

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ



ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΛΕΙΤΟΥΡΓΙΑ ΟΛΟΚΛΗΡΩΜΕΝΩΝ ΕΦΑΡΜΟΓΩΝ ΔΙΑΧΥΤΟΥ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕ ΧΡΗΣΗ ΑΣΥΡΜΑΤΩΝ ΕΤΕΡΟΓΕΝΩΝ ΣΥΣΚΕΥΩΝ, ΑΙΣΘΗΤΗΡΩΝ ΚΑΙ ΕΛΕΓΚΤΩΝ

Ορέστης Ακριβόπουλος
Α.Μ. : 713

Επιβλέπων : Καθηγητής Παύλος Σπυράκης
Συνεπιβλέπων : Δρ. Ιωάννης Χατζηγιαννάκης

Τριμελής επιτροπή : κ.Παύλος Σπυράκης, Καθηγητής
κ.Χρήστος Κακλαμάνης, Καθηγητής
κ.Σωτήρης Νικολετσέας, Αναπλ. Καθηγητής

Πάτρα
Ιούλιος 2012

Περίληψη

Στην παρούσα μεταπτυχιακή διπλωματική εργασία μελετάται ο σχεδιασμός, η ανάπτυξη, η εγκατάσταση και λειτουργία ολοκληρωμένων εφαρμογών διάχυτου υπολογισμού με χρήση ετερογενών ασυρμάτων αισθητήρων. Οι συγκεκριμένες συσκευές μπορούν να λειτουργήσουν ως απλοί αισθητήρες αλλά και ως ελεγκτές ή ενεργοποιητές. Η ασύρματη επικοινωνία ετερογενών συσκευών σε ένα διάχυτο σύστημα παρουσιάζει σημαντικά προβλήματα καθώς χρησιμοποιούνται συσκευές με εντελώς διαφορετική αρχιτεκτονική, διαφορετικά χαρακτηριστικά και διαφορετικές δυνατότητες όσον αφορά τις τεχνολογίες που υποστηρίζουν για την υλοποίηση εφαρμογών σε αυτές.

Στα πλαίσια της συγκεκριμένης διπλωματικής, εστιάζουμε αρχικά στον εντοπισμό των αιτιών που δεν επιτρέπουν την ασύρματη επικοινωνία των συσκευών και προτείνουμε συγκεκριμένες λύσεις. Προτείνουμε επίσης, μια αρχιτεκτονική ενός συγκεκριμένου συστήματος η οποία βασίζεται σε μια ιεραρχία επιπέδων προσφέροντας επεκτασιμότητα καθώς επίσης και χρήση διαφορετικών συσκευών (heterogeneity). Το συγκεκριμένο σύστημα είναι ανεξάρτητο από τις συσκευές που χρησιμοποιούνται (platform & hardware independent) καθώς οι λειτουργίες και τα δεδομένα παρέχονται με δομημένο τρόπο, π.χ., με την χρήση Web Services.

Για την μελέτη της συμπεριφοράς του συστήματος αναπτύσσονται συγκεκριμένες ολοκληρωμένες εφαρμογές οι οποίες παρουσιάζουν τις δυνατότητες του. Κάθε μία από αυτές εστιάζει σε διαφορετικά χαρακτηριστικά του. Στο συγκεκριμένο σύστημα επιτρέπεται, μεταξύ άλλων, η ανάπτυξη παιχνιδιών απόκρισης πραγματικού χρόνου με χρήση ετερογενών ασυρμάτων αισθητήρων. Οι εφαρμογές χρησιμοποιούν ένα σύνολο διαφορετικών συσκευών και συγκεκριμένα iSense, Arduino, SunSPOT, TelosB και XBee.

Abstract

Within the scope of this MSc dissertation, we discuss the design and implementation of pervasive applications on top of heterogeneous wireless sensor network environment. The wireless communication between heterogeneous devices is an inherently difficult research problem due to fundamental differences in system architecture, properties and capabilities of these devices.

Initially, our research focused on the identification of the problems related to the intercommunication among the devices of a heterogeneous wireless sensor network. As a solution, we propose a new abstract system that provides the key qualities needed for a successful pervasive system; expandability, scalability and performance. The new architecture achieves interoperability among the devices by introducing abstraction in the communication protocols (MAC, Transport and Application Layers).

In order to demonstrate the applicability of our system we include various representative use case scenarios, that illustrate the usage of our infrastructure. Each scenario focuses on different properties of the system and uses a combination of devices such as iSense, Arduino, SunSPOT, TelosB and XBee.

Δημοσιεύσεις

Μέρος της δουλειάς που αναπτύχθηκε στα πλαίσια της μεταπτυχιακής διπλωματικής εργασίας δημοσιεύτηκε στα ακόλουθα:

Chapters in Books:

- **Akribopoulos Orestis**, Amaxilatis Dimitrios, Chatzigiannakis Ioannis, Georgitzikis Vasileios and Logaras Marios, **Distributed Self-organized Societies in Action**, chapter in the book “Distributed Self-organized Societies of Tiny Artefacts: Design & Implementation”, editors: I.Chatziannakis and P.Spirakis. Printed by PerAda, the Future and Emerging Technologies coordination action funded by the European Commission under FP7. Lulu Publishers, Raleigh, N.C., May 2011 (ISBN 5800059245538).
- **Akribopoulos Orestis**, Cord-Landwehr Andreas, Hasemann Henning and Kempkes Barbara, **Motion Coordination**, chapter in the book “Distributed Self-organized Societies of Tiny Artefacts: Design & Implementation”, editors: I.Chatziannakis and P.Spirakis. Printed by PerAda, the Future and Emerging Technologies coordination action funded by the European Commission under FP7. Lulu Publishers, Raleigh, N.C., May 2011 (ISBN 5800059245538).
- **Akribopoulos Orestis**, Bergamini Lorenzo, Cord-Landwehr Andreas, Koninis Christos and Segura Antoni, **End-to-End Communication**, chapter in the book “Distributed Self-organized Societies of Tiny Artefacts: Design & Implementation”, editors: I.Chatziannakis and P.Spirakis. Printed by PerAda, the Future and Emerging Technologies coordination action funded by the European Commission under FP7. Lulu Publishers, Raleigh, N.C., May 2011 (ISBN 5800059245538).

Publications:

- **Akribopoulos Orestis**, Georgitzikis Vasileios, Protopapa Anastasia and Chatzigiannakis Ioannis, Building a Platform-agnostic Wireless Network of Interconnected Smart Objects, in: 15th Panhellenic Conference on Informatics with international participation (**PCI 2011**), IEEE, Kastoria, Greece, 2011
- **Akribopoulos Orestis**, Chatzigiannakis Ioannis, Koninis Christos and Theodoridis Vaggelis, A Web services-oriented Architecture for Integrating Small Programmable Objects in the Web Of Things, in: 3rd International Conference on Developments in eSystems Engineering (**DeSE 2010**), London, UK, 2010.

Technical Reports:

- **O.Akribopoulos**, Georgitzikis Vasileios, Koninis Christos, Papavasileiou Ioannis and Chatzigiannakis Ioannis: Deployment and Evaluation of a 802.15.4 Heterogeneous Network, 2011. Technical Report in **ACM Computing Research Repository**, arXiv:1102.0058v2, February, 2011.

Ευχαριστίες

Αρχικά θα ήθελα να ευχαριστήσω τον Καθηγητή του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών και Πληροφορικής, και επιβλέποντα καθηγητή μου, κ. Παύλο Σπυράκη που με τίμησε με την συνεργασία του και μου έδωσε την ευκαιρία να ασχοληθώ με το συγκεκριμένο αντικείμενο στα πλαίσια της μεταπτυχιακής διπλωματικής εργασίας μου.

Στη συνέχεια θα ήθελα να ευχαριστήσω τα μέλη της τριμελούς επιτροπής, τον Καθηγητή του ΤΜΗΥΠ, κ. Χρήστο Κακλαμάνη και τον Καθηγητή του ΤΜΗΥΠ, κ. Σωτήρη Νικολετσέα για την καθοδήγηση τους και την υποστήριξη τους στην ολοκλήρωση αυτής της εργασίας.

Επίσης θα ήθελα να εκφράσω τις εγκάρδιες ευχαριστίες μου στον Δρ. Ιωάννη Χατζηγιαννάκη, για την φιλική συμπαράσταση και καθοδήγηση του καθ'όλη τη διάρκεια των μεταπτυχιακών μου σπουδών. Η ουσιαστική επικοινωνία και η καθημερινή συνεργασία μου μαζί του, με βοήθησαν στην ανάπτυξη ώριμου τρόπου σκέψης και αποδοτικού και αποτελεσματικού τρόπου δράσης.

Τέλος, θα ήθελα να ευχαριστήσω τον φίλο και συνάδελφο Βασίλη Γεωργιτζίκη, για την πολύτιμη βοήθεια που μου προσέφερε κατά την εκπόνηση της διπλωματικής εργασίας μου.

Περιεχόμενα

Περίληψη	3
Ευχαριστίες	iii
1 Εισαγωγή	1
1.1 Κίνητρο και σημασία του θέματος	1
1.2 Στόχοι της διπλωματικής εργασίας	2
1.3 Συνεισφορά της διπλωματικής εργασίας	2
1.4 Δομή της διπλωματικής εργασίας	2
2 Ετερογενές επίπεδο επικοινωνίας με βάση το IEEE 802.15.4	5
2.1 Εισαγωγή	5
2.2 Πλατφόρμες Υλοποίησης	5
2.3 IEEE 802.15.4	8
2.4 Ασυμβατότητες και προτεινόμενες τροποποιήσεις	11
2.5 Πειραματική αξιολόγηση	13
3 Διασύνδεση αυτοκινούμενων συσκευών με ασύρματα δίκτυα αισθητήρων	21
3.1 Εισαγωγή	21
3.2 Επικοινωνία από άκρη σε άκρη	21
3.3 Moway Robots και SunSPOTs	23
3.4 Delay Tolerant Υπηρεσίες	25
3.5 Σενάριο Χρήσης: Testbed	28
4 Ανάπτυξη εφαρμογών με το μοντέλο των Web Services	31
4.1 Εισαγωγή	31
4.2 Αρχιτεκτονική	31
4.3 Ελεγκτής	33
4.3.1 Ελεγκτής ασυρμάτου δικτύου	33
4.3.2 Ρυθμίσεις με χρήση της WiseML	35
4.3.3 Proxy	36
4.4 Small Programmable Objects	36
4.4.1 Nano HTTP server	37
4.4.2 Senselets	37
4.4.3 Discovery protocol	37
4.4.4 Virtual Radio Routing Protocol	38
4.4.5 WSN Runtime	38
4.5 Υλοποίηση πρωτοτύπου και εγκατάσταση	38

4.5.1	Σενάριο Χρήσης: Αυτοματισμοί γραφείου	40
4.5.2	Σενάριο Χρήσης: Απομακρυσμένη Παρακολούθηση Κτιρίων	41
5	Ανάπτυξη εφαρμογών με το μοντέλο των Distributed Objects	43
5.1	Εισαγωγή	43
5.2	Αρχιτεκτονική και Υπηρεσίες	44
5.2.1	Αρχιτεκτονική του Controller	45
5.2.2	Αρχιτεκτονική των Smart Objects	47
5.3	Σενάρια χρήσης	48
5.3.1	Αυτοματισμοί Γραφείων	48
5.3.2	Εφαρμογή Ψυχαγωγίας	49
5.4	Συμπεράσματα	50
6	Συμπεράσματα	51

1

Εισαγωγή

1.1 Κίνητρο και σημασία του θέματος

Τα τελευταία χρόνια η χρήση κινητών συσκευών σχετίζεται με τις καθημερινές μας ανάγκες και οι τεχνολογικές εξελίξεις επηρεάζουν άμεσα και σε μεγάλο βαθμό την καθημερινή μας ζωή. Κινητά τηλέφωνα, notebooks, PDAs δημιουργούν διαφορετικά συστήματα με διαφορετικές δυνατότητες, διαφορετικές απαιτήσεις και διαφορετικά χαρακτηριστικά. Η διασύνδεση αυτών των ετερογενών συστημάτων παρουσιάζει αρκετές προκλήσεις και δεν είναι μία αυτόματη διαδικασία.

Το όραμα είναι η δημιουργία του Web of Things [2], ενός ιστού στον οποίο θα είναι συνδεδεμένες όλες οι συσκευές και τα 'έξυπνα' αντικείμενα (Smart Objects [3]) που χρησιμοποιούμε στην καθημερινή μας ζωή και είναι μια επέκταση του Internet of Things [1]. Όλα τα Smart Objects θα είναι συνδεδεμένα μεταξύ τους αλλά και με το Web με διαφανή τρόπο, δημιουργώντας ένα περιβάλλον διάχυτου υπολογισμού. Χαρακτηριστικό παράδειγμα Smart Objects που μπορούν να χρησιμοποιηθούν για την υλοποίηση του οράματος είναι τα ασύρματα δίκτυα αισθητήρων με τα οποία μελετώνται στην παρούσα μεταπτυχιακή διπλωματική εργασία. Τυπικές εφαρμογές των δικτύων αυτών έχουν να κάνουν με την παρακολούθηση του φυσικού, βιομηχανικού και οικιακού περιβάλλοντος και τις συνθήκες που επικρατούν σε αυτό. Με την προσθήκη actuators (ενεργοποιητών) στα δίκτυα αυτά επιτυγχάνουμε τον απομακρυσμένο ή ακόμη και τον αυτοματοποιημένο έλεγχο των συγκεκριμένων περιβάλλοντων.

Τα μέχρι σήμερα υλοποιημένα συστήματα όμως, είτε κάνουν περιορισμένη χρήση αισθητήρων είτε είναι αποσυνδεδεμένα μεταξύ τους. Η προσπάθεια απόκρυψης της ύπαρξης διαφορετικών συστημάτων και η παροχή ενός απλοποιημένου φαινομενικά συστήματος στους τελικούς χρήστες υλοποιεί κατά βάση την ιδέα ενός διάχυτου υπερσυστήματος. Η ανάπτυξη λογισμικού για τέτοια συστήματα παρουσιάζει πολλές προκλήσεις. Η επικοινωνία διαφορετικών συσκευών μέσω διαφορετικών καναλιών, η αποθήκευση δεδομένων και αντικειμένων καθώς και ένα κοινό μοντέλο αρχιτεκτονικής αποτελούν τις σημαντικότερες από αυτές.

Στην διπλωματική εργασία μελετάται ο σχεδιασμός, η ανάπτυξη, η εγκατάσταση και λειτουργία ολοκληρωμένων εφαρμογών Διάχυτου Υπολογισμού με χρήση Ετερογενών Ασύρματων Αισθητήρων. Οι συγκεκριμένες συσκευές μπορούν να λειτουργήσουν ως απλοί αισθητήρες αλλά και ως ελεγκτές ή ενεργοποιητές. Η ασύρματη επικοινωνία ετερογενών συσκευών σε ένα διάχυτο σύστημα παρουσιάζει σημαντικά προβλήματα καθώς χρησιμοποιούνται συσκευές με εντελώς διαφορετική αρχιτεκτονική, διαφορετικά χαρακτηριστικά και διαφορετικές δυνατότητες όσον αφορά της τεχνολογίες που υποστηρίζουν για την υλοποίηση εφαρμογών σε αυτές.

1.2 Στόχοι της διπλωματικής εργασίας

Στα πλαίσια της συγκεκριμένης διπλωματικής, εστιάζουμε αρχικά στον εντοπισμό των προβλημάτων που δεν επιτρέπουν την ασύρματη επικοινωνία των συσκευών και προτείνουμε συγκεκριμένες λύσεις. Προτείνεται μια αρχιτεκτονική ενός συγκεκριμένου συστήματος η οποία βασίζεται σε μια ιεραρχία επιπέδων προσφέροντας επεκτασιμότητα καθώς επίσης και χρήση διαφορετικών συσκευών (heterogeneity). Το συγκεκριμένο σύστημα είναι ανεξάρτητο από τις συσκευές που χρησιμοποιούνται (platform & hardware independent) καθώς οι λειτουργίες και τα δεδομένα παρέχονται με δομημένο τρόπο, π.χ., με την χρήση Web Services.

Για την μελέτη της συμπεριφοράς του συστήματος αναπτύσσονται συγκεκριμένες ολοκληρωμένες εφαρμογές οι οποίες παρουσιάζουν τις δυνατότητες του. Κάθε μία από αυτές εστιάζει σε διαφορετικά χαρακτηριστικά του. Οι εφαρμογές χρησιμοποιούν ένα σύνολο διαφορετικών συσκευών και συγκεκριμένα iSense, Arduino, SunSPOT, TelosB και XBee.

1.3 Συνεισφορά της διπλωματικής εργασίας

Στα πλαίσια της παρούσας διπλωματικής εργασίας μελετάται σε βάθος το πρωτόκολλο IEEE 802.15.4 το οποίο είναι το βασικό πρωτόκολλο που χρησιμοποιείται για την επικοινωνία στα ασύρματα δίκτυα αισθητήρων. Για να εντοπιστούν οι λόγοι που δεν επιτρέπουν την επικοινωνία μεταξύ ετερογενών συσκευών παρ' όλο που χρησιμοποιούν το 802.15.4 απαιτείται η ενδελεχής μελέτη των Network Stack των συσκευών SunSPOT, iSense, Arduino, TelosB και XBee.

Επίσης μελετάται και προτείνεται ένα ολοκληρωμένο πακέτο λογισμικού το οποίο επιτρέπει την από άκρο σε άκρο (end-to-end) επικοινωνία και παρέχει και υπηρεσίες Delay Tolerant με την χρήση αυτοκινούμενων συσκευών και συγκεκριμένα των Moway. Επίσης μελετήθηκαν εις βάθος οι συσκευές της Sun Microsystems SunSPOTs, και ο τρόπος σχεδιασμού και ανάπτυξης λογισμικού για αυτά. Μελετήθηκαν θέματα επικοινωνίας και δρομολόγησης πακέτων σε κινητά δίκτυα αισθητήρων.

Στη συνέχεια μελετάται το μοντέλο των Web Services και των Distributed Objects και προτείνονται διάχυτα συστήματα τα οποία αναπτύχθηκαν βάση των ανωτέρω μοντέλων.

1.4 Δομή της διπλωματικής εργασίας

Σε αυτή την ενότητα παρουσιάζεται η δομή της παρούσας διπλωματικής εργασίας, η οποία χωρίζεται σε τέσσερις θεματικές ενότητες. Η πρώτη ενότητα περιλαμβάνει την εισαγωγή

της διπλωματικής, όπου αναφέρονται η χρησιμότητα της διπλωματικής, το κίνητρο ενασχόλησης με το συγκεκριμένο θέμα καθώς και η σημασία του θέματος, οι στόχοι της διπλωματικής εργασίας και η συνεισφορά της, και τέλος η δομή της. Στο δεύτερο μέρος γίνεται η παρουσίαση του πρωτοκόλλου 802.15.4 το οποίο χρησιμοποιείται κατά κόρον για την επικοινωνία στα ασύρματα δίκτυα αισθητήρων. Παρουσιάζονται και απαντώνται οι λόγοι που δεν επιτρέπουν την χρήση του για την επικοινωνία ετερογενών συσκευών. Το τρίτο μέρος ασχολείται με τη βιβλιοθήκη επικοινωνίας που αναπτύχθηκε στα πλαίσια της διπλωματικής και στη συνέχεια με δυο συστήματα τα οποία χρησιμοποιούν την συγκεκριμένη βιβλιοθήκη προσφέροντας δύο διαφορετικά μοντέλα ανάπτυξης εφαρμογών, με Web Services και σύμφωνα με το μοντέλο των Distributed Objects. Στο τελευταίο μέρος καταγράφονται τα συμπεράσματα και ο απολογισμός της συγγραφής της διπλωματικής καθώς επίσης και η βιβλιογραφία που χρησιμοποιήθηκε.

Πιο αναλυτικά, στο **κεφάλαιο 2** παρουσιάζεται το πρωτόκολλο 802.15.4 και οι λόγοι που δεν επιτρέπουν την επικοινωνία ανάμεσα σε συσκευές διαφορετικού Hardware παρόλο που χρησιμοποιούν το ίδιο πρωτόκολλο. Αφού μελετηθούν προσεκτικά οι λόγοι προτείνεται συγκεκριμένη λύση στο πρόβλημα.

Στο **κεφάλαιο 3** παρουσιάζεται ο σχεδιασμός και η υλοποίηση ενός end-to-end module και η χρήση του σε δίκτυα αισθητήρων σε συνδυασμό με αυτοκινούμενες συσκευές που προσφέρουν υπηρεσίες Delay Tolerant.

Το **κεφάλαιο 4** περιέχει τον σχεδιασμό και την υλοποίηση της αρχιτεκτονικής που επιτρέπει την ανάπτυξη εφαρμογών διάχυτου υπολογισμού με το μοντέλο των Web Services.

Το **κεφάλαιο 5** περιέχει τον σχεδιασμό και την υλοποίηση της αρχιτεκτονικής που επιτρέπει την ανάπτυξη εφαρμογών διάχυτου υπολογισμού με το μοντέλο των Distributed Objects.

Τέλος, στο **κεφάλαιο 6** παρουσιάζονται κάποια συμπεράσματα τα οποία προέκυψαν από τη συγγραφή της διπλωματικής εργασίας, και γίνεται ο απολογισμός όλων όσων επιτεύχθηκαν σε αυτή την προσπάθεια.

2

Ετερογενές επίπεδο επικοινωνίας με βάση το IEEE 802.15.4

2.1 Εισαγωγή

Στο παρόν κεφάλαιο παρουσιάζεται το πρωτόκολλο επικοινωνίας IEEE 802.15.4, ο τρόπος λειτουργίας του και οι λόγοι που δεν μπορεί να χρησιμοποιηθεί για την επικοινωνία μεταξύ ετερογενών συσκευών. Στη συνέχεια παρουσιάζεται η προτεινόμενη λύση για την δημιουργία ενός ετερογενούς επιπέδου επικοινωνίας βασισμένο στο IEEE 802.15.4 και τέλος τα σενάρια αξιολόγησης που εφαρμόσαμε για τον έλεγχο της απόδοσης του συγκεκριμένου επιπέδου.

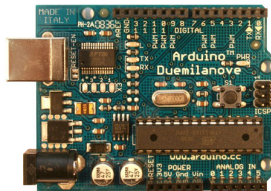
2.2 Πλατφόρμες Υλοποίησης

Σε αυτή την ενότητα παρουσιάζουμε τις πλατφόρμες - τεχνολογίες που χρησιμοποιήσαμε για την ανάπτυξη και το έλεγχο του ετερογενούς ασύρματου επιπέδου.

