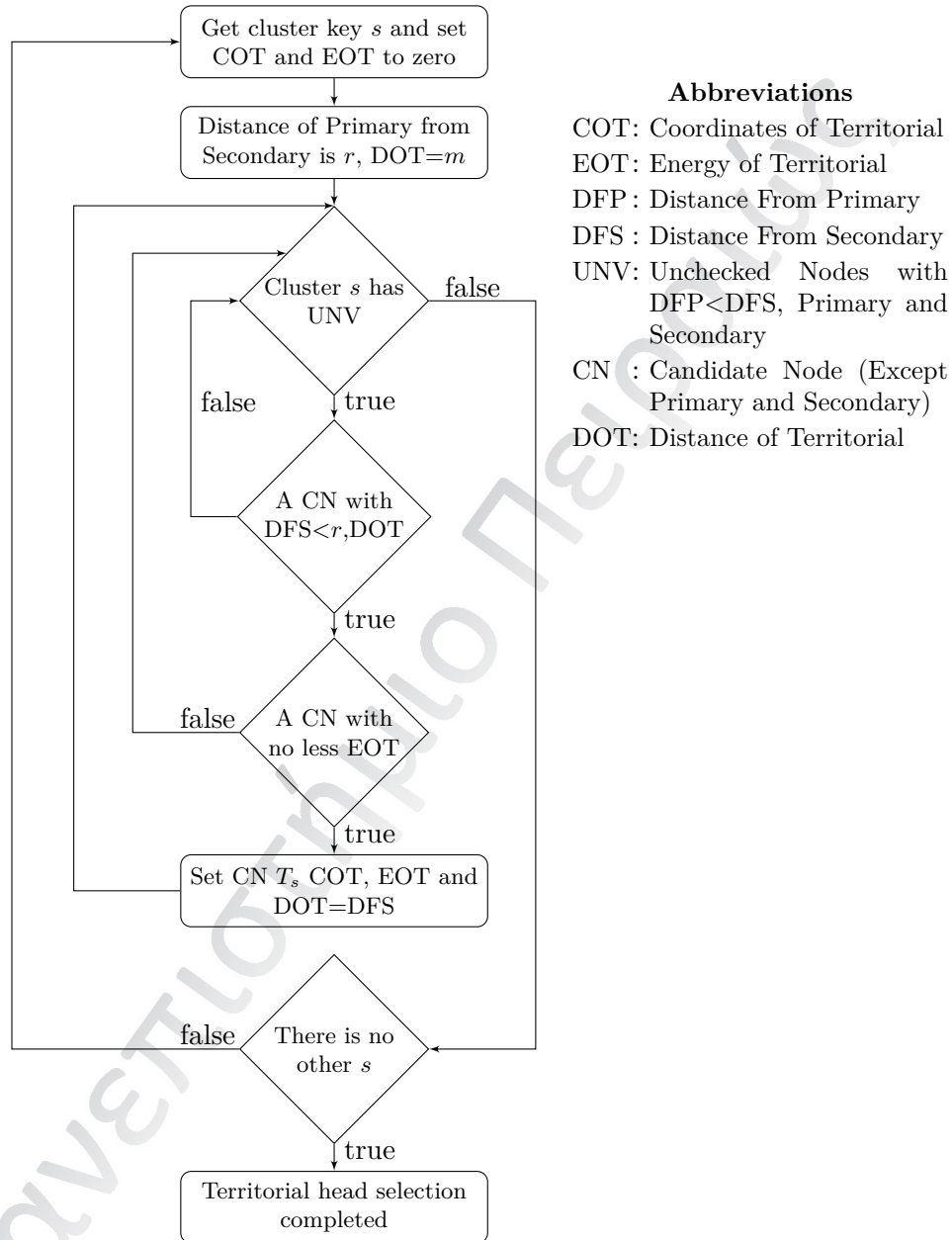


Σχήμα 3.8.: Διάγραμμα ροής για την επιλογή του κόμβου Secondary Head.

Ο επόμενος cluster head είναι ο secondary (Σχήμα 3.8). Αυτή την φορά, η συστάδα με κωδικό  $s$  θα πρέπει να διαθέτει ένα κόμβο primary. Ο CN θα πρέπει να διαθέτει κάθε

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

φορά περισσότερη ή ίση ενέργεια από έναν προκάτοχό του. Επίσης, η DFP (Distance from Primary) του υποψήφιου κόμβου θα πρέπει να είναι μεγαλύτερη από αυτή του προκάτοχου. Ο προκάτοχος είναι ένας κόμβος που έχει ήδη ελεγχθεί και διαθέτει τα προσόντα να αναλάβει την θέση του secondary.



Σχήμα 3.9.: Διάγραμμα ροής για την επιλογή του κόμβου Territorial Head.

Η επιλογή του τελευταίου cluster head territorial (Σχήμα 3.9) είναι πιο σύνθετη από

### 3.5. Δημιουργία TDMA Προγράμματος

αυτή των προηγούμενων. Κάθε συστάδα  $s$  θα πρέπει να διαθέτει κόμβους primary και secondary. Οι κόμβοι που μπορούν να αναλάβουν την θέση διαθέτουν συγκεκριμένες ιδιότητες:

1. Η απόσταση του υποψήφιου κόμβου (CN) από τον primary (DFP) θα πρέπει να είναι μικρότερη από την απόστασή του από τον secondary (DFS).
2. Η απόσταση του CN από τον secondary (DFS) θα πρέπει να είναι μικρότερη από την απόσταση  $r$  (όπου  $r$  η απόσταση του primary από τον secondary) και της απόστασης του territorial (Distance of Territorial – DOT). Η DOT ορίζεται αρχικά ως μία πολύ μεγάλη απόσταση  $m$ , μεγαλύτερη από οποιαδήποτε πιθανή στο δίκτυο.

Εκτός από τα κριτήρια των αποστάσεων, ο CN θα πρέπει να διαθέτει κάθε φορά περισσότερη ή ίση ενέργεια από έναν προκάτοχό του.

Οι διαδικασίες της επιλογής των cluster head εκτελούνται για κάθε κλειδί συστάδας  $s$  και σε κάθε κύκλο δικτύου (round). Όταν κανένας από τους υποψήφιους κόμβους δεν πληροί τις προϋποθέσεις ή δεν υπάρχουν υποψήφιοι κόμβοι για την επιλογή ενός cluster head, τότε οι συντεταγμένες του παραμένουν μηδενικές (zero). Αυτό σημαίνει ότι ο συγκεκριμένος cluster head δεν θα υπάρχει στην συστάδα  $s$ . Μία συστάδα μπορεί να λειτουργήσει αν δεν διαθέτει κόμβους territorial ή secondary. Αν υπάρχει μόνο ένας κόμβος θα αναλάβει την θέση του primary. Όταν χανθεί όλη η ενέργεια, από όλους τους κόμβους της συστάδας, τότε αυτή τίθεται εκτός διαδικασιών συλλογής και δρομολόγησης.

### 3.5. Δημιουργία TDMA Προγράμματος

Η δημιουργία του TDMA προγράμματος περιλαμβάνει την δρομολόγηση της πληροφορίας μεταξύ των cluster leader. Βασίζεται στον κωδικό-κλειδί κάθε συστάδας αποστολέα και στον κωδικό-κλειδί κάθε συστάδας παραλήπτη.

Συστάδα Αποστολέας	Συστάδα Προορισμός
0	2
1	2
2	BS
3	1

Πίνακας 3.1.: Παράδειγμα επιλογής κόμβων προορισμών για τους cluster leader σε ένα round.

Στον Πίνακα 3.1 έχουμε ένα παράδειγμα όπου ο αλγόριθμος δρομολόγησης μη αρνητικής απόστασης (Αλγόριθμος 3) υπολόγισε τις αποστολές και τις λήψεις των δεδομένων μεταξύ των cluster leader. Μέσω αυτού του πίνακα μπορούμε να υπολογίσουμε την χρονική στιγμή (προτεραιότητα) μετάδοσης για κάθε cluster leader. Η διαδικασία ξεκινά

### 3. Μέθοδος Εξοικονόμησης Ενέργειας

με ένα κόμβο ο οποίος ονομάζεται “κόμβος ελέγχου”. Κόμβος ελέγχου μπορεί να γίνει μόνο ένας κόμβος αποστολέας. Αρχικά, επιλέγεται εκείνος που έχει ως παραλήπτη τον σταθμό βάσης. Αυτός ο κόμβος αποκτά την προτεραιότητα 1. Στην συνέχεια, όποιος αποστολέας έχει ως παραλήπτη τον κόμβο ελέγχου, αποκτά την επόμενη προτεραιότητα και ορίζεται ως κόμβος ελέγχου. Η διαδικασία επαναλαμβάνεται έως ότου βρεθούν όλες οι προτεραιότητες για όλους τους κόμβους. Σε κάθε βήμα μπορεί να υπάρχουν περισσότεροι από ένας κόμβοι ελέγχου. Κάθε κόμβος αποστολέας θα πρέπει να επιλεγεί ως κόμβος ελέγχου (αν δεν έχουν βρεθεί όλες οι προτεραιότητες) ακόμα και αν δεν υπάρχει στους παραλήπτες.

S	D	P
0	2	-
1	2	-
2	BS	1
3	1	-

a)

S	D	P
0	2	-
1	2	-
2	BS	1
3	1	-

b)

S	D	P
0	2	2
1	2	2
2	BS	1
3	1	-

c)

S	D	P
0	2	2
1	2	2
2	BS	1
3	1	-

d)

S	D	P
0	2	2
1	2	2
2	BS	1
3	1	3

e)

S	D	P
0	2	2
1	2	2
2	BS	1
3	1	3

f)

Σχήμα 3.10.: Υπολογισμοί προτεραιοτήτων (P) των συστάδων αποστολέων (S) και των παραληπτών (D).

Στο Σχήμα 3.10 παρουσιάζεται ένα παράδειγμα για τέσσερα ζεύγη κόμβων cluster leader. Οι συστάδες αποστολείς εμφανίζονται ταξινομημένοι κατά αύξουσα σειρά. Με γκριζό χρώμα και έντονη γραμματοσειρά εμφανίζονται οι κόμβοι ελέγχου. Με γκριζό χρώμα εμφανίζονται οι παραλήπτες των κόμβων ελέγχου. Στην αρχή (a) αναζητούνται οι κόμβοι αποστολείς με προορισμό τον σταθμό βάσης (BS). Η προτεραιότητά τους ορίζεται 1 και ο αποστολέας του (b) με κλειδί 2, ως κόμβος ελέγχου. Έπειτα (c) αναζητούνται οι κόμβοι παραλήπτες με κλειδί 2 και η προτεραιότητά τους ορίζεται 2. Στο επόμενο βήμα (d) έχουμε δύο κόμβους ελέγχου 0 και 1. Η επόμενη αναζήτηση (e) πραγματοποιείται και για τους δύο. Στους παραλήπτες υπάρχει μόνο ένας κόμβος με κλειδί 1. Ο παραλήπτης αυτός αποκτά την προτεραιότητα 3. Όλες οι προτεραιότητες έχουν υπολογιστεί και η διαδικασία ολοκληρώνεται (f).

Όταν υπολογιστούν οι προτεραιότητες, ξεκινά η δημιουργία του TDMA προγράμματος. Οι κόμβοι αποστολείς με την μεγαλύτερη προτεραιότητα προωθούν τα δεδομένα τους στην χρονοθυρίδα (timeslot)  $t_i$ . Εκείνοι με την αμέσως μικρότερη προτεραιότητα ξεκινούν την μετάδοσή τους την χρονική στιγμή  $t_{i+1}$ . Οι υπολογισμοί συνεχίζονται έως την προτεραιότητα 1 όπου ο κόμβος αποστολέας προωθεί τελευταίος τα δεδομένα του στον σταθμό βάσης. Κάθε χρονοθυρίδα παρέχει αρκετό χρόνο στους αποστολείς, ώστε να ολοκληρώσουν με επιτυχία την μετάδοσή τους. Η εκτέλεση του προγράμματος συνεχίζεται ακόμη και αν αποτύχει μία μετάδοση.

### 3.6. Συμπεράσματα

Στο κεφάλαιο αυτό περιγράψαμε την λειτουργία ενός πρωτοκόλλου το οποίο μπορεί να χρησιμοποιηθεί για την εξοικονόμηση ενέργειας στα ασύρματα δίκτυα αισθητήρων. Η λειτουργία του βασίζεται στην διαδικασία της συσταδοποίησης, της συλλογής της πληροφορίας και της δρομολόγησης. Εκτός από αυτή της συσταδοποίησης, η συλλογή και η δρομολόγηση της πληροφορίας, είναι διαδικασίες που αλληλοεξαρτώνται. Βασίζόμενοι στις διαδικασίες αυτές, μπορούμε να διακρίνουμε κάποιες ιδιότητες που χαρακτηρίζουν την ΜΕΕΑΔΑ, σε σχέση με τα υπόλοιπα πρωτόκολλα της ίδιας κατηγορίας. Οι ιδιότητες αυτές είναι:

- *Συσταδοποίηση δύο επιπέδων:* Οι αποστάσεις μεταξύ των κόμβων, η πυκνότητα τους, ο αριθμός τους και η θέση στην οποία βρίσκονται αποτελούν τους παράγοντες που καθορίζουν τις συστάδες. Ο κύριος σκοπός της συσταδοποίησης είναι να χωρίζει το δίκτυο σε υποδίκτυα.
- *Προσαρμοστική συλλογή πληροφορίας:* Η συγκέντρωση της πληροφορίας σε μία συστάδα γίνεται με την χρήση τριών κόμβων. Η θέση τους επιλέγεται με τέτοιο τρόπο, έτσι ώστε να μειώνονται οι αποστάσεις μετάδοσης για κάθε κόμβο. Τα δεδομένα ολόκληρης της συστάδας καταλήγουν σε έναν κόμβο με την ονομασία cluster leader.
- *Δρομολόγηση πληροφορίας:* Οι cluster leader συνεργάζονται μεταξύ τους προκειμένου να δημιουργήσουν μία χαμηλού κόστους διαδρομή και να προωθήσουν από κοινού τα δεδομένα του δικτύου στον προορισμό.

Η ΜΕΕΑΔΑ έχει σχεδιαστεί έτσι ώστε να λειτουργεί για οποιοδήποτε αριθμό αισθητήρων και έκταση της περιοχής εφαρμογής, πάντα σε σχέση με την θέση του σταθμού βάσης. Ο σταθμός βάσης μπορεί να τοποθετηθεί οπουδήποτε στο δίκτυο εντός ή εκτός αυτού. Για την επαλήθευση της ορθής λειτουργίας του πρωτοκόλλου, όπως επίσης και για τον έλεγχο της απόδοσής του, είναι σημαντική η υλοποίησή του. Μέσα από τα πειραματικά αποτελέσματα, θα είμαστε σε θέση να το συγκρίνουμε με τα άλλα πρωτόκολλα.

Πανεπιστήμιο Πειραιώς

Η υλοποίηση της μεθόδου εξοικονόμησης ενέργειας στα ασύρματα δίκτυα αισθητήρων μπορεί να μας δώσει μία σαφή εικόνα για τις δυνατότητές της. Η έκδοση που θα παρουσιαστεί σε αυτό το κεφάλαιο υλοποιήθηκε με την χρήση της γλώσσας προγραμματισμού C. Αφού παρουσιάσουμε την λειτουργία του προγράμματος, θα προχωρήσουμε στην σύγκρισή του με το πρωτόκολλο LEACH. Η σύγκριση μεταξύ των δύο θα γίνει χρησιμοποιώντας τις παραμέτρους που θα καθοριστούν στην αρχή του κεφαλαίου. Στόχος του κεφαλαίου αυτού είναι να παρουσιάσει τις δυνατότητες του προτεινόμενου πρωτοκόλλου δρομολόγησης σε ό,τι αφορά την διάρκεια ζωής και την μέση κατανάλωση ενέργειας του δικτύου.

#### 4.1. Υλοποίηση ΜΕΕΑΔΑ

Βασιζόμενοι στο Κεφάλαιο 3 και την θεωρητική παρουσίαση της λειτουργίας της ΜΕΕΑΔΑ, προχωράμε στο επόμενο βήμα, αυτό της υλοποίησης. Η υλοποίηση πραγματοποιήθηκε στην γλώσσα C, λαμβάνοντας υπόψιν τους αλγόριθμους 1, 2, 3. Το πρόγραμμα δέχεται ως είσοδο ένα σύνολο αισθητήρων  $C'$  και τα όρια της περιοχής όπου αυτοί περιέχονται. Η θέση των αισθητήρων στο δίκτυο είναι τυχαία και η πυκνότητά τους μπορεί επίσης να καθοριστεί. Στην έξοδό του παράγει το τελικό δίκτυο, έτσι όπως υπολογίστηκε κατά την διαδικασία της συσταδοποίησης. Ο σταθμός βάσης τοποθετείται σε σταθερή θέση και μακριά από το δίκτυο, χωρίς αυτό να σημαίνει ότι δεν είναι δυνατή η τοποθέτησή του ακόμα και εντός δικτύου.

Από εκεί και πέρα, η λειτουργία του περιλαμβάνει την έναρξη της διαδικασίας της συλλογής της πληροφορίας και της δρομολόγησης της στον σταθμό βάσης. Ο χρήστης μπορεί να επιβλέπει το σχέδιο συλλογής και δρομολόγησης του δικτύου ανά round. Επίσης, είναι δυνατή η παρακολούθηση της κατανάλωσης ενέργειας σε κάθε κόμβο. Η κατανάλωση της ενέργειας αφορά την λήψη και την αποστολή δεδομένων και υπολογίζεται σύμφωνα με το μοντέλο ενέργειας που χρησιμοποιείται. Επιπλέον, για τους κόμβους cluster head και cluster leader, προστίθεται ένα επιπλέον κόστος κατανάλωσης ener-

#### 4. Έλεγχος Επίδοσης

γείας, που αφορά στην διαδικασία της συγχώνευσης των δεδομένων (data aggregation, data compression).

Αρχικά, όλοι οι κόμβοι του δικτύου διαθέτουν την ίδια ενέργεια. Η ενέργειά τους ενημερώνεται όταν εκτελούν κάποια λειτουργία. Λόγω του ότι διαθέτουν πεπερασμένα ποσά ενέργειας, μετά από κάποιο χρονικό διάστημα είναι φυσικό να μην μπορούν να ανταποκριθούν στις μετέπειτα ενεργειακές απαιτήσεις. Οι κόμβοι αυτοί αφαιρούνται και το δίκτυο συνεχίζει με τους υπόλοιπους τις διαδικασίες της συλλογής και της δρομολόγησης της πληροφορίας. Θεωρούμε ότι η λειτουργία του δικτύου έχει τερματιστεί, όταν όλοι οι κόμβοι που το αποτελούν, δεν διαθέτουν ενεργειακά αποθέματα. Καθ' όλη την διάρκεια της προσομοίωσης, μας ενδιαφέρει ο χρόνος  $t$  (round) στον οποίο κάθε κόμβος έχασε όλη την ενέργειά του. Επίσης, ως προς την συνολική απόδοση του πρωτοκόλλου, μας ενδιαφέρει και η μέση κατανάλωση ενέργειας ανά round.

##### 4.1.1. Παράμετροι Δικτύου

Οι παράμετροι που θα χρησιμοποιήσουμε κατά την διάρκεια του ελέγχου της επίδοσης της ΜΕΕΑΔΑ είναι όμοιες με αυτές που παρουσιάστηκαν στο Κεφάλαιο 2.1. Το μέγεθος των πακέτων που ανταλλάσσουν οι κόμβοι ορίζεται ως 2000-bit και η θέση του σταθμού βάσης θα είναι ανάλογη με το σενάριο της εκάστοτε προσομοίωσης. Σε κάθε περίπτωση, η θέση του θα θεωρείται σταθερή και η ελάχιστη απόστασή του από το δίκτυο τα 100m. Από το μέγεθος του πακέτου, υπολογίζουμε επίσης και το κόστος της συγχώνευσης των δεδομένων, που επιβαρύνει κάθε cluster head ή cluster leader.

