



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ, ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάλυση Της Κατανάλωσης Ενέργειας Στα Ασύρματα
Δίκτυα Αισθητήρων

Asem A.Abujoudeh

Επιβλέπων : Καθηγητής Γεώργιος Σταμούλης

2^{ος} Βαθμολογητής: Δρ. Παναγιώτης Κίικρας

Βόλος 2010

ΠΕΡΙΛΗΨΗ

Σκοπός αυτής της εργασίας είναι η μελέτη της καταναλισκόμενης ενέργειας σε ασύρματα δίκτυα αισθητήρων, χρησιμοποιώντας δύο βασικούς προσομοιωτές ενέργειας PowerTossim και Avrora, καθώς και η μελέτη των ενεργειακών μηχανισμών και του τρόπου χειρισμού τους, καθώς η προσπάθεια δημιουργίας κατάλληλων ενεργειακών αρχείων για ένα γενικό μοντέλο.

Αρχικά, έγινε μια εισαγωγή των ασύρματων δικτύων αισθητήρων, των χαρακτηριστικών, των εφαρμογών και των προβλημάτων τους

Έπειτα, έγινε μια αναφορά στους δύο προσομοιωτές που χρησιμοποιούνται για τη μέτρηση και εκτίμηση της ενέργειας ενός δικτύου αισθητήρων.

Αντίστοιχα, εγκαταστάθηκε και χρησιμοποιήθηκε το λειτουργικό σύστημα TinyOS μαζί με τον εξομοιωτή του τον TOSSIM, αλλά και στον εξομοιωτή ενέργειας PowerTossim και ύστερα εγκαταστάθηκε ο προσομοιωτής Avrora .

Στη συνέχεια, βρέθηκαν τα σχετικά αρχεία τα οποία καθορίζουν την ενέργεια που χρειάζεται ο κάθε αισθητήρας για κάθε λειτουργία που εκτελεί και έπειτα δημιουργήθηκαν δύο παρόμοια αρχεία για μελλοντικές χρήσεις. Επίσης εκτελέστηκαν κάποιες εφαρμογές σε διαφορετικές πλατφόρμες και έγινε αποθήκευση των αποτελεσμάτων για σαφέστερη παρατήρηση.

Στο τέλος έγινε μία εκτέλεση της έτοιμης εφαρμογής του TinyOS “CntToLedsAndRfm” για διαφορετικό αριθμό κόμβων, τόσο στον PowerTossim χρησιμοποιώντας το περιβάλλον οπτικοποίησής του TinyViz, όσο και στον Avrora στις πλατφόρμες pc, mica2 αντίστοιχα, έπειτα οπτικοποιήθηκαν τα αποτελέσματα σε γραφήματα με την βοήθεια του προγράμματος Matlab.

Λέξεις – Κλειδιά : ασύρματα δίκτυα αισθητήρων, καταναλισκόμενη ενέργεια, TinyOS, TOSSIM, PowerTossim, Matlab, μέτρηση, εκτίμηση, ενεργειακό μοντέλο, CntToLedsAndRfm, TinyViz .

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

1. ΕΙΣΑΓΩΓΗ

- 1.1 Γενικά (σελ. 6)
- 1.2 Ο Σκοπός της πτυχιακής (σελ. 8)
- 1.3 Δομή της εργασίας (σελ. 9)

2. ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ

- 2.1 Χαρακτηριστικά των Ασύρματων Δικτύων Αισθητήρων (σελ. 10)
 - 2.1.1 Περιβάλλοντα λειτουργίας (σελ. 11)
- 2.2 Εφαρμογές των δικτύων αισθητήρων (σελ. 12)
- 2.3 Προβλήματα και προκλήσεις (σελ. 14)
 - 2.3.1 Διαφορές με υπάρχοντα δίκτυα (σελ. 14)
 - 2.3.2 Σημαντικές προκλήσεις (σελ. 16)

3. ΟΙ ΠΡΟΣΟΜΟΙΩΤΕΣ TOSSIM-POWER TOSSIM ΚΑΙ AVRORA

- 3.1 Tossim (σελ. 19)
 - 3.1.1 Ο PowerTossim (σελ. 22)
 - 3.1.2 Η Αρχιτεκτονική του PowerTossim (σελ. 22)
 - 3.1.3 Η μονάδα (module) PowerState (σελ. 24)
 - 3.1.4 Αποτίμηση του PowerTossim (σελ. 25)
- 3.2 Avrora (σελ. 26)
- 3.3 Ενδεικτικά αποτελέσματα συγκρίσεων (σελ. 30)

4. ΑΝΑΛΥΣΗ ΤΗΣ ΚΑΤΑΝΑΛΩΣΗΣ ΕΝΕΡΓΕΙΑΣ

- 4.1 PowerTossim (σελ. 32)
 - 4.1.1 Επεξεργαστής (σελ. 35)
 - 4.1.2 Πομποδέκτης (σελ. 35)
 - 4.1.3 Επιπλέον καταναλώσεις (σελ. 36)
- 4.2 Avrora (σελ. 36)
- 4.3 Ενδεικτικά τρεξίματα και αποθήκευση αποτελεσμάτων σε Trace Files (σελ. 43)
- 4.4 Δημιουργία Εναλλακτικών Αρχείων (σελ. 44)

5. ΕΚΤΕΛΕΣΗ ΤΟΥ CntToLedsAndRfm ΜΕ ΤΟΥΣ POWERTOSSIP & AVRORA

5.1 PowerTossim (σελ. 45)

5.2 Avrora (σελ. 51)

6. ΕΠΙΛΟΓΟΣ (σελ. 54)

7. ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ (σελ. 55)

8. ΠΑΡΑΡΤΗΜΑ (σελ. 56)

ΕΙΣΑΓΩΓΗ

1.1 Γενικά

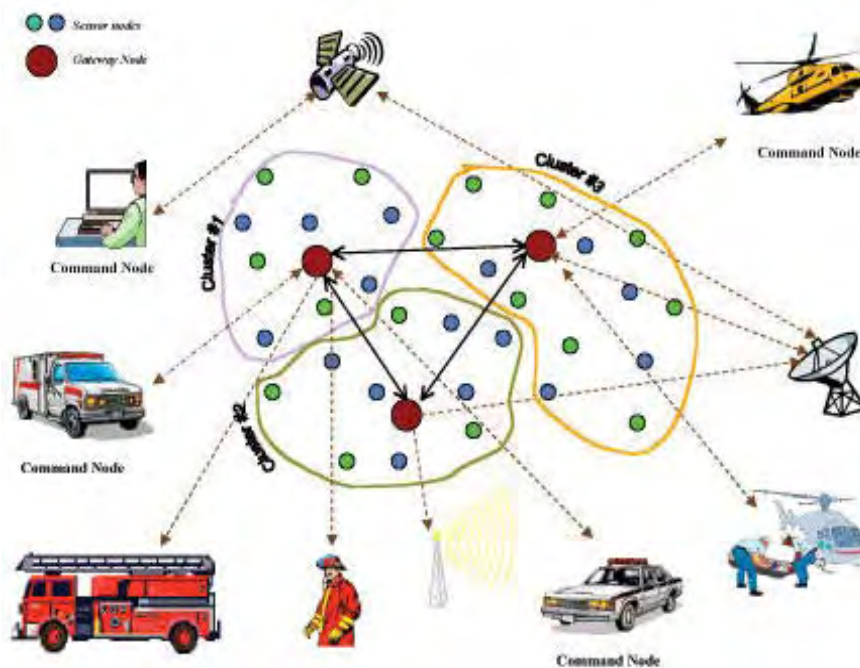
Εδώ και πολλούς αιώνες, βασική μέριμνα της ανθρωπότητας αποτελεί η μεταφορά – μετάδοση πληροφορίας, με τον ταχύτερο και ασφαλέστερο δυνατό τρόπο. Άμεσο επακόλουθο της ανάγκης αυτής ήταν η έλευση των τηλεπικοινωνιών, στις αρχές του περασμένου αιώνα, για να καταλήξουμε στις μέρες της τηλεπικοινωνιακής επανάστασης που διανύουμε, όπου η επικοινωνία έχει μπει στην καθημερινή μας ζωή με ποικίλους διαφορετικούς τρόπους. Ο τομέας δε, στον οποίο παρατηρείται η εντονότερη κινητικότητα σε ερευνητικό επίπεδο, έχει να κάνει με τα δίκτυα υπολογιστικών μηχανών. Με την είσοδο στον 21^ο αιώνα και στα πλαίσια της γενικής παγκοσμιοποίησης σε πολλούς τομείς της ανθρώπινης δραστηριότητας, ριζικές αλλαγές παρατηρούνται κυρίως στις ασύρματες επικοινωνίες, δηλαδή εκείνες στις οποίες η πληροφορία μεταδίδεται με τη μορφή ραδιοκύματος στον αέρα. Έτσι, οι ασύρματες επικοινωνίες βρίσκονται στην κορυφή της πυραμίδας του ερευνητικού ενδιαφέροντος και παρουσιάζουν μια αλματώδη εξέλιξη, χαρακτηριζόμενη από την προσπάθεια για τη συνεχή δημιουργία νέων, φιλικών προς το χρήστη τεχνολογιών, που έχουν τη δυνατότητα να διευκολύνουν τη ζωή μας σε επίπεδο πάμπολλων εφαρμογών. Η πρόοδος αυτή της τεχνολογίας των ασυρμάτων επικοινωνιών επισύρει οφέλη για τους χρήστες και κέρδη για τους δημιουργούς της.

Η ουσιαστική εξέλιξη των προσωπικών ασύρματων επικοινωνιακών συστημάτων ξεκινά γύρω στα 1980 με συστήματα που βασίζονταν σε μεθόδους αναλογικής διαμόρφωσης. Οι ταχύτατα αυξανόμενες ανάγκες όμως σε απαιτήσεις, σε συνδυασμό με την αδιάκοπη ερευνητική προσπάθεια πάνω σε νέες τεχνολογίες, έφεραν στο προσκήνιο νέα τηλεπικοινωνιακά συστήματα, στις αρχές τις δεκαετίας του '90, σηματοδοτώντας την έναρξη της εποχής των ψηφιακών συστημάτων ασύρματων επικοινωνιών.

Η ανάπτυξη και ευρεία χρήση των υπηρεσιών που παρέχονταν από τα συστήματα αυτά άνοιξε διάπλατα νέους ορίζοντες στον τομέα των ασύρματων επικοινωνιών. Ορόσημο αποτέλεσε η ανάπτυξη ασύρματων δικτύων στα οποία οι

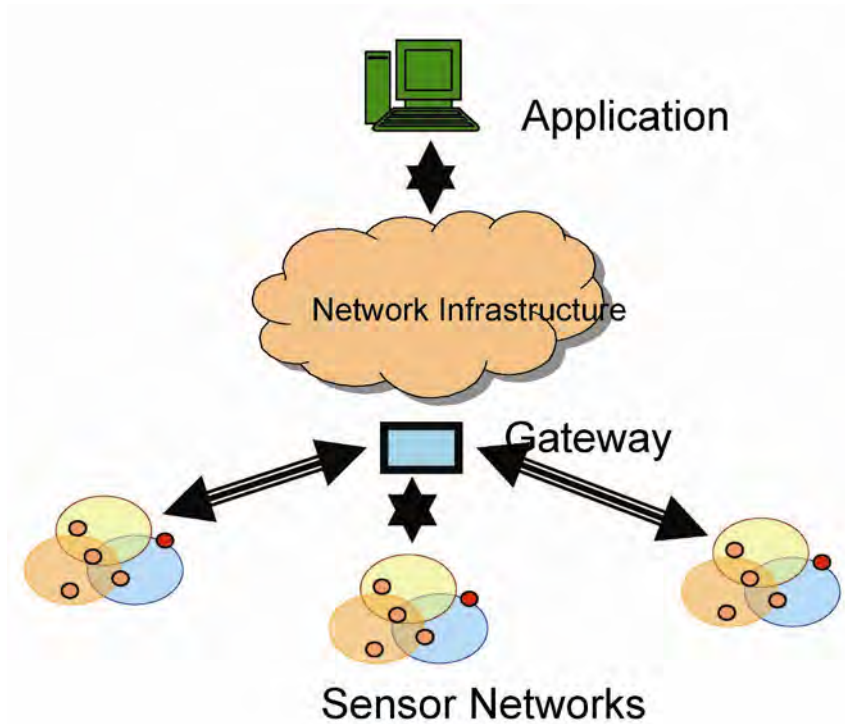
χρήστες ανταλλάσσουν δεδομένα απευθείας μεταξύ τους οργανωμένοι σε τοπικές ασύρματες δομές, οι οποίες είναι δυνατό να μεταβάλλονται δυναμικά. Τα δίκτυα αυτά, γνωστά στη βιβλιογραφία σαν «ad – hoc δίκτυα», έθεσαν ένα πλήθος νέων προκλήσεων στον παγκόσμιο ερευνητικό χώρο και άνοιξαν την πόρτα, σε ακόμα νεότερες τεχνολογίες που αναμένεται να βοηθήσουν σημαντικά την ανθρωπότητα στο άμεσο μέλλον.

Τα τελευταία χρόνια προωθείται πολύ η χρήση μικροσκοπικών συσκευών σε όλες τις πτυχές της τεχνολογικής ανάπτυξης. Δηλαδή, απλές και χαμηλής ισχύος συσκευές που ενσωματώνουν ασύρματες τηλεπικοινωνιακές ζεύξεις υψηλής απόδοσης και βελτιωμένες μικρής κλίμακας πηγές ενέργειας έχουν συνδυαστεί με μειωμένα κόστη παραγωγής για να κάνουν ένα νέο τεχνολογικό όνειρο πραγματικότητα. Τα Ασύρματα Δίκτυα Αισθητήρων (Wireless Sensor Networks) .



Τα Ασύρματα Δίκτυα Αισθητήρων αποτελούνται από ένα ή περισσότερα (base station) και από μερικές δεκάδες ή χιλιάδες κόμβους αισθητήρες, οι οποίοι διασκορπίζονται σε ένα χώρο. Οι κόμβοι αυτοί συλλέγουν πληροφορίες από το

περιβάλλον και ανάλογα με την εφαρμογή, είτε επεξεργάζονται τις πληροφορίες και τις στέλνουν, είτε τις στέλνουν χωρίς καμία επεξεργασία.



1.2 Ο Σκοπός της πτυχιακής

Βάσει των παραπάνω, είναι σαφές ότι τα (WSNs) αποτελούν ένα ιδιαίτερο είδος δικτύων, με ιδιαίτερες απαιτήσεις αλλά και πληθώρα εφαρμογών, που μάλιστα συνεχώς αυξάνονται.

Στόχος λοιπόν της παρούσης εργασίας είναι να εστιάσει σε μοντέλα προσομοίωσης κατανάλωσης ενέργειας για τέτοιου είδους δίκτυα, καθώς κρίνεται ιδιαίτερα σημαντικό να μπορούν να μελετηθούν τα ασύρματα δίκτυα αισθητήρων και σε επίπεδο προσομοίωσης.

1.3 Δομή της εργασίας

Έχοντας παρουσιάσει μια εισαγωγή στα ασύρματα δίκτυα αισθητήρων καθώς και τον σκοπό της εργασίας, τα επόμενα κεφάλαια δομούνται ως ακολούθως.

Στο επόμενο κεφάλαιο, θα παρουσιαστούν τα βασικά χαρακτηριστικά αυτών των δικτύων σχετικά με τις ιδιαιτερότητές τους και τις προκλήσεις που συνιστούν την κατανάλωση ενέργειας να είναι ένα ιδιαίτερα σημαντικό πρόβλημα.

Εν συνεχεία, θα παρουσιαστούν τα βασικά χαρακτηριστικά των δύο κυριότερων προσομοιωτών, του PowerTossim και του Avrora, και στο 4^ο κεφάλαιο θα γίνει εκτενής αναφορά του τρόπου λειτουργίας τους και των βασικών αρχείων που καθορίζουν τα ενεργειακά μοντέλα που χρησιμοποιούνται στους δύο προσομοιωτές.

Τέλος, στο 5^ο κεφάλαιο παρουσιάζεται η δημιουργία εναλλακτικών τέτοιων αρχείων, που μπορεί να χρησιμοποιηθεί για την μελέτη διαφορετικών κόμβων – αισθητήρων ή για άλλες πλατφόρμες από τις ήδη υπάρχουσες.