Arduino Duemilanove with XBee 1mW Chip Antenna

Το Arduino [25] (Σχήμα 2.1(α')) είναι μια πλατφόρμα ανοιχτού κώδικα. Το Arduino Duemilanove είναι μια μικροελεγκτική συσκευή βασισμένη στον μικροεπεξεργαστή ATmega328. Η μνήμη RAM του έχει μέγεθος 16KB. Μπορεί να *αισθάνθαι* το περιβάλλον και να το επηρεάσει ελέγχοντας φώτα, κινητήρες, κτλ. Διαθέτει 14 ακροδέκτες εισόδου/εξόδου, 6 αναλογικές εισόδους, ένα ψηφιακό ταλαντωτή 16MHz, μία θύρα USB, ακροδέκτη τροφοδοσίας, ένα διασυνδετικό αγωγό ICSP και ένα κουμπί επαναφοράς (reset button). Οι βασικές βιβλιοθήκες του Arduino είναι γραμμένες στην γλώσσα C και C++ και έχουν μεταγλωττιστεί χρησιμοποιώντας τους μεταγλωττιστές avr-gcc και Avr Libc. Προγραμματίζεται χρησιμοποιώντας την γλώσσα προγραμματισμού Wiring [26], η οποία μοιάζει με την C++, με κάποιες απλοποιήσεις και αλλαγές. Δεν διαθέτει κάποιο λειτουργικό σύστημα όπως οι υπόλοιπες συσκευές, διαθέτει όμως πληθώρα βιβλιοθηκών και οδηγών (drivers)

που κάνουν την ανάπτυξη του κώδικα πολύ εύκολη. Το Arduino δεν έχει ενσωματωμένο πομποδέκτη, γι' αυτό το λόγο το συνδέουμε με ένα XBee Series 1 [27] κύκλωμα κεραίας. Το XBee (Σχήμα 2.1(β')) παρέχει δικτυακή σύνδεση στο Arduino μέσω του πρωτοκόλλου MAC IEEE 802.15.4 [24].



(α')



(β')

Σχήμα 2.1: Arduino (α') και XBee (β')

SunSPOT

Το SunSPOT [20] (SUN Small Programable Object Technology) (Σχήμα 2.2) είναι μια μικρή ασύρματη συσκευή με ενσωματωμένους αισθητήρες που λειτουργεί με μπαταρία και εκτελεί την εικονική μηχανή της Java: Squawk JVM. Αναπτύχθηκε στα εργαστήρια της Sun Microsystems (θυγατρική της Oracle πλέον) και ο κώδικας που εκτελεί είναι γραμμένος σε Java 2 Micro Edition (J2ME).

Ένα free-range SunSPOT διαθέτει ένα processor board, ένα sensor board και την μπαταρία. Στο processor board είναι συνδεδεμένος ο επεξεργαστής και ο πομποδέκτης. Ο επεξεργαστής του είναι ο 32-bit ARM920T που είναι χρονισμένος στα 180MHz. Διαθέτει μνήμη RAM 512KB και εξωτερική μνήμη Flash 4MB. Ο πομποδέκτης είναι ο CC2420 Chipcon που υλοποιεί το πρωτόκολλο MAC IEEE 802.15.4 [24]. Το sensor board διαθέτει αισθητήρες θερμοκρασίας, φωτός και επιταχυνσιόμετρο 3 αξόνων. Επιπλέον έχει 8 LED τριών χρωμάτων, 6 αναλογικές εισόδους που διαβάζονται μέσω Analog to Digital Converter, 2 κουμπιά, 5 pins εισόδου/εξόδου γενικού σκοπού και 4 pins εξόδου υψηλού ρεύματος. Διαθέτει επίσης θύρα mini-USB για την φόρτιση και τον προγραμματισμό του.



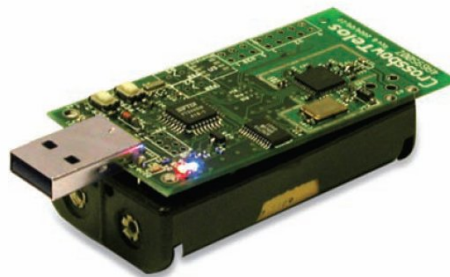
Σχήμα 2.2: SunSPOT

TelosB mote

Το TelosB [21] (Σχήμα 2.3) είναι μια ασύρματη συσκευή συμβατή με το πρωτόκολλο MAC IEEE 802.15.4 [24]. Διαθέτει μικροεπεξεργαστή TI-MSP430 16-bit που είναι χρονισμένος στα 8MHz. Έχει μνήμη RAM 10KB και εξωτερική μνήμη Flash 48KB. Η τροφοδοσία του γίνεται είτε μέσω της θύρας USB που διαθέτει για συλλογή δεδομένων και τον προγραμματισμό του, είτε μέσω της υποδοχής για 2 μπαταρίες AA που διαθέτει. Το TelosB έχει λειτουργικό σύστημα TinyOS [22] και το λογισμικό που εκτελεί είναι γραμμένο στην γλώσσα προγραμματισμού nesC. Ο πομποδέκτης που διαθέτει είναι ίδιος με αυτόν του SunSPOT, ο CC2420.

Το TinyOS είναι λειτουργικό σύστημα ανοιχτού κώδικα, σχεδιασμένο για ασύρματες συσκευές μικρής ισχύος. Αυτές μπορεί να είναι είτε συσκευές που χρησιμοποιούνται σε δίκτυα αισθητήρων, είτε ενσωματωμένες (embedded) συσκευές, κτλ. Ο προγραμματισμός στο TinyOS βασίζεται σε συνιστώσες (component-based programming). Κάθε διεργασία του συστήματος είναι τοποθετημένη σε διαφορετικές συνιστώσες έτσι ώστε όλα τα δεδομένα και οι συναρτήσεις μέσα σε μια συνιστώσα να είναι σημασιολογικά συσχετισμένες. Οι συνιστώσες είναι συνδεδεμένες μεταξύ τους μέσω διεπαφών (interfaces). Κάθε συνιστώσα παρέχει και χρησιμοποιεί άλλες συνιστώσες για να αποπερατώσει το έργο της. Το TinyOS παρέχει συνιστώσες για την διαχείριση μερών του συστήματος όπως: επικοινωνία, δρομολόγηση πακέτων, αισθητήρων και αποθήκευσης.

Το TinyOS έχει μία στοίβα (stack) με αποτέλεσμα να είναι απολύτως non-blocking. Έτσι διεργασίες που καθυστερούν αρκετά, όπως αποστολή μηνυμάτων, εκτελούνται στο παρασκήνιο και όταν ολοκληρωθούν πυροδοτείται η εκτέλεση γεγονότων (events) υπεύθυνων για την διαχείριση τους. Συνεπώς, κάθε εφαρμογή που είναι γραμμένη για το TinyOS υπακούει στο μοντέλο προγραμματισμού εφαρμογών οδηγούμενων από γεγονότα (event-driven applications).



Σχήμα 2.3: TelosB

iSense

Το iSense [15] (Σχήμα 2.4) αναπτύχθηκε από την Coalesenses GmbH που έχει βάση στο Λούμπεκ της Γερμανίας. Το iSense Core module περιλαμβάνει τον πομποδέκτη JN5139 που χρησιμοποιεί το πρωτόκολλο MAC IEEE 802.15.4 Zigbee. Ο επεξεργαστής του είναι ο 32 bit RISC Controller που είναι χρονισμένος στα 16MHz. Διαθέτει μνήμη RAM 96KB και εξωτερική σειριακή μνήμη Flash 128KB. Διαθέτει υποδοχές επέκτασης για όλα τα είδη από άλλες μονάδες και ενεργειακές πηγές. Διαθέτει επίσης ρυθμιστή

τάσης ελεγχόμενη από το λογισμικό, ο οποίος συνδυάζει υψηλή ενεργειακή απόδοση με ένα ευρύ φάσμα τάσης εφοδιασμού. Επίσης διαθέτει και υποδοχή κεραίας SMA εκτός από την ενσωματωμένη κεραμική κεραία.

Το λειτουργικό σύστημα που διαθέτει και το λογισμικό που μπορεί να εκτελέσει είναι γραμμένο σε C++. Το iSense λογισμικό σύστημα παρέχει ένα μεγάλο αριθμό από έτοιμες προς εκτέλεση υπηρεσίες και πρωτόκολλα, όπως δρομολόγηση, χρονικό συγχρονισμό, προγραμματισμό *πάνω από τον αέρα* (ασύρματα), αξιόπιστη μεταφορά μεμονωμένων πακέτων αλλά και ροής πακέτων (streams), υπηρεσίες λειτουργικού συστήματος και οδηγούς (drivers) ανεξαρτήτως πλατφόρμας για μια ευρεία ποικιλία αισθητήρων και άλλων επεκτάσεων.



Σχήμα 2.4: iSense

Στον παρακάτω πίνακα φαίνονται συγκεντρωμένες όλες οι πλατφόρμες, Πίνακας 2.1.

	Arduino	SunSPOT	TelosB	iSense
Επεξεργαστής	ATmega328	ARM920T	MSP430	JN5139
Συχνότητα λειτουργίας(MHz)	16	180	8	16
Μνήμη RAM	16 KB	512 KB	10 KB	96 KB
Μνήμη Flash	32 KB	4 MB	48 KB	128 KB
Πομποδέκτης	XBee Series 1	CC2420	CC2420	JN5139
Γλώσσα προγραμματισμού	Wiring (C++)	J2ME	nesC	C++
Λειτουργικό Περιβάλλον	—	Squawk	TinyOS	iSense

Πίνακας 2.1: Σύγκριση των διαφόρων συσκευών.

2.3 IEEE 802.15.4

Όπως έχουμε δει, οι συσκευές που χρησιμοποιούμε για την ανάπτυξη του επιπέδου μας χρησιμοποιούν το πρωτόκολλο MAC IEEE 802.15.4 [24]. Το πρωτόκολλο αναπτύχθηκε

από την ομάδα εργασίας 4 του IEEE 802.15 [28] και έχει σκοπό να ορίσει το φυσικό επίπεδο (Physical Layer – PHY) και το επίπεδο πρόσβασης μέσου (Medium Access Control – MAC), για ασύρματη επικοινωνία.

Ο σκοπός του 802.15.4 είναι να παρέχει ένα πρότυπο με χαμηλή πολυπλοκότητα, χαμηλό κόστος και χαμηλή κατανάλωση ενέργειας προσφέροντας παράλληλα ικανοποιητικούς ρυθμούς ασύρματης μετάδοσης δεδομένων. Ο ρυθμός μετάδοσης σε συγκεκριμένες περιπτώσεις αγγίζει τα 250 kb/sec .

Γενικά χαρακτηριστικά

Το IEEE 802.15.4 είναι σχεδιασμένο για να μπορεί να υποστηρίξει δίκτυα με λίγους κόμβους μέσα σε ένα κτήριο αλλά και χιλιάδες κόμβους σε κάποια γεωγραφική περιοχή. Οι κόμβοι αυτοί μπορεί να δημιουργήσουν κυκλοφορία είτε με κάποιο σταθερό ρυθμό είτε με πιθανές εκρήξεις. Έτσι λοιπόν ο φόρτος του δικτύου δεν είναι σταθερός αλλά εξαρτάται σε μεγάλο βαθμό από την εφαρμογή. Η λίστα που ακολουθεί παρουσιάζει κάποια από τα βασικά χαρακτηριστικά του πρωτοκόλλου.

- Ρυθμός μετάδοσης δεδομένων 250 kb/s , 40 kb/s και 20 kb/s
- Λειτουργία αστέρα ή peer-to-peer
- Εύρος διευθύνσεων 16 bit ή 64 bit
- Κατανομή εγγυημένων χρονοθυρίδων (guaranteed time slots – GTSS)
- Πολλαπλή πρόσβαση καναλιού με ανίχνευση φέροντος και αποφυγή συγκρούσεων (Carrier sense multiple access with collision avoidance – CSMA-CA)
- Αξιόπιστη μεταφορά με Acknowledgement μηνύματα
- Χαμηλή κατανάλωση ενέργειας
- Ανίχνευση Ενέργειας (Energy detection – ED)
- Ένδειξη ποιότητας σύνδεσης (Link quality indication – LQI)
- 16 κανάλια μέσα στη 2450 MHz ζώνη, 10 κανάλια στη 915 MHz ζώνη και 1 κανάλι στη 868 MHz ζώνη

MAC frames

Τα πλαίσια MAC (MAC frames) είναι συλλογές δεδομένων που στέλνονται μέσω του φυσικού επιπέδου (PHY). Για κάθε πακέτο δεδομένων που έρχεται από τα ανώτερα επίπεδα, το επίπεδο MAC προσθέτει δικά του δεδομένα ώστε να επιτευχθεί η ασύρματη επικοινωνία. Έτσι δημιουργείται το πλαίσιο MAC. Η δομή των πλαισίων έχει σχεδιαστεί για να διατηρεί την πολυπλοκότητα απλή και ταυτόχρονα να το καθιστά αρκετά ισχυρό για την μετάδοση σε ένα θορυβώδες κανάλι. Κάθε επίπεδο πρωτοκόλλου προσθέτει στη δομή δεδομένα στην αρχή και στο τέλος του πλαισίου (headers and footers). Κάθε πλαίσιο έχει μέγεθος μέχρι 127 bytes και αποτελείται από τις ακόλουθες βασικές συνιστώσες.

1. **MAC header – MHR:** Η κεφαλίδα του πλαισίου
2. **MAC payload:** Το ωφέλιμο φορτίο του πλαισίου, το οποίο περιέχει πληροφορίες σχετικές με τον τύπο του πλαισίου. Εδώ υπάρχουν και τα δεδομένα από τα υψηλότερα επίπεδα.
3. **MAC footer – MFR:** Η κατάληξη του πλαισίου, η οποία περιέχει μια σειρά ελέγχου (frame check sequence)

MAC header Όπως είδαμε και στα γενικά χαρακτηριστικά, το 802.15.4 παρέχει δυνατότητα διευθυνσιοδότησης 16 bit ή 64 bit . Αυτό καθορίζεται μέσα στο πεδίο Frame control, του οποίου τα περιεχόμενα φαίνονται στο Σχήμα 2.5. Στα bits 0-2 του Frame control καθορίζεται ο τύπος του πλαισίου. Τα bits αυτά μπορούν να πάρουν διάφορες τιμές, καθορίζοντας αν το πλαίσιο θα είναι τύπου Beacon, δεδομένων, αναγνώρισης (Acknowledgment) ή διαχείρισης MAC. Ο αποστολέας των δεδομένων μπορεί να ζητήσει, για το πλαίσιο που έστειλε, την αναγνώριση της επιτυχούς λήψης από τον παραλήπτη. Έτσι ο παραλήπτης είναι υποχρεωμένος να απαντήσει με ένα πλαίσιο αναγνώρισης (Acknowledgment) μέσα σε 12 συμβολικές περιόδους (symbol periods), δηλαδή 192 μsec . Το bit 6 από το υποπεδίο Frame control καθορίζει αν το πλαίσιο έχει προορισμό κάποιο προσωπικό δίκτυο με διαφορετικό αναγνωριστικό από αυτό που ανήκει η συσκευή. Τα bits 10-11 (Destination addressing mode) και τα 14-15 (Source addressing mode) δίνουν την δυνατότητα καθορισμού του είδους διευθυνσιοδότησης. Οι τιμές που μπορούν να πάρουν φαίνονται στο Σχήμα 2.6. Με αυτό τον τρόπο δίνεται η δυνατότητα καθορισμού διαφορετικού εύρους διευθυνσιοδότησης στον αποστολέα απ' ό,τι στον παραλήπτη. Αν επιθυμούμε διευθυνσιοδότηση 16 bit σε αποστολέα και παραλήπτη θα πρέπει τα bits 10-11 και τα 14-15 του πεδίου Frame control να πάρουν την τιμή 10.

Bits: 0–2	3	4	5	6	7–9	10–11	12–13	14–15
Frame type	Security enabled	Frame pending	Ack. request	Intra-PAN	Reserved	Dest. addressing mode	Reserved	Source addressing mode

Σχήμα 2.5: MAC Frame control.

Addressing mode value $b_1 b_0$	Description
00	PAN identifier and address field are not present.
01	Reserved.
10	Address field contains a 16 bit short address.
11	Address field contains a 64 bit extended address.

Σχήμα 2.6: Τρόποι διευθυνσιοδότησης.

Η οκτάδα 1 από το γενικό πλαίσιο (Σχήμα 2.7) περιέχει τον αριθμό ακολουθίας (sequence number), ο οποίος χρησιμοποιείται για την ταυτοποίηση των πλαισίων αναγνώρισης (Acknowledgment MAC Frames). Το επίπεδο MAC διαιρεί όλους τους κόμβους που λειτουργούν στο ίδιο κανάλι σε πολλαπλά προσωπικά δίκτυα (PAN). Κάθε ένα από τα δίκτυα έχει το μοναδικό του αναγνωριστικό. Το πεδίο αναγνωριστικού προσωπικού δικτύου προορισμού (Destination PAN identifier), Σχήμα 2.7, έχει εύρος 16 bits και καθορίζει το μοναδικό αναγνωριστικό του δικτύου που έχει ο παραλήπτης. Αν αυτό το πεδίο πάρει τιμή 0xffff τότε πρόκειται για broadcast αναγνωριστικό και έτσι θα γίνει αποδεκτό από κάθε συσκευή που *ακούει* στο κανάλι. Το πεδίο διεύθυνσης παραλήπτη (Destination address field) έχει, όπως είδαμε, μεταβλητό εύρος (16 bit ή 64 bit), και καθορίζει την διεύθυνση του παραλήπτη. Τα πεδία αναγνωριστικού προσωπικού δικτύου αποστολέα (Source PAN identifier) και διεύθυνσης αποστολέα (Source address field) καθορίζουν τις αντίστοιχες τιμές για το αναγνωριστικό και την διεύθυνση στον αποστολέα του πλαισίου.

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	FCS
		Addressing fields					
MHR						MAC payload	MFR

Σχήμα 2.7: Γενικό πλαίσιο MAC.

MAC payload

Το ωφέλιμο φορτίο του πλαισίου έχει μεταβλητό μέγεθος και περιέχει πληροφορίες σχετικά με μεμονωμένα είδη πλαισίων. Αν το υποπεδίο ενεργοποίησης ασφάλειας (security enabled subfield) έχει την τιμή 1 (Σχήμα 2.5) τότε το ωφέλιμο φορτίο προστατεύεται όπως έχει οριστεί η ασφάλεια γι' αυτή τη σχέση.

MAC footer

Η κατάληξη του πλαισίου περιέχει την σειρά ελέγχου, η οποία έχει μέγεθος 16 bit και έχει ITU-T κυκλικό έλεγχο πλεονασμού (CRC). Η σειρά ελέγχου υπολογίζεται μέσω του ακόλουθου πολυωνύμου γεννήτορα, βαθμού 16:

$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1$$

2.4 Ασυμβατότητες και προτεινόμενες τροποποιήσεις

Οι πλατφόρμες που παρουσιάσαμε υλοποιούν το 802.15.4 με διαφορετικό τρόπο, χωρίς να είναι συμβατοί αυτοί οι τρόποι μεταξύ τους. Γι αυτό το λόγο η επικοινωνία των τεσσάρων αυτών συσκευών δεν είναι δυνατή χωρίς να υπάρξει κάποια τροποποίηση σε κάποια από αυτές. Από την προηγούμενη ενότητα βλέπουμε ότι υπάρχει η δυνατότητα σε αυτές τις

συσκευές να έχουν διευθυνσιοδότηση 16 *bit* ή 64 *bit*. Παρόλα αυτά η στοίβα δικτύου του SunSPOT υποστηρίζει μόνο διευθύνσεις των 64 *bit*, ενώ το TelosB υποστηρίζει μόνο διευθύνσεις των 16 *bit*. Οι στοίβες των Arduino και iSense υποστηρίζουν και τα δύο είδη διευθυνσιοδότησης.

Βασισμένο στις προδιαγραφές του LowPan [16], η βιβλιοθήκη του SunSPOT παρέχει δρομολόγηση (routing), προώθηση πακέτων σε άλλους κόμβους (meshing) και κατακερματισμό πακέτων (fragmentation), δημιουργώντας πακέτα μεγέθους μέχρι και 1260 bytes. Το LowPan προσθέτοντας επιπλέον κεφαλίδες στα 802.15.4 πλαίσια επιτυγχάνει να δημιουργήσει αυτές τις δυνατότητες. Γνωρίζουμε ότι όταν γίνει λήψη ενός πλαισίου που έχει το bit “ACK” στην τιμή 1 ότι πρέπει να απαντήσει με ένα πλαίσιο τύπου acknowledgement (σε 12 συμβολικές περιόδους). Αυτό στο SunSPOT υλοποιείται με την αυτόματη απάντηση που παρέχει το κύκλωμα ασύρματης επικοινωνίας. Αντίθετα το TinyOS αφήνει την διαχείριση αυτής της απαίτησης στο λογισμικό του χρήστη, καθώς με την λήψη ενός πακέτου δεν εγγυάται η παράδοσή του από το radio chip CC2420 στον μικροεπεξεργαστή. Γι’ αυτό το λόγο η χρήση αυτόματης αποστολής πλαισίων αναγνώρισης από το υλικό (hardware-ACKs) μπορεί να έχει αποτέλεσμα εσφαλμένες αναγνώρισεις.

Οι διαφορές αυτές που μόλις αναφέραμε μπορεί να φανούν συνολικά στον Πίνακα 2.2. Την θέση του Arduino στον πίνακα έχει πάρει το XBee καθώς αυτό είναι που του προσφέρει την ασύρματη δικτύωση.

	XBee	SunSPOT	TelosB	iSense
μέγιστο payload	100 bytes	113 bytes	114 bytes	116 bytes
διευθύνσεις 16 bit	✓	✗	✓	✓
διευθύνσεις 64 bit	✓	✓	✗	✓
port messaging	✗	✓	✗	✗
PAN ID	0x1-0xffff	0x1-0xffff	0x1-0xffff	0x1
κανάλι	0x0B-0x1A	0x0B-0x1A	0x0B-0x1A	0x0B-0x1A
ασυμβατότητες	κεφαλίδες MaxStream	κεφαλίδες LowPan	το auto ack είναι ανενεργό	

Πίνακας 2.2: Διαφορές στην υλοποίηση για κάθε πλατφόρμα.

Για την επίτευξη της επικοινωνίας των τεσσάρων συσκευών, έπρεπε να ξεπεράσουμε αυτές τις διαφορετικές υλοποιήσεις που παρουσιάσαμε προηγουμένως. Έτσι λοιπόν έγιναν οι ακόλουθες επιλογές:

- εύρος διευθύνσεων: 16 bit
- αναγνωριστικό προσωπικού δικτύου: 0x1
- προσθήκη 2 επιπρόσθετων byte στα πακέτα επιπέδου εφαρμογής (application layer) με σκοπό την παράκαμψη του LowPan στο SunSPOT
- προσθήκη αριθμού θύρας (port number) σε κάθε πακέτο
- ενεργοποίηση των αυτόματων μηνυμάτων αναγνώρισης στο TelosB (hardware-ACKs)

Η προσθήκη αυτών των 2 επιπρόσθετων byte είναι κομβικής σημασίας, καθώς βοηθάει στην παράκαμψη του LowPan στο SunSPOT. Παρόλα αυτά πακέτα που έχουν κεφαλίδες για το LowPan θα γίνουν δεκτά από το SunSPOT χωρίς κανένα πρόβλημα, αφού η τροποποίηση που έγινε ήταν η προσθήκη αυτού του κώδικα για την παράκαμψη και όχι η απενεργοποίηση του. Έτσι επιτρέπουμε σε υπάρχουσες εφαρμογές που χρησιμοποιούν το προηγούμενο SDK (v5) να μπορούν να εκτελεστούν κανονικά.