Λειτουργία	Κατανάλωση Ενέργειας
Κύκλωμα Μετάδοσης ( $E_{elec}$ )	50nJ/bit
Κύκλωμα Λήψης ( $E_{elec}$ )	
Ενισχυτής Μετάδοσης ( $E_{amp}$ )	100pJ/bit/m <sup>2</sup>
Συγχώνευση Δεδομένων	5nJ/bit/message

Πίνακας 4.1.: Παράμετροι κατανάλωσης ενέργειας.

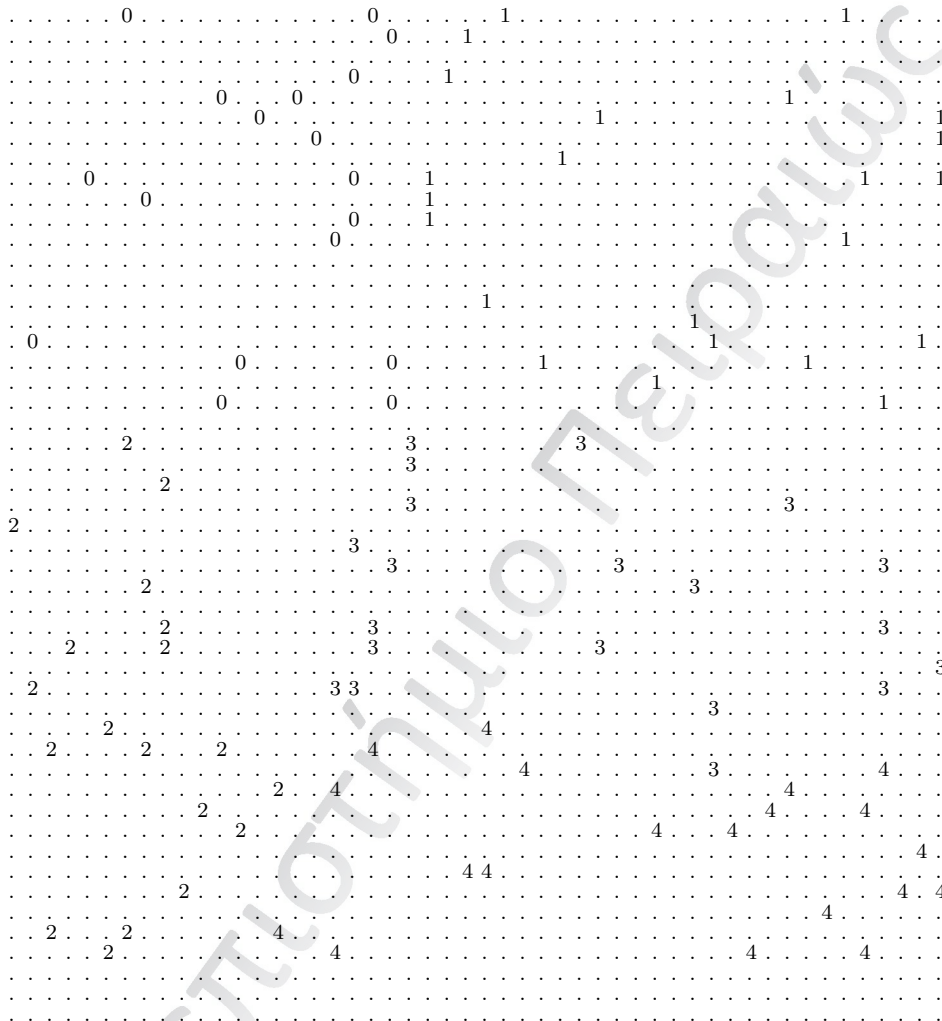
Χρησιμοποιώντας τις παραμέτρους του Πίνακα 4.1, θα ακολουθήσουν παρομοιώσεις για διάφορα μεγέθη δικτύων ξεκινώντας από μικρά ( $50m \times 50m$ ) και καταλήγοντας σε μεγαλύτερα ( $200m \times 200m$ ). Η πυκνότητα των κόμβων θα είναι επίσης διαφορετική για κάθε σενάριο και θα εναλλάσσεται από χαμηλή σε υψηλή (100-500 κόμβους). Η ελάχιστη απόσταση μεταξύ των κόμβων θα είναι το 1m. Θεωρείται ότι όλοι οι κόμβοι διαθέτουν τις ίδιες δυνατότητες, όπως επεξεργαστική ισχύ, μνήμη (βλ. Κεφάλαιο 3). Σε ότι αφορά το περιβάλλον εφαρμογής, δεν λαμβάνονται υπόψιν φαινόμενα (θορυβώδεις περιβάλλον) που μπορούν να επηρεάσουν την διαδικασία της ασύρματης μετάδοσης.

##### 4.1.2. Λειτουργία Προσομοίωσης ΜΕΕΑΔΑ

Για την παρουσίαση του προγράμματος που υλοποιεί την ΜΕΕΑΔΑ, θα χρησιμοποιήσουμε ένα δίκτυο διαστάσεων  $50m \times 50m$ . Στο δίκτυο αυτό τοποθετούνται τυχαία, ένα



σύνολο 100 κόμβων. Όλοι διαθέτουν αρχικά την ίδια ενέργεια 0.25J και η πυκνότητά τους είναι ίδια για όλα τα σημεία του δικτύου. Ο σταθμός βάσης ορίζεται στην θέση [25]–[100]. Αυτό σημαίνει ότι η ελάχιστη απόστασή του από τον κοντινότερο κόμβο είναι τουλάχιστον 100m. Εκτός από τις παραμέτρους αυτές, ορίζουμε επίσης και ένα όριο (threshold) 5m, απαραίτητο για την λειτουργία του πρώτου επιπέδου της συσταδοποίησης (βλ. Αλγόριθμος 1).



Σχήμα 4.1.: Πίνακας με τους κωδικούς (κλειδιά) 100 κόμβων σε δίκτυο διαστάσεων  $50m \times 50m$ , μετά την εκτέλεση της συσταδοποίησης.

Στο Σχήμα 4.1 αναπαρίστανται η έξοδος των αλγορίθμων συσταδοποίησης. Οι υπολογισμοί του προγράμματος γίνονται με την χρήση ενός πίνακα δύο διαστάσεων. Από τους τυχαία τοποθετημένους κόμβους δημιουργείται ένας αρχικός πίνακας όπου όλοι οι κόμβοι διαθέτουν κωδικό συστάδας 0. Με το σύμβολο “.” αναπαριστούμε τον κενό χώρο

#### 4. Έλεγχος Επίδοσης

της περιοχής του δικτύου. Ξεκινώντας τους υπολογισμούς, το πρόγραμμα ακολουθεί τον αλγόριθμο 1 και δημιουργεί ένα πίνακα με τους αρχικούς κωδικούς συστάδων. Στο επόμενο και τελικό βήμα ακολουθείται ο αλγόριθμος 2 και παράγεται το αποτέλεσμα του σχήματος. Οι κόμβοι με ίδιο κωδικό ανήκουν στην ίδια συστάδα. Οι υπολογισμοί που αφορούν στην συλλογή και στην δρομολόγηση της πληροφορίας θα γίνονται με βάση αυτές τις συστάδες.

Η λειτουργία της ΜΕΕΑΔΑ ξεκινά αφού δημιουργηθούν οι συστάδες. Πριν περιγράψουμε τα επόμενα βήματα του προγράμματος είναι σημαντικό να σημειώσουμε κάποια από τα χαρακτηριστικά της συσταδοποίησης. Όπως παρατηρούμε και στο Σχήμα 4.1, κάθε κόμβος διαθέτει ένα κωδικό συστάδας. Το εύρος των κωδικών είναι από 0 ως 4 γεγονός που υποδηλώνει το σύνολο των διαφορετικών συστάδων. Το πλήθος των διαφορετικών κωδικών, προκύπτει από το  $\tilde{N} = 5\%$  των 100 κόμβων. Οι κόμβοι χωρίζονται ανάλογα με τον αναμενόμενο αριθμό κόμβων ανά συστάδα και την πυκνότητα τους στο δίκτυο. Σύμφωνα με τα στοιχεία του παρακάτω πίνακα, οι κόμβοι του δικτύου έχουν χωριστεί σχεδόν ισοδύναμα σε συστάδες.

Κωδικός Συστάδας	Αριθμός Κόμβων
0	18
1	23
2	19
3	20
4	20

Πίνακας 4.2.: Αριθμός κόμβων ανά συστάδα για το δίκτυο του Σχήματος 4.1.

Μετά την συσταδοποίηση, έχουμε την συλλογή και την δρομολόγηση της πληροφορίας. Για την αναζήτηση των κόμβων και την εκτέλεση των παραπάνω διαδικασιών θα χρησιμοποιήσουμε δύο σημεία ελέγχου  $i, j$ . Η θέση ενός κόμβου χαρακτηρίζεται από την γραμμή και την στήλη στην οποία βρίσκεται. Οι γραμμές του πίνακα θα αναπαριστούνται με τον χαρακτήρα  $i$  και οι στήλες με τον χαρακτήρα  $j$ . Αν έχουμε ένα πίνακα διαστάσεων  $100m \times 100m$ , τότε οι δυνατές τιμές των σημείων  $i, j$  είναι από 1 ως 100 και η θέση ενός κόμβου ίση με  $[i][j]$ . Η αρίθμηση ξεκινά από την πάνω αριστερή γωνία του πίνακα (βλ. Σχήμα 4.1) και καταλήγει στην κάτω δεξιά. Ο κόμβος που βρίσκεται στην πάνω αριστερή γωνία καταλαμβάνει την θέση  $[1][1]$  ενώ εκείνος που βρίσκεται στην κάτω δεξιά την θέση  $[100][100]$ .

Η διαδικασία της συλλογής και της δρομολόγησης προϋποθέτει τον υπολογισμό των τριών cluster head. Ένας από αυτούς θα αναλάβει την θέση του cluster leader. Έπειτα, κάθε κόμβος θα επιλέξει τον κοντινότερο σε αυτόν για να προωθήσει τα δεδομένα που συνέλεξε. Η πληροφορία θα σταλεί από τους cluster head στον cluster leader και από εκεί στον σταθμό βάσης, με απευθείας μετάδοση ή μέσω κάποιας άλλης συστάδας. Το πρόγραμμα έχει την ικανότητα να μας προβάλλει, μέσω μίας λίστας, όλη αυτή την πληροφορία για κάθε κόμβο του δικτύου και σε κάθε round. Επιπλέον, μπορούμε να ελέγξουμε και το κόστος σε ενέργεια που επωμίζεται κάθε κόμβος για κάθε αποστολή

και λήψη δεδομένων. Ένα παράδειγμα για τον τρόπο με τον οποίο το πρόγραμμα μάς ενημερώνει για τις διαδικασίες συλλογής και δρομολόγησης ανά round περιγράφεται στο Σχήμα 4.2.

primary head [22][7]	
secondary head [47][6]	
in turf head [34][2]	
	B[24][9] : euclidean = 2.828427 transmission energy = 0.00010160 receive energy = 0.00010000
	B[26][1] : euclidean = 7.211103 transmission energy = 0.00011040 receive energy = 0.00010000
	B[29][8] : euclidean = 7.071068 transmission energy = 0.00011000 receive energy = 0.00010000
inturfnode	B[31][9] : euclidean = 7.615773 transmission energy = 0.00011160 receive energy = 0.00010000
inturfnode	B[32][4] : euclidean = 2.828427 transmission energy = 0.00010160 receive energy = 0.00010000
inturfnode	B[32][9] : euclidean = 7.280110 transmission energy = 0.00011060 receive energy = 0.00010000
inturfnode	B[36][6] : euclidean = 4.472136 transmission energy = 0.00010400 receive energy = 0.00010000
inturfnode	B[37][3] : euclidean = 3.162278 transmission energy = 0.00010200 receive energy = 0.00010000
inturfnode	B[37][8] : euclidean = 6.708204 transmission energy = 0.00010900 receive energy = 0.00010000
inturfnode	B[37][12] : euclidean = 10.440307 transmission energy = 0.00012180 receive energy = 0.00010000
secondary	B[39][15] : euclidean = 12.041595 transmission energy = 0.00012900 receive energy = 0.00010000
secondary	B[40][11] : euclidean = 8.602325 transmission energy = 0.00011480 receive energy = 0.00010000
secondary	B[41][13] : euclidean = 9.219544 transmission energy = 0.00011700 receive energy = 0.00010000
secondary	B[44][10] : euclidean = 5.000000 transmission energy = 0.00010500 receive energy = 0.00010000
secondary	B[46][3] : euclidean = 3.162278 transmission energy = 0.00010200 receive energy = 0.00010000
secondary	B[46][7] : euclidean = 1.414214 transmission energy = 0.00010040 receive energy = 0.00010000
head[47][6] to head[34][2]	: euclidean=13.601471 transmission energy = 0.00014700 receive energy = 0.00010000
head[34][2] to head[22][7]	: euclidean=13.000000 transmission energy = 0.00014380 receive energy = 0.00010000
B[22][7] to B[25][-100]	: min euclidean=107.042048 transmission energy = 0.00240160 receive energy = 0.00010000

Σχήμα 4.2.: Σχέδιο δρομολόγησης για την συστάδα με κωδικό 2 του Σχήματος 4.1.

Στο Σχήμα 4.2 παρουσιάζονται οι πράξεις που εκτελεί το πρόγραμμα και η έξοδος

#### 4. Έλεγχος Επίδοσης

του προς τον χρήστη. Για λόγους χώρου, προβάλλεται μόνο το σχέδιο που αφορά την συστάδα με κωδικό 2. Αρχικά μπορούμε να παρατηρήσουμε την θέση των τριών cluster head. Πιο συγκεκριμένα, εκλέγονται οι κόμβοι *primary head* (Primary), *secondary head* (Secondary) και *in turf head* (Territorial). Στην συνέχεια, καθένας από τους κόμβους του πίνακα  $B[i][j]$  επιλέγει τον κοντινότερο σε αυτόν κόμβο. Η επιλογή φαίνεται στα αριστερά της θέσης κάθε κόμβου, με το κενό να συμβολίζει τον Primary και το *secondary* και *inturfnode* τους Secondary και Territorial αντίστοιχα. Για κάθε κόμβο (που δεν είναι cluster head), μπορούμε να δούμε πληροφορίες όπως η απόσταση από τον κόμβο επιλογής, καθώς και το κόστος της ενέργειας μετάδοσης και λήψης που απαιτείται από τους δύο κόμβους.

Όταν η πληροφορία συγκεντρωθεί στους cluster head εκείνοι την προωθούν στον cluster leader. Ο τρόπος με τον οποίο επιτυγχάνεται αυτό παρουσιάζεται στις τελευταίες γραμμές του σχήματος (Σχήμα 4.2). Στο επόμενο βήμα ο cluster leader επιλέγει τον προορισμό που θα μεταδώσει τα δεδομένα της συστάδας. Στο παράδειγμά μας, ο cluster leader επιλέγει τον σταθμό βάσης. Το κόστος μετάδοσης είναι σημαντικά υψηλότερο λόγω της μεγαλύτερης απόστασης. Επιπλέον, υπολογίζεται και το κόστος λήψης του σταθμού βάσης, χωρίς όμως να έχει ιδιαίτερη σημασία, καθώς υποθέτουμε ότι διαθέτει απεριόριστη ενέργεια. Το πρόγραμμα μας ενημερώνει για τις επιλογές μετάδοσης κάθε cluster leader ανά round μέσω μίας ουράς, αλλά και μέσω της λίστας κάθε συστάδας.

Συστάδα Αποστολέας	Συστάδα Προορισμός
0	2
1	0
2	BS
3	2
4	3

Πίνακας 4.3.: Επιλογές κόμβων προορισμών από τους cluster leader ανά round.

Οι cluster leader θα πρέπει να είναι προσεκτικοί κατά την μετάδοση προς τον σταθμό βάσης. Όπως αναφέρθηκε και στο Κεφάλαιο 3, θα πρέπει να υπάρξει συνεργασία μεταξύ των κόμβων, μέσω ενός TDMA προγράμματος, ώστε να ολοκληρωθεί η διαδικασία χωρίς να χαθούν δεδομένα. Το βήμα της μετάδοσης των cluster leader είναι και το τελευταίο σε ότι αφορά την ολοκλήρωση ενός round. Το επόμενο round, θα ξεκινήσει με τους κόμβους να διαθέτουν νέα αποθέματα ενέργειας. Αν η ενέργεια ενός κόμβου είναι μικρότερη ή ίση από το 0 τότε αφαιρείται από τον πίνακα του δικτύου και δεν υπολογίζεται στις λίστες των συστάδων. Η λειτουργία του προγράμματος τερματίζει όταν όλοι οι κόμβοι χάνουν την ενέργειά τους.

#### 4.2. Διάρκεια Ζωής του Δικτύου

Ένα από τα σημαντικότερα ζητήματα λειτουργίας ενός πρωτοκόλλου είναι η απόδοσή του σε σχέση με τα αντίστοιχα πρωτόκολλα της ίδιας κατηγορίας. Έτσι λοιπόν, θα

συγκρίνουμε την ΜΕΕΑΔΑ με το πρωτόκολλο LEACH. Η σύγκριση θα γίνει με βάση τις παραμέτρους του Πίνακα 4.1. Οι ίδιες παράμετροι θα χρησιμοποιηθούν και για την θέση του σταθμού βάσης και το μέγεθος πακέτου. Μας ενδιαφέρει ο χρόνος στον οποίο χάνουν την ενέργειά τους ο πρώτος και ο τελευταίος κόμβος του δικτύου. Τα δίκτυα που θα δημιουργηθούν για τις ανάγκες των προσομοιώσεων, θα περιλαμβάνουν 100 κόμβους με διάφορα αρχικά ποσά ενέργειας (0.25J, 0.5J, 1J). Το μέγεθός τους θα είναι  $50m \times 50m$  και  $100m \times 100m$ .

	Ενέργεια (J/node)	Πρωτόκολλο	Πρώτος κόμβος εκτός λειτουργίας (Round)	Τελευταίος κόμβος εκτός λειτουργίας (Round)
50m × 50m	0.25	LEACH	409	639
		ΜΕΕΑΔΑ	803	1179
	0.5	LEACH	865	1252
		ΜΕΕΑΔΑ	1623	2350
	1	LEACH	1752	2508
		ΜΕΕΑΔΑ	3245	4678
100m × 100m	0.25	LEACH	167	317
		ΜΕΕΑΔΑ	382	773
	0.5	LEACH	329	562
		ΜΕΕΑΔΑ	773	1516
	1	LEACH	668	1059
		ΜΕΕΑΔΑ	1547	3011

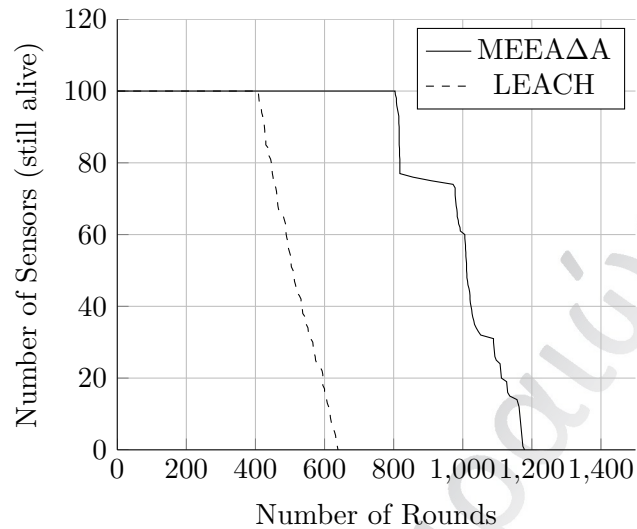
Πίνακας 4.4.: Αποτελέσματα προσομοιώσεων των πρωτοκόλλων δρομολόγησης ΜΕΕΑΔΑ και LEACH.