ΑΣΥΡΜΑΤΑ ΔΙΚΤΥΑ ΑΙΣΘΗΤΗΡΩΝ

2.1 Χαρακτηριστικά των Ασύρματων Δικτύων Αισθητήρων

Η ραγδαία εξέλιξη της τεχνολογίας τα τελευταία χρόνια έχει ως αποτέλεσμα τη συνεχή δημιουργία και εισαγωγή στη ζωή μας νέων τεχνολογιών, μερικές από τις οποίες αποδεικνύονται περισσότερο από χρήσιμες. Σε αυτή την κατηγορία ανήκει και η τεχνολογία των κόμβων - αισθητήρων. Πρόκειται για μικρούς πομποδέκτες που έχουν τη δυνατότητα να συλλέγουν πληροφορίες από το περιβάλλον, να τις επεξεργάζονται, αλλά και να δημιουργούν μεταξύ τους ένα δυναμικό δίκτυο μέσω του οποίου μπορούν να ανταλλάσσουν τις πληροφορίες αυτές, επικοινωνώντας ασύρματα. Και φυσικά όλες τις παραπάνω λειτουργίες τις πραγματοποιούν με τη βοήθεια πηγής ενέργειας που εμπεριέχουν.

Οι αισθητήρες συνήθως χρησιμοποιούνται σε μεγάλες ποσότητες μαζί, και διασκορπίζονται κοντά στο φαινόμενο το οποίο θέλουμε να παρατηρήσουμε συλλέγοντας δεδομένα. Όπως προαναφέρθηκε, όλοι οι κόμβοι έχουν πομπούς και δέκτες για επικοινωνία, και το δίκτυο που δημιουργείται μεταξύ τους είναι δυναμικό, δηλαδή οι θέσεις τους δεν είναι προκαθορισμένες αλλά τυχαίες και δυνατόν να αλλάζουν. Τα δεδομένα που μπορούν να συλλέγουν ποικίλλουν και κατά συνέπεια είναι πάρα πολλές και οι δυνατές εφαρμογές της τεχνολογίας αυτής.

Για παράδειγμα, οι αισθητήρες αυτοί μπορούν να φανούν πολύ χρήσιμοι σε τομείς όπως η υγεία, ο στρατός, η μετεωρολογία, η ασφάλεια κλπ. Θα μπορούσαν να δίνουν σημαντικότερες πληροφορίες σε ένα γιατρό για κάποιον ασθενή, εφόσον τοποθετηθούν σε κατάλληλο σημείο. Επίσης είναι δυνατόν να ανιχνεύουν διάφορες χημικές ουσίες στον αέρα ή στο νερό ή ακόμα και να ανιχνεύουν επίπεδα μόλυνσης στην ατμόσφαιρα. Γενικότερα, η δυνατότητά τους τόσο να συλλέγουν δεδομένα όσο και να επικοινωνούν μεταξύ τους, τους κάνει ένα πολύ χρήσιμο εργαλείο σε πάρα πολλές περιπτώσεις.

Ένα ακόμα σημαντικό χαρακτηριστικό των αισθητήρων αυτών, είναι οι περιορισμένες δυνατότητες που έχουν όσον αφορά την ισχύ που μπορούν να καταναλώνουν. Οι πηγές που ενσωματώνονται μέσα στους αισθητήρες δεν είναι δυνατό να είναι πολύ ισχυρές, με αποτέλεσμα τελικά στα δίκτυα αυτά πρωταρχικός στόχος να είναι η επίτευξη της κατανάλωσης όσο το δυνατόν λιγότερης ενέργειας, στόχος ακόμα σημαντικότερος και από την ποιότητα των υπηρεσιών του δικτύου, καθώς σχετίζεται άμεσα με τη βιωσιμότητά του.

2.1.1 Περιβάλλοντα λειτουργίας

Έχοντας τονίσει την πληθώρα των εφαρμογών όπου ένα ασύρματο δίκτυο αισθητήρων μπορεί να λειτουργήσει, είναι προφανές ότι αντίστοιχα ευρύ είναι και το φάσμα των διαφορετικών περιβαλλόντων, στα οποία τα δίκτυα αυτά καλούνται να μεταδώσουν πληροφορίες. Το περιβάλλον στο οποίο μπορεί να λειτουργεί και να αναπτύσσεται ένα δίκτυο αισθητήρων είναι ιδιαίτερα σημαντικό και μπορεί να μας βοηθήσει να καταλάβουμε το μέγεθος των προβλημάτων που μπορεί να προκύψουν. Συγκεκριμένα, ένα τέτοιο δίκτυο μπορεί να βρίσκεται:

- Στο εσωτερικό μηχανών
- Στο βάθος του ωκεανού
- Μέσα σε έναν τυφώνα
- Σε ένα πεδίο μάχης
- Πάνω ή και μέσα στο σώμα ανθρώπων ή ζώων
- Μέσα σε ένα ποτάμι ή λίμνη

Αυτές είναι μερικές μόνο περιπτώσεις, από τις οποίες φαίνεται καθαρά ότι οι συνθήκες κάτω από τις οποίες λειτουργούν οι αισθητήρες είναι μπορεί να είναι αντίξοες: σε υψηλή πίεση, σε πολύ ζέστη ή κρύο, σε ανώμαλα εδάφη. Είναι επιτακτική λοιπόν η ανάγκη του σχεδιασμού των αισθητήρων με γνώμονα και την φυσική αντοχή τους στις καταστάσεις αυτές.

2.2 Εφαρμογές των Δικτύων Αισθητήρων

Η τεχνολογία των Ασύρματων Δικτύων Αισθητήρων (WSNs) μπορεί να εφαρμοστεί σε πολλές εφαρμογές του πραγματικού κόσμου και να φέρει στην επιφάνεια κάποιες εντελώς καινούριες.

Ένα κρίσιμο και πρωτεύον συστατικό των κόμβων των ασύρματων δικτύων αισθητήρων είναι ο αισθητήρας. Για πολλές παραμέτρους του φυσικού περιβάλλοντος υπάρχει η κατάλληλη τεχνολογία αισθητήρα που μπορεί να ενσωματωθεί σε ένα (WSN). Οι πιο ευρέως χρησιμοποιούμενοι είναι οι αισθητήρες θερμοκρασίας, υγρασίας, ήχου, πίεσης και οι χημικοί αισθητήρες.

Μια σύντομη λίστα με τις πιο αστικές εφαρμογές παρουσιάζεται παρακάτω :

Πρόληψη Καταστροφών. Μια από τις πιο συχνά αναφερόμενες εφαρμογές των (WSNs) είναι στην πρόληψη καταστροφών. Ένα τυπικό σενάριο για εφαρμογές αυτής της κατηγορίας είναι η ανίχνευση πυρκαγιών. Οι κόμβοι αισθητήρων είναι εξοπλισμένοι με θερμομέτρα και μπορούν να υπολογίσουν τη θέση τους τρέχοντας κάποιον αλγόριθμο εντοπισμού θέσης (localization). Τους κόμβους αυτούς μπορούμε να τους απλώσουμε σε ένα δάσος, πετώντας τους από ένα αεροπλάνο. Έτσι σχηματίζεται ένας θερμοκρασιακός χάρτης της περιοχής και σε περίπτωση υψηλών θερμοκρασιών και χαμηλής υγρασίας που υπονοούν πυρκαγιά ενημερώνουν τους πυροσβέστες.

Έλεγχος του περιβάλλοντος . Τα (WSNs) μπορούν να χρησιμοποιηθούν για να ελέγχουν το περιβάλλον ως προς τους χημικούς ίππους ή ακόμα και για το σχηματισμό μίας εικόνας ως προς τον αριθμό των διαφορετικών ειδών πανίδας και χλωρίδας μια περιοχής.

Ευφυή Κτίρια. Τα μεγάλα κτίρια συχνά καταναλώνουν μεγάλα ποσά ενέργειας εξαιτίας λανθασμένης χρήσης των συσκευών Air Conditioning (HVAC). Μια αποδοτικότερη, πραγματικού-χρόνου και ακριβέστερη παρακολούθηση της θερμοκρασίας, της υγρασίας και άλλων παραμέτρων μπορεί να μειώσει την

κατανάλωση ενέργειας. Επίσης, μπορούν να χρησιμοποιηθούν για την παρακολούθηση των μηχανικών καταπονήσεων σε κτίρια ή γέφυρες που βρίσκονται σε σεισμικά ενεργές ζώνες, ενώ άλλου τύπου αισθητήρες μπορούν να χρησιμοποιηθούν για τον εντοπισμό εγκλωβισμένων ανθρώπων σε περιπτώσεις σεισμού. Οι αισθητήρες μπορούν να τοποθετηθούν στα κτίρια τη στιγμή της κατασκευής τους ή αφού έχουν κατασκευαστεί. Σε αυτές τις εφαρμογές η εξοικονόμηση ενέργειας για τους αισθητήρες είναι πολύ σημαντική απαίτηση.

Διαχείριση Εγκαταστάσεων. Τα (WSNs) μπορούν να χρησιμοποιηθούν για εφαρμογές διαχείρισης μεγάλων εγκαταστάσεων, όπως θέματα ασφαλείας. Η είσοδος των ανθρώπων στις εγκαταστάσεις μπορεί να γίνεται χωρίς κλειδιά, αλλά με τη χρήση κάποιου πομπού, ενώ μπορούν να εντοπίζονται πιθανοί εισβολείς. Επίσης σε χημικές εγκαταστάσεις τα (WSNs) θα μπορούσαν να χρησιμοποιηθούν για τον εντοπισμό διαρροών.

Συντήρηση Μηχανών. Αισθητήρες μπορούν να τοποθετηθούν σε δυσπρόσιτα σημεία μηχανών για να ελέγχουν τους κραδασμούς που υποδεικνύουν ανάγκη για συντήρηση. Παραδείγματα τέτοιων μηχανών είναι οι αυτόματες μηχανές ή οι άξονες των τροχών των τρένων.

Εφαρμογές στη Γεωργία. Η εφαρμογή (WSNs) σε καλλιεργήσιμες εκτάσεις με τοποθέτηση αισθητήρων μέτρησης υγρασίας και ανάλυσης της σύστασης του εδάφους επιτρέπει την ακριβέστερη και αποδοτικότερη λίπανση και άρδευση των εκτάσεων. Επίσης, η εκτροφή ζώων μπορεί να ωφεληθεί τοποθετώντας αισθητήρες στα ώα που ελέγχουν την κατάσταση της υγείας τους.

Εφαρμογές στον τομέα της υγείας. Η χρήση (WSNs) στον τομέα της υγείας μπορεί να αποδειχτεί πολύ ωφέλιμη. Όμως υπάρχουν αρκετά ηθικά διλήματα πάνω στο θέμα αυτό. Οι πιθανές εφαρμογές εκτείνονται από την άμεση τοποθέτηση αισθητήρων στον ίδιο τον ασθενή για την παρακολούθηση της υγείας του και ίσως αυτόματη χορήγηση φαρμάκων, μέχρι την παρακολούθηση των ιατρών και των ασθενών στα νοσοκομεία.

Ευφυή οδικά συστήματα. Στα ευφυή οδικά συστήματα αισθητήρες τοποθετούνται στους δρόμους, ακόμα και στα κράσπεδα των δρόμων οι οποίοι συλλέγουν πληροφορίες για την κίνηση και την κατάσταση του

οδικού δικτύου γενικότερα και επικοινωνούν με τους οδηγούς δίνοντάς τους χρήσιμες πληροφορίες.

Στρατιωτικές Εφαρμογές. Τα (WSNs) μπορούν να είναι ενιαίο και αναπόσπαστο τμήμα των στρατιωτικών συστημάτων. Τα χαρακτηριστικά των (WSNs), όπως είναι η γρήγορη τοποθέτηση τους, η αυτοοργάνωση και η ανοχή στα σφάλματα, τα μετατρέπουν σε μια υποσχόμενη τεχνολογία για τα στρατιωτικά συστήματα. Κάποιες από τις πιθανές στρατιωτικές εφαρμογές τους είναι η παρακολούθηση της κατάστασης των εξοπλισμών και των πολεμοφοδίων, η στενή παρακολούθηση του πεδίου της μάχης, η αναγνώριση των εχθρικών δυνάμεων, η εκτίμηση των καταστροφών μετά από μάχη καθώς και ο εντοπισμός και η αναγνώριση χημικής, ατομικής ή βιολογικής επίθεσης.

2.3 Προβλήματα και Προκλήσεις

Η ιδιαίτερη δομή των ασύρματων δικτύων αισθητήρων (WSNs) έχει ως αποτέλεσμα τη δημιουργία ξεχωριστών απαιτήσεων και προβλημάτων προς επίλυση, σε σχέση με τις υπάρχουσες δομές. Με μια πρώτη προσέγγιση, τα (WSNs) παρουσιάζουν σημαντικές ομοιότητες με τα ad – hoc δίκτυα. Παρόλα αυτά, και παρά την φαινομενική ομοιότητα τους, τα δίκτυα αισθητήρων που μελετάμε διαφέρουν αρκετά, τόσο με τα παραδοσιακά δίκτυα, όσο και με τα ad – hoc δίκτυα, ειδικά σε θέματα κατανάλωσης και απαιτήσεων ενέργειας. Γι' αυτό, εξάλλου, και τα πρωτόκολλα αλλά και οι προσομοιωτές που έχουν προταθεί κατά καιρούς και χρησιμοποιούνται στα ad-hoc δίκτυα δεν είναι κατάλληλα για τους αισθητήρες αυτούς.

2.3.1 Διαφορές με Υπάρχοντα ad – hoc Δίκτυα

Στα δίκτυα αισθητήρων συνήθως έχουμε πολύ μεγαλύτερο αριθμό κόμβων απ' ότι στα ad-hoc δίκτυα. Για παράδειγμα, ενώ στα δίκτυα αισθητήρων ο αριθμός των κόμβων μπορεί να ξεκινάει από μερικές εκατοντάδες και να φτάνει τις πολλές χιλιάδες, στα ad-hoc δίκτυα ο αριθμός αυτός δεν μπορεί να ξεπερνάει τις λίγες

εκατοντάδες. Εξαιτίας της διαφοράς αυτής εξάλλου, οι κόμβοι-αισθητήρες δεν έχουν ταυτότητα (ID) μέσα στο δίκτυο, όπως συμβαίνει με τους κόμβους των ad-hoc δικτύων. Στα δίκτυα αισθητήρων άλλωστε δεν χρειάζεται να υπάρχουν ταυτότητες στους κόμβους, καθώς δεν ενδιαφέρει συνήθως η πληροφορία που δίνει ο κάθε συγκεκριμένος κόμβος, αλλά για παράδειγμα η τοποθεσία των κόμβων των οποίων οι πληροφορίες έχουν κάποια ιδιαίτερα χαρακτηριστικά.

Μια άλλη ιδιαίτερα σημαντική διαφορά μεταξύ των δύο τύπων δικτύων είναι το γεγονός ότι στα ad-hoc δίκτυα οι κόμβοι είναι όλοι και συνεχώς σε κίνηση. Αυτό δε συμβαίνει στα δίκτυα αισθητήρων, όπου είναι δυνατό, ανάλογα με την περίπτωση και την εφαρμογή του δικτύου, ένα μέρος των κόμβων να κινείται και οι υπόλοιποι να παραμένουν σταθεροί στις θέσεις τους. Αυτό έχει ως αποτέλεσμα να δημιουργούνται πολλές διαφορές και στην τοπολογία των δικτύων αυτών. Κατά συνέπεια, και τα πρωτόκολλα που χρησιμοποιούνται στα ad-hoc δίκτυα σχετικά με την τοπολογία (όπως π.χ. για τον εντοπισμό θέσης-localization) θα έχουν διαφορές από αυτά που χρησιμοποιούνται στα δίκτυα αισθητήρων.

Κάτι εξίσου σημαντικό που διαχωρίζει σαφώς τα δύο αυτά είδη δικτύων είναι το γεγονός ότι η γενικότερη λειτουργία των ad-hoc δικτύων πραγματοποιείται με σκοπό την βελτίωση της ποιότητας των υπηρεσιών (Quality Of Service). Αυτός είναι ο βασικός γνώμονας τόσο για την οργάνωση του δικτύου, όσο και για την διαδικασία μετάδοσης πληροφοριών. Η ενέργεια που καταναλώνεται κατά τη λειτουργία του δικτύου, είναι δευτερεύουσας σημασίας. Αντίθετα, στα δίκτυα αισθητήρων ο βασικός παράγοντας που λαμβάνεται υπ' όψη είναι η διάρκεια ζωής του δικτύου και ο περιορισμός σε μεγάλο βαθμό της κατανάλωσης ενέργειας. Αυτό συμβαίνει, καθώς στα δίκτυα αυτά οι πηγές ενέργειας των κόμβων είναι περιορισμένες και στις περισσότερες περιπτώσεις αδύνατο να αντικατασταθούν, όπως μπορεί να συμβεί συνήθως στους κόμβους των ad-hoc δικτύων, αλλά και διότι τα περιβάλλοντα στα οποία λειτουργούν τα δίκτυα αισθητήρων σε πολλές περιπτώσεις μπορεί να είναι δυσμενή. Αναγκαστικά λοιπόν, η φροντίδα για την ποιότητα των υπηρεσιών έρχεται σε δεύτερη μοίρα.