Η προσθήκη του αριθμού θύρας (port number) σε κάθε πακέτο έγινε κι αυτή για λόγους συμβατότητας του SunSPOT με τις άλλες συσκευές. Στο SunSPOT κάθε πακέτο που στέλνεται έχει ακόμη 1 byte πριν από το ωφέλιμο φορτίο ώστε η επικοινωνία των εφαρμογών να γίνεται μέσω θυρών. Όπως φαίνεται και στον πίνακα 2.2 η δυνατότητα αυτή δεν υπάρχει στις άλλες πλατφόρμες. Η δομή των πακέτων σε επίπεδο εφαρμογής (application layer) φαίνεται στον Πίνακα 2.3. Οι τιμές των σταθερών είναι:

1 byte {	LP1
1 byte {	LP2
1 byte {	port No.
113 bytes {	payload

Πίνακας 2.3: Δομή πακέτου στο επίπεδο εφαρμογής.

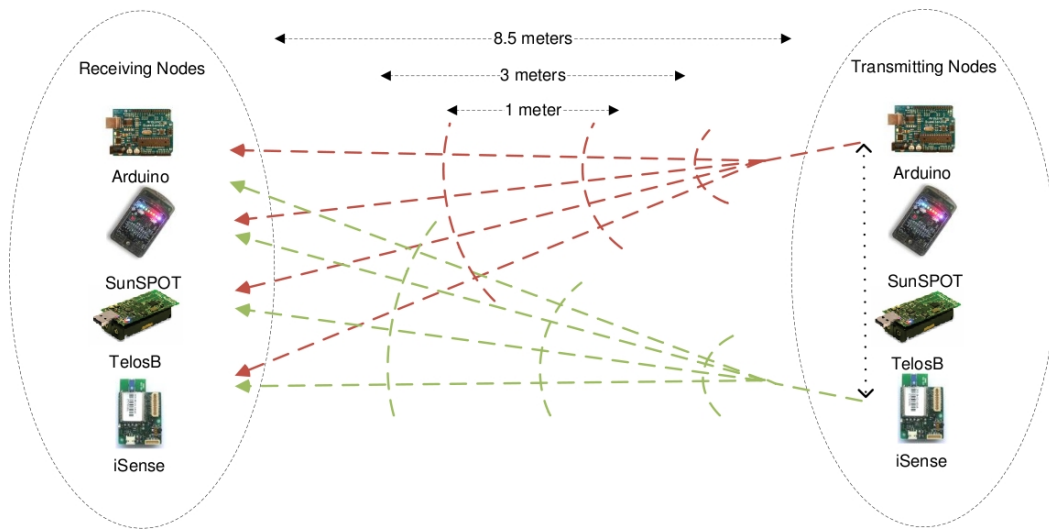
LP1 = 0x7f και **LP2** = 105 = 0x69.

2.5 Πειραματική αξιολόγηση

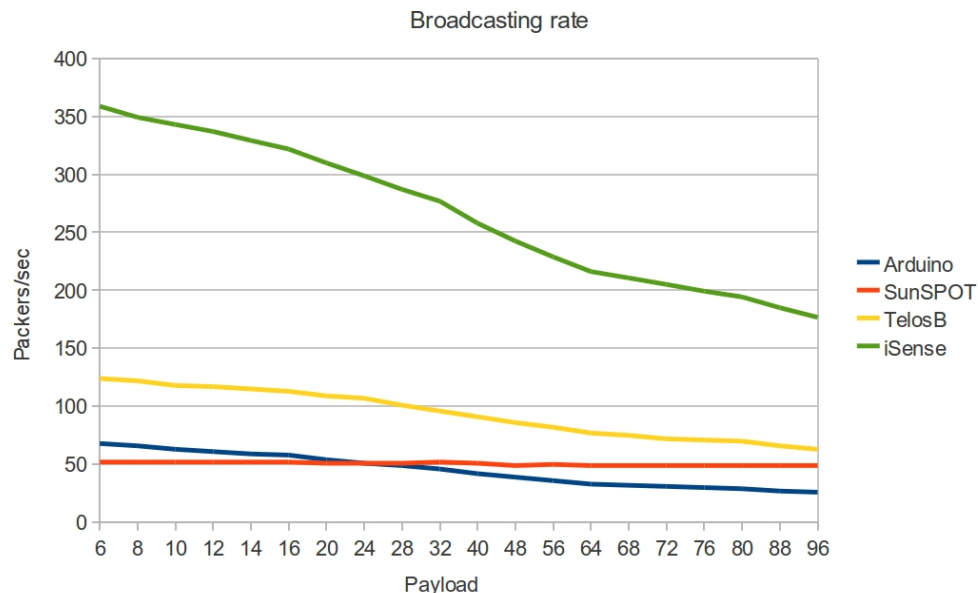
Οι μετρήσεις γίνονται ανάμεσα σε δύο ομάδες συσκευών, τις συσκευές λήψης και τις συσκευές αποστολής. Κάθε ομάδα αποτελείται από τις 4 διαφορετικές συσκευές: SunSPOT [20], TelosB [21], Arduino [25] και iSense [15]. Οι μετρήσεις γίνονται σε εξωτερικό χώρο ώστε να ελαχιστοποιηθούν οι ηλεκτρονικές παρεμβολές από άλλες συσκευές, οι ανακλάσεις από τους τοίχους κτλ. Κάθε ομάδα συσκευών είναι τοποθετημένη σε απόσταση 60 εκατοστόμετρων (cm) από το έδαφος ώστε να περιοριστούν οι ανακλάσεις του εδάφους. Η απόσταση μεταξύ των 2 ομάδων συσκευών είναι μεταβλητή για κάθε πείραμα που πραγματοποιούνται και παίρνει τιμές: 1 μέτρο, 3 μέτρα και 8.5 μέτρα. Η τοποθέτηση των 4 συσκευών κάθε ομάδας ήταν παρόμοια, με τρόπο ώστε κάθε συσκευή μιας ομάδας να έχει προσανατολισμό στις κεραίες της άλλης ομάδας. Η τοπολογία των πειραμάτων φαίνεται και στο Σχήμα 2.8.

Αρχικά παρουσιάζουμε την ταχύτητα αποστολής πακέτων (pps) ανά συσκευή. Το payload των πακέτων είναι μεταβλητό και κυμαίνεται από 6 έως 96 bytes. Σε κάθε συσκευή μετριέται η συνολική απαιτούμενη χρονική περίοδος για την αποστολή 500 πακέτων. Οι τιμές που παρουσιάζονται στο Σχήμα 2.9 είναι οι μέσες τιμές έπειτα από επανάληψη των πειραμάτων για 10 φορές.

Παρατηρούμε ότι το iSense έχει τον μεγαλύτερο ρυθμό αποστολής πακέτων ο οποίος μειώνεται όσο αυξάνεται το μέγεθος των πακέτων. Το Arduino έχει τον μικρότερο ρυθμό αποστολής ενώ το SunSPOT έχει σταθερό ρυθμό. Τα SunSPOT εκτελούν το Java Squawk JVM το οποίο δεν απαιτεί μεγαλύτερους υπολογιστικούς πόρους και εκεί δημιουργείται και το bottleneck το οποίο δεν επιτρέπει υψηλότερους ρυθμούς αποστολής.



Σχήμα 2.8: Η τοπολογία του πειράματος.

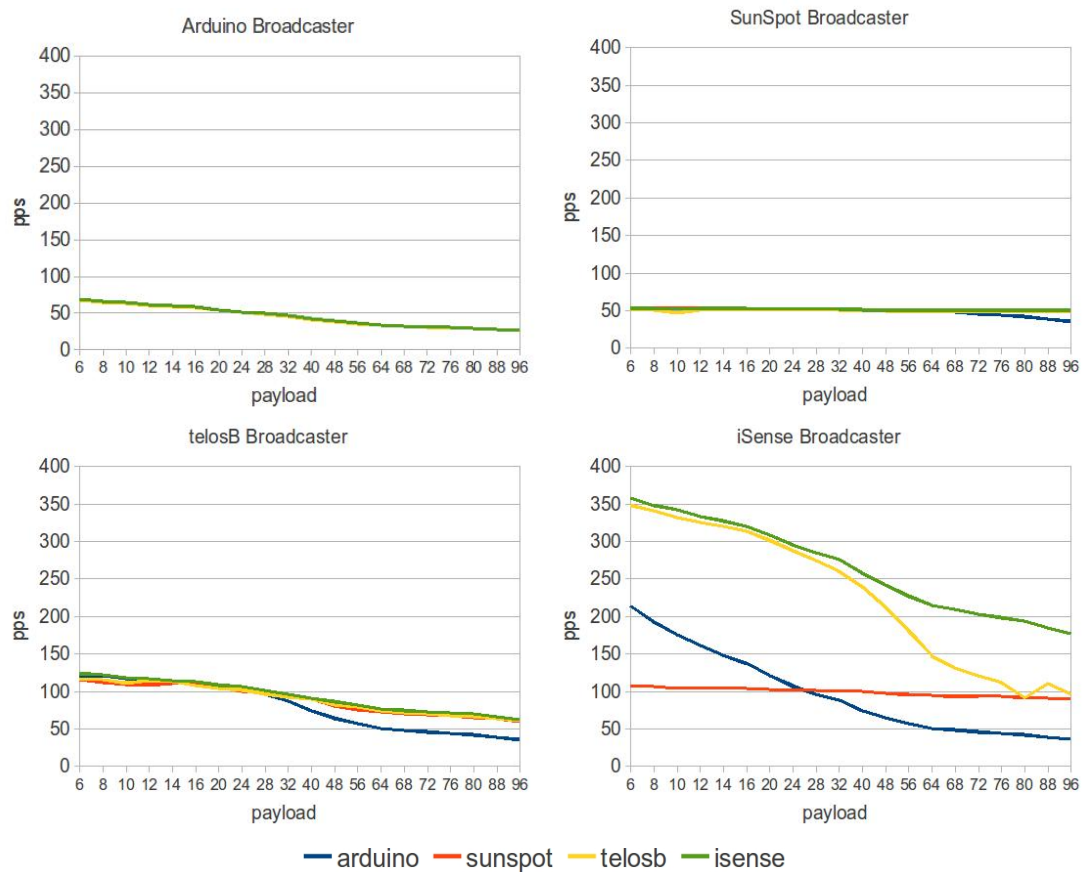


Σχήμα 2.9: Πακέτα/δευτερόλεπτο(pps) συναρτήσει του payload σε bytes

Ο ρυθμός λήψης πακέτων στις συσκευές εξαρτάται από τον ρυθμό μετάδοσης πακέτων του αποστολέα και δεν μπορεί να τον ξεπεράσει. Βλέπουμε ότι για συσκευές με μικρό ρυθμό μετάδοσης, όπως Arduino και SunSPOT, ο ρυθμός λήψης πακέτων από αυτές τις συσκευές είναι σχεδόν παρόμοιος με τον ρυθμό αποστολής. Για συσκευές με υψηλούς ρυθμούς μετάδοσης, όπως το iSense, οι ρυθμοί λήψης πακέτων είναι μικρότεροι σε σχέση με τον ρυθμό αποστολής. Αυτό οφείλεται στο γεγονός ότι το iSense στέλνει πιο γρήγορα από όσο μπορούν να λάβουν οι υπόλοιπες συσκευές με αποτέλεσμα να χάνονται πακέτα. Το φαινόμενο γίνεται πιο έντονο όσο αυξάνεται το μέγεθος των πακέτων. Σε γενικές εκτιμήσεις μέσα από τα ακόλουθα αποτελέσματα έχουμε ότι το iSense έχει τον καλύτερο ρυθμό λήψης, ανεξαρτήτως συσκευής αποστολής. Το Arduino έχει τους χαμηλότερους ρυθμούς λήψης, ενώ το SunSPOT με το telosB βρίσκονται ενδιάμεσα με το SunSPOT να

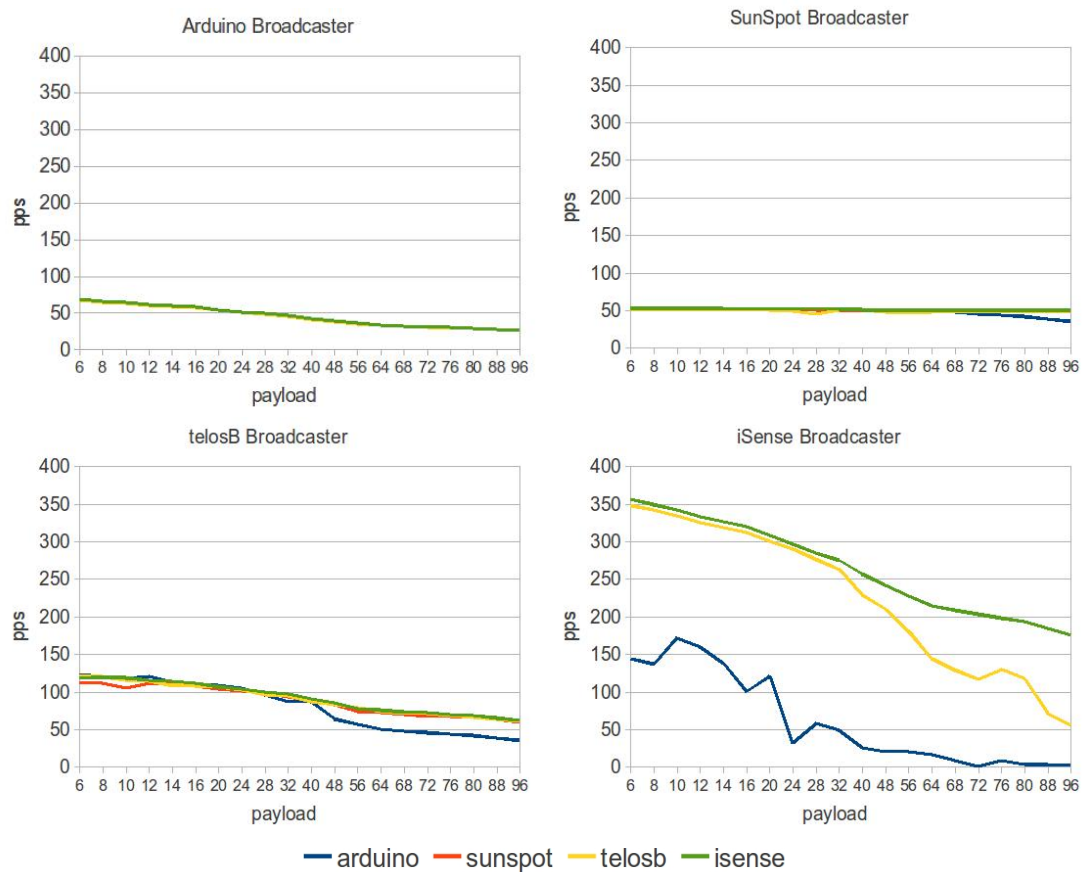
κατατάσσεται ελαφρώς χαμηλότερα.

Στην εικόνα 2.10 παρουσιάζονται οι ρυθμοί λήψεις για όλες τις συσκευές σε απόσταση ενός μέτρου.



Σχήμα 2.10: Ταχύτητα λήψης σε πακέτα ανά δευτερόλεπτο pps (1 μέτρο)

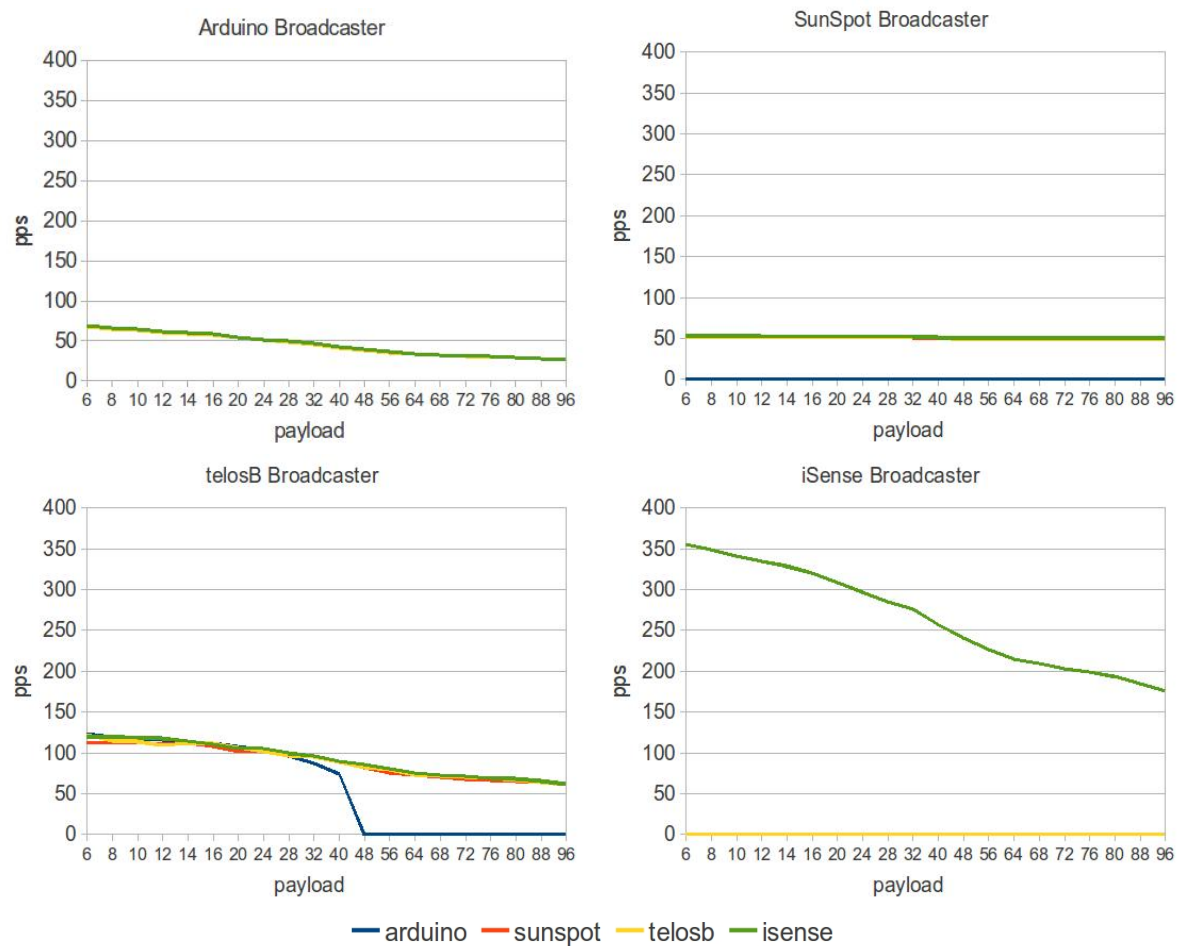
Στις εικόνες 2.11 και 2.12 παρουσιάζονται οι ρυθμοί λήψεις σε αποστάσεις 3 και 8 μέτρων αντίστοιχα. Όλες οι συσκευές δεν έχουν την ίδια εμβέλεια επικοινωνίας και όσο μεγαλώνει η απόσταση παρατηρούμε σε κάποιες συσκευές σχεδόν μηδενικούς ρυθμούς λήψεις. Πριν φτάσει ένα πακέτο στο λογισμικό της συσκευής ελέγχεται από την κεραία η ισχύς τους σήματος το οποίο θα μεταβεί στην συνέχεια σε πακέτο και αν η ισχύς είναι μικρότερη από κάποιο συγκεκριμένο όριο το πακέτο γίνεται drop. Κάθε συσκευή έχει διαφορετική κεραία η οποία έχει διαφορετικά όρια τα οποία δεν μπορούμε να πειράζουμε και γιαυτό το λόγο παρατηρείται το πρόβλημα με τους μηδενικούς ρυθμούς λήψεις.



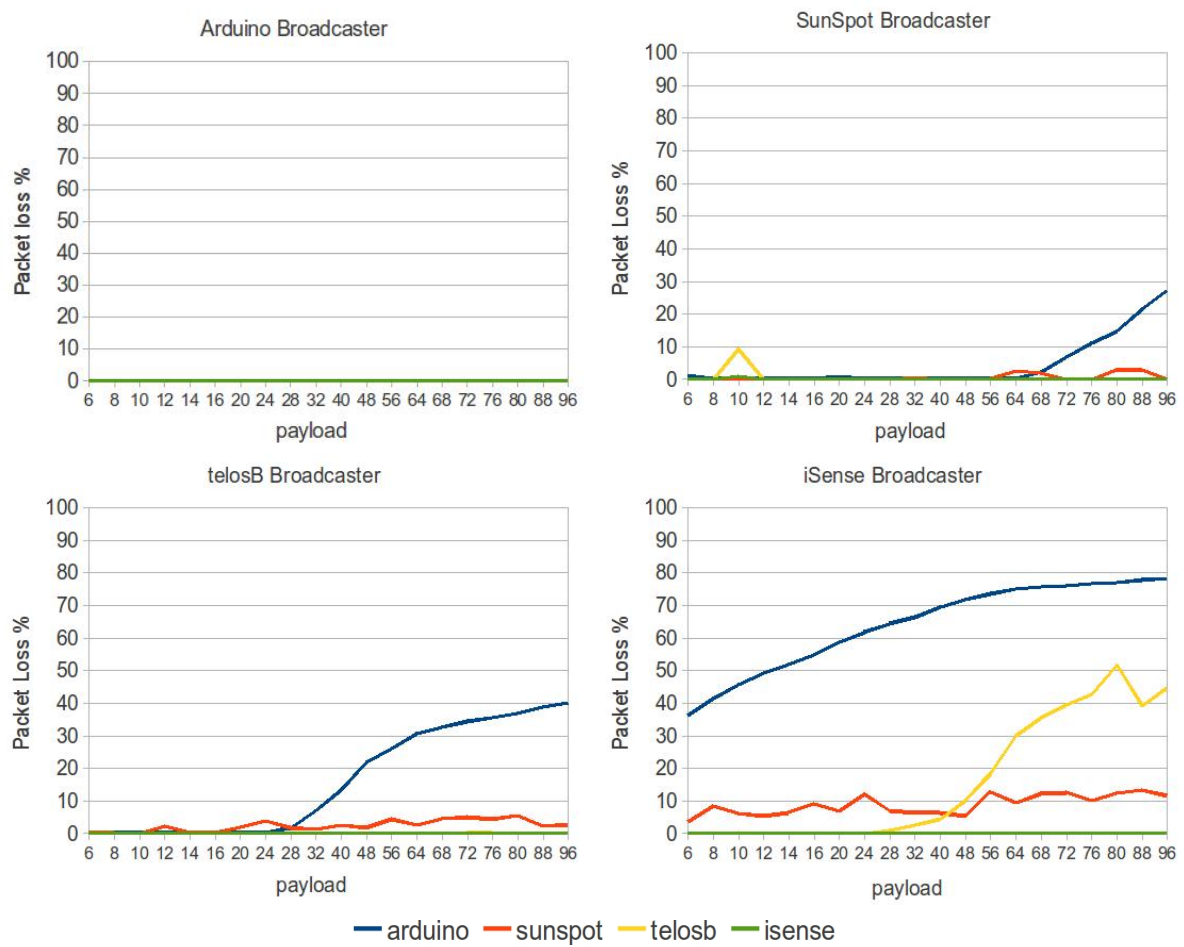
Σχήμα 2.11: Ταχύτητα λήψης σε πακέτα ανά δευτερόλεπτο pps (3 μέτρα)

Η απώλεια πακέτων συνδέεται άμεσα με τον μέγιστο δυνατό ρυθμό λήψης καθώς επίσης και με την απόσταση ανάμεσα σε πομπό και δέκτη. Όπως είναι φυσιολογικό όταν η ταχύτητα αποστολής είναι μεγαλύτερη από την μέγιστη δυνατή ταχύτητα λήψης, παρουσιάζεται αυξημένη απώλεια πακέτων. Επίσης όσο αυξάνεται η απόσταση, παρατηρείται εξασθένιση του σήματος και γιαυτό το λόγο αυξημένη απώλεια πακέτων.