Τα αποτελέσματα των προσομοιώσεων (Πίνακας 4.4) δείχνουν ότι η ΜΕΕΑΔΑ υπερτερεί του πρωτοκόλλου LEACH. Ο σταθμός βάσης τοποθετείται στην θέση [25][150] και [50][200] για τα δίκτυα μεγέθους  $50m \times 50m$  και  $100m \times 100m$  αντίστοιχα. Στην πρώτη περίπτωση, η απόσταση του από τον κοντινότερο κόμβο είναι τα 100 μέτρα, ενώ στην δεύτερη τα 200 μέτρα. Στα περισσότερα αποτελέσματα, η ΜΕΕΑΔΑ τείνει να διπλασιάσει την διάρκεια ζωής των κόμβων του δικτύου, σε ό,τι αφορά τον χρόνο όπου ο πρώτος και ο τελευταίος κόμβος χάνουν την ενέργεια τους. Οι διαφορές στην απόδοση μεταξύ των δύο πρωτοκόλλων γίνονται πιο αισθητές όσο το μέγεθος του δικτύου αυξάνεται και η πυκνότητα των κόμβων μειώνεται.

Χρησιμοποιώντας την ΜΕΕΑΔΑ, ο πρώτος κόμβος του δικτύου χάνει την ενέργεια του πολύ μετά από τον τελευταίο του πρωτοκόλλου LEACH. Αυτό συμβαίνει διότι στην ΜΕΕΑΔΑ οι αποστάσεις μετάδοσης των κόμβων σε κάθε συστάδα μειώνονται σημαντικά ανά round χάρη στους τρεις cluster head. Ακόμα, μειώνονται και οι αποστάσεις μετάδοσης από τους cluster head προς τον σταθμό βάσης, καθώς χρησιμοποιείται ο αλγόριθμος δρομολόγησης μη αρνητικής απόστασης. Αντίθετα, στο πρωτόκολλο LEACH κάθε cluster head επικοινωνεί απευθείας με τον σταθμό βάσης. Οι μικρότερες αποστάσεις μετάδοσης μπορεί να οδηγήσουν και στην μείωση των παρεμβολών που προκα-

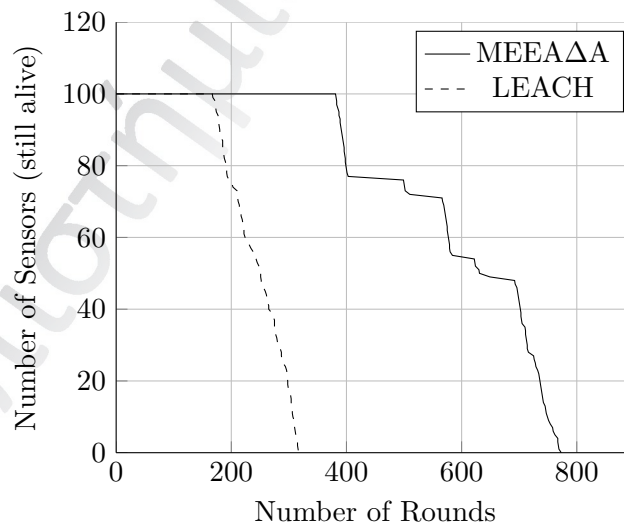
#### 4. Έλεγχος Επίδοσης

λούνται από τους ίδιους τους κόμβους του δικτύου.



Σχήμα 4.3.: Διάρκεια ζωής του δικτύου μεγέθους  $50m \times 50m$  και  $0.25J/\text{κόμβο}$ .

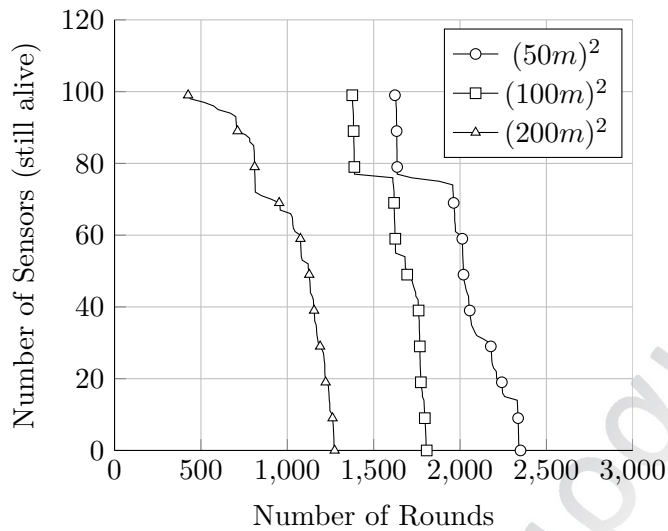
Στο διάγραμμα του Σχήματος 4.3, παρατηρούμε τον χρόνο όπου καθένας από τους 100 κόμβους του δικτύου, χάνει την ενέργειά του. Το διάγραμμα δημιουργήθηκε με τις ίδιες παραμέτρους που χρησιμοποιήθηκαν και για τα αποτελέσματα που παρουσιάστηκαν στον Πίνακα 4.4.



Σχήμα 4.4.: Διάρκεια ζωής του δικτύου μεγέθους  $100m \times 100m$  και  $0.25J/\text{κόμβο}$ .

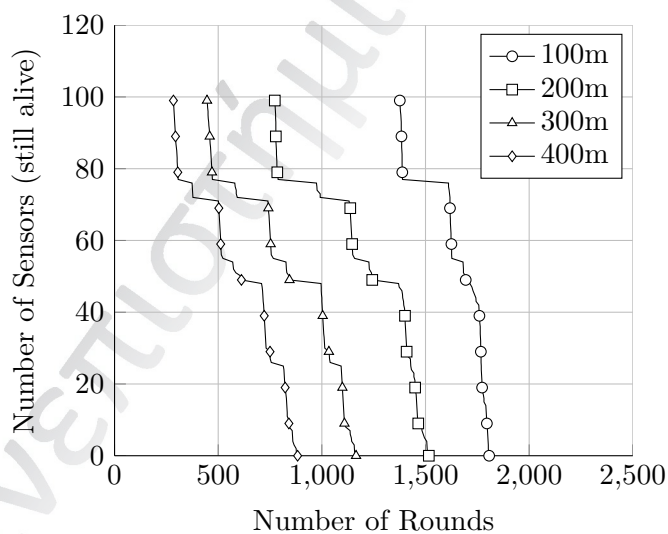
Στο επόμενο διάγραμμα (Σχήμα 4.4) παρατηρούμε την διάρκεια ζωής των κόμβων, για ένα δίκτυο μεγέθους  $100m \times 100m$ . Όπως και στο προηγούμενο παράδειγμα, οι

παράμετροι είναι ίδιες με αυτές του προηγούμενου πίνακα (Πίνακας 4.4).



Σχήμα 4.5.: Απόδοση MEEAΔA για διαφορετικά μεγέθη δικτύου και 0.5J/κόμβο.

Για όλα τα αποτελέσματα του διαγράμματος (Σχήμα 4.5), ο σταθμός βάσης απέχει τουλάχιστον 100m από τον κοντινότερο κόμβο. Τοποθετείται στην θέση [25][150] για τα δίκτυα 50m × 50m, στην [50][−100] για τα δίκτυα 100m × 100m και στην [100][−100] για αυτά μεγέθους 200m × 200m.

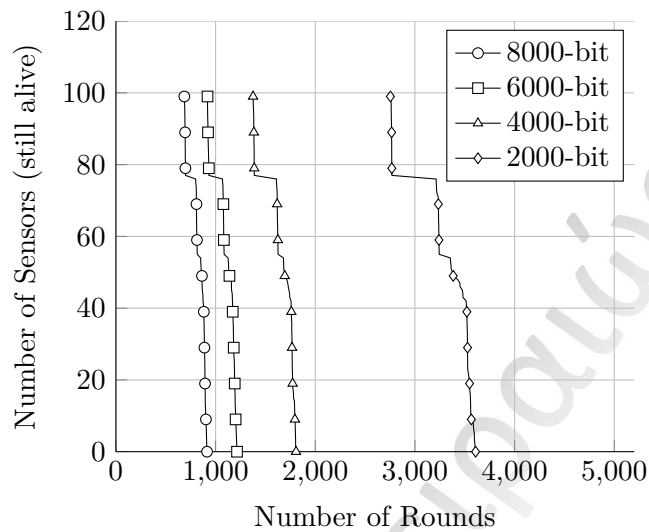


Σχήμα 4.6.: Η απόδοση της MEEAΔA για διαφορετική θέση του σταθμού βάσης και 0.5J/κόμβο.

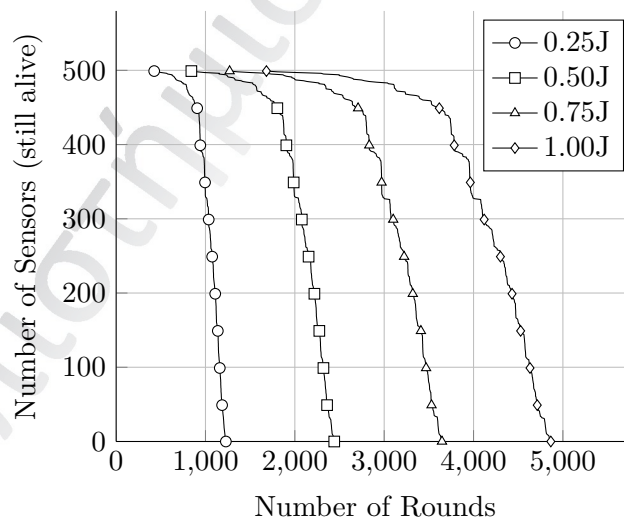
Συνεχίζουμε τον έλεγχο της απόδοσης (Σχήμα 4.6) τοποθετώντας τον σταθμό βάσης

#### 4. Έλεγχος Επίδοσης

σε διαφορετικές αποστάσεις από τον κοντινότερο κόμβο. Οι διαστάσεις του δικτύου είναι  $100m \times 100m$  και η αρχική ενέργεια των κόμβων  $0.5J$ .



Σχήμα 4.7.: Αποτελέσματα προσομοιώσεων για διαφορετικό μέγεθος πακέτου, σε δίκτυο διαστάσεων  $100m \times 100m$  με 100 κόμβους και αρχική ενέργεια  $1J/\text{κόμβο}$ . Ο σταθμός βάσης τοποθετείται στην θέση  $[50][100]$ .



Σχήμα 4.8.: Απόδοση MEEAΔA για διαφορετική αρχική ενέργεια 500 κόμβων, σε ένα δίκτυο διαστάσεων  $100m \times 100m$ . Ο σταθμός βάσης τοποθετείται στην θέση  $[50][100]$ .

Στο διάγραμμα του Σχήματος 4.8 αυξάνουμε την πυκνότητα των κόμβων από τους



100 των προηγούμενων σχημάτων (βλ. Σχήμα 4.6, απόσταση σταθμού βάσης 100m) σε 500. Από τα αποτελέσματα, μπορούμε να διαπιστώσουμε ότι ο μεγαλύτερος αριθμός κόμβων, οδηγεί στην αύξηση της συνολικής διάρκειας ζωής του δικτύου. Βέβαια, αυτό ισχύει από ένα σημείο και μετά, καθώς οι πρώτοι κόμβοι (βλ. Σχήμα 4.8 με 0.5J) χάνουν την ενέργειά τους πιο γρήγορα. Επίσης, θα πρέπει να επισημάνουμε ότι οι περισσότεροι κόμβοι οδηγούν σε περισσότερες συστάδες. Ως επακόλουθο, αυξάνονται οι καθυστερήσεις συλλογής και δρομολόγησης των δεδομένων του δικτύου.

Παρατηρήσαμε την απόδοση της ΜΕΕΑΔΑ για διάφορες παραμέτρους που επηρεάζουν την διάρκεια ζωής των κόμβων ενός δικτύου ασύρματων αισθητήρων. Για όλες τις περιπτώσεις, η υπεροχή του προτεινόμενου πρωτοκόλλου σε σχέση με το αντίστοιχο LEACH είναι εμφανής. Από εκεί και πέρα, το μέγεθος του δικτύου, η απόσταση του σταθμού βάσης, το μέγεθος του πακέτου και η αρχική ενέργεια των κόμβων είναι παράγοντες που επηρεάζουν την διάρκεια ζωής των κόμβων. Ξεκινώντας από το μέγεθος του δικτύου, όσο αυτό αυξάνεται, διαπιστώνουμε περισσότερη κατανάλωση ενέργειας από τους κόμβους. Αυτό είναι απόλυτα λογικό, καθώς ο ίδιος αριθμός κόμβων καλύπτει μεγαλύτερη περιοχή με αποτέλεσμα να αυξάνονται οι μεταξύ τους αποστάσεις.

Ανάλογη μείωση στην διάρκεια ζωής του δικτύου παρατηρείται και από την απόσταση που τοποθετείται ο σταθμός βάσης. Αν και σε κάθε round μόνο ένας κόμβος προωθεί όλη την πληροφορία του δικτύου, μακροπρόθεσμα επηρεάζεται η κατανάλωση όλων. Χρησιμοποιώντας τον αλγόριθμο δρομολόγησης μη αρνητικής απόστασης, όλοι οι κόμβοι του δικτύου θα αναλάβουν το βάρος της μετάδοσης των δεδομένων. Αυτοί που βρίσκονται στην κοντινότερη απόσταση από τον σταθμό βάσης, θα έχουν την προτεραιότητα σε σχέση με τους πιο απομακρυσμένους.

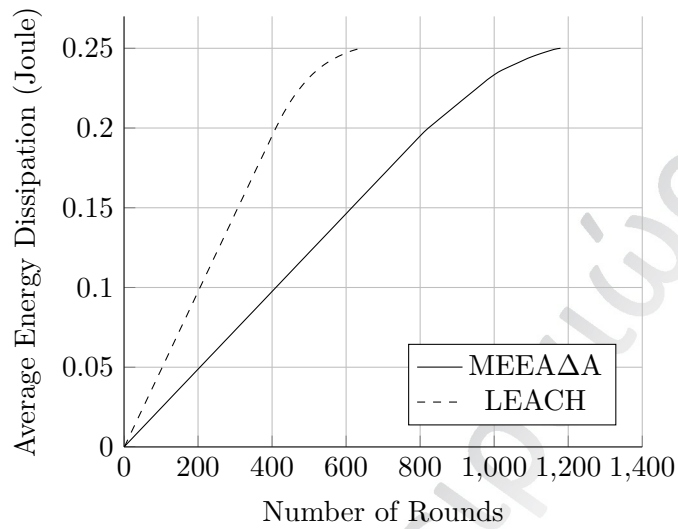
Το μέγεθος του πακέτου είναι ακόμη ένας βασικός παράγοντας που καθορίζει την διάρκεια ζωής των κόμβων. Όπως δείξαμε και στο Κεφάλαιο 2.1 (Σχέση (2.1), (2.2)), το μοντέλο ενέργειας που χρησιμοποιούμε βασίζεται το μέγεθος του πακέτου και την απόσταση μετάδοσης. Κατά συνέπεια, επηρεάζεται άμεσα η κατανάλωση των κυκλωμάτων μετάδοσης, λήψης καθώς και το κόστος συγχώνευσης των πακέτων. Όπως παρατηρούμε και από το διάγραμμα που αφορά το μέγεθος του πακέτου (Σχήμα 4.7) ο διπλασιασμός του μεγέθους του μπορεί να επιφέρει σχεδόν διπλάσια κατανάλωση ενέργειας. Για τον λόγο αυτό, άλλωστε, γίνονται προσπάθειες για την ανεύρεση τεχνικών που να μειώνουν αυτό το μέγεθος, χωρίς να χάνεται η ποιότητα των δεδομένων του.

### 4.3. Μέση Κατανάλωση Ενέργειας

Στο επόμενο βήμα της αξιολόγησης των επιδόσεων του πρωτοκόλλου ΜΕΕΑΔΑ, ακολουθεί ο έλεγχος της μέσης κατανάλωσης ενέργειας του δικτύου. Κάθε πρωτόκολλο που δημιουργείται για το πρόβλημα της διαχείρισης ενέργειας στα WSNs στοχεύει στην διαχείριση της ενέργειας κάθε αισθητήριου κόμβου. Ακολουθώντας μία τέτοια στρατηγική μπορούμε να αυξήσουμε την διάρκεια ζωής του δικτύου. Για να μετρήσουμε το πόσο αποτελεσματική είναι αυτή η διαχείριση, ελέγχουμε την μέση ενέργεια που καταναλώνεται από όλους τους κόμβους ανά round. Η ΜΕΕΑΔΑ έχει ως προτεραιότητα την ελάχιστη δυνατή κατανάλωση ανά round. Βέβαια, δεν σημαίνει ότι η επιλογή αυτή δεν

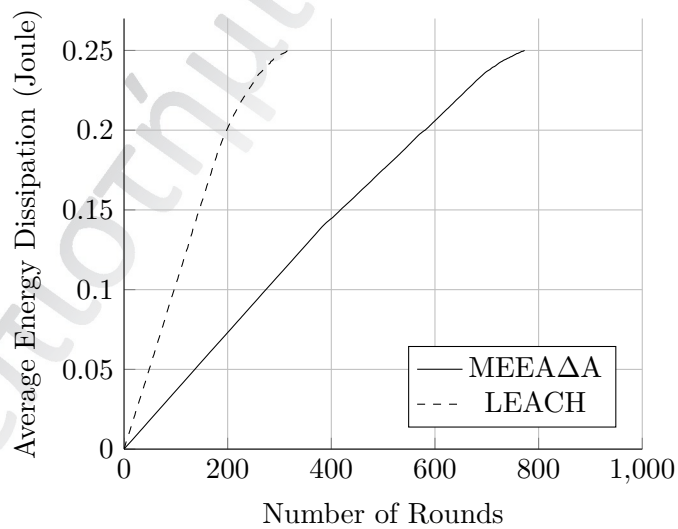
#### 4. Έλεγχος Επίδοσης

έχει μειονεκτήματα, που μπορούν να λειτουργήσουν αρνητικά στον τρόπο με τον οποίο οι κόμβοι χάνουν την ενέργειά τους.



Σχήμα 4.9.: Μέση κατανάλωση ενέργειας για δίκτυα μεγέθους  $50m \times 50m$ , 100 κόμβους και  $0.25J/\text{κόμβο}$ . Ο σταθμός βάσης τοποθετείται στην θέση [25][150].

Οι παράμετροι που χρησιμοποιήθηκαν για τα αποτελέσματα του σχήματος (Σχήμα 4.9), είναι ίδιες με αυτές του Σχήματος 4.3.