Ιδιαίτερα σημαντικές είναι οι διαφορές ανάμεσα στα ad-hoc δίκτυα και στα δίκτυα αισθητήρων όσον αφορά τη ροή των δεδομένων. Στα ad-hoc δίκτυα, υπάρχει συνήθως πυκνή και συνεχόμενη ροή πληροφοριών, ενώ αντίθετα στα δίκτυα αισθητήρων η ροή των δεδομένων είναι σίγουρα μικρότερη ενώ επιπλέον δεν είναι ντετερμινιστική και συνήθως προσεγγίζεται με στοχαστικά μοντέλα, κάτι που έχει

άμεση συνέπεια τόσο στην καταναλισκόμενη ενέργεια όσο και στα αντίστοιχα μοντέλα προσομοίωσης που πρέπει να χρησιμοποιούνται. Αυτός είναι και ο λόγος που το εύρος ζώνης που απαιτείται για την μετάδοση πληροφοριών στα δίκτυα αισθητήρων είναι σχετικά μικρό, της τάξης των 1-100 kbps.

Τα δύο αυτά είδη δικτύων διαφέρουν σημαντικά και στις δυνατότητες επεξεργαστικής ισχύος, υπολογισμού και μνήμης. Συγκεκριμένα, οι αισθητήρες έχουν σαφώς περιορισμένες δυνατότητες στους παραπάνω τομείς σε σχέση με τους κόμβους των ad-hoc δικτύων, και αυτό είναι συνέπεια κατά κύριο λόγο των περιορισμών στις πηγές ενέργειας που συναντάμε στα δίκτυα αισθητήρων.

Τέλος, είναι δυνατόν να συναντήσει κάποιος πολλές άλλες μικροδιαφορές με την περαιτέρω μελέτη των δικτύων αισθητήρων και των ad-hoc δικτύων, όπως για παράδειγμα η πυκνότητα των κόμβων, που είναι πολύ μεγαλύτερη στα (WSNs), κάτι απόλυτα φυσιολογικό αν αναλογιστεί κανείς και τη σημαντική διαφορά στον αριθμό των κόμβων.

2.3.2 Σημαντικές Προκλήσεις

Τα Ασύρματα Δίκτυα Αισθητήρων παρουσιάζουν ένα εντελώς διαφορετικό σύνολο από περιορισμούς σε σύγκριση με τους περιορισμούς που παρουσιάζονται στα παραδοσιακά δίκτυα. Το πιο σημαντικό από αυτά είναι η ενέργεια.

Η διαφορετικότητα των (WSNs) σε σχέση με τα υπάρχοντα δίκτυα συνεπάγεται την εμφάνιση νέων προκλήσεων και την ανάγκη επανασχεδίασης πολλών τμημάτων της τηλεπικοινωνιακής αλυσίδας έτσι ώστε να διασφαλιστεί η σωστή και αποδοτική λειτουργία τους. Έτσι, και με βάση τα γενικά χαρακτηριστικά των δικτύων αυτών αλλά και τις διαφορές τους με τα υπάρχοντα δίκτυα, όπως περιγράφηκαν πιο πάνω, προκύπτει ότι είναι απαραίτητη η εκ νέου σχεδίαση των περισσότερων στρωμάτων λειτουργίας τους.

Στα πλαίσια του Φυσικού Στρώματος, απαιτείται η σχεδίαση και υλοποίηση τεχνικών που να λαμβάνουν υπόψη τα ιδιαίτερα χαρακτηριστικά των κόμβων – αισθητήρων. Προσαρμοστικές τεχνικές που θα δίνουν περισσότερη έμφαση στην εξοικονόμηση ενέργειας και στην απλότητα υλοποίησης θα πρέπει να αναπτυχθούν. Επίσης, τα ιδιαίτερα και πολλές φορές δυσπρόσιτα περιβάλλοντα στα οποία καλείται

να λειτουργήσει ένα (WSN) ανάλογα με την εφαρμογή, δημιουργούν επιπλέον προκλήσεις σχετικές με τον δίαυλο επικοινωνίας και κατά προέκταση με τις απαιτήσεις στη σχεδίαση του τηλεπικοινωνιακού τμήματος των κόμβων.

Παράλληλα, η επιλογή του καταλληλότερου τρόπου διαμόρφωσης, οι τεχνικές για την μελέτη της ραδιοκάλυψης αλλά και η ανάγκη για το σχεδιασμό κυκλωμάτων πολύ μικρών σε διάσταση και πολύ οικονομικών σε κατανάλωση ενέργειας, είναι τομείς που επιδέχονται πολλές βελτιώσεις.

Ομοίως, και όσον αφορά στο Στρώμα Διασύνδεσης Δεδομένων, κρίνεται απαραίτητη η επανασχεδίαση των πρωτοκόλλων μετάδοσης και άρα και η χρήση νέων προσομοιωτών για τη μελέτη λειτουργίας τους. Τα πρωτόκολλα που χρησιμοποιούνται στα σημερινά δίκτυα χαρακτηρίζονται από ιδιαίτερη πολυπλοκότητα και υπολογιστικές απαιτήσεις, με αποτέλεσμα να μην είναι εφαρμόσιμα σε δίκτυα αποτελούμενα από μικρούς κόμβους που καλούνται να λειτουργήσουν βασιζόμενοι σε μικρές και δύσκολα αντικαταστάσιμες μπαταρίες. Συνεπώς, απαιτείται η εφαρμογή απλούστερων τεχνικών που θα στοχεύουν στην βελτίωση της βιωσιμότητας των δικτύων.

Ιδιαίτερα απαιτητική είναι και η σχεδίαση των τεχνικών σχετικά με το Στρώμα Δικτύου και την δρομολόγηση των δεδομένων μέσα σε ένα (WSNs). Η μετάδοση πληροφορίας με χρήση πολλαπλών βημάτων, η πιθανή μετακίνηση μέρους των κόμβων λόγω περιβάλλοντος ή η παύση λειτουργίας άλλων λόγω εξάντλησης των ενεργειακών αποθεμάτων, αποτελούν παράγοντες που συνεπάγονται την ανάγκη για νέες τεχνικές προώθησης της πληροφορίας, με προσαρμοστικά χαρακτηριστικά και ευελιξία. Επιπλέον, η δομή των δικτύων αυτών αφήνει ανοιχτό το ενδεχόμενο και για την εφαρμογή «έξυπνων» αλγορίθμων δρομολόγησης, στους οποίους θα αναπτύσσονται διάφορες μορφές συνεργασίας μεταξύ των κόμβων – αισθητήρων. Τέλος, και σε ανώτερα στρώματα λειτουργίας ενός δικτύου, εμφανίζονται αντίστοιχες προκλήσεις που η επιστημονική κοινότητα καλείται να αντιμετωπίσει τα αμέσως επόμενα χρόνια.

Και σε διαστρωματικό επίπεδο όμως, οι προκλήσεις που συναντώνται σχετικά με τη σχεδίαση αλγορίθμων για τη λειτουργία ενός (WSN) είναι σημαντικότερες. Καταρχήν, με κάθε νέα τεχνολογία που προτείνεται, πρέπει να εξασφαλίζεται η αύξηση της χωρητικότητας τόσο των επιμέρους ζεύξεων όσο και του συνολικού δικτύου. Επιπροσθέτως, η ποιότητα των παρεχόμενων υπηρεσιών μπορεί μεν να μην είναι πρωτεύουσας σημασίας στα (WSNs), αλλά είναι δεδομένο ότι πρέπει

πάντοτε να λαμβάνεται υπόψη κατά τον σχεδιασμό οποιωνδήποτε τεχνικών. Ένας άλλος σημαντικός παράγοντας είναι η αύξηση της αξιοπιστίας και της ασφάλειας στη μετάδοση των διάφορων δεδομένων, από τη στιγμή που ένα (WSNs) μπορεί να χρησιμοποιείται για παράδειγμα σε ποικίλες στρατιωτικές εφαρμογές. Τέλος, ανεξάρτητα από το στρώμα στο οποίο αναφερόμαστε, βασικός γνώμονας για οποιαδήποτε προτεινόμενη τεχνολογία, είναι η αύξηση του χρόνου ζωής του δικτύου.

Συνοψίζοντας, η σχεδίαση, οργάνωση και υλοποίηση τεχνικών και πρωτοκόλλων για ένα (WSNs), και κατά συνέπεια η σχεδίαση προσομοιωτών για τη μελέτη των δικτύων αυτών παρουσιάζει σημαντικά προβλήματα και κενά σε σχέση με τις ως τώρα τεχνολογικές λύσεις. Σε αυτό συμβάλλουν οι ιδιαίτερες απαιτήσεις των (WSNs) σε εξοικονόμηση ενέργειας, όπως προκύπτουν από την ανάγκη για αύξηση της βιωσιμότητάς τους με γνώμονα την λειτουργία με τροφοδότηση από μπαταρίες περιορισμένων δυνατοτήτων.

Κατά προέκταση, η δημιουργία των κατάλληλων εργαλείων προσομοίωσης για την μελέτη της καταναλισκόμενης ενέργειας σε τέτοια δίκτυα, είναι ιδιαίτερα σημαντική. Σε αυτό το πλαίσιο, στα επόμενα κεφάλαια θα μελετήσουμε τις βασικές αρχές λειτουργίας δύο εκ των σημαντικότερων τέτοιων εργαλείων, του PowerTossim και του Avrora.

Οι Προσομοιωτές Powertossim και Avrora

Η ανάπτυξη σωστών εργαλείων προσομοίωσης ήταν ένα βήμα κλειδί για την περαιτέρω έρευνα και ανάπτυξη των συστημάτων. Γενικά, η εξομοίωση μπορεί να παρέχει ένα τρόπο μελέτης των εναλλακτικών τρόπων σχεδίασης ενός συστήματος, σε ένα ελεγχόμενο από όλες τις απόψεις περιβάλλον. Έτσι ελέγχεται τι επιφέρουν οι διάφορες ρυθμίσεις σε ένα σύστημα που είναι δύσκολο να κατασκευαστεί και να ρυθμιστεί επανειλημμένα. Επίσης παρατηρούνται οι διάφορες αλληλεπιδράσεις του συστήματος για κάθε σύνολο ρυθμίσεων που εφαρμόζεται, κάτι που μπορεί να είναι δύσκολο να ανιχνευθεί σε ένα πραγματικό σύστημα .

Τέτοια εργαλεία προσομοίωσης είναι πολύ χρήσιμα και στα δίκτυα ασύρματων αισθητήρων, τα οποία λόγω της φύσης κατασκευής και εγκατάστασης τους είναι δύσκολο και οικονομικά ασύμφορο να μελετηθούν σε πραγματική κλίμακα, ώστε να βρεθούν οι κατάλληλες ρυθμίσεις που θα τα καθιστούν ικανά να φέρουν σε πέρας την αποστολή τους. Επίσης καθώς τέτοια συστήματα είναι ακόμα σε στάδιο μελέτης και λίγα από αυτά έχουν πραγματοποιηθεί, η εξομοίωση βοηθά ακόμα περισσότερο προς αυτή την κατεύθυνση. Τέτοια εργαλεία θα πρέπει να μπορούν να μελετούν συμπεριφορές της εφαρμογής μέχρι το χαμηλότερο επίπεδο των συστατικών ενός κόμβου, καθώς επίσης θα πρέπει να μπορούν να χειριστούν ένα μεγάλο πλήθος κόμβων και τις αλληλεπιδράσεις μεταξύ τους αλλά και με το περιβάλλον.

3.1 Λίγα λόγια για τον TOSSIM

Το βασικό εργαλείο μέσω του οποίου γίνεται η προσομοίωση γεγονότων στον TinyOS είναι ο TOSSIM. Πρόκειται για έναν διακριτό εξομοιωτή γεγονότων (events) για ασύρματα δίκτυα αισθητήρων, ο οποίος αντιλαμβάνεται τη συμπεριφορά και τη διαδραστικότητα των πομποδεκτών σε επίπεδο bit, σε αντίθεση με άλλους εξομοιωτές που κάνουν το ίδιο σε επίπεδο πακέτου.

Η αρχιτεκτονική του TOSSIM αποτελείται από πέντε μέρη:

- Οι Γράφοι συστατικών του TinyOS
- Μοντέλο εκτέλεσης
- Μοντέλα RF και ADC
- Συστατικά «γενίκευσης» υλικού
- Υπηρεσίες επικοινωνίας

Το 1^ο μέρος αφορά στην υποστήριξη της μεταγλώττισης γράφων των συστατικών του TinyOS στην εξομοίωση από το χρήστη. Ένα συστατικό του TinyOS αποτελείται από πέντε αλληλοσχετιζόμενα μέρη : χειριστές εντολών (command handlers), χειριστές γεγονότων (event handlers), ένα συμπυκνωμένο προσωπικό πλαίσιο δεδομένων, μια δομή από προσωπικές μεταβλητές και μια ομάδα απλών tasks. Τα tasks, οι χειριστές εντολών και γεγονότων εκτελούνται στο περιβάλλον του πλαισίου. Οι εντολές και τα γεγονότα είναι μηχανισμοί επικοινωνίας μεταξύ των συστατικών και τα tasks χρησιμοποιούνται εσωτερικά.

Το 2^ο μέρος, το μοντέλο εκτέλεσης, αφορά σε μια διακριτή ουρά γεγονότων υπεύθυνη για την ενεργοποίηση της εξομοίωσης. Τα μοντέλα του TOSSIM προκαλούν διακοπή μέσω γεγονότων εξομοίωσης, καθένα από τα οποία σχετίζεται με ένα mote, αφού ο προγραμματιστής (scheduler) εκτέλεσης των γεγονότων εκτελεί tasks στην ουρά των tasks του TinyOS, και τα εκτελεί όλα (τα γεγονότα εξομοίωσης) σε σειρά FIFO. Η ουρά των tasks του σχετιζόμενου mote εκτελείται μαζί με το γεγονός, μέχρι ολόκληρη η ουρά να έχει τελειώσει.

Το 3^ο μέρος, δηλαδή το RF και ο μετατροπέας αναλογικού-σε-ψηφιακό (ADC) χρησιμοποιούνται για να μοντελοποιήσουν τα χαρακτηριστικά των TinyOS motes. Τα RF μοντέλα ορίζουν τα χαρακτηριστικά που αφορούν στη μετάδοση από κόμβο σε κόμβο. Ένα ελάττωμα στο RF μοντέλο του TOSSIM είναι ότι η απόσταση δεν επηρεάζει την ισχύ του σήματος δημιουργώντας παρεμβολές γενικά χειρότερες από πραγματικές συμπεριφορές. Υπάρχουν δύο είδη RF μοντέλων.

Το «απλό» και αυτό που προκαλεί απώλειες ενέργειας, το «lossy». Το απλό RF μοντέλο τοποθετεί όλους τους κόμβους σε ένα μοναδικό «κελί», όπου κάθε bit που μεταδίδεται, λαμβάνεται χωρίς καθόλου σφάλμα. Ενώ, όμως, κανένα bit δεν καταστρέφεται λόγω σφάλματος, δύο motes που εκπέμπουν ταυτόχρονα οδηγούν στο πρόβλημα της επικάλυψης των σημάτων, αλλά η πιθανότητα να εκπέμπουν δύο motes είναι πολύ χαμηλή εξαιτίας του πρωτοκόλλου CSMA του TinyOS.

Το «lossy» RF μοντέλο τοποθετεί τους κόμβους σε έναν κατευθυνόμενο γράφο. Κάθε ακμή (x,y) του γράφου σημαίνει ότι το σήμα του κόμβου x μπορεί να ακουστεί στον κόμβο y. Κάθε ακμή έχει μια τιμή που αναπαριστά την πιθανότητα ένα bit που αποστέλλεται από τον κόμβο x να είναι κατεστραμμένο (αντεστραμμένο), όταν ο y το ακούσει.

Ο TOSSIM παρέχει δύο ADC μοντέλα : το τυχαίο (random) και το γενικής χρήσης (generic). Ο ADC έχει διάφορα κανάλια που μπορούν να δειγματοληφθούν. Στο μοντέλο αυτό, όποτε ένα κανάλι δειγματοληπτείται επιστρέφει μια τυχαία 10-bit τιμή. Το γενικό μοντέλο παρέχει τις τυχαίες τιμές. Επιπλέον, παρέχει την πιθανότητα να ενεργοποιηθεί από εξωτερικές εφαρμογές, θέτοντας την τιμή για οποιαδήποτε ADC θύρα σε οποιοδήποτε mote.

