

Ατομική Διπλωματική Εργασία

**ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ ΗΤΑΡ
ΓΙΑ ΕΛΕΓΧΟ ΣΥΜΦΟΡΗΣΗΣ ΣΕ ΑΣΥΡΜΑΤΑ ΔΙΚΥΑ
ΑΙΣΘΗΤΗΡΩΝ**

Μαρίνος Στυλιανού

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2011

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

**Υλοποίηση και Αξιολόγηση του Αλγόριθμου HTAP για Έλεγχο Συμφόρησης σε
Ασύρματα Δίκτυα Αισθητήρων**

Μαρίνος Στυλιανού

Επιβλέπων Καθηγητής

Βάσος Βασιλείου

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

Μάιος 2011

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Βάσο Βασιλείου για την καθοδήγηση και την βοήθεια που μου παρείχε κατά τη διάρκεια της υλοποίησης αυτής της εργασίας. Επίσης ευχαριστώ για την πολύτιμη βοήθεια τους διδακτορικούς φοιτητές Χριστιάνα Ιωάννου, Ζήνωνα Ζήνωνος και Μάριο Κούτρουλλο.

Ακόμη θέλω να ευχαριστήσω μερικούς από τους συμφοιτητές μου για την συμπαράσταση τους και τέλος θέλω να ευχαριστήσω την οικογένεια μου που με στήριξε όλο αυτό το καιρό κατά τη διάρκεια των σπουδών μου .

Περίληψη

Τα Ασύρματα Δίκτυα Αισθητήρων με την ραγδαία ανάπτυξη της τεχνολογίας γνωρίζουν μεγάλη άνθιση και ανάπτυξη αυτή έχει σαν αποτέλεσμα και την αυξημένη παρουσία τους αλλά και μελέτη σε ερευνητικό επίπεδο. Στόχος αυτής της εργασίας είναι να παρουσιάσει σε πρώτο στάδιο τα Ασύρματα Δίκτυα Αισθητήρων και κατά δεύτερο από ερευνητικής άποψης να ασχοληθεί με την μελέτη αλγορίθμων που είναι σχεδιασμένοι για να αντιμετωπίζουν το πρόβλημα της συμφόρησης στα δίκτυα αυτά.

Πιο συγκεκριμένα θα παρουσιαστεί ο αλγόριθμος HTAP (Hierarchical Tree Alternative Path) που έχει σαν στόχο την βελτίωση της απόδοσης των δικτύων από πλευράς συμφόρησης αλλά και διαχείρισης ενέργειας. Επίσης γίνεται μια αναφορά στο λειτουργικό που χρησιμοποιήθηκε για την υλοποίηση αυτού του αλγορίθμου και στον τρόπο λειτουργίας του. Τέλος παρουσιάζονται τα αποτελέσματα προσομοιώσεων του αλγορίθμου αυτού και εξάγονται κάποια συμπεράσματα με βάση τα αποτελέσματα.

Περιεχόμενα

Κεφάλαιο 1 Εισαγωγή

1.1 Γενική αναφορά και ο σκοπός της διπλωματικής εργασίας.....	1
1.2 Δομή της διπλωματικής Εργασίας	2

Κεφάλαιο 2 Υπόβαθρο για τα Ασύρματα Δίκτυα Αισθητήρων

2.1 Γενικά για Ασύρματα Δίκτυα Αισθητήρων	3
2.2 Κόμβοι Αισθητήρων	4
2.3 Τρόποι Διάταξης Αισθητήρων	7
2.4 Πρότυπα Επικοινωνίας.....	8
2.5 Εφαρμογές των ασύρματων δικτύων αισθητήρων.....	9
2.6 Προβλήματα-Προκλήσεις στα Ασύρματα Δίκτυα Αισθητήρων.....	10

Κεφάλαιο 3 Έλεγχος Συμφόρησης

3.1 Γενικά.....	13
3.2 Έλεγχος Συμφόρησης στα Ενσύρματα και Ασύρματα Δίκτυα	13
3.3 Έλεγχος Συμφόρησης στα Ασύρματα Δίκτυα Αισθητήρων.....	14
3.4 Λόγοι Δημιουργίας και οι Επιπτώσεις της Συμφόρησης	16
3.5 Τύποι Συμφόρησης.....	19

Κεφάλαιο 4 Αντιμετώπιση Συμφόρησης

4.1 Γενικά.....	21
4.2 Πρωτόκολλα για Αντιμετώπιση της Συμφόρησης	21
4.3 Περιγραφή Πρωτόκολλων	22
4.3.1 Congestion Detection and Avoidance (CODA).....	22
4.3.2 Mitigating Congestion (Fusion).....	24
4.3.3 SenTCP	25
4.3.4 HTAP (Hierarchical Tree Alternative Path)	26

Κεφάλαιο 5 Contiki Λειτουργικό Σύστημα για Ασύρματα Δίκτυα Αισθητήρων

5.1 Γενικά.....	29
-----------------	----

5.2 Τρόποι Επικοινωνίας.....	30
5.3 RDC Drivers.....	33
5.4 MAC Drivers.....	34
5.5 Event Timers	35
Κεφάλαιο 6 Υλοποίηση αλγορίθμου HTAP σε Contiki	
6.1 Γενικά για υλοποίηση	36
6.2 Φάση πρώτη: Ανακάλυψη δικτύου και καθορισμός επιπέδων	40
6.3 Φάση δεύτερη: Υλοποίηση ιεραρχικού δένδρου	43
6.4 Φάση τρίτη: Αποστολή δεδομένων και εύρεση εναλλακτικού μονοπατιού	45
Κεφάλαιο 7 Πειραματική αξιολόγηση Αλγορίθμου	
7.1 Γενικά.....	47
7.2 Περιβάλλον Προσομοίωσης.....	48
7.3 Με βάση την πυκνότητα των κόμβων.....	50
7.4 Με σταθερό αριθμό κόμβων αλλά με περισσότερα Source's	53
7.5 Με διαφορετικές τοπολογίες.....	57
7.6 Προβλήματα του αλγορίθμου HTAP	58
Κεφάλαιο 8 Συμπεράσματα	
8.1 Γενική σύνοψη	61
8.2 Συμπεράσματα	62

Κεφάλαιο 1

Εισαγωγή

1.1 Γενική αναφορά και ο σκοπός της διπλωματικής εργασίας

1.2 Δομή της διπλωματικής Εργασίας

1.1 Γενική αναφορά και ο σκοπός της διπλωματικής εργασίας

Αυτή η διπλωματική εργασία έχει σαν θέμα τα Ασύρματα Δίκτυα Αισθητήρων και ένα από τα προβλήματα τα οποία αντιμετωπίζουν σαν τεχνολογία τα δίκτυα αυτά. Στα πλαίσια αυτής της διπλωματικής εργασίας θα γίνει μία εκτενής αναφορά τόσο στα Ασύρματα Δίκτυα Αισθητήρων όσο και στο πρόβλημα το οποίο πρέπει να αντιμετωπιστεί.

Για την αντιμετώπιση του προβλήματος αυτού που είναι η συμφόρηση κλήθηκα να υλοποιήσω ένα αλγόριθμο, και με βάση αυτήν την υλοποίηση να τον αξιολογήσω με την βοήθεια κάποιων εργαλείων. Τα εργαλεία αυτά και το λειτουργικό που χρησιμοποιήθηκε για την υλοποίηση του αλγορίθμου είναι αυτά τα οποία στην πραγματικότητα μπορούν να τρέξουν στους κόμβους έτσι είναι μία ποιά αντικειμενική προσέγγιση σε σχέση με υλοποίηση σε άλλες γλώσσες προγραμματισμού ή την χρήση άλλων εργαλείων.

Με την αξιολόγηση αυτή κρίθηκε η ικανότητα του αλγορίθμου να αντιμετωπίσει το πρόβλημα της συμφόρησης για το οποίο σχεδιάστηκε καθώς και η λειτουργικότητα του ίδιου του αλγορίθμου. Σκοπός είναι η υλοποίηση του αλγορίθμου HTAP που έχει σαν στόχο να γίνει η αξιολόγηση του μέσα από ένα περιβάλλον προσομοιώσεων. Έτσι μπορούν να βρεθούν βελτιώσεις στον τρόπο λειτουργίας του αλγορίθμου να αντιμετωπιστούν λάθη που τυχόν αντιμετωπίζει και δεν εντοπίζεται μέχρι τώρα καθώς και να γίνουν πιθανές βελτιώσεις για καλύτερη απόδοση στο πρόβλημα της συμφόρησης.

Γενικά η εργασία αυτή περιστρέφεται γύρω από τα Ασύρματα Δίκτυα Αισθητήρων , το πρόβλημα συμφόρησης που αντιμετωπίζουν και την αξιολόγηση ενός πιθανού τρόπου αντιμετώπισης του που είναι ο αλγόριθμος HTAP.

1.2 Δομή της διπλωματικής Εργασίας

Η δομή που ακολουθήθηκε για την ανάπτυξη της διπλωματικής εργασίας αυτή είναι πολύ κατατοπιστική αφού πριν αναφερθεί κάτι πρώτα επεξηγείται και μετά παρουσιάζεται. Έτσι η σειρά με την οποία τα θέματα παρουσιάζονται στην εργασία αυτή έχει ως εξής.

Αρχικά γίνεται μία γενική αναφορά στα δίκτυα αισθητήρων στα δομικά τους στοιχεία. Επεξηγείται ο τρόπος λειτουργίας τους οι εφαρμογές που μπορούν να έχουν αλλά και τα προβλήματα που αντιμετωπίζουν. Στη συνέχεια παρουσιάζεται ποιο αναλυτικά το πρόβλημα της συμφόρησης στα Ασύρματα Δίκτυα Αισθητήρων, οι λόγοι για τους οποίους εμφανίζεται οι συμφόρηση αλλά και τί επιπτώσεις έχει η συμφόρηση στην τεχνολογία αυτή.

Μετά ακολουθεί η παρουσίαση και οι επεξήγηση τρόπων με τους οποίους μπορεί να αντιμετωπιστεί η συμφόρηση αλλά και να γίνει έλεγχος για αποφυγή της. Σε αυτό το σημείο γίνεται και η αναφορά του HTAP αλγορίθμου ο οποίος είναι και ο αλγόριθμος ο οποίος υλοποιείται και αξιολογείται στα πλαίσια αυτή της διπλωματικής. Στην συνέχεια αφού γίνεται η παρουσίαση του αλγόριθμου αλλά και η επεξήγηση του τρόπου λειτουργίας του γίνεται μία αναφορά στο λειτουργικό σύστημα στο οποίο υλοποιείτε ο αλγόριθμος. Με την επεξήγηση του περιβάλλοντος αυτού και του τρόπου λειτουργίας του λειτουργικού CONTIKI μπορεί μετά να γίνει ποιο κατανοητή η επεξήγηση της υλοποίησης που έγινε για τον αλγόριθμο.

Τέλος αφού γίνει η επεξήγηση της υλοποίησης παρουσιάζεται ο τρόπος και τα σενάρια με τα οποία γίνεται η αξιολόγηση του αλγορίθμου HTAP καθώς και τα συμπεράσματα που βγαίνουν για τον αλγόριθμο και την λειτουργικότητα του αλλά και τα γενικά συμπεράσματα τα οποία αφορούν την τεχνολογία των ασύρματων δικτύων αισθητήρων.

Κεφάλαιο 2

Υπόβαθρο για τα Ασύρματα Δίκτυα Αισθητήρων

2.1 Γενικά για Ασύρματα Δίκτυα Αισθητήρων

2.2 Κόμβοι Αισθητήρων

2.3 Τρόποι Διάταξης Αισθητήρων-Τοπολογία

2.4 Εφαρμογές των ασύρματων δικτύων αισθητήρων

2.5 Προβλήματα-Προκλήσεις στα Ασύρματα Δίκτυα Αισθητήρων

2.1 Γενικά για Ασύρματα Δίκτυα Αισθητήρων

Με την ραγδαία πρόοδο που έχουν οι ασύρματες επικοινωνίες και τα ηλεκτρονικά στις μέρες μας οδήγησε στην ανάπτυξη των χαμηλού κόστους ασύρματων αισθητήρων. Είναι αισθητήρες συνήθως μικροί σε μέγεθος που έχουν τη ικανότητα να ανιχνεύουν και να επεξεργάζονται δεδομένα ανάλογα με την λειτουργία που είναι αρμόδια να εκτελούν επίσης μπορούν να επικοινωνούν μεταξύ τους μέσω κάποιων ράδιο-δέκτων και να οργανώνονται σε δίκτυα αισθητήρων τα οποία ονομάζονται Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks, WSN).

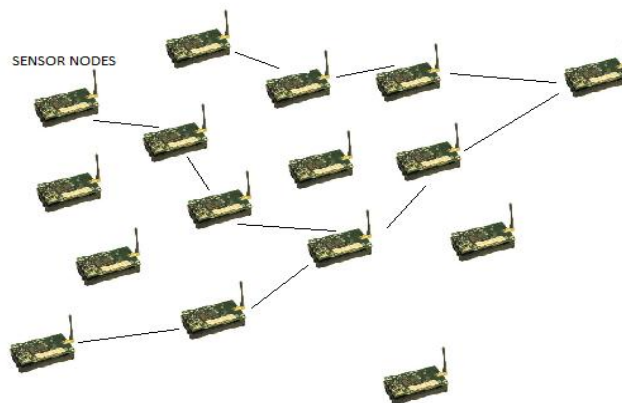
Τα δίκτυα αυτά αποτελούνται από λίγους ή ακόμα και από μεγάλο αριθμό κόμβων όπου κάθε κόμβος είναι ένας αισθητήρας σταθερός σε μια θέση είτε κινητός. Μερικά βασικά χαρακτηριστικά αυτού του τύπου δικτύων είναι :

- Ικανότητα να οργανώνονται μονοί τους σε δίκτυα.
- Αναμετάδοση πληροφοριών μέσω πολλών κόμβων μέχρι να φτάσουν στον τελικό αποδέκτη κόμβο
- Πυκνή διάταξη των κόμβων και συνεταιριστική προσπάθεια των κόμβων για να έχουν επιτυχή αποτελέσματα.
- Συχνά μεταβαλλόμενη τοπολογία λόγω των αποτυχιών ή εξασθένησης των κόμβων.

Τα πιο πάνω χαρακτηριστικά είναι αυτά τα οποία ξεχωρίζουν τα ασύρματα δίκτυα αισθητήρων από τα άλλα ασύρματα δίκτυα.

Τα κύρια όμως χαρακτηριστικά στα ασύρματα δίκτυα αισθητήρων είναι η περιορισμένη αποθηκευμένη ενέργεια που έχουν οι κόμβοι, η αντοχή σε αντίξοες συνθήκες, η ικανότητα να αντιμετωπίζουν την πιθανή διακοπή λειτουργίας κόμβων ανάλογα με το πώς είναι προγραμματισμένοι οι κόμβοι να το κάνουν, κινητικότητα των κόμβων σε κάποιες περιπτώσεις όπου απαιτείται, η δυναμική τοπολογία του δικτύου, ετερογένεια στους κόμβους, επέκταση του δικτύου σε μεγάλη κλίμακα και η αυτόνομη λειτουργία τους.

Μέσα από αυτά τα χαρακτηριστικά προκύπτουν όμως και κάποιες απαιτήσεις τις οποίες πρέπει να ικανοποιεί κάθε δίκτυο ασύρματων αισθητήρων και αυτές είναι η ακριβής και αξιόπιστη μεταφορά των δεδομένων μέσα από τους διάφορους κόμβους του δικτύου επίσης πρέπει να παρέχονται μηχανισμοί ασφαλείας για τους κόμβους του δικτύου οι οποίοι να είναι προσαρμοσμένοι ανάλογα με το περιβάλλον χρήσης κάθε κόμβου και τέλος αφού τα δίκτυα είναι αυτόνομα και βασίζονται σε μια περιορισμένη μονάδα ενέργειας χωρίς να έχουν συνεχή παροχή όλοι οι αλγόριθμοι και πρωτόκολλα τα οποία χρησιμοποιούνται να είναι οικονομικά όσον αφορά την κατανάλωση ενέργειας.



Σχημα 2.1 Ένα Δίκτυο Ασύρματων Αισθητήρων

2.2 Κόμβοι Αισθητήρων

Ένας κόμβος αισθητήρα συνδυάζει τις ικανότητες για τον υπολογισμό, την επικοινωνία και την αίσθηση. Ένας αισθητήρας μπορεί να τοποθετηθεί σε δύσκολα προσβάσιμες περιοχές ανάλογα με το τι θέλουμε να εξετάσουμε σε αυτές τις περιοχές και να συλλέγει πληροφορίες τις οποίες θα μεταδίδει μετά σε άλλους κόμβους όπου μπορεί να

γίνει επεξεργασία τους. Οι κόμβοι όπως και οι αισθητήρες που έχουν οι κόμβοι ποικίλουν για παράδειγμα ένας κόμβος μπορεί να έχει διάφορους αισθητήρες όπως αισθητήρες κίνησης (radar), αισθητήρες ήχου, φωτός υπέρυθρου ή ορατού, χημικούς αισθητήρες, αισθητήρες πίεσης, μαγνητικούς αισθητήρες και διάφορους άλλους αφού η τεχνολογία των αισθητήρων γνωρίζει μεγάλη ανάπτυξη.

Λόγο όμως του ότι οι αισθητήρες προσπαθούν να συνδυάσουν τις τρεις λειτουργίες που ανάφερα αρχικά η ανάπτυξη των κόμβων αισθητήρων επηρεάζεται από τους ποιο κάτω παράγοντες.

- Από την αύξηση της πολυπλοκότητας της συσκευής λόγω των μικροτσίπ που απαιτούνται για υπολογισμούς.
- Από τις υψηλής απόδοσης τεχνολογίες ασύρματης δικτύωσης.
- Από τον συνδυασμό της ψηφιακής επεξεργασίας σήματος και της συλλογής / καταγραφής των δεδομένων από των αισθητήρα.
- Από την πρόοδο στην ανάπτυξη των μικροηλεκτρονικών συστημάτων (MEMS).
- Από την διαθεσιμότητα των εργαλείων ανάπτυξης υψηλής απόδοσης [1].

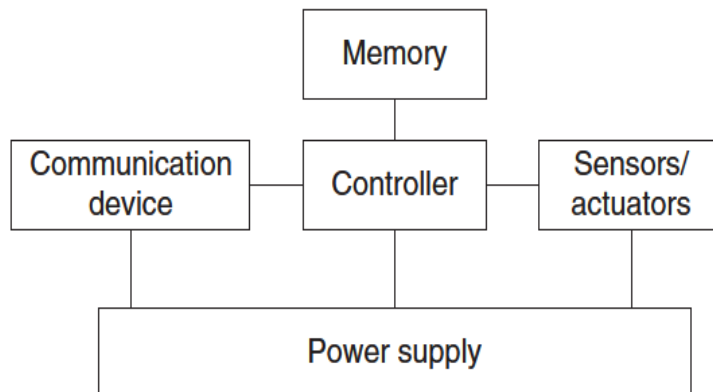
Οι κόμβοι αισθητήρες με βάση τις λειτουργίες που κάνουν αλλά και του τύπου τους χωρίζονται σε τρεις κατηγορίες

Παθητικοί, Προς όλες τις κατευθύνσεις αισθητήρες όπου μετρούν κάποιο μέγεθος με βάση τον αισθητήρα που έχουν χωρίς να επεμβαίνουν στο περιβάλλον .

Παθητικοί, Κατευθυνόμενοι αισθητήρες όπου πάλι χωρίς να επεμβαίνουν στο περιβάλλον η μέτρηση ή η παρατήρηση που κάνουν είναι προς μια κατεύθυνση ή μια συγκεκριμένη γωνία.

Ενεργοί αισθητήρες όπου πρέπει να επέμβουν στο περιβάλλον για να πάρουν κάποια μέτρηση.

Ποιο κάτω μια εικόνα με την τυπική δομή ενός αισθητήρα.



Σχήμα 2.2: Η τυπική δομή ενός κόμβου αισθητήρα [5]

Τα κυρία δομικά στοιχεία ενός κόμβου όπως φαίνονται στη εικόνα είναι τα ακόλουθα.

- Μικροεπεξεργαστής είναι το τμήμα το οποίο μπορεί να τρέξει τον κώδικα αλλά και υπεύθυνο για την λειτουργία και συντονισμό των επιμέρους κομματιών.
- Η μνήμη στη οποία μπορούν να αποθηκευτούν προγράμματα και δεδομένα αλλά συνήθως η μνήμη για προγράμματα είναι διαφορετική από την μνήμη για δεδομένα.
- Αισθητήρες είναι το κομμάτι που αλληλεπιδρά με τον υπόλοιπο κόσμο παίρνοντας μετρήσεις ή σήματα και συλλέγει τα δεδομένα
- Συσκευή Επικοινωνίας Πομποδέκτης είναι το μέρος με το οποίο μπορούν να επικοινωνούν οι συσκευές σε wireless κανάλια μέσω ραδιοσυχνοτήτων.
- Η πηγή ενέργειας σε ένα κόμβο αισθητήρα μπορεί να είναι είτε κάποια μπαταρία είτε κάποιοι πυκνωτές που συγκρατούν ενέργεια. Τα τελευταία χρόνια έχουν αναπτυχθεί κόμβοι με τη δυνατότητα να παίρνουν ενέργεια από το περιβάλλον τους και να ανανεώνουν τα αποθέματα που έχουν, όπως με χρήση της ηλιακής ενέργειας.

Τέλος οι κόμβοι ανάλογα με την εφαρμογή και την λειτουργία που εκτελούν μπορούν να διακριθούν σε δύο κατηγορίες.

Οι κομβοί αισθητήρες (source nodes) οι οποίοι είναι οι κόμβοι οι οποίοι είναι υπεύθυνη για τις μετρήσεις / παρατηρήσεις που πρέπει να γίνουν και να αναφέρουν παρακάτω

αυτά τα δεδομένα χωρίς να τα επεξεργαστούν μέχρι να φτάσουν στους κόμβους αποδέκτες των δεδομένων (Sinks) που είναι η δεύτερη κατηγορία.

Τα Sinks είναι οι κόμβοι στους οποίους τα δεδομένα τα οποία “αισθάνθηκαν” οι κόμβοι αισθητήρες πρέπει να συγκεντρωθούν εκεί και ανάλογα με το πώς είναι προγραμματισμένοι τα χειρίζονται, είτε με την ενημέρωση κάποιας άλλης μονάδας εκτός του δικτύου αισθητήρων είτε απλά συγκεντρώνοντας τα. Τα Sinks μερικές φορές είναι μέρος του ίδιου του δικτύου αισθητήρων και κάποιες φορές δεν αποτελούν μέρος του δικτύου, αλλά είναι καθαρά συστήματα έξω από αυτό. Τις πλείστες φορές οι κόμβοι αισθητήρων (source nodes) είναι αρκετά πιο πολλοί από τα Sink Nodes.



Σχημα 1.3 Κόμβοι Αισθητήρες

2.3 Τρόποι Διάταξης Αισθητήρων

Ο τρόπος διάταξης των αισθητήρων σε ένα τέτοιο δίκτυο είναι ένας αρκετά σημαντικός παράγοντας όσον αφορά την καλή λειτουργία του δικτύου αλλά και την διάρκεια ζωής του. Οι θέσεις των αισθητήρων παίζουν ρολό στο αν θα μπορούν να “αισθανθούν” κάτι εκεί που βρίσκονται αλλά και αν μπορούν να προωθήσουν τις πληροφορίες προς τους Sink nodes.

Η τοποθέτηση των κόμβων μπορεί να διακριθεί σε δύο κατηγορίες τα καλά οργανωμένα, με σταθερή ανάπτυξη των κόμβων αισθητήρων σε συγκεκριμένα σημεία η καθορισμένη περιοχή και σε δίκτυα με τυχαία ανάπτυξη.

Μερικοί τρόποι ανάπτυξης των κόμβων που ακολουθούν αυτές τις δύο αρχές είναι

Απλή Διάχυση (Simple Diffusion) π.χ. ρίχνοντας μεγάλη ποσότητα κόμβων από ένα αεροπλάνο σε ένα δάσος.

Σταθερή τοποθέτηση (Constant Placement) όπου γίνεται με σκοπό η πυκνότητα των κόμβων να είναι ομοιόμορφη και οι κόμβοι τοποθετούνται στα ανάλογα σημεία.

R- Τυχαία Τοποθέτηση (R- Random Placement) είναι η περίπτωση όπου οι κόμβοι διασκορπίζονται ομοιόμορφα με βάση την ακτίνα και την γωνιακή κατεύθυνση από το κέντρο - Sink. Με αυτή τη μέθοδο έχουμε μεγαλύτερη πυκνότητα κόμβων κοντά στο Sink παρά στο υπόλοιπο πεδίο.

Μια άλλη μέθοδος είναι η τοποθέτηση των κόμβων με βάση κάποιο πλέγμα (Grid Placement) όπου η τοποθέτηση των κόμβων γίνεται ακριβώς εκεί όπου διασταυρώνονται οι γραμμές του πλέγματος [6].

2.4 Πρότυπα Επικοινωνίας

Εδώ θα αναφέρω μερικά από τα πρότυπα που χρησιμοποιούνται για επικοινωνία στα WSN [10].