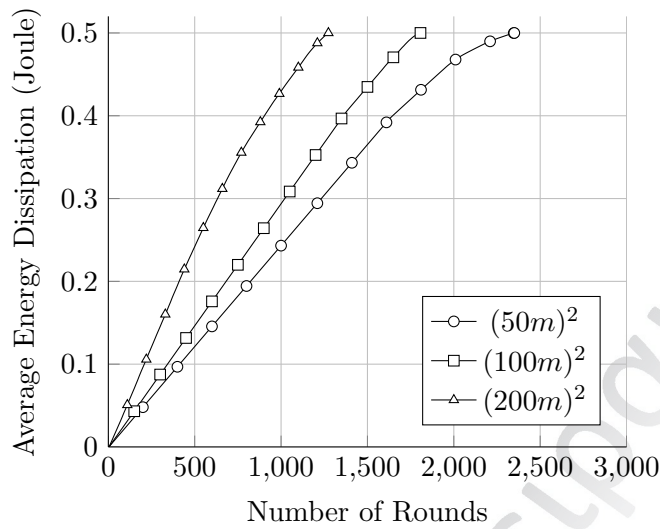


Σχήμα 4.10.: Μέση κατανάλωση ενέργειας για δίκτυα μεγέθους  $100m \times 100m$ , 100 κόμβους και  $0.25J/\text{κόμβο}$ .

Συνεχίζουμε τον έλεγχο της μέσης κατανάλωσης της ενέργειας (Σχήμα 4.10) για

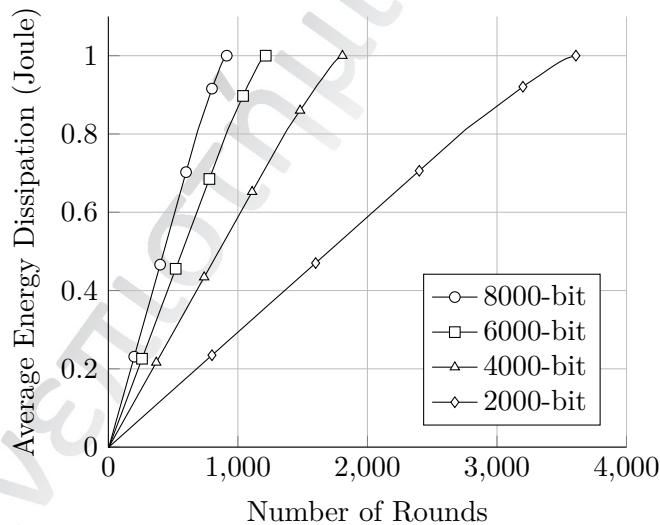
### 4.3. Μέση Κατανάλωση Ενέργειας

παραμέτρους όμοιες με αυτές του Σχήματος 4.4. Ο σταθμός βάσης τοποθετείται στην θέση [50] [-200].



Σχήμα 4.11.: Μέση κατανάλωση ενέργειας για διαφορετικό μέγεθος δικτύου, 100 κόμβους και αρχική ενέργεια 0.5J/κόμβο.

Στο Σχήμα 4.11 χρησιμοποιήθηκαν οι ίδιες παράμετροι με αυτές του Σχήματος 4.5. Ο σταθμός βάσης απέχει τουλάχιστον 100m από τον κοντινότερο κόμβο.

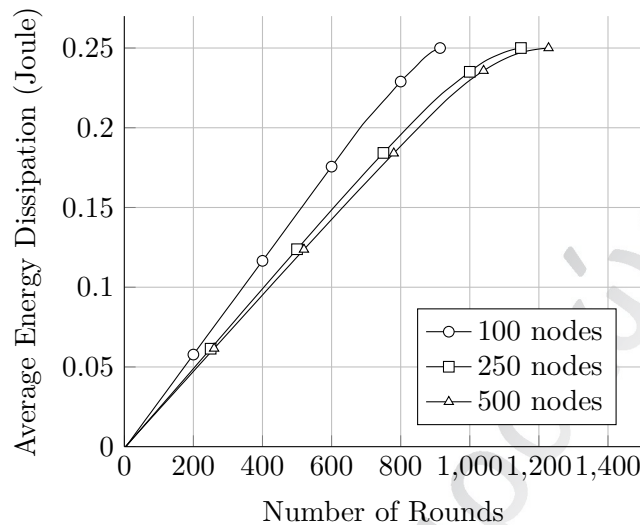


Σχήμα 4.12.: Μέση κατανάλωση ενέργειας για διαφορετικό μέγεθος πακέτου, 1J/κόμβο, σε δίκτυο διαστάσεων 100m × 100m.

Οι παράμετροι του σχήματος (Σχήμα 4.12) είναι όμοιες με αυτές που χρησιμοποιήθη-

#### 4. Έλεγχος Επίδοσης

καν και στο Σχήμα 4.7. Ο σταθμός βάσης τοποθετείται στην θέση [50][-100].



Σχήμα 4.13.: Μέση κατανάλωση ενέργειας για διαφορετικό αριθμό κόμβων με 0.25J/κόμβο. Η περιοχή εφαρμογής έχει διαστάσεις 100m × 100m.

Τα αποτελέσματα του σχήματος (Σχήμα 4.13) δημιουργήθηκαν με τον σταθμό βάσης να τοποθετείται στην θέση [50][-100]. Όπως παρατηρούμε και στο σχήμα, μπορούμε να μειώσουμε την μέση κατανάλωση ενέργειας αν αυξήσουμε τον αριθμό των κόμβων. Ωστόσο, κάτι τέτοιο δεν μπορεί να ισχύει πάντοτε. Από ένα σημείο και μετά, υπάρχει μία ισορροπία ανάμεσα στις αποστάσεις των κόμβων και στο μέγεθος των δεδομένων που διαχειρίζεται το δίκτυο ανά round.

Τα αποτελέσματα της μέσης κατανάλωσης ενέργειας δείχνουν αυτό που παρατηρήθηκε και κατά τον έλεγχο της διάρκειας ζωής του δικτύου. Η MEEAΔA παρουσιάζει υψηλότερες επιδόσεις από το αντίστοιχο πρωτόκολλο LEACH. Αυτό που πρέπει να σημειωθεί είναι ότι περισσότερη ενέργεια καταναλώνεται για μεγαλύτερες αποστάσεις μετάδοσης και μεγαλύτερο αριθμό κόμβων. Οποιοσδήποτε συνδυασμός μεταξύ των δύο μπορεί να αποφέρει πλεονεκτήματα και μειονεκτήματα στον τρόπο λειτουργίας του δικτύου.

#### 4.4. Συμπεράσματα Επιδόσεων

Στο κεφάλαιο αυτό υλοποιήσαμε το πρωτόκολλο δρομολόγησης MEEAΔA και παρουσιάσαμε την απόδοσή του σε σύγκριση με το πρωτόκολλο LEACH. Επιπλέον, με την χρήση διαγραμμάτων παρατηρήσαμε τις μεταβολές που μπορούν να επιφέρουν διάφοροι παράγοντες όπως η απόσταση του σταθμού βάσης, το μέγεθος του δικτύου, και ο αριθμός των κόμβων. Μέσα από τα αποτελέσματα, μπορούμε να βγάλουμε χρήσιμα συμπεράσματα που θα βοηθήσουν στην περαιτέρω εξέλιξη της προτεινόμενης μεθόδου. Η βελτίωση της MEEAΔA θα πρέπει να γίνει με σκοπό την αντιμετώπιση των αδυ-

ναμιών σε τομείς στους οποίους μπορεί να υστερεί έναντι των άλλων πρωτοκόλλων δρομολόγησης.

Από τα αποτελέσματα των προσομοιώσεων, μας δίνεται μια εικόνα για τον γενικότερο τρόπο λειτουργίας της ΜΕΕΑΔΑ. Υπάρχουν κάποιες περιπτώσεις όπου η χρονική απόσταση στην οποία ο πρώτος και ο τελευταίος κόμβος χάνουν την ενέργειά τους, είναι υψηλότερη από αυτή του πρωτοκόλλου LEACH. Επειδή χρησιμοποιείται ο αλγόριθμος μη αρνητικής απόστασης και οι συστάδες που βρίσκονται στην κοντινότερη απόσταση από τον σταθμό βάσης, χάνουν τους κόμβους τους πιο γρήγορα από τις υπόλοιπες. Για κάποιες εφαρμογές, ίσως θα ήταν προτιμότερο να ελέγξουμε την διάρκεια ζωής των κόμβων, ώστε να χάνουν την ενέργειά τους σχεδόν ταυτόχρονα.

Η απώλεια των πιο κοντινών στον σταθμό βάσης συστάδων έχει ως αποτέλεσμα, κάποια μέρη του δικτύου να μην διαθέτουν αισθητήριους κόμβους για ένα χρονικό διάστημα. Για να αποφευχθεί κάτι τέτοιο, θα πρέπει να αλλάξει ο τρόπος λειτουργίας του αλγορίθμου μη αρνητικής δρομολόγησης. Όλες οι συστάδες του δικτύου, θα πρέπει να μοιραστούν το βάρος της μετάδοσης όλων των δεδομένων από την αρχή της λειτουργίας του δικτύου. Η έκδοση που παρουσιάστηκε σε αυτή την εργασία προσφέρει ισορροπημένη κατανάλωση ενέργειας μεταξύ των συστάδων για συγκεκριμένες παραμέτρους. Όταν ο σταθμός βάσης βρίσκεται σε σχετικά κοντινή απόσταση (π.χ. 100m από τον κοντινότερο κόμβο) ή μέσα στο δίκτυο και οι αποστάσεις μετάδοσης κάθε cluster leader είναι σχεδόν ίσες. Αν η κατανάλωση ενέργειας ανά συστάδα είναι περίπου ίση, τότε αυτές θα χάνουν τους κόμβους τους με τον ίδιο ρυθμό και σε σύντομο χρονικό διάστημα.

### 4.5. Βελτίωση πρωτοκόλλου ΜΕΕΑΔΑ

Κατά την διάρκεια της συσταδοποίησης δημιουργήθηκαν συστάδες με σχεδόν ίσο αριθμό κόμβων. Οι κόμβοι μίας μικρότερης συστάδας, σε αντίθεση με αυτούς μίας μεγαλύτερης, επιβαρύνονται συνήθως με μεγαλύτερες αποστάσεις μετάδοσης και αναλαμβάνουν πιο συχνά την ενεργοβόρα θέση του cluster head ή cluster leader. Από την άλλη, ο όγκος των δεδομένων που διαχειρίζονται είναι μικρότερος. Με αυτό τον τρόπο γίνεται μία προσπάθεια ώστε να υπάρχει ισορροπία στην μέση κατανάλωση ενέργειας των συστάδων. Βασίζόμενοι λοιπόν σε αυτές τις παρατηρήσεις, θα μετασχηματίσουμε την λειτουργία της ΜΕΕΑΔΑ έτσι ώστε να πετύχουμε τους ακόλουθους στόχους:

- Μείωση του χρόνου όπου ο πρώτος και ο τελευταίος κόμβος χάνουν την ενέργειά τους.
- Σχεδόν ίσες πιθανότητες απώλειας της ενέργειας για όλους τους κόμβους και όχι περισσότερες για αυτούς που βρίσκονται πιο κοντά στον σταθμό βάσης.

#### 4.5.1. Αλλαγή Δρομολόγησης Πληροφορίας

Η βελτίωση που προτείνεται στο κεφάλαιο αυτό αφορά τον αλγόριθμο δρομολόγησης μη αρνητικής απόστασης. Ο αλγόριθμος αυτός λειτουργεί με γνώμονα τις αποστάσεις των cluster leader από τον σταθμό βάσης. Εδώ, ο cluster leader που προωθεί τα δεδομένα

#### 4. Έλεγχος Επίδοσης

του δικτύου αποφασίζεται από τις ιδιότητες της συστάδας στην οποία ανήκει. Πιο συγκεκριμένα, ο cluster leader που θα προωθήσει τα δεδομένα του δικτύου στον σταθμό βάσης υπολογίζεται από ένα βάρος μετάδοσης  $w_s$  όπου  $s$  ο κωδικός της συστάδας. Το βάρος αυτό υπολογίζεται από την ειδική μέση ενέργεια κάθε συστάδας  $save_s$  και το κόστος μετάδοσης  $lt_s$ , του cluster leader στον σταθμό βάσης.

Η ειδική μέση τιμή  $save_s$  μιας συστάδας με κωδικό  $s$  προκύπτει από την σχέση:

$$save_s = \frac{1}{p} \sum_{i=1}^n (v_i) \quad (4.1)$$

όπου  $v_i$  η ενέργεια που διαθέτει ένας κόμβος  $i$  της συστάδας με κωδικό  $s$ , και  $p$  η τιμή ισορροπίας των συστάδων του δικτύου. Η τιμή ισορροπίας  $p = C'/N'$  υπολογίζεται μία φορά μετά το τέλος της συσταδοποίησης. Προκύπτει από τον αρχικό αριθμό των κόμβων του δικτύου  $C'$  και τον αριθμό συστάδων  $N'$  που δημιουργούνται. Χρησιμοποιείται επειδή η μετάδοση στον σταθμό βάσης οδηγεί συνήθως στην κατανάλωση υψηλών ποσών ενέργειας. Η κατανάλωση αυτή, επηρεάζει περισσότερο την μέση ενέργεια των μικρότερων συστάδων. Με την χρήση της τιμής  $p$ , οι συστάδες με μικρό αριθμό κόμβων θα έχουν μικρότερη ειδική μέση τιμή από τις μεγαλύτερες και κατά συνέπεια μικρότερες πιθανότητες επιλογής. Βέβαια, αυτό θα ισχύει έως ότου οι μεγαλύτερες συστάδες χάσουν αρκετή από την ενέργεια τους και επέλθει ισορροπία με τις μικρότερες.

Η σχέση που περιγράφει το κόστος μετάδοσης  $lt_s$  είναι:

$$lt_s = E_{Tx-amp}(k, d) = E_{amp} * k * d^2 \quad (4.2)$$

όπου  $E_{Tx-amp}(k, d)$  η κατανάλωση ενέργειας του ενισχυτή μετάδοσης, η οποία προκύπτει από το κόστος μετάδοσης ενός μηνύματος  $k$ ,  $E_{Tx}(k, d) = E_{Tx-amp}(k, d) + E_{Tx-elec}(k)$  του μοντέλου ενέργειας που χρησιμοποιείται από το δίκτυο (βλ. Κεφ. 2.1). Η κατανάλωση ενέργειας των κυκλωμάτων μετάδοσης  $E_{Tx-elec}(k)$  δεν υπολογίζεται καθώς είναι ίση για όλους τους cluster leader.

Το βάρος μετάδοσης  $w_s$  υπολογίζεται από τις σχέσεις (4.1) και (4.2):

$$w_s = save_s - lt_s \quad (4.3)$$

Σύμφωνα με την σχέση (4.3), περισσότερες πιθανότητες επιλογής έχουν οι συστάδες με μεγαλύτερη ειδική μέση ενέργεια και μικρότερο κόστος μετάδοσης. Ο αλγόριθμος δρομολόγησης μη αρνητικής απόστασης θεωρεί ότι ο cluster leader της συστάδας  $s$  με το μεγαλύτερο βάρος  $w_s$ , είναι αυτός που βρίσκεται στην κοντινότερη απόσταση από τον σταθμό βάσης. Έχοντας αυτό ως δεδομένο, υπολογίζει το σχέδιο δρομολόγησης για όλους τους cluster leader όλων των συστάδων.

#### 4.5.2. Επιδόσεις Προτεινόμενης Βελτίωσης

Οι επιδόσεις της προτεινόμενης βελτίωσης του πρωτοκόλλου ΜΕΕΑΔΑ (ΒΜΕΕΑΔΑ) θα συγκριθούν με αυτές της κύριας έκδοσης (ΜΕΕΑΔΑ) και του πρωτοκόλλου LEACH. Η σύγκριση θα αφορά την διάρκεια ζωής του δικτύου και την μέση κατανάλωση

#### 4.5. Βελτίωση πρωτοκόλλου ΜΕΕΑΔΑ

ενέργειας. Θα χρησιμοποιηθούν οι ίδιες παράμετροι με αυτές των Σχημάτων 4.3, 4.4, για τον έλεγχο της διάρκειας ζωής του δικτύου και αυτές των Σχημάτων 4.9, 4.10, για την μέση κατανάλωση ενέργειας. Πριν από την παρουσίαση των διαγραμμάτων θα πραγματοποιηθεί η ενημέρωση του Πίνακα 4.4 (βλ. Κεφ. 4.2), ώστε να συμπεριληφθούν και οι επιδόσεις της ΒΜΕΕΑΔΑ. Έτσι, θα έχουμε μία πολύ καλή εικόνα για το πώς ανταποκρίνεται στα ίδια σενάρια που δοκιμάστηκαν τα πρωτόκολλα LEACH και ΜΕΕΑΔΑ.

	Ενέργεια (J/node)	Πρωτόκολλο	Πρώτος κόμβος εκτός λειτουργίας (Round)	Τελευταίος κόμβος εκτός λειτουργίας (Round)
50m × 50m	0.25	LEACH	409	639
		ΜΕΕΑΔΑ	803	1179
		ΒΜΕΕΑΔΑ	855	1036
	0.5	LEACH	865	1252
		ΜΕΕΑΔΑ	1623	2350
		ΒΜΕΕΑΔΑ	1715	2055
	1	LEACH	1752	2508
		ΜΕΕΑΔΑ	3245	4678
		ΒΜΕΕΑΔΑ	3441	4091
100m × 100m	0.25	LEACH	167	317
		ΜΕΕΑΔΑ	382	773
		ΒΜΕΕΑΔΑ	555	660
	0.5	LEACH	329	562
		ΜΕΕΑΔΑ	773	1516
		ΒΜΕΕΑΔΑ	1161	1297
	1	LEACH	668	1059
		ΜΕΕΑΔΑ	1547	3011
		ΒΜΕΕΑΔΑ	2287	2546

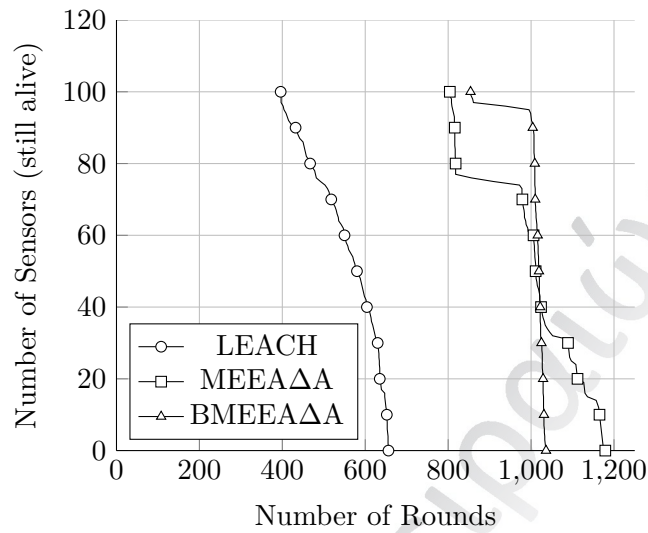
Πίνακας 4.5.: Αποτελέσματα προσομοιώσεων των πρωτοκόλλων δρομολόγησης ΜΕΕΑΔΑ, LEACH και ΒΜΕΕΑΔΑ.

Παρατηρώντας τα αποτελέσματα του Πίνακα 4.5 μπορούμε να συμπεράνουμε ότι ο χρόνος (round ή κύκλος δικτύου) όπου ο πρώτος κόμβος χάνει την ενέργειά του, είναι μεγαλύτερος στην ΒΜΕΕΑΔΑ απ' ό,τι στην ΜΕΕΑΔΑ. Το αντίθετο συμβαίνει για τον χρόνο που αφορά την απώλεια του τελευταίου κόμβου του δικτύου. Η διαφορά της ΒΜΕΕΑΔΑ από το πρωτόκολλο LEACH είναι εμφανής. Για τις προσομοιώσεις που αφορούν τα δίκτυα μεγέθους 100m × 100m η διαφορά στον χρόνο όπου ο πρώτος κόμβος τίθεται εκτός λειτουργίας είναι παραπάνω από τριπλάσια.