Αναφορικά με το 4^ο μέρος, τα συστατικά «γενίκευσης» υλικού (hardware abstraction components), πρέπει να σημειώσουμε ότι το TinyOS θεωρεί κάθε πόρο υλικού ως συστατικό (component). Ο TOSSIM αντικαθιστά ένα μικρό μόνο αριθμό από αυτά τα συστατικά, όπως το ADC, το ρολόι, το EEPROM, κτλ., και προσομοιώνει τη συμπεριφορά του υποκείμενου υλικού. Ο TOSSIM μοντελοποιεί αυτά τα συστατικά στο κομμάτι των συστατικών «γενίκευσης» υλικού της αρχιτεκτονικής.

Τέλος, σε σχέση με τις υπηρεσίες επικοινωνίας, αυτές αφορούν στις απαραίτητες διαδικασίες για να επιτρέψει σε εφαρμογές που τρέχουν στο PC να επικοινωνήσει με τον TOSSIM μέσω TCP/IP.

Ο TOSSIM διαβαθμίζεται σε χιλιάδες κόμβους, που μεταγλωττίζονται κατευθείαν από το TinyOS. Ο TOSSIM εξομοιώνει σε επίπεδο bit και αντικαθιστά το υλικό με συστατικά λογισμικού καθιστώντας δυνατές κλιμακούμενες εξομοιώσεις δικτύων αισθητήρων. Ο TOSSIM στοχεύει στην επίτευξη τεσσάρων στόχων:

Κλιμάκωση (scalability) – να είναι δυνατόν να χειριστεί μεγάλα δίκτυα από χιλιάδες κόμβους σε μεγάλο εύρος ρυθμίσεων.

Αρτιότητα (completeness) – να αντιλαμβάνεται με ακρίβεια τη συμπεριφορά ολόκληρου του συστήματος.

Αξιοπιστία (fidelity) – να αντιλαμβάνεται τη συμπεριφορά του δικτύου σε λειτουργία fine grain αποκαλύπτοντας απροσδόκητες αλληλεπιδράσεις. Ο TOSSIM βοήθησε στην αποσφαλμάτωση (debugging) των προβλημάτων στοίβας των δικτύων του TinyOS. Εξομοιώνει το δίκτυο σε επίπεδο bit.

Γεφύρωση (bridging) – γεφυρώνει το κενό ανάμεσα στον αλγόριθμο και την υλοποίηση, επιτρέποντας σχεδιαστές να ελέγξουν και να επιβεβαιώσουν τον κώδικα που θα τρέξει σε πραγματικό υλικό. Ο ελεγμένος κώδικας μπορεί να τοποθετηθεί κατευθείαν χωρίς καμία αλλαγή στην εφαρμογή.

3.1.1 Ο PowerTOSSIM

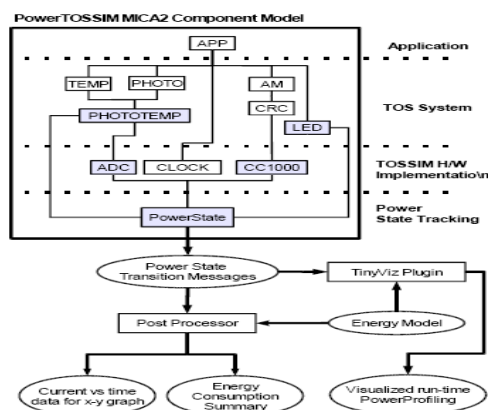
Στην ενότητα αυτή, θα περιγράψουμε την ολοκληρωμένη σχεδίαση του PowerTOSSIM, όπως δίνεται από τους σχεδιαστές του, δηλαδή την ερευνητική ομάδα των Schnayder et al. [1] του πανεπιστημίου του Harvard. Ο PowerTOSSIM περιλαμβάνει την τροποποίηση του κώδικα του TinyOS για να παρακολουθούνται μεταβάσεις στην ενεργειακή κατάσταση του υλικού, έναν ακριβή μηχανισμό μέτρησης του κύκλου της CPU, ο οποίος βασίζεται σε περιγραφή επιπέδου βασικού δομικού στοιχείου, και εργαλεία ανάλυσης για την οπτικοποίηση και την ανάλυση των αποτελεσμάτων της κατανάλωσης ενέργειας σε κάθε mote.

3.1.2 Η αρχιτεκτονική του PowerTOSSIM

Η Εικόνα δείχνει την αρχιτεκτονική του PowerTOSSIM. Ο PowerTOSSIM παρακολουθεί την ενεργειακή κατάσταση κάθε συστατικού (component) του υλικού των εξομοιούμενων motes, παράγοντας συγκεκριμένα μηνύματα μεταβάσεων από μία κατάσταση σε μία άλλη, τα οποία καταγράφονται κατά την εκτέλεση της προσομοίωσης. Αυτό επιτυγχάνεται με την τροποποίηση των εικονικών συστατικών

υλικού του TOSSIM μέσω κλήσεων σε ένα νέο συστατικό, το PowerState, το οποίο παρακολουθεί την ενεργειακή κατάσταση του κάθε mote και την καταγράφει σε ένα αρχείο κατά τη διάρκεια του τρεξίματος. Η εκτίμηση της χρήσης της CPU είναι κάπως πιο πολύπλοκη : αφού ο PowerTOSSIM τρέχει το λογισμικό του mote σε ένα τοπικό πρόγραμμα στη host μηχανή, δεν έχει πληροφορίες για το χρόνο που ένα mote χρησιμοποιεί τη CPU. Έτσι, η μοντελοποίηση της επιτυγχάνεται με χαρτογράφηση των βασικών κομματιών (blocks) της προσομοίωσης σε μετρήσεις κύκλων στο αντίστοιχο πρόγραμμα του mote.

Τα δεδομένα της ενεργειακής κατάστασης που δημιουργεί ο PowerTOSSIM μπορούν να συνδυαστούν με ένα ενεργειακό μοντέλο, για να αποφασιστεί η ανα-mote και ανα-component ενεργειακή κατανάλωση. Αυτή η επεξεργασία μπορεί να πραγματοποιηθεί χρησιμοποιώντας offline εργαλεία για να αποκτηθούν λεπτομερή «ίχνη» (traces) της κατανάλωσης ενέργειας του κάθε συστατικού του κάθε mote ή να περαστεί στο πρόγραμμα οπτικοποίησης TinyViz για να φανεί η κατανάλωση ενέργειας σε πραγματικό χρόνο.



Εικόνα – Αρχιτεκτονική του PowerTOSSIM . Τα components της εξομοίωσης (RF, sensors, LEDs, κτλ) καλούν το module PowerState και αυτό εκπέμπει μηνύματα αλλαγής ενεργειακής κατάστασης για το καθένα. Αυτά τα μηνύματα μπορούν να συνδυαστούν με ένα ενεργειακό μοντέλο για την παραγωγή λεπτομερών δεδομένων κατανάλωσης ενέργειας ή οπτικοποιήσεων.

Επιλέγουμε να αποδεσμεύσουμε την παραγωγή και επεξεργασία των δεδομένων της μετάβασης μεταξύ ενεργειακών καταστάσεων για δύο λόγους : αποδοτικότητα και ελαστικότητα.

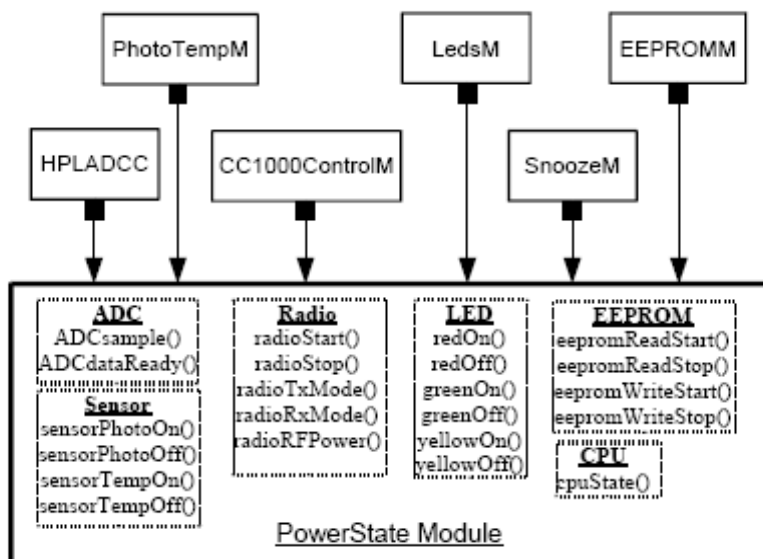
Αποδοτικότητα : Ένα από τα σημαντικά χαρακτηριστικά που ο PowerTOSSIM κληρονομεί από τον TOSSIM είναι η πολύ υψηλή αποδοτικότητά του στην εξομοίωση μεγάλων δικτύων που κλιμακώνονται σε χιλιάδες κόμβων. Για τη διατήρηση αυτής είναι σημαντικό να αποφεύγονται υψηλά overheads στην ίδια την εξομοίωση. Ομοίως, με την εκτέλεση της εξομοίωσης ως τοπικό πρόγραμμα αποφεύγουμε το overhead της εξομοίωσης σε επίπεδο εντολής.

Ελαστικότητα : Είναι, επίσης, σημαντικό να παρέχεται υψηλός βαθμός ελαστικότητας για την καταμέτρηση και την μοντελοποίηση της ενεργειακής κατάστασης του mote. Δεν υποτέθηκε η χρήση μιας πλατφόρμας, αφού συνέχεια χρησιμοποιούνται νέα σχέδια. Με την αποσυνδεδεμένη σχεδίαση είναι δυνατό να εκτιμηθεί η ενεργειακή αποδοτικότητα πιθανών σχεδιάσεων υλικού, μόνο βάζοντας ένα νέο ενεργειακό μοντέλο στα εργαλεία ανάλυσης. Η εξομοίωση δε χρειάζεται να επανεκτελεστεί.

3.1.3 Η μονάδα (module) PowerState

Για την αποφυγή της ύπαρξης διασκορπισμένου κώδικα για την παρακολούθηση της ενέργειας σε ολόκληρο τον εξομοιωτή, δημιουργήθηκε ένα μόνο module του TinyOS, το PowerState, το οποίο καλούν τα υπόλοιπα components του TinyOS για να καταγράψουν αλλαγές στην ενεργειακή τους κατάσταση. Κάθε συνάρτηση ελέγχει αν είναι ενεργοποιημένο το power profiling και, αν ναι, εκπέμπει ένα μήνυμα καταγραφής (log message) που περιέχει τον αριθμό του mote, τη συγκεκριμένη ενεργειακή μετάβαση και το χρόνο της προσομοίωσης. Ένα απόσπασμα αυτού δίνεται παρακάτω :

```
O: POWER: Mote 0 LED_STATE RED_OFF at 18677335
O: POWER: Mote 0 LED_STATE YELLOW_OFF at 18677335
O: POWER: Mote 0 ADC_SAMPLE RSSI_PORT at 18990479
O: POWER: Mote 0 ADC_DATA_READY at 18990679
O: POWER: Mote 0 RADIO_STATE TX at 18993551
O: POWER: Mote 0 RADIO_STATE RX at 19199375
```

Εικόνα – Η διασύνδεση (wiring) του PowerState module με τα υπόλοιπα components

Απαιτούνται πολύ λίγες τροποποιήσεις στον αυθεντικό κώδικα του TOSSIM για την παρακολούθηση της ενεργειακής κατάστασης των διαφόρων συσκευών. Συνολικά, 46 γραμμές κώδικα άλλαξαν ή προστέθηκαν σε υπάρχοντα TinyOS components. Αυτές οι γραμμές ήταν απλά διασυνδέσεις και κλήσεις συναρτήσεων σε εντολές του module PowerState. Ολόκληρο το PowerState αποτελείται από περίπου 400 γραμμές για τη διαμόρφωση και εκτύπωση μηνυμάτων κατάστασης.

3.1.4 Αποτίμηση του PowerTOSSIM

Για την αξιολόγηση του προσομοιωτή PowerTOSSIM παρατίθεται ο παρακάτω πίνακας, ο οποίος δείχνει ότι ο PowerTOSSIM επιτυγχάνει πολύ καλή ακρίβεια, με μέσο σφάλμα 4.7% και μέγιστο 13% σε σχέση με το πραγματικό mote. Ένα μέρος της διαφοράς ανάμεσα στην προσομοιούμενη και μετρούμενη ενέργεια μπορεί να αποδοθεί σε διακυμάνσεις τάσης, θόρυβο και σφάλματα στρογγυλοποίησης στο στήσιμο του πειράματος. Άλλες διαφορές μπορεί να οφείλονται σε ανακρίβειες στο ενεργειακό μοντέλο ή στη μέτρηση των κύκλων της CPU.

Benchmark	Simulated	Measured	Error (%)
Beacon	92.93	106.73	-12.9
Blink	940.26	931.72	0.85
BlinkTask	940.28	917.90	2.5
CntToLeds	1336.49	1330.00	0.45
CntToLedsAndRfm	2620.37	2562.00	2.3
CntToRfm	2028.09	1985.00	2.1
Oscilloscope	867.94	801.60	8.3
OscilloscopeRF	2136.45	2021.90	5.7
Sense	865.59	900.72	-3.8
SenseLightToLog	2133.89	2005.26	6.4
SenseTask	865.62	944.74	-8.3
SenseToLeds	868.70	977.73	-11.1
SenseToRfm	2152.27	2059.16	4.5
Average			4.7
TinyDB (idle)	2001.31	2275.55	-12.1
TinyDB (select light)	2144.86	2465.30	-13.0
Surge	2089.09	2028.40	3.0
Average			9.5

Πίνακας – Αποτελέσματα προσομοίωσης και μέτρησης για διάφορες εφαρμογές ύστερα από αντίστοιχο χρόνο 60 λεπτών. Όλες οι τιμές ενέργειας σε millijoules.

3.2 Ο AVRORA

Από την άλλη μεριά, οι ερευνητές που δημιούργησαν τον Avgora, δηλώνουν πως ο simulator αυτός αγγίζει την απόδοση του TOSSIM ενώ παράλληλα διατηρεί την ακρίβεια του «κύκλου».

Ο simulator αυτός προσομοιώνει ένα δίκτυο αισθητήρων, τρέχει τα ακριβή προγράμματα των μικροελεγκτών (και όχι μοντέλα σε λογισμικό), και πραγματοποιεί ακριβείς προσομοιώσεις αναφορικά τόσο με τις συσκευές όσο και με την επικοινωνία τους [4].

Ο Avgora έχει υλοποιηθεί σε Java, κάτι που βοηθάει στην ευχρηστία και την επεκτασιμότητά του. Αντίθετα, ο TOSSIM είναι υλοποιημένος σε C. Επιπλέον είναι ιδιαίτερα χρήσιμο το γεγονός ότι χαρακτηρίζεται από ανεξαρτησία στις υλοποιήσεις, ενώ για παράδειγμα ο TOSSIM μπορεί να προσομοιώσει μόνο TinyOS περιβάλλοντα.

Αναφορικά με τη λειτουργία του, ο Avgora τρέχει ένα thread για κάθε κόμβο. Αυτά τα threads συγχρονίζονται μόνο όταν υπάρξει ανάγκη. Ο συγχρονισμός αυτός μπορεί να χρειάζεται για παράδειγμα για να βεβαιώσει ότι ο κοινός χρονισμός και η

σειρά των επικοινωνιών διατηρείται κατά τη διάρκεια της προσομοίωσης. Μάλιστα, οι ερευνητές που δημιούργησαν τον Avroga, έδειξαν ότι ο συγχρονισμός μπορεί να πραγματοποιείται αρκετά πιο αραιά από κάθε κύκλο του ρολογιού, κάτι που οδηγεί, όπως είναι φυσικό, σε σημαντικά κέρδη όσον αφορά την απόδοση του προσομοιωτή.

Για την επίτευξη αποδοτικής εκτέλεσης των προγραμμάτων, οι συγκεκριμένοι ερευνητές χρησιμοποίησαν την αρκετά διαδεδομένη ιδέα της ουράς γεγονότων (event queue) [2]. Για να υπάρχει αποδοτικός συγχρονισμός που να διατηρεί σωστή την προσομοίωση και την ακρίβεια του ρολογιού, στα πλαίσια του Avroga διαμορφώθηκαν και λύθηκαν δύο συγκεκριμένα προβλήματα συγχρονισμού, το πρόβλημα Αποστολής – Λήψης και το πρόβλημα Δειγματοληψίας.