- IEEE 802.15.4

Το IEEE 802.15.4 είναι για ασύρματη, χαμηλής ροής και περιορισμένης εμβέλειας επικοινωνία μεταξύ σχετικά απλών συσκευών, που καταναλώνουν ελάχιστη ενέργεια και τυπικά λειτουργούν σε ένα προσωπικό χώρο λειτουργίας. Είναι για χαμηλού ρυθμού μετάδοσης δίκτυα κάτι που χαρακτηρίζει η ευκολία στην εγκατάσταση, η αξιόπιστη μετάδοση δεδομένων, η λειτουργία περιορισμένης έκτασης, το χαμηλό κόστος και η λογική διάρκεια ζωής της μπαταρίας, ενώ διατηρούν παράλληλα μια απλή και ευέλικτη στοίβα πρωτοκόλλων. Λειτουργεί σε τρεις ζώνες συχνοτήτων και συνολικά έχει 27 κανάλια και στις τρεις ζώνες

- WirelessHart

Το WirelessHart standard παρέχει ένα πρωτόκολλο ασύρματης επικοινωνίας για μετρήσεις διεργασιών και εφαρμογές ελέγχου, στηρίζεται στο πρότυπο IEEE 802.15.4 για λειτουργία χαμηλής ισχύος 2.4GHz και είναι συμβατό με όλες τις υπάρχουσες συσκευές, εργαλεία και συστήματα. Είναι ασφαλές αξιόπιστο και αποδοτικό από άποψη ενέργειας.

- 6LoWPAN

Επιτρέπει την IPv6 επικοινωνία πακέτων σε ένα IEEE 802.15.4 δίκτυο. Το πρότυπο αυτό παρέχει όλα τα πλεονεκτήματα της IP επικοινωνίας. Δεδομένου πως το μέγεθος των IP πακέτων είναι μεγαλύτερο από αυτά που χρησιμοποιούνται σε ένα 802.15.4 δίκτυο γίνεται χρήση ακόμα ενός επιπέδου που είναι υπεύθυνο για την προσαρμογή των πακέτων

- Wibree

Το Wibree συνιστά τεχνολογία ασύρματης επικοινωνίας, σχεδιασμένη για χαμηλή κατανάλωση ισχύος, μετάδοση περιορισμένου βεληνεκούς και χαμηλού κόστους συσκευές. Επιτρέπει την επικοινωνία μεταξύ μικρών συσκευών τροφοδοτούμενων με μπαταρία και Bluetooth συσκευών. Λειτουργεί στην περιοχή συχνοτήτων 2.4 GHz, παρέχοντας ρυθμό δεδομένων 1Mbps για αποστάσεις 5-10m.

2.5 Εφαρμογές των ασύρματων δικτύων αισθητήρων

Οι τομείς στους οποίους μπορούν να έχουν εφαρμογή τα δίκτυα ασύρματων αισθητήρων είναι πάρα πολλοί. Μερικοί από αυτούς είναι

- Εφαρμογές βοήθειας σε περίπτωση καταστροφών πχ πλημμύρες, πυρκαγιές όπου μέσω των αισθητήρων αναγνωρίζονται οι καταστροφές αυτές και ενημερώνονται ανάλογα οι αρμόδιες υπηρεσίες για έγκαιρη αντιμετώπιση.
- Εφαρμογές για έλεγχο και για χαρτογράφηση βιοποικιλότητας.
- Εφαρμογές για έξυπνα κτίρια πχ για θέματα διατήρησης θερμοκρασίας και υγρασία μέσα στο κτίριο αφού σε περίπτωση που μειωθεί η υγρασία ή η υγρασία σε ένα χώρο τότε με κάποιο τρόπο πρέπει να ενημερωθεί η υπεύθυνη μονάδα ελέγχου του κτιρίου για κάνει τις κατάλληλες ενέργειες για επαναφορά των κατάλληλων συνθηκών.
- Εφαρμογές για τη διαχείριση διαφόρων εγκαταστάσεων που απαιτούν ασφάλεια όπου με τους κόμβους μπορεί να γίνει ανίχνευση κίνησης και άλλων παραμέτρων με τις οποίες μπορούμε να έχουμε καλύτερο έλεγχο ασφάλειας. Εδώ μπορεί να εφαρμοστεί και κινητικότητα κόμβων σε τέτοιες εφαρμογές βασισμένους σε κάποιες άλλες ρομποτικές μονάδες με αλγόριθμους για “έξυπνη” κίνηση για καλύτερη κάλυψη του χώρου.
- Εφαρμογές για επιτήρηση μηχανών και για προληπτική συντήρηση αφού μπαίνουν σε σημεία χωρίς να απαιτείται καλωδίωση άρα λιγότερο κόστος και επίσης μπορούν να μουν σε σημεία όπου λόγω επικινδυνότητας δεν μπορεί να έχει πρόσβαση ο άνθρωπος ανά πάσα στιγμή.
- Εφαρμογές που αφορούν τη γεωργία.

- Εφαρμογές που αφορούν την ιατρική και την υγεία πχ μέτρηση ποιότητας αέρα ή ανίχνευση επιβλαβών ουσιών σε νερό σε καθορισμένη περιοχή.

2.6 Προβλήματα-Προκλήσεις στα Ασύρματα Δίκτυα Αισθητήρων

Τα ασύρματα δίκτυα αισθητήρων από την φύση τους είναι αυτόνομα και οι κόμβοι που χρησιμοποιούνται είναι μικροί σε μέγεθος. Αυτά τα δύο χαρακτηριστικά τους είναι τα πιο περιοριστικά όσον αφορά την τεχνολογία αυτή γιατί τα πιο συνηθισμένα προβλήματα που αντιμετωπίζει πηγάζουν από αυτά τα δύο χαρακτηριστικά. Σε αυτό το σημείο θα αναφέρω μερικά από τα προβλήματα αυτά αλλά και με μερικές προκλήσεις τις οποίες καλούνται να αντιμετωπίσουν τα Ασύρματα Δίκτυα Αισθητήρων.

- Ο περιορισμένος αριθμός πόρων που διαθέτουν οι κόμβοι λόγω του μικρού τους μεγέθους αλλά και λόγω του ότι οι κόμβοι και γενικά το δίκτυο πρέπει να είναι αυτόνομα αποτελεί ένα σημαντικό πρόβλημα αλλά και πρόκληση που πρέπει να αντιμετωπιστεί. Η αδυναμία ύπαρξης συνεχόμενης παροχής ενέργειας προς του κόμβους είναι ένα περιοριστικό μετρώ αφού αλυσιδωτά επηρεάζονται όλες οι μονάδες του κόμβου επεξεργαστής, μνήμη, και εμβέλεια μετάδοσης. Έτσι για να πετύχουμε την μέγιστη διάρκεια ζωής πρέπει να γίνεται χρήση αλγορίθμων και πρωτοκόλλων με χαμηλή κατανάλωση ενέργειας. Στις μέρες μας ωστόσο αρχίζουν να βγαίνουν και αισθητήρες που μπορούν να αναπληρώσουν την αποθηκευμένη ενέργεια από ανανεώσιμες πηγές όπως η ηλιακή.
- Οι κόμβοι έχουν περιορισμένες υπολογιστικές δυνατότητες λόγω μεγέθους που δεν επιτρέπει χρήση μεγάλων και ακριβών επεξεργαστών αλλά λόγω του ότι πρέπει να κρατηθεί χαμηλά το κόστος κάθε κόμβου αφού λόγω της φύσης των WSN's απαιτούνται μεγάλοι αριθμοί από κόμβους για να υλοποιηθούν.
- Άλλη μια πρόκληση των ασύρματων δικτύων αισθητήρων είναι αδυναμία πρόβλεψης τόσο της φυσικής κατάστασης του κόμβου όσο και το αν μπορεί ο κόμβος να εκπληρώσει τις διεργασίες που πρέπει. Αυτό συμβαίνει γιατί τα περιβάλλοντα στα οποία εγκαθίστανται τα δίκτυα αυτά συμβαίνουν ανεξέλεγκτες αλλά και απρόβλεπτες καταστάσεις. Για παράδειγμα δεν μπορείς να ξέρεις ποτέ ένας κόμβος θα χαθεί από το δίκτυο λόγω έλλειψης ενέργειας ή λόγω εμποδίων που επηρεάζουν την επικοινωνία. Έτσι πρέπει να υλοποιούνται αλγόριθμοι που να μπορούν αυτόνομα να αντιμετωπίζουν τέτοιου είδους

καταστάσεις και να μπορούν δυναμικά να διαχειρίζονται αλλαγές στην τοπολογία αλλά και απώλειες κόμβων.

- Τα δίκτυα αισθητήρων πρέπει να είναι επεκτάσιμα ανά πάσα στιγμή αν χρειαστεί και να μπορούν να δεχτούν νέους κόμβους. Οι κομβοί αυτοί πρέπει να έχουν την δυνατότητα να προσαρμοστούν δυναμικά και να δουλεύουν στο δίκτυο χωρίς προβλήματα όπως τους υπόλοιπους κόμβους που ήταν αρχικά στο δίκτυο.
- Ένα ακόμα θέμα το οποίο προβληματίζει για την χρήση των ασυρμάτων δικτύων αισθητήρων είναι το θέμα της ασφάλειας. Οι κόμβοι αισθητήρες λόγω του μικρού μεγέθους τους και του χαμηλού κόστους τους δεν έχουν την δυνατότητα να παρέχουν την απαιτούμενη ασφάλεια. Ένα από τα προβλήματα είναι το σταθερό εύρος συχνοτήτων στις οποίες εκπέμπουν για να επικοινωνούν και αυτό καθιστά πολύ εύκολο να μπορέσει κάποιος να “ακούσει” τα δεδομένα τα οποία μεταφέρονται. Επίσης λόγω τις δυναμικότητας στην δημιουργία των δικτύων μπορεί κάποιος να εισάγει ένα δικό του κόμβο στο δίκτυο και να προκαλέσει προβλήματα στην λειτουργία του. Για να αποφευχθούν αυτά τα προβλήματα πρέπει με κάποιο τρόπο να εφαρμοστούν μηχανισμοί ασφάλειας στους κόμβους είτε μηχανικά είτε μέσω του λογισμικού τους αλλά εδώ μπαίνει η μεγάλη πρόκληση να κρατηθεί χαμηλά η πολυπλοκότητα των κόμβων.
- Ένα άλλο πρόβλημα είναι κατά πόσο υπάρχει συνέπεια και αξιοπιστία στην παράδοση των δεδομένων. Τα δίκτυα αυτά είναι εγκατεστημένα σε πραγματικά περιβάλλοντα και για αυτό πρέπει να έχουν και την ανάλογη απόκριση. Για παράδειγμα θα αναφέρω πω μπορεί να εμφανιστή αυτό το πρόβλημα με βάση μια εφαρμογή στην οποία χρησιμοποιούνται τα WSN. Για παράδειγμα σε μια εγκατάσταση όπου υπάρχουν αποθηκευμένα διάφορα επιβλαβή για το περιβάλλον υλικά και ο έλεγχος για απώλειες από την εγκατάσταση ή διαρροές ελέγχεται με κόμβους αισθητήρες τότε οι κόμβοι πρέπει να ανταποκρίνονται άμεσα και να ενημερώνετε η ασφάλεια αλλά και να δίνουν έγκυρες πληροφορίες. Δεν υπάρχουν περιθώρια λάθους αλλά ούτε και λανθασμένων ενδείξεων.
- Ανοχή Σφαλμάτων (Fault tolerance). Στα Ασύρματα Δίκτυα Αισθητήρων είναι δεδομένο ότι υπάρχει μεγάλη πιθανότητα οι κόμβοι να μείνουν χωρίς ενέργεια ή να καταστραφούν ή ακόμα, η ασύρματη επικοινωνία μεταξύ δύο κόμβων μπορεί να διακοπεί μόνιμα. Για αυτούς τους λόγους είναι πολύ σημαντικό τα Ασύρματα

Δίκτυα Αισθητήρων να είναι σε θέση να ανεχτούν τέτοια ελαττώματα. Για να υπάρχει ανοχή στην αποτυχία κάποιων κόμβων, η περιττή επέκταση τους είναι απαραίτητη, για παράδειγμα, χρησιμοποιώντας περισσότερους κόμβους.

- Προσαρμοστικότητα Maintainability. Το σύστημα πρέπει να προσαρμόζεται δεδομένου ότι τόσο το περιβάλλον ενός Ασύρματου Δικτύου Αισθητήρων όσο και το ίδιο είναι πιθανό να αλλάξουν (μειωμένες μπαταρίες, κόμβοι που αποτυγχάνουν, νέοι στόχοι). Το σύστημα πρέπει να παρακολουθεί την κατάσταση του για να μπορεί να αλλάζει τις λειτουργικές παραμέτρους του ή για να μπορεί να επιλέγει τις διάφορες παραμέτρους (π.χ. για να παρέχουν τη χαμηλότερη ποιότητα όταν ο ενεργειακός πόρος γίνεται λιγοστός). Από αυτή την άποψη, το δίκτυο πρέπει να μπορεί να διατηρείται. Θα πρέπει επίσης, να είναι ικανό να αλληλεπιδρά με τους εξωτερικούς μηχανισμούς συντήρησης για να εξασφαλίζει την εκτεταμένη λειτουργία του σε ένα απαραίτητο επίπεδο ποιότητας.

Κεφάλαιο 3

Έλεγχος Συμφόρησης

3.1 Γενικά

3.2 Έλεγχος Συμφόρησης στα Ενσύρματα και Ασύρματα Δίκτυα

3.3 Έλεγχος Συμφόρησης στα Ασύρματα Δίκτυα Αισθητήρων

3.4 Λόγοι Δημιουργίας και οι Επιπτώσεις της Συμφόρησης

3.5 Τύποι Συμφόρησης

3.1 Γενικά

Ο έλεγχος συμφόρησης είναι ένας μηχανισμός που υλοποιείται στο πρωτόκολλο μεταφοράς μαζί με το μηχανισμό για αξιόπιστη μεταφορά δεδομένων. Ο έλεγχος συμφόρησης αφορά τον έλεγχο της κίνησης σε ένα δίκτυο, προσπαθώντας να αποφύγει τη ενεργή συμμετοχή οποιασδήποτε διεργασίας ή τις δυνατότητες του καλωδίου των ενδιάμεσων κόμβων και δικτύων, ακολουθώντας κάποια βήματα περιορισμού των πόρων μειώνοντας το ρυθμό αποστολής πακέτων.

Στη συνέχεια, θα δούμε πολύ σύντομα τον έλεγχο συμφόρησης στα ενσύρματα και ασύρματα δίκτυα και αναλυτικότερα τον έλεγχο συμφόρησης στα Ασύρματα Δίκτυα Αισθητήρων. Θα δούμε το τι προκαλεί τη συμφόρηση αλλά και τις επιπτώσεις που έχει η συμφόρηση στα δίκτυα αυτά.

3.2 Έλεγχος Συμφόρησης στα Ενσύρματα και Ασύρματα Δίκτυα

Στα ενσύρματα και ασύρματα δίκτυα, η συμφόρηση συμβαίνει όταν η κίνηση στο δίκτυο ξεπερνά τη χωρητικότητα σε οποιοδήποτε σημείο στο δίκτυο. Ο έλεγχος της συμφόρησης στα ενσύρματα δίκτυα, συνήθως γίνεται χρησιμοποιώντας μηχανισμούς από άκρο σε άκρο (end-to-end) όπως στο TCP, οι οποίοι είναι απλοί και εύρωστοι και επιπέδου δικτύου. Παρόλα αυτά, η μέθοδος αυτή δεν επιλύει το πρόβλημα στα ασύρματα δίκτυα γιατί οι ταυτόχρονες μεταδόσεις στα διάφορα «καλώδια»

αλληλεπιδρούν και επηρεάζουν το ένα το άλλο, και αυτό επειδή η ποιότητα του καναλιού δείχνει μεγάλη μεταβλητότητα όσο προχωρά ο χρόνος.

Ο μηχανισμός από άκρο σε άκρο δεν μπορεί να χρησιμοποιηθεί στα Ασύρματα Δίκτυα Αισθητήρων γιατί δεν είναι αρκετά αποδοτικός στα δίκτυα αυτά. Οι λόγοι για τους οποίους δεν είναι αποδοτικός είναι οι εξής: τα μηνύματα ελέγχου που χρησιμοποιούνται για τον εντοπισμό των χαμένων πακέτων στο μηχανισμό από άκρο σε άκρο χρησιμοποιούν ένα αντίστροφο μονοπάτι επιστροφής που αποτελείται από πολλά hops και αυτό δεν είναι αποδοτικό από πλευράς ενέργειας αλλά και συμφόρησης.

Επίσης, τα μηνύματα ελέγχου που «ταξιδεύουν» διαμέσου πολλών hops έχουν πολύ μεγάλη πιθανότητα να χαθούν είτε λόγω λαθών στο σύνδεσμο είτε λόγω συμφόρησης. Και τέλος, ο μηχανισμός από άκρο σε άκρο για την ανακάλυψη κάποιου χαμένου πακέτου αναπόφευκτα χρησιμοποιεί από άκρο σε άκρο επανάληψη της μετάδοσης των πακέτων που χρειάζονται περισσότερη ενέργεια σε σύγκριση με τις hop-by-hop επαναλήψεις μετάδοσης των πακέτων.

Το άλλο είδος μηχανισμού που μπορεί να χρησιμοποιηθεί είναι το hop-by-hop. Παρόλο που στο μηχανισμό αυτό όλοι οι ενδιάμεσοι κόμβοι που συμμετέχουν στη διαδικασία από τον source μέχρι το Sink πρέπει να αλλάξουν συμπεριφορά, γίνεται γρήγορη μείωση της συμφόρησης στο δίκτυο μειώνοντας των αριθμό των πακέτων που βρίσκονται στο δίκτυο .

Η όλη διαδικασία γίνεται χρησιμοποιώντας λιγότερα πακέτα στο δίκτυο, άρα γίνεται διαφύλαξη της ενέργειας. Ο έλεγχος της συμφόρησης εφαρμόζεται με μια επαναλαμβανόμενη μέθοδο στον κάθε κόμβο και οι αποφάσεις βασίζονται μόνο σε τοπικές πληροφορίες [8].

3.3 Έλεγχος Συμφόρηση στα Ασύρματα Δίκτυα Αισθητήρων

Πιο πάνω είδαμε, ότι η συμφόρηση στα ασύρματα δίκτυα διαφέρει από τη συμφόρηση στα ενσύρματα δίκτυα και επίσης, είδαμε ότι οι κόμβοι αισθητήρες στα Ασύρματα Δίκτυα Αισθητήρων αντιμετωπίζουν περιορισμούς όπως περιορισμένες υπολογιστικές δυνατότητες, περιορισμένο χώρο μνήμης, επικοινωνιακό εύρος ζώνης και πάνω από όλα ενέργεια. Όταν κάποιο γεγονός συμβεί στα δίκτυα αυτά, παρατηρείται μια μεγάλη παραγωγή δεδομένων από τους κόμβους αισθητήρων που έχουν σαν κατεύθυνση το Sink.

Για τους πιο πάνω λόγους, ο έλεγχος συμφόρησης θα πρέπει να γίνεται με γνώμονα τόσο τις δυνατότητες που έχουν οι κόμβοι αλλά και την συνέπεια που πρέπει να έχει το δίκτυο στην μεταφορά των πακέτων αφού συνήθως είναι σημαντικά στοιχεία που δεν πρέπει να έχουμε απώλειες.

Γι' αυτό θα πρέπει να σχεδιαστούν αποδοτικοί μηχανισμοί και μερικοί από αυτούς είναι:

- Εντοπισμού συμφόρησης (congestion detection), όπως βάση το μήκος της ουράς του κόμβου ή ο χρόνος εξυπηρέτησης του πακέτου.
- Γνωστοποίησης συμφόρησης (congestion notification), αφού εντοπιστεί η συμφόρηση υπάρχουν δύο τρόποι για να ενημερωθεί το δίκτυο. Αυτονόητα (implicit), δηλαδή με ένα δυαδικό bit που γίνεται piggyback σε άλλα πακέτα ενημερώνεται όλο το δίκτυο για τη συμφόρηση ή κατηγορητικά (explicit) στέλνονται ειδικά μηνύματα για να ενημερώσουν το δίκτυο.
- Αποφυγής συμφόρησης (congestion avoidance). Ο έλεγχος συμφόρησης ψάχνει να βρει ένα βέλτιστο (και δίκαιο) τρόπο δέσμευσης ρυθμού αποστολής δεδομένων ανά αισθητήρα (optimal fair rate allocation) ο οποίος να είναι αποδοτικός, έτσι ώστε όταν οι αισθητήρες στέλνουν με αυτό το ρυθμό, το δίκτυο να χρησιμοποιείται πλήρως, και ο ρυθμός μετάδοσης δεδομένων(ωφέλιμο ρυθμό - καλά δεδομένα) ανά αισθητήρα να είναι κοντά στο ρυθμό αποστολής (όχι πολλές επαναλήψεις μετάδοσης πακέτων). Με άλλα λόγια τα πρωτόκολλα ελέγχου συμφόρησης για τα WSN's πρέπει να περιορίζουν το πέταγμα πακέτων αφού το πέταγμα πακέτων σημαίνει και απώλεια ενέργειας λόγω της επανάληψης της αποστολής. Αυτό μπορεί να γίνει με τον ενεργό έλεγχο συμφόρησης (Active Congestion Control), δηλαδή με μια διαδικασία κατά την οποία κάνουμε αποφυγή συμφόρησης με πολύ λίγο κόστος. Ένα παράδειγμα αποφυγής συμφόρησης είναι να μειώσουμε το ρυθμό αποστολής δεδομένων όταν ο αποθηκευτικός χώρος (Buffer space) του παραλήπτη ξεπερνά μια τιμή αυτός ο μηχανισμός είναι κάτι παρόμοιο με αυτό που υλοποιώ στα πλαίσια αυτής της διπλωματικής εργασίας.
- Μείωσης συμφόρησης (congestion mitigation): Έστω ένα δίκτυο στο οποίο ο κάθε αισθητήρας στέλνει με σταθερό ρυθμό. Όταν ο συνολικός ρυθμός δεδομένων ξεπεράσει τη χωρητικότητα του δικτύου (network capacity), η τεχνική της μείωσης συμφόρησης θα καθορίσει το πως θα πρέπει οι αισθητήρες

να ρυθμίσουν τη μετάδοση δεδομένων τους, έτσι ώστε ο χρήσιμος ρυθμός δεδομένων του δικτύου (network goodput) να ελαττωθεί ήπιο - θετικό τρόπο (graceful degradation). Δηλαδή, αφού γίνει εντοπισμός της συμφόρησης και ενημερωθεί το δίκτυο για τη συμφόρηση πρέπει κάποιοι μηχανισμοί να ενεργοποιηθούν για να ελέγξουν τη συμφόρηση και έτσι να μπορέσει να λειτουργήσει ξανά σωστά το δίκτυο. Κάποιος μηχανισμός που χρησιμοποιείται στα Ασύρματα Δίκτυα Αισθητήρων είναι ο hop-by-hop.

3.4 Λόγοι Δημιουργίας και οι Επιπτώσεις της Συμφόρησης

Σε αυτήν την ενότητα θα δούμε κάποια συμπτώματα, δηλαδή το τι προκαλεί τη συμφόρηση στα Ασύρματα Δίκτυα Αισθητήρων.

Ο ρυθμός με τον οποίο ωθούνται τα δεδομένα, δηλαδή ο ρυθμός με τον οποίο οι κόμβοι αισθητήρες αισθάνονται κάποιο γεγονός και το ωθούν εξαρτάται από τον τύπο της εφαρμογής. Δηλαδή, για κάποιες εφαρμογές κάποιο γεγονός μπορεί να είναι περιοδικό ενώ για κάποιες άλλες μπορεί να είναι πιο συχνό. Επίσης, σε κάποιες εφαρμογές μπορεί να δημιουργείται πολύ μικρή κίνηση σε μια μεμονωμένη περιοχή του δικτύου αισθητήρων για παράδειγμα, target detection, ενώ σε άλλες όπως, τη φωτιά, τους σεισμούς, δημιουργούνται μεγάλα κύματα δεδομένων που πρέπει να μεταφερθούν σε ολόκληρη την περιοχή για να φτάσουν το Sink. Παρόλα αυτά, και στις δύο περιπτώσεις μπορεί να παρατηρηθεί συμφόρηση γιατί το κανάλι μετάδοσης αλλάζει στο χρόνο, συχνά δραματικά, και ταυτόχρονες μεταφορές δεδομένων σε διαφορετικά κανάλια μεταδόσεων αλληλεπιδρούν, κάνοντας έτσι την ποιότητα να εξαρτάται όχι μόνο από το θόρυβο αλλά και από την πυκνότητα των κόμβων αισθητήρων στο δίκτυο.