Στην ΒΜΕΕΑΔΑ, ο πρώτος και ο τελευταίος κόμβος χάνουν την ενέργειά τους σε συντομότερο χρονικό διάστημα από την ΜΕΕΑΔΑ. Για τις προσομοιώσεις δικτύων με μέγεθος 50m × 50m, 100 κόμβων και 0.25J/κόμβο, η μέση διάρκεια ζωής κάθε κόμβου είναι  $\approx 1013$  round για την ΒΜΕΕΑΔΑ και  $\approx 1002$  για την ΜΕΕΑΔΑ. Αυτό είναι ιδιαίτερα σημαντικό αν αναλογιστούμε ότι ο χρόνος που ο τελευταίος κόμβος του

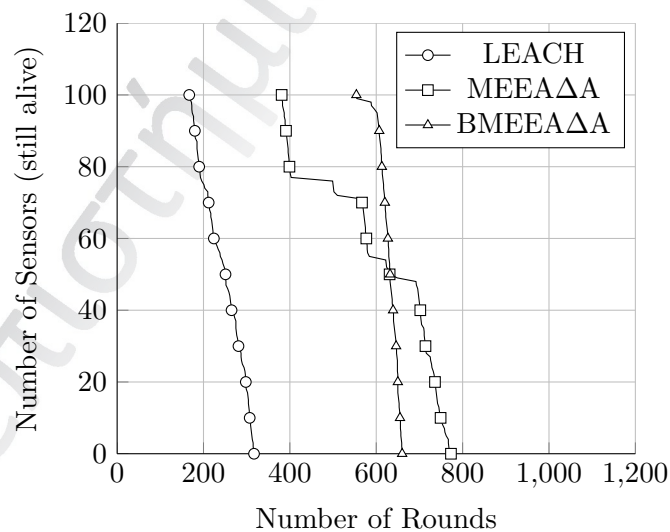
#### 4. Έλεγχος Επίδοσης

δικτύου χάνει την ενέργειά του είναι 1036 round για την BMEEAΔA και 1179 round για την MEEAΔA.



Σχήμα 4.14.: Διάρκεια ζωής του δικτύου μεγέθους  $50m \times 50m$ , 100 κόμβων και  $0.25J/\text{κόμβο}$  (με την BMEEAΔA).

Στο Σχήμα 4.14 παρατηρούμε τις επιδόσεις των τριών πρωτοκόλλων για παραμέτρους όμοιες με αυτές του Σχήματος 4.3. Η θέση του σταθμού βάσης είναι η [25][150].

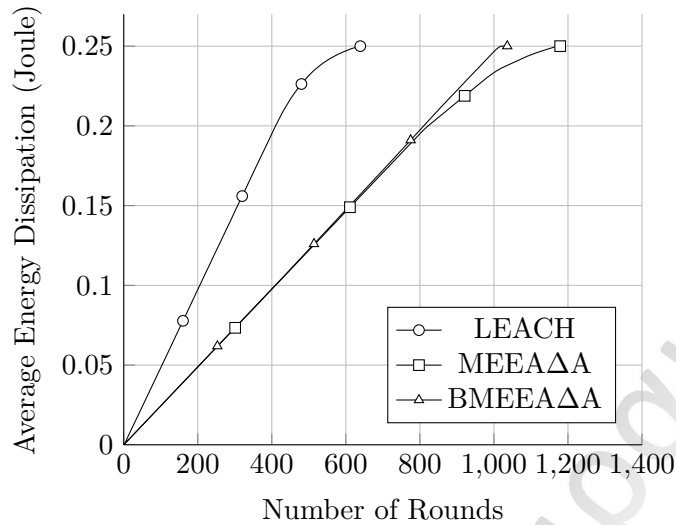


Σχήμα 4.15.: Διάρκεια ζωής του δικτύου μεγέθους  $100m \times 100m$ , 100 κόμβων και  $0.25J/\text{κόμβο}$  (με την BMEEAΔA).

Για τα δίκτυα μεγέθους  $100m \times 100m$  (Σχήμα 4.15) χρησιμοποιήθηκαν οι ίδιες παράμε-

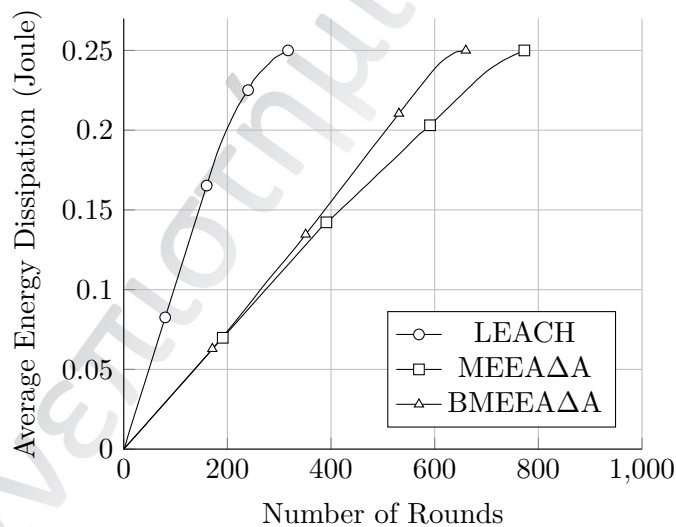


τροι με αυτές του Σχήματος 4.4. Ο σταθμός βάσης τοποθετείται στην θέση [50][200].



Σχήμα 4.16.: Μέση κατανάλωση ενέργειας για δίκτυα μεγέθους  $50m \times 50m$ , 100 κόμβους και  $0.25J/κόμβο$ .

Αυτή την φορά (Σχήμα 4.16), ελέγχουμε την μέση κατανάλωση της ενέργειας για δίκτυα μεγέθους  $50m \times 50m$  (Σχήμα 4.14). Παρατηρούμε ότι δεν υπάρχουν μεγάλες διαφορές πριν από τους 800 κύκλους δικτύου.



Σχήμα 4.17.: Μέση κατανάλωση ενέργειας για δίκτυα μεγέθους  $100m \times 100m$ , 100 κόμβους και  $0.25J/κόμβο$  (με την BMEEAΔA).

Στο Σχήμα 4.17 παρατηρούμε την μέση κατανάλωση ενέργειας για τα δίκτυα μεγέθους

#### 4. Έλεγχος Επίδοσης

$100m \times 100m$  (Σχήμα 4.15). Οι διαφορές γίνονται πιο αισθητές από ένα σημείο και μετά της λειτουργίας του δικτύου.

##### 4.5.3. Συμπεράσματα Προτεινόμενης Βελτίωσης

Τα αποτελέσματα των προσομοιώσεων έδειξαν ότι οι στόχοι που τέθηκαν στην αρχή του Κεφαλαίου 4.5 επιτεύχθηκαν σε μεγάλο βαθμό. Οι διαφορές στον χρόνο που οι κόμβοι χάνουν την ενέργειά τους μειώθηκαν αρκετά. Αυτό οφείλεται στον τρόπο λειτουργίας της δρομολόγησης των συστάδων. Η ΒΜΕΕΑΔΑ, αν και χρησιμοποιεί περισσότερη ενέργεια για την επικοινωνία των cluster leader με τον προορισμό, καταφέρνει να προσδώσει στους κόμβους μεγαλύτερη μέση διάρκεια ζωής. Αν όμως ο σταθμός βάσης βρίσκεται κοντά στο δίκτυο, τότε οι αποστάσεις μετάδοσης των cluster leader είναι σχεδόν ίσες. Σε αυτές τις περιπτώσεις οι επιδόσεις της ΜΕΕΑΔΑ θα είναι όμοιες με αυτές της ΒΜΕΕΑΔΑ.

Το πλεονέκτημα της ΒΜΕΕΑΔΑ αυξάνεται όσο αυξάνονται και οι διαφορές στις αποστάσεις μετάδοσης των cluster leader. Όπως παρατηρήσαμε και στα αποτελέσματα των προσομοιώσεων, οι διαφορές των δύο πρωτοκόλλων είναι μεγαλύτερες για τα δίκτυα μεγέθους  $100m \times 100m$  (βλ. Πίνακας 4.5,  $0.5J/\text{κόμβο}$ ). Αυτό συμβαίνει γιατί ο σταθμός βάσης τοποθετείται σε απόσταση  $200m$  τουλάχιστον ( $[50][200]$ ) από τον κοντινότερο κόμβο. Στην ΒΜΕΕΑΔΑ, η μέση διάρκεια ζωής κάθε κόμβου είναι  $\approx 1267$  round (τελευταίος κόμβος εκτός λειτουργίας 1297 round) ενώ στην ΜΕΕΑΔΑ  $\approx 1201$  (τελευταίος τελευταίος κόμβος εκτός λειτουργίας 1516 round).

Στην εργασία αυτή παρουσιάστηκε μία νέα προτεινόμενη μέθοδος εξοικονόμησης ενέργειας για τα ασύρματα δίκτυα αισθητήρων (ΜΕΕΑΔΑ). Οι αλγόριθμοι που αποτελούν αυτό το πρωτόκολλο δρομολόγησης περιγράφηκαν λεπτομερώς και με την χρήση παραδειγμάτων. Μετά την θεωρητική ανάλυση ακολούθησε η υλοποίησή του. Η έξοδος του προγράμματος παρουσιάστηκε μέσα από πραγματικά παραδείγματα εκτέλεσης. Στην συνέχεια διαπιστώσαμε τις επιδόσεις της ΜΕΕΑΔΑ για διάφορες παραμέτρους και τις συγκρίναμε με αυτές του πρωτοκόλλου LEACH. Στο κεφάλαιο αυτό παρατίθενται τα συμπεράσματα της εργασίας.

### 5.1. Προτεινόμενο Πρωτόκολλο Δρομολόγησης

Η ΜΕΕΑΔΑ σχεδιάστηκε με σκοπό να αποτελέσει ένα ευέλικτο πρωτόκολλο με πολλές δυνατότητες παραμετροποίησης. Ξεκινώντας με την διαδικασία της συσταδοποίησης, έχουμε την εισαγωγή μία πολυεπίπεδης διαδικασίας, η οποία στοχεύει στον σχηματισμό υποδικτύων. Με την χρήση των υποδικτύων περιορίζεται ο αριθμός των κόμβων κάθε συστάδας ή υποδικτύου με αποτέλεσμα να είναι δυνατή η συμμετοχή των κόμβων στους υπολογισμούς της δρομολόγησης. Οι συστάδες-υποδίκτυα σχηματίζονται μόνο μία φορά στην αρχή για να μην επιβαρύνονται οι κόμβοι με επιπλέον κόστος κατανάλωσης ενέργειας. Ο αριθμός τους υπολογίζεται από το πλήθος των κόμβων του δικτύου. Αφού ολοκληρωθεί η συσταδοποίηση, κάθε κόμβος λαμβάνει πίνακες δρομολόγησης περιορισμένου μεγέθους.

Σε κάθε συστάδα-υποδίκτυο η διαδικασία της συλλογής της πληροφορίας ανατίθεται σε τρεις κόμβους που ονομάζονται cluster head. Καθένας από τους cluster head εκλέγεται με κριτήρια την διαθέσιμη ενέργεια και τις μεταξύ αποστάσεις των κόμβων, για να επιτελέσει ένα συγκεκριμένο σκοπό. Από μόνοι τους σχηματίζουν μία ιεραρχία, όπου αυτός που βρίσκεται σε χαμηλότερο επίπεδο, συγκεντρώνει πληροφορία για αυτόν που βρίσκεται στο υψηλότερο. Η θέση των κόμβων είναι τέτοια ώστε να ελαχιστοποιούνται οι αποστάσεις μετάδοσης κάθε κόμβου της συστάδας. Η επιλογή μετάδοσης από τους

## 5. Συμπεράσματα

υπόλοιπους κόμβους, είναι αυτή που δημιουργεί τρεις συστάδες (υποσυστάδες) μέσα στην συστάδα. Οι κόμβοι που αναλαμβάνουν τους ρόλους του cluster head σε κάθε συστάδα-υποδίκτυο αλλάζουν σε κάθε round και έτσι έχουμε την δημιουργία τριών νέων συστάδων (adaptive subclustering).

Σε κάθε συστάδα, η πληροφορία που συνέλεξαν τα αισθητήρια όργανα των κόμβων καταλήγει σε έναν από τους τρεις κόμβους. Ο κόμβος αυτός είναι ο υψηλότερος στην ιεραρχία και ονομάζεται cluster leader. Για να φτάσει η πληροφορία των cluster leader στον προορισμό χρησιμοποιείται ένας απληστος αλγόριθμος δρομολόγησης, ο αλγόριθμος δρομολόγησης μη αρνητικής απόστασης, ο οποίος δημιουργεί ένα μονοπάτι χαμηλού κόστους. Το μονοπάτι αυτό εμπεριέχει όλους τους cluster leader και καταλήγει στον προορισμό. Η απληστία του αλγορίθμου οφείλεται στο γεγονός ότι ένας κόμβος επιλέγει να προωθήσει τα δεδομένα του στον κοντινότερο σε αυτόν cluster leader. Η απόσταση του κόμβου παραλήπτη από τον σταθμό βάσης πρέπει να είναι μικρότερη από την αντίστοιχη του κόμβου αποστολέα.

Όλα τα παραπάνω χαρακτηριστικά συμβάλλουν στην αποτελεσματική διαχείριση της ενέργειας των κόμβων του δικτύου. Η ΜΕΕΑΔΑ μπορεί να λειτουργήσει με την βοήθεια του σταθμού βάσης αλλά και αυτόνομα, χάρη στους περιορισμένους πίνακες δρομολόγησης. Δεν τίθενται περιορισμοί σε ό,τι αφορά το πλήθος των κόμβων και την αρχική τους ενέργεια. Επιπλέον, η δρομολόγηση μεταξύ των cluster leader δεν θέτει περιορισμούς σε ότι αφορά την θέση του σταθμού βάσης. Η υλοποίηση του πρωτοκόλλου είναι σχετικά απλή και η πολυπλοκότητα των πράξεων που εκτελεί χαμηλή. Το προτεινόμενο πρωτόκολλο εξοικονόμησης ενέργειας, συνιστάται για περιπτώσεις όπου είναι απαραίτητη η βελτιστοποίηση της διάρκειας ζωής του δικτύου.

### 5.2. Αξιολόγηση Επιδόσεων

Η αξιολόγηση της ΜΕΕΑΔΑ πραγματοποιήθηκε με την χρήση διαφορετικών παραμέτρων. Οι παράμετροι αυτές είναι το μέγεθος του δικτύου και το μέγεθος του πακέτου, η απόσταση και η θέση του σταθμού βάσης, ο αριθμός και η αρχική ενέργεια των κόμβων. Οι προσομοιώσεις πραγματοποιήθηκαν μέσα από την υλοποίηση της μεθόδου στην γλώσσα προγραμματισμού C. Τα αποτελέσματα έδειξαν την ανωτερότητά της σε σχέση με το αντίστοιχο πρωτόκολλο LEACH. Με την χρήση της ΜΕΕΑΔΑ, ο πρώτος κόμβος χάνει την ενέργειά του πολύ μετά από όταν τερματίσει η λειτουργία του πρωτοκόλλου LEACH. Σε μερικές περιπτώσεις και σε ό,τι αφορά τον χρόνο που κάθε κόμβος χάνει την ενέργειά του, η διαφορά στην απόδοση των δύο πρωτοκόλλων είναι παραπάνω από διπλάσια.

Η μέση κατανάλωση ενέργειας υπολογίστηκε ανά round και συγκρίθηκε με αυτή του πρωτοκόλλου LEACH. Επιπλέον, προσομοιώσεις πραγματοποιήθηκαν και για διαφορετικές παραμέτρους. Η στρατηγική της ελαχιστοποίησης της κατανάλωσης για κάθε round ξεχωριστά είναι αυτή που οδηγεί σε μία καθολική υπεροχή της ΜΕΕΑΔΑ έναντι του LEACH. Ο τρόπος με τον οποίο οι κόμβοι και οι συστάδες χάνουν την ενέργειά τους είναι πάντοτε ανάλογη με τις παραμέτρους του δικτύου. Τα σημεία κλειδιά της ΜΕΕΑΔΑ (συσταδοποίηση, συλλογή πληροφορίας, δρομολόγηση) μπορούν να προσαρ-

## 5.2. Αξιολόγηση Επιδόσεων

μοστούν ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής. Όπως αναφέραμε και στο Κεφάλαιο 4.5, η λειτουργία της μπορεί να βελτιωθεί στον τομέα της δρομολόγησης. Η βελτίωση αυτή αποδεικνύεται ιδιαίτερα σημαντική όταν οι αποστάσεις μετάδοσης των cluster leader στον σταθμό βάσης είναι μεγαλύτερες από αυτές μεταξύ τους.

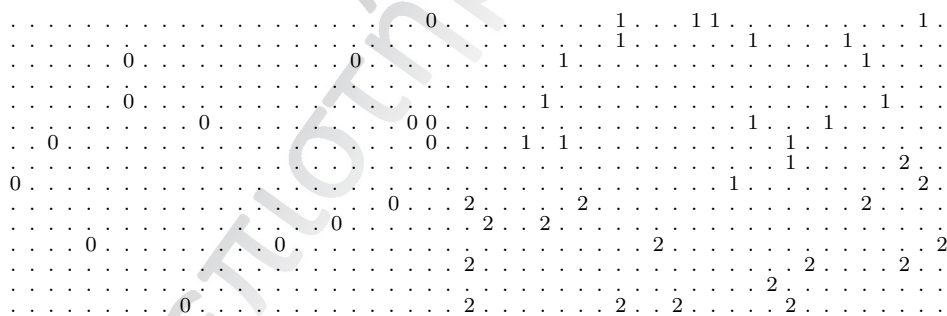
Πανεπιστήμιο Πειραιώς

Πανεπιστήμιο Πειραιώς

Στο κεφάλαιο αυτό παρουσιάζεται ο κώδικας της μεθόδου εξοικονόμησης ενέργειας στα ασύρματα δίκτυα αισθητήρων (ΜΕΕΑΔΑ). Η υλοποίηση έγινε με την χρήση της γλώσσας προγραμματισμού C. Επίσης, παρουσιάζονται μερικά παραδείγματα με τον τρόπο που η ΜΕΕΑΔΑ χωρίζει το δίκτυο σε συστάδες.

### Α'.1. Παραδείγματα Συσταδοποίησης ΜΕΕΑΔΑ

Όλα τα παραδείγματα συσταδοποίησης προέρχονται από την εκτέλεση του προγράμματος. Οι προσομοιώσεις περιλαμβάνουν δίκτυα με διαφορετικό μέγεθος και αριθμό κόμβων.



Σχήμα Α'.1.: Πίνακας με τους κωδικούς (κλειδιά) 50 κόμβων σε δίκτυο διαστάσεων  $15m \times 50m$ , μετά την εκτέλεση της συσταδοποίησης.

Στο Σχήμα Α'.1 παρουσιάζεται το αποτέλεσμα της συσταδοποίησης για 50 κόμβους, σε ένα δίκτυο με διαστάσεις  $15m \times 50m$ . Η συστάδα με κλειδί-κωδικό 0 καταλαμβάνει περισσότερο χώρο στο δίκτυο επειδή η πυκνότητα των κόμβων που την αποτελούν είναι χαμηλή. Οι συστάδες με κλειδί 1 και 2, καταλαμβάνουν σχεδόν ίσο χώρο στο δίκτυο.