Στην περίπτωση της απόδοσης ενός κόμβου, ο διαχειριστής του Avroga μπορεί να προσομοιώσει προγράμματα μικρο – ελεγκτών ως και 25MHz, κάτι που σημαίνει ότι η διαδικασία γίνεται ως και τρεις φορές πιο γρήγορα από τον πραγματικό χρόνο. Όταν οι κόμβοι γίνονται περισσότεροι, ο Avroga μπορεί να διαχειριστεί μέχρι και 1750 κόμβους ανάλογα και με τον υπολογιστή στον οποίο γίνεται η διαδικασία. Για παράδειγμα, σε υπολογιστικό σύστημα με μεγάλες δυνατότητες ο προσομοιωτής έχει διαχειριστεί και δίκτυο αποτελούμενο από 10000 κόμβους.

Σε κάθε περίπτωση, αποδεικνύεται πως ο Avroga επιτυγχάνει κατά κανόνα γραμμική συσχέτιση του χρόνου που διαρκεί η προσομοίωση και του αριθμού των κόμβων. Όταν μάλιστα το δίκτυο αποτελείται από λιγότερους από 25 κόμβους, οι χρόνοι του Avroga είναι μικρότεροι από τους πραγματικούς. Μάλιστα, ο μόνος περιορισμός που εισάγεται αναφορικά με τον αριθμό των κόμβων που μπορεί να προσομοιωθεί, έχει να κάνει με τα όρια του λειτουργικού συστήματος αναφορικά με τις διαδικασίες που μπορούν να εκτελούνται ταυτόχρονα.

Μάλιστα, οι δημιουργοί του Avroga κατέληξαν στο συμπέρασμα ότι ακριβείς προσομοιώσεις δικτύων από 1000άδες κόμβους μπορούν να γίνουν με ικανοποιητική απόδοση, ενώ όταν οι κόμβοι είναι απλά 10άδες, οι χρόνοι είναι της τάξης των πραγματικών. Ο Avroga δίνει τη δυνατότητα στον χρήστη να αναλύσει ένα δίκτυο αισθητήρων με μεγάλη ακρίβεια, τόση που δεν ήταν δυνατόν προηγουμένως. Μάλιστα, ο Avroga συνοδεύεται πλέον και από ένα ακριβές μοντέλο κατανάλωσης ενέργειας, [3] που μεταξύ άλλων δίνει τη δυνατότητα πρόβλεψης του χρόνου ζωής του δικτύου.

Το κλειδί για την πραγματοποίηση μιας αποδοτικής προσομοίωσης ενός μικροελεγκτή είναι η ουρά γεγονότων. Παραδείγματος χάριν, μια από τις σημαντικές παραμέτρους ειδικά σε εφαρμογές που αναφέρονται σε ενεργειακά – αποδοτικούς μικρό – ελεγκτές (όπως στις περιπτώσεις των δικτύων αισθητήρων), αυτοί περνούν σε κατάσταση «ύπνου» για ένα μεγάλο χρονικό διάστημα. Οι κόμβοι / αισθητήρες έχουν ενσωματωμένο ένα mode λειτουργίας «ύπνου», το οποίο θέτει τον μικρό – ελεγκτή σε κατάσταση χαμηλής κατανάλωσης ισχύος κατά την οποία δεν εκτελεί διεργασίες.

Η ουρά γεγονότων που δημιουργεί ο Avroga εκμεταλλεύεται το χαρακτηριστικό αυτό των κόμβων ενός δικτύου αισθητήρων, με αποτέλεσμα να κερδίζει σημαντικά όσον αφορά την απόδοσή του. Πιο συγκεκριμένα, φαίνεται πως όταν το πρόγραμμα θέσει τον μικρό- ελεγκτή σε κατάσταση ύπνου, τότε μόνο ένα χρονικό «event» που προκαλεί διακοπή μπορεί να «αφύπνισει» τον μικρό – ελεγκτή. Ένα τέτοιο γεγονός μπορεί να δημιουργηθεί στο chip ή στο περιβάλλον. Και οι δύο αυτές περιπτώσεις υλοποιούνται με την εισαγωγή τέτοιων γεγονότων στην ουρά γεγονότων των κόμβων τα οποία εκτελούνται σε συγκεκριμένες χρονικές στιγμές [4].

Συμπερασματικά, τα γεγονότα πρέπει να έχουν καταχωρηθεί στην ουρά εκ των προτέρων, και μόνο το γεγονός που είναι στην κεφαλή της ουράς έχει τη δυνατότητα να επηρεάσει την προσομοίωση όταν ο μικρό – ελεγκτής είναι σε κατάσταση ύπνου. Ο Avroga λοιπόν το μόνο που χρειάζεται να κάνει, είναι να διαχειρίζεται τα γεγονότα που υπάρχουν στην ουρά μέχρις ότου κάποιο από αυτά να προκαλέσει μια διακοπή στο υλικό, κάτι που οδηγεί τον μικρό – ελεγκτή και πάλι σε αφύπνιση. Μεγάλες ποσότητες χρόνου προσομοίωσης μπορούν να περάσουν με αυτόν τον τρόπο, οι οποίες απαιτούν ιδιαίτερα μικρό φόρτο εργασίας προσομοίωσης. Ένα ακραίο παράδειγμα είναι το παράδειγμα του προγράμματος Blink του TinyOS, το οποίο αναβοσβήνει ένα LED κάθε ένα δευτερόλεπτο. Στο πρόγραμμα αυτό, ο μικρό ελεγκτής είναι σε κατάσταση ύπνου το 99,96% του χρόνου. Χωρίς την βελτιστοποίηση της κατάστασης ύπνου, ο προσομοιωτής τρέχει στα 31MHz στον Avroga. Με την βελτιστοποίηση όμως, η προσομοίωση γίνεται 500 φορές πιο γρήγορα.

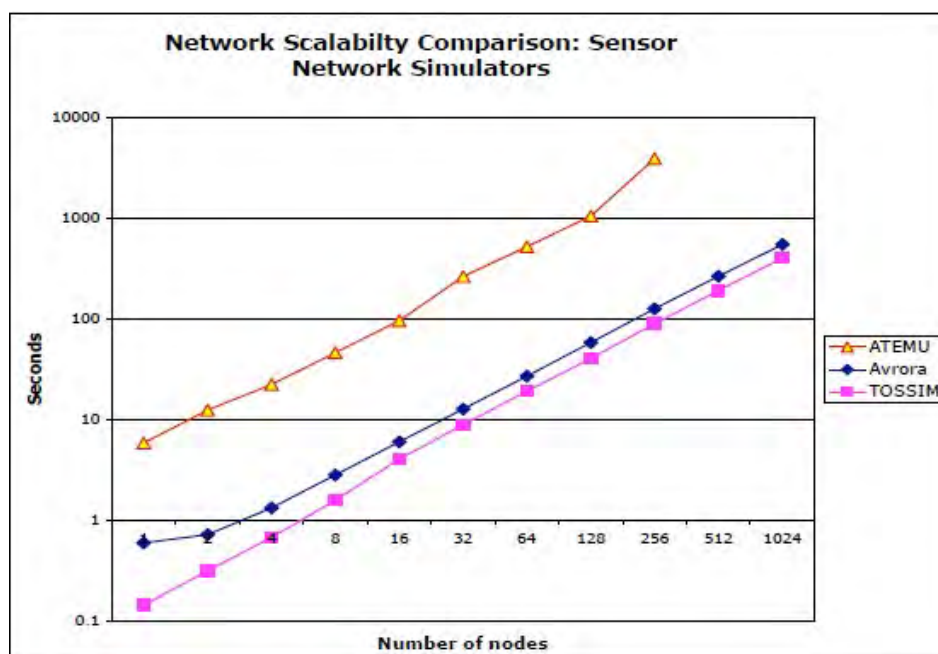
Ο ψευδοκώδικας που ακολουθεί παριστάνει το βασικό loop προσομοίωσης.

```
infinite loop {  
  if (sleeping)  
    eventqueue.advance (head.delta);  
  else {  
    execute instruction;  
    eventqueue.advance(cyclesconsumed);  
  }  
}  
  
eventqueue.advance(cycles) {  
  if (head) {  
    head.delta -= cycles;  
    if (head.delta == 0) {  
      head.fire();  
      head = head.next;  
    }  
  }  
}
```

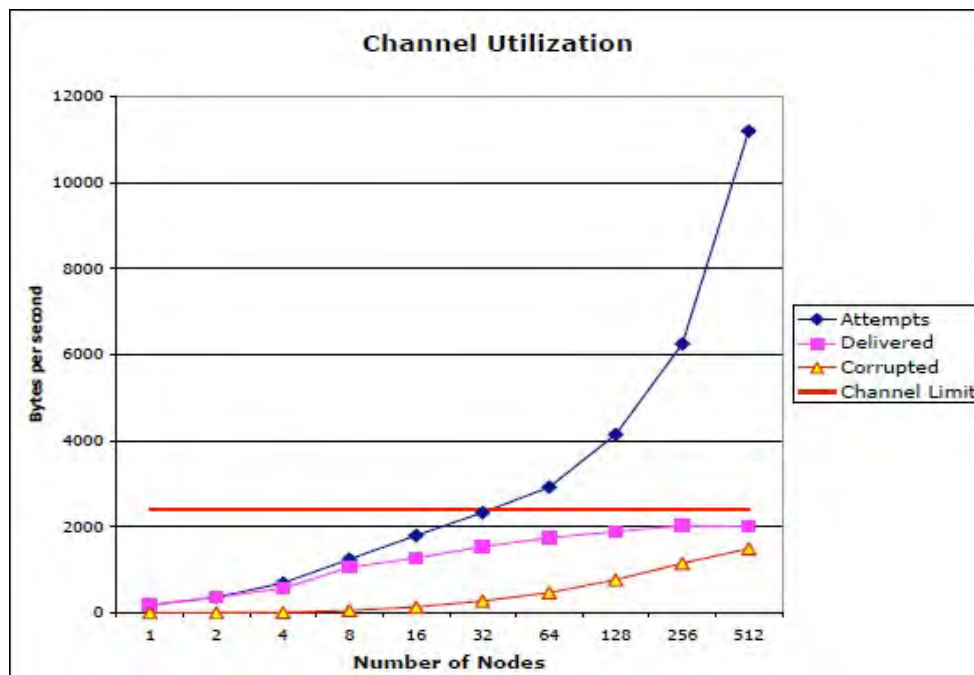
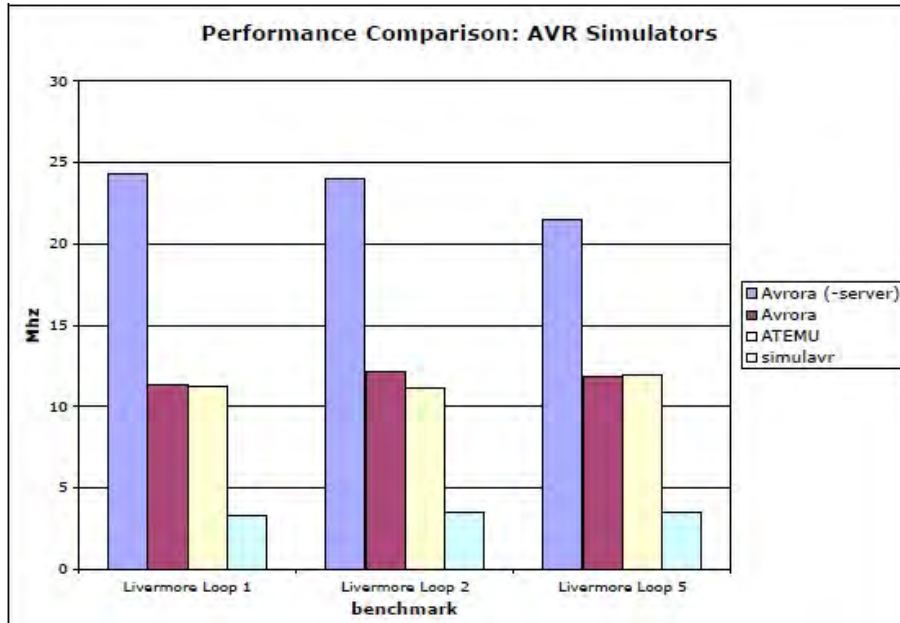
3.3 Ενδεικτικά αποτελέσματα συγκρίσεων

Στα επόμενα τρία γραφήματα, παρουσιάζονται ενδεικτικά κάποιες συγκρίσεις μεταξύ των δύο αυτών προσομοιωτών αλλά και άλλων, όπως για παράδειγμα του ATEMU.

Στο γράφημα που ακολουθεί, φαίνεται ότι αναφορικά με την ανεκτικότητα των προσομοιωτών στο μέγεθος του δικτύου, ο Avroga αποδίδει σχεδόν με παρόμοιο τρόπο με τον TOSSIM.



Το επόμενο γράφημα παριστάνει την απόδοση των προσομοιωτών σε MHz, ενώ ενδιαφέρον παρουσιάζει και η δυνατότητα του Avroga σχετικά με τη χρήση του καναλιού (utilization) που φαίνεται στο 3ο γράφημα, με τη χρήση εφαρμογών με ποικίλες συμπεριφορές.



Ανάλυση της κατανάλωσης ενέργειας

4.1 PowerTossim

Το αρχείο που είναι υπεύθυνο για τον καθορισμό του ενεργειακού μοντέλου βάσει του προσομοιωτή PowerTOSSIM βρίσκεται στο μονοπάτι:

Opt/tinyos-1.x/tools/scripts/PowerTOSSIM

Το βασικό αρχείο που καθορίζει το ενεργειακό μοντέλο για τους αισθητήρες mica2 είναι το:

Mica2_energy_model.txt

Μέσα στο αρχείο αυτό, καθορίζονται οι βασικές παράμετροι από τις οποίες ρυθμίζονται τα ποσά ενέργειας που καταναλώνει ο κάθε κόμβος, ανάλογα με την λειτουργία που επιτελεί.

Το σύνολο του αρχείου φαίνεται παρακάτω.

```
# Energy model for the mica2

# see http://www.eecs.harvard.edu/~shnayder/ptossim/ for details
# Lines starting with '#' are comments

# Format: KEY\s+VALUE
# Unless otherwise specified, VALUE is a current in mA
# Value is always assumed to be a number

VOLTAGE    3.0
# CPU numbers:
# Idle mode is what the CPU is in when not actively executing, but
# no special action is taken. The per-active-cycle numbers are added
# on top of that
```



```
# Added .93mA to idle and active cpu power to account for the
# oscillator. Need to make it separate later...
```

```
#CPU_ACTIVE          8.0
CPU_ACTIVE           8.93
#CPU_IDLE            3.2
CPU_IDLE             4.13
CPU_ADC_NOISE_REDUCTION 1.0
CPU_POWER_DOWN       0.103
CPU_POWER_SAVE       0.110
CPU_STANDBY          0.216
CPU_EXTENDED_STANDBY 0.223
```

```
# The startup mode for the cpu
```

```
CPU_INIT            3.2
```

```
CPU_FREQ 4000000
```

```
# It's actually 7382800, but the simulator is hardcoded for 4Mhz, and
```

```
# the fixes for that aren't merged yet
```

```
#CPU_FREQ 7382800
```

```
# These are actual measured numbers
```

```
RADIO_RX 7.03
```

```
RADIO_TX_00 3.72
```

```
RADIO_TX_01 5.21
```

```
RADIO_TX_03 5.37
```

```
RADIO_TX_06 6.47
```

```
RADIO_TX_09 7.05
```

```
RADIO_TX_0F 8.47
```

```
RADIO_TX_60 11.57
```

```
RADIO_TX_80 13.77
```

```
RADIO_TX_C0 17.37
```

```
RADIO_TX_FF 21.48
```

```
RADIO_OFF 0
```

```
# These are interpolated from above
```

```

RADIO_TX_02  5.3
RADIO_TX_04  5.75
RADIO_TX_05  6.1
RADIO_TX_07  6.67
RADIO_TX_08  6.87
RADIO_TX_0A  7.33
RADIO_TX_0B  7.6
RADIO_TX_0C  7.93
RADIO_TX_0E  8.15
RADIO_TX_40  9.5
RADIO_TX_50  10.5
RADIO_TX_70  12.6
RADIO_TX_90  15.0
RADIO_TX_E0  19.5
# The default power mode (0x0F, but python doesn't like hex numbers in floats)
RADIO_DEFAULT_POWER  15.00
# LED
LED          2.2
#default to off
LED_INIT    0
# EEPROM
EEPROM_READ  6.24
EEPROM_WRITE 18.40

#ADC
# we thought it took power, so it's tracked, but it actually doesn't
ADC  0.0

# If the sensor board is plugged in, it draws this much current at all times
SENSOR_BOARD  0.69
# Sensors - don't actually use power beyond the general sensor board power
SENSOR_PHOTO  0
SENSOR_TEMP   0

```

Στο αρχείο αυτό, όλες οι τιμές αναφέρονται σε mA, εκτός αν αναφέρεται διαφορετικά.