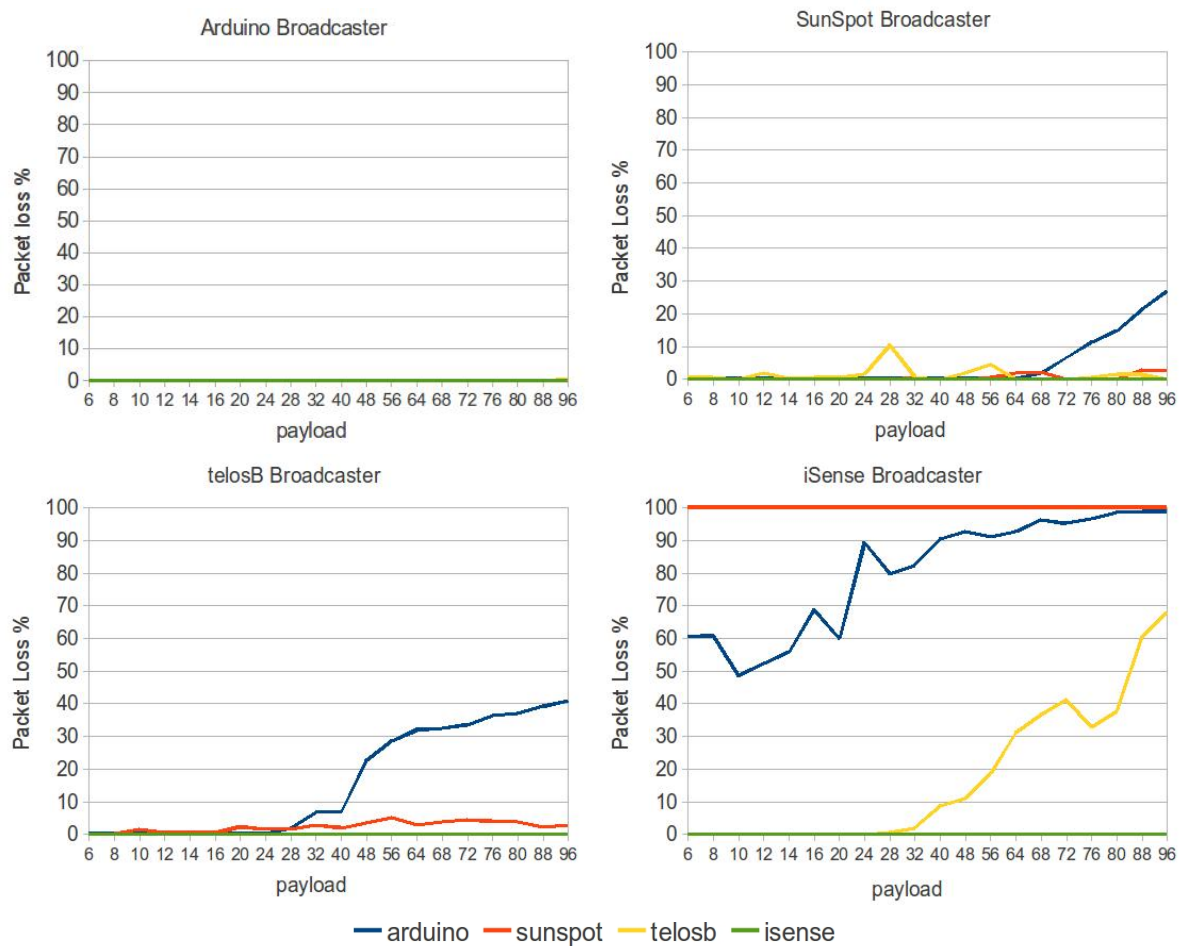
Στις εικόνες 2.13, 2.14, 2.15 παρουσιάζεται ο ρυθμός απώλειας πακέτων σε αποστάσεις 1,3 και 8 μέτρων αντίστοιχα.



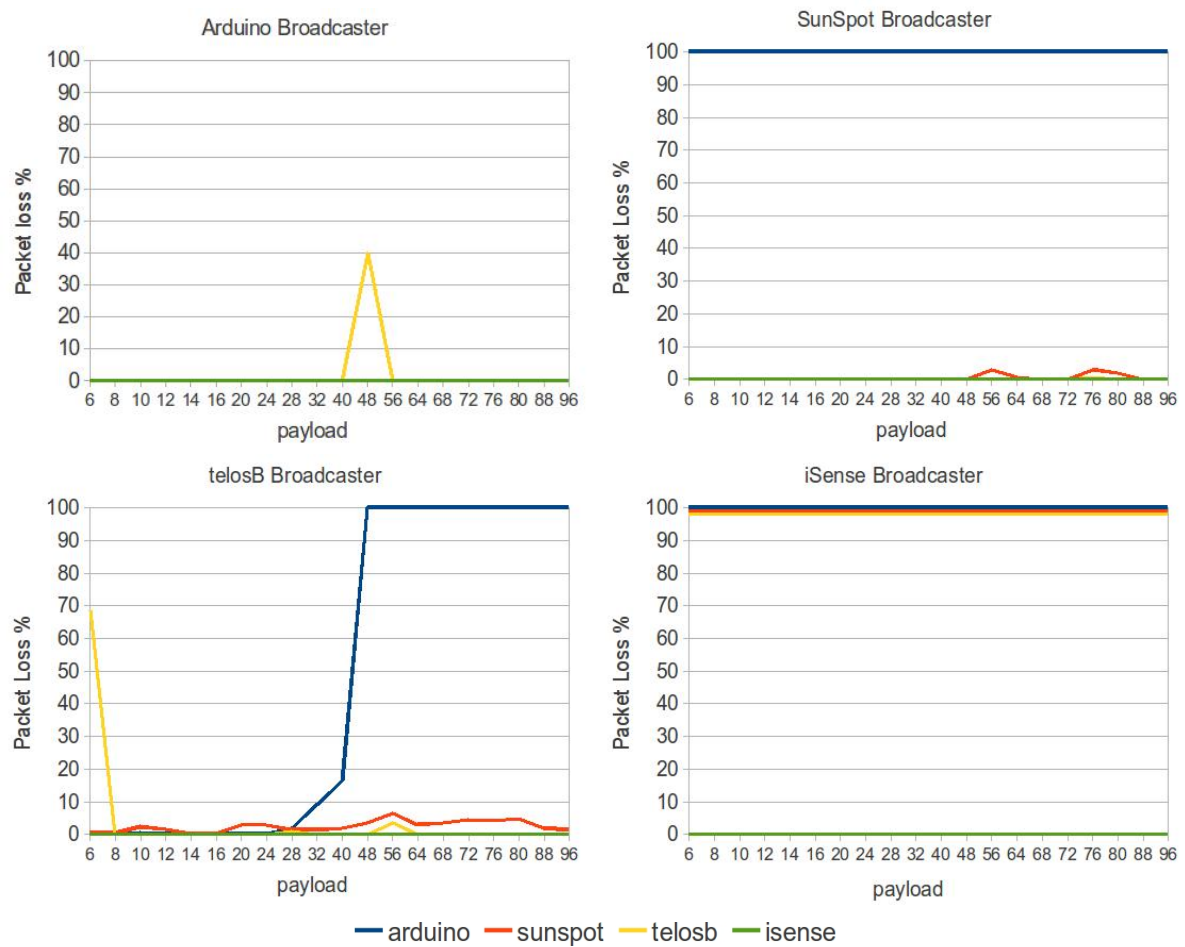
Σχήμα 2.12: Ταχύτητα λήψης σε πακέτα ανά δευτερόλεπτο pps (8 μέτρα)



Σχήμα 2.13: Ρυθμός απώλειας πακέτων (1 μέτρο)



Σχήμα 2.14: Ρυθμός απώλειας πακέτων (3 μέτρα)



Σχήμα 2.15: Ρυθμός απώλειας πακέτων (8 μέτρα)

3

Διασύνδεση αυτοκινούμενων συσκευών με ασύρματα δίκτυα αισθητήρων

3.1 Εισαγωγή

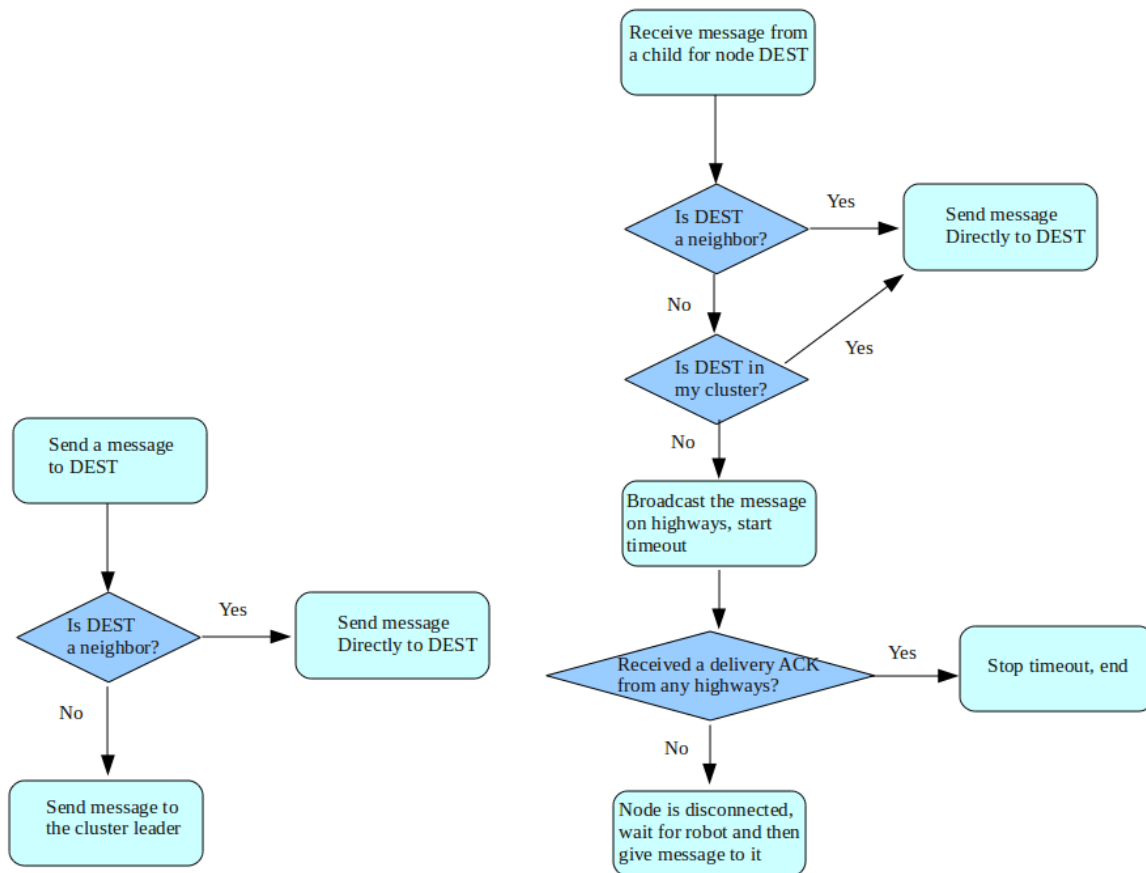
Ένα από τα πιο σημαντικά θέματα που πρέπει να αντιμετωπιστεί σε ένα ασύρματο δίκτυο αισθητήρων είναι η δυνατότητα επικοινωνίας ανάμεσα σε κάθε ζεύγος κόμβων. Εάν οι κόμβοι βρίσκονται μέσα στην εμβέλεια επικοινωνίας τους τότε δεν είναι αναγκαία η δρομολόγηση πακέτων καθώς υπάρχει δυνατότητα απευθείας επικοινωνίας. Όμως στις περισσότερες περιπτώσεις είναι αναγκαία η μεσολάβηση ενός ή περισσότερων ενδιάμεσων κόμβων οι οποίοι λειτουργώντας ως δρομολογητές (router) θα προωθήσουν τα πακέτα στον προορισμό τους.

Επίσης, στα WSN η τοπολογία του δικτύου αλλάζει συνεχώς και δημιουργεί προβλήματα στην μετάδοση και στην παράδοση μηνυμάτων. Οι αλγόριθμοι δρομολόγησης που θα εφαρμοστούν θα πρέπει να μπορούν να αντιληφθούν και να προσαρμοστούν στις αλλαγές της τοπολογίας έτσι ώστε να είναι δυνατή η δρομολόγηση μηνυμάτων. Συνεπώς, σκοπός είναι να έχουμε επικοινωνία από άκρη σε άκρη (end-to-end communication).

Στο παρών κεφάλαιο παρουσιάζουμε ένα end-to-end communication module το οποίο αποτελείται από έναν αλγόριθμο δρομολόγησης και υπηρεσίες delay-tolerant συνδεσιμότητας, ο συνδυασμός των οποίων προσφέρει προσαρμοστικό και αξιόπιστο τρόπο παράδοσης μηνυμάτων στον προορισμό τους. Ακόμη και σε περιπτώσεις κατακερματισμένων δικτύων εγγυάται την επικοινωνία ανάμεσα σε οποιοδήποτε ζεύγος κόμβων του δικτύου.

3.2 Επικοινωνία από άκρη σε άκρη

Το E2E communication module είναι ένα πακέτο λογισμικού το οποίο αποκρύπτει όλες τις πληροφορίες των διαφόρων πρωτοκόλλων του δικτύου και παρέχει ένα abstract layer στο χρήστη. Ο χρήστης όταν θέλει να στείλει ένα μήνυμα το μόνο που κάνει είναι να καλεί την συνάρτηση *send (destination)*, αποκρύπτοντας όλες τις εσωτερικές πληροφορίες που



Σχήμα 3.1: Διάγραμμα ροής του E2E module

απαιτούνται για την αποστολή του μηνύματος στον προορισμό.

Το E2E module είναι η υλοποίηση ενός αλγορίθμου δρομολόγησης, όπου το μήνυμα αποστέλλεται στον προορισμό του επιλέγοντας την καταλληλότερη στρατηγική βάση των πληροφοριών που παρέχονται από τους neighbor-discovery, clustering και highways αλγορίθμους. Επίσης μπορεί να χρησιμοποιήσει και τους κινητούς κόμβους του δικτύου (Moways) προσφέροντας delay-tolerant και αξιόπιστη συνδεσιμότητα, σε περιπτώσεις όπου ο αποστολέας και ο παραλήπτης δεν είναι μέλη του ίδιου δικτύου. Στην εικόνα 3.1 παρουσιάζεται το διάγραμμα ροής του E2E module.

Όταν ένα νέο μήνυμα αποστέλλεται μέσω του E2E, αρχικά το module ελέγχει αν ο προορισμός είναι γείτονας, και εάν είναι τότε στέλνει απευθείας το μήνυμα στον προορισμό. Την συγκεκριμένη πληροφορία την παρέχει το neighbor-discovery(ND) module. Εάν ο κόμβος δεν είναι γείτονας τότε το E2E εξετάζει αν υπάρχει κάποια διαδρομή προς τον προορισμό αποθηκευμένη στην route cache του. Στην περίπτωση που δεν υπάρχει αποθηκευμένη κάποια διαδρομή αποθηκεύει προσωρινά σε έναν εσωτερικό buffer το μήνυμα και στέλνει ένα route request μήνυμα στον cluster-head.

Ο cluster-head μόλις λάβει ένα route request μήνυμα εξετάζει εάν ο προορισμός του μηνύματος είναι κάποιος κόμβος ο οποίος ανήκει στο συγκεκριμένο cluster. Εάν δεν είναι, προωθεί το route request μήνυμα στους γειτονικούς cluster-heads και η συγκεκριμένη διαδικασία συνεχίζεται μέχρι να καλυφθεί όλο το δίκτυο ή να βρεθεί ο κόμβος. Κάθε

κόμβος που προωθεί το route request μήνυμα προσθέτει το ID του στο μήνυμα. Εάν κάποιο cluster-head είναι ο προορισμός ή ο προορισμός είναι κόμβος που ανήκει σε κάποιο cluster τότε αποστέλλεται πίσω ένα route reply μήνυμα στον κόμβο που δημιούργησε το route request μήνυμα (originator), ακολουθώντας το αντίστροφο μονοπάτι (reverse path). Μόλις ένα route reply μήνυμα φτάσει στον originator, τότε ο originator προσθέτει το μονοπάτι στην route cache του και στη συνέχεια ελέγχει τον buffer που διατηρεί για μηνύματα προς αποστολή στο συγκεκριμένο προορισμό. Κάθε διαδρομή στην route cache και κάθε μήνυμα στον buffer έχει κάποιο συγκεκριμένο timeout και ανά τακτά χρονικά διαστήματα διαγράφονται οι παλιές εγγραφές.

Το E2E module παρέχει επίσης την δυνατότητα ανταλλαγής μηνυμάτων ανάμεσα σε απομακρυσμένα δίκτυα. Αυτό επιτυγχάνεται με την χρήση των κινητών κόμβων τα οποία ελέγχονται από το Delay Tolerant Service module(DTS). Εάν ένα μήνυμα παραμένει για περισσότερα από 2 δευτερόλεπτα στον εσωτερικό buffer κάποιου κόμβου τότε σημειώνεται ως υποψήφιο για αποστολή μέσω του DTS και παραδίδεται στον επόμενο κινητό κόμβο που θα βρεθεί κοντά. Η κίνηση που θα ακολουθήσει ο κόμβος είναι ανεξάρτητη από τη λειτουργία παράδοσης μηνυμάτων, η οποία ελέγχεται αποκλειστικά από το E2E module.

Στην επόμενη ενότητα θα παρουσιαστούν οι κινητοί κόμβοι και ο τρόπος λειτουργίας τους.

3.3 Moway Robots και SunSPOTs

Η συνδεσιμότητα των δικτύων αποτελεί σημαντικό σημείο στην end-to-end επικοινωνία καθώς πρέπει διασφαλιστεί η επιτυχής παράδοση των μηνυμάτων στον προορισμό τους. Στη συγκεκριμένη ενότητα θεωρούμε ότι έχουμε δίκτυα ή κόμβους τα οποία δεν μπορούν να επικοινωνήσουν απευθείας μεταξύ τους και χρησιμοποιούνται ρομπότ (Moway [19]) τα οποία συλλέγουν μηνύματα και τα παραδίδουν στον προορισμό τους. Ένα παράδειγμα τέτοιων δικτύων είναι η χρήση μετεωρολογικών σταθμών σε απομακρυσμένες παραλίες. Οι σταθμοί συλλέγουν τα δεδομένα όμως δεν έχουν την δυνατότητα να τα αποστείλουν κεντρικά καθώς δεν είναι συνδεδεμένα στο δίκτυο. Για την συλλογή των δεδομένων υπάρχουν 2 επιλογές: α) Συλλογή από κινητούς κόμβους που επισκέπτονται στοχευμένα τους σταθμούς για να συλλέξουν τα δεδομένα ή β) η αποστολή των δεδομένων σε διερχόμενους κόμβους που βρέθηκαν τυχαία στην εμβέλεια κάποιου σταθμού με σκοπό την παράδοση τους στον προορισμό.

Για την μελέτη της end-to-end επικοινωνία και των κινητών κόμβων χρησιμοποιούμε τα Moway. Τα Moway είναι μικρά ρομπότ, τα οποία μπορούν να προγραμματιστούν μέσω του Moway GUI, το οποίο είναι μια διεπαφή η οποία επιτρέπει στους χρήστες να προγραμματίσουν το Moway με τη χρήση διαγραμμάτων ροής (flow charts) ή χρησιμοποιώντας τις γλώσσες προγραμματισμού C ή Assembly.

Τα Moway χρησιμοποιούνται κυρίως σε πρακτικές εφαρμογές μικρο-ρομποτικής. Διαθέτουν ένα σύστημα οδήγησης και ελέγχου της κίνησης το οποίο μας επιτρέπει:

- Έλεγχο ταχύτητας κίνησης,
- Έλεγχο χρόνου κίνησης,
- Έλεγχο απόσταση κίνησης,
- Έλεγχο γωνία κίνησης.



Σχήμα 3.2: Moway Robot

Διαθέτει επίσης έναν επεξεργαστή στα 4 MHz, τον PIC18F87J50, αισθητήρες ανίχνευσης φωτός, 4 LEDs, έναν αισθητήρα θερμοκρασίας, 4 αισθητήρες αποφυγής εμποδίων και μία υποδοχή επέκτασης με 8 IO PINs εκ των οποίων τα 2 είναι ρεύμα και γείωση.

Το μειονέκτημα που έχουν είναι ότι δεν μπορούν να επικοινωνήσουν ασύρματα μεταξύ τους ή με άλλες συσκευές καθώς επίσης ότι έχουν αδύναμο επεξεργαστή, ελάχιστη μνήμη και πολύ μικρή αυτονομία. Οι συγκεκριμένοι περιορισμοί καθώς επίσης και ο τρόπος προγραμματισμού τους, δυσκολεύουν πολύ την υλοποίηση σύνθετων εφαρμογών. Από την άλλη τα SunSPOTs έχουν πολύ δυνατό επεξεργαστή, μεγάλη χωρητικότητα σε μνήμη, είναι πολύ εύκολα στον προγραμματισμό τους και παρέχουν και ασύρματη επικοινωνία.