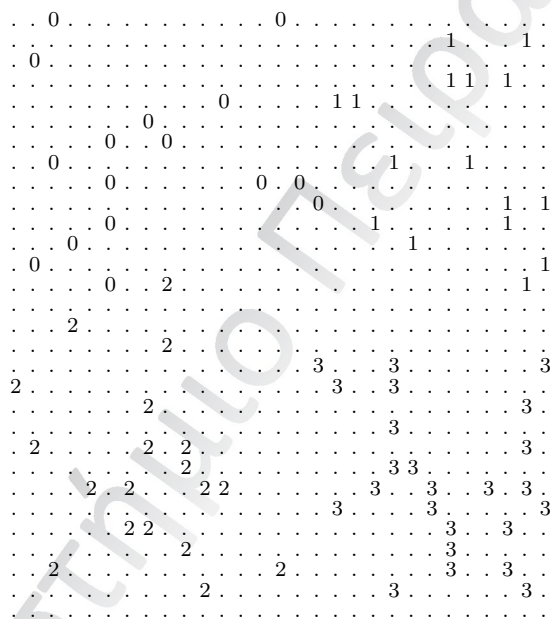
A'. Επιπλέον Υλικό

Ο αριθμός των κόμβων κάθε συστάδας είναι:

Κωδικός Συστάδας	Αριθμός Κόμβων
0	15
1	18
2	17

Πίνακας Α'.1.: Αριθμός κόμβων για κάθε συστάδα του δικτύου του Σχήματος Α'.1.

Στον Πίνακα Α'.1 παρατηρούμε ότι οι συστάδες αποτελούνται από σχεδόν ίσο αριθμό κόμβων. Η συστάδα με κλειδί 0 είναι αυτή με τους λιγότερους κόμβους λόγω της χαμηλότερης πυκνότητας.



Σχήμα Α'.2.: Πίνακας με τους κωδικούς (κλειδιά) 75 κόμβων σε δίκτυο διαστάσεων  $30m \times 30m$ , μετά την εκτέλεση της συσταδοποίησης.

Στο παράδειγμα του Σχήματος Α'.2 έχουμε ένα δίκτυο διαστάσεων  $30m \times 30m$ . Το δίκτυο αυτό αποτελείται από 75 κόμβους που δημιουργούν τέσσερις συστάδες. Ο σχηματισμός τους επηρεάζεται σημαντικά από την πυκνότητα των κόμβων. Για παράδειγμα, οι κόμβοι της συστάδας με κλειδί 3, εμφανίζουν υψηλότερη πυκνότητα από αυτούς της συστάδας 0. Η πυκνότητα των κόμβων καθορίζεται από την εκτέλεση του πρώτου βήματος της συσταδοποίησης (βλ. Κεφ 3.2.1, Αλγόριθμος 1). Στο δεύτερο βήμα της συσταδοποίησης, αναζητείται ίσος αριθμός συστάδων του πρώτου βήματος (βλ. Κεφ 3.2.2, Αλγόριθμος 2).

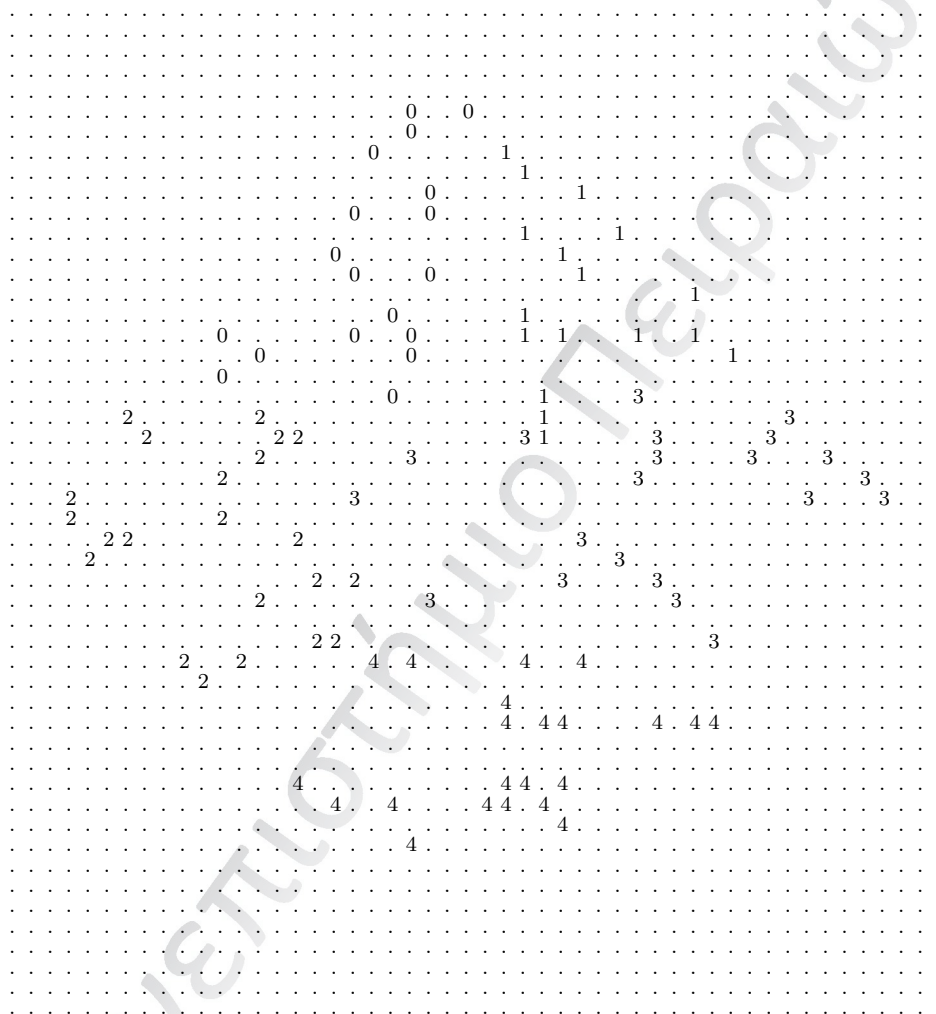
Ο αριθμός των κόμβων κάθε συστάδας είναι:



Α'.1. Παραδείγματα Συσταδοποίησης ΜΕΕΑΔΑ

Κωδικός Συστάδας	Αριθμός Κόμβων
0	16
1	16
2	19
3	24

Πίνακας Α'.2.: Αριθμός κόμβων για κάθε συστάδα του δικτύου του Σχήματος Α'.2.



Σχήμα Α'.3.: Πίνακας με τους κωδικούς (κλειδιά) 100 κόμβων στο κέντρο του δικτύου διαστάσεων  $50m \times 50m$ , μετά την εκτέλεση της συσταδοποίησης.

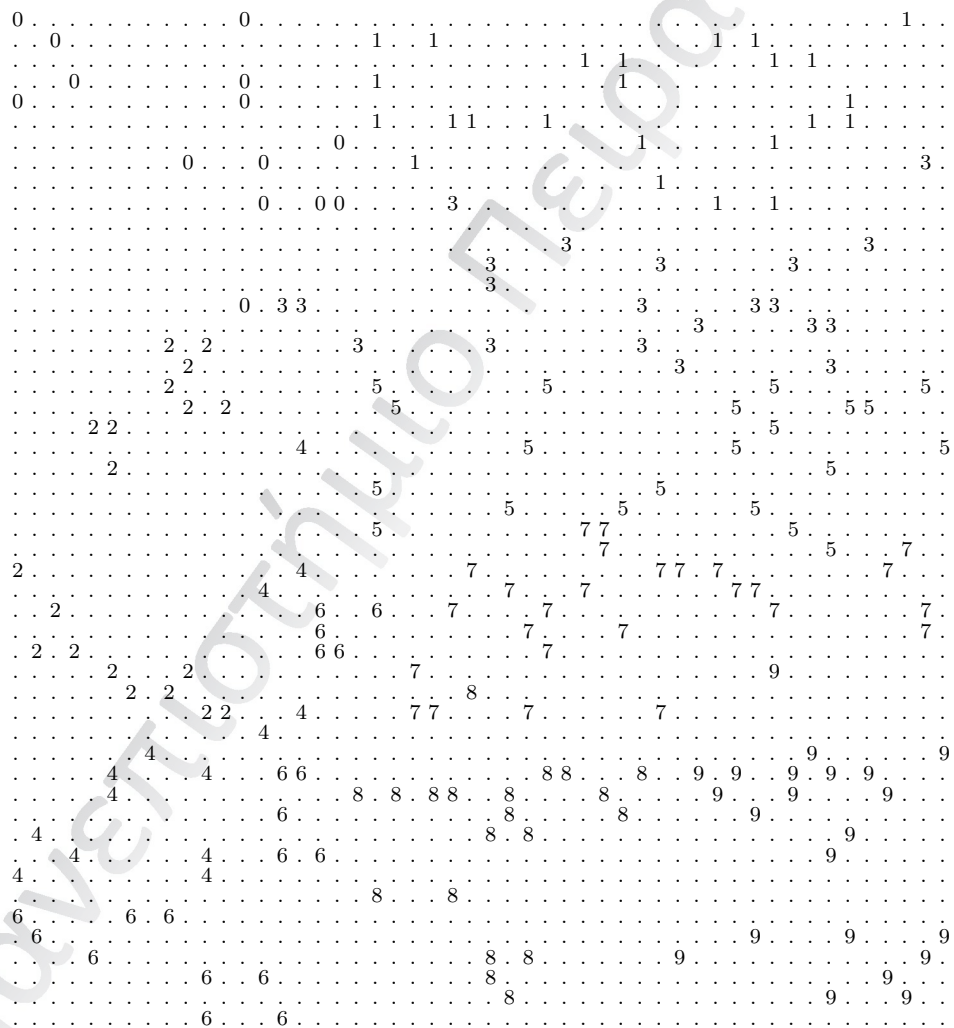
Στο επόμενο παράδειγμα (Σχήμα Α'.3) οι κόμβοι τοποθετούνται στο κέντρο σχημα-

Α'. Επιπλέον Υλικό

τίζοντας ένα κυκλικό δίκτυο.

Κωδικός Συστάδας	Αριθμός Κόμβων
0	18
1	17
2	22
3	21
4	22

Πίνακας Α'.3.: Αριθμός κόμβων για κάθε συστάδα του δικτύου του Σχήματος Α'.3.



Σχήμα Α'.4.: Πίνακας με τους κωδικούς (κλειδιά) 200 κόμβων σε δίκτυο 50m x 50m.

## A'.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

Στο Σχήμα A'.4 έχουμε ένα παράδειγμα συσταδοποίησης (10 συστάδες) για 200 κόμβους, σε δίκτυο διαστάσεων  $50m \times 50m$ . Η πυκνότητα του δικτύου είναι αρκετά υψηλή, με αποτέλεσμα οι περισσότεροι κόμβοι να καταλαμβάνουν μία μικρή περιοχή. Σε κάποιες συστάδες (π.χ. συστάδα με κλειδί 4) παρατηρούμε ότι κάποιοι κόμβοι βρίσκονται σε μεγάλη απόσταση σε σχέση με τους υπόλοιπους. Οι κόμβοι αυτοί επιβαρύνονται με μεγαλύτερες αποστάσεις μετάδοσης. Αυτό όμως δεν αποτελεί σημαντικό μειονέκτημα. Οι κόμβοι αυτοί θα αναλάβουν λιγότερες φορές την θέση του primary, ο οποίος θα επιλέγεται ως cluster leader για τους περισσότερους κύκλους δικτύου.

Κωδικός Συστάδας	Αριθμός Κόμβων
0	14
1	24
2	19
3	21
4	14
5	21
6	19
7	26
8	20
9	22

Πίνακας A'.4.: Αριθμός κόμβων για κάθε συστάδα του δικτύου του Σχήματος A'.4.

Ο σταθμός βάσης τοποθετείται στην θέση  $[25][-100]$  για όλα τα παραδείγματα. Η θέση του σταθμού βάσης επηρεάζει τον σχηματισμό των συστάδων (βλ. Αλγόριθμος 1). Σε ό,τι αφορά την θέση των κόμβων στο δίκτυο, θα πρέπει να είναι εντός των ορίων αναζήτησης των αλγορίθμων συσταδοποίησης.

## A'.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

Πριν από την παρουσίαση του κώδικα, είναι σημαντικό να αναφέρουμε κάποιες παραμέτρους που καθορίζουν την λειτουργία του προγράμματος. Τα σημεία  $i, j$  (βλ. Κεφάλαιο 4.1.2) διαφέρουν από αυτά των  $x, y$  στο Σχήμα 3.2. Για την εύκολη αναζήτηση αυτών των παραμέτρων στον κώδικα, χρησιμοποιούμε και τον αριθμό της γραμμής που περιέχονται. Οι πιο χρήσιμοι παράμετροι είναι:

- `msensors` (γραμμή 13): Ορίζει τον μέγιστο αριθμό των αισθητήριων κόμβων που θα χρησιμοποιήσει το πρόγραμμα.
- `maximum_i` (γραμμή 14): Το μέγεθος του δικτύου, μέγιστο σημείο  $i$ .
- `maximum_j` (γραμμή 15): Το μέγεθος του δικτύου, μέγιστο σημείο  $j$ .
- `bspos_x` (γραμμή 17): Η θέση του σταθμού βάσης στο σημείο  $i$ .

## A'. Επιπλέον Υλικό

- `bspos_y` (γραμμή 18): Η θέση του σταθμού βάσης στο σημείο `j`.
- `initenergy` (γραμμή 23): Ορίζει την αρχική ενέργεια που διαθέτει κάθε κόμβος του δικτύου.
- `density` (γραμμή 34): Ορίζει την πυκνότητα των κόμβων στο δίκτυο. Όσο αυξάνεται αυτή η τιμή, τόσο αυξάνεται η περιοχή (πίνακας) που καταλαμβάνουν οι κόμβοι. Αν είναι πολύ υψηλή, μπορεί να οδηγήσει στην δημιουργία μικρότερου αριθμού κόμβων, από αυτόν του `msensors`.
- `packetize` (γραμμή 36): Ορίζει το μέγεθος του πακέτου που χρησιμοποιείται από τους κόμβους.

Όταν εκτελείται το πρόγραμμα, παρουσιάζει τον αρχικό πίνακα με τους κόμβους και μετέπειτα το αποτέλεσμα των βημάτων της συσταδοποίησης. Με εντολή του χρήστη (πληκτρο `Enter`), ξεκινά η κύρια λειτουργία. Για την προβολή των υπολογισμών του προγράμματος ανά `round`, πρέπει να αφαιρεθεί το σχόλιο `/*` από την εντολή `getchar()` (γραμμή 1012).

Με την ολοκλήρωση της εκτέλεσης του προγράμματος δημιουργείται ένα αρχείο `.txt` με το όνομα `round_record`. Το αρχείο αυτό περιέχει πληροφορίες όπως η ώρα έναρξης, ο αριθμός των κόμβων, η αρχική ενέργεια κάθε κόμβου και διάφορες παραμέτρους σχετικά με την προσομοίωση. Επίσης, καταγράφεται ο χρόνος (`round`) όπου κάθε κόμβος του δικτύου χάνει την ενέργειά του. Τα περιεχόμενα του αρχείου `round_record` αντικαθίστανται όταν ξεκινά μία νέα προσομοίωση.

```
1  /*
2  Author: Emmanouil P. Krikelis
3  Name: A method for energy efficient routing in WSNs – MEEADA
4  Version: v2wqar
5  */
6
7  #include <stdio.h>
8  #include <math.h>
9  #include <time.h>
10 #include <stdlib.h>
11 #include <string.h>
12
13 #define msensors 100
14 #define maximum_i 50
15 #define maximum_j 50
16
17 #define bspos_x 25
18 #define bspos_y -100
19 #define bsclust -1
20
21 #define knode (msensors*2)+10
22
23 #define initenergy 0.5
```

```

24
25
26 int i=0;
27 int j=0;
28
29 int max_i=maximum_i;
30 int max_j=maximum_j;
31 int A[maximum_i][maximum_j];
32
33 int max_sensors=msensors;
34 int density=22;
35
36 int packetsize=2000;
37
38 int x=bspos_x;
39 int y=bspos_y;
40
41 int x2=bspos_x;
42 int y2=bspos_y;
43
44 int network_rounds=0;
45 double max_header=0;
46
47 typedef struct
48 {
49     int clust;
50     double energy;
51     int head;
52     int wcounter;
53 }wsensor;
54
55 wsensor B[maximum_i][maximum_j];
56
57 typedef struct schnode
58 {
59     int cluster1;
60     int cluster2;
61     struct schnode *right;
62 }schnode;
63
64 schnode *newitem (schnode *rpointer, int clust1, int clust2);
65 schnode *DelSchedule (schnode *rpointer);
66
67 double minimumdist=10000;
68
69 double threshold=5;
70 int clr_grade=msensors;
71
72 int node_num=0;
73 double node_numa=0;
74 int divider=0;
75 int number=0; //sensors
76
77 int number_2=0; //clusters

```

```

78 | int tot_sensor_clu=0;
79 |
80 | int k=knode;
81 |
82 | char m = '.';
83 | int sensor=0;
84 |
85 | int z=0;
86 | int number_of_nodes=0;
87 | int extrahead=1;
88 |
89 | void swap(int *a, int *b, int *c, int *d);
90 |
91 | double get_elec(int packet);
92 | double get_fusion(int packet_f);
93 | double get_eamp(int packet_amp, double dist);
94 |
95 | double get_euclidean(int xA, int yA, int xB, int yB);
96 |
97 | void check_state(int x1, int y1);
98 | void state_on_receive(int xx1, int yy1);
99 |
100 | void clear_records();
101 |
102 | void siminfo();
103 | void write_rounds(int sens, int wrounds);
104 |
105 | void viewSchedule (schnode *rpointer);
106 |
107 | double simenergy=initenergy;
108 |
109 | int main()
110 | {
111 |     double inienergy=initenergy;
112 |
113 |     schnode newnode, *rpointer;
114 |     clear_records();
115 |
116 |     siminfo();
117 |
118 |     srand(time(0));
119 |
120 |     for (i=1;i<=max_i;i++)
121 |     {
122 |         for (j=1;j<=max_j;j++)
123 |         {
124 |             A[i][j] = (rand()%density) + 1;
125 |             if (i==bspos_x && j==bspos_y)
126 |             {
127 |                 A[i][j]=2;
128 |             }
129 |             if (A[i][j]<=1)
130 |             {
131 |                 if (z<max_sensors)

```

## A'.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

```
132     {
133         A[i][j]=1;
134         z++; //count sensors
135     }
136     else
137     {
138         A[i][j]=k;
139     }
140 }
141 }
142 }
143
144 for (i=1;i<=max_i;i++)
145 {
146     for (j=1;j<=max_j;j++)
147     {
148         if (A[i][j]>1)
149         {
150             A[i][j]=k;
151             printf("%2c", m);
152         }
153         else
154         {
155             A[i][j]=0;
156             number_of_nodes++;
157             printf("%2d", A[i][j]);
158         }
159     }
160     printf("\n");
161 }
162
163 printf("\n\nNumber of sensors generated: %d\n",z);
164 printf("Number of sensors opened: %d",number_of_nodes);
165
166 double euclidean=1;
167
168 int min_x=0;
169 int min_y=0;
170 int tmp=clr_grade;
171 int update=0;
172 double separate_clust=1;
173 int counter=1;
174
175 //mincalculation of euc2;
176
177
178 double euclidean_min=1;
179 double euclideanb_min=0;
180
181 while (counter!=number_of_nodes+1)
182 {
183     double firstmin=minimumdist;
184     for (i=1;i<=max_i;i++)
185     {
```

```

186     for (j=1;j<=max_j;j++)
187     {
188         if (A[i][j]==sensor)
189         {
190
191             euclidean=get_euclidean(x,y,i,j);
192
193             if (euclidean<firstmin)
194             {
195                 min_x=i;
196                 min_y=j;
197                 firstmin=euclidean;
198                 if (separate_clust<euclidean)
199                 {
200                     separate_clust=euclidean;
201                     euclidean_min=minimumdist;
202                 }
203                 else
204                 {
205                     separate_clust=euclidean;
206                 }
207             }
208
209             euclidean_min=get_euclidean(x,y,min_x,min_y);
210
211             if (euclidean_min>euclidean)
212             {
213                 euclidean_min=euclidean;
214             }
215             euclidean=1;
216         }
217     }
218 }
219 }
220
221 counter++;
222 update=tmp-1;
223
224 A[min_x][min_y]=update;
225
226 for (i=1;i<=max_i;i++)
227 {
228     for (j=1;j<=max_j;j++)
229     {
230
231         if (euclideanb_min<euclidean_min || euclidean_min>threshold)
232         {
233             A[min_x][min_y]=tmp++;
234             goto break_this_loop;
235         }
236     }
237 }
238 }
239