4.1.1 Επεξεργαστής

Έτσι, αναφορικά με την CPU, φαίνεται ότι όταν είναι σε «ενεργή» κατάσταση (active mode) και λειτουργεί κανονικά, χρησιμοποιεί 8.93 mA. Από αυτά, τα 0.93 αναφέρονται σε λειτουργίες του ταλαντωτή (oscillator).

Όταν ο επεξεργαστής είναι σε κατάσταση «ηρεμίας» και δεν επιτελεί κάποια εργασία (idle mode) τότε η κατανάλωση πέφτει στα 4.13 mA.

Η λειτουργία μείωσης θορύβου «κοστίζει» στον επεξεργαστή επιπλέον 1 mA, ενώ η διαδικασία κατά την οποία «κλείνει» καταναλώνει 0.103 mA.

Ακόμη, η αποθήκευση πριν από αυτή τη διαδικασία καταναλώνει άλλα 0.11 mA, η μετάβαση σε κατάσταση «standby» αντιστοιχεί σε 0.216 mA και το «παρατεταμένο» standby σε 0.223 mA.

Τέλος, η εκκίνηση του επεξεργαστή απαιτεί ρεύμα 3.2 mA.

4.1.2 Πομποδέκτης

Όπως με τον επεξεργαστή, έτσι και με τον πομποδέκτη το αρχείο καθορίζει τιμές ρεύματος ανάλογα με τις λειτουργίες που αυτός επιτελεί.

Ο δέκτης (receiver) καταναλώνει σταθερή τιμή ρεύματος (7.03 mA) ενώ ο πομπός μπορεί να καταναλώνει πληθώρα διαφορετικών τιμών, ανάλογα με το είδος λειτουργίας και τύπο πακέτων που μεταδίδει. Η τυπική τιμή είναι τα 15.00 mA

4.1.3 Επιπλέον καταναλώσεις

Στις επιπλέον καταναλώσεις περιλαμβάνονται τα φωτάκια του sensor (led), η κατανάλωση από την μνήμη EEPROM, και από την πλατφόρμα του αισθητήρα.

Το φως (led) καταναλώνει 2.2 mA όταν ανάβει, και καθόλου ρεύμα όταν είναι σβηστό.

Η ανάγνωση από τη μνήμη EEPROM κοστίζει 6.24 mA στον αισθητήρα, ενώ η καταγραφή στη μνήμη στοιχίζει 18,4 mA.

Τέλος, η πλατφόρμα του αισθητήρα καταναλώνει 0.69 mA.

4.2 Avrora

Ο προσομοιωτής Avrora μετά την εγκατάστασή του, φαίνεται πως λειτουργεί με βάση προ – εγκατεστημένα modules από πομποδέκτες (radios) που υπάρχουν σε πλατφόρμες των mica και άλλων κόμβων – αισθητήρων.

Επομένως, η άντληση δεδομένων για τον υπολογισμό των ενεργειακών καταναλώσεων γίνεται μέσω αυτών των αρχείων. Έτσι, με την εισαγωγή των κατάλληλων παραμέτρων για την επιθυμητή πλατφόρμα (πχ mica2), το επιθυμητό radio κλπ γίνεται και η απαραίτητη προσομοίωση.

Τα modules αυτά είναι σε μορφή class αρχείων της java.

Ο προσομοιωτής αυτός δίνει την δυνατότητα «παρέμβασης» μόνο στο βαθμό δημιουργίας εξ αρχής επιπλέον modules και τοποθέτησής τους στα κατάλληλα folders.

Τα modules που καθορίζουν την ενεργειακή κατανάλωση είναι στο μονοπάτι:

Avrora/sim/energy

Το πρόβλημα της ύπαρξης μόνο class αρχείων ξεπεράστηκε με τη χρήση ενός de – compiler της java, ο οποίος δημιουργεί από class αρχεία τα αντίστοιχα java αρχεία.

Με τη χρήση ενός τέτοιου προγράμματος, έγινε δυνατή η επεξεργασία του αρχείου **energy.class**, το οποίο είναι το βασικό αρχείο που καθορίζει τις παραμέτρους της ενέργειας, και το περιεχόμενό του είναι το ακόλουθο:

```
// Decompiled by DJ v3.11.11.95 Copyright 2009 Atanas Neshkov Date: 6/6/2010 5:13:09
PM
// Home Page: http://members.fortunecity.com/neshkov/dj.html
http://www.neshkov.com/dj.html - Check often for new version!
// Decompiler options: packimports(3)
// Source File Name: Energy.java
package avrora.sim.energy;
import avrora.sim.FiniteStateMachine;
import avrora.sim.clock.Clock;
// Referenced classes of package avrora.sim.energy:
// EnergyControl
public class Energy
    implements avrora.sim.FiniteStateMachine.Probe
{
    public Energy(String s, double ad[], FiniteStateMachine finitestatemachine)
    {
        deviceName = s;
        clock = finitestatemachine.getClock();
        ampere = ad;
        stateMachine = finitestatemachine;
        currentMode = finitestatemachine.getStartState();
        freq = (int)clock.getHZ();
        cycleTime = 1.0D / (double)freq;
        instance = EnergyControl.getCurrentInstance();
        EnergyControl.addConsumer(this);
    }
    public void fireBeforeTransition(int i, int j)
    {
    }
    public void fireAfterTransition(int i, int j)
```

```

{
    if(j != currentMode)
    {
        cycles[currentMode] += clock.getCount() - lastChange;
        oldMode = currentMode;
        currentMode = j;
        lastChange = clock.getCount();
        instance.stateChange(this);
    }
}
public double getTotalConsumedEnergy()
{
    double d = 0.0D;
    for(int i = 0; i < ampere.length; i++)
        d += getConsumedEnergy(i);
    return d;
}
public double getConsumedEnergy(int i)
{
    return 3D * (double)getCycles(i) * ampere[i] * cycleTime;
}
public int getModeNumber()
{
    return ampere.length;
}
public int getCurrentMode()
{
    return currentMode;
}
public String getModeName(int i)
{
    return stateMachine.getStateName(i);
}
public double getModeAmpere(int i)
{
    return ampere[i];
}
public long getCycles(int i)
{

```

```

    if(cycles == null)
        return 0L;
    long l = cycles[i];
    if(i == currentMode)
        l += clock.getCount() - lastChange;
    return l;
}
public String getName()
{
    return deviceName;
}
public int getOldMode()
{
    return oldMode;
}
public double getCurrentAmpere()
{
    return ampere[currentMode];
}
public double getOldAmpere()
{
    return ampere[oldMode];
}
public void activate()
{
    stateMachine.insertProbe(this);
    cycles = new long[ampere.length];
    for(int i = 0; i < cycles.length; i++)
        cycles[i] = 0L;
}
private String deviceName;
private double ampere[];
private long cycles[];
private int currentMode;
private int oldMode;
private long lastChange;
private static final double voltage = 3D;
private int freq;
private double cycleTime;

```

```

private FiniteStateMachine stateMachine;
private Clock clock;
private EnergyControl.Instance instance;
}

```

Ουσιαστικά, όπως φαίνεται από το παρακάτω αρχείο, ο βασικός έλεγχος σχετικά με την καταναλισκόμενη ενέργεια γίνεται με τις

```

public double getTotalConsumedEnergy()
{
    double d = 0.0D;
    for(int i = 0; i < ampere.length; i++)
        d += getConsumedEnergy(i);
    return d;
}
public double getConsumedEnergy(int i)
{
    return 3D * (double)getCycles(i) * ampere[i] * cycleTime;
}

```

Η βασική παράμετρος είναι η τιμή του ρεύματος (ampere) που πολλαπλασιάζεται με τον αριθμό των cycles καθώς και τη διάρκειά τους, και το 3D που είναι η τάση.

Από εκεί και πέρα, οι τιμές των παραμέτρων αυτών καθορίζονται ανάλογα με την πλατφόρμα και τον πομποδέκτη που χρησιμοποιείται.

Παραδείγματος χάριν, για τα MICA το αντίστοιχο αρχείο είναι το παρακάτω:

```

package avrora.sim.platform;
import avrora.core.Program;
import avrora.sim.InterpreterFactory;
import avrora.sim.Simulator;
import avrora.sim.clock.ClockDomain;
import avrora.sim.mcu.*;
import avrora.sim.platform.sensors.LightSensor;
import avrora.sim.platform.sensors.SensorBoard;
import avrora.sim.radio.CC1000Radio;
import avrora.sim.radio.Radio;

```



```

// Referenced classes of package avrora.sim.platform:
// Platform, LED, ExternalFlash, PlatformFactory
public class Mica2 extends Platform
{
    public static class Factory
        implements PlatformFactory
    {
        public Platform newPlatform(int i, InterpreterFactory interpreterfactory, Program
program)
        {
            ClockDomain clockdomain = new ClockDomain(0x708000L);
            clockdomain.newClock("external", 32768L);
            return new Mica2(new ATmega128(i, clockdomain, interpreterfactory, program));
        }
        public Factory()
        {
        }
    }
    private Mica2(Microcontroller microcontroller)
    {
        super(microcontroller);
        sim = microcontroller.getSimulator();
        addDevices();
    }
    protected void addDevices()
    {
        LED led = new LED(sim, 11, "Yellow");
        LED led1 = new LED(sim, 2, "Green");
        LED led2 = new LED(sim, 1, "Red");
        led.enablePrinting();
        led1.enablePrinting();
        led2.enablePrinting();
        mcu.getPin("PA0").connect(led);
        mcu.getPin("PA1").connect(led1);
        mcu.getPin("PA2").connect(led2);
        radio = new CC1000Radio(mcu, 0xe10000L);
        addDevice("radio", radio);
        sensorboard = new SensorBoard(sim);
        externalFlash = new ExternalFlash(mcu);
    }
}

```

```
AtmelMicrocontroller atmelmicrocontroller = (AtmelMicrocontroller)mcu;
lightSensor = new LightSensor(atmelmicrocontroller, 1, "PC2", "PE5");
addDevice("light-sensor", lightSensor);
}
```

```
protected static final int MAIN_HZ = 0x708000;
protected final Simulator sim;
protected Radio radio;
protected SensorBoard sensorboard;
protected ExternalFlash externalFlash;
protected LightSensor lightSensor;
}
```

Από το αρχείο αυτό, φαίνεται ότι για την συγκεκριμένη πλατφόρμα χρησιμοποιείται για παράδειγμα ο CC1000Radio, οι τιμές για τον οποίο καθορίζονται στο αντίστοιχο αρχείο.

Σε κάθε περίπτωση όμως, πρέπει να γίνει σαφές ότι δεδομένου πως ο συγκεκριμένος κώδικας δεν είναι «ανοιχτός», δηλαδή η εταιρία δίνει μόνο τα class αρχεία, θεωρείται αδύνατο να παραχθεί αρχείο με εναλλακτικό ενεργειακό μοντέλο που να είναι δυνατόν να ελεγχθεί μέσω του TinyOS.

4.3 Ενδεικτικά τρεξίματα και αποθήκευση αποτελεσμάτων σε trace files

Στο επόμενο βήμα της μελέτης, έγιναν κάποια ενδεικτικά «τρεξίματα» εφαρμογών με τη βοήθεια των δύο προσομοιωτών.

Και οι δύο δίνουν στον χρήστη τη δυνατότητα να αποθηκεύει τις ενέργειες που εκτελούνται με βάση των έτοιμων εφαρμογών και των διαφορετικών αισθητήρων, δηλαδή του συνόλου των κόμβων που εκτελούνται, σε ειδικά αρχεία, τα trace files.

Σε αυτό το πλαίσιο, μελετήθηκε η λειτουργία δύο έτοιμων εφαρμογών των MICA2, τόσο με τον έναν όσο και με τον άλλο προσομοιωτή.

Κάποια ενδεικτικά κομμάτια των trace files που δημιουργήθηκαν είναι διαθέσιμα στο παράρτημα.

Πάντως είναι σαφές πως το είδος των δεδομένων και η μορφή με την οποία παρουσιάζονται αυτά, διαφέρουν από προσομοιωτή σε προσομοιωτή.



MICA2

4.4 Δημιουργία Εναλλακτικών Αρχείων

Έχοντας μελετήσει τα αρχεία που είναι υπεύθυνα για τον καθορισμό των ενεργειακών παραμέτρων για τους δύο προσομοιωτές, είναι εύκολο να δημιουργηθούν νέα αρχεία με σκοπό την μικρή παραλλαγή των βασικών αυτών παραμέτρων.

Σχετικά με τον PowerTossim, δημιουργήθηκε το αρχείο “alternative_energy_model.txt”, το οποίο φαίνεται στο παράρτημα. Σαν βασικές διαφορές σε σχέση με το ενεργειακό μοντέλο του MICA2, θεωρήθηκε το ότι καταρχήν δεν υπάρχουν LED και άρα οι αντίστοιχες καταναλώσεις. Επιπλέον, υπάρχει κατανάλωση ενέργειας για τον ADC.

Οι καταναλώσεις του ταλαντωτή (oscillator) έχουν ενσωματωθεί στις καταναλώσεις της CPU για τις διάφορες λειτουργίες της, ενώ κάποιες καταναλώσεις έχουν αυξηθεί ή μειωθεί. Προφανώς οι τιμές μπορούν να αλλάξουν ανάλογα με το ενεργειακό μοντέλο που θέλουμε να υιοθετήσουμε.

Σε σχέση με τον Avroga, για να χρησιμοποιηθεί εναλλακτικό ενεργειακό μοντέλο πρέπει να δημιουργηθεί μια νέα κλάση εναλλακτικής πλατφόρμας.

Ονομάσαμε αυτή την κλάση «alternative_avroga». Η κλάση αυτή έχει στηριχθεί στην κλάση που αντιστοιχεί στην πλατφόρμα MICA2, και ο αντίστοιχος κώδικας φαίνεται στο παράρτημα. Χρησιμοποιεί τον CC1000Radio για την επικοινωνία, όπως και τους αισθητήρες φωτός. Η βασική λειτουργία της έχει να κάνει με τον έλεγχο των LED.

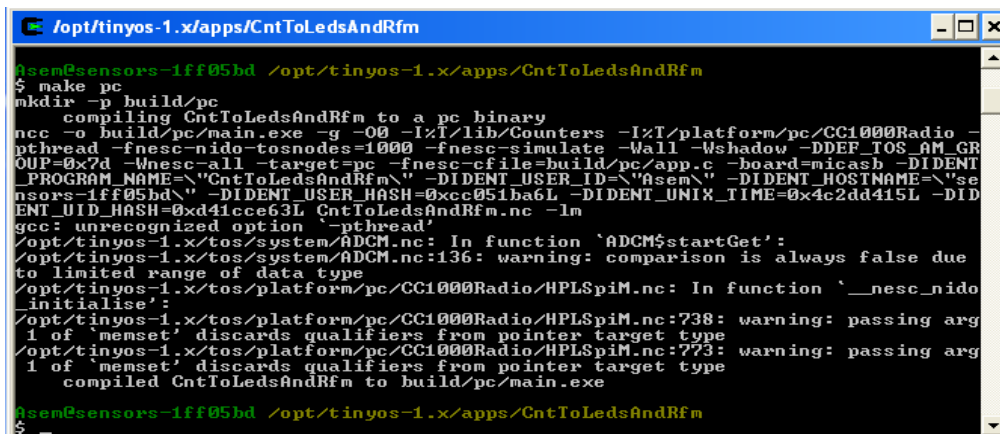
Τονίζεται για άλλη μια φορά, πως ο Avroga σου δίνει την δυνατότητα επέκτασης μέσω της δημιουργίας τέτοιων κλάσεων για εναλλακτικές πλατφόρμες, αλλά το γεγονός ότι ο κώδικας στην αντίστοιχη ιστοσελίδα της εταιρίας δεν είναι ανοιχτός (διαθέσιμος μόνο σε *.class αρχεία), δεν αφήνει πολλά περιθώρια καθορισμού ενεργειακών παραμέτρων αναφορικά με τιμές ρεύματος για τις διάφορες λειτουργίες.