Η πρόσθεση κόμβων αισθητήρων για σκοπούς αντικατάστασης κάποιων κόμβων που απέτυχαν ή για αύξηση της πυκνότητας τους για να μπορούν να έχουν καλύτερη απόδοση. Δηλαδή, όταν οι υπάρχον αισθητήρες δεν είναι αρκετοί για μια εφαρμογή και προσθέτονται και άλλοι για να μπορούν να αισθάνονται πιο γρήγορα ένα γεγονός και έτσι να έχουν καλύτερη απόδοση. Ακόμα και η αφαίρεση κόμβων αισθητήρων από το δίκτυο επειδή απέτυχαν για παράδειγμα, λόγω έλλειψης ενέργειας, μπορούν να κάνουν μια περιοχή όπου δεν υπήρχε συμφόρηση να την εφοδιάσουν με κόμβους και τελικά, να γίνει συμφόρηση στη συγκεκριμένη περιοχή.

Στα παραδοσιακά ενσύρματα και ασύρματα δίκτυα, η υπερχείλιση των αποθηκευτικών χώρων και το πέταγμα των πακέτων, και η αύξηση της καθυστέρησης είναι τα κύρια

συμπτώματα που προκαλούν τη συμφόρηση. Οι διάφοροι ερευνητές έχουν καταφέρει με συνδυασμό της αναπροσαρμογής του ρυθμού στα άκρα του δικτύου και το πέταγμα πακέτων που γίνεται στο επίπεδο δικτύου, να εγγυώνται ότι αυτά τα δίκτυα θα μπορούν να λειτουργούν χωρίς να καταρρέουν λόγω συμφόρησης. Στα Ασύρματα Δίκτυα Αισθητήρων εκτός από την υπερχείλιση των αποθηκευτικών χώρων ένα σημαντικό σύμπτωμα τη συμφόρησης είναι η μείωση της ποιότητας του καναλιού μετάδοσης, που προκαλείται από την αύξηση της ποσότητας της κίνησης που στέλνεται από άλλα κομμάτια του δικτύου. Επειδή τα κανάλια μετάδοσης δεν προστατεύονται το ένα από το άλλο όπως γίνεται στα ενσύρματα δίκτυα, η κίνηση ταξιδεύει σε ένα δεδομένο κομμάτι του δικτύου έχοντας επιβλαβής επίδραση στην ποιότητα του καναλιού σε άλλα κομμάτια του δικτύου. Η φτωχή και εναλλασσόμενη στο χρόνο ποιότητα του καναλιού, η ασυμμετρία στο επικοινωνιακό κανάλι και οι κρυμμένοι κόμβοι (hidden terminal) όλα δουλεύουν εναντίον μιας καλά οργανωμένης κίνησης έτσι ώστε να κάνουν την παράδοση της δύσκολη.

Πιο πάνω αναφέραμε τα πιθανά συμπτώματα που μπορεί να προκαλέσουν συμφόρηση σε ένα δίκτυο ασύρματων αισθητήρων. Τώρα πρέπει να δούμε και τις επιπτώσεις που έχει η συμφόρηση στα δίκτυα αυτά. Στην κορυφή της πυραμίδας των επιπτώσεων που προκαλεί η συμφόρηση στα Ασύρματα Δίκτυα Αισθητήρων βρίσκεται η σπατάλη ενέργειας. Η ενέργεια είναι ένα από τα σημαντικότερα προβλήματα που έχουν να αντιμετωπίσουν τα δίκτυα αυτά, αφού η ενέργεια στους κόμβους αισθητήρων είναι μειωμένη, και η συμφόρηση κάνει ακόμα πιο έντονο το πρόβλημα. Αυτό που συμβαίνει είναι ότι κόμβος-πηγή σπαταλάει την ενέργεια του για να στείλει κάποιο πακέτο δεδομένων και οι ενδιάμεσοι κόμβοι σπαταλούν την ενέργεια τους για να μεταφέρουν το πακέτο προς το Sink, και μόλις γίνει συμφόρηση το πακέτο πετάγεται, μαζί όλη την ενέργεια όλων των κόμβων που συμμετείχαν για τη μεταφορά του. Τελικά, σπαταλείται όλη αυτή η ενέργεια χωρίς τελικά το πακέτο να φτάσει στον προορισμό του. Στη συνέχεια, πρέπει να σπαταληθεί ξανά αυτή η ενέργεια για να μεταφερθεί ξανά το ίδιο πακέτο. Η σπατάλη αυτής της ενέργειας μπορεί να κάνει κάποιο κόμβο να αποτύχει λόγω έλλειψης ενέργειας και να οδηγήσει ακόμα και σε διαχωρισμό του δικτύου αν πολλοί κόμβοι αποτύχουν λόγω έλλειψης ενέργειας. Δηλαδή, η συμφόρηση προκαλεί και μείωση της διάρκειας ζωής των κόμβων αισθητήρων ή ακόμα, και ολόκληρου του δικτύου.

Μαζί με την έλλειψη ενέργειας και το διαχωρισμό του δικτύου, αυτό που μπορεί να συμβεί είναι η μείωση της ακρίβειας των πληροφοριών ή ακόμα και χάσιμο σημαντικών πληροφοριών. Για παράδειγμα, σε περίπτωση που η πληροφορία που μεταφέρεται σε ένα δίκτυο ασύρματων αισθητήρων ήταν για να ειδοποιήσει για πυρκαγιά, και αυτό το μήνυμα χάθηκε λόγω της συμφόρησης, αυτό μπορεί να προκαλέσει ολέθριες καταστροφές αφού δεν θα ενημερωθούν ποτέ οι πυροσβέστες για την πυρκαγιά.

Η συμφόρηση προκαλεί μείωση του throughput και αύξηση της καθυστέρησης. Σε συνέχεια με το προηγούμενο παράδειγμα, για την πυρκαγιά καθυστερημένες πληροφορίες για φωτιά μπορεί να οδηγήσουν και πάλι σε ολέθριες καταστροφές. Είναι σημαντικό να αναφέρουμε ότι όλες οι επιπτώσεις από τη συμφόρηση, δηλαδή η σπατάλη ενέργειας, η μείωση ζωής του δικτύου ή των κόμβων, η μείωση της ακρίβειας των πληροφοριών, το χάσιμο πακέτων, η μείωση του throughput και η αύξηση της καθυστέρησης είναι όλα απαιτήσεις για την ποιότητα υπηρεσιών στα Ασύρματα Δίκτυα Αισθητήρων, όπως είδαμε και προηγουμένως.

3.5 Τύποι Συμφόρησης

Η συμφόρηση στα ασύρματα δίκτυα αισθητήρων μπορεί να εμφανιστεί σε διάφορους τύπους και αυτό εξαρτάτε με το πώς χάθηκαν τα πακέτα. Οι τύποι με τους οποίους μπορεί να εμφανιστεί σε σχέση με το πώς χάθηκαν τα πακέτα είναι οι εξής [6]:

Τύπος 1 Σε μια συγκεκριμένη περιοχή πολλοί κόμβοι είναι ο ένας μέσα στην εμβέλεια του άλλου και προσπαθούν να μεταδώσουν ταυτόχρονα με αποτέλεσμα να έχουν απώλειες πακέτων λόγω παρεμβολών και έτσι μειώνετε το throughput όλων των κόμβων στη περιοχή αυτή. Αυτού του είδους η συμφόρηση μπορεί να αντιμετωπιστεί με ρητό τοπικό συγχρονισμό μεταξύ των γειτονικών κόμβων αλλά δεν μπορεί να εξαλειφτεί γιατί οι κόμβοι οι οποίοι δεν γειτονεύουν αλλά είναι στην ίδια εμβέλεια μπορούν να κάνουν παρεμβολές

Τύπος 2 Είναι όταν σε ένα κόμβο η ουρά(queue) η ο αποθηκευτικός χώρος (buffer) που χρησιμοποιείται για να φυλάει τα πακέτα που θα μεταδοθούν υπερχειλίσουν (overflow) με αποτέλεσμα να μην μπορούν αλλά πακέτα να πάρουν θέση για να εξυπηρετηθούν και έτσι να έχουμε απώλεια πακέτων. Αυτός ο τύπος συμφόρησης παρατηρείται ευρέως στα ενσύρματα δίκτυα, και το χάσιμο πακέτων αφορά την επιτυχημένη παραλαβή του πακέτου αλλά το πακέτο πρέπει να πεταχτεί λόγω της υπερχειλίσης.

Επιπρόσθετα μπορεί να γίνει διαχωρισμός της συμφόρησης βασισμένη και σε ποιο σημείο του δικτύου συμβαίνει το πρόβλημα

Τύπος 1 Είναι όταν υπάρχει θερμό σημείο (hotspot) δηλαδή το σημείο που αντιμετωπίζει το πρόβλημα κοντά στην πηγή (source). Αυτό ονομάζεται Συμφόρηση Πηγής (source congestion) και δημιουργείται όταν υπάρχει πυκνή κατανομή κόμβων έτσι έχουμε μεγάλη παραγωγή δεδομένων στο σημείο του συμβάντος αυτό θα δημιουργήσει επίμονα hotspots πολύ κοντά στην πηγή κάπου ένα με δύο κόμβους. Σε αυτή την περίπτωση μηχανισμοί οι οποίοι μπορούν να λειτουργήσουν με ταχύτητα εντοπίζοντας τα σημεία μπορούν να απαλείψουν την συμφόρηση στα σημεία αυτά όπως επίσης και με το να γίνουν οι πηγές ασύγχρονες μπορεί να είναι αποτελεσματικός.

Τύπος 2 Είναι όταν υπάρχει θερμό σημείο κοντά στο Sink και ονομάζεται Sink Congestion. Ακόμη και αραιά αναπτυγμένοι κόμβοι που παράγουν δεδομένα σε χαμηλούς ρυθμούς μπορούν ενδεχομένως να δημιουργήσουν παροδικά hotspots οπουδήποτε μέσα στο πεδίο που είναι αναπτυγμένοι οι κόμβοι (sensor field) αλλά ποιά συχνά συμβαίνει μακριά από τις πηγές και ποιά κοντά στα Sink's. Μια γρήγορη ανάλυση της κλίμακας των εντοπισμένων hotspots και η χρήση ενός μηχανισμού αποσυμφόρησης με συνδυασμό απόρριψη μερικών πακέτων είναι αποτελεσματικά. Οι πηγές λόγω της παροδικής φύσης αυτού του προβλήματος μπορεί να μην λάβουν μέρος στον μηχανισμό αποσυμφόρησης. Ακόμα μία μέθοδος η οποία μπορεί να αντιμετωπίσει το Sink Congestion είναι να κατανεμηθούν ομοιόμορφα στο πεδίο πολλά Sinks και έτσι να ισοζυγιστεί η συμφόρηση προς αυτά χωρίς να χρειάζεται να αντιμετωπιστεί μόνο από μερικούς κόμβους κάτι που δημιουργούσε αυτό το πρόβλημα.

Τύπος 3 Αυτή η μορφή συμφόρησης ονομάζεται Forwarder Congestion και παρατηρείται όταν στο δίκτυο έχουμε περισσότερες από μια ροές δεδομένων-πακέτων και αυτές οι ροές διασταυρώνονται μέσα στο δίκτυο. Με τις ροές εννοώ την μεταφορά πακέτων από Source στο Sink. Το πρόβλημα εντοπίζεται στο σημείο όπου οι ροές διασταυρώνονται και επομένως θα έχουμε hotspot. Σε ένα δίκτυο με επικοινωνία σε μορφή δέντρου κάθε ενδιάμεσος κόμβος μπορεί να αντιμετωπίσει αυτό το πρόβλημα.

Τώρα σε σύγκριση με τους δύο προηγούμενους τύπους συμφόρησης με βάση τον διαχωρισμό του πού συμβαίνει το Forwarder Congestion είναι με διαφορά το ποίο προκλητικό πρόβλημα λόγω τις δυσκολίας να προβλέψουμε στα σημεία όπου μπορεί να συμβεί λόγω της δυναμικότητας του δικτύου. Σε αυτή την περίπτωση ακόμα και αραιά κατανεμημένοι κόμβοι που παράγουν δεδομένα μπορούν να χρησιμοποιήσουν επίμονα αλλά και παροδικά hotspots διασκορπισμένα στο sensor field. Ο συνδυασμός μιας γρήγορης κλιμακωτής ενέργειας για επίλυση των εντοπισμένων παροδικών hotspots μαζί με μία κλειστού βρόγχου ρύθμιση του ρυθμού όλων των πηγών οι οποίες συμβάλουν στην δημιουργία επίμονων hotspots είναι αποτελεσματικός σε αυτές τις περιπτώσεις.

Κεφάλαιο 4

Αντιμετώπιση Συμφόρησης

4.1 Γενικά

4.2 Πρωτόκολλα για Αντιμετώπιση της Συμφόρησης

4.3 Περιγραφή Πρωτόκολλων

4.1 Γενικά

Η συμφόρηση, και ειδικότερα ο έλεγχος συμφόρησης είναι ένας μηχανισμός που υλοποιείται στο επίπεδο μεταφοράς από τα γνωστά κλασσικά επίπεδα (φυσικό, εφαρμογής, δικτύου). Στα Ασύρματα Δίκτυα Αισθητήρων έχουν σχεδιαστεί και αναπτυχθεί ειδικά πρωτόκολλα που να υλοποιούν έλεγχο της συμφόρησης. Ειδικότερα, έχουν σχεδιαστεί ειδικοί μηχανισμοί για εντοπισμό της συμφόρησης, για γνωστοποίηση της συμφόρησης στο υπόλοιπο δίκτυο και για αποφυγή της συμφόρησης, όπως είδαμε και προηγουμένως. Σημαντικό είναι το γεγονός ότι τα περισσότερα πρωτόκολλα που αναπτύχθηκαν αφορούν τον έλεγχο της συμφόρησης στην περιοχή του Sink, αφού έχει αποδειχτεί από πολλούς ερευνητές ότι η συμφόρηση συμβαίνει πιο συχνά στη συγκεκριμένη περιοχή, όπου φτάνουν όλα τα πακέτα δεδομένων των κόμβων αισθητήρων.

4.2 Πρωτόκολλα για Αντιμετώπιση της Συμφόρησης

Κάποια από τα πρωτόκολλα που αναπτύχθηκαν μέχρι στιγμής είναι το Congestion Control and Fairness for Many-to-One Routing (CCF) , το Congestion Detection and Avoidance (CODA) [7], το Event-to-Sink Reliable Transport (ESRT) , το Mitigating Congestion (Fusion) [3] και το SenTCP [6] . Επιπρόσθετα με αυτούς τους μηχανισμούς οι οποίοι είναι υλοποιημένα στα πλαίσια αυτής της διπλωματικής εργασίας προσπάθησα να υλοποιήσω και εγώ έναν υπαρχών αλγόριθμο που έχει σκοπό τον έλεγχο και αντιμετώπιση της συμφόρησης αλλά και τον χειρισμό κόμβων που έχουν σχεδόν εξαντληθεί τα αποθέματα ενέργειας τους. Αυτός ο αλγόριθμος είναι ο HTAP (Hierarchical Tree Alternative Path) [5][6] .

Πιο κάτω θα δούμε αναλυτικότερα μηχανισμούς που αναπτύχθηκαν σε μερικά από τα πρωτόκολλα που υπάρχουν για να κάνουν έλεγχο της συμφόρησης και θα περιγραφούν και οι αλγόριθμοι-μηχανισμοί με τους οποίους ασχολείται αυτή η διπλωματική εργασία.

4.3 Περιγραφή Πρωτόκολλων

Σε αυτό το σημείο θα επεξηγήσω μερικούς μηχανισμούς που έχουν κάποια από τα πρωτόκολλα καθώς και τους μηχανισμούς με τους οποίους ασχολήθηκα.

4.3.1 Congestion Detection and Avoidance (CODA)

Το CODA χρησιμοποιείται σε Ασύρματα Δίκτυα Αισθητήρων που λειτουργούν οδηγούμενα από τα γεγονότα (event-driven). Δηλαδή, παραμένουν σιωπηλά και ξαφνικά γίνονται ενεργά λόγω κάποιου γεγονότος που συνέβη και οι αισθητήρες το αισθάνθηκαν. Ανάλογα με την εφαρμογή διαφέρει και ο βαθμός της συμφόρησης στο δίκτυο. Κατά τη διάρκεια τέτοιων γεγονότων υπάρχει η πιθανότητα να γίνει η μεγαλύτερη συμφόρηση στο δίκτυο και στο διάστημα αυτό οι πληροφορίες που πρέπει να μεταφερθούν έχουν τη μεγαλύτερη σημασία για τους χρήστες. Το CODA προτείνει ένα ενεργειακά αποδοτικό έλεγχο συμφόρησης χρησιμοποιώντας τους εξής τρεις μηχανισμούς:

- I. receiver-based congestion detection
- II. open-loop hop-by-hop backpressure
- III. closed- loop multi-source regulation

Πιο κάτω θα περιγράψουμε αναλυτικότερα τους τρεις μηχανισμούς αφού το CODA δουλεύει με βάση αυτούς.

- **Receiver-based congestion detection**

Το CODA για να κάνει ακριβή εντοπισμό της συμφόρησης σε κάθε παραλήπτη κόμβο με μικρό κόστος συνδυάζοντας την παρούσα και την προηγούμενη κατάσταση του καναλιού μαζί με τον διαθέσιμο αποθηκευτικό χώρο. Τα Ασύρματα Δίκτυα Αισθητήρων πρέπει να γνωρίζουν την κατάσταση του καναλιού, εφόσον το κανάλι είναι ένα μέσο το οποίο μοιράζονται πολλοί και υπάρχει πιθανότητα να παρατηρείται συμφόρηση λόγω της κίνησης μεταξύ άλλων συσκευών που βρίσκονται στη γειτονιά. Για το συνεχή έλεγχο της κίνησης που υπάρχει σε ένα κανάλι χρειάζονται μεγάλα ποσά ενέργειας. Γι' αυτό το CODA χρησιμοποιεί κάποιο σχήμα έτσι ώστε να ενεργοποιεί τον τοπικό

έλεγχου του καναλιού την κατάλληλη στιγμή, για να μειώνει έτσι το κόστος ενώ ταυτόχρονα να κάνει ένα ακριβή υπολογισμό. Εφόσον ανιχνευτεί συμφόρηση οι κόμβοι αισθητήρες ενημερώνουν τους γείτονες τους προς την αντίθετη φορά του Sink για τη συμφόρηση με τη βοήθεια του μηχανισμού backpressure.

- **Open-loop hop-by-hop backpressure**

Στο CODA μόλις κάποιος κόμβος εντοπίσει συμφόρηση στέλνει σε όλους μηνύματα backpressure. Τα μηνύματα αυτά κατευθύνονται προς τους κόμβους πηγές. Και στην περίπτωση πυκνών δικτύων αισθητήρων τα μηνύματα αυτά πιθανό να κατευθυνθούν κατευθείαν στους κόμβους πηγές. Οι κόμβοι που παραλαμβάνουν τέτοια μηνύματα μειώνουν το ρυθμό αποστολής δεδομένων ή πετάνε τα πακέτα βασιζόμενοι στην τοπική τους πολιτική για τη συμφόρηση. Για παράδειγμα, πετάνε τα πακέτα ή χρησιμοποιούν τον αλγόριθμο AIMD(Additive Increase Multiplicative Decrease). Επίσης, κάποιος κόμβος που παραλαμβάνει τέτοιο μήνυμα πρέπει να αποφασίσει κατά πόσο θα μεταφέρει το backpressure μήνυμα παρακάτω βασιζόμενος και πάλι, στις τοπικές του συνθήκες.

- **Closed- loop multi-source regulation**

Στο CODA η ρύθμιση closed-loop λειτουργεί με χαμηλή κλίμακα στο χρόνο και έχει τη δυνατότητα να προσθέσει επιπλέον έλεγχο συμφόρησης σε πολλαπλές πηγές. Όταν ο ρυθμός των γεγονότων στην πηγή είναι πιο κάτω από το κατώφλι του μεγαλύτερου θεωρητικού throughput του καναλιού, τότε ο κόμβος κάνει μόνος του την ρύθμιση που χρειάζεται. Όταν αυτή η τιμή ξεπερνιέται τότε πολύ πιθανόν η πηγή να συμβάλει στη συμφόρηση και γι' αυτό πυροδοτείται ο closed-loop έλεγχος συμφόρησης. Η πηγή μπαίνει στη ρύθμιση που γίνεται από το Sink μόνο αν αυτό το κατώφλι ξεπεραστεί. Σε αυτό το σημείο η πηγή χρειάζεται σταθερό feedback με χαμηλή κλιμάκωση στο χρόνο , για παράδειγμα ACK από το Sink για να διατηρήσει το ρυθμό της. Η παραλαβή ACK από τις πηγές χρησιμοποιείτε ως self-clocking μηχανισμός επιτρέποντας στις πηγές να συντηρούν το τρέχον ρυθμό των γεγονότων τους. Σε αντίθετη περίπτωση που αποτύχει να παραλάβει τα ACK η πηγή αναγκάζεται να μειώσει το δικό της ρυθμό [7].

4.3.2 Mitigating Congestion (Fusion)

Το Fusion ως μηχανισμός συνδυάζει τρεις τεχνικές για έλεγχο της συμφόρησης .

- I. Hop-By-Hop Flow Control
- II. Rate Limiting
- III. Prioritized Mac

Στην συνέχεια θα εξηγήσω αναλυτικότερα τους τρεις αυτούς μηχανισμούς

- **Hop-By-Hop Flow Control**

Σχεδιάστηκε για να εμποδίζει τους κόμβους να μεταδίδουν όταν τα πακέτα τους προορίζονται να πεταχτούν λόγω του ανεπαρκή χώρου στον αποθηκευτικό χώρο του κόμβου αισθητήρα που θα τα παραλάβει. Στο Fusion κάθε κόμβος καθορίζει ένα bit συμφόρησης στο header κάθε πακέτου που πρόκειται να μεταφερθεί. Χρησιμοποιεί τα χαρακτηριστικά του ασύρματου καναλιού που μεταδίδει σε όλους τους κόμβους (broadcast). Κάθε πακέτο υποστηρίζει ανάδραση στη συμφόρηση προς όλους τους κόμβους στην ίδια γειτονιά δηλαδή στην εμβέλεια του κόμβου που το παραλαμβάνει σε κάθε μετάδοση και έτσι δεν υπάρχει ανάγκη για κατηγορηματικά μηνύματα ελέγχου σπαταλώντας έτσι το ήδη περιορισμένο εύρος ζώνης bandwidth που έχουν. Ο hop-by-hop flow control αποτελείται από δύο μέρη: τον εντοπισμό της συμφόρησης και τη μείωση της. Ο εντοπισμός της συμφόρησης βασίζεται στην παρακολούθηση του μεγέθους του αποθηκευτικού χώρου του κόμβου αισθητήρα. Αν ξεπεράσει μια τιμή εφαρμόζεται ένα bit συμφόρησης, αλλιώς αυτό αφαιρείται. Η μείωση της συμφόρησης βασίζεται στον έλεγχο που γίνεται στο ρυθμό μεταφοράς πακέτων των κόμβων αισθητήρων προς τους άμεσα γείτονες τους για να μην τους υπερχειλίζουν. Όταν κάποιος κόμβος παραλαμβάνει κάποιο πακέτο το οποίο έχει το bit συμφόρησης τότε σταματά να μεταφέρει πακέτα. Διαφορετικά η συμφόρηση θα γίνει ακόμα μεγαλύτερη οδηγώντας ακόμα και σε κατάρρευση ολόκληρου του δικτύου

- **Rate Limiting**

Μετρά την κίνηση που υπάρχει στο δίκτυο για από της πηγές που. Κάθε κόμβος αισθητήρας ακούει την κίνηση που μεταφέρει ο πατέρας του για να υπολογίσει το N , που είναι ο συνολικός αριθμός των πηγών που μοναδικά δρομολογούνται

τα πακέτα τους μέσω του πατέρα. Τότε χρησιμοποιείται ένα σχήμα με κουβά κουπονιών (token bucket) για να ρυθμιστεί ο ρυθμός αποστολής του κάθε αισθητήρα. Ο αισθητήρας παίρνει ένα κουπόνι κάθε φορά που ακούει τον πατέρα του να μεταφέρει Ν πακέτα μέχρι να συμπληρώσει τον μέγιστο αριθμό κουπονιών. Ο κάθε αισθητήρας επιτρέπεται να στέλνει μόνο όταν ο αριθμός των κουπονιών είναι πάνω από μηδέν και κάθε αποστολή στοιχίζει ένα κουπόνι. Αυτή η προσέγγιση βάζει κάποιο όριο στον ρυθμό μεταφοράς των κόμβων και τους περιορίζει να στέλλουν με τον ίδιο ρυθμό τον οποίο στέλλουν αυτή που βρίσκονται κάτω από αυτούς ή αλλιώς οι κόμβοι παιδιά.