Επομένως προσαρμόζουμε ένα SunSPOT πάνω σε κάθε Moway, το οποίο μέσω των IO Pins του συνδέεται στο Expansion Connector του Moway έτσι ώστε να στέλνει εντολές στο σύστημα οδήγησης του Moway ενώ ταυτόχρονα μπορεί να εκτελεί οποιαδήποτε άλλη λειτουργία. Η τοποθέτηση λοιπόν του SunSPOT στην κορυφή του κάθε ρομπότ, επιτρέπει την εκτέλεση σύνθετων αλγορίθμων και καταστεί δυνατή την επικοινωνία με το υπόλοιπο δίκτυο αισθητήρων.

Συγκεκριμένα συνδέουμε 6 από τα I/O Pins του SunSPOT, τα D0-D4 και το H0 στα Pins 3-8 του Moway καθώς επίσης και την γείωση του (GND) του SunSPOT στη γείωση του Moway. Στην εικόνα 3.3 φαίνεται η σύνδεση του SunSPOT με το Moway.

Τα 6 Pins του SunSPOT μπορούν να αναπαραστήσουν 12 διαφορετικές καταστάσεις, οι οποίες αντιστοιχούν σε 12 διαφορετικές κινήσεις οι οποίες θα εκτελούνται από το Moway. Στον πίνακα 3.1, παρουσιάζεται η αντιστοίχιση των καταστάσεων των Pins με τις κινήσεις που θα εκτελεί το Moway.

Το SunSPOT για να χειριστεί το Moway θέτει σε υψηλή κατάσταση τα αντίστοιχα Pins. Το Moway διαβάζει συνεχόμενα τα Pins εισόδου του Expansion Connector και αποφασίζει, σύμφωνα με το διάγραμμα ροής που έχουμε σχεδιάσει, ποια κίνηση πρέπει να εκτελέσει. Με αυτό τον τρόπο οι κινήσεις του ρομπότ καθορίζονται από το SunSPOT, όπου υπάρχει δυνατότητα ασύρματης δικτύωσης καθώς επίσης είναι ευκολότερη η υλοποίηση εφαρμογών. Στην εικόνα 3.4 παρουσιάζεται το διάγραμμα ροής του Moway.

Βασίζόμενοι στα παραπάνω, υλοποιήσαμε 2 ειδών κινήσεις στο SunSPOT, τον απομακρυσμένο έλεγχο (*Remote Control*) και τις τυχαίες διαδρομές βασιζόμενες στο θεωρητικό μοντέλο των (*Random waypoint [4]*):

1. *Remote Control*: Στην συγκεκριμένη περίπτωση, δίνεται η δυνατότητα να κατευθύ-



Σχήμα 3.3: Διασύνδεση SunSPOT με Moway Robot

νουμε απομακρυσμένα το Moway στέλνοντας τις κατάλληλες εντολές στο SunSPOT. Η επικοινωνία επιτυγχάνεται με τη χρήση του 802.15.4 δικτύου. Ο χρήστης έχει την δυνατότητα να ορίσει, όχι μόνο την επιθυμητή κίνηση αλλά καθώς επίσης και την διάρκεια εκτέλεσης της εκάστοτε κίνησης.

2. *Random waypoint model*: Αρχικά, ο αλγόριθμος επιλέγει τυχαία μία από τις 3 δυνατές κινήσεις προς τα εμπρός καθώς επίσης και την χρονική διάρκεια της κίνησης. Η χρονική διάρκεια ορίζεται ανάμεσα σε 2 σταθερές τιμές που δηλώνουν την ελάχιστη και την μέγιστη διάρκεια κάθε κίνησης. Στη συνέχεια επιλέγει τυχαία μία από της αριστερές ή δεξιές στροφές και αφού την εκτελέσει επιλέγει και πάλι μία κίνηση προς τα εμπρός. Η συγκεκριμένη ακολουθία (κίνηση-στροφή-κίνηση..) συνεχίζεται για όσο χρονικό διάστημα έχει ορίσει ο χρήστης. Σε περίπτωση που χτυπήσει σε κάποιο εμπόδιο, αυτομάτως στρίβει είτε δεξιόστροφα είτε αριστερόστροφα κατά 45° , ανάλογα με την θέση που βρίσκεται το εμπόδιο.

Όπως αναφέραμε και στην προηγούμενη ενότητα θέλουμε η κίνηση των Moway να είναι ανεξάρτητη από το E2E module καθώς επίσης και από το DTS. Το συγκεκριμένο module ονομάζεται robot motion module και είναι υπεύθυνο αποκλειστικά για την κίνηση των robot. Στην επόμενη ενότητα θα παρουσιάσουμε το DTS.

3.4 Delay Tolerant Υπηρεσίες

Το DTS module παρέχεται όπως αναφέραμε από το E2E module. Τα Moway αλληλεπιδρούν μέσω του E2E με τους σταθερούς κόμβους του δικτύου και συλλέγουν τα μηνύματα τα οποία δεν μπορούν να παραδοθούν στον προορισμό τους. Διατηρούν τα μηνύματα και μόλις εισέλθουν στην εμβέλεια του δικτύου στο οποίο βρίσκεται ο τελικός προορισμός τα παραδίδουν.

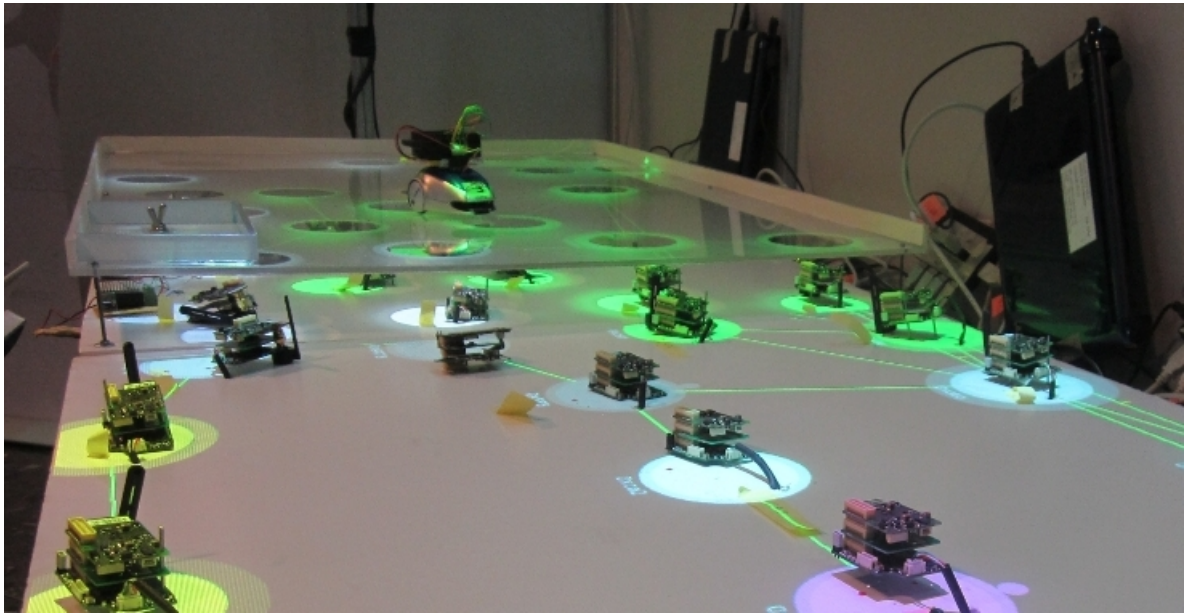
SunSPOT's pins states						Moway Operation
D0	D1	D2	D3	D4	H0	
0	0	0	0	0	0	Stop
1	0	0	1	0	0	Forward
1	0	0	1	1	0	Slow Forward
1	0	0	1	1	1	Fast Forward
0	1	1	0	0	0	Right turn 30°
0	1	1	1	0	0	Right turn 60°
0	1	1	1	1	0	Right turn 90°
0	1	1	1	1	1	Right turn 120°
0	0	1	0	0	0	Left turn 30°
0	0	1	1	0	0	Left turn 60°
0	0	1	1	1	0	Left turn 90°
0	0	1	1	1	1	Left turn 120°

Πίνακας 3.1: Συσχέτιση των καταστάσεων των SunSPOT's pins με τις κινήσεις του Moway.

Η βασική λειτουργικότητα του DTS module υλοποιείται στα SunSPOT όπου έχει υλοποιηθεί και το E2E module. Το E2E module αποφασίζει εάν ένα μήνυμα μπορεί να δρομολογηθεί μέχρι τον προορισμό του ή όχι. Στην περίπτωση που δεν μπορεί να δρομολογηθεί το μήνυμα παραδίδεται στο Moway, συλλέγει τα μηνύματα και τα αποθηκεύει σε έναν εσωτερικό Buffer.

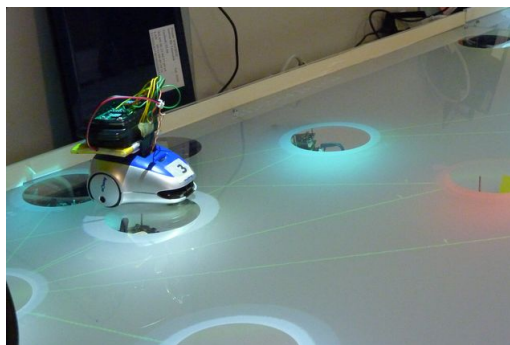
Συγκεκριμένα, μόλις ο DTS παραλάβει ένα μήνυμα από έναν κόμβο του δικτύου, ελέγχει το προορισμό του μηνύματος και το αποθηκεύει σε μια ουρά. Ο DTS διατηρεί μια thread-safe ουρά ανά προορισμό. Οι ουρές έχουν προκαθορισμένο μέγεθος και σε περίπτωση που γεμίσουν αφαιρούνται τα παλαιότερα μηνύματα. Μόλις το Neighbor Discovery module ανακαλύψει έναν νέο γείτονα ειδοποιεί τον DTS. Εάν ο DTS έχει αποθηκευμένα μηνύματα για το συγκεκριμένο κόμβο, προσπαθεί να του τα αποστείλει όλα. Ο DTS ακολουθεί το Observable/Observer design pattern, και ενημερώνεται για τυχόν αλλαγές του Neighbor Discovery module αποφεύγοντας με αυτό τον τρόπο τις συνεχόμενες ερωτήσεις για αλλαγές στο ND. Μόλις ο DTS ξεκινήσει την αποστολή μηνυμάτων προς κάποιον κόμβο ειδοποιεί ταυτόχρονα το Robot Motion module και σταματάει την κίνηση του robot. Μετά την ολοκλήρωση της αποστολής, ο DTS ξαναειδοποιεί το Robot Motion module για να ξεκινήσει και πάλι την κίνηση του. Ο DTS αποθηκεύει επίσης σε ξεχωριστές λίστες τα μηνύματα που παραδόθηκαν επιτυχώς στον προορισμό τους.

Επίσης, μόλις συναντηθούν 2 Moway robots συγχωνεύουν τις ουρές τους που περιέχουν τα μηνύματα που δεν έχουν παραδώσει ακόμη, έτσι ώστε να μειωθεί ο χρόνος παράδοσης. Για να εξασφαλίσουν ότι ένα μήνυμα θα παραδοθεί μόνο μία φορά και ότι δεν μεταφέρουν μηνύματα που έχουν ήδη παραδοθεί τα Moway συγχωνεύουν και τις ουρές που περιέχουν τα μηνύματα που έχουν παραδώσει. Συγκεκριμένα μόλις ο DTS ειδοποιηθεί από το Neighbor Discovery module για την ύπαρξη ενός νέου bidirectional neighbor ο οποίος είναι Moway, ο DTS του ενός Moway προσπαθεί να στείλει τόσο τα μηνύματα που έχουν παραδοθεί όσο και αυτά που δεν έχουν παραδοθεί ακόμη. Ο DTS του Moway που έλαβε τις δύο λίστες απαντάει πίσω μόνο με τα διαφορετικά μηνύματα που περιέχονται στις αποθηκευμένες λίστες του και σε αυτές που μόλις έλαβε. Μόλις



Σχήμα 3.6: Το testbed με το Moway και τους υπόλοιπους κόμβους.

Όπως αναφέρθηκε και προηγουμένως, έχουν τοποθετηθεί 3 Arduino τα οποία επικοινωνούν με το δίκτυο των iSense. Το Arduino που ελέγχει τον ανεμιστήρα υποστηρίζει 3 διαφορετικές λειτουργίες: κλείσιμο ανεμιστήρα, γρήγορη λειτουργία του ανεμιστήρα και αργή. Για τα LEDs έχουμε και πάλι 3 λειτουργίες: σθήσιμο, γρήγορο αναβόσμημα και αργό. Το 3ο Arduino είναι αυτό που ελέγχει τους διακόπτες και το κουμπί. Ο πρώτος διακόπτης καθορίζει ποια συσκευή θα ελέγξουμε, τα LEDs ή τον ανεμιστήρα, ενώ ο δεύτερος καθορίζει την ταχύτητα (αργά ή γρήγορα). Το κουμπί μόλις πατηθεί δημιουργεί μηνύματα τα οποία αποστέλλονται από το Arduino προς το δίκτυο των iSense και δημιουργούνται σύμφωνα με την κατάσταση των διακοπών. Όσο είναι πατημένο το κουμπί υπάρχει μια συνεχόμενη ροή μηνυμάτων προς τους Actuators οι οποίοι εκτελούν την αντίστοιχη ενέργεια. Μόλις αφήσει ο χρήστης το κουμπί διακόπτεται η ροή οπότε σταματάει και η λειτουργία των Actuators (LEDs ή ανεμιστήρας).



Σχήμα 3.7: Τα Moway στο testbed

Υπάρχει επίσης και μία web-based εφαρμογή η οποία ελέγχει το testbed και εκτελείται σε μία συσκευή iPad και παρουσιάζεται στην εικόνα 3.8. Ο χρήστης μπορεί να αλλάξει την τοπολογία του δικτύου, να διακόψει την λειτουργία ή να επανεκκινήσει τους κόμβους

κτλ. Επίσης μπορεί να κατευθύνει τα Moway πάνω στο testbed.



Σχήμα 3.8: Η web-based εφαρμογή του iPad

Τα Moway πραγματοποιούν τυχαίες διαδρομές, βασιζόμενα στο Random waypoint model, πάνω στο testbed ενώ ταυτόχρονα εκτελείται τόσο στα Moway όσο και στα iSense το E2E module. Ο χρήστης τροποποιώντας την τοπολογία του δικτύου και δημιουργεί 3 ασύνδετα δίκτυα τα οποία δεν μπορούν να επικοινωνήσουν μεταξύ τους. Οπότε όταν πατηθεί το κουμπί θα δημιουργηθούν μηνύματα τα οποία λόγω της τροποποιημένης τοπολογίας δεν θα μπορούν να φτάσουν στον προορισμό τους επειδή δεν θα υπάρχει μονοπάτι. Τα Moway όμως, όπως περιγράψαμε παραπάνω, συλλέγουν τα μηνύματα μέσω του E2E module, τα αποθηκεύουν τοπικά και στην συνέχεια είτε εκτελώντας τυχαία κίνηση είτε κατευθυνόμενα από τον iPad τα παραδίδουν στον προορισμό τους. Στη συνέχεια μόλις γίνεται επαναφορά της τοπολογίας τους δικτύου στην αρχική της κατάσταση, δημιουργούνται και πάλι links μεταξύ των τριών υποδικτύων και τα μηνύματα φτάνουν στον προορισμό τους χωρίς την χρήση των Moway. Με αυτόν τον τρόπο επαληθεύουμε και παρατηρούμε σε πραγματικό χρόνο την λειτουργία του DTS module και τις τυχαίες κινήσεις του των Moway.

4

Ανάπτυξη εφαρμογών με το μοντέλο των Web Services

4.1 Εισαγωγή

Στο συγκεκριμένο κεφάλαιο, παρουσιάζουμε ένα ολοκληρωμένο σύστημα, όπου σε κάθε επίπεδο της αρχιτεκτονικής εκτελούνται Web Services μέσω των οποίων παρέχεται η απαραίτητη λειτουργικότητα κάθε επιπέδου. Ο σκοπός της δημιουργίας ενός συστήματος σύμφωνα με την προσέγγιση των Web Services είναι η δημιουργία ενός δικτύου αισθητήρων όπου επιπρόσθετοι κόμβοι μπορούν να εισέρχονται στο δίκτυο μετά από την αρχική τοποθέτηση/εγκατάσταση. Η λειτουργικότητα και τα δεδομένα κάθε κόμβου παρέχονται με δομημένο τρόπο έτσι ώστε να μπορούν να χρησιμοποιηθούν από πολλαπλές εφαρμογές. Το αποτέλεσμα είναι η δημιουργία ενός συστήματος όπου εκτελούνται διάφορες εφαρμογές χρησιμοποιώντας την ίδια υποδομή και τους ίδιους αισθητήρες του δικτύου.

Η κύρια πρόκληση ενός συστήματος αισθητήρων με περιορισμένους υπολογιστικούς πόρους που χρησιμοποιεί web services είναι η επιπρόσθετη κατανάλωση ενέργειας καθώς επίσης και οι αυξημένες απαιτήσεις σε εύρος ζώνης λόγω του δομημένου τρόπου που παρέχονται τα δεδομένα.

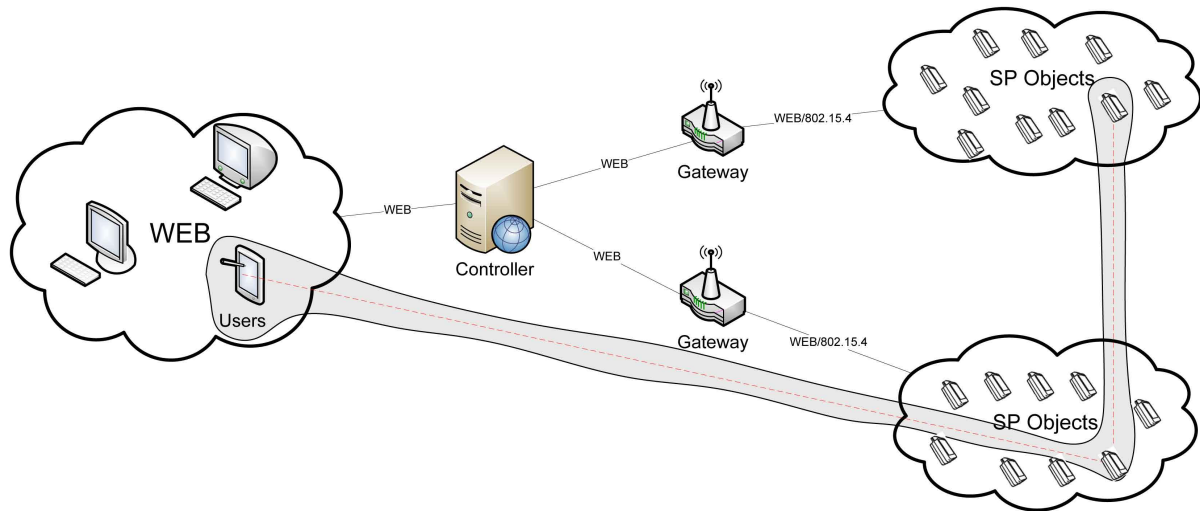
Το συγκεκριμένο σύστημα είναι ανεξάρτητο από τις συσκευές που χρησιμοποιούνται (platform & hardware independent) καθώς οι λειτουργίες και τα δεδομένα παρέχονται με την χρήση web services. Η μοναδική προϋπόθεση για την λειτουργία του συστήματος είναι η χρήση συσκευών που επιτρέπουν την επικοινωνία μέσω του 802.15.4 πρωτοκόλλου.

Στη συνέχεια παρουσιάζεται η αρχιτεκτονική του συγκεκριμένου συστήματος.

4.2 Αρχιτεκτονική

Το σύστημα βασίζεται σε μία ιεραρχία επιπέδων και η αρχιτεκτονική του παρουσιάζεται στην εικόνα 5.1.

Στο χαμηλότερο επίπεδο της αρχιτεκτονική βρίσκονται τα **Small Programmable Ob-**



Σχήμα 4.1: Αρχιτεκτονική του Συστήματος

jects (SPO). Κάθε ασύρματος κόμβος εκτελεί ένα συγκεκριμένο λογισμικό το οποίο προσφέρει πρωτόκολλα επικοινωνίας και εντοπισμού της υποδομής και των υπολοίπων κόμβων που βρίσκονται στο δίκτυο. Επίσης έχει υλοποιηθεί ο *nano HTTP server* ο οποίος χειρίζεται τις HTTP αιτήσεις. Μόλις ο *nano HTTP server* λάβει μια νέα HTTP αίτηση, την επεξεργάζεται και στην συνέχεια καλεί το αντίστοιχο *Senselet*. Κάθε *Senselet* είναι ένα *tiny Web Service* το οποίο υλοποιεί μια συγκεκριμένη λειτουργία.