ΕΚΤΕΛΕΣΗ ΤΟΥ CntToLedsAndRfm ΜΕ ΤΟΥΣ POWERTOSSIM & AVRORA

5.1 PowerTossim

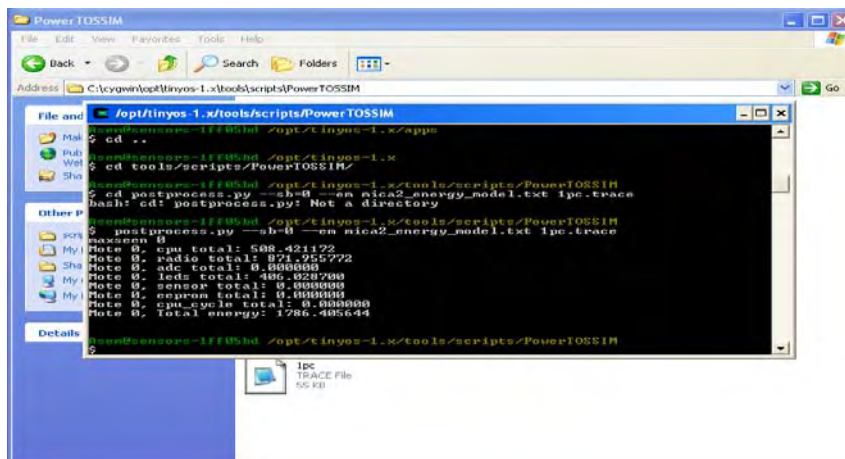
Αρχικά εκτελώντας στην έτοιμη εφαρμογή CntToLedsAndRfm που διαθέτει ο TinyOS με τον PowerTossim και τον τύπο αισθητήρα της πλατφόρμας (PC), και ξεκινώντας με ένα mote ώστε να παρατηρήσουμε την κατανάλωση ενέργειας που απαιτείται στην αρχική φάση.

Στο μονοπάτι που φαίνεται παρακάτω κάνοντας compile την πλατφόρμα (pc).



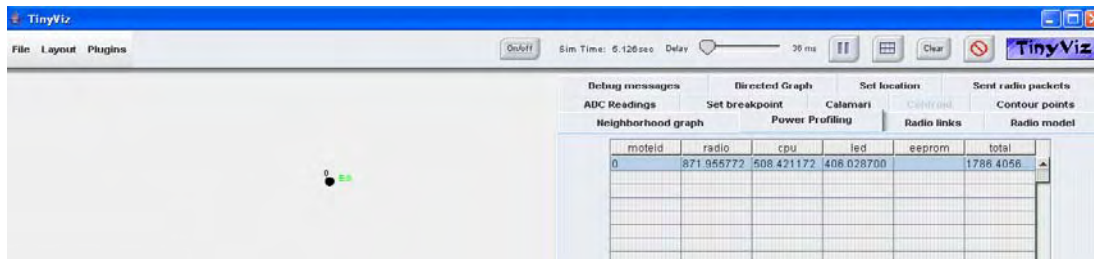
```
Assem@sensors-1ff05bd /opt/tinyos-1.x/apps/CntToLedsAndRfm
$ make pc
mkdir -p build/pc
compiling CntToLedsAndRfm to a pc binary
ncc -o build/pc/main.exe -g -O0 -I/I/lib/Counters -I/I/platform/pc/CC1000Radio -pthread -fnesc-nido-tosnodes=1000 -fnesc-simulate -Wall -Wshadow -DDEF_TOS_AM_GRP=0x7d -Wnesc-all -target=pc -fnesc-cfile=build/pc/app.c -board=micasb -DIDENT_PROGRAM_NAME="CntToLedsAndRfm" -DIDENT_USER_ID="Assem" -DIDENT_HOSTNAME="sensors-1ff05bd" -DIDENT_USER_HASH=0xcc051ba6L -DIDENT_UNIX_TIME=0x4c2dd415L -DIDENT_UID_HASH=0xd41cce63L CntToLedsAndRfm.nc -lm
gcc: unrecognized option '-pthread'
/opt/tinyos-1.x/tos/system/ADCM.nc: In function 'ADCM$startGet':
/opt/tinyos-1.x/tos/system/ADCM.nc:136: warning: comparison is always false due to limited range of data type
/opt/tinyos-1.x/tos/platform/pc/CC1000Radio/HPLSpiM.nc: In function '__nesc_nido_initialise':
/opt/tinyos-1.x/tos/platform/pc/CC1000Radio/HPLSpiM.nc:738: warning: passing arg 1 of 'memset' discards qualifiers from pointer target type
/opt/tinyos-1.x/tos/platform/pc/CC1000Radio/HPLSpiM.nc:773: warning: passing arg 1 of 'memset' discards qualifiers from pointer target type
compiled CntToLedsAndRfm to build/pc/main.exe
Assem@sensors-1ff05bd /opt/tinyos-1.x/apps/CntToLedsAndRfm
$
```

Και στην συνέχεια αποθηκεύοντας τα αποτελέσματα στο αρχείο 1pc.trace.

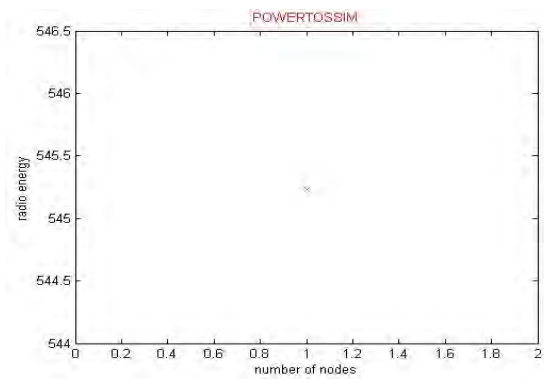
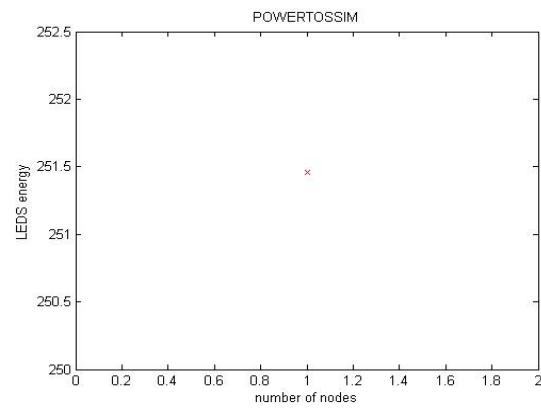
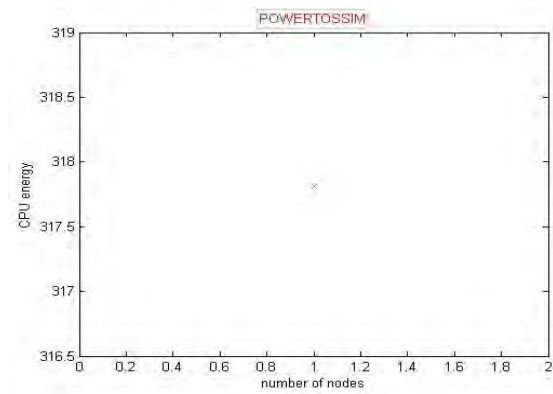


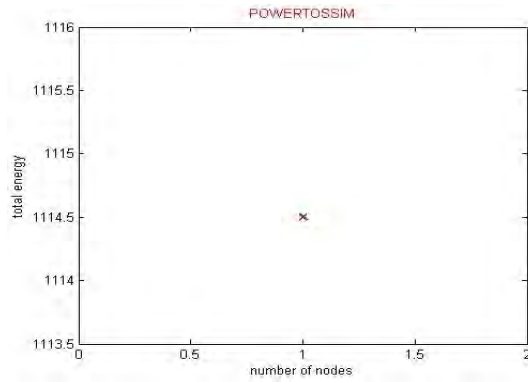
```
PowerTOSSIM
File Edit View Favorites Tools Help
Back Search Folders
Address C:\cywin\opt\tinyos-1.x\tools\scripts\PowerTOSSIM
File and /opt/tinyos-1.x/tools/scripts/PowerTOSSIM
$ cd ..
Assem@sensors-1ff05bd /opt/tinyos-1.x
$ cd tools/scripts/PowerTOSSIM/
Assem@sensors-1ff05bd /opt/tinyos-1.x/tools/scripts/PowerTOSSIM
$ postprocess.py --h-0 --m nica2_energy_model.txt 1pc.trace
bash: cd: postprocess.py: Not a directory
Assem@sensors-1ff05bd /opt/tinyos-1.x/tools/scripts/PowerTOSSIM
$ postprocess.py --h-0 --m nica2_energy_model.txt 1pc.trace
Assem@
Note 0: cpu total: 508.421172
Note 0: radio total: 874.257772
Note 0: adc total: 0.000000
Note 0: leds total: 406.020000
Note 0: sensor total: 0.000000
Note 0: eeprom total: 0.000000
Note 0: cpu_cycles total: 0.000000
Note 0: Total energy: 1786.405644
Assem@sensors-1ff05bd /opt/tinyos-1.x/tools/scripts/PowerTOSSIM
$
1pc TRACE File
95 KB
```

Οπτικοποιώντας τα αποτελέσματα με την χρήση του TinyViz:

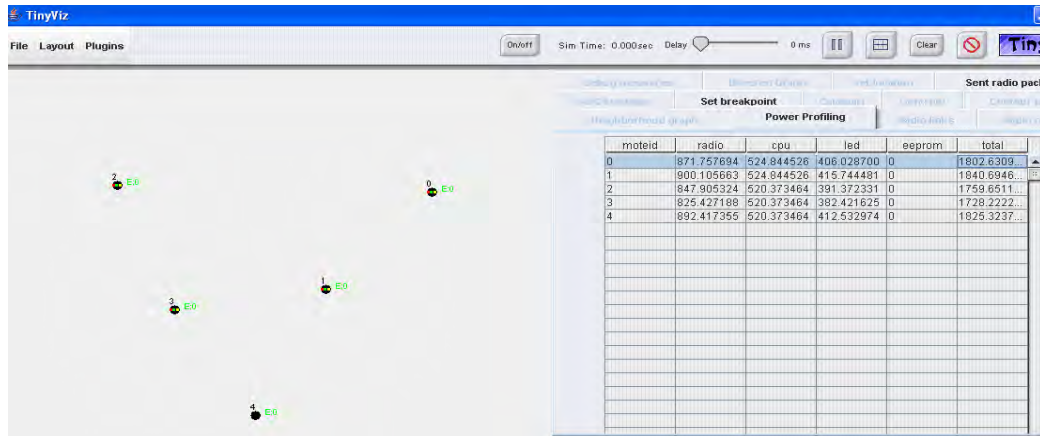


Και με την χρήση του προγράμματος Matlab:

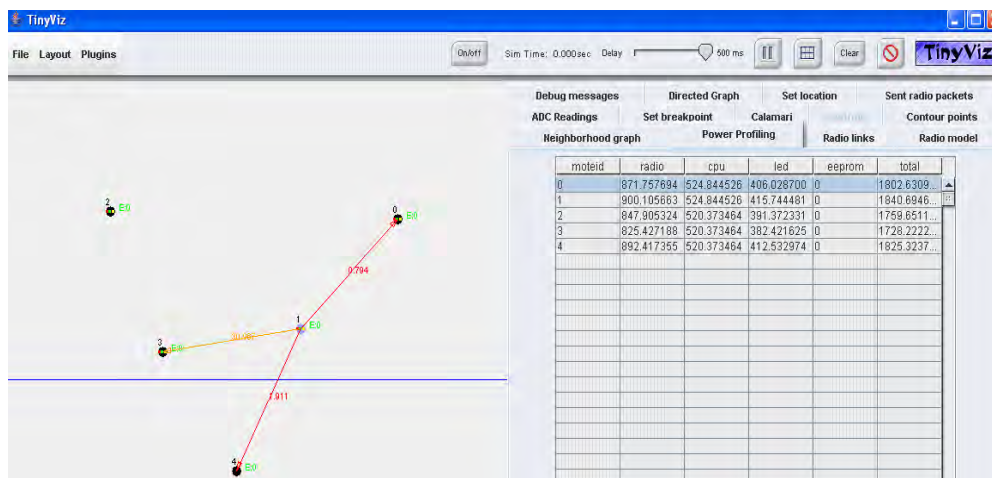


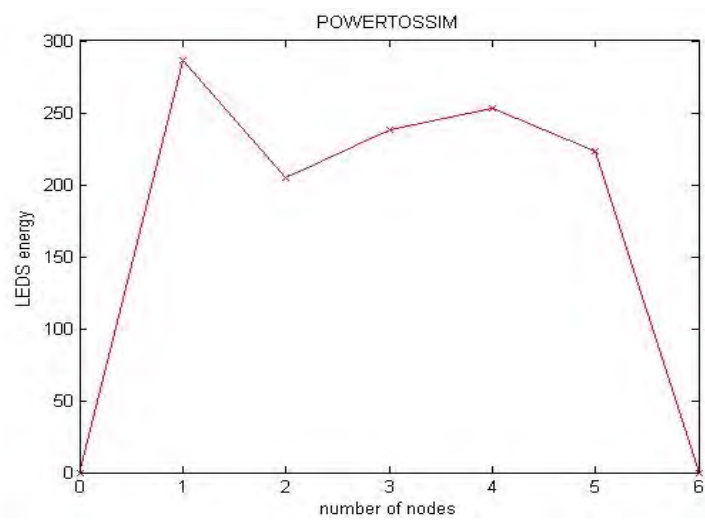
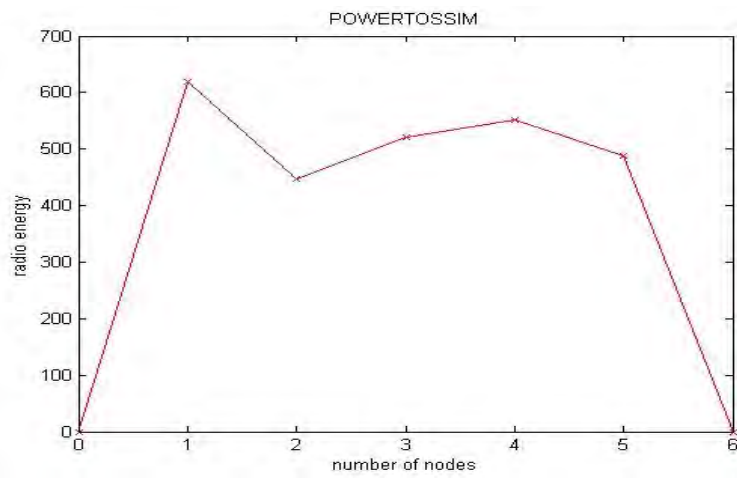
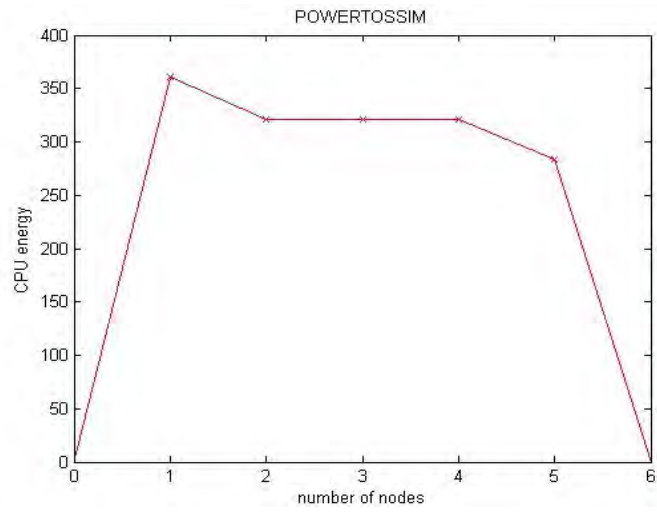


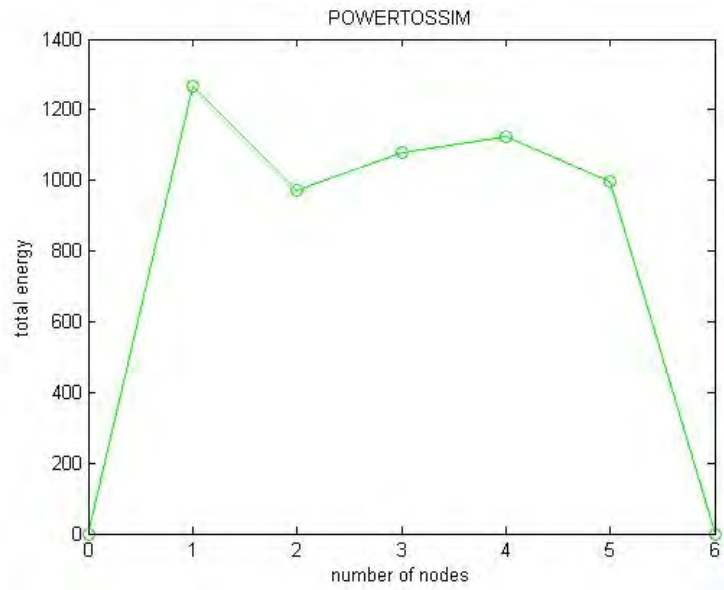
Και για 5 motes επαναλαμβάνεται η διαδικασία και αποθηκεύεται η ενέργεια στο **5pc.trace** παρατηρώντας τα αποτελέσματα της κατανάλωσης της ενέργειας στο συγκεκριμένο μοντέλο:



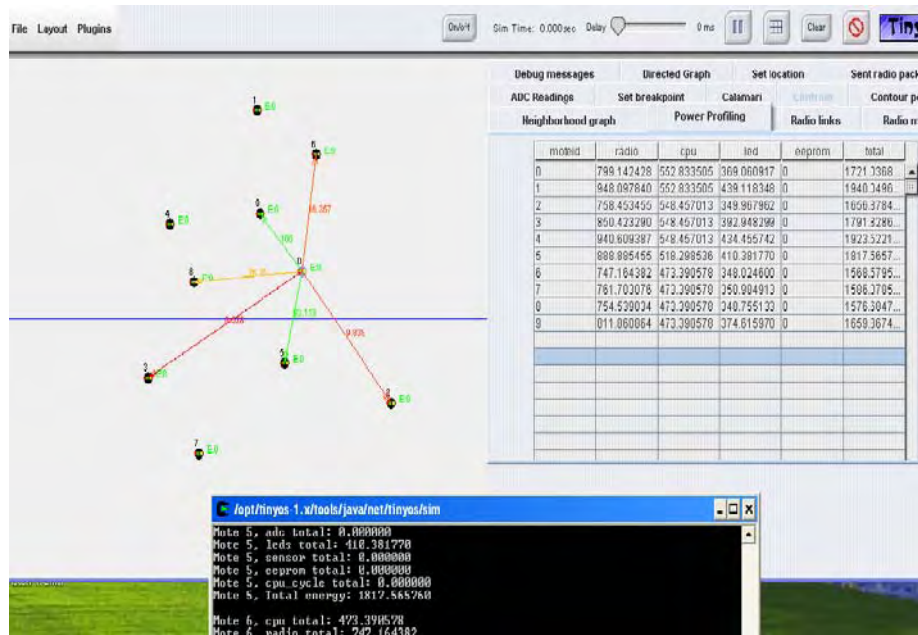
Και φυσικά παρατηρείται ότι οι κόμβοι αρχίζουν και ανταλλάσσουν μηνύματα μεταξύ τους .