- **Prioritized Mac**

Εγγυάται ότι οι κόμβοι που είχαν συμφόρηση παίρνουν προτεραιότητα για πρόσβαση στο κανάλι επιτρέποντας τις ουρές αποστολής τους να αδειάσουν. Το επίπεδο MAC παρέχει κάποια βοήθεια στους αισθητήρες έτσι ώστε να αντιδρούν αρκετά γρήγορα σε περίπτωση συμφόρησης. Χρησιμοποιείται ένα κανονικό CSMA(Carrier Sense Medium Access) επιπέδου MAC κάτι που θα εξηγηθεί στο επόμενο κεφάλαιο, με την αλλαγή ότι υλοποιεί ένα σχήμα προτεραιότητας. Σύμφωνα με αυτό το σχήμα, οι κόμβοι που έχουν συμφόρηση έχουν πιο ψηλές προτεραιότητες σε σύγκριση με τους υπόλοιπους κόμβους. Ειδικότερα, κάποιος κόμβος ο οποίος έχει συμφόρηση, το μέγεθος του παραθύρου του είναι το ένα τέταρτο του μεγέθους του παραθύρου κάποιου κόμβου που δεν έχει συμφόρηση κάνοντας πιθανότερο το γεγονός ο κόμβος με τη συμφόρηση να κερδίσει την περίοδο διαμάχης επιτρέποντας την ουρά του να αδειάσει και μεταδίδοντας το μήνυμα συμφόρησης σε όλη τη γειτονιά του [3] .

4.3.3 SenTCP

Το SenTCP είναι ένα open-loop hop-by-hop πρωτόκολλο ελέγχου συμφόρησης και χρησιμοποιεί στην ουσία δύο μηχανισμούς. Πρώτα χρησιμοποιεί τη χωρητικότητα του αποθηκευτικού χώρου , τον μέσο χρόνο άφιξης για κάθε πακέτο τοπικά και τον μέσο χρόνο εξυπηρέτησης του για να μετρήσει το βαθμό της τρέχουσας συμφόρησης τοπικά σε κάθε ενδιάμεσο κόμβο. Η χρησιμοποίηση του χρόνου άφιξης και του χρόνου εξυπηρέτησης μπορεί επίσης να βοηθήσει για να διαχωρίσουμε το λόγο για τον οποίο κάποιο πακέτο χάθηκε στο ασύρματο περιβάλλον. Αυτό θα συμβεί γιατί ένας από τους δύο χρόνους που αναφέραμε ότι χρησιμοποιούνται θα γίνει μεγάλος ή μικρός λόγω του λάθους στο ασύρματο κανάλι. Ο δεύτερος μηχανισμός που υλοποιεί σε συνδυασμό με

τον πρώτο είναι η ανάδραση hop-by-hop. Κάθε ενδιάμεσος κόμβος πρέπει να στήλη ανάδραση στο προηγούμενο κόμβο, το πακέτο ανάδρασης περιέχει τον τοπικό βαθμό συμφόρησης και τη συχνότητα με την οποία γεμίζει ο αποθηκευτικός του χώρος έτσι ώστε αυτός που το παραλαμβάνει να προσαρμόζει ανάλογα την ταχύτητα στο επίπεδο αποστολής του. Με αυτόν τον τρόπο δηλαδή με την hop-by-hop ανάδραση που παρέχει ο μηχανισμός μπορεί να εξαλείψει την συμφόρηση γρήγορα και να μειώσει σημαντικά το χάσιμο πακέτων αφού ελέγχεται η υπερχειλίση των αποθηκευτικών χώρων. Έτσι στην ουσία το SenTCP αποτελείται από τρία μέρη που περιλαμβάνουν στοιχεία για την ανάδραση του σήματος, ρύθμιση του ρυθμού αποστολής και αναμετάδοση του σήματος ανάδρασης. Τα τρία αυτά μέρη αναπτύσσονται σε κάθε ενδιάμεσο κόμβο αισθητήρα. Οι κόμβοι πηγές χρησιμοποιούν μόνο τη ρύθμιση του ρυθμού αποστολής για να μειώσουν το ρυθμό τους όταν παραλάβουν σήμα ανάδρασης. Η παρακολούθηση της χωρητικότητας του αποθηκευτικού χώρου και η διεργασία που γίνεται όταν φτάνει το πακέτο μπορεί να παράγουν το σήμα ανάδρασης προς τους κόμβους πηγές όταν οι συνθήκες ικανοποιούνται. Αλλά όσο οι ενδιάμεσοι τόσο και οι κόμβοι πηγής πρέπει να ρυθμίζουν το ρυθμό αποστολής δεδομένων όταν παραλάβουν σήμα ανάδρασης από τους κόμβους γείτονες τους [6].

4.3.4 HTAP (Hierarchical Tree Alternative Path)

Αυτός είναι ο αλγόριθμος οποίος υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής στην πλατφόρμα υλοποίησης του Contiki. Για το Contiki θα μιλήσω στο επόμενο κεφάλαιο.

Ο HTAP χρησιμοποιεί ένα ευέλικτο και κατανεμημένο πλαίσιο στην προσπάθεια του να μειώσει την συμφόρηση στο δίκτυο και να επιτύχει της ασφαλή μετάδοση των δεδομένων σε Event Based δίκτυα. Για αυτό το λόγο αυτός ο αλγόριθμος δεν κάνει ενέργειες περιορισμού ροής των δεδομένων αλλά προσπαθεί να διατηρεί υψηλή την ροή των δεδομένων καθώς προσπαθεί να μειώσει την απώλεια τους. Ο αλγόριθμος αυτός είναι βασισμένος στην δημιουργία εναλλακτικών διαδρομών από τους κόμβους αισθητήρες source nodes στους κόμβους όπου συλλέγονται τα δεδομένα sink nodes χρησιμοποιώντας την πληθώρα των κόμβων που δεν χρησιμοποιούνται στο δίκτυο με σκοπό την ασφαλή μετάδοση των δεδομένων που συλλέχτηκαν.

Η δημιουργία αυτών των εναλλακτικών διαδρομών όμως συμπεριλαμβάνει και κόμβους οι οποίοι δεν βρίσκονται στην αρχική συντομότερη διαδρομή που δημιουργείται μεταξύ

των κόμβων αισθητήρων και των sinks. Η χρήση αυτών των κόμβων οδηγεί στην ισορροπημένη κατανάλωση ενέργειας στο δίκτυο και αποτρέπεται έτσι η δημιουργία νεκρών σημείων ή τρύπες στο δίκτυο λόγω εξασθενημένων κόμβων και κατά συνέπεια παρατείνεται η ζωή-διάρκεια του δικτύου.

Ο αλγόριθμος αυτός αποτελείται από τέσσερα κύρια μέρη

- I. Flooding για ανακάλυψη του δικτύου και ανακάλυψη των επιπέδων του δικτύου.
- II. Αλγόριθμο ιεραρχικού δένδρου(The Hierarchical Tree Algorithm HT)
- III. Δημιουργία εναλλακτικής διαδρομής (Alternative Path Creation APC)
- IV. Διαχείριση κόμβων με χαμηλή ενέργεια.

- Flooding για ανακάλυψη του δικτύου και ανακάλυψη των επιπέδων του δικτύου.

Σε αυτό το μέρος κάθε κόμβος ανακαλύπτει τους γειτονικούς του κόμβους και ενημερώνει τους πίνακες με τους γείτονες του και επιπρόσθετα οι κομβοί διαχωρίζονται σε επίπεδα από τους κόμβους αισθητήρες μέχρι τα sinks αρχίζοντας οι πηγές είναι οι κόμβοι με επίπεδο 0. Η διαδικασία αυτή σταματά όταν όλοι οι κόμβοι έχουν μπει σε επίπεδα και οι κομβοί Sink έχουν αναγνωρισθεί στο δίκτυο

- Αλγόριθμο ιεραρχικού δένδρου(The Hierarchical Tree Algorithm HT)
Ένα ιεραρχικό δένδρο υλοποιείται ξεκινώντας από τον κόμβο αισθητήρα. Κάθε κόμβος πρέπει να αποφασίσει σε ποιόν από τους κόμβους γείτονες του πρέπει να προωθεί τα πακέτα και αυτή είναι η δουλειά αυτής της φάσης του αλγορίθμου. Όταν μέσα από μία συνάρτηση που επιλέγει με βάση τον κόμβο με το ψηλότερο επίπεδο και τη χαμηλότερη συμφόρηση, η σύνδεση μεταξύ αποστολέα και παραλήπτη υλοποιείται με 2 way handshake και με την ανταλλαγή αυτών των πακέτων γίνεται και η ενημέρωση του αποστολέα για το πόση συμφόρηση υπάρχει στον παραλήπτη. Όταν το επίπεδο συμφόρησης ενός κόμβου παραλήπτη είναι πάνω από ένα κατώφλι τότε υλοποιείται ο APC.
- Δημιουργία εναλλακτικής διαδρομής (Alternative Path Creation APC)

Σε αυτό το σημείο για να αποφευχθεί η συμφόρηση κάθε κόμβος που πάει προς υπερφόρτωση που έχει ως αποτέλεσμα τη συμφόρηση σε ένα δίκτυο στέλλει ένα μήνυμα και ενημερώνει τον κόμβο ο οποίος του αποστέλλει τα δεδομένα. Έτσι με την σειρά του ο αποστολέας σταματά την αποστολή των πακέτων σε αυτών τον κόμβο και ψάχνει στους πίνακες με τους γείτονες του να βρει τον λιγότερο φορτωμένο κόμβο ούτως ώστε να συνεχίσει την μετάδοση πακέτων σε αυτόν τον κόμβο και στον πίνακα με τους γείτονες του ανανεώνει την κατάσταση συμφόρησης του παλιού κόμβου ούτως ώστε να μην γίνει ξανά επιλογή του ίδιου κόμβου. Αυτή η δυναμική αλλαγή των αποδεχτών οδηγεί στην δημιουργία νέων διαδρομών από τους κόμβους αισθητήρες στα sinks. Ο συνδυασμός των δυο αλγορίθμων οδηγεί στην υλοποίηση του HTAP Hierarchical Tree Alternative Path (HTAP).

- Διαχείριση κόμβων με χαμηλή ενέργεια.
Όταν ένας κόμβος πάει να ξεμείνει από ενέργεια τότε αμέσως αφαιρείται από το δίκτυο και ενημερώνονται οι πίνακες των γειτόνων του. Αυτή η διαδικασία είναι όσο πιο απλή γίνεται γιατί μπορεί να συμβεί σε στιγμή κρίσης του δικτύου [6] [5].

Κεφάλαιο 5

Contiki Λειτουργικό Σύστημα για Ασύρματα Δίκτυα Αισθητήρων

-
- 5.1 Γενικά
 - 5.2 Τρόποι Επικοινωνίας
 - 5.3 RDC Drivers
 - 5.4 MAC Drivers
 - 5.5 Event Timers
-

5.1 Γενικά

Το Contiki είναι ένα ανοιχτού κώδικα, εξαιρετικά φορητό και πολλαπλών διεργασιών λειτουργικό σύστημα το οποίο χρησιμοποιήθηκε σε αυτή τη διπλωματική εργασία. Έχει πάρει αυτό το όνομα του από την πλωτή σχεδία Kon –Tiki η οποία διέσχισε των Ειρηνικό Ωκεανό με ελάχιστους πόρους κάτι που είναι και η κεντρική ιδέα του λειτουργικού συστήματος αυτού. Το λειτουργικό αυτό έχει αναπτυχθεί από επαγγελματίες προγραμματιστές αλλά και ακαδημαϊκούς υπό την επίβλεψη του Adam Dunkels από το Σουηδικό Ινστιτούτο Πληροφορικής. Ο κύριος στόχος του λειτουργικού αυτού είναι η χρήση σε ενσωματωμένα δικτυακά συστήματα με στόχο την υψηλή απόδοση σε ζητήματα μνήμης αλλά και στα ασύρματα δίκτυα αισθητήρων.

Τα λειτουργικό είναι σχεδιασμένο για συστήματα μικροελεγκτών με περιορισμένες δυνατότητες όσων αφορά την μνήμη. Μια τυπική αναφορά στις δυνατότητες μνήμης του Contiki είναι πως κάνει χρήση 2 kilobytes για RAM και 40 kilobytes για ROM

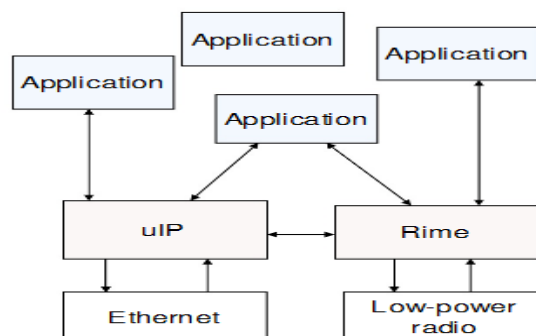
Το λειτουργικό αυτό μπορεί να τρέξει σε μια μεγάλη ποικιλία πλατφόρμων από ενσωματωμένους μικροελεγκτές όπως ο MSP430 ο οποίος είναι ένας μικροελεγκτής μικτού σήματος σχεδιασμένος για να έχει χαμηλό κόστος και χαμηλή κατανάλωση ενέργειας στις ενσωματωμένες εφαρμογές που χρησιμοποιείται αλλά μέχρι και σε πολιούς ηλεκτρονικούς υπολογιστές. Είναι γραμμένο στην γλώσσα προγραμματισμού C. Το λειτουργικό αυτό έχει εφαρμοστή σε πολλά έργα και εφαρμογές όπως σε δίκτυα παρακολούθησης για ασφάλεια, στην ανίχνευση εισβολής σε καθορισμένη περιοχή,

στην ανίχνευση πυρκαγιών σε οδικές σήραγγες αλλά ακόμη και στην παρακολούθηση και ερευνά του νερού τις Βαλτικής θάλασσας.

5.2 Τρόποι Επικοινωνίας

Το Contiki μπορεί να παρέχει δύο στοίβες επικοινωνίας. Επικοινωνία με της στοίβα uIP που στην ουσία είναι σε TCP/IP με διευθύνσεις IPv4 αλλά και IPv6 και επιπλέον έχει και την δική του μορφή επικοινωνίας το Rime stack που είναι ένας πιο “οικονομικός” τρόπος.

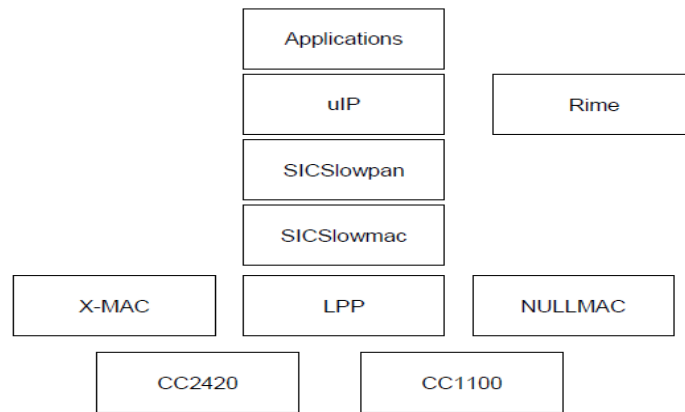
Οι εφαρμογές μπορούν να χρησιμοποιήσουν μια από τις δύο μορφές ή και τις δύο ή και καμία.



Σχημα 5.1 Σχηματική αναπαράσταση των στοιβών επικοινωνίας

Στην uIP στοίβα οι διεργασίες ανοίγουν TCP και UDP συνδέσεις με τις συναρτήσεις που παρέχει το λειτουργικό `tcp_connect()`, `tcp_listen()`, `udp_new()`.

Σε αυτή την μορφή επικοινωνίας έχει δύο APIs το απλό event-driven API και το Protosockets το οποίο είναι βασισμένο σε socket-like προγραμματισμό σε protothreads. Το event - driven είναι καταλληλότερο για μικρά προγράμματα ενώ το Protosockets δουλεύει καλύτερα για μεγαλύτερα προγράμματα.



Σχημα 5.2 Η θέση του uIP στην αρχιτεκτονική των WSN

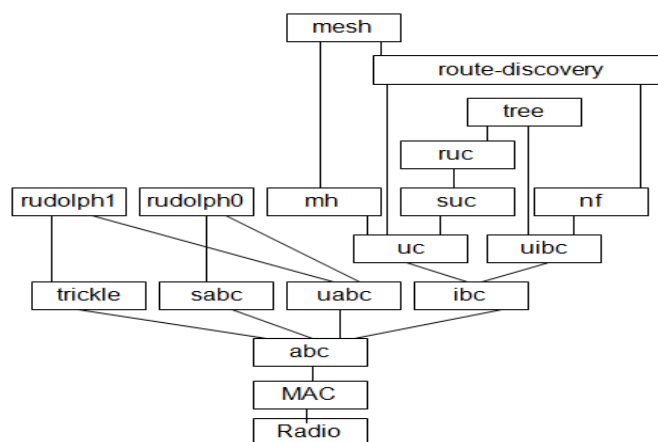
Σε αυτή τη διπλωματική εργασία εγώ δεν χρησιμοποιώ την uIP στοίβα επικοινωνίας αλλά την στοίβα Rime.

Η στοίβα Rime είναι μια μορφή ραδιοεπικοινωνίας με χαμηλή κατανάλωση ενέργειας και αποτελούμενη από διάφορα στρώματα (Layers). Διάφορα αρχέτυπα επικοινωνίας παρέχονται για την επικοινωνία και κυμαίνονται από ένα απλό best - effort local area broadcast μέχρι και αξιόπιστο Network Flooding τα λεγόμενα modules. Τα modules τα οποία υλοποιεί το Rime Stack είναι οι πιο κάτω και θα εξηγηθούν περεταίρω σε επόμενο κεφάλαιο όσα έχουν χρησιμοποιηθεί :

- Anonymous best-effort single-hop broadcast (abc)
- Identified best-effort single-hop broadcast (ibc)
- Stubborn identified best-effort single-hop broadcast (sibc)
- Best-effort single-hop unicast (uc)
- Stubborn best-effort single-hop unicast (suc)
- Reliable single-hop unicast (ruc)
- Unique anonymous best-effort single-hop broadcast (uabc)
- Unique identified best-effort single-hop broadcast (uibc)
- Best-effort multi-hop unicast (mh)
- Best-effort multi-hop flooding (nf)

- Reliable multi-hop flooding (trickle)
- Hop-by-hop reliable data collection tree routing (tree)
- Hop-by-hop reliable mesh routing (mesh)
- Best-effort route discovery (route-discovery)
- Single-hop reliable bulk transfer (rudolph0)
- Multi-hop reliable bulk transfer (rudolph1)

Κάθε μία από αυτές τις λειτουργίες που παρέχονται είναι σχετικά απλές αφού αποτελούνται από μεταγλωττισμένο κώδικα μικρού μεγέθους από 114 μέχρι 598 bytes. Η απλότητα τους οφείλεται στην στρωματοποίηση αφού τα ποιο σύνθετα υλοποιούνται με την χρήση των ποιο απλών modules. Κάθε στρώμα μπορεί να προσθέσει το δικό του Header που είναι μικρό σε μέγεθος αφού αποτελείται μόνο από μερικά bytes. Παρόλα αυτά όμως δεν αποτελεί ένα πλήρως αρθρωτό πλαίσιο με την χρήση αυτών των modules γιατί κάτι τέτοιο απαιτεί μεγάλη πολυπλοκότητα και για αυτό χρησιμοποιεί αυστηρή στρωματοποίηση.



Σχημα 4.3 Ο χαρτης των modules του Rime Stack

Τα κανάλια τα οποία χρησιμοποιούνται για επικοινωνία χαρακτηρίζονται από 16 bits το κάθε ένα. Οι κόμβοι στο δίκτυο για να καταφέρουν να επικοινωνήσουν πρέπει να συμφωνούν στο module και στο κανάλι το οποίο θα χρησιμοποιήσουν. Για παράδειγμα για το module uc στο κανάλι 5 πρέπει και οι υπόλοιποι κόμβοι που θα επικοινωνήσουν μαζί του να χρησιμοποιούν το module uc στο κανάλι 5. Τα κανάλια που

χρησιμοποιούνται είναι μικρότερα του 128 αφού τα αλλά είναι κρατημένα από το ίδιο το σύστημα.

5.3 RDC Drivers

Στα δίκτυα χαμηλής κατανάλωσης ενέργειας ο πομποδέκτης πρέπει να σβήνει όσο το δυνατό περισσότερο ούτως ώστε να έχουμε εξοικονόμηση ενέργειας. Στο Contiki αυτό γίνεται από το RDC(Radio Duty Cycling) layer. Το λειτουργικό παρέχει αρκετούς RDC μηχανισμούς οι οποίοι έχουν ο κάθε ένας ξεχωριστά τα δικά τους χαρακτηριστικά και είναι χάρη σε αυτούς που μπορούν οι κόμβοι να επικοινωνούν έχοντας την περισσότερη ώρα κλειστό τον πομποδέκτη τους.

Μερική από τους μηχανισμούς αυτούς είναι το ContikiMAC, X-MAC, CX-MAC, LPP και το NullRDC. Το ContikiMAC είναι ο προκαθορισμένος μηχανισμός που χρησιμοποιείται και έχει μεγάλη απόδοση όσο αφορά την εξοικονόμηση ενέργειας αλλά κατά κάποιο τρόπο είναι προσαρμοσμένος για το πρωτόκολλο επικοινωνίας IEEE 802.15.4 και τον πομποδέκτη CC2420. Το X-MAC είναι ένας παλιότερος μηχανισμός ο οποίος δεν έχει όση απόδοση έχει το ContikiMAC στην εξοικονόμηση ενέργειας αλλά δεν έχει τόσο αυστηρές απαιτήσεις συγχρονισμού. Το CX –MAC (Compatibility X-MAC) είναι μία υλοποίηση του X-MAC η οποία έχει πιο χαλαρό συγχρονισμό επομένως επιτρέπει επικοινωνία σε ένα πιο μεγάλο εύρος συχνοτήτων. Το LPP (Low-Power Probing) είναι πρωτόκολλο το οποίο έχει να κάνει κυρίως με των παραλήπτη και τέλος το NullRDC είναι ένα κενό RDC πρωτόκολλο το οποίο ποτέ δεν κλείνει τον πομποδέκτη και συνήθως χρησιμοποιείται απλά ως βάση για να γίνει σύγκριση με άλλα RDC πρωτόκολλα.

Γενικά η λειτουργία των πρωτοκόλλων αυτών είναι να κρατούν κλειστό τον πομποδέκτη όσο το δυνατό πιο πολύ και περιοδικά να ελέγχουν στην συχνότητα αν υπάρχει δραστηριότητα. Όταν ανιχνευτεί κίνηση στην συχνότητα τότε ο πομποδέκτης παραμένει ενεργός για να παραλάβει το πακέτο.

Η συχνότητα με την οποία γίνεται ο έλεγχος στο κανάλι δίνετε σε Hz(Hertz) και καθορίζει τον αριθμό των ελέγχων στο κανάλι ανά δευτερόλεπτο με προκαθορισμένη συχνότητα τα 8 Hz. Συνήθως η συχνότητα αυτή είναι αριθμός δύναμη του 2 και ορισμένες συχνότητες είναι 2, 4, 6, 8 και 16.

Στην δική μου υλοποίηση εγώ θα κάνω χρήση του ContikiMAC γιατί χρειάζομαι όσο το δυνατό καλύτερη απόδοση στην εξοικονόμηση ενέργειας.

5.4 MAC Drivers

Το MAC(Medium Access Control) layer βρίσκεται πάνω από το RDC layer και είναι υπεύθυνο για να αποφεύγονται σύγκρουσης πακέτων στις συχνότητες που χρησιμοποιούνται αλλά σε περίπτωση που υπάρχουν είναι υπεύθυνο για την αναμετάδοση των πακέτων αυτών.

Το MAC επίπεδο παίρνει πακέτα από το RDC και χρησιμοποιεί το RDC επίπεδο για να στείλει πακέτα στο δίκτυο. Στα WSN υπάρχει μια πολύ σημαντική ιδιαιτερότητα που πρέπει να ληφθεί υπόψη στη σχεδίαση ενός MAC πρωτοκόλλου και αυτή είναι η περιορισμένη ενέργεια των κόμβων. Η λήψη και μετάδοση μηνυμάτων είναι εκ των πραγμάτων η πιο ακριβή από άποψη ενέργειας διαδικασία που καλείται να εκτελέσει ένας κόμβος, οπότε ένα MAC πρωτόκολλο για WSN πρέπει να έχει ως πρωτεύων στόχο την ελαχιστοποίηση του χρόνου λειτουργίας του κυκλώματος ραδιοπομπού κάτι που γίνεται και στο RDC.

Υπάρχουν τέσσερις παράγοντες που προκαλούν σπατάλη ενέργειας κατά την επικοινωνία [9].

- Συγκρούσεις: Οι συγκρούσεις πακέτων προκαλούν την αμοιβαία καταστροφή τους, με αποτέλεσμα η ενέργεια που δαπανήθηκε για την αποστολή και λήψη τους να πηγαίνει χαμένη.
- Τυχαία ακούσματα - Overhearing: Τα μηνύματα συνήθως έχουν μια πηγή και έναν προορισμό. Λόγω της φύσης των ηλεκτρομαγνητικών σημάτων οι κόμβοι λαμβάνουν μηνύματα που δεν τους αφορούν. Παρόλα αυτά υπάρχουν περιπτώσεις όπου το overhearing μπορεί να χρησιμοποιηθεί με ωφέλιμο τρόπο (π.χ. σε data – centric αλγόριθμους).
- Επιπρόσθετος φόρτος - Overhead πρωτοκόλλου: Το overhead εισάγεται από πλαίσια ελέγχου, σχετικά με το MAC, όπως από κεφαλίδες (headers) ή επίμετρα (trailers) που προστίθενται στα πακέτα.
- Αναμονή Ακούσματος - Idle Listening: Ένας κόμβος που βρίσκεται σε κατάσταση ετοιμότητας περιμένοντας την εμφάνιση μηνύματος, ξοδεύει άσκοπα ενέργεια κατά τη διάρκεια της αναμονής.