Το αμέσως επόμενο επίπεδο είναι το **Controller layer**. Το συγκεκριμένο επίπεδο παρέχει τις περισσότερες από τις λειτουργίες του συστήματος και λειτουργεί ως το ενδιαμέσο επίπεδο της αρχιτεκτονικής. Οι λειτουργίες του *Controller layer* παρέχονται και αυτές με την χρήση *Web Services*. Τα υποσυστήματα από τα οποία αποτελείται παρουσιάζονται στην εικόνα 4.2 και αναλύονται στην συνέχεια :

- Ο **wireless sensor controller (WSN Controller)** είναι μια υπηρεσία η οποία είναι υπεύθυνη για τον έλεγχο και την διαχείριση όλων των SPO υποδικτύων. Οι βασικές λειτουργίες του *WSN Controller* είναι έναρξη και ο τερματισμός μιας εφαρμογής σε κάθε κόμβο καθώς επίσης και το 'deploy' μιας νέας εφαρμογής σε κόμβους.
- Το **Routing module** είναι υπεύθυνο για την σύνδεση διαφορετικών SPO υποδικτύων μεταξύ τους, την εγκαθίδρυση εικονικών συνδέσεων (*Virtual Links*) και τοπολογιών ανάμεσα τους και την δρομολόγηση πακέτων στο δίκτυο. Συγκεκριμένα σκοπός του *Routing module* είναι:
 - η διασύνδεση ενός μεγάλου αριθμού από μικρά υποδίκτυα και η δημιουργία ενός ενοποιημένου δικτύου πολύ μεγαλύτερου μεγέθους,
 - η υποστήριξη της επικοινωνίας ανάμεσα σε ετερογενείς συσκευές, διαφορετικού hardware μέσα στο ίδιο δίκτυο,
 - επιτρέπει την γραφική απεικόνιση του φυσικού δικτύου μειώνοντας έτσι την ανάγκη για την επανατοποθέτηση συσκευών.
- Ο **Proxy** είναι υπεύθυνος για την δημοσίευση και την έκθεση των *Senselets* στους *Clients*. Το συγκεκριμένο επιτυγχάνεται με τις ακόλουθες δύο βασικές λειτουργίες:

- Ο βασικός ρόλος του *Proxy* είναι η προώθηση HTTP αιτήσεων από το IP δίκτυο στα χαμηλότερα non-IP δίκτυα αισθητήρων. Κάθε κόμβος καταχωρείται σε ένα συγκεκριμένο gateway. Για να επιτευχθεί επικοινωνία μεταξύ των κόμβων του δικτύου και των υψηλότερων επιπέδων, τα gateways υλοποιούν 2 διεπαφές (interfaces). Η πρώτη παρέχει επικοινωνία με τους κόμβους με την χρήση των Senselets και η δεύτερη προσφέρει επικοινωνία με τα υψηλότερα επίπεδα με την χρήση των Java Web Services.
 - Επίσης το συγκεκριμένο module παρέχει όλες τις πληροφορίες για το σύνολο των gateways και τους καταχωρημένους κόμβους στα υψηλότερα επίπεδα. Κάθε gateway συνδέεται στον *Proxy* έτσι ώστε να καταχωρηθεί. Ο proxy δέχεται ειδοποιήσεις κάθε φορά που κάποιος νέος κόμβος καταχωρείται. Επίσης είναι υπεύθυνος για τον χειρισμό και την προώθηση HTTP αιτήσεων σε συγκεκριμένο gateway ανάλογα με το Senselet που έχει κληθεί.
- Για να ρυθμίσουμε την συνολική αρχιτεκτονική τους συστήματος χρησιμοποιούμε την, *WiseML*¹ [7, 5]. Η *WiseML* είναι βασισμένη στην XML, η οποία είναι μία γλώσσα σήμανσης, που περιέχει ένα σύνολο κανόνων για την ηλεκτρονική κωδικοποίηση κειμένων. Η *WiseML* χρησιμοποιείται για να ορίσει την τοπολογία του δικτύου και για να περιγράψει της πληροφορίες που παρέχει κάθε Senselet .
 - Όσον αφορά την ταυτοποίηση των χρηστών, υπάρχει το *authentication* module στο επίπεδο του Controller layer όπου κάθε client πρέπει να ταυτοποιηθεί πριν συνδεθεί στο σύστημα. Με αυτόν τον τρόπο κάθε χρήστης έχει διαφορετικά δικαιώματα που του επιτρέπουν να χειριστεί συγκεκριμένα Senselets και συγκεκριμένες συναρτήσεις. Επίσης με την χρήση των *virtual private SPO networks* τοπολογιών μπορούμε να ορίσουμε ποιοι πόροι του συστήματος επιτρέπεται να χρησιμοποιηθούν από κάθε εικονικό δίκτυο.

Το επίπεδο του **Client** είναι το υψηλότερο στην ιεραρχία και περιέχει δύο διαφορετικά υποσυστήματα που επιτρέπουν στους χρήστες να χειριστούν το σύνολο των λειτουργιών του συστήματος. Το πρώτο από τα δύο υποσυστήματα είναι το **Web Site**, το οποίο συνδέεται στον proxy, και προσφέρει στο χρήστη ένα ολοκληρωμένο περιβάλλον για την παρακολούθηση και τον έλεγχο του συνολικού συστήματος. Το δεύτερο είναι τα **standalone applications**, τα οποία επιπροσθέτως προσφέρουν απευθείας επικοινωνία με τους αισθητήρες παραλείποντας τα ενδιάμεσα επίπεδα των proxy και gateways. Να σημειωθεί ότι και το SPO έχει την δυνατότητα να λειτουργήσει ως client .

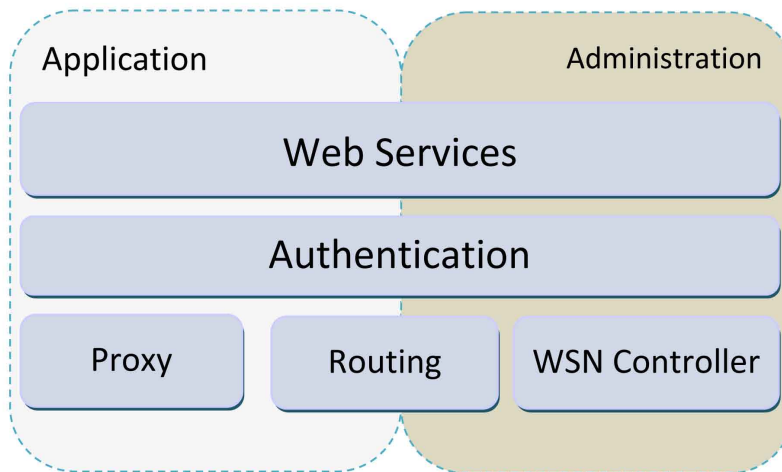
4.3 Ελεγκτής

4.3.1 Ελεγκτής ασυρμάτου δικτύου

Το WSN controller είναι ένα web service το οποίο διαχειρίζεται σε χαμηλό επίπεδο τα φυσικά δίκτυα των SPO. Το API της συγκεκριμένης υπηρεσίας έχει σχεδιαστεί έτσι να υπάρχει σαφής διαχωρισμός ανάμεσα στον Controller του συνολικού συστήματος και σε αυτόν των SPO υποδικτύων (Σχήμα 4.2). Οι πιο βασικές λειτουργίες της συγκεκριμένης υπηρεσίας είναι οι ακόλουθες:

¹WiseML designed and published by European project WISEBED [7, 5]

Controller



Σχήμα 4.2: Controller Stack

- **GETPROPERTYFROMSPO**: Επιστρέφει την τιμή της ζητούμενης ιδιότητας, π.χ, το επίπεδο της μπαταρίας, το ποσοστό της ελεύθερης μνήμης.
- **GETNEIGHBOURHOODOFSP**O: Επιστρέφει τους γειτονικούς κόμβους του συγκεκριμένου SPO.
- **DESCRIBECAPABILITIESOFSP**O: Επιστρέφει το σύνολο των δυνατοτήτων της συγκεκριμένης συσκευής.
- **ARESPOLIVE**: Ενημερώνει αν το συγκεκριμένο SPO λειτουργεί κανονικά.
- **RESETSPO**: Επανεκκινεί την συγκεκριμένη συσκευή.
- **SEND**: Στέλνει ένα μήνυμα από ένα συγκεκριμένο SPO στο Gateway του ή σε κάποιο SPO ενός άλλου υποδικτύου.
- **FLASHSENSELETS**: Εγκαθιστά ένα σύνολο από Senselets σε ένα συγκεκριμένο SPO.
- **GETNETWORK**: Επιστρέφει την τοπολογία του δικτύου.
- **DEFINENETWORK**: Ορίζει την τοπολογία του δικτύου.
- **SETVIRTUALLINK**: Εγκαθιδρύει μία εικονική σύνδεση
- **DESTROYVIRTUALLINK**: Καταργεί μία εικονική σύνδεση ανάμεσα σε δύο SPO.
- **ENABLEPHYSICALLINK**: Ενεργοποιεί την επικοινωνία ανάμεσα σε δύο SPO.
- **DISABLEPHYSICALLINK**: Απενεργοποιεί την επικοινωνία ανάμεσα σε δύο SPO.
- **DISABLENODE**: Ενεργοποιεί ένα SPO.
- **ENABLENODE**: Απενεργοποιεί ένα SPO.

4.3.2 Ρυθμίσεις με χρήση της WiseML

Όσον αφορά την περιγραφή και την αναπαράσταση των δεδομένων στο Internet of Things, μέχρι πρόσφατα υπήρχε έλλειψη ενός καλά τυποποιημένου προτύπου. Οι υπάρχουσες προσεγγίσεις είτε προσφέρουν πολύ περιορισμένες δυνατότητες, όπως η GraphML [9]) είτε είναι πολύ γενικές, όπως η SensorML [30]. Στο ευρωπαϊκό έργο Wisebed [7], έγινε μια συστηματική μελέτη με σκοπό την επίλυση του συγκεκριμένου προβλήματος και είχε ως αποτέλεσμα την WiseML [7]. Η WiseML είναι βασισμένη στην XML και έχει οριστεί έτσι ώστε να μπορεί να αναπαραστήσει τα δεδομένα που συλλέγονται από ασύρματα δίκτυα αισθητήρων. Προσφέρει δυνατότητες αναπαράστασης της τοπολογίας του δικτύου καθώς επίσης και των χαρακτηριστικών των καναλιών επικοινωνίας. Αρχικά, η WiseML παρείχε ένα τυποποιημένο και δομημένο τρόπο αποθήκευσης των πληροφοριών που εξάγονται κατά την εκτέλεση πειραμάτων, όμως είναι αρκετά επεκτάσιμη έτσι μπορεί να ενσωματώσει επιπλέον χαρακτηριστικά. Η χρήση αυτής της μορφής δεδομένων εξελίχθηκε και χρησιμοποιείται για την διευκόλυνση της επικοινωνίας μεταξύ αργαλειών, όπως εξομοιωτές, προγράμματα διαχείρισης πειραμάτων, κτλ, στο τομέα της έρευνας σε ασύρματα δίκτυα αισθητήρων.

Εμείς χρησιμοποιήσαμε την WiseML για την επικοινωνία ανάμεσα στον Controller (4.3) και των SPO gateways (4.4).

Ένα WiseML αρχείο περιέχει όλες της απαραίτητες πληροφορίες που απαιτούνται για να οριστεί σωστά η εγκατάσταση. Απαριθμεί τα στοιχεία του δικτύου, τις δυνατότητες τους, πληροφορίες για το περιβάλλον εγκατάστασης κτλ. Η ενότητα *setup* αποτελείται από τρεις υποενότητες: *installation setup block*, *node related block* και *link related block*. Η πρώτη υποενότητα περιγράφει το σύστημα συντεταγμένων βάση του οποίου έγινε η τοποθέτηση των συσκευών στο χώρο, πληροφορίες σχετικά με το χρόνο που έγινε η συλλογή των δεδομένων από το δίκτυο και πληροφορίες σχετικά με την κινητικότητα των συσκευών στο δίκτυο. Στην δεύτερη υποενότητα εκτός από την απαρίθμηση των διαθέσιμων συσκευών ορίζονται και τα προεπιλεγμένα χαρακτηριστικά που ισχύουν για όλους τους κόμβους του δικτύου καθώς επίσης και περιγράφονται και τα ειδικά χαρακτηριστικά συγκεκριμένων κόμβων. Ως προεπιλεγμένα χαρακτηριστικά ορίζονται: η αρχική θέση τοποθέτησης της συσκευής και τα χαρακτηριστικά της, όπως, αριθμός έκδοσης λογισμικού, εάν είναι gateway, πληροφορίες σχετικές με το hardware κτλ. Επίσης ορίζεται και μια λίστα με τις δυνατότητες της συσκευής (π.χ., *temperature*, *light*, κτλ), οι οποίες χαρακτηρίζονται από τα *name*, *data type*, *unit* και *default* τιμή. Τέλος το *link related block* απαριθμεί όλους τους δυνατούς συνδέσμους επικοινωνίας και ορίζει και τα χαρακτηριστικά του συνδέσμου, πχ εάν χρησιμοποιείται κωδικοποίηση στο κανάλι, εάν ο σύνδεσμος είναι εικονικός κτλ.

Για τα *links* και τους κόμβους, εκτός από την περιγραφή των χαρακτηριστικών και των δυνατοτήτων τους, ορίζεται και ένα μοναδικό αναγνωριστικό έτσι ώστε να είναι δυνατή η πρόσβαση σε όλα αυτά τα αντικείμενα από εξωτερικούς δικτυακούς τόπους μέσω του Controller Layer. Συγκεκριμένα για τους κόμβους ορίζεται το *id* ενώ για τα *links* ορίζονται τα *source* και *target*. Επίσης ορίζονται και τα δικαιώματα και η αδειοδότηση, πχ, ποια Senselets μπορούν να εκτελεστούν και σε ποια SPO, ποιοι Clients έχουν πρόσβαση και σε ποιους πόρους κτλ.

4.3.3 Proxy

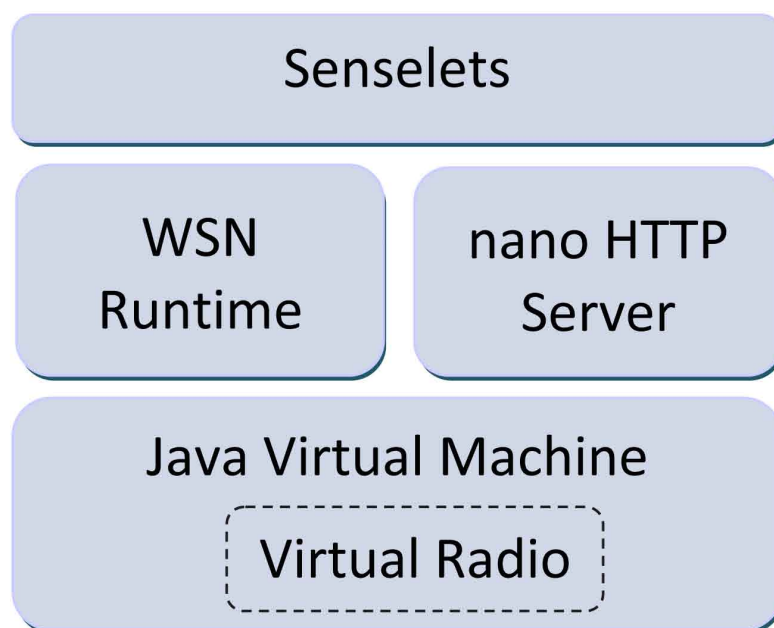
Ο Controller αποθηκεύει τις MAC διευθύνσεις όλων των εγγεγραμμένων SPO καθώς επίσης και μία λίστα με όλα τα διαθέσιμα Senselets. Μόλις ένα gateway λάβει ένα broadcast μήνυμα από ένα SPO καλεί το GETINFO Senselet του συγκεκριμένου SPO, διαβάζει το επιστρεφόμενο XML, εξάγει τα υποστηριζόμενα Senselets και τα αποθηκεύει τοπικά. Στη συνέχεια, ειδοποιεί τον Proxy και του στέλνει την ενημερωμένη λίστα με τα Senselets.. Ο Proxy μόλις λάβει μια HTTP αίτηση για ένα συγκεκριμένο Senselet ενός SPO προωθεί το αίτημα στο αντίστοιχο SPO μέσω του Gateway. Το σύστημα υποστηρίζει και multihop επικοινωνία με την χρήση ενός ad-hoc routing πρωτοκόλλου του AODV [11].

Ο Proxy αποθηκεύει και αυτός όλες τις MAC διευθύνσεις των εγγεγραμμένων SPO καθώς επίσης και τα υποστηριζόμενα Senselets και τις MAC διευθύνσεις των gateway στα οποία είναι εγγεγραμμένα τα SPO. Ο Proxy μόλις δεχθεί μία Senselet αίτηση για ένα συγκεκριμένο SPO από τα κατώτερα επίπεδα, προωθεί στο αντίστοιχο gateway την αίτηση και εκεί μέσω του πρωτοκόλλου δρομολόγησης η αίτηση φτάνει στο ζητούμενο SPO. Οι HTTP αιτήσεις εξυπηρετούνται από τον HTTP server στον οποίο εκτελείται και ο Proxy.

4.4 Small Programmable Objects

Τα Small Programmable Objects αποτελούνται από τα εξής μέρη: Nano HTTP server, Senselets, Discovery protocol, το Virtual Radio Routing Protocol και το WSN Runtime. Η αρχιτεκτονική των SPO παρουσιάζεται στο σχήμα 4.3.

SP Object



Σχήμα 4.3: SP Object Stack

4.4.1 Nano HTTP server

Ο Nano HTTP server είναι ένας server ο οποίος είναι σε θέση να λαμβάνει HTTP αιτήσεις με την χρήση Radiograms. Τα Radiograms είναι μια επέκταση των Datagrams μέσω των οποίων επιτυγχάνεται η αποστολή και λήψη μηνυμάτων ανάμεσα στα SPOs. Μόλις ο Nano HTTP server λάβει ένα νέο Radiogram, δημιουργείται ένα καινούριο Thread το οποίο εξυπηρετεί το συγκεκριμένο αίτημα. Στη συνέχεια γίνεται ανάλυση του αιτήματος και προώθεται το συγκεκριμένο αίτημα στον SenseletManager ο οποίος καλεί το αντίστοιχο Senselet. Μόλις ολοκληρωθεί η εκτέλεση του Senselet, ο Nano HTTP server αποστέλει ένα μήνυμα στον εκάστοτε client το οποίο είναι σε μορφή XML.

4.4.2 Senselets

Τα Senselets υλοποιούν τα Web Services στα SPO. Ο σκοπός του SENSELETMANAGER είναι η προσθήκη και η εκτέλεση ενός συγκεκριμένου Senselet. Η κλήση της ADD μεθόδου στο SenseletManager είναι υποχρεωτική γιατί μέσω της συγκεκριμένης μεθόδου γίνεται η ενεργοποίηση του Senselet. Κάθε Senselet βασίζεται στο abstract Senselet το οποίο έχει υλοποιηθεί και παρέχει της εξής λειτουργίες:

- DOGET
- INIT
- CONFIG
- GETSENSELETINFO

Κάθε νέο Senselet είναι μία επέκταση του abstract. Η μέθοδος doGet καλείται από τον SenseletManager έτσι ώστε να εκτελεστεί ένα συγκεκριμένο Senselet. Οι INIT και CONFIG είναι υπεύθυνες για την αρχικοποίηση και την ρύθμιση των Senselets. Η GETSENSELETINFO επιστρέφει πληροφορίες σχετικά με τις λειτουργίες που προσφέρει το κάθε Senselet.

Έχει υλοποιηθεί ένα σύνολο από Senselets το οποίο καλύπτει τις βασικές λειτουργίες ενός SPO. Το βασικό Senselet σε κάθε SPO είναι το GETINFO το οποίο επιστρέφει ένα XML αρχείο με πληροφορίες σχετικά με την συσκευή και όλα τα εγγεγραμμένα Senselets της συσκευής.

4.4.3 Discovery protocol

Το Discovery protocol χωρίζεται σε δύο μέρη: Broadcaster και το Register Senselet. Ο Broadcaster είναι ένα Java Thread το οποίο είναι υπεύθυνο για την αποστολή μηνυμάτων με την χρήση Radiogram συνδέσεων. Το SPOT για να εντοπίσει ένα Gateway ακολουθεί την ακόλουθη διαδικασία: Ξεκινάει ένα περιορισμένο flooding με συνεχώς αυξανόμενο hop count, με τιμές από 1 μέχρι 15. Μόλις ένα Gateway λάβει ένα broadcast μήνυμα από ένα SPO καλεί το REGISTER Senselet της συγκεκριμένης συσκευής. Το Gateway ανά τακτά χρονικά διαστήματα καλεί το REGISTER Senselet για όλα τα εγγεγραμμένα SPOs για να επιβεβαιώσει της λειτουργία τους. Το REGISTER Senselet αποθηκεύει την χρονική στιγμή κατά την οποία έγινε κλήση του συγκεκριμένου Senselet από κάποιο Gateway και θέτει το Broadcaster Thread σε sleep mode. Στην συνέχεια, ο Broadcaster περιοδικά

ελέγχει το πότε κλήθηκε τελευταία φορά το Register Senselet. Αν η χρονική στιγμή είναι πολύ παλιά τότε θεωρεί ότι έχει χαθεί η επικοινωνία με το Gateway και ξαναρχίζει το Discovery Protocol.

4.4.4 Virtual Radio Routing Protocol

Το ROUTING module (σχήμα 5.1) είναι αυτό που χειρίζεται τις εικονικές τοπολογίες που έχουν εγκαθιδρυθεί ανάμεσα στα φυσικά SPO δίκτυα [29]. Οι συνδέσεις (links) του εικονικού δικτύου μπορούν να εγκαθιδρυθούν με τους εξής τρόπους:

- Ανάμεσα σε κόμβους μέσα στο ίδιο SPO δίκτυο, ορίζοντας έτσι μία προσαρμοσμένη τοπολογία δικτύου.
- Ανάμεσα σε δύο SPO δίκτυα τα οποία είναι σαφώς διαχωρισμένα (λόγο τοποθεσίας, μεγέθους κτλ) δημιουργώντας ένα ενιαίο δίκτυο.
- Μεταξύ συγκεκριμένων κόμβων των δικτύων αυτών, με αποτέλεσμα την δημιουργία ενός μεγαλύτερου δικτύου, με προσαρμοσμένες ακμές ανάλογα με τις εκάστοτε ανάγκες.

Ένα εικονικό link επιτρέπει την επικοινωνία ανάμεσα σε δύο SPO, τα οποία δεν μπορούν να επικοινωνήσουν απευθείας με φυσική σύνδεση, με διαφανή τρόπο. Έτσι η εφαρμογή του χρήστη δεν μπορεί να αντιληφθεί ότι πρόκειται για εικονικό link. Επιπλέον, οι φυσικοί σύνδεσμοι, δηλαδή τα links ανάμεσα σε κόμβους με απόσταση ενός hop, μπορούν επιλεκτικά να απενεργοποιηθούν. Και οι δύο δυνατότητες του Virtual Radio γίνονται με τρόπο εντελώς διαφανή στα ενεργά Senselets.

Συγκεκριμένα, τα virtual links ενεργοποιούνται μέσω ενός ειδικού λογισμικού που είναι εγκατεστημένο σε κάθε SPO κόμβο και ονομάζεται Virtual Radio. Το Virtual Radio διατηρεί ένα πίνακα δρομολόγησης της μορφής <ID, INTERFACE> και έτσι όταν κάποιος κόμβος ζητήσει να αποστείλει ένα μήνυμα σε κάποιον κόμβο, ανάλογα με το ID αποφασίζει σε ποια διεπαφή θα το προωθήσει, είτε στο Physical Radio είτε στο Virtual Radio.