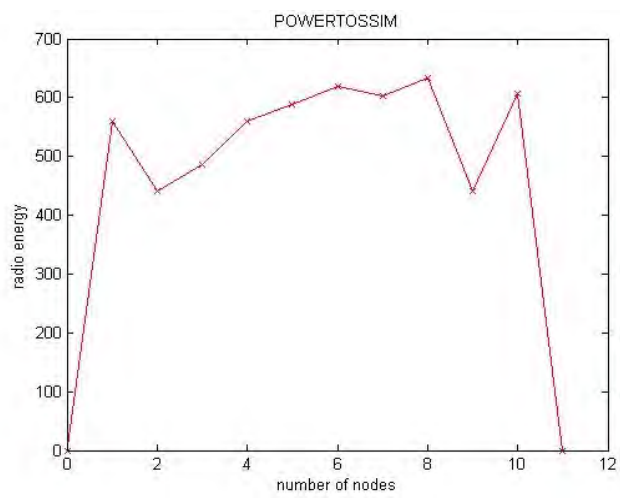
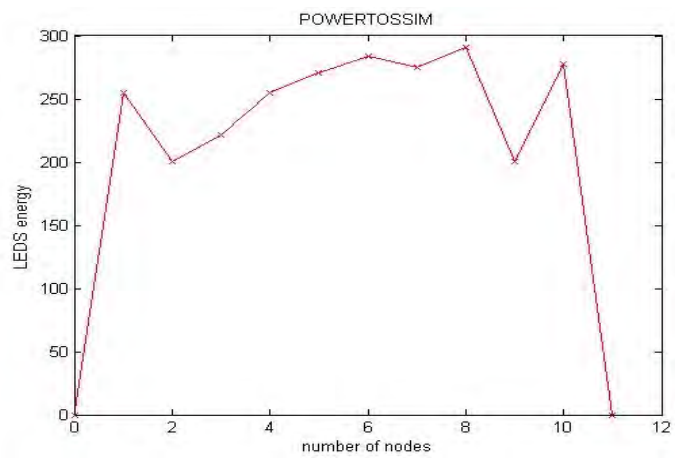
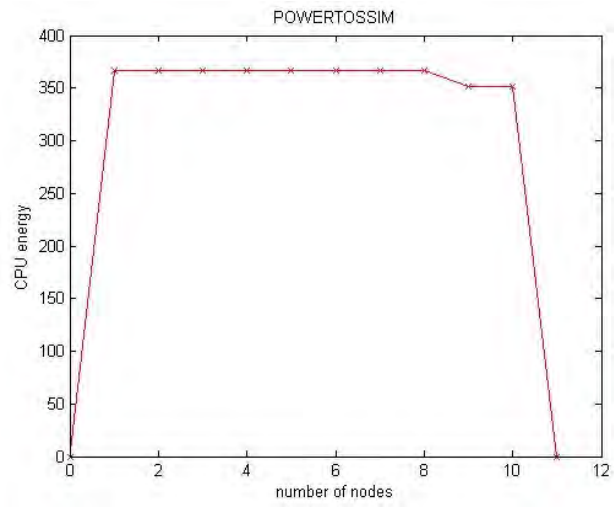


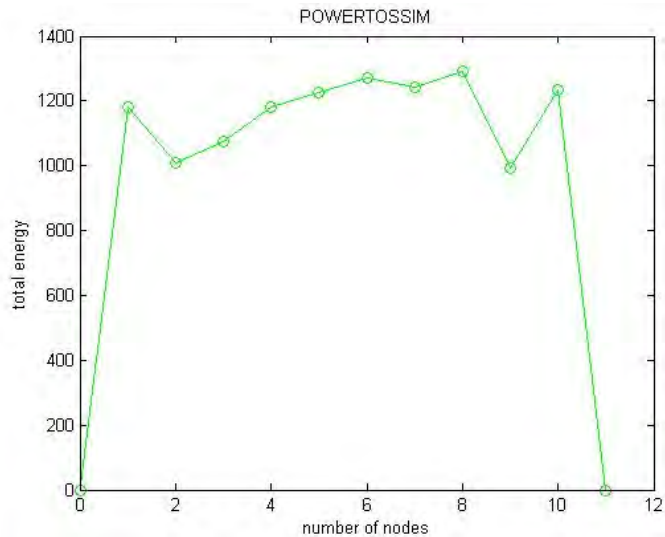




Και για 10 κόμβους από το αρχείο **10pc.trace** :







Σύμφωνα με τα παραπάνω τρεξίματα συμπεραίνουμε ότι όσο αυξάνεται ο αριθμός των κόμβων τόσο μειώνεται η καταναλισκόμενη συνολική ενέργεια, τόσο για τη CPU όσο για τα LED και RADIO. Επίσης μπορεί αυτό να μην είναι τόσο προφανές εάν στους αρχικούς κόμβους μεταβάλλεται η ενέργεια με μη προκαθορισμένο τρόπο (αύξηση/μείωση), όπως φαίνεται στα γραφήματα παραπάνω, αλλά τελικά όσο αυξάνονται οι κόμβοι γίνεται όλο και πιο ξεκάθαρο.

5.2 Avrora

Μετά την εγκατάσταση του Avrora, όπως και στο PowerTossim, πηγαίνοντας στο αρχείο CntToLedaAndRfm κάνουμε compile την πλατφόρμα mica2 για παράδειγμα, και αφού δημιουργηθεί το αντίστοιχο αρχείο θα πρέπει να γίνουν κάποιες μεταφράσεις. Η διαδικασία εκτίμησης της καταναλισκόμενης ενέργειας στο συγκεκριμένο μοντέλο (Avrora) γίνεται μόνο σε αρχεία τύπου File.od .

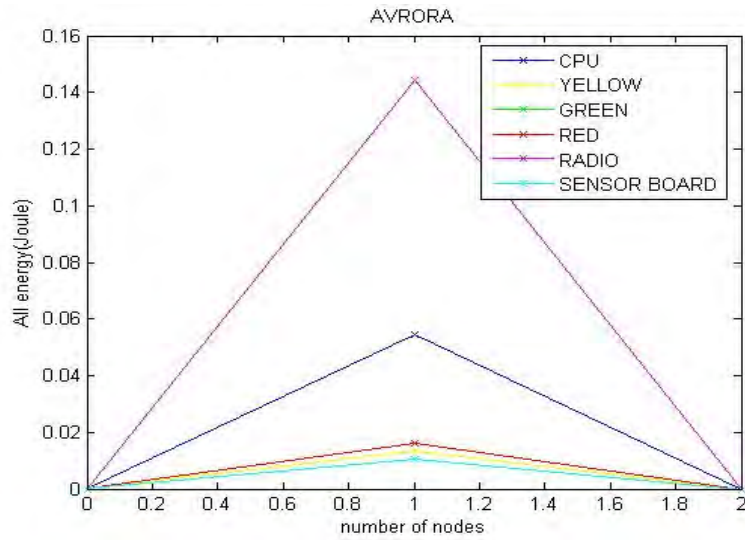
Εκτελώντας τις εξής εντολές μπορούμε να αποκτήσουμε τον τύπο αυτό:

```
make mica2
```

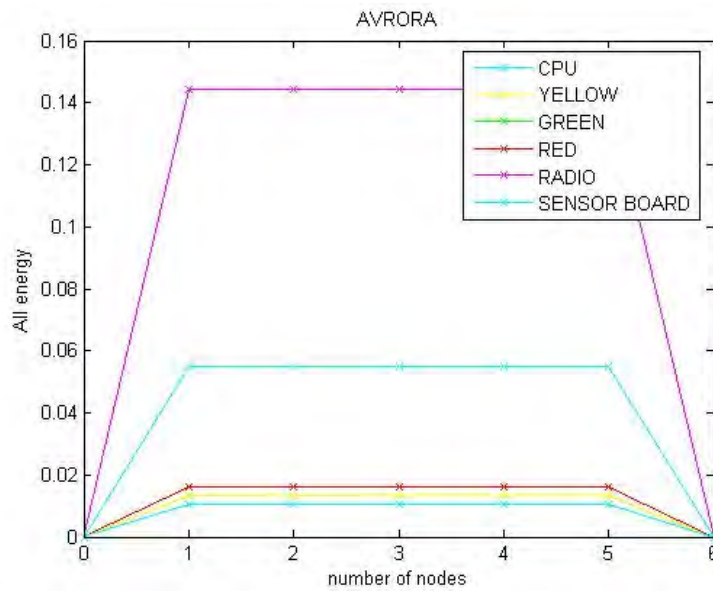
```
mv build/mica2/main.exe CntToLedaAndRfm.elf
```

```
avr-objdump -zhD CntToLedaAndRfm.elf > CntToLedaAndRfm.od
```

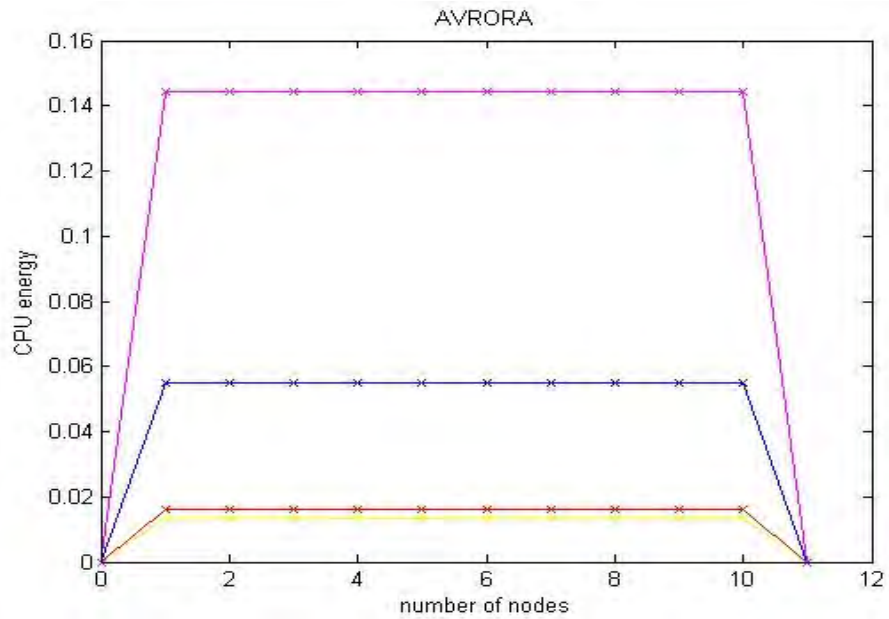
Ξεκινώντας την ανάλυση ενέργειας όπως και πριν, αρχικά για ένα mote και στην συνέχεια για πέντε, δέκα, δεκαπέντε κλπ.



Αποτελέσματα για ένα κόμβο



Αποτελέσματα για πέντε κόμβους



Αποτελέσματα για δέκα κόμβους

Χρησιμοποιώντας τον Ansgora στην πλατφόρμα mica2 για διαφορετικό αριθμό κόμβων, παρατηρούμαι ότι η κατανάλωση ενέργειας δεν μεταβάλλεται (αυξάνεται/μειώνεται) σημαντικά, και αυτό φαίνεται από τα γραφήματα παραπάνω για κάθε λειτουργία ξεχωριστά, και προφανώς η συνολική ενέργεια.

Επίλογος

Η ραγδαία εξέλιξη στον τομέα των ασύρματων επικοινωνιών την τελευταία δεκαετία έχει ως αποτέλεσμα την ανάγκη για συνεχή μελέτη πάνω στις νέες τεχνολογίες. Η επιστημονική κοινότητα πραγματοποιώντας γοργά βήματα προτείνει διαρκώς νέες λύσεις για τα προβλήματα και τις ανάγκες των καταναλωτών. Στα πλαίσια της εξέλιξης αυτής, έχει παρατηρηθεί τα τελευταία χρόνια ιδιαίτερη κινητικότητα στην έρευνα πάνω σε ασύρματα δίκτυα αισθητήρων, τα οποία καλούνται να καλύψουν πολλές ιδιαίτερες ανάγκες σε πληθώρα εφαρμογών.

Οι πολλές ιδιαιτερότητες των δικτύων αυτών συνεπάγονται και αντίστοιχες προκλήσεις στον ερευνητικό τομέα, τις οποίες οι ερευνητές στον παγκόσμιο χώρο καλούνται να αντιμετωπίσουν και οι οποίες αποτέλεσαν και το έναυσμα για την παρούσα εργασία. Από το πλήθος των προβλημάτων προς επίλυση, η συγκεκριμένη εργασία ασχολείται με την μελέτη των εργαλείων προσομοίωσης, και δη με τα ενεργειακά μοντέλα που αυτά χρησιμοποιούν.

Στόχος της εργασίας ήταν να παρουσιάσει με όσο το δυνατόν αναλυτικότερο τρόπο τα βασικά χαρακτηριστικά των δύο βασικών προσομοιωτών, του PowerTossim και του Avrora, και να εμβαθύνει στον τρόπο λειτουργίας τους.

Βιβλιογραφία-Αναφορές

- [1] “Simulating the Power Consumption of LargeScale Sensor Network Applications”, V. Schnayder, M. Hempstead, B. Chen, G.W. Allen, M. Welsh, Harvard University, Sensys '04, Nov 3-5, 2004
- [2] L. Lamport, “Time, clocks, and the ordering of events in a distributed system,” Communications of the ACM, vol. 21, no. 7, pp. 558–565, 1978.
- [3] O. Landsiedel, K. Wehrle, S. Rieche, S. Gotz, and L. Petrak, “Accurate prediction of power consumption in sensor networks,” 2004, manuscript
- [5] http://en.wikipedia.org/wiki/Sensor_node
- [6] <http://www.tinyos.net/>
- [7] <http://www.moteiv.com>
- [8] <http://en.wikipedia.org/wiki/Smardust>
- [9] Ben L. Titzer, Daniel K. Lee, Jens Palsberg «Aurora: Scalable Sensor Network Simulation with Precise Timing»
- [10] TOSSIM: A Simulator for TinyOS Networks
Philip Levis and Nelson Lee
- [11] TOSSIM: The Tiny OS Simulator
Nitish Jha
- [12] CS 441 – WIRELESS & MOBILE COMPUTING
Instructor – Dr. Kemal Akkaya
- [13] http://www.docs.tinyos.net/index.php/Main_Page

Παράρτημα

1. Powertossim, trace file with 1 mote
 2. Powertossim, trace file with 5 motes
 3. PowerTossim, trace file with 10 motes
 4. PowerTossim, trace file with 20 motes
 5. Avrora, trace file with 1 mote
 6. Avrora, trace file with 5 mote
 7. Avrora, trace file with 10 mote
 8. Avrora, trace file with 15 mote
-
- ✓ Το αρχείο του PowerTossim: Alternative_energy_model.txt
 - ✓ Το αρχείο alternative_avrora.java