Η ταυτόχρονη αντιμετώπιση των παραπάνω παραγόντων είναι αδύνατη. Τα περισσότερα MAC πρωτόκολλα για WSN, επικεντρώνονται συνήθως στην ελαχιστοποίηση των επιπτώσεων ενός ή δυο παραγόντων από τους ποιο πάνω.

Το λειτουργικό παρέχει δύο MAC Drivers το CSMA(Carrier Sense Medium Access) και το NullMac το οποίο το οποίο δεν κάνει τίποτα από πλευράς MAC-level επεξεργασίας.

Ο προκαθορισμένος μηχανισμός είναι το CSMA και είναι ο μοναδικός μηχανισμός που παρέχει αναμετάδοση πακέτων σε περίπτωση σύγκρουσης και είναι αυτός ο οποίος χρησιμοποιώ για να αναπτύξω τον αλγόριθμο στα πλαίσια αυτής της διπλωματικής εργασίας.

5.5 Event Timers

Τα Event Timers είναι ένας τρόπος με τον οποίο μπορεί το λειτουργικό να δημιουργήσει συμβάντα σε καθορισμένο χρόνο. Ένα χρονόμετρο τέτοιου είδους θα δώσει ένα συμβάν στην διεργασία η οποία το έχει καλέσει και έχει καθορίσει τον χρόνο του την στιγμή που ο χρόνος του χρονομέτρου θα τελειώσει. Ένα event timer δηλώνεται ως ένα struct etimer και κάθε πρόσβαση σε αυτό μπορεί να γίνει με ένα δείκτη στο δηλωμένο event timer.

Το Contiki παρέχει ακόμα 3 τύπους από χρονόμετρα.

- struct timer είναι παθητικό χρονόμετρο και έχει να κάνει μόνο με το πότε θα τελειώσει ο χρόνος του.
- struct ctimer είναι και αυτό ένα ενεργό χρονόμετρο οποίο καλεί μία συνάρτηση όποτε τελειώσει ο χρόνος του. Αυτό χρησιμοποιείται από rime stack
- struct rtimer είναι και αυτό ένα ενεργό χρονόμετρο το οποίο είναι πραγματικού χρόνου και έχει την δυνατότητα να καλέσει μία συνάρτηση σε συγκεκριμένη ώρα.

Κεφάλαιο 6

Υλοποίηση αλγορίθμου HTAP σε Contiki

- 6.1 Γενικά για υλοποίηση
 - 6.2 Φάση πρώτη ανακάλυψη δικτύου και καθορισμός επιπέδων
 - 6.3 Φάση δεύτερη υλοποίηση ιεραρχικού δένδρου
 - 6.4 Φάση τρίτη αποστολή δεδομένων και εύρεση εναλλακτικού μονοπατιού
-

6.1 Γενικά για υλοποίηση

Σε αυτό το κεφάλαιο θα γίνει μία εκτενής περιγραφή για το πώς υλοποιήθηκε ο αλγόριθμος αυτός επίσης θα γίνει επεξήγηση συναρτήσεων που χρησιμοποιήθηκαν .

Ο αλγόριθμος αυτός όπως αναφέρθηκε στο προηγούμενο κεφάλαιο έχει ως στόχο να καταφέρει να αντιμετωπίσει το πρόβλημα της συμφόρησης στα ασύρματα δίκτυα αισθητήρων αλλά και να επιτύχει την παράταση της ζωής του δικτύου κάνοντας σωστή διαχείριση της ενέργειας των κόμβων.

Στο προηγούμενο κεφάλαιο γίνεται μια επεξήγηση για το λειτουργικό Contiki το οποίο είναι το λειτουργικό στο οποίο έγινε η υλοποίηση του αλγορίθμου. Συγκεκριμένα το λογισμικό αυτό διαθέτει πολλές εκδόσεις αφού συνεχώς αναβαθμίζεται, αλλά στην υλοποίηση αυτή έγινε χρήση της contiki-2.x έκδοσης του contiki-2.4 αφού όταν έγινε προσπάθεια για να χρησιμοποιηθεί η έκδοση contiki-2.5.rc1 η οποία είναι και η πιο πρόσφατη μέχρι στιγμής αντιμετωπίστηκαν διάφορα προβλήματα τόσο με τα MAC και RDC πρωτόκολλα όσο και με βιβλιοθήκες και συναρτήσεις που παρέχει το λειτουργικό.

Στην περιγραφή του αλγορίθμου ο αλγόριθμος διαχωρίζεται σε τρία στάδια. Με βάση αυτά τα στάδια θα εξηγήσω και εγώ την υλοποίηση μα αρχικά ας ξεκινήσουμε λίγο πιο γενικά .

Κάθε κόμβος σε αυτήν την υλοποίηση τρέχει με το ίδιο λειτουργικό και τον ίδιο αλγόριθμο έτσι ότι αναφέρεται εννοείται ότι είναι το ίδιο σε όλους τους κόμβους του δικτύου αλλά υλοποιείται για κάθε κόμβο τοπικά ξεχωριστά .

Στην υλοποίηση του αλγορίθμου έγινε χρήση της `time stack` για επικοινωνία και σύμφωνα με τους στόχους που καλείται να υλοποιήσει πρέπει να κάνει χρήση κάποιων από τα RDC αλλά και MAC πρωτόκολλα του λειτουργικού συστήματος τα οποία αναφέρθηκαν στο προηγούμενο κεφάλαιο.

Αρχικά όσον αφορά το MAC είχε γίνει χρήση του προκαθορισμένου από το Contiki CSMA το οποίο ικανοποιεί πλήρως της απαιτήσεις αφού παρέχει αναμεταδώσεις πακέτων σε περίπτωση συγκρούσεων έτσι πετυχαίνεται ο στόχος του αλγορίθμου σε αυτό το επίπεδο που είναι όσο το δυνατό λιγότερες απώλειες δεδομένων. Τώρα όσον αφορά τα RDC πρωτόκολλα έγινε μια μικρή αλλαγή κατά την διάρκεια της υλοποίησης του αλγορίθμου αφού η αρχική μας υπόθεση για το ποιο έπρεπε να χρησιμοποιηθεί δεν βγήκε ορθή. Αρχικά θεωρήσαμε ότι έπρεπε να γίνει χρήση του X-MAC πρωτοκόλλου το οποίο παρέχει μια πολύ καλή διαχείριση όσον αφορά την ενέργεια του δικτύου αλλά και στις συγκρούσεις πακέτων. Τελικά δεν προχωρήσαμε στην υλοποίηση με αυτό γιατί στην πορεία και μέσα από διάφορες δοκιμές και διαφορετικές υλοποιήσεις μερικών κομματιών του αλγορίθμου παρατηρήθηκε ότι δεν μπορούσε να παρέχει πλήρη κάλυψη από τις συγκρούσεις έτσι προτιμήθηκε το ContikiMAC το οποίο παρέχει και αυτό μεγάλη αποδοτικότητα στην διαχείριση ενέργειας αλλά χωρίς συγκρούσεις και απώλειες πακέτων.

Στην συνέχεια είναι βιβλιοθήκες που χρησιμοποιήθηκαν μετά θα εξηγήσω τις δομές που χρησιμοποιούνται και τέλος τα τρία μέρη του αλγόριθμου και πως υλοποιήθηκαν.

Η βασικότερες βιβλιοθήκες οι οποίες χρησιμοποιούνται είναι η `contiki.h` η οποία είναι αυτή η οποία συμπεριλαμβάνει γενικά την όλη δομή του `contiki` λειτουργικού και ποια έκδοση χρησιμοποιείται και η `time.h` η οποία περιέχει συναρτήσεις οι οποίες χρησιμοποιούνται για την επικοινωνία στον αλγόριθμο. Γίνετε χρήση της βιβλιοθήκης `lib/list.h` η οποία παρέχει την δυνατότητα δημιουργίας μια λίστας αντικειμένων και συναρτήσεις που με τις οποίες μπορεί να γίνει επεξεργασία της λίστας αυτής.

Επίσης μια βιβλιοθήκη η οποία χρησιμοποιείται `lib/memb.h` και είναι κατά κάποιο τρόπο αλληλένδετη με την προηγούμενη αφού παρέχει την δυνατότητα δέσμευσης μνήμης η οποία δηλώνετε σαν κάποιου τύπου στην περίπτωση μας οι κόμβοι τις λίστας και κάνει δέσμευση μνήμης για αυτούς.

Η βιβλιοθήκη `lib/random.h` η οποία είναι για την παράγωγή τυχαίων αριθμών από τις συναρτήσεις της. Η χρήση της γίνεται στα σημεία όπου υπάρχουν `timers` για τον

υπολογισμό της διάρκειας τους. Ακόμα μια βιβλιοθήκη που καλείται είναι η `packetqueue.h` η οποία υλοποιεί μία δομή ουράς στην οποία μπορείς να βάλεις μέσα πακέτα που έρχονται στον κάθε κόμβο δηλαδή να τη χρησιμοποιήσεις σαν `buffer`. Επίσης παρέχει κάποιες συναρτήσεις με τις οποίες μπορείς να χειριστής αυτή την ουρά ανάλογα με το πώς θέλεις να την χειριστής.

Μέσα από αυτές τις βιβλιοθήκες έκανα χρήση αρκετών δομών που υλοποιούσαν σε διάφορα σημεία του αλγορίθμου έτσι τώρα θα επεξηγήσω τις δομές που χρησιμοποιήθηκαν για την υλοποίηση πριν συνεχίσω με την περιγραφή των σταδίων του αλγορίθμου.

Οι δομές οι οποίες χρησιμοποιώ είναι μια ουρά τύπου `PACKETQUEUE` η οποία υλοποιείται με την βοήθεια της δομής της λίστας η οποία ορίζεται από την ανάλογη βιβλιοθήκη και ονομάζεται `q`. Το μέγεθος της ουράς αυτής καθορίζεται από μια σταθερά την `MAX_PACKETQUEUE`. Επίσης γίνεται χρήση μια λίστας και στην υλοποίηση του πίνακα με τους γείτονες κάθε κόμβου. Η δομή αυτή ονομάζεται `neighbor_list` και κάθε κόμβος της δομής είναι τύπου `neighbor` τύπος ο οποίος όρισα για να περιγραφεί τα απαραίτητα στοιχεία τα οποία χρειάζεται να ξέρει κάθε κόμβος για τους γείτονες του. Τα στοιχεία αυτά είναι ένα πεδίο δείκτης που δείχνει στο επόμενο στοιχείο της λίστας την διεύθυνση του κόμβου στη ουσία τον αριθμό του κόμβου, το `level` του κόμβου σύμφωνα με τον αλγόριθμο ένα πεδίο που καθορίζει αν ο κόμβος είναι `Sink` η αν είναι συνδεδεμένος με το `Sink` άρα `Sink_neighbor`. Το μέγεθος αυτής της λίστας καθορίζεται από μια σταθερά την `MAX_NEIGHBORS` η οποία παίρνει ως τιμή τον μέγιστο αριθμό γειτονικών κόμβων που μπορεί να φυλάξει τα στοιχεία τους κάθε κόμβος.

Για επικοινωνία χρησιμοποιούνται δύο τύποι μηνυμάτων τους οποίους δεν παρέχει το λειτουργικό αλλά καθορίζουμε εμείς την δομή τους. Ο ένας τύπος είναι τα `broadcast_message` που είναι τα πακέτα που αποστέλλονται με `broadcasting` όπως λέει και το όνομα τους, έχουν μέσα τα πεδία `seqno` που είναι ο αριθμός κάθε πακέτου, το πεδίο `level` που είναι όπως λέει και το όνομα το επίπεδο του κόμβου που το αποστέλλει, και το `sender_addr` την διεύθυνση του κόμβου που το αποστέλλει. Ο δεύτερος τύπος είναι τα πακέτα τα οποία αποστέλλονται με `unicast` μέθοδο και είναι τα τύπου `ack_message`. Τα μηνύματα αυτά περιέχουν τα εξής πεδία `type` που είναι ο τύπος του `unicast` μηνύματος βασικά καθορίζει τον σκοπό για τον οποίο αποστέλλεται το μήνυμα,

το `m_addr` πού είναι η διεύθυνση του αποστολέα του πακέτου, το `sink_connected` που καθορίζει αν ο κόμβος αποστολέας είναι συνδεδεμένος με το `sink`, το `buffer_state` που μέσα έχει την κατάσταση την μνήμης του κόμβου αποστολέα, το `level` που είναι το επίπεδο του κόμβου αποστολέα και τέλος το πεδίο `sink` που καθορίζει κατά πόσο ο κόμβος αποστολέας είναι Sink.

Και τέλος μια δομή που υλοποιείται μόνο στους κόμβους αποδέκτες τα `sink` είναι ένας πίνακας ο οποίος καθορίζει πόσοι κόμβοι μπορούν να συνδεθούν στο `sink` με βάση και πάλι μία σταθερά `MAX_SINK_NEIGHBORS`.

Ακολουθεί η ανάλυση του κώδικα που γράφτηκε για την υλοποίηση του αλγορίθμου και στην επεξήγηση της υλοποίησης των τριών φάσεων του αλγορίθμου.

Αρχικά ο τρόπος με τον οποίο δημιουργείς ένα πρόγραμμα στο Contiki δηλώνεις μια διεργασία μέσω της συνάρτησης `PROCESS` (όνομα της διεργασίας) και με την συνάρτηση `AUTOSTART_PROCESSES` (&όνομα της διεργασίας) δηλώνει ποια διεργασία να ξεκινήσει όταν φορτωθεί αυτό το `module`. Έπειτα μέσα στο `PROCESS_THREAD` καθορίζεις την διεργασία σου και με τις συναρτήσεις `PROCESS_BEGIN()` και `PROCESS_END()` καθορίζουν το πότε ξεκινά και πότε τελειώνει μία διεργασία και με την συνάρτηση `PROCESS_EXITHANDLER()` καθορίζεις με ποια ενέργεια η διεργασία βγαίνει από την εκτέλεση.

Κάθε κόμβος έχει τις δικές του `global` μεταβλητές οι οποίες δηλώνονται στην αρχή. Οι μεταβλητές αυτές έχουν κάποια ιδιότητα ως σημαίες για κάποια σημεία είτε είναι μετρητές αλλά και κάποιες άλλες οι οποίες απλά κρατούν μερικά στοιχεία του κόμβου τα οποία θέλουμε να μπορούμε να έχουμε πρόσβαση από παντού. Αυτές οι μεταβλητές θα επεξηγηθούν κατά την διάρκεια στα σημεία που χρησιμοποιούνται όπως και διαφορές συναρτήσεις κάτω από αυτές.

Στην αρχή στο σημείο που γράφεται το κυρίως μέρος της διεργασίας δηλώνονται οι διαφορές μεταβλητές οι οποίες όσες δεν είναι γενικής εμβέλειας είναι `static` για λόγους εξοικονόμησης μνήμης. Μετά καθορίζεται το πού ξεκινά η διεργασία να εκτελείται και τότε αρχίζει και το πρώτο κομμάτι του αλγορίθμου. Αλλά πριν από αυτό είναι ανοίγονται όλες οι μορφές επικοινωνίας που θα χρειαστούν με τις ανάλογες συναρτήσεις που παρέχει η `time stack` επικοινωνίας αλλά αρχικοποιούνται και οι ουρά μνήμη για τα πακέτα αλλά και. Η συναρτήσεις αυτές είναι η `broadcast_open()`, `unicast_open`, και `trickle_open` οι οποίες ανοίγουν την επικοινωνία αλλά γίνεται και ο

καθορισμός του καναλιού στο οποίο θα γίνεται η επικοινωνία. Οι πρώτες δύο είναι ξεκάθαρο από το όνομα τι κάνουν αλλά για την trickle αν και αναφέρεται στο προηγούμενο κεφάλαιο στην εξήγηση του rime stack θα αναφερθεί ξανά ότι είναι μια συνάρτηση που υλοποιεί ένα αξιόπιστο multi-hop broadcast πλημμύρισμα στο δίκτυο.

Στην συνέχεια με έλεγχο της ταυτότητας του κόμβου μπορεί ο προγραμματιστής να καθορίσει τους κόμβους πηγές αλλά και τους κόμβους αποδέκτες ενημερώνοντας την τιμή της global μεταβλητής του κόμβου ανάλογα. Αν ο κόμβος δηλωθεί ως κόμβος πηγή τότε ρητά το επίπεδο του καθορίζεται 0. Έπειτα αν ο κόμβος είναι αποδέκτης τότε αρχικοποιείται ο πίνακας που καθορίζει τους πόσους γείτονες μπορεί να έχει το Sink.

6.2 Φάση πρώτη: Ανακάλυψη δικτύου και καθορισμός επιπέδων

Κάπου εδώ είναι που ξεκινά και η πρώτη φάση του αλγορίθμου που είναι η αναγνώριση του δικτύου και ο καθορισμός επιπέδων των κόμβων. Κάθε μία από τις φάσεις έχει την δική της επανάληψη η οποία καθορίζεται από μία σημαία που είναι global μεταβλητή και σταματά μόνο όταν τελειώσει αυτά πρέπει να κάνει ο κόμβος και αλλάξει η τιμή της σημαίας. Άμεσος μετά την επανάληψη είναι η αρχικοποίηση του χρονομέτρου που στην περίπτωση αυτή γίνεται χρήση ενός event timer και ο χρόνος υπολογίζεται με μία φόρμουλα οι οποία δίνει αποτελέσματα με τιμές από 18 μέχρι 32 δευτερόλεπτα κάθε φορά αλλά με διαφορετικό συγχρονισμό για κάθε κόμβο αφού γίνεται πολλαπλασιασμός με τον αριθμό του κόμβου κάθε φορά. Και έπειτα καλείται η συνάρτηση `PROCESS_WAIT_EVENT_UNTIL()` η οποία παγώνει την διεργασία κάθε φορά μέχρι αν παρουσιαστεί κάποιο event το οποίο event θα δοθεί με την λήξη του event timer.

Σε πρώτη φάση η πηγή πρέπει να αναγνωριστεί και να είναι αυτή η οποία θα αρχίσει την διαδικασία με broadcast και όχι οποιοσδήποτε άλλος κόμβος. Έτσι με έλεγχο αναγνωρίζεται αν είναι πηγή με βάση την σημαία της κάθε πηγής και επειδή μπορεί πριν τελειώσει αυτή η φάση να έρθει ξανά η σειρά της πηγής να τρέξει αλλά δεν πρέπει να κάνει ξανά broadcast έτσι αυτό εξασφαλίζεται με ένα μετρητή οποίος δεν αφήνει την πηγή να το επαναλάβει. Ακόλουθος η πηγή ετοιμάζει το μήνυμα το οποίο είναι να στήλη και το φορτώνει στην μνήμη με την συνάρτηση `packetbuf_copyfrom()` και το στέλλει με την συνάρτηση `broadcast_send()` και μέσα το όνομα της σύνδεσης την οποία αρχικοποίησε στην αρχή.

Όταν ένα πακέτο broadcast παραληφθεί από τους γείτονες τότε καλείτε η συνάρτηση `broadcast_recv()` η οποία έχει σαν παράμετρο την σύνδεση που έγινε για broadcast και την διεύθυνση του αποστολέα. Είναι μία από τις callback συναρτήσεις που παρέχει το broadcast module , callbacks παρέχουν όλες οι μορφές επικοινωνίας και καλούνται όταν ένα πακέτο της αντίστοιχης μορφής παραλειφθεί από κάποιο κόμβο. Έτσι όταν η συνάρτηση αυτή καλεστεί μετά την παραλαβή του πακέτου το πακέτο μπαίνει στην ουρά αν η ουρά δεν είναι γεμάτη με την συνάρτηση `packetqueue_enqueue_packetbuf()` με διάρκεια ζωής στη μνήμη όση είναι η τιμή τις μεταβλητής `lifetime`. Τα περιεχόμενα του πακέτου πρέπει να είναι προσβάσιμα και αυτό γίνεται με την συνάρτηση `packetbuf_dataptr()` η οποία επιστρέφει ένα δείκτη στο πρώτο byte δεδομένων του πακέτου.

Ο κάθε κόμβος παραλήπτης τότε ελέγχει το επίπεδο του πακέτου δηλαδή το επίπεδο του αποσταλεί με το δικό του. Αν είναι μεγαλύτερο από το δικό του τότε απορρίπτει το πακέτο αλλά σε περίπτωση που είναι μικρότερο αλλάζει το επίπεδο του και το κάνει ίσο με το επίπεδο του μηνύματος συν ένα. Αν είναι ίσο τότε δεν αλλάζει το επίπεδο του αλλά δεν απορρίπτει το μήνυμα. Αν το μήνυμα δεν απορριφθεί τότε αλλάζει και η τιμή του flag μια σημαία που καθορίζει κατά πόσο το μήνυμα έγινε αποδεκτό η όχι. Στην συνέχεια γίνεται ένας επιπλέον έλεγχος ο οποίος είναι για την περίπτωση του κόμβου αποδέκτη ο οποίος πρέπει να ξέρει πόσους κόμβους θα δεχτεί σαν γείτονες και είναι εδώ που γίνεται χρήση του δυσδιάστατου πίνακα και ενός μετρητή οποίος κρατά τον αριθμό των γειτόνων του κόμβου αποδέκτη του `sink_counter_flag`. Η λογική εδώ είναι πως κάθε φορά που ο κόμβος αυτός παραλάβει ένα πακέτο ελέγχει τον μετρητή αυτό με την σταθερά που καθορίζει τον αριθμό των γειτονικών κόμβο του Sink. Αν ο μετρητής είναι μικρότερος τότε στην θέση του μετρητή στον πίνακα κάνει τον πίνακα από 0 σε 1, αυξάνει τον μετρητή και στο πεδίο του unicast μηνύματος με το οποίο θα απαντήσει κάνει το πεδίο που δηλώνει στον παραλήπτη αν είναι συνδεδεμένος με το Sink 1 η 0 αναλόγως. Έπειτα συμπληρώνει και τα υπόλοιπα πεδία του unicast μηνύματος και τον τύπο του και το αποστέλλει με την `unicast_send()` πίσω στον κόμβο που έστειλε το broadcast. Μετά την αποστολή του πακέτου τότε βγάζουμε από την ουρά το broadcast πακέτο. Αυτή διαδικασία επαναλαμβάνεται σε κάθε κόμβο που παραλαμβάνει broadcast πακέτο.

Αφού αποστέλλονται unicast πακέτα τότε όποιος κόμβος παραλαμβάνει θα καλεί την συνάρτηση ή αλλιώς το callback `unicast_recv()`. Στην συνάρτηση αυτή μόλις

παραληφθεί πακέτο μπαίνει στην ουρά και ανάλογα με τον τύπο του λαμβάνονται οι ανάλογες ενέργειες. Στην περίπτωση αυτή ο τύπος του μηνύματος είναι 1 άρα είναι η περίπτωση στην οποία βάζει τον αποστολέα στην λίστα με τους γείτονες του. Πως γίνεται η εισαγωγή αυτή είναι απλό. Γίνεται έλεγχος αν είδη είναι μέσα στους γείτονες διατρέχοντας την λίστα με ένα For loop και ελέγχοντας την διεύθυνση του κόμβου αποστολέα με αυτές τον είδη υπάρχοντων, αν υπάρχει τότε δεν θα γίνει εισαγωγή. Αν δεν υπάρχει δεσμεύεται χώρος με την συνάρτηση `memb_alloc` συμπληρώνονται τα πεδία του νέου γείτονα στην νέα εγγραφή και τότε μπαίνει στην λίστα ο νέος γείτονας με τη συνάρτηση `list_add()`. Μετά βγαίνει από την ουρά το μήνυμα και επιστρέφει η συνάρτηση. Σε κάθε unicast μήνυμα με τύπο 1 που παραλαμβάνεται ακολουθείται αυτή η διαδικασία.

Τώρα ξανά πίσω στην διεργασία μας όπου αν η πηγή έχει ξεκινήσει και οι γείτονες έχουν πάρει τα μηνύματα τότε η σημαία `flag` έχει υψωθεί από τη συνάρτηση και οι κόμβοι με υψωμένη σημαία οι οποίοι όμως δεν είναι πηγές ή παραλήπτες μπορούν να πράξουν και αυτοί όπως η πηγή στη αρχή και να στείλουν με τον ίδιο τρόπο broadcast μηνύματα προς τους γείτονες τους οι οποίοι με την σειρά τους θα κάνουν την ίδια διαδικασία με αυτούς όταν τα παραλάβουν. Κάθε κόμβος κάνει μία φορά μόνο την αποστολή broadcast και αυτό ελέγχεται με την σημαία `broadcast_flag`.