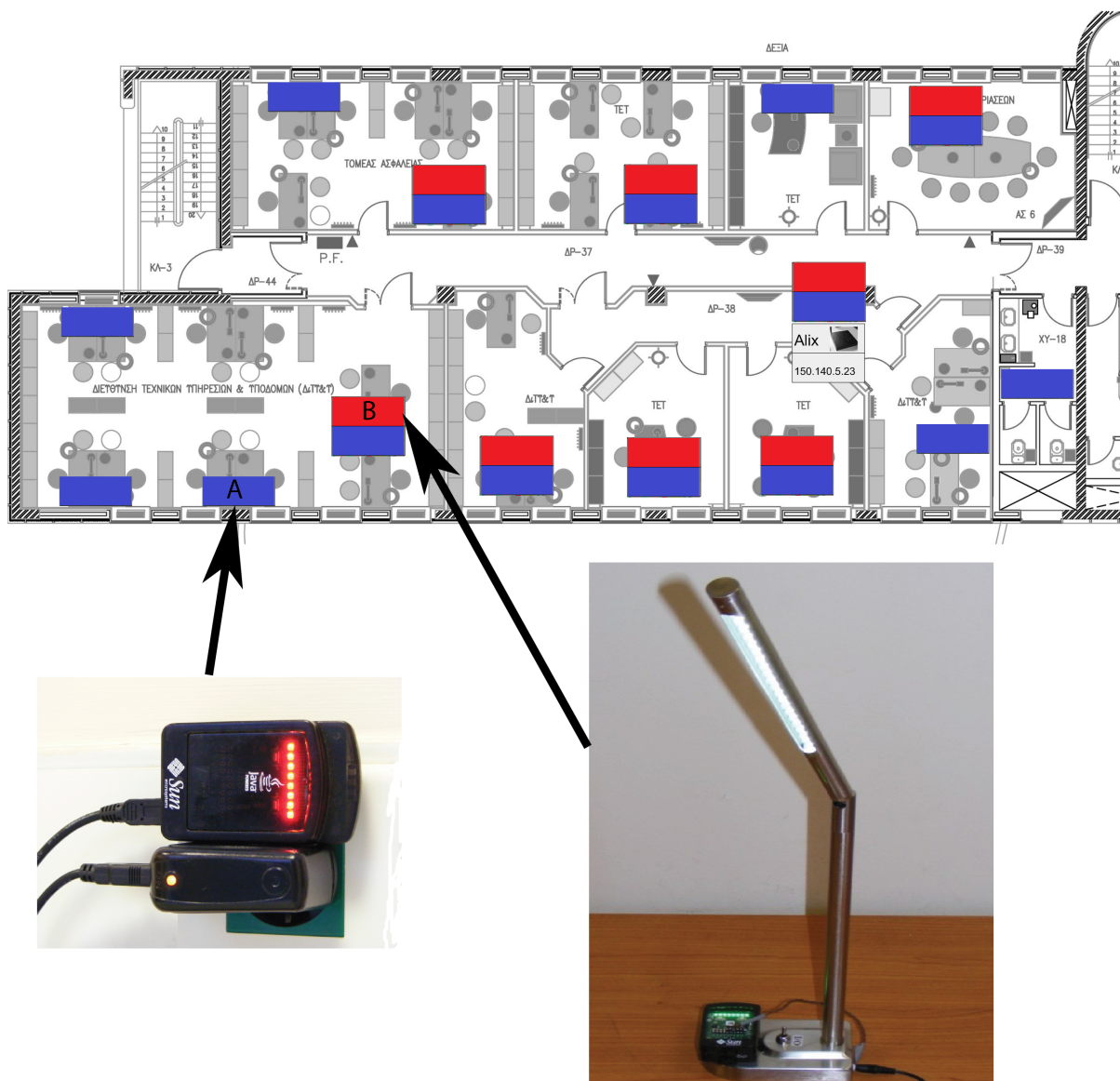
Κατά την εκκίνηση του συστήματος ή όποτε αλλάξει κάποια ρύθμιση, τα IDs του virtual radio ελέγχονται από το Routing Component έτσι ώστε να παραμένουν μοναδικά. Κάθε εικονική τοπολογία ρυθμίζεται από τον εκάστοτε WSN Controller ο οποίος ενημερώνει τα SPO τα οποία βρίσκονται στο υποδίκτυο του. Με αυτόν το τρόπο όποιο μήνυμα αποσταλεί μέσω του virtual radio αρχικά προωθείται στον WSN Controller ο οποίος αποφασίζει την διαδρομή που θα δρομολογήσει το πακέτο έτσι ώστε να φτάσει στον προορισμό του.

4.4.5 WSN Runtime

Το WSN Runtime module είναι το end-point του WSN Controller στο επίπεδο του Controller. Όλες οι λειτουργίες που περιγράφονται στην ενότητα 4.3.1, παρέχονται μέσω του συγκεκριμένου module σε κάθε SPO.

4.5 Υλοποίηση πρωτοτύπου και εγκατάσταση

Για την υλοποίηση του πρωτοτύπου αναπτύξαμε αναπτύξαμε τα ακόλουθα:



Σχήμα 4.4: Σενάριο Χρήσης

- τον Nano HTTP server
- το Controller
- το Discovery protocol
- και το WSN runtime για της συσκευές SunSPOT.

Στα SunSPOT υλοποιήσαμε μια σειρά από Senselets τα οποία δίνουν σε εφαρμογές την δυνατότητα πρόσβασης στις τιμές των αισθητήρων καθώς επίσης και την δυνατότητα ορισμού κανόνων στις συσκευές, π.χ., κλήση ενός Senselet ενός άλλου SunSPOT όταν προκληθεί ένα συγκεκριμένο συμβάν.

Για gateways χρησιμοποιήσαμε τα ALIX PC [12], τα οποία είναι συνδεδεμένα μέσω του τοπικού δικτύου σε ένα κεντρικό υπολογιστή όπου και εκτελείται ο Proxy και ο Controller.

Ο Proxy και ο Controller έχουν υλοποιηθεί σε J2SE 1.6 ενώ το Web Site αναπτύχθηκε με την τεχνολογία JavaServer Pages (JSP).

Επίσης έχει υλοποιηθεί και μία εφαρμόγηση σε κινητό τηλέφωνο, συγκεκριμένα χρησιμοποιήθηκε το κινητό OpenMoko Neo FreeRunner [13] με λειτουργικό Android [14]. Στην συγκεκριμένη εφαρμογή υλοποιήθηκαν όλες οι λειτουργίες του Controller έτσι ώστε η κινητή συσκευή να έχει άμεση επικοινωνία με το δίκτυο.

Για την απόδειξη της λειτουργικότητας της αρχιτεκτονικής μας παρουσιάζουμε στην συνέχεια 2 αντιπροσωπευτικά σενάρια χρήσης που απεικονίζουν τον τρόπο με τον οποίο μπορεί να χρησιμοποιηθεί το σύστημα μας σε έξυπνα κτίρια.

4.5.1 Σενάριο Χρήσης: Αυτοματισμοί γραφείου

Στο συγκεκριμένο σενάριο (Εικόνα 4.4) υπάρχει ένας actuator ο οποίος έχει εγκατασταθεί σέ έναν από τους αισθητήρες ενός γραφείου σε ένα κτίριο, και ελέγχει τα φώτα στο συγκεκριμένο γραφείο.

Ο χρήστης ο οποίος θέλει να διασυνδέσει το συγκεκριμένο actuator στο σύστημα μας πρέπει να ακολουθήσει τα ακόλουθα βήματα:

1. Αρχικά Υλοποιεί ένα Senselet το οποίο μας επιτρέπει να χρησιμοποιήσουμε τον actuator που έχει συνδεθεί στο SunSPOT B
2. Στη συνέχεια συνδέεται στον Controller όπου πρέπει να αποκτήσει δικαιώματα χρήσης της συγκεκριμένης υπηρεσίας και στην συνέχεια συνδέεται στο σύστημα
3. Μετά από την επιτυχή σύνδεση, ενημερώνει τα wiseml entries με πληροφορίες για το συγκεκριμένο Senselet
4. Ο Controller διαβάζει τις νέες πληροφορίες και εγκαθιστά το Senselet στο συγκεκριμένο SunSPOT
5. Το Senselet μόλις ξεκινήσει να εκτελείται συνδέεται στον Proxy όπου και καταγράφεται.
6. Τώρα ο χρήστης και οποιοδήποτε άλλο SunSPOT το οποίο είναι στο δίκτυο έχει πρόσβαση και μπορεί να χρησιμοποιήσει το συγκεκριμένο Senselet μέσω http calls για να ελέγχει τα φώτα.

7. Ο χρήστης δημιουργεί έναν εικονικό σύνδεσμο μεταξύ του Sensor A το οποίο μετράει την φωτεινότητα στο χώρο και του SunSPOT B το οποίο ελέγχει τα φώτα.
8. Στη συνέχεια ο χρήστης ορίζει την επιθυμητή φωτεινότητα του χώρου. Μόλις η φωτεινότητα ξεπεράσει τα επιθυμητά επίπεδα το SunSPOT με τον αισθητήρα φωτεινότητας συνδέεται στο SunSPOT B ανοίγει ή σβήνει το φως αναλόγως,

Με αυτόν τον τρόπο το SunSPOT B δέχεται input από το SunSPOT A, και βάση των ορίων που έχει θέσει ο χρήστης τα φώτα ανάβουν ή σβήνουν αυτόματα.

4.5.2 Σενάριο Χρήσης: Απομακρυσμένη Παρακολούθηση Κτιρίων

Στο συγκεκριμένο σενάριο ο χρήστης επιθυμεί να παρακολουθεί την θερμοκρασία κάθε γραφείου στο κτίριο απομακρυσμένα, μέσω του κινητού τηλεφώνου του.

1. Χρησιμοποιώντας το Phone application ο χρήστης συνδέεται στον Controller όπου γίνεται η ταυτοποίηση του χρήστη και στην συνέχεια συνδέεται στο σύστημα.
2. Ο χρήστης ανακτά μία λίστα με όλους τους εγγεγραμμένους αισθητήρες από τον Proxy
3. Στην συνέχεια η εφαρμογή στέλνει απευθείας HTTP GET αιτήσεις σε όλους τους εγγεγραμμένους αισθητήρες. Η αίτηση είναι της ακόλουθης μορφής:
`http://Proxy/<sensor_ID>/temperatureSenselet` και επιστρέφει την θερμοκρασία του συγκεκριμένου αισθητήρα.
4. Η εφαρμογή εμφανίζει τα δεδομένα που έχει ανακτήσει από το δίκτυο και ο χρήστης είναι ενήμερος για την θερμοκρασία σε κάθε χώρο.

5

Ανάπτυξη εφαρμογών με το μοντέλο των Distributed Objects

5.1 Εισαγωγή

Στο παρόν κεφάλαιο μελετάται η ανάπτυξη εφαρμογών σύμφωνα με το μοντέλο των Distributed Objects [10]. Στη συνέχεια παρουσιάζεται ο σχεδιασμός και η υλοποίηση μιας βιβλιοθήκης η οποία μπορεί να χρησιμοποιηθεί για τον έλεγχο ετερογενών συσκευών προσφέροντας APIs για την εύκολη ανάπτυξη εφαρμογών.

Με τον όρο Distributed Objects αναφερόμαστε συνήθως σε software modules τα οποία έχουν σχεδιαστεί έτσι ώστε να λειτουργούν μαζί, αλλά είτε εκτελούνται σε διαφορετικούς υπολογιστές και επικοινωνούν μεταξύ τους μέσω δικτύου, είτε εκτελούνται σε διαφορετικές διεργασίες στον ίδιο υπολογιστή. Η αρχή λειτουργίας τους βασίζεται στην ανταλλαγή μηνυμάτων και επί τις ουσίας ένα object στέλνει μήνυμα σε κάποιο άλλο object με σκοπό την εκτέλεση κάποιας συγκεκριμένης εργασίας. Με την ολοκλήρωση της εργασίας τα αποτελέσματα αποστέλλονται πίσω στο object που έστειλε το αρχικό μήνυμα. Ο όρος Distributed Objects χρησιμοποιείται επίσης και στα ασύρματα δίκτυα αισθητήρων καθώς οι εφαρμογές που αναπτύσσονται είναι κατά κόρον κατανεμημένες.

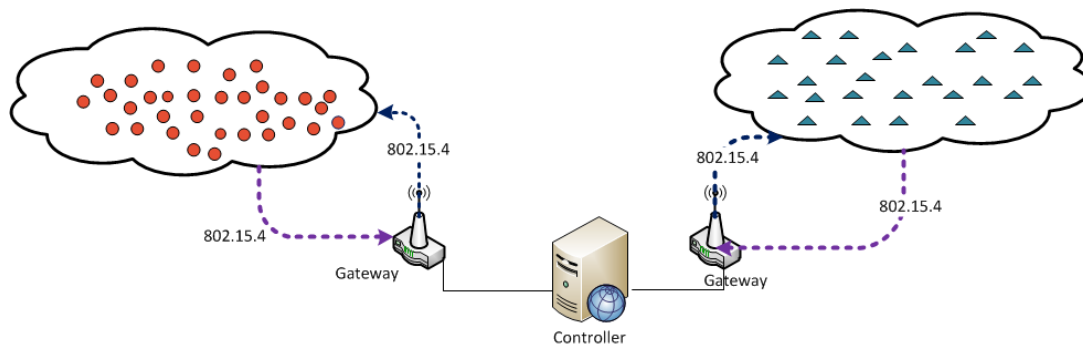
Τα ασύρματα δίκτυα αισθητήρων τα τελευταία χρόνια έχουν παρουσιάσει ραγδαία ανάπτυξη και χρήση. Οι περισσότεροι από τους κόμβους των συγκεκριμένων δικτύων λειτουργούν ως Smart Objects και η λειτουργικότητα τους μπορεί να χρησιμοποιηθεί μέσω του Web of Things. Αυτό που παρατηρήσαμε είναι ότι υπάρχει υποδομή από κόμβους αλλά λείπουν οι εφαρμογές των τελικών χρηστών που θα μπορέσουν να χρησιμοποιήσουν τις λειτουργίες των συγκεκριμένων δικτύων. Τα βασικά προβλήματα τα οποία οδηγούν στην έλλειψη πληθώρας εφαρμογών είναι:

- Ανομοιογένεια Hardware, Software,
- Περιορισμένη επικοινωνία ανάμεσα σε ετερογενείς συσκευές

Στο πρώτο μέρος της συγκεκριμένη διπλωματική, (κεφάλαιο 2), μελετήσαμε τους λόγους στους οποίους οφείλεται η περιορισμένη επικοινωνία λόγω Hardware και προτείναμε συγκεκριμένες λύσεις. Στο υπόλοιπο του κεφαλαίου θα παρουσιάσουμε τον σχεδιασμό και την υλοποίηση του κατάλληλου software το οποίο μπορεί να χρησιμοποιηθεί για την ανάπτυξη εφαρμογών για το Future Internet.

5.2 Αρχιτεκτονική και Υπηρεσίες

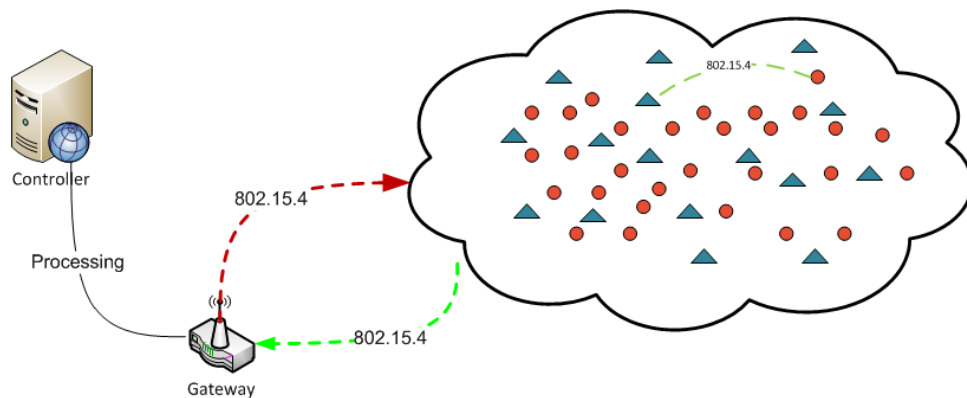
Υπάρχουν δύο κατηγορίες από Smart Objects, τα παθητικά (passive), που ‘αισθάνονται’ και παρατηρούν το περιβάλλον, και τα ενεργητικά (active), που λαμβάνουν μεμονωμένες αποφάσεις και στην ενεργούν αντιστοίχως. Οι δύο αυτές συμπεριφορές απαιτούν διαφορετικές συσκευές για να υλοποιηθούν, κάθε μία από τις οποίες χρησιμοποιεί διαφορετικές τεχνολογίες. Μέχρι στιγμής η αρχιτεκτονική ενός συστήματος με ετερογενείς συσκευές είναι όπως αυτή που παρουσιάζεται στην εικόνα 5.1.



Σχήμα 5.1: Αρχιτεκτονική υπαρχόντων συστημάτων ετερογενών δικτύων

Όπως φαίνεται και στην εικόνα 5.1, η αρχιτεκτονική αποτελείται από δύο ή περισσότερα υποδίκτυα, ανάλογα με τον αριθμό των διαφορετικών συσκευών που υπάρχουν στο σύστημα, τα οποία είναι απομονωμένα μεταξύ τους και δεν υπάρχει απευθείας επικοινωνία μεταξύ τους. Για την επικοινωνία μεταξύ των διαφορετικών υποδικτύων απαιτείται ένα gateway για κάθε υποδίκτυο το οποίο θα δρομολογεί τα μηνύματα στον κεντρικό Controller όπου συλλέγονται και επεξεργάζονται τα δεδομένα και στην συνέχεια προωθούνται στο επιθυμητό υποδίκτυο μέσω του αντίστοιχου gateway. Παρόλο που στην συγκεκριμένη αρχιτεκτονική η παραπάνω διαδικασία γίνεται με διαφανή τρόπο ως προς τον χρήστη, παρουσιάζονται αρκετοί περιορισμοί. Πρώτα από όλα απαιτείται ένα gateway για κάθε υποδίκτυο που αποτελείται από διαφορετικό hardware έτσι ώστε να μπορούν να δρομολογηθούν μηνύματα από τον Controller προς το υποδίκτυο. Επιπροσθέτως για κάθε νέο τύπο συσκευής που προστίθεται στο σύστημα θα πρέπει να αναπτύσσεται καινούριο λογισμικό έτσι ώστε να μπορεί να συνδεθεί στο υπάρχον σύστημα είτε ως active Smart Object είτε ως passive Smart Object. Το πιο αξιοσημείωτο, ωστόσο, πρόβλημα, είναι ο περιορισμός και η δυσκολία στον προγραμματισμό ενός τόσο περίπλοκου συστήματος όπως αυτό, γεγονός που καθιστά δύσκολο το σχεδιασμό, την ανάπτυξη και την επέκτασή του.

Λόγω των ανωτέρω περιορισμών, προτείνουμε μία νέα αρχιτεκτονική όπως αυτή που παρουσιάζεται στην εικόνα 5.2. Η βασική ιδέα πίσω από την συγκεκριμένη αρχιτεκτονική, είναι η δημιουργία ενός συστήματος από active και passive Smart Objects ανεξαρτήτως



Σχήμα 5.2: Προτεινόμενη αρχιτεκτονική

τεχνολογίας. Έχοντας λύσει τα προβλήματα της απευθείας επικοινωνίας μεταξύ ετερογενών συσκευών, όπως παρουσιάσαμε στο κεφάλαιο 2, λύνουμε και το πρόβλημα των πολλαπλών gateways, χρησιμοποιώντας μόνο έναν. Επίσης αναπτύσσουμε το λογισμικό του Controller το οποίο παρέχει απλοποιημένα APIs που επιτρέπουν τον προγραμματισμό και τον έλεγχο ενός ετερογενούς δικτύου.

Οι βασικές υπηρεσίες που παρέχονται είναι οι ακόλουθες:

- **Registration:** Smart Objects εγγράφονται στο σύστημα παρέχοντας την MAC διεύθυνση τους.
- **Data collection:** ο Controller μπορεί απευθείας να επικοινωνήσει με τους αισθητήρες των Smart Objects ζητώντας συγκεκριμένες τιμές.
- **Service request:** ο Controller ενεργοποιεί το σύστημα ειδοποιήσεων έτσι ώστε να ειδοποιείται μόλις συμβεί κάποιο συγκεκριμένο συμβάν στο δίκτυο, π.χ., αύξηση της θερμοκρασίας πάνω από κάποιο συγκεκριμένο όριο.
- **Action request:** ο Controller ζητάει από κάποια συσκευή να εκτελέσει μια συγκεκριμένη ενέργεια.

Στη συνέχεια μελετάται ξεχωριστά η αρχιτεκτονική και η υλοποίηση του Controller και των Smart Objects. Οι συσκευές στις οποίες υλοποιήθηκε η προτεινόμενη αρχιτεκτονική είναι τα SunSPOT και Arduino αλλά μπορεί πολύ εύκολα να επεκταθεί και σε άλλες.

5.2.1 Αρχιτεκτονική του Controller

Ο Controller εκτελείται κεντρικά σε κάποιο υπολογιστή και είναι υπεύθυνος για τον χειρισμό και τον έλεγχο του δικτύου και των Smart Objects. Η αρχιτεκτονική του Controller αποτελείται από μία ιεραρχία επιπέδων όπως παρουσιάζεται στην εικόνα 5.3.

Το Registration Layer παρέχει την λίστα με όλα τα εγγεγραμμένα Smart Objects του δικτύου. Για την εγγραφή ενός νέου Smart Object ο χρήστης απλά προσθέτει τη νέα συσκευή εισάγοντας στο σύστημα την MAC address του. Στο συγκεκριμένο επίπεδο έχει υλοποιηθεί και το Discovery Service το οποίο με το που ειδοποιηθεί για την νέα συσκευή επικοινωνεί μαζί της και συλλέγει όλα τα απαραίτητα δεδομένα έτσι ώστε ο χρήστης να γνωρίζει τις υπηρεσίες που μπορεί να προσφέρει.



Σχήμα 5.3: Αρχιτεκτονική του Controller

Μόλις προστεθεί μια συσκευή στο σύστημα και αφού ολοκληρωθεί επιτυχώς η εγγραφή της, ο χρήστης μέσω του Service Layer μπορεί να την χρησιμοποιήσει. Στο Service Layer έχουν υλοποιηθεί όλες οι κοινές και βασικές λειτουργίες που μπορούν να παρέχουν τα Smart Objects στα οποία υλοποιήθηκε το σύστημα. Συγκεκριμένα, παρέχει μια διεπαφή (Interface) μέσω της οποίας μπορούμε να χειριστούμε τα ακόλουθα :

- LEDs
- Αισθητήρα Θερμοκρασίας
- Αισθητήρα Φωτεινότητας
- Έλεγχο κατάστασης κουμπιών
- Έλεγχο κατάστασης Input/Output Pins

Η συγκεκριμένη διεπαφή υποστηρίζει 3 διαφορετικές λειτουργίες:

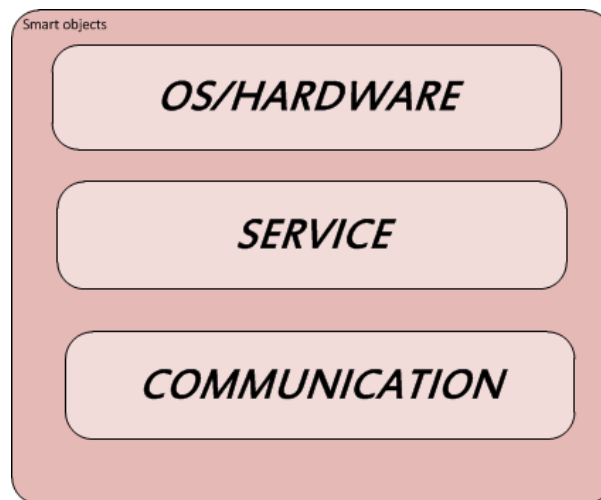
1. *Data request*: Στη συγκεκριμένη περίπτωση ο Controller επικοινωνεί με κάποια συσκευή και της ζητάει την τελευταία τιμή από κάποιον αισθητήρα, π.χ., ο Controller ζητάει την τιμή του αισθητήρα θερμοκρασίας από μία συσκευή.
2. *Service request*: Ο Controller θέλει να ειδοποιείται για συγκεκριμένα συμβάντα που συμβαίνουν στο δίκτυο. Στην ουσία πρόκειται για μία Publish - subscribe υπηρεσία. π.χ, ο Controller ζητάει να ειδοποιείται μόλις πατηθεί κάποιο από τα κουμπιά ενός Smart Object ή μόλις η φωτεινότητα ξεπεράσει το όριο το οποίο έχει θέσει.