```



## A'.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

```
240 break_this_loop ;
241 euclideanb_min=euclidean_min ;
242 x=min_x ;
243 y=min_y ;
244
245 }
246
247 printf("\n\n\n");
248 for (i=1;i<=max_i;i++)
249 {
250     for (j=1;j<=max_j;j++)
251     {
252         if (A[i][j]==sensor)
253         {
254             printf("%2d",sensor);
255         }
256         else if (A[i][j]!=0 && A[i][j]!=k)
257         {
258             printf("%2d", A[i][j]-(clr_grade-1));
259         }
260         else
261         {
262             printf("%2c",m);
263         }
264     }
265     printf("\n");
266 }
267
268 printf("\n\nNumber of Clusters: %d\n\n", tmp-clr_grade);
269
270
271 number=z;          //sensors
272
273 number_2=tmp-clr_grade;    //clusters
274 tot_sensor_clu=tmp-clr_grade;
275 double n=number;
276 node_numa = (n*5)/100;
277
278 int node_num = (int) node_numa;
279
280 if(node_numa - node_num == 0)
281 {
282     node_num ;
283 }
284 else
285 {
286     node_num=node_num+1;
287 }
288
289 if (node_num<=1)
290 {
291     node_num=1;
292 }
293
```

```

294 divider=node_num;
295
296 int maxi_x=1;
297 int maxi_y=1;
298 int mini_x=1;
299 int mini_y=1;
300
301 counter=0;
302 int cur_val=0;
303 int cur_clust_val=0;
304 int total_sensors=0;
305 int trigger=1;
306
307 int horizontal=1;
308 int vertical=1;
309
310 while (divider!=0)
311 {
312     if (divider==2)
313     {
314         number_2=(number_2+1)/divider;
315     }
316     else
317     {
318         number_2=number_2/divider;
319     }
320     counter=0;
321     while (counter!=number_2)
322     {
323
324         if (maxi_y==max_j && maxi_x==max_i)
325         {
326             mini_x=1;
327             mini_y=max_j/2;
328             maxi_x=mini_x;
329             maxi_y=mini_y;
330             vertical=1;
331             horizontal=1;
332             counter++;
333         }
334
335         if (maxi_x > 0 && maxi_x < max_i)
336         {
337             maxi_x=maxi_x+horizontal;
338             if (maxi_x>=max_i)
339             {
340                 maxi_x=max_i;
341             }
342         }
343
344         if (maxi_y > 0 && maxi_y < max_j)
345         {
346             maxi_y=maxi_y+vertical;
347             if (maxi_y>=max_j)

```

## A'.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

```
348     {
349         maxi_y=max_j;
350     }
351 }
352
353 for (i=mini_x; i<=maxi_x; i++)
354 {
355     for (j=mini_y; j<=maxi_y; j++)
356     {
357         if (counter!=number_2)
358         {
359             if (A[i][j]!=k && A[i][j]>cur_clust_val)
360             {
361                 counter++;
362                 cur_val = A[i][j];
363                 for (i=1; i<=max_i; i++)
364                 {
365                     for (j=1; j<=max_j; j++)
366                     {
367                         if (A[i][j]==cur_val)
368                         {
369                             A[i][j]=cur_clust_val;
370                             total_sensors++;
371                         }
372                     }
373                 }
374             }
375         }
376     }
377 }
378
379 }
380
381 if (trigger==0)
382 {
383     mini_x=1;
384     mini_y=1;
385     maxi_x=1;
386     maxi_y=1;
387     vertical=1;
388     horizontal=max_j/10 + (max_j%100!=0);
389     trigger++;
390 }
391 else
392 {
393     mini_x=1;
394     mini_y=maxi_y;
395     maxi_x=mini_x;
396     maxi_y=mini_y;
397     vertical=max_i/10 + (max_i%100!=0);
398     horizontal=1;
399     trigger--;
400 }
401 cur_clust_val++;
```

```

402     tot_sensor_clu=tot_sensor_clu-number_2;
403     number_2=tot_sensor_clu;
404     divider--;
405 }
406
407 cur_clust_val--;
408 node_num--;
409
410 for (i=1; i<=max_i; i++)
411 {
412     for (j=1; j<=max_j; j++)
413     {
414
415         if (A[i][j]>cur_clust_val && A[i][j]!=k)
416         {
417             A[i][j]=node_num;
418         }
419         if (A[i][j]!=k)
420         {
421             printf("%2d", A[i][j]);
422         }
423         else
424         {
425             printf("%2c",m);
426         }
427     }
428     printf("\n");
429 }
430 getchar();
431 printf("\n");
432
433 for (i=1;i<=max_i;i++)
434 {
435     for (j=1;j<=max_j;j++)
436     {
437         if (A[i][j]!=k)
438         {
439             B[i][j].clust=A[i][j];
440             B[i][j].energy=inienergy;
441             B[i][j].head=0;
442             B[i][j].wcounter=0;
443         }
444         else
445             B[i][j].clust=k;
446     }
447 }
448
449 //print structure
450 for (i=1;i<=max_i;i++)
451 {
452     for (j=1;j<=max_j;j++)
453     {
454         if (B[i][j].clust!=k)
455         {

```

## A'.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

```
456     printf("%2d", B[i][j].clust);
457     }
458     else
459     {
460         printf("%2c", m);
461     }
462     }
463     printf("\n");
464     }
465
466     int chk_node_num=0;
467     int i_max=0;
468     int j_max=0;
469
470     double euc_maxim=1;
471     double euc_maxim2=0;
472     int euci=0;
473     int eucj=0;
474
475     double shead_energy=0;
476
477     while (z!=0)
478     {
479
480     for (i=1;i<=max_i;i++)
481     {
482         for (j=1;j<=max_j;j++)
483         {
484             if (B[i][j].head!=0)
485             {
486                 B[i][j].head=0;
487             }
488         }
489     }
490
491     //set-up headers
492     while (chk_node_num != node_num+1)
493     {
494         for (i=1;i<=max_i;i++)
495         {
496             for (j=1;j<=max_j;j++)
497             {
498                 if (B[i][j].clust==chk_node_num && B[i][j].clust!=k)
499                 {
500                     if (B[i][j].energy > max_header)
501                     {
502                         max_header=B[i][j].energy;
503                         i_max=i;
504                         j_max=j;
505                     }
506                 }
507             }
508         }
509     }
```

```

510  if (i_max!=0 && j_max!=0)
511  {
512    B[i_max][j_max].head=1;
513    for (i=1;i<=max_i;i++)
514    {
515      for (j=1;j<=max_j;j++)
516      {
517        if (B[i][j].clust==chk_node_num && B[i][j].head!=1)
518        {
519          euc_maxim=get_euclidean(i_max,j_max,i,j);
520
521          if (euc_maxim > euc_maxim2 && B[i][j].energy>=shead_energy)
522          {
523            euc_maxim2=euc_maxim;
524            euc_i=i;
525            euc_j=j;
526            shead_energy=B[i][j].energy;
527          }
528        }
529        euc_maxim=1;
530      }
531    }
532    B[euc_i][euc_j].head=2;
533    euc_maxim2=0;
534  }
535  chk_node_num++;
536  max_header=0;
537  shead_energy=0;
538  i_max=0;
539  j_max=0;
540  euc_i=0;
541  euc_j=0;
542 }
543 printf("\n");
544 // end set-up headers
545
546 for (i=1;i<=max_i;i++)
547 {
548   for (j=1;j<=max_j;j++)
549   {
550     if (B[i][j].clust!=k)
551     {
552       printf("%2d", B[i][j].head);
553     }
554     else
555     {
556       printf("%2c", m);
557     }
558   }
559   printf("\n");
560 }
561
562 int head_i=0;
563 int head_j=0;

```

```

564
565 int shead_i=0;
566 int shead_j=0;
567 int turf_i=0;
568 int turf_j=0;
569
570 double euclidean_rout=1;
571 double seuclidean_rout=1;
572 double sseuclidean_rout=1;
573 chk_node_num=0;
574
575 double euclidean_rout1=1;
576 double euclidean_rout2=1;
577 double euclidean_rout3=1;
578 double euclidean_rout4=1;
579 double euclidean_rout5=1;
580
581 double euclidean_rout6=1;
582 double euclidean_rout7=0;
583
584 double bseuc=1;
585 double bseucA=1;
586
587 double mineuclidean=1;
588
589 int mineuc_i=0;
590 int mineuc_j=0;
591 int mineucA_i=0;
592 int mineucA_j=0;
593
594 int mineu6_i=0;
595 int mineu6_j=0;
596 int mineu7_i=0;
597 int mineu7_j=0;
598
599 double transmission_energy=0;
600 double e_elec=0;
601 double e_amp=0;
602 double fusion=0;
603
604 double turfdist1=1;
605 double turfdist2=minimumdist;
606 double turfdist3=1;
607 double turf_energy=0;
608
609 double adv_min=minimumdist;
610
611 while (chk_node_num != node_num+1)
612 {
613     adv_min=minimumdist;
614     head_i=0;
615     head_j=0;
616     shead_i=0;
617     shead_j=0;

```

```

618   turf_i=0;
619   turf_j=0;
620
621   for (i=1;i<=max_i;i++)
622   {
623     for (j=1;j<=max_j;j++)
624     {
625       if (B[i][j]. clust==chk_node_num)
626       {
627         if (B[i][j]. head==1)
628         {
629           head_i=i;
630           head_j=j;
631           goto exit_loop1;
632         }
633       }
634     }
635   }
636   exit_loop1 ;;
637
638   if (head_i==0 && head_j==0)
639   {
640     goto headiszero;
641   }
642
643   for (i=1;i<=max_i;i++)
644   {
645     for (j=1;j<=max_j;j++)
646     {
647       if (B[i][j]. clust==chk_node_num)
648       {
649         if (B[i][j]. head==2)
650         {
651           shead_i=i;
652           shead_j=j;
653           goto exit_loop2;
654         }
655       }
656     }
657   }
658   exit_loop2 ;;
659
660   printf("primary head[%d][%d]\n", head_i, head_j);
661   printf("secondary head[%d][%d]\n", shead_i, shead_j);
662
663   for (i=1;i<=max_i;i++)
664   {
665     for (j=1;j<=max_j;j++)
666     {
667       if (B[i][j]. clust==chk_node_num && B[i][j]. head!=1 && B[i][j]. head
668       !=2)
669       {
670         euclidean_rout=get_euclidean(head_i, head_j, i, j);

```



## A'.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

```
671
672     if (shead_i != 0 && shead_j != 0)
673     {
674         seuclidean_rout=get_euclidean(shead_i ,shead_j ,i ,j);
675     }
676     }
677     else
678     {
679         seuclidean_rout=minimumdist;
680     }
681     //first level energy
682
683     if (euclidean_rout <= seuclidean_rout)
684     {
685     /*turf distance*/
686         turfdist1=seuclidean_rout;
687
688         turfdist3=get_euclidean(head_i ,head_j ,shead_i ,shead_j);
689
690         if (turfdist1<turfdist2 && turfdist1<turfdist3)
691         {
692             if (B[i][j].energy>=turf_energy)
693             {
694                 turf_i=i;
695                 turf_j=j;
696                 turfdist2=turfdist1;
697                 turf_energy=B[i][j].energy;
698             }
699         }
700     }
701     }
702     turfdist1=1;
703     euclidean_rout=1;
704     seuclidean_rout=1;
705     turfdist3=1;
706 }
707 }
708
709 turfdist2=minimumdist;
710 turf_energy=0;
711
712 if (shead_i != 0 && shead_j != 0)
713 {
714     if (turf_i == 0 && turf_j == 0)
715     {
716         turf_i=head_i;
717         turf_j=head_j;
718     }
719     else
720     {
721         B[turf_i][turf_j].head=3;
722     }
723 }
724
```

```

725 printf("in turf head[%d][%d]\n\n", turf_i , turf_j);
726
727 for ( i=1;i<=max_i;i++)
728 {
729     for ( j=1;j<=max_j;j++)
730     {
731         if (B[i][j]. clust==chk_node_num && B[i][j]. head!=1 && B[i][j]. head
732         !=2 && B[i][j]. head!=3)
733         {
734             euclidean_rout=get_euclidean ( head_i , head_j , i , j);
735
736             if ( shead_i != 0 && shead_j != 0)
737             {
738
739                 seuclidean_rout=get_euclidean ( shead_i , shead_j , i , j);
740
741                 if (B[turf_i][turf_j]. head != B[head_i][head_j]. head)
742                 {
743                     sseuclidean_rout=get_euclidean ( turf_i , turf_j , i , j);
744                 }
745                 else
746                 {
747                     sseuclidean_rout=minimumdist+1;
748                 }
749             }
750             else
751             {
752                 seuclidean_rout=minimumdist;
753                 sseuclidean_rout=minimumdist+1;
754             }
755             //first level energy
756
757             if ( euclidean_rout <= seuclidean_rout && euclidean_rout <=
758             sseuclidean_rout)
759             {
760                 e_elec= get_elec (packetsize);
761                 e_amp= get_eamp (packetsize , euclidean_rout);
762                 transmission_energy=e_elec + e_amp;
763
764                 printf("      B[%d][%d] : euclidean= %f\n", i , j , euclidean_rout);
765                 printf("      transmission energy = %.8f\n",
766                 transmission_energy);
767
768                 B[i][j]. energy=B[i][j]. energy-transmission_energy;
769                 check_state(i , j);
770
771                 printf("      receive energy = %.8f\n", e_elec);
772
773                 B[head_i][head_j]. energy=B[head_i][head_j]. energy-e_elec;
774                 state_on_receive(head_i , head_j);
775             }
776             //inturfnode

```

## Α.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

```

776     else if (euclidean_rout > sseuclidean_rout && seuclidean_rout >
777             sseuclidean_rout)
778     {
779         e_elec=get_elec(packetsize);
780         e_amp= get_eamp(packetsize ,sseuclidean_rout);
781         transmission_energy=e_elec + e_amp;
782
783         printf("    inturfnode  B[%d][%d] : euclidean= %f\n",i,j,
784             sseuclidean_rout);
785         printf("                transmission energy = %.8f\n",
786             transmission_energy);
787
788         B[i][j].energy=B[i][j].energy-transmission_energy;
789         check_state(i,j);
790
791         printf("                receive energy = %.8f\n",e_elec);
792
793         B[turf_i][turf_j].energy=B[turf_i][turf_j].energy-e_elec;
794         state_on_receive(turf_i ,turf_j);
795     }
796 //second head
797 else
798 {
799     e_elec= get_elec(packetsize);
800     e_amp= get_eamp(packetsize ,seuclidean_rout);
801     transmission_energy=e_elec + e_amp;
802
803     printf("    secondary B[%d][%d] : euclidean= %f\n",i,j,
804         seuclidean_rout);
805     printf("                transmission energy = %.8f\n",
806         transmission_energy);
807
808     B[i][j].energy=B[i][j].energy-transmission_energy;
809     check_state(i,j);
810
811     printf("                receive energy = %.8f\n",e_elec);
812
813     B[shead_i][shead_j].energy=B[shead_i][shead_j].energy-e_elec;
814     state_on_receive(shead_i ,shead_j);
815 }
816 }
817 euclidean_rout=1;
818 seuclidean_rout=1;
819 sseuclidean_rout=1;
820 }
821 //calculate heads
822 for (i=1;i<=max_i;i++)
823 {
824     for (j=1;j<=max_j;j++)

```

```

825     {
826     if (B[i][j]. clust==chk_node_num && (B[i][j]. head==1 || B[i][j].
head==extrahead))
827     {
828     mineucA_i=i;
829     mineucA_j=j;
830     mineuc_i=x2;
831     mineuc_j=y2;
832     if (B[i][j]. head==1)
833     {
834     euclidean_rout6=get_euclidean(x2,y2,i,j);
835     mineu6_i=i;
836     mineu6_j=j;
837     }
838     if (B[i][j]. head==2)
839     {
840     euclidean_rout7=get_euclidean(x2,y2,i,j);
841     mineu7_i=i;
842     mineu7_j=j;
843     }
844     }
845
846     }
847     }
848     if (euclidean_rout7==0)
849     {
850     euclidean_rout7=minimumdist;
851     }
852
853     if (euclidean_rout6<euclidean_rout7)
854     {
855     mineucA_i=mineu6_i;
856     mineucA_j=mineu6_j;
857     }
858     else
859     {
860     mineucA_i=mineu7_i;
861     mineucA_j=mineu7_j;
862     }
863
864     euclidean_rout1=get_euclidean(mineucA_i,mineucA_j,x2,y2);
865     bseuc=euclidean_rout1;
866
867     euclidean_rout6=1;
868     euclidean_rout7=0;
869
870     for (i=1;i<=max_i;i++)
871     {
872     for (j=1;j<=max_j;j++)
873     {
874     if (B[i][j]. clust!=chk_node_num && (B[i][j]. head==1 || B[i][j].
head==extrahead))
875     {
876

```

## Α.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

```

877     euclidean_rout2=get_euclidean(x2,y2,i,j);
878
879     euclidean_rout3=get_euclidean(mineucA_i,mineucA_j,i,j);
880
881     if (euclidean_rout2 < euclidean_rout1)
882     {
883         if (euclidean_rout3 < adv_min)
884         {
885             mineuc_i=i;
886             mineuc_j=j;
887             adv_min=euclidean_rout3;
888         }
889     }
890     euclidean_rout2=1;
891     euclidean_rout3=1;
892 }
893 }
894 }
895
896 bseucA=get_euclidean(mineucA_i,mineucA_j,mineuc_i,mineuc_j);
897
898 if (bseuc<bseucA)
899 {
900     mineuc_i=x2;
901     mineuc_j=y2;
902 }
903
904 bseuc=1;
905 bseucA=1;
906
907 mineuclidean=get_euclidean(mineucA_i,mineucA_j,mineuc_i,mineuc_j);
908
909 if (B[mineucA_i][mineucA_j].head==2)
910 {
911     swap(&head_i,&head_j,&shead_i,&shead_j);
912 }
913
914 if (shead_i != 0 && shead_j != 0)
915 {
916     if (B[turf_i][turf_j].head != B[shead_i][shead_j].head)
917     {
918
919         euclidean_rout4=get_euclidean(turf_i,turf_j,shead_i,shead_j);
920
921         e_elec= get_elec(packetsize);
922         fusion= get_fusion(packetsize);
923         e_amp= get_eamp(packetsize,euclidean_rout4);
924         transmission_energy=e_elec + e_amp + fusion;
925         printf("head[%d][%d] to head[%d][%d] : euclidean= %f\n",shead_i,
shead_j,turf_i,turf_j,euclidean_rout4);
926         printf("    transmission energy = %.8f\n",transmission_energy);
927
928         B[shead_i][shead_j].energy=B[shead_i][shead_j].energy-
transmission_energy;