Κάθε φορά που ένας κόμβος περάσει τα σημεία που κάνουν έλεγχο για broadcast θα ελέγχεται αν είναι Sink και τότε με μια επανάληψη ελέγχει τον πίνακα με τους γείτονες του αν έχει συμπληρωθεί κάτι που αν έχει γίνει σημαίνει και το τέλος της πρώτης φάσης του αλγόριθμου αφού όλοι θα έχουν λάβει broadcast μηνύματα θα έχουν απαντήσει με unicast και θα έχει κάθε κόμβος ενημερώσει το επίπεδο αλλά και τον πίνακα με τους γείτονες του. Σε αυτή την περίπτωση για να ενημερωθεί όλο το δίκτυο για τον τερματισμό της φάσης αυτής η πηγή κάνει τη δική της σημαία για την φάση αυτή 0, έτσι σταματά την επανάληψη. Έτσι μόλις η σημαία του sink για την πρώτη φάση γίνει 0 ορίζεται ένα χρονόμετρο το οποίο έχει σκοπό να καθυστερήσει την αποστολή του μηνύματος για τερματισμό της φάσης σε όλο το δίκτυο. Αυτό γίνεται για να προλάβουν όλοι κόμβοι να ενημερώσουν τον πίνακα με τους γείτονες τους σε περίπτωση που δεν πρόλαβαν να το κάνουν. Το χρονόμετρο αυτό καθορίζεται ανάλογα με τον αριθμό των κόμβων στο δίκτυο και μόλις λήξει συνεχίζει ο κόμβος Sink και στέλλει με την συνάρτηση `trickle_send()` μήνυμα "STOP PHASE 1" ούτως ώστε να κάνουν και αυτοί την `neighb_disc=0` όταν παραλάβουν το μήνυμα. Έτσι έχω

υλοποίηση την πρώτη φάση του αλγορίθμου και όλοι οι κόμβοι ανάλογα με την σειρά που θα καλεστούν θα αρχίσουν την επόμενη φάση.

6.3 Φάση δεύτερη: Υλοποίηση ιεραρχικού δένδρου

Σε αυτή τη φάση με οι κόμβοι εκτελούν της εντολές με παρόμοιο τρόπο δηλαδή μέσα σε μία επανάληψη η οποία εξαρτάτε από μία σημαία το `htc_flag` και βάση κάποιο event timer που αρχικοποιείται με την ίδια φόρμουλα που χρησιμοποιείται και στην προηγούμενη φάση.

Αρχικά γίνεται χρήση πάλι μίας σημαίας του `second=1` με το οποίο ελέγχεται ότι κάθε κόμβος θα μπει στην διαδικασία να στείλει μήνυμα σε όλους τους γείτονες του για να μάθει έτσι τα νέα στοιχεία τους πριν να κάνει επιλογής σε ποιο κόμβο θα αποστέλλει τα δεδομένα. Αυτή η διαδικασία της αποστολής γίνεται με ένα `for loop` διατρέχει την λίστα με τους γείτονες του και παίρνοντας από κάθε γείτονα την διεύθυνση, τους αποστέλλει όλους unicast μήνυμα τύπου 2 με σκοπό να του απαντήσουν για να ενημερώσει έτσι τα πεδία τους στην λίστα και να μπορέσει να κάνει την επιλογή του καλύτερου κόμβου για να προωθεί τα δεδομένα του. Αφού στείλει σε όλους που είναι στην λίστα κάνει την σημαία `second=0` για μην μπορεί να επανέλθει αυτή η διαδικασία. Τώρα για το πώς αντιμετωπίζει τα μηνύματα τύπου 2 ο παραλήπτης γίνεται όπως και πριν στο callback του unicast. Πάλι μπαίνει στην ουρά το πακέτο ελέγχεται ο τύπος του και όταν είναι τύπου 2 τότε ο παραλήπτης συμπληρώνει τα πεδία που πρέπει στο unicast μήνυμα που θα απαντήσει και το στέλλει πίσω στον αποστολέα αυτή την φορά σαν τύπου 3 μήνυμα και το βγάζει από την ουρά των μηνυμάτων του. Όταν ο αποστολέας λάβει το μήνυμα τύπου 3 που ουσιαστικά είναι η απάντηση στο τύπου 2 μήνυμα που του έστειλε προηγούμενος πρέπει με βάση τα στοιχεία που έχει το μήνυμα να ενημερώσει την λίστα με τους γείτονες του και συγκεκριμένα την εγγραφή του αποστολέα. Έτσι με τον ίδιο τρόπο με τις άλλες φορές βρίσκει τον αποστολέα μέσα στην λίστα που έχει και ενημερώνει τα πεδία του με βάση το μήνυμα που πήρε. Μετά από αυτό για να μπορεί να ξέρει ποτέ έχουν ενημερωθεί όλες οι εγγραφές των γειτόνων του αυξάνει ένα μετρητή κατά ένα που είναι μια global μεταβλητή την `deb`.

Έτσι πίσω στην διεργασία θα μπορέσει να μπει στην διαδικασία επιλογής του καταλληλότερου γείτονα όταν ο μετρητής εξισωθεί με το μέγεθος τις λίστας των γειτόνων το οποίο περνούμε από την συνάρτηση `list_lenght()`, αλλά πρέπει να έχει πρώτα ολοκληρώσει την μαζική αποστολή για να ενημερωθεί και ο κόμβος να μην

είναι κόμβος Sink. Όταν πλήρη αυτές τις προϋποθέσεις τότε μπορεί να γίνει η επιλογή του γείτονα παραλήπτη με την συνάρτηση `htc()` που παίρνει σαν παράμετρο μόνο την λίστα γειτόνων του κόμβου. Η συνάρτηση θα εξηγηθεί ξεχωριστά πως λειτουργεί στο τέλος. Μετά την επιλογή του κόμβου που θα προωθεί τα πακέτα αυξάνεται ξανά ο μετρητής και έτσι με ένα έλεγχο για το αν ο μετρητής ισούται με 1+ το μέγεθος τις λίστας των γειτόνων του ή είναι Sink αλλάζει την σημαία της φάσης αυτής σε γίνεται `htc_flag=0` και τερματίζει η φάση αυτή.

Τώρα θα εξηγήσω την λειτουργία της `htc()` συνάρτησης κάθε κόμβου.

Αφού αποστέλλεται η λίστα με τους γείτονες κάθε κόμβου και η συνάρτηση πρέπει να κάνει τη επιλογή διατρέχει πάλι την λίστα. Καθώς την διατρέχει η επιλογή γίνεται μέσα από διαφόρους ελέγχους τον κόμβο γειτόνων που είναι στη λίστα. Εάν δεν έχει γείτονες κάποιος από τους κόμβους τότε δεν μπαίνει καν στους ελέγχους για να τον επιλέξει και προχωρά στον επόμενο καθώς επίσης και το ίδιο κάνει εάν ο κόμβος είναι σε κατάσταση συμφόρησης να επιλέξει έτσι επιστρέφει η συνάρτηση. Εάν δεν ισχύει κάτι από τα ποιο πάνω για τον γειτονικό κόμβο συνεχίζει ως εξής. Αρχικά αν ο γειτονικός κόμβος είναι Sink τότε τα στοιχεία θα αποσταλούν σε αυτόν αν δεν είναι Sink αλλά είναι κόμβος που είναι γείτονας με Sink τότε κατά την διάρκεια αν δεν βρεθεί άλλος στη λίστα που να είναι Sink τότε θα είναι αυτός επιλογή κάθε φορά που γίνεται αντιγράφεται στην `global` μεταβλητή `sendto` η διεύθυνση του κόμβου με την συνάρτηση `rimeaddr_copy()`. Σε περίπτωση που κάτι από τα ποιο πάνω δεν ισχύει τότε με τις τοπικές μεταβλητές `minbuffer`, `minbufferlevel` και `minbufaddr` και βρίσκουμε την ποιο ελεύθερη μνήμη μέσα από τους κόμβους με την πρώτη μεταβλητή με την δεύτερη το επίπεδο του κόμβου που την έχει και με την τρίτη την διεύθυνση του κόμβου αυτού.

Τώρα με τις τοπικές μεταβλητές `maxlevel` `maxlevelbuff` `maxlevaddr` κρατούμε αντίστοιχα τον κόμβο με το ψηλότερο επίπεδο, την μνήμη που έχει καθώς και την διεύθυνση του. Τότε με κάποιους ελέγχους δίνετε προτεραιότητα στους κόμβους με την ποιο ελεύθερη μνήμη και το ψηλότερο επίπεδο και ανάλογα με την περίπτωση επιλέγεται ο ανάλογος κόμβος από τους δύο. Υπάρχουν όμως και ειδικές περιπτώσεις όπου ένας κόμβος έχει μόνο φίλο έτσι αναγκαστικά θα προωθούσε σε αυτόν τα πακέτα και σε περίπτωση που αυτός δεν είχε κάποιον άλλο είχαμε μαύρη τρύπα στο δίκτυο. Αυτό αντιμετωπίστηκε με ελέγχους και την ύψωση πάλι μιας σημαίας της `sendtoless`

σε περίπτωση που λάβει κάποιο μήνυμα ο κόμβος να αντιληφθεί ότι δεν μπορεί να στείλει παρακάτω και να ενημερώσει αυτόν που του έστειλε να βρει κάποιον άλλον για να στέλλει. Ακόμα κάτι αφού στην ουσία ένας κόμβος με ένα γείτονα ο οποίος γείτονας δεν έχει κάποιο γείτονα είναι και αυτός άχρηστος για αυτό ακολουθήθηκε αυτή η υλοποίηση για να αντιμετωπίζεται ένα βήμα πιο γρήγορα το πρόβλημα .

Αυτή είναι η ανάλυση της υλοποίησης και της δεύτερης φάσης του αλγορίθμου.

6.4 Φάση τρίτη: Αποστολή δεδομένων και εύρεση εναλλακτικού μονοπατιού

Μετά και την ολοκλήρωση της δεύτερης φάσης ο κάθε κόμβος στο δίκτυο έχει τον δικό του πίνακα με τους γείτονές του αλλά έχει επιλέξει επίσης και ποιος θα είναι ο κόμβος στον οποίο θα προωθεί τα πακέτα σε περίπτωση που η πηγή αρχίσει την αποστολή πακέτων. Η αποστολή πακέτων όμως δεν πρέπει να αρχίσει αυτόματα γιατί μπορεί η πηγή να τρέξει πριν όλοι οι κόμβοι αποφασίζουν σε ποίο κόμβο θα προωθούν τα πακέτα και έτσι να έχουμε αποστολή πακέτων και το ιεραρχικό δένδρο να μην είναι ολοκληρωμένο με αποτέλεσμα να έχουμε χάσιμο πακέτων ή να μην φτάνουν ποτέ στην πηγή. Έτσι όπως και μετά το τέλος της πρώτης φάσης σε περίπτωση που η πηγή πάει για αποστολή πακέτων να την καθυστερήσει για κάποιο χρονικό διάστημα. Η λογική είναι όπως και στο προηγούμενο χρονόμετρο ανάλογα με τους κόμβους που υπάρχουν στο δίκτυο .Μετά στο κομμάτι αυτό υλοποιείται η διαδικασία της αποστολής πακέτων δεδομένων καθώς επίσης και οι ελέγχει για την αλλαγή κόμβου στον οποίο τα στέλλει σε περίπτωση που αυτός που στέλλει παρουσιάσει συμφόρηση ή πέσει κάτω από κάποιο όριο το επίπεδο της ενέργειας του ή δεν έχει κάποιον κόμβο για να προωθήσει τα πακέτα του. Όπως και στις άλλες δυο φάσεις η διαδικασία είναι μέσα σε μια επανάληψη μαζί με ένα event timer το οποίο καθορίζει ποτέ τρέχει κάθε κόμβος.

Με έλεγχο στο πεδίο που καθορίζει αν κόμβος είναι πηγή όσοι κόμβοι είναι πηγές αρχίζουν την αποστολή unicast πακέτων στην διεύθυνση sendto τα οποία είναι τα πακέτα δεδομένων και δηλώνονται σαν τύπου 4.Κάθε κόμβος που παραλαμβάνει πακέτα δεδομένων στην συνάρτηση unicast_recv() τα βάζει στη ουρά ,ελέγχοντας βλέπει πως το πακέτο είναι τύπου 4 άρα είναι πακέτο δεδομένων τότε με τη συνάρτηση packetqueue_len() παίρνει το μέγεθος της ουράς που έχει σαν μνήμη για τα πακέτα. Τότε για να ελέγξει αν στον κόμβος πρόκειται να εμφανιστεί συμφόρηση γίνεται ένας έλεγχος του μεγέθους της ουράς +3 το οποίο δεν πρέπει να είναι μεγαλύτερο ή ίσο με την σταθερά η οποία καθορίζει το μέγεθος της ουράς. Αν είναι μεγαλύτερο τότε ο

κόμβος αυτός αποστέλλει πίσω ένα πακέτο το οποίο έχει σαν σκοπό να τον ενημερώσει ούτως ώστε να αλλάξει τον κόμβο στον οποίο στέλλει πακέτα. Το πακέτο αυτό είναι τύπου 5. Ο επόμενος έλεγχος είναι κατά πόσο ο κόμβος ακόμα έχει την απαιτούμενη ενέργεια για να είναι στο μονοπάτι των πακέτων προς το Sink αν δεν έχει αυτή την φορά στέλλει άλλο πακέτο για να ενημερώσει τον αποστολέα ότι δεν έχει αρκετή ενέργεια. Το πακέτο αυτό είναι τύπου 6. Ο τελευταίος έλεγχος είναι εάν ο κόμβος έχει κάπου να προώθηση τα πακέτα του ή εάν ο κόμβος που στέλλει είναι ο κόμβος που του τα έστειλε τότε στέλλει και αυτός ένα μήνυμα τύπου 7 για να ενημερώσει τον αποστολέα του να βρει άλλον κόμβο να στέλλει τα πακέτα. Τώρα σε περίπτωση που ο κόμβος δεν αντιμετωπίζει κάποιο πρόβλημα από αυτά απλά αντιγράφει το μήνυμα που έλαβε με την συνάρτηση `packetbuf_copyto()` το βάζει μετά στην δική του μνήμη μηνυμάτων για αποστολή και το προωθεί στον δικό του `sendto` κόμβο. Τώρα όσον αφορά τα μηνύματα που αποστέλλονται σε περίπτωση που υπάρχει κάποιο πρόβλημα η κυριότερη τους δουλειά είναι να αλλάξουν τις τιμές κάποιων μεταβλητών οι οποίες δηλώνουν ότι ο κόμβος πρέπει να βρει άλλον `sendto` γείτονα. Στην περίπτωση μηνύματος τύπου 5 που είναι το μήνυμα που αφορά συμφόρηση τότε όταν το παραλάβει ο κόμβος αλλάζει την τιμή της μεταβλητής `conget_flag=1` και πάει στην λίστα με τους γείτονες του και αλλάζει την κατάσταση μνήμης αυτού του γείτονα με την νέα κατάσταση. Τώρα όταν το μήνυμα είναι τύπου 6 που είναι για την έλλειψη ενέργειας του κόμβου πάλι αλλάζει η τιμή της μεταβλητής αλλά αυτή την φορά της `batteryflag=1` και δεν ανανεώνει την κατάσταση του γείτονα του με την νέα κατάσταση αλλά τον αφαιρεί από την λίστα με τους γείτονες του και ελέγχει αν έχει κάπου να στέλλει δηλαδή αν έχει άλλους γείτονες για να ενημερώσει την σημαία του ότι είναι ακατάλληλος σε περίπτωση που δεν έχει. . Την ίδια ακριβώς λειτουργία με τα τύπου 6 μηνύματα κάνουν και με μήνυμα τύπου 7 οι κόμβοι που τα παραλαμβάνουν μόνο που η σημαία αυτή την φορά είναι το `sendtoless_flag=1`.

Έτσι τώρα πίσω στην διεργασία κάθε φορά που μια από τις τρεις μεταβλητές αλλάζει τιμή και μπορέσει ο κόμβος να περάσει από τους ελέγχους θα καλείται η συνάρτηση `htc()` η οποία με βάση την λίστα των γειτόνων κάθε κόμβου καλείται να επιλέξει των καταλληλότερο, δηλαδή διαλέγει εναλλακτικό μονοπάτι για αποστολή των πακέτων. Έτσι με την τρίτη φάση τελειώνει και η υλοποίηση του αλγορίθμου.

Κεφάλαιο 7

Πειραματική αξιολόγηση Αλγορίθμου

7.1 Γενικά

7.2 Περιβάλλον Προσομοίωσης

7.3 Με βάση την πυκνότητα των κόμβων

7.4 Με σταθερό αριθμό κόμβων αλλά με περισσότερα Sources και Sinks

7.5 Με διαφορετικές τοπολογίες

7.1 Γενικά

Σε αυτό το σημείο θα γίνει η πειραματική αξιολόγηση του αλγορίθμου. Η αξιολόγηση του αλγορίθμου θα γίνει μέσα από ένα εργαλείο του Contiki το COOJA το οποίο είναι προσομοιωτής υλοποιημένος σε γλωσσά προγραμματισμού Java για αλγορίθμους υλοποιημένους σε Contiki.

Έτσι θα έγινε μια σειρά από προσημειώσεις σε αυτό το εργαλείο με σκοπό να εξεταστούν κάποιες μετρικές που αφορούν τα δίκτυα αισθητήρων και πως επηρεάζονται από τον αλγόριθμο αυτό. Οι μετρικές οι οποίες θα εξεταστούν είναι η απώλεια πακέτων που παρατηρείται στο δίκτυο packet loss μαζί με την καθυστέρηση που υπάρχει στο δίκτυο για να φτάσουν τα πακέτα από την πηγή στο κόμβο αποδέκτη. Η ροή στο δίκτυο δεν θα μελετηθεί αφού ο αλγόριθμος δεν κάνει κάτι για να μειώσει την ροή των πακέτων έτσι παραμένει πάντα σταθερή σύμφωνα με τον ρυθμό αποστολής των πηγών. Η κατανάλωση ενέργειας του δικτύου παρότι είναι μια πού σημαντική μετρική και έχει γίνει στην υλοποίηση η δομή της αλλά και ο τρόπος που λαμβάνονται τα μέτρα για να αντιμετωπίζεται δεν θα αξιολογηθεί αυτή τη στιγμή. Ο λόγος είναι ότι δεν μπόρεσα βρω τον τρόπο με τον οποίο θα υπολογιζόταν η ενέργεια του κάθε κόμβου του δικτύου. Για αυτό το λόγο θα στην αξιολόγηση δεν μπορώ να συμπεριλάβω αποτελέσματα και για την ενέργεια αφού οι κόμβοι θεωρείται ότι πάντα έχουν το 100% της ενέργειας του .

Οι μετρικές που θα εξεταστούν οι οποίες αναφέρονται πιο πάνω θα παρατηρηθούν πως επηρεάζονται με την αλλαγή μερικών τεχνικών παραμέτρων του δικτύου. Αυτές είναι η πυκνότητα των κόμβων στο δίκτυο Density είναι η τοπολογία του δικτύου, και

το πώς επηρεάζονται όταν αλλάξει ο αριθμός των πηγών ή των Sink's που υπάρχουν στο δίκτυο. Αλλά πριν προχωρήσουμε με την αξιολόγηση θα μιλήσουμε στο επόμενο υποκεφάλαιο για το περιβάλλον της προσομοίωσης και κάποια ζητήματα τα οποία προκύπτουν και πρέπει να ληφθούν υπόψη.

7.2 Περιβάλλον Προσομοίωσης

Αρχικά σε όλες τις προσομοιώσεις και τα σενάρια τα οποία επιλέχθηκαν οι κόμβοι είναι σκορπισμένη μέσα σε ένα τετραγωνισμένο πλέγμα μεγέθους 100m x100m και οι παράμετροι στον Πίνακα1 ποιο κάτω είναι πάντα σταθερές.

Πινάκας 1.Παράμετροι Προσομοίωσης

Απόσταση X (m)	100
Απόσταση Y (m)	100
Τύπος κόμβων	Sky mote
Πληρότητα μνήμης(%)	85
Απόσταση Μετάδοσης(m)	30
MAC Driver	CSMA
RDC Driver	ContikiMAC

Ο λόγος για τον οποίο ξεκινούν οι κόμβοι με 85% πληρότητα μνήμης είναι γιατί λόγο του μικρού αριθμού κόμβων αλλά και χρόνου που γίνονταν οι προσομοιώσεις δεν εμφανιζόταν συμφόρηση έτσι για σκοπούς αξιολόγησης δημιουργούμε συνθήκες τις οποίες είναι δυνατό να εμφανιστεί. Επιπρόσθετα με αυτό όμως υπάρχουν μερικές απαιτήσεις αλλά και προβλήματα που πρέπει να αντιμετωπιστούν για να λειτουργήσει ο αλγόριθμος και να μπορέσει να γίνει μελέτη για τα αποτελέσματα του.

Αρχικά κάτι που καθιστά δύσκολή την αξιολόγηση του αλγορίθμου μέσα από αυτό το εργαλείο είναι ο τρόπος λειτουργίας του εργαλείου. Το COOJA εκτελεί τον αλγόριθμο σειριακά για κάθε κόμβο ενώ στην πραγματικότητα κάθε κόμβος εκτελεί παράλληλα με τους υπόλοιπους τον αλγόριθμο με βάση τον οποίο λειτουργεί. Έτσι δημιουργούνται προβλήματα χρονισμού σε ορισμένα σενάρια και δεν μπορούν να ληφθούν αντικειμενικές μετρήσεις.

Επίσης για να μπορεί να λειτουργήσει με επιτυχία πρέπει με βάση την ποσότητα των κόμβων να ρυθμίζονται ανάλογα και τα χρονόμετρα τα οποία καθορίζουν και την ροή εκτέλεσης του κώδικα αλλά και την σειρά που οι κόμβοι εκτελούν τον κώδικα. Αυτό πρέπει να γίνεται γιατί όπως παρατήρησα όσο αυξάνονταν οι κόμβοι στο δίκτυο μερικοί έμεναν εκτός δικτύου με αποτέλεσμα η δρομολόγηση των πακέτων να γίνεται λανθασμένα. Συγκεκριμένα στον αλγόριθμο υπάρχουν τέσσερα χρονόμετρα που απαιτείτε ο συντονισμός τους αλλά τα δύο είναι τα πιο σημαντικά στην ορθή λειτουργία του αλγορίθμου.

Πρώτο είναι αυτό στην πρώτη φάση που καθορίζει πόσο θα περιμένει το Sink για να στήσει το μήνυμα στο δίκτυο να σταματήσει αυτή η φάση. Αυτό γίνεται γιατί όσο μεγαλώνει ο αριθμός των κόμβων χρειάζονται περισσότερο χρόνο να γίνει η ανακάλυψη του δικτύου έτσι είναι υποχρεωμένος να περιμένει για να μην μείνουν κόμβοι εκτός δικτύου. Το δεύτερο είναι στην φάση που ξεκινά η πηγή την αποστολή πακέτων και είναι για παρόμοιο λόγο που υπάρχει δηλαδή να μην μείνουν κόμβοι χωρίς να έχουν βρει που θα προωθούν τα πακέτα και έτσι να μην γίνει ορθά το στήσιμο του ιεραρχικού δένδρου. Η διαφορές που έγιναν στα χρονόμετρα για κάθε φορά που άλλαζε ο αριθμός των κόμβων φαίνεται στον Πίνακα 2 πιο κάτω.

Πίνακας 2. Ρυθμίσεις χρονομέτρων ανάλογα με τον αριθμό των κόμβων

ΦΑΣΕΙΣ	ΧΡΟΝΟΜΕΤΡΟ	20 Κόμβοι	40 Κόμβοι	60 Κόμβοι
ΦΑΣΗ 1 ^η	Καθορισμός πόσο ξυπνά κάθε κόμβος για ανακάλυψη του δικτύου	20s	40s	40s
	Πόσο χρόνο περιμένει το Sink για να σταματήσει αυτή την φάση	40s	60s	60s
ΦΑΣΗ 2 ^η	Καθορισμός πόσο ξυπνά κάθε κόμβος για να βρει που θα στέλλει τα πακέτα του	20s	22s	22s
	Πόσο χρόνο περιμένει ο source για να αρχίσει να στέλλει πακέτα	60s	60s	80s
ΦΑΣΗ 3 ^η	Συχνότητα αποστολής πακέτων δεδομένων από την πηγή	1s	1s	1s

Τέλος κάτι ακόμα που πρέπει να ληφθεί υπόψη είναι πως για σκοπούς πιο εύκολης αξιολόγησης αλλά και της προσομοίωσης οι κόμβοι πηγές αλλά και κόμβοι αποδέκτες

καθορίζονται προγραμματιστικά πριν καν αρχίσει η προσομοίωση έτσι μπορώ να καθορίσω την θέση τους ανάλογα με το σενάριο που ακολουθείται.