```

```

929     check_state(shead_i , shead_j);
930
931     printf("    receive energy = %.8f\n", e_elec);
932
933     B[turf_i][turf_j].energy=B[turf_i][turf_j].energy-e_elec;
934     state_on_receive(turf_i , turf_j);
935
936     euclidean_rout4=1;
937 //third
938 }
939
940 if (B[turf_i][turf_j].head != B[head_i][head_j].head)
941 {
942
943     euclidean_rout5=get_euclidean(turf_i , turf_j , head_i , head_j);
944
945     e_elec= get_elec(packetsize);
946     fusion= get_fusion(packetsize);
947     e_amp= get_eamp(packetsize , euclidean_rout5);
948     transmission_energy=e_elec + e_amp + fusion;
949     printf("head[%d][%d] to head[%d][%d] : euclidean= %f\n", turf_i ,
950 turf_j , head_i , head_j , euclidean_rout5);
951     printf("    transmission energy = %.8f\n", transmission_energy);
952
953     B[turf_i][turf_j].energy=B[turf_i][turf_j].energy-
954 transmission_energy;
955     check_state(turf_i , turf_j);
956
957     printf("    receive energy = %.8f\n", e_elec);
958
959     B[head_i][head_j].energy=B[head_i][head_j].energy-e_elec;
960     state_on_receive(head_i , head_j);
961
962     euclidean_rout5=1;
963 }
964 }
965 //heads
966 //second level energy
967
968 e_elec= get_elec(packetsize);
969 fusion= get_fusion(packetsize);
970 e_amp= get_eamp(packetsize , mineuclidean);
971 transmission_energy=e_elec + e_amp + fusion;
972 printf("\nB[%d][%d] to B[%d][%d] : min_euclidean= %f \n\n", mineucA_i ,
973 mineucA_j , mineuc_i , mineuc_j , mineuclidean);
974
975 printf("    transmission energy = %.8f\n", transmission_energy);
976
977 B[mineucA_i][mineucA_j].energy=B[mineucA_i][mineucA_j].energy-
978 transmission_energy;
979 check_state(mineucA_i , mineucA_j);

```

## Α.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

```

979     printf("         receive energy = %.8f\n", e_elec);
980     if (mineuc_i != x2 && mineuc_j != y2)
981     {
982         B[mineuc_i][mineuc_j].energy=B[mineuc_i][mineuc_j].energy-e_elec;
983         check_state(mineuc_i,mineuc_j);
984     }
985
986     if (mineuc_i==x2 && mineuc_j==y2)
987     {
988         rpointer = newitem (rpointer , B[mineucA_i][mineucA_j].clust , bsclust
989         );
990     }
991     else
992     {
993         rpointer = newitem (rpointer , B[mineucA_i][mineucA_j].clust , B[
994         mineuc_i][mineuc_j].clust);
995     }
996
997     headiszero ;
998     chk_node_num++;
999     mineuclidean=1;
1000 }
1001 if (z<max_sensors)
1002 {
1003     extrahead=2;
1004 }
1005
1006 network_rounds++;
1007 chk_node_num=0;
1008 viewSchedule (rpointer);
1009 while (rpointer != NULL)
1010 {
1011     rpointer = DelSchedule (rpointer);
1012 }
1013 //getchar ();
1014 }
1015 printf("\n\n");
1016 printf("\n\nNetwork Rounds : %d\n", network_rounds-1);
1017 printf("\n\nz : %d\n", z);
1018 printf("\n\n");
1019 }
1020 }
1021
1022
1023 void swap(int *a, int *b, int *c, int *d)
1024 {
1025     int temp1 = *a;
1026     *a = *c;
1027     *c = temp1;
1028
1029     int temp2 = *b;
1030     *b = *d;

```

```

1031     *d = temp2;
1032 }
1033
1034 double get_elec(int packet)
1035 {
1036     double en_elec=0;
1037
1038     en_elec=(1/(double)1000000000)*50*packet;
1039     return en_elec;
1040 }
1041
1042 double get_fusion(int packet_f)
1043 {
1044     double en_fusion=0;
1045
1046     en_fusion=(1/(double)1000000000)*5*packet_f;
1047     return en_fusion;
1048 }
1049
1050 double get_eamp(int packet_amp, double dist)
1051 {
1052     double en_eamp=0;
1053
1054     en_eamp=(1/(double)1000000000000)*100*packet_amp*(dist*dist);
1055     return en_eamp;
1056 }
1057
1058 double get_euclidean(int xA, int yA, int xB, int yB)
1059 {
1060     double euc_par=0;
1061     double distance=1;
1062     double com_par=0;
1063
1064     euc_par=((xA-xB)*(xA-xB))+((yA-yB)*(yA-yB));
1065     while (com_par!=distance)
1066     {
1067         com_par=distance;
1068         distance=0.5*(com_par+(euc_par/com_par));
1069     }
1070
1071     return distance;
1072 }
1073 }
1074
1075 void check_state(int x1, int y1)
1076 {
1077     if (B[x1][y1].energy <= 0 && B[x1][y1].clust!=k)
1078     {
1079         B[x1][y1].wcounter--;
1080         B[x1][y1].clust=k;
1081         B[x1][y1].energy=0;
1082         z--;
1083         write_rounds(z, network_rounds);
1084     }

```



```

1085     else
1086     {
1087         B[x1][y1].wcounter++;
1088     }
1089 }
1090
1091 void state_on_receive(int xx1, int yy1)
1092 {
1093     if (B[xx1][yy1].energy <= 0 && B[xx1][yy1].clust!=k)
1094     {
1095         B[xx1][yy1].wcounter--;
1096         B[xx1][yy1].energy=0;
1097     }
1098 }
1099 else
1100 {
1101     B[xx1][yy1].wcounter++;
1102 }
1103 }
1104
1105 void clear_records()
1106 {
1107     char text[20]="round_record";
1108
1109     if(remove(text) != 0)
1110     {
1111         fprintf(stderr, "Error deleting the file %s.\n", text);
1112     }
1113 }
1114 }
1115
1116 void siminfo()
1117 {
1118     FILE *fp;
1119
1120     time_t get_time;
1121     char* time_string;
1122     char error[40]="Error: simulation date/time undefined";
1123
1124     get_time = time(NULL);
1125     time_string = ctime(&get_time);
1126
1127     if (time_string == NULL)
1128     {
1129         time_string=error;
1130     }
1131
1132     fp = fopen ("round_record", "a");
1133     fprintf(fp, "Network size is %dx%d\nNetwork nodes are %d\nBase Station
        position is [%d][%d]\nInitial energy %f J\nSimulation date/time: %s\
        n\n", maximum_i, maximum_j, msensors, bspos_x, bspos_y, simenergy,
        time_string);
1134
1135     fclose (fp);

```

```

1136 }
1137 }
1138
1139 void write_rounds(int sens, int wrounds)
1140 {
1141     FILE *fp;
1142
1143     fp = fopen ("round_record", "a");
1144     fprintf(fp, "Nodes are %d at round %d\n", sens, wrounds);
1145     fclose (fp);
1146 }
1147
1148 schnode *newitem (schnode *rpointer, int clust1, int clust2)
1149 {
1150     schnode *fpa = rpointer;
1151
1152     if (rpointer != NULL)
1153     {
1154         while (rpointer -> right != NULL)
1155         {
1156             rpointer = rpointer -> right;
1157         }
1158
1159         rpointer -> right = (struct schnode *) malloc (sizeof (schnode));
1160         rpointer = rpointer -> right;
1161         rpointer -> right = NULL;
1162         rpointer -> cluster1 = clust1;
1163         rpointer -> cluster2 = clust2;
1164         return fpa;
1165     }
1166     else
1167     {
1168         rpointer = (struct schnode *) malloc (sizeof (schnode));
1169         rpointer -> right = NULL;
1170         rpointer -> cluster1 = clust1;
1171         rpointer -> cluster2 = clust2;
1172         return rpointer;
1173     }
1174 }
1175 }
1176 }
1177
1178 void viewSchedule (schnode *rpointer)
1179 {
1180     if (rpointer == NULL)
1181         printf ("no schedule!\n");
1182     else
1183     {
1184         printf("Cluster Schedule\nError code: %d (no energy)\n\n", knode);
1185         printf("From clr | To clr\n");
1186         while (rpointer != NULL)
1187         {
1188             printf ("%5d", rpointer -> cluster1);
1189             if (rpointer -> cluster2 == bsclust)

```

A'.2. Κώδικας Μεθόδου Εξοικονόμησης Ενέργειας

```
1190     {
1191         printf ("      BS\n");
1192     }
1193     else
1194     {
1195         printf ("%9d\n", rpointer -> cluster2);
1196     }
1197     rpointer = rpointer -> right;
1198 }
1199 printf ("\n");
1200 }
1201
1202 }
1203 }
1204
1205 schnode *DelSchedule (schnode *rpointer)
1206 {
1207     schnode *temp;
1208     temp = rpointer -> right;
1209     free (rpointer);
1210     return temp;
1211 }
```

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑ

Πανεπιστήμιο Πειραιώς

- [1] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM.
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Comput. Netw.*, 38(4):393–422, March 2002.
- [3] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52(12):2292–2330, August 2008.
- [4] S. Ehsan and B. Hamdaoui. A survey on energy-efficient routing techniques with qos assurances for wireless multimedia sensor networks. *Communications Surveys Tutorials, IEEE*, 14(2):265–278, Second 2012.
- [5] A. Gkikopouli, G. Nikolakopoulos, and S. Manesis. A survey on underwater wireless sensor networks and applications. In *Control Automation (MED), 2012 20th Mediterranean Conference on*, pages 1147–1154, July 2012.
- [6] M. Johnson, M. Healy, P. van de Ven, M.J. Hayes, J. Nelson, T. Newe, and E. Lewis. A comparative review of wireless sensor network mote technologies. In *Sensors, 2009 IEEE*, pages 1439–1442, Oct 2009.
- [7] S. Gajjar, N. Choksi, M. Sarkar, and K. Dasgupta. Comparative analysis of wireless sensor network motes. In *Signal Processing and Integrated Networks (SPIN), 2014 International Conference on*, pages 426–431, Feb 2014.
- [8] Yun Wang, Xiaodong Wang, Bin Xie, Demin Wang, and D.P. Agrawal. Intrusion detection in homogeneous and heterogeneous wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 7(6):698–711, June 2008.

- [9] I. Butun, S.D. Morgera, and R. Sankar. A survey of intrusion detection systems in wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 16(1):266–282, First 2014.
- [10] Liyang Yu, Neng Wang, and Xiaoqiao Meng. Real-time forest fire detection with wireless sensor networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, volume 2, pages 1214–1217, Sept 2005.
- [11] Santosh Bhima, Anil Gogada, and Rammurthy Garimella. A tsunami warning system employing level controlled gossiping in wireless sensor networks. In *Proceedings of the 4th International Conference on Distributed Computing and Internet Technology, ICDCIT'07*, pages 306–313, Berlin, Heidelberg, 2007. Springer-Verlag.
- [12] Chien-Hao Wang, Yu-Kai Huang, Xiang-Yao Zheng, Tzu-Shiang Lin, Cheng-Long Chuang, and Joe-Air Jiang. A self sustainable air quality monitoring system using wsn. In *Service-Oriented Computing and Applications (SOCA), 2012 5th IEEE International Conference on*, pages 1–6, Dec 2012.
- [13] M.A. Razzaque, M. Akhtaruzzaman Adnan, and A.H. Abdullah. Energy efficient gas emission monitoring systems using wireless sensor networks. In *Ubiquitous and Future Networks (ICUFN), 2013 Fifth International Conference on*, pages 588–593, July 2013.
- [14] Shi Lan, Miao Qilong, and Jinglin Du. Architecture of wireless sensor networks for environmental monitoring. In *Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing. ETT and GRS 2008. International Workshop on*, volume 1, pages 579–582, Dec 2008.
- [15] Jeongyeup Paek, K. Chintalapudi, R. Govindan, J. Caffrey, and S. Masri. A wireless sensor network for structural health monitoring: Performance and experience. In *Embedded Networked Sensors, 2005. EmNetS-II. The Second IEEE Workshop on*, pages 1–10, May 2005.
- [16] H. Furtado and R. Trobec. Applications of wireless sensors in medicine. In *MIPRO, 2011 Proceedings of the 34th International Convention*, pages 257–261, May 2011.
- [17] S. Dengler, A. Awad, and F. Dressler. Sensor/actuator networks in smart homes for supporting elderly and handicapped people. In *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, volume 2, pages 863–868, May 2007.
- [18] J. Burrell, T. Brooke, and R. Beckwith. Vineyard computing: sensor networks in agricultural production. *Pervasive Computing, IEEE*, 3(1):38–45, Jan 2004.

- [19] Wei Dong, Chun Chen, Xue Liu, and Jiajun Bu. Providing os support for wireless sensor networks: Challenges and approaches. *Communications Surveys Tutorials, IEEE*, 12(4):519–530, Fourth 2010.
- [20] G.S. Ramachandran, S. Michiels, W. Joosen, D. Hughes, and B. Porter. Analysis of sensor network operating system performance throughout the software life cycle. In *Network Computing and Applications (NCA), 2013 12th IEEE International Symposium on*, pages 211–218, Aug 2013.
- [21] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, and Andrea Passarella. Energy conservation in wireless sensor networks: A survey. *Ad Hoc Netw.*, 7(3):537–568, May 2009.
- [22] R. Marfievici, A.L. Murphy, G.P. Picco, F. Ossi, and F. Cagnacci. How environmental factors impact outdoor wireless sensor networks: A case study. In *Mobile Ad-Hoc and Sensor Systems (MASS), 2013 IEEE 10th International Conference on*, pages 565–573, Oct 2013.
- [23] N.A. Pantazis and D.D. Vergados. A survey on power control issues in wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 9(4):86–107, Fourth 2007.
- [24] B. Kaur and S. Kaushal. Qos based evaluation of routing protocols in wsn. In *Engineering and Computational Sciences (RAECS), 2014 Recent Advances in*, pages 1–7, March 2014.
- [25] H.R. Karkvandi, E. Pecht, and O. Yadid-Pecht. Performance evaluation of lifetime-aware routing in wireless sensor networks with practical design considerations. In *Electrical Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, pages 1–4, April 2012.
- [26] P. Suriyachai, U. Roedig, and A. Scott. A survey of mac protocols for mission-critical applications in wireless sensor networks. *Communications Surveys Tutorials, IEEE*, 14(2):240–264, Second 2012.
- [27] N.A. Pantazis, S.A. Nikolidakis, and D.D. Vergados. Energy-efficient routing protocols in wireless sensor networks: A survey. *Communications Surveys Tutorials, IEEE*, 15(2):551–591, Second 2013.
- [28] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*, pages 10 pp. vol.2–, Jan 2000.
- [29] Stefanos A. Nikolidakis, Dionisis Kandris, Dimitrios D. Vergados, and Christos Douligeris. Energy efficient routing in wireless sensor networks through balanced clustering. *Algorithms*, 6(1):29–42, 2013.

- [30] S. Lindsey, C. Raghavendra, and K.M. Sivalingam. Data gathering algorithms in sensor networks using energy metrics. *Parallel and Distributed Systems, IEEE Transactions on*, 13(9):924–935, Sep 2002.
- [31] David Lee Hall. *Mathematical Techniques in Multisensor Data Fusion*. Artech House, Inc., Norwood, MA, USA, 1992.
- [32] Marwah M Almasri and Khaled M Elleithy. Data fusion models in wsns: Comparison and analysis. In *American Society for Engineering Education (ASEE Zone 1), 2014 Zone 1 Conference of the*, pages 1–6, April 2014.
- [33] S. Lindsey and C.S. Raghavendra. Pegasus: Power-efficient gathering in sensor information systems. In *Aerospace Conference Proceedings, 2002. IEEE*, volume 3, pages 3–1125–3–1130 vol.3, 2002.
- [34] S.D. Muruganathan, D.C.F. Ma, R.I. Bhasin, and A. Fapojuwo. A centralized energy-efficient routing protocol for wireless sensor networks. *Communications Magazine, IEEE*, 43(3):S8–13, March 2005.
- [35] Arati Manjeshwar and D.P. Agrawal. Teen: a routing protocol for enhanced efficiency in wireless sensor networks. In *Parallel and Distributed Processing Symposium., Proceedings 15th International*, pages 2009–2015, April 2001.
- [36] Yang Yang, Hui-Hai Wu, and Hsiao-Hwa Chen. Short: Shortest hop routing tree for wireless sensor networks. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 8, pages 3450–3454, June 2006.
- [37] Q. Nadeem, M.B. Rasheed, N. Javaid, Z.A. Khan, Y. Maqsood, and A. Din. M-gear: Gateway-based energy-aware multi-hop routing protocol for wsns. In *Broadband and Wireless Computing, Communication and Applications (BWCCA), 2013 Eighth International Conference on*, pages 164–169, Oct 2013.
- [38] Wenjing Guo, Wei Zhang, and Gang Lu. Pegasus protocol in wireless sensor network based on an improved ant colony algorithm. In *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on*, volume 3, pages 64–67, March 2010.
- [39] R. A. Roseline and P. Sumathi. Local clustering and threshold sensitive routing algorithm for wireless sensor networks. In *Devices, Circuits and Systems (ICDCS), 2012 International Conference on*, pages 365–369, March 2012.
- [40] Arati Manjeshwar and Dharma P. Agrawal. Apteen: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In *Proceedings of the 16th International Parallel and Distributed Processing Symposium, IPDPS '02*, pages 48–, Washington, DC, USA, 2002. IEEE Computer Society.



- [41] T.M. Rahayu, Sang-Gon Lee, and Hoon-Jae Lee. Survey on leach-based security protocols. In *Advanced Communication Technology (ICACT), 2014 16th International Conference on*, pages 304–309, Feb 2014.
- [42] M. Aslam, N. Javaid, A. Rahim, U. Nazir, A. Bibi, and Z. A. Khan. Survey of extended leach-based clustering routing protocols for wireless sensor networks. In *High Performance Computing and Communication 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESSE), 2012 IEEE 14th International Conference on*, pages 1232–1238, June 2012.
- [43] W.B. Heinzelman, A.P. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4):660–670, Oct 2002.

Πανεπιστήμιο Πειραιώς

Πανεπιστήμιο Πειραιώς

- αναλογία, 20  
άπληστος αλγόριθμος, 62  
αποδοτικότητα, 3
- βιωσιμότητα, 15
- διατάξεις, 7, 12
- εμβέλεια μετάδοσης, 23  
ενδιάμεσων κόμβων, 8  
ενεργειακά αποθέματα, 42  
επίπεδο υλικού, 7  
ευκλείδεια απόσταση, 17
- ζητήματα λειτουργίας, 46
- θεωρητική παρουσίαση, 41
- ιδιότητες των κόμβων, 15  
ισορροπία, 17, 54
- καθολική γνώση, 27  
κατευθυνόμενο άκυκλο μονοπάτι, 28  
κόστος υλοποίησης, 3
- κύκλοι δρομολόγησης, 32  
κύριες διαδικασίες, 33, 39
- λειτουργικό σύστημα, 2
- μονοπάτια χαμηλού κόστους, 30
- ξεχωριστά υποδίκτυα, 20
- ομαλή λειτουργία, 19  
ομοιόμορφη κατανομή, 12, 24  
ορθότητα της δρομολόγησης, 32
- πολυεπίπεδης διαδικασίας, 61  
προσαρμοστική συσταδοποίηση, 10
- σημεία κλειδιά, 62  
στρατηγικές θέσεις, 16  
συσκευές, 1
- τοπολογία, 3, 10
- υπολογιστικό κόστος, 9
- χαμηλή πολυπλοκότητα, 2