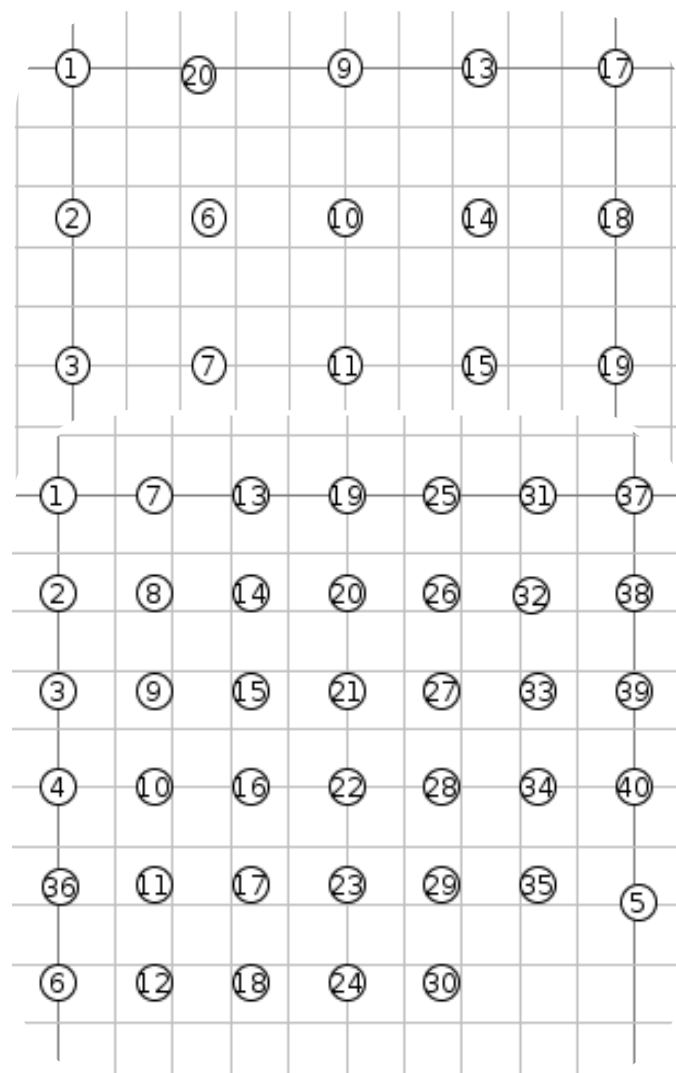
Αυτά είναι τα ζητήματα τα οποία πρέπει ληφθούν υπόψη έτσι τώρα θα συνεχίσουμε με την αξιολόγηση. Αρχικά θα ξεκινήσω με τον έλεγχο στην αλλαγή της πυκνότητας του δικτύου.

7.3 Με βάση την πυκνότητα των κόμβων

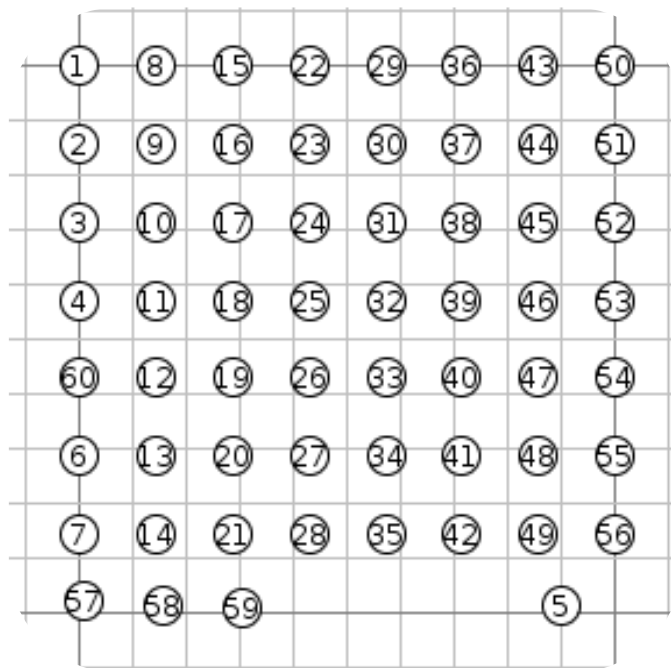
Εδώ θα γίνει η αξιολόγηση του δικτύου με την αλλαγή της πυκνότητας του. Θα δοκιμαστεί με τρεις διαφορετικές περιπτώσεις κάθε φορά για 20 κόμβους για 40 και 60. Δεν θα χρησιμοποιηθούν μεγαλύτερης κλίμακας δίκτυα λόγω των προβλημάτων που ανέφερα προηγουμένως τα οποία εμποδίζουν την ορθή λειτουργία των προσομοιώσεων.

Οι κόμβοι κάθε φορά είναι κατανεμημένοι σύμφωνα με το τετραγωνισμένο πλέγμα και στο δίκτυο υπάρχει ένας κόμβος πηγή και ένας κόμβος αποδέκτης. Ο κόμβος πηγή είναι ο κόμβος με τον αριθμό 1 και ο κόμβος αποδέκτης είναι ο κόμβος 5 σε κάθε φορά.

Τα τρία δίκτυα στα οποία έγιναν οι προσημειώσεις είναι τα πιο κάτω:

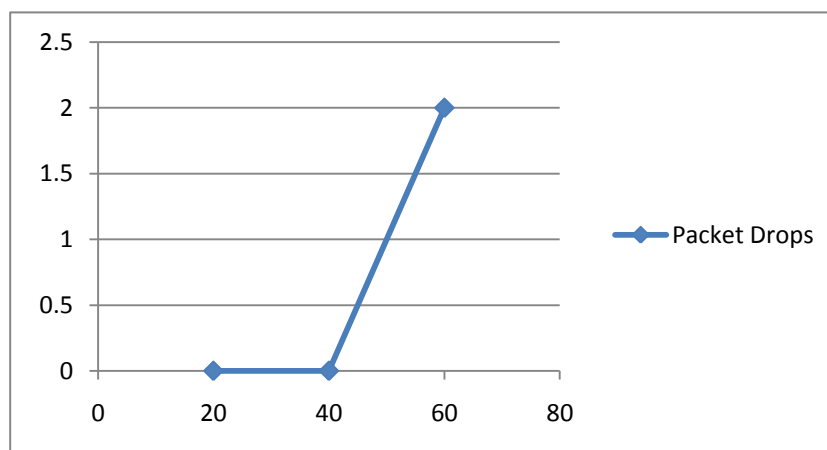


Σχημα 6.2 Δίκτυο 40 κόμβων



Σχήμα 6.3 Δίκτυο 60 κόμβων

Θα δούμε την απώλεια πακέτων στο δίκτυο πως μεταβάλλεται με την πυκνότητα του δικτύου αλλά για τα πακέτα δεδομένων και όχι για τα πακέτα έλεγχου του δικτύου. Οι μετρήσεις που λήφθηκαν είναι σε δείγμα αποστολής 100 πακέτων δεδομένων και τα δεδομένα που λήφθηκαν είναι τα ποιο κάτω.

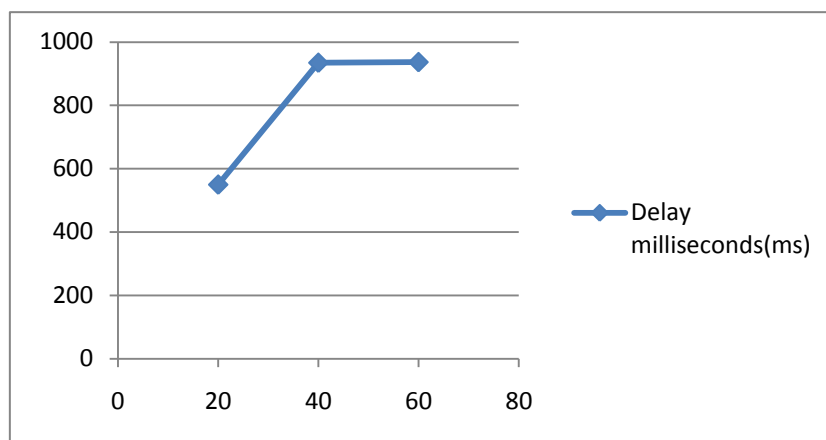


Σχήμα 6.4 Packet Drops σε σχέση με τον αριθμό των κόμβων

Όπως φαίνεται και στο Σχήμα 6.4 υπάρχει αύξηση των χαμένων πακέτων όσο μεγαλώνει ο αριθμός των κόμβων. Στην συγκεκριμένη περίπτωση στις δύο πρώτες μετρήσεις δεν είχαμε καθόλου απώλειες αφού δεν δημιουργήθηκε κανένα περιστατικό συμφόρησης αλλά στην τρίτη περίπτωση όπου δημιουργήθηκαν δύο περιστατικά είχαμε δύο απώλειες πακέτων. Η απώλειες είναι σχετικά μικρές αφού ήταν μία σε κάθε περιστατικό κάτι που δείχνει πώς με την χρήση του αλγόριθμου για έλεγχο

συμφόρησης αλλά και λόγω της έγκαιρης ενημέρωσης για χρήση εναλλακτικής διαδρομής για τα πακέτα έχουμε καλύτερη διαχείριση των κρίσεων και μειωμένες απώλειες πακέτων.

Η επόμενη μετρική με βάση την οποία θα αξιολογηθεί ο αλγόριθμος είναι η καθυστέρηση των πακέτων να φτάσουν από τον κόμβο πηγή στο Sink. Αυτό επηρεάζεται ανάλογα και με το πόσους κόμβους χρησιμοποιεί το μονοπάτι για να σταλεί το πακέτο στο Sink. Ποία κάτω είναι τα δεδομένα τα οποία συλλέχτηκαν από της προσημιώσεις.



Σχήμα 6.5 End-to-End delay σε σχέση με τον αριθμό των κόμβων

Βασικά σε αυτό το σημείο βλέπουμε πώς επηρεάζεται ο μέσος χρόνος μεταφοράς των πακέτων από την πηγή στο Sink ανάλογα με την αύξηση των κόμβων στο δίκτυο. Στην συγκεκριμένη περίπτωση λόγω της αύξησης των κόμβων άλλαξε και το μέγεθος του μονοπατιού από την πηγή στο Sink. Στους 20 κόμβους ήταν 7 hops ενώ για 40 και 60 κόμβους ήταν 10 hops για αυτό και παρατηρείται αυτή η διάφορα στον χρόνο. Αυτό μπορεί να θεωρηθεί σαν μειονέκτημα του αλγορίθμου η αύξηση του χρόνου αλλά σίγουρα είναι θετικό ότι και στις δύο τελευταίες περίπτωση έχουμε το ίδιο μέγεθος μονοπατιού αλλά στους 60 κόμβους όπως είπα πριν εμφανίστηκαν και δύο περιστατικά συμφόρησης και παρόλα αυτά η διάφορα τους είναι σχεδόν αμελητέα κάτι που δηλώνει την ταχύτατη λειτουργία του αλγορίθμου στην διαδικασία της εύρεσης εναλλακτικού μονοπατιού που είναι εξίσου σύντομο με το αρχικό. Όπως είπα όμως στην αρχή είναι ο μέσος χρόνος που εμφανίζεται άρα μπορεί σε αυτό να οφείλεται και το γεγονός του ίδιου χρόνου. Γενικά όμως η αύξηση του χρόνου ανάλογα με την πυκνότητα του

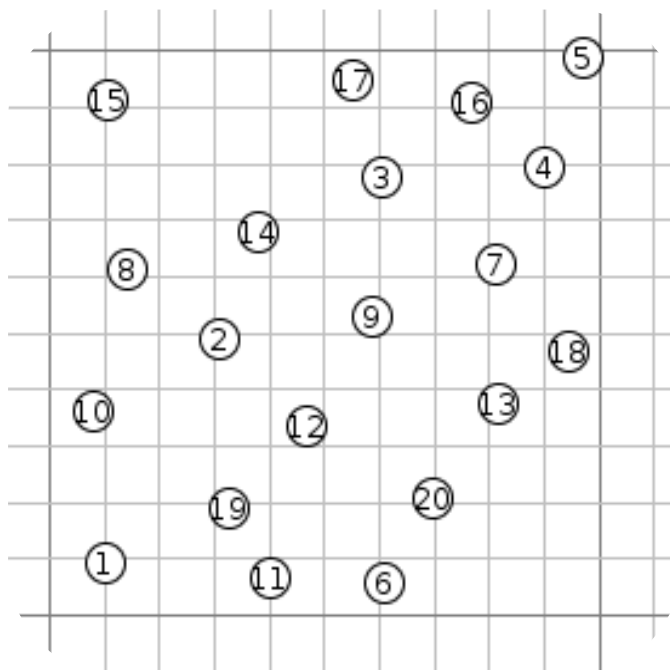
δικτύου μπορεί να θεωρηθεί μειονέκτημα του αλγορίθμου αφού σε πολλές περιπτώσεις και εφαρμογές ο χρόνος ανταπόκρισης είναι το κρισιμότερο σημείο.

Εδώ αφού δεν θα μελετηθεί η μετρική της ενέργειας τελειώνει εδώ η αξιολόγηση με βάση την πυκνότητα στο δίκτυο και θα συνεχιστεί με την μελέτη των μετρικών με βάση των αλλαγών στις πηγές και στα Sinks

7.4 Με σταθερό αριθμό κόμβων αλλά με περισσότερα Source's

Εδώ θα γίνει αξιολόγηση όχι τόσο των μετρικών αλλά της αποδοτικότητας του αλγορίθμου από πλευράς υλοποίησης. Αυτό θα γίνει μέσα από τρία διαφορετικά σενάρια όπου η πυκνότητα του δικτύου παραμένει σταθερή αλλά μεταβάλλεται ο αριθμός και οι θέσεις των πηγών μέσα στο δίκτυο στο οποίο οι κόμβοι είναι τοποθετημένη τυχαία και ανάλογα με το σενάριο αλλάζει ο αριθμός των πηγών ή η θέση τους. Η πυκνότητα του δικτύου και στα τρία σενάρια παραμένει σταθερή και είναι 20 κόμβοι. Στα σενάρια αυτά δεν θα δοθεί έμφαση στις μετρικές αφού τα δεδομένα που εξάγονται σύμφωνα με την υλοποίηση που έγινε δεν είναι αντικειμενικά αν συγκριθούν με την απλή περίπτωση του πρώτου σεναρίου

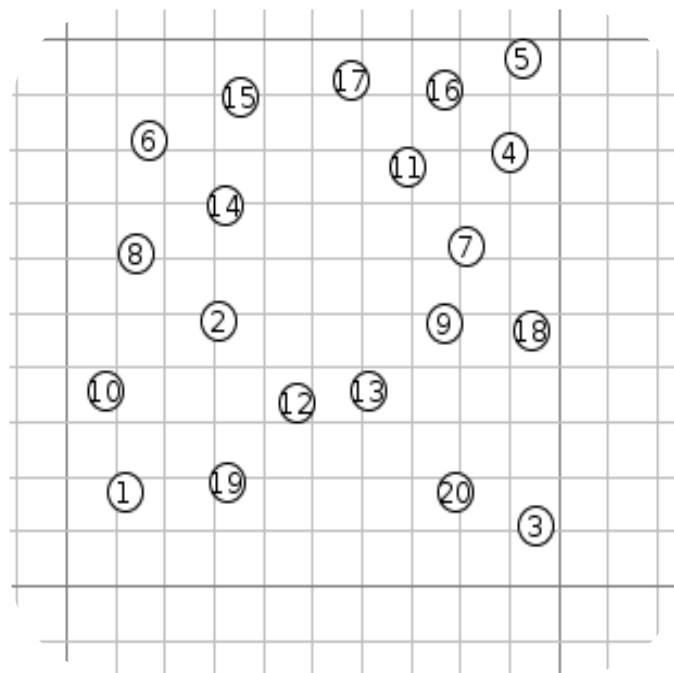
Ας αρχίσουμε με το πρώτο σενάριο όπου είναι μια απλή τυχαία διάταξη των κόμβων με ένα κόμβο πηγή και τον κόμβο 1 και ένα Sink το 5. Η διάταξη φαίνεται στο Σχήμα 6.6 που ακολουθεί.



Σχήμα 6.6 Πρώτο σενάριο

Αυτό το σενάριο μπήκε βασικά για να είναι το μέτρο σύγκρισης για τα αλλά δύο αφού δεν παρουσιάζει συμφόρηση. Είναι ένα υγιές δίκτυο το οποίο στήνεται με την βοήθεια του αλγορίθμου έχει ένα μονοπάτι από τον κόμβο 1 στον κόμβο 5 αποτελούμενο από 8 κόμβους και η καθυστέρηση του ενός πακέτου να φτάσει από την πηγή στο Sink είναι 640ms. Τα επόμενα δυο σενάρια είναι τροποποίησης αυτού του δικτύου και πως αυτό το δίκτυο μπορεί να αντιμετωπίζει κάποιες κατάστασης ή αντίθετα δημιουργεί με την βοήθεια του HTAP αλγορίθμου.

Στο δεύτερο σενάριο προστίθεται ακόμα μία πηγή στο δίκτυο ο κόμβος 3 και αλλάζει λίγο η τοπολογία και γίνεται όπως το Σχήμα 6.7



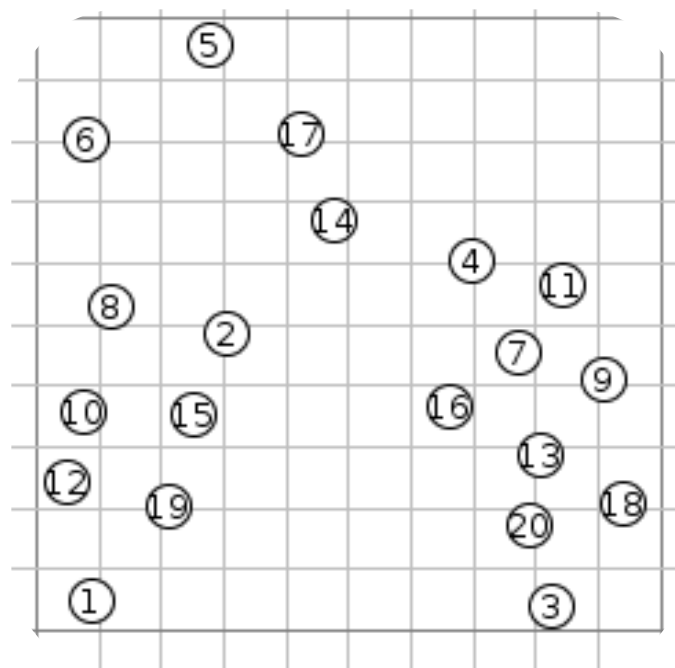
Σχημα 6.7 Δεύτερο σενάριο

Εδώ οι δύο πηγές είναι στα άκρα του δικτύου μακριά από το Sink αλλά με πολλούς κόμβους στο ενδιάμεσο για να μπορούν να προωθήσουν τα πακέτα.

Εδώ σε αυτή την προσημείωση είναι που άρχισαν να εμφανίζονται και τα προβλήματα που έχουμε λόγο του COOJA αφού στην πραγματικότητα οι δύο πηγές αν ήταν σε κανονικούς κόμβους τα πακέτα θα αποστέλλονταν και από της δύο ταυτόχρονα ή έστω ανεξάρτητα η μία από την άλλη εδώ λόγο των χρονομέτρων που έχουμε η σειρά αποστολής των πακέτων μπερδεύεται και καθιστά δύσκολο να βγουν αντικειμενικά συμπεράσματα. Για παράδειγμα το end-to-end delay για την μία πηγή είναι κανονικό ενώ για την άλλη η οποία ξεκίνα δεύτερη την αποστολή αφού δεν μπορούν να το

κάνουν ταυτόχρονα λόγο τον χρονομέτρων και τις ανεξαρτησίας του κώδικα για κάθε κόμβο είναι πολύ μεγάλο γιατί ουσιαστικά είναι σαν να παρεμβαίνει στην διαδικασία της πηγής που ξεκίνησε πρώτη. Η παρατήρηση που μπορώ να κάνω εδώ σχετικά με τον αλγόριθμο είναι ότι λόγω των δυο πηγών ο φόρτος του δικτύου ανεβαίνει για αυτό και παρατηρούνται και δύο συμβάντα συμφόρησης αλλά έχουμε και άλλου είδους προβλήματα που μέσα από τον αλγόριθμο πρέπει να αντιμετωπιστούν. Το πρόβλημα δημιουργείται λόγω του ότι και οι δυο πηγές θέλουν με την σειρά τους να κάνουν αναγνώριση του δικτύου και καθορισμό των κόμβων σε επίπεδα στο δίκτυο. Έτσι αυτή που ξεκινά πρώτη μπορεί να ανάθεση τα επίπεδα με βάση τον εαυτό της αλλά μετά όταν η άλλη πηγή ξεκινήσει μερικά από τα επίπεδα να αλλάξουν και έτσι ούτε η μια πηγή αλλά ούτε και η άλλη έχει ορθά υλοποιημένο το ιεραρχικό της δένδρου προς το Sink. Το αποτέλεσμα αυτού είναι λάθος δρομολογήσεις πακέτων, τα πακέτα να κάνουν κύκλους είτε δημιουργία κόμβων που είναι μαύρες τρύπες. Αυτά τα προβλήματα προσπάθησα να τα αντιμετωπίσω προγραμματιστικά αλλά δεν είναι η καλύτερη λύση γιατί αυξάνει την πολυπλοκότητα του αλγορίθμου επομένως και την κατανάλωση ενέργειας.

Σε αυτό το σενάριο αυτά ήταν τα συμπεράσματα μου για τον αλγόριθμο και ακολουθεί και το τρίτο σενάριο. Στο τρίτο σενάριο δεν μεταβάλλεται ο αριθμός των πηγών αλλά μόνο η τοπολογία του δικτύου. Η τοπολογία φαίνεται στο Σχήμα 6.8



Σχήμα 6.8 Τρίτο σενάριο

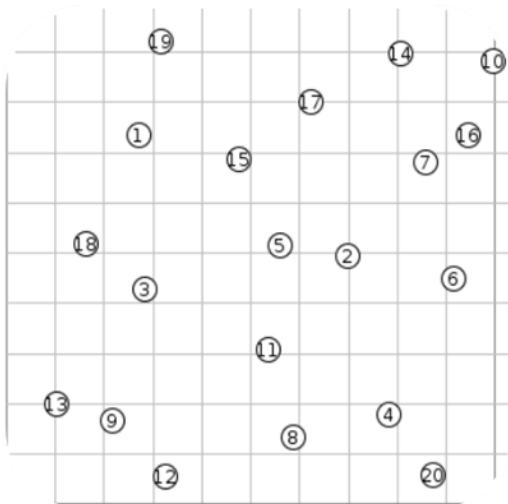
Στο σενάριο αυτό σκοπό έχει να μελετήσει την αντίδραση του αλγορίθμου όταν οι πρόσβαση στο Sink είναι περιορισμένη. Τα δεδομένα και σε αυτή την περίπτωση για τις μετρικές δεν μπορούν να χρησιμοποιηθούν για αξιολόγηση του αλγόριθμου αλλά και πάλι μπορεί να γίνει μια αξιολόγηση αποτελεσματικότητας του λόγο τον προβλημάτων που προκύπτουν. Στο προηγούμενο σενάριο αναφέρθηκα στο φαινόμενο όπου τα ιεραρχικά δέντρα κάθε πηγής επηρεάζονται από την ύπαρξη της άλλης έτσι σε αυτό το σενάριο για να αποφευχθεί αυτό διαμορφώθηκε έτσι η τοπολογία.

Εδώ το πρόβλημα που παρουσιάστηκε είναι πως η μια πηγή μονοπωλεί τον κόμβο αποδέκτη λόγο τον περιορισμένων κόμβων που έχουν πρόσβαση σε αυτών και η άλλη καταστρέφει την πλευρά του δικτύου της από μόνη της. Δηλαδή αν η πηγή 1 αρχίζει πρώτη την αποστολή πακέτων τότε θα πιάσει τους κόμβους 14 και 17 που είναι κόμβοι με πρόσβαση στο Sink. Έτσι όταν ξεκινήσει η πηγή 3 την αποστολή και φτάσουν τα πακέτα σε εκείνους τους κόμβους θα δημιουργηθεί συμφόρηση με αποτέλεσμα να πρέπει να βρει εναλλακτικό μονοπάτι που δεν υπάρχει άλλο προς τον Sink. Η πηγή από τον κόμβο 14 έστω ότι απέχει 4 hops όταν ο 14 ενημερώσει τον αποστολέα του για την συμφόρηση αυτός θα ψάξει να βρει άλλον από τους γείτονες του, μέχρι να το κάνει όμως η πηγή εξακολουθεί να στέλλει πακέτα αφού δεν μπορεί να ενημερωθεί για την συμφόρηση. Αυτή η ενέργεια και παράληψη που δεν μπορεί να υλοποιηθεί στα πλαίσια του αλγορίθμου μπορεί να έχει διάφορες καταστροφικές για το δίκτυο καταλήξεις. Μια είναι η εκδοχή όπου συμφώνα με τον αλγόριθμο θα επιλέξει τον γείτονα του με το πιο άδειο buffer μα ούτε και αυτός δεν θα μπορεί να το προωθήσει αφού δεν έχει πρόσβαση κάπου άρα ενημερώνει ξανά πίσω τον αποστολέα του και αυτός πάλι ψάχνει για γείτονα και αν δεν έχει πάει και αυτός πίσω και συνεχίζεται μέχρις ότου όλοι οι κόμβοι γεμίσουν και παραλύει το μισό δίκτυο με την πηγή να μην έχει κόμβο να προωθήσει τα δεδομένα. Αυτό είναι και το αποτέλεσμα της προσομοίωσης του σεναρίου αυτού.

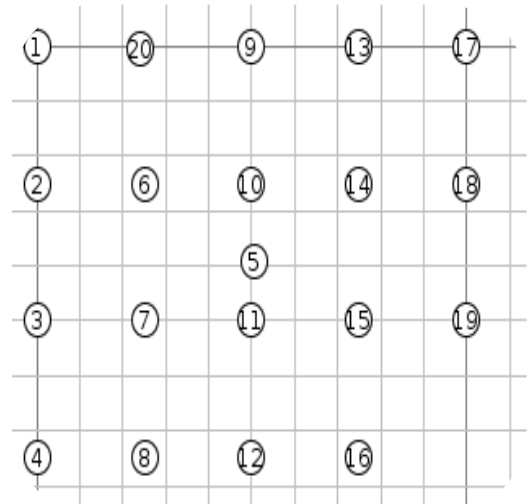
Έτσι το συμπέρασμα από το σενάριο αυτό είναι πάλι πως η χρήση του COOJA το οποίο τρέχει γραμμικά τους κόμβους και οι πηγές δεν τρέχουν παράλληλα οδηγεί στην αποκοπή της μίας πηγής από το να αποστέλλει στο Sink και ως αποτέλεσμα ο αλγόριθμος που δεν φροντίζει για να ενημερώνεται η πηγή να πλημυρίζει το μισό δίκτυο

7.5 Με διαφορετικές τοπολογίες

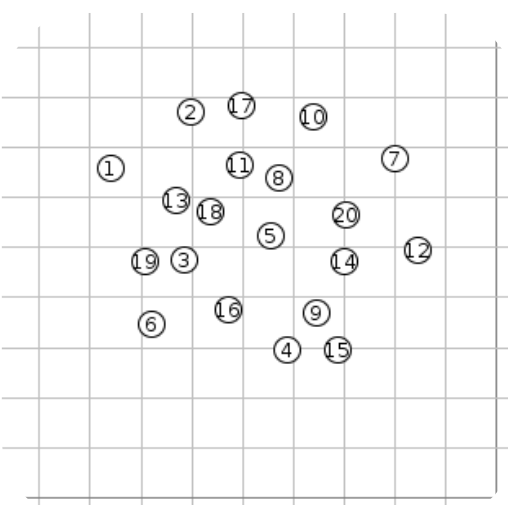
Μέσα από αυτές τις προσομοιώσεις ο αλγόριθμος θα δοκιμαστή σε διαφορετικές τοπολογίες με σταθερή την πυκνότητα των 20 κόμβων. Η τοποθέτηση τους που έγινε με τέσσερις διαφορετικούς τρόπους την Diffusion με συγκέντρωση στο κέντρο όπως όταν οι κομβοί αφαιρούν από ένα αεροπλάνο, Random Placement, Grid placement και manual τοποθέτηση όπως θα έμπαιναν οι κόμβοι στο ενδιαμέσο του χώρου αν στις δύο απέναντι διαγώνιες γωνίες του πλέγματος υπήρχαν κύκλοι που δεν μπορείς να βάλεις μέσα κόμβους. Ο αλγόριθμος εδώ θα αξιολογηθεί με βάση τις βασικές μετρικές που αναφέρθηκαν πιο πριν. Οι τοπολογίες που χρησιμοποιήθηκαν φαίνονται στα πιο κάτω σχήματα.



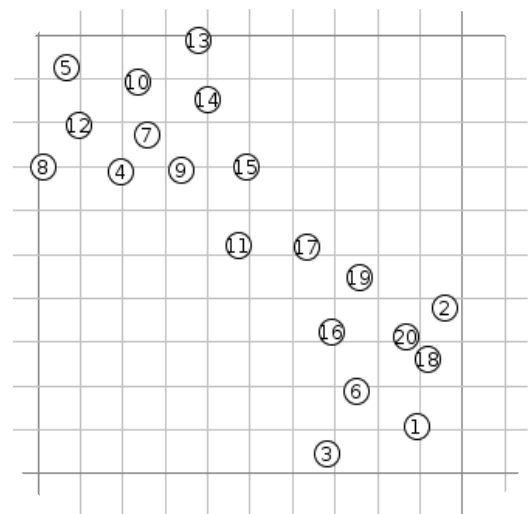
Σχήμα 6.9 Random placement



Σχήμα 6.10 Grid placement

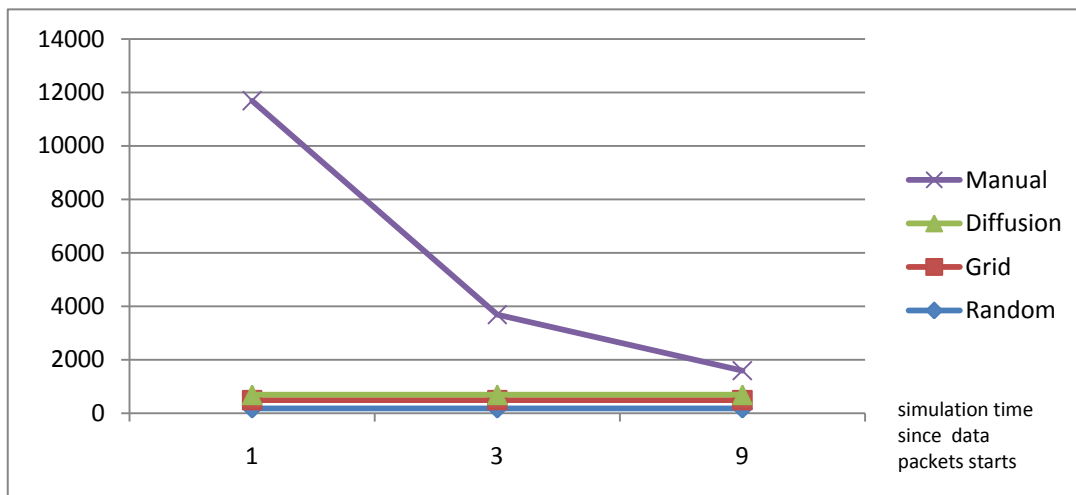


Σχήμα 6.11 Diffusion placement



Σχήμα 6.12 Custom placement

Τα αποτελέσματα των προσομοιώσεων φαίνονται πιο κάτω στο Σχήμα 6.13



Σχήμα 6.13 End-to-End delay σε σχέση με τον χρόνο αποστολής πακέτων δεδομένων

Στον έλεγχο του Average Packet Drop παρατηρήθηκε ότι μόνο η manual διάταξη των κόμβων είχε απώλειες και αυτό οφείλεται στο ότι υπάρχει μεγάλη συγκέντρωση των κόμβων κοντά στην πηγή για αυτό και αρχικά έχει τόσο μεγάλη καθυστέρηση στην αποστολή των πακέτων και μετά σταθεροποιείται. Ο λόγος που δημιουργείται συμφόρηση είναι η μεγάλη πυκνότητα των κόμβων αφού για την αρχικοποίηση του δικτύου αλλάζουν πολλά πακέτα ελέγχου έτσι με το που ξεκινά η μετάδοση πολύ σύντομα παρουσιάζεται συμφόρηση ενώ προς το Sink δεν παρουσιάζεται γιατί μέχρι τα πακέτα να φτάσουν εκεί οι μνήμες των κόμβων αυτών έχουν αδειάσει. Οι υπόλοιπες τοπολογίες δεν αντιμετώπισαν πρόβλημα συμφόρησης για αυτό είναι σταθερός ο χρόνος τους.

Τα συμπεράσματα είναι ότι η μέση ροή δεδομένων ανεξάρτητα από την τοπολογία παρέμενε σταθερή αφού ο αλγόριθμος από την φύση του δεν κάνει έλεγχο τις ροής αφού ο σκοπός του είναι να αποφεύγεται η συμφόρηση με την χρήση των έξτρα κόμβων.

7.6 Προβλήματα του αλγορίθμου HTAP

Μέσα από τις προσομοιώσεις αλλά και τα σενάρια τα οποία ακολουθήθηκαν για την αξιολόγηση της υλοποίηση του αλγορίθμου HTAP διαφάνηκαν κάποια προβλήματα που αντιμετωπίζει ο αλγόριθμος αλλά βγήκαν και κάποια συμπεράσματα τα οποία αφορούν την λειτουργικότητα του αλγορίθμου.

Αρχικά θα αναφερθούν τα προβλήματα του αλγορίθμου και στη συνέχεια τα συμπεράσματα. Συνολικά εντοπίστηκαν δύο προβλήματα όσον αφορά την σχεδίαση του αλγορίθμου. Πρώτον είναι το γεγονός ότι η κάθε πηγή καθορίζει τα επίπεδα του δικτύου και δεν αποτρέπει στις περιπτώσεις που υπάρχουν και άλλες πηγές στο δίκτυο ο καθορισμός των επιπέδων να γίνεται και από τις δύο ξεχωριστά. Αυτό έχει ως αποτέλεσμα να αλλοιώνονται τα επίπεδα που θέτει κάθε πηγή από τα επίπεδα άλλων πηγών και να έχει σαν αποτέλεσμα τις λάθος δρομολογήσεις πακέτων αλλά και αδυναμία δημιουργίας σωστών μονοπατιών προς τους κόμβους Sink.

Δεύτερο είναι πώς ο κάθε κόμβος στο σημείο όπου καλείται να βρει σε ποιόν κόμβο θα αποστέλλει τα πακέτα του επιλέγει πάντα τον κόμβο που έχει την μικρότερη πιθανότητα να εμφανίσει συμφόρηση ανεξάρτητα από το επίπεδο του κόμβου αυτού. Η ενέργεια αυτή αποφέρει δύο προβλήματα όσον αφορά τον αλγόριθμο αυτό για τα οποία δεν λαμβάνει κανένα μέτρο για να αντιμετωπιστούν. Το ότι δεν γίνεται χρήση των επιπέδων κάθε κόμβου κατά την επιλογή μπορεί να οδηγήσει στην δημιουργία μεγάλων μονοπατιών μεταξύ κόμβων ίδιου επιπέδου αφού τα πακέτα δεν προωθούνται προς το Sink έτσι να έχει μεγαλύτερη καθυστέρηση στο να φτάσουν τα πακέτα στο κόμβο αποδέκτη γίνεται χρήση περισσότερων κόμβων του δικτύου άρα περισσότερη κατανάλωση ενέργειας και μειώνει την διάρκεια ζωής του δικτύου. Τα άλλα προβλήματα λόγω του ότι τα επίπεδα κάθε κόμβου δεν χρησιμοποιούνται είναι ότι δύο κόμβοι του ίδιου επιπέδου να μπορούν να επιλέξουν ο ένας των άλλων ως παραλήπτες των πακέτων τους και να χρειάζεται επιπλέον ενέργεια για να αντιμετωπιστεί αυτό με συνέπειες σπατάλη ενέργειας και καθυστερήσεις στο δίκτυο.

Αυτά είναι τα δύο κυριότερα προβλήματα τα οποία εντοπίστηκαν μέσα από τα διάφορα σενάρια και τις προσημειώσεις τις οποίες έγιναν. Τώρα τα συμπεράσματα που βγήκαν όσον αφορά τον αλγόριθμο HTAP είναι πως λόγω της αυξημένης πολυπλοκότητας που έχει μέχρι να αρχικοποιηθεί το δίκτυο και να αρχίσει η ροή πακέτων δεδομένων εμφανίζεται μεγάλη καθυστέρηση κάτι που μπορεί να είναι καταστροφικό αφού πρέπει να είναι άμεση η ανταπόκριση του δικτύου στην εμφάνιση ενός συμβάντος.

Επίσης ακόμα ένα συμπέρασμα είναι πώς με την πληροφόρηση κάθε κόμβου απλά για αλλαγή κόμβου στον οποίο προωθεί τα πακέτα και χωρίς καθόλου έλεγχο ροής θα υπάρχουν περιπτώσεις στις οποίες το δίκτυο θα καταλήγει με την πηγή να μπορεί να προωθήσει τα πακέτα αφού δεν υπάρχει διαθέσιμος κόμβος είτε λόγω συμφόρησης είτε

λόγο μειωμένης ενέργειας. Αυτό γίνεται γιατί χωρίς την μείωση του φόρτου του δικτύου, το δίκτυο θα υπερφορτώνεται και θα εμφανίζει συμφόρηση ανεξάρτητα από το μονοπάτι που ακολουθεί.

Αυτά είναι τα προβλήματα τα οποία εμφανίστηκαν στον αλγόριθμο αλλά και κάποια συμπεράσματα τα οποία αφορούν τον τρόπο λειτουργίας του αλγορίθμου. Ακολουθούν τα γενικά συμπεράσματα τα οποία βγήκαν μέσα από αυτήν την διπλωματική εργασία

Κεφάλαιο 8

Συμπεράσματα

8.1 Γενική σύνοψη

8.2 Συμπεράσματα

8.1 Γενική σύνοψη

Αυτή η διπλωματική εργασία ασχολήθηκε με θέματα που αφορούν τα ασύρματα δίκτυα αισθητήρων και το πρόβλημα της συμφόρησης σε αυτά αλλά και τρόπους με τους οποίους μπορεί να αντιμετωπιστεί. Ποίο συγκεκριμένα είδαμε στην αρχή μία γενική αναφορά στα δίκτυα αισθητήρων για το τι είναι αυτά τα δίκτυα από τι αποτελούνται πως επικοινωνούν αλλά και που μπορούν να βρουν εφαρμογές στην καθημερινή μας ζωή.

Μετά σχετικά με το ποια είναι τα δομικά στοιχεία στα δίκτυα αισθητήρων είδαμε τους κόμβους αισθητήρες. Για τους αισθητήρες είδαμε την βασική αρχιτεκτονική και πως δουλεύει κάθε κομμάτι, είδαμε σε τι κατηγορίες μπορούν να καταχτούν αλλά και κάποια ζητήματα τα οποία αντιμετωπίζουν οι κόμβοι αισθητήρων λόγω της φύσης τους. Είδαμε τα προβλήματα που υπάρχουν στον χώρο αλλά και ζητήματα πάνω στα οποία γίνονται έρευνες για να αντιμετωπιστούν και επικεντρωθήκαμε στο πρόβλημα της συμφόρησης στα ασύρματα δίκτυα αισθητήρων.

Μιλήσαμε για την συμφόρηση στα δίκτυα γενικά αλλά και για την συμφόρηση στα ασύρματα δίκτυα αισθητήρων. Όσον αφορά τα WSN's έγινε μια ποιο εκτενής αναφορά στα προβλήματα συμφόρησης και στις επιπτώσεις που μπορεί να έχουν στα δίκτυα, και υπό ποιες μορφές αλλά και κάτω από ποιες συνθήκες μπορεί να δημιουργηθεί συμφόρηση.

Στην συνέχεια ακολούθησε το κεφάλαιο που ασχολείται με την αντιμετώπιση της συμφόρησης όπου παρουσιάστηκαν μερικά πρωτόκολλα και στην συνέχεια αναλύθηκαν το κάθε πρωτόκολλο ξεχωριστά. Σε αυτά τα πρωτόκολλα συμπεριλήφθηκε και ο αλγόριθμος ο οποίος υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας και εξηγήθηκε ο τρόπος λειτουργίας του. Αφού έγινε η

επεξήγηση του αλγορίθμου ακολούθησε η περιγραφή του περιβάλλοντος στο οποίο υλοποιήθηκε ο αλγόριθμος. Έτσι είδαμε το λειτουργικό Contiki τα χαρακτηριστικά του καθώς και τη προσφέρει που μπορεί να χρησιμοποιηθεί στην υλοποίηση. Μετά ακολούθησε η περιγραφή τις υλοποίησης του αλγορίθμου η οποία χωριστικέ σε τρεις φάσεις με βάση και τις φάση του αλγορίθμου. Σε αυτό το σημείο επεξηγήθηκαν δομές που χρησιμοποιήθηκαν καθώς επίσης και ο τρόπος σκέψης για την υλοποίηση σε κάποια σημεία.

Μετά και την περιγραφή της υλοποίησης ακολουθεί η φάση της αξιολόγησης του αλγορίθμου μέσα από προσομοιώσεις η οποία και πάλι έσπασε σε τρεις διαφορετικές κατευθύνσεις. Ασχολήθηκε στην μελέτη του δικτύου με την συμπεριφορά του αλγορίθμου στις αλλαγές τις πυκνότητας του, με αλλαγές στον αριθμό των πηγών του δικτύου καθώς επίσης και το πώς αλλάζει η απόδοση του δικτύου με την χρήση του αλγορίθμου σε διαφορετικές τοπολογίες.

Τέλος έγινε σχολιασμός των αποτελεσμάτων αλλά και της αποδοτικότητας του αλγορίθμου.

Ακολουθεί το τελευταίο κεφάλαιο αυτής της διπλωματικής εργασίας που είναι τα γενικά συμπεράσματα.

8.2 Συμπεράσματα

Τα συμπεράσματα αυτά βγήκαν μέσα από τα διάφορα στάδια που ακολουθήθηκαν για την περάτωση αυτής της διπλωματικής εργασίας. Μέσα από τις διάφορες φάσεις οι οποίες ήταν η μελέτη γενικά γύρο από τα WSN's, η πιο ειδικευμένη μελέτη γύρο από την συμφόρηση αλλά και η προσπάθεια για υλοποίηση του αλγορίθμου και η διαδικασία αξιολόγησης του οδήγησαν η κάθε μια στην εξαγωγή κάποιων συμπερασμάτων.

Το συμπέρασμα το οποίο βγαίνει μέσα από την μελέτη για τα WSN's είναι ότι ως τεχνολογία τα WSN's είναι πολλά υποσχόμενη και με μεγάλα ωφελήματα για τον άνθρωπο. Αυτό φαίνεται μέσα από την ραγδαία ανάπτυξη που έχουν την εφαρμογή τους σε πολλούς τομείς της ζωής μας αν και ακόμα βρίσκεται σε πρώιμο στάδιο η ανάπτυξη τους αλλά και η πολύ έντονη ερευνητική δραστηριότητα γύρω από αυτά. Όπως αναφέρεται και ποίο πάνω στο κεφάλαιο που αφορά τον έλεγχο συμφόρησης για τα WSN's η συμφόρηση είναι ένα κρίσιμο σημείο όσων αφορά τα δίκτυα και δεν

μπορεί να βρεθεί ο μοναδικός και βέλτιστος τρόπος για επίλυση της. Έτσι αυτό την κατατάσσει ως μία ερευνητική περιοχή η οποία πάντα θα απασχολεί τους ερευνητές σε όλα τα είδη δικτύων αφού δεν είναι μόνο τα WSN's που αντιμετωπίζουν αυτό το πρόβλημα. Το συμπέρασμα είναι πως πάντα θα υπάρχουν περιθώρια βελτίωσης των μηχανισμών που υλοποιούνται για τον έλεγχο συμφόρησης αφού κάθε προσέγγιση διαφέρει έστω και σε μικρό βαθμό από τις άλλες κάτι που αλλάζει την απόδοση είτε συνολικά είτε σε συγκεκριμένα σημεία.

Τώρα ποιο συγκεκριμένα όσον αφορά τον αλγόριθμο και την υλοποίηση που έγινε σε αυτή τη διπλωματική εργασία έχουν βγει κάποια συμπεράσματα. Τα συμπεράσματα αυτά είναι δυνατό να αλλάξουν ή να μην ισχύουν αν η υλοποίηση γίνει διαφορετικά με ποιο αποδοτικό τρόπο ή χρήση διαφορετικών βιβλιοθηκών και δομών. Τα συμπεράσματα είναι τα ακόλουθα.

Αρχικά ένα συμπέρασμα το οποίο βγαίνει είναι πως ο αλγόριθμος αυτός λόγω του ότι απαιτεί το ιεραρχικό δένδρο να ξεκινά από τον κόμβο πηγή του προσθέτει επιπλέον πολυπλοκότητα. Αυτό εξηγείται γιατί με το να ξεκινά η πηγή την δημιουργία του δένδρου δεν έχει την δυνατότητα να γνωρίζει πότε να σταματήσει ή πότε το δένδρο έχει ολοκληρωθεί έτσι για να γίνουν αυτές οι διεργασίες απαιτούνται περισσότεροι υπολογισμοί σε κάθε κόμβο περισσότερες αποστολές πακέτων έλεγχου κάτι που οδηγεί αλυσιδωτά στις μεγαλύτερες περιόδους χρόνου που χρειάζεται ένας κόμβος να είναι ενεργός στο δίκτυο, επιπλέον καθυστέρηση σπάταλη υπολογιστικής ισχύος και που όλα αυτά έχουν αντίκτυπο και στη κατανάλωση ενέργειας.

Ακόμα ένα συμπέρασμα είναι πως χωρίς καθόλου έλεγχο ροής στο δίκτυο σε κάποιες περιπτώσεις είναι αδύνατο να αποφύγεις την συμφόρηση ή την κατάρρευση του δικτύου. Αυτό το συμπέρασμα βγαίνει μέσα από τις προσομοιώσεις που έγιναν και συγκεκριμένα στα σενάρια που αφορούν την αλλαγή του αριθμού των πηγών για παράδειγμα στο σενάριο 3 όπου συνέβη κάτι αντίστοιχο.

Επίσης στα ασύρματα δίκτυα έχουν αναπτυχθεί μερικοί μηχανισμοί για παράδειγμα τα MAC drivers τα RDC drivers τα οποία κάνουν πάρα πολύ καλή διαχείριση και της ενέργειας αλλά και στις περιπτώσεις των συγκρούσεων πακέτων έτσι η δημιουργία συμφόρησης είναι δύσκολη ή τουλάχιστον δεν εμφανίζεται άμεσα σε ένα δίκτυο για αυτό και σε αυτόν τον αλγόριθμο κατά την προσημείωση για να εμφανιστή συμφόρηση οι κόμβοι ξεκινούσαν με το 85% της μνήμης τους είδη γεμάτο.

Υπάρχουν ακόμα και μερικά άλλα συμπεράσματα που αφορούν κυρίως την υλοποίηση του αλγορίθμου τα οποία αναφέρονται και σε προηγούμενα κεφάλαια καθώς αποτελούν εξήγηση για ορισμένα θέματα που έπρεπε να αντιμετωπιστούν για να μπορέσει ο αλγόριθμος να προσημειωθεί και να βγουν αποτελέσματα.

Τελικά καταλήγουμε πως μέσα από την από την μελέτη και την έρευνα γύρο από τα WSN's θα καταστεί εφικτή η αντιμετώπιση των προβλημάτων τους και θα μουν στη ζωή μας ακόμα περισσότερο αφού μπορούν να βοηθήσουν τον άνθρωπο σε χιλιάδες πρακτικές εφαρμογές από διαφορετικούς τομείς. Βασική προϋπόθεση πάντα αποτελεί η εξέλιξη της τεχνολογίας και η σωστή εκμετάλλευση της ενσωματώνοντας την στα δίκτυα αυτά.

Βιβλιογραφία

- [1] J. Blumenthal, M. Handy, F. Golasowski, M. Haase, and D. Timmermann. “Wireless Sensor Networks - New Challenges in Software Engineering.”, In Proc. 2003 Emerging Technologies and Factory Automation, pages 551--556. IEEE Computer Society Press, Sept. 2003.
- [2] K. Holger, A. Willig, “Protocols and Architectures for Wireless Sensor Networks “, Wiley, Published Online January 2006
- [3] B. Hull, K. Jamieson, and H. Balakrishnan, “Mitigating congestion in wireless sensor networks,” in SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems. New York, NY, USA: ACM, 2004, pp. 134–147.
- [4] The Contiki Operating System [Online], Available: <http://www.sics.se/contiki/>
- [5] C. Sergiou, V. Vassiliou, A. Pitsillides, “Reliable Data Transmission in Event-based Sensor Networks during Overload Situation,” in Proc. of the 2nd International Workshop on Performance Control in Wireless Sensor Networks (PWSN 2007), Austin, Texas, October 23, 2007.
- [6] V. Vassiliou and C. Sergiou, “Performance study of node placement for congestion control in wireless sensor networks,” in 3rd International Conference on New Technologies, Mobility and Security (NTMS), Cairo, Egypt, Dec., 2009, pp. 1 –8.
- [7] C. Y. Wan, S. B. Eisenman, and A. T. Campbell, “CODA: Congestion Detection and Avoidance in Sensor Networks”, In Proceedings of the ACM SenSys, November 2003.
- [8] C. Wang, K. Sohraby, B. Li and W. Tang, “Issues of Transport Control Protocols for Wireless Sensor Networks,” In Proc. of the International Conference on Communication, Circuits and Systems, Vol. 1, May 2005, pp. 422-426.
- [9] W. Ye, J. Heidemann and D. Estrin, ” An Energy-Efficient MAC Protocol for Wireless Sensor Networks.” In INFOCOM 2002. IEEE Press, June 2002.
- [10] Jennifer. Y, Biswanath. M, Dipak. G, “Wireless Sensor Networks: A Survey.” Computer Networks 52 (2008) 2292-2330, April 2008, pp2299-22300, [ONLINE], Available: <http://ece.ut.ac.ir/Classpages/F87/ECE637/FILES/material/Article%20WSN%20survey%202008.pdf>
- [11] Contiki On-line Documentation [Online], Available: <http://www.sics.se/~adam/contiki/docs/>