3. *Action request*: Η συγκεκριμένη λειτουργία παρέχει την δυνατότητα στον Controller να ζητήσει από ένα Smart Object να εκτελέσει κάποια ενέργεια, π.χ. Να ανάψει ένα φως. Χρησιμοποιείται κυρίως για τον έλεγχο των I/O Pins τα οποία χειρίζονται το Relay το οποίο ανοιγοκλείνει τα φώτα.

Το τελευταίο επίπεδο είναι το Communication Layer το οποίο είναι υπεύθυνο για επικοινωνία με τα Smart Objects και προωθεί μηνύματα από τα ανώτερα επίπεδα στο δίκτυο και αντίστροφα. Κάτω από το συγκεκριμένο επίπεδο βρίσκεται μόνο το λειτουργικό του συστήματος. Η υλοποίηση έγινε εξόλοκληρου σε Java η οποία είναι Platform Independent και μπορεί να εκτελεστεί σε οποιοδήποτε λειτουργικό.

5.2.2 Αρχιτεκτονική των Smart Objects

Στα Smart Objects υλοποιούμε το λογισμικό το οποίο είναι υπεύθυνο για την επικοινωνία με τον Controller. Η αρχιτεκτονική του παρουσιάζεται στην εικόνα 5.4. Όλες οι συσκευές παρέχουν κοινές λειτουργίες εκτός από τις περιπτώσεις που λόγω Hardware δεν μπορούν να τις υποστηρίξουν, π.χ., έλλειψη αισθητήρα θερμοκρασίας. Στην συγκεκριμένη ειδοποιείται μέσω ενός Exception ότι η συσκευή δεν προσφέρει την συγκεκριμένη υπηρεσία.



Σχήμα 5.4: Αρχιτεκτονική των Smart Objects

Κάθε συσκευή δέχεται αιτήσεις στο Communication Layer και τις προωθεί στο Service Layer. Το Service Layer παρέχει όλα τα resources της συσκευής που μπορούν να χρησιμοποιηθούν στο Controller.

Το λογισμικό έχει υλοποιηθεί για κάθε μία από τις συσκευές με τέτοιο τρόπο ώστε να παρέχει τις ίδιες υπηρεσίες ανεξάρτητα με την συσκευή. Εκμεταλλευόμενοι τα interfaces που υλοποιήσαμε προσφέρουμε στον τελικό χρήστη ένα διαφανή τρόπο χρήσης των υπηρεσιών. Ο χρήστης απλά δηλώνει τον τύπο της συσκευής και στην συνέχεια χρησιμοποιεί κοινά interfaces. Με αυτό τον τρόπο μπορεί ανά πάσα στιγμή να αντικαταστήσει μία συσκευή με κάποια άλλη χωρίς να χρειάζεται να αλλάξει ολόκληρη την υλοποίηση της εφαρμογής του.

Στην συνέχεια παρουσιάζεται το interface του Service Layer του Controller:

- `public synchronized void addSPOT(String address)`
Η συγκεκριμένη συνάρτηση προσθέτει ένα καινούριο SunSPOT στο σύστημα. Το μόνο που δέχεται σαν είσοδο είναι η MAC address του.
- `public synchronized void addArduino(String address)`
Η αντίστοιχη συνάρτηση για το Arduino.
- `public synchronized SunSPOT getSPOT(String address)`
Η συγκεκριμένη συνάρτηση επιστρέφει το αντικείμενο του SunSPOT με την συγκεκριμένη διεύθυνση.
- `public synchronized Arduino getArduino(String address)`
Η συγκεκριμένη συνάρτηση επιστρέφει το αντικείμενο του Arduino με την συγκεκριμένη διεύθυνση.

Σύμφωνα με την υλοποίηση που έχει γίνει οποιαδήποτε συσκευή είτε είναι SunSPOT είτε είναι Arduino 'επεκτείνει' την κλάση Smart Object που έχει υλοποιηθεί κληρονομώντας τις λειτουργίες της. Με αυτό τον τρόπο μπορούμε να υλοποιήσουμε την βιβλιοθήκη σε οποιαδήποτε συσκευή και θα είμαστε σε θέση να την χρησιμοποιήσουμε αρκεί να δημιουργήσουμε μια αντίστοιχη με τις SunSPOT, Arduino κλάση.

5.3 Σενάρια χρήσης

Για να αναδείξουμε τις δυνατότητες της βιβλιοθήκης μας παρουσιάζουμε ένα σενάριο χρήσης μέσω του οποίου επεξηγείται ο τρόπος με τον οποίο μπορεί να χρησιμοποιηθεί για την δημιουργία 'έξυπνων εφαρμογών' στο Future Internet.

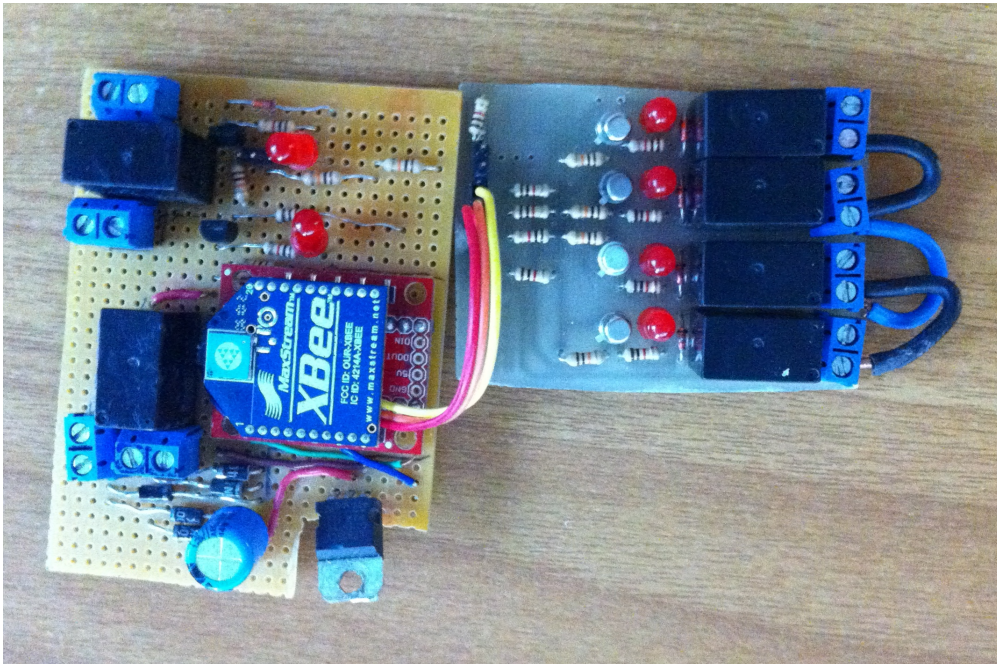
5.3.1 Αυτοματισμοί Γραφείων

Το συγκεκριμένο σενάριο χρήσης αφορά τους αυτοματισμούς σε εργασιακούς χώρους. Σκοπός της εφαρμογής είναι ο έλεγχος του φωτισμού στον χώρο σε περίπτωση που η φωτεινότητα είναι χαμηλή και υπάρχει παρουσία ανθρώπων στο χώρο ή σε περίπτωση που κάποιος κάθεται σε κάποιο γραφείο. Η λογική της εφαρμογής είναι η εξής:

Εάν η φωτεινότητα στο χώρο είναι κάτω από κάποιο συγκεκριμένο όριο, τότε ανάβουν όλα τα φώτα στα γραφεία τα οποία κάθεται κάποιος. Εάν επίσης η φωτεινότητα είναι χαμηλή και υπάρχει κάποιος μέσα στο χώρο, τότε ανάβουν κάποια από τα φώτα έτσι ώστε να είναι ομοιόμορφα φωτισμένη η αίθουσα.

Στην συγκεκριμένη εφαρμογή χρησιμοποιούνται 2 διαφορετικές συσκευές, SunSPOT και Arduino. Τα SunSPOT, λόγω των μεγαλύτερων υπολογιστικών δυνατοτήτων τους χρησιμοποιούνται για την να παίρνουν γρήγορα τις αποφάσεις και να για να ειδοποιούν την κεντρική εφαρμογή, ενώ τα Arduino για να ελέγχουμε την παρουσία ανθρώπων στο χώρο με την χρήση PIR αισθητήρων και την παρουσία ανθρώπων σε κάποιο γραφείο με την χρήση αισθητήρων proximity στις οθόνες. Καμία από τις 2 συσκευές δεν είχε αισθητήρες PIR και proximity. Στα Arduino όμως είναι πολύ εύκολο να προσθέσουμε αισθητήρες και γιαυτό το λόγο προτιμήθηκαν σε σχέση με τα SunSPOT.

Στην εικόνα 5.5



Σχήμα 5.5: Το Arduino με τους αισθητήρες που ανοιγοκλείνει τα φώτα

5.3.2 Εφαρμογή Ψυχαγωγίας

Στο συγκεκριμένο σενάριο χρησιμοποιούμε τα SunSPOT και τα Arduino για να δημιουργήσουμε μια διαδραστική εφαρμογή ψυχαγωγίας. Η συγκεκριμένη εφαρμογή χρησιμοποιεί διαδραστικούς κουβάδες με χρώματα οι οποίοι μέσα τους έχουν Arduino και διαδραστικά πινέλα τα οποία κατασκευάστηκαν από SunSPOT και παρουσιάζονται στην εικόνα 5.6.

Κάθε κουβάς έχει ένα συγκεκριμένο χρώμα και τα LEDs που βρίσκονται μέσα στον κουβά φωτίζουν στο αντίστοιχο χρώμα. Ο παίχτης βουτάει το πινέλο που κρατάει στο χέρι του μέσα στον κουβά με το χρώμα που θέλει να πάρει. Μόλις πλησιάσει το πινέλο κοντά στον κουβά το πινέλο φωτίζει στο αντίστοιχο χρώμα. Στην συνέχεια κάνει μια απότομη κίνηση με το πινέλο και 'πετάει' στον καμβά ο οποίος είναι μια επιφάνεια προβολής όπου και εμφανίζεται το χρώμα. Χρησιμοποιώντας το σύστημα που αναφέραμε προηγουμένως, η υλοποίηση της εφαρμογής ήταν αρκετά εύκολη. Στον Controller όπου και εκτελείται η λογική της εφαρμογής ορίστηκε πως θέλει να ειδοποιείται μόλις οι τιμές του αζελερόμετρου των SunSPOT(πινέλων) ξεπεράσουν το όριο που έχει θέσει. Με αυτόν τον τρόπο μπορούμε να καταλάβουμε πότε ο παίχτης κάνει την κίνηση για να πετάξει το χρώμα. Εκείνη την στιγμή SunSPOT ειδοποιεί τον Controller για την κίνηση καθώς και το χρώμα που έχουν εκείνη τη στιγμή τα LEDs του. Στα Arduino που βρίσκονται μέσα στους κουβάδες ο Controller έχει ορίσει το χρώμα των LEDs τους. Τα Arduino εκπέμπουν επίσης με πολύ χαμηλή εμβέλεια κάθε 500ms ένα πακέτο μέσα στο οποίο περιέχεται το χρώμα των LEDs τους. Ο Controller επίσης έχει στείλει ένα Action Request στα SunSPOT με το οποίο τα SunSPOT μόλις λάβουν ένα πακέτο από κάποιον κουβά ανάβουν και αυτά τα LEDs τους στο αντίστοιχο χρώμα.



Σχήμα 5.6: Διαδραστική εφαρμογή ψυχαγωγίας με πινέλα και κουβάδες χρωμάτων

5.4 Συμπεράσματα

Στο συγκεκριμένο κεφάλαιο παρουσιάσαμε μια βιβλιοθήκη η οποία μπορεί να χρησιμοποιηθεί για την ανάπτυξη 'έξυπνων εφαρμογών' στο Future Internet. Σκοπός της είναι η παροχή απλών διεπαφών έτσι ώστε να μπορεί ο χρήστης να ορίσει εύκολα κανόνες και να δημιουργήσει εφαρμογές ανεξαρτήτως συσκευής. Η βιβλιοθήκη απαντά όλα τα προβλήματα και τα θέματα της επικοινωνίας και του κατανεμημένου υπολογισμού που απαιτείται για την ανάπτυξη εφαρμογών σε ασύρματα δίκτυα αισθητήρων. Επιτρέπει τον κεντρικοποιημένο χειρισμό των συσκευών με διαφανή τρόπο στον χρήστη. Επίσης έχει υλοποιηθεί με τέτοιο τρόπο ώστε να μπορεί εύκολα να επεκταθεί και σε νέες συσκευές με περισσότερες λειτουργίες και δυνατότητες.

6

Συμπεράσματα

Σκοπός της παρούσας διπλωματικής εργασίας ήταν η εγκατάσταση και λειτουργία ολοκληρωμένων εφαρμογών διάχυτου υπολογισμού με χρήση ασυρμάτων ετερογενών συσκευών, αισθητήρων και ελεγκτών. Το βασικό πρόβλημα που υπήρχε στα ετερογενή ασύρματα δίκτυα αισθητήρων ήταν η αδυναμία επικοινωνίας συσκευών διαφορετικού hardware μεταξύ τους παρόλο που χρησιμοποιούν το ίδιο μέσο και το ίδιο πρωτόκολλο επικοινωνίας, το 802.15.4. Έπειτα από την μελέτη τεσσάρων διαφορετικών συσκευών, των SunSPOT, iSense, TelosB και Arduino, παρατηρήθηκε ότι σε κάθε μία από τις συσκευές υπάρχει μερική υλοποίηση του πρωτοκόλλου και οδηγεί σε ασυμβατότητες. Επιλέχθηκαν οι συγκεκριμένες συσκευές καθώς είναι οι συνηθέστερες που χρησιμοποιούνται από την ερευνητική κοινότητα. Για κάθε συσκευή μετρήσαμε τις δυνατότητες των συσκευών όσο αφορά την ταχύτητα αποστολής και λήψης και έπειτα από την πειραματική αξιολόγηση καταλήξαμε στο συμπέρασμα ότι τα iSense είναι η πιο σταθερή συσκευή με τους ταχύτερους ρυθμούς αποστολής και λήψης.

Πέρα από την ασυμβατότητα λόγω του hardware ασχοληθήκαμε και με την δυνατότητα επικοινωνίας από άκρο σε άκρο σε δίκτυα ασυρμάτων αισθητήρων. Στα WSN η τοπολογία του δικτύου αλλάζει συνεχώς και δημιουργεί προβλήματα στην μετάδοση και στην παράδοση μηνυμάτων. Προτείνουμε ένα πακέτο λογισμικού το οποίο παρέχει end-to-end επικοινωνία και υπηρεσίες delay-tolerant συνδεσιμότητας. Ο συνδυασμός του λογισμικού με την επίλυση των προβλημάτων επικοινωνίας λόγω του 802.15.4 επιτρέπει την δημιουργία ετερογενών συστημάτων διάχυτου υπολογισμού με χρήση ασυρμάτων δικτύων αισθητήρων.

Η ανάπτυξη λογισμικού όμως για τέτοια συστήματα παρουσιάζει πολλές προκλήσεις καθώς ο προγραμματισμός των συσκευών απαιτεί την χρήση διαφορετικών τεχνολογιών και διαφορετικών μοντέλων αρχιτεκτονικής. Η προσπάθεια απόκρυψης της ύπαρξης διαφορετικών συστημάτων και η παροχή ενός απλοποιημένου φαινομενικά συστήματος στους τελικούς χρήστες υλοποιεί κατά βάση την ιδέα ενός διάχυτου υπερσυστήματος. Στο υπόλοιπο της διπλωματικής παρουσιάσαμε δύο διαφορετικά συστήματα που αναπτύξαμε για την προγραμματισμό τέτοιων συστημάτων, ένα βασισμένο στο μοντέλο των Web Services

και ένα με το μοντέλο των Distributed Objects προσφέροντας διαφορετικές δυνατότητες.

Για κάθε ένα από τα μοντέλα που χρησιμοποιήθηκαν αναπτύχθηκαν και διαφορετικά σενάρια χρήσης αναδεικνύοντας τις δυνατότητες των συστημάτων. Συγκεκριμένα αναπτύχθηκαν οι εξής εφαρμογές:

1. End to End Communication module και πειραματική αξιολόγηση στο Portable Testbed
2. Αυτοματισμοί κτιρίων σύμφωνα με το μοντέλο των Web Services
3. Απομακρυσμένη παρακολούθηση κτιρίων σύμφωνα με το μοντέλο των Web Services
4. Εφαρμογή ψυχαγωγίας σύμφωνα με το μοντέλο των Distributed Objects

Βιβλιογραφία

- [1] Internet of Things: http://en.wikipedia.org/wiki/Internet_of_Things
- [2] Web of Things: http://en.wikipedia.org/wiki/Web_of_Things
- [3] Smart Objects: http://en.wikipedia.org/wiki/Smart_objects
- [4] Mobile Computing, editors, Tomasz Imielinski and Hank Korth, volume 353, Kluwer Academic Publishers, 1996.
- [5] Chatzigiannakis, I., Fischer, S., Koninis, C., Mylonas, G. and Pfisterer, D.. WISEBED: an Open Large-Scale Wireless Sensor Network Testbed. In Proc. 1st ICST International Conference on Sensor Networks Applications, Experimentation and Logistics (SENSAPPEAL 2009), Athens, Greece, September 24-25, 2009. Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, (Springer-Verlag).
- [6] Nissanka B. Priyantha, Aman Kansal, Michel Goraczko and Feng Zhao *Tiny Web Services: Design and Implementation of Interoperable and Evolvable Sensor Networks*, ACM/IEEE Conference on Embedded Networked Sensor Systems
- [7] Wisebed: <http://www.wisebed.eu/>
- [8] T. Baumgartner, I. Chatzigiannakis, S. Fekete, C. Koninis, A. Kroller, and A. Pyrgelis. Wiselib: A generic algorithm library for heterogeneous sensor networks. In *7th European Conf. on Wireless Sensor Networks*, EWSN '10, pages 162–177.
- [9] The GraphML file format, <http://graphml.graphdrawing.org/>
- [10] Distributed Objects: http://en.wikipedia.org/wiki/Distributed_objects
- [11] Ad hoc On-Demand Distance Vector Routing:
http://en.wikipedia.org/wiki/Ad_hoc_OnDemand_Distance_Vector_Routing
- [12] ALIX system boards: <http://pcengines.ch/alix.htm>
- [13] Openmoko: http://wiki.openmoko.org/wiki/Main_Page
- [14] Android: <http://www.android.com/>
- [15] iSense: <http://www.coalesenses.com/>

- [16] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler: Transmission of IPv6 Packets over IEEE 802.15.4 Networks
- [17] RF Monolithics: <http://www.rfm.com>
- [18] Srinivasan, K., & Levis, P. (2006). "RSSI is Under Appreciated". Power, 2006. Citeseer. Διαθέσιμο στον σύνδεσμο :
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.3630&rep=rep1&type=pdf>
- [19] Moway Robots: <http://www.moway-robot.com/>
- [20] SunSPOT: <http://www.sunspotworld.com/>
- [21] TelosB: <http://www.xbow.com>
- [22] TinyOS: <http://www.tinyos.net>
- [23] TinyOS στην wikipedia:
<http://en.wikipedia.org/wiki/TinyOS>
- [24] IEEE std. 802.15.4 - 2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs):
<http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>
- [25] Arduino: <http://www.arduino.cc/>
- [26] Γλώσσα προγραμματισμού Wiring:
<http://wiring.org.co/>
- [27] Xbee module: <http://www.digi.com/>
- [28] IEEE 802.15 WPAN Task Group 4 (TG4):
<http://www.ieee802.org/15/pub/TG4.html>
- [29] Baumgartner, T., Chatzigiannakis, I., Danckwardt, M., Koninis, C., Kroller, A., Mylonas, G., Pfisterer, D., and Porter, B.: Virtualising Testbeds to Support Large-Scale Reconfigurable Experimental Facilities. In Proc. "7th European Conference on Wireless Sensor Networks (EWSN 2010), Coimbra, Portugal, 2010. Lecture Notes in Computer Science, Volume 5970, pp. 210-223 (Springer-Verlag).
- [30] Botts, M. and Robin, A.. OpenGIS: Sensor Model Language (SensorML) Implementation Specification, 2007.
- [31] Duquennoy, S., Grimaud, G., and Vandewalle, J. 2009. Serving embedded content via web applications: model, design and experimentation. In Proceedings of the Seventh ACM international Conference on Embedded Software (Grenoble, France, October 12 - 16, 2009).

- [32] Oliveira, L. B., Kansal, A., Priyantha, B., Goraczko, M., and Zhao, F. 2009. Secure-TWS: Authenticating node to multi-user communication in shared sensor networks. In *Proceedings of the 2009 international Conference on information Processing in Sensor Networks*.
- [33] Glombitza, N., Pfisterer, D., and Fischer, S. 2009. Integrating wireless sensor networks into web service-based business processes. In *Proceedings of the 4th international Workshop on Middleware Tools, Services and Run-Time Support For Sensor Networks*.
- [34] Orestis Akribopoulos, Georgitzikis Vasileios, Koninis Christos, Papavasileiou Ioannis and Chatzigiannakis Ioannis, Deployment and Evaluation of a 802.15.4 Heterogeneous Network, 2011. Technical Report in ACM Computing Research Repository, arXiv:1102.0058v2, February, 2011.
- [35] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. Dbpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154–165, 2009.
- [36] M. Compton, C. Henson, H. Neuhaus, L. Lefort, and A. Sheth. A survey of the semantic specification of sensors. In *2nd Int. Workshop on Semantic Sensor Networks*.
- [37] O. Corcho and R. García-Castro. Five challenges for the semantic sensor web. *Semantic Web*, 1(1,2):121–125, 2010.
- [38] S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. smap: a simple measurement and actuation profile for physical information. In *8th ACM Conf. on Embedded Networked Sensor Systems, SenSys '10*, pages 197–210.
- [39] P. Hayes, editor. *RDF Semantics*. W3C Recommendation. World Wide Web Consortium, 2004.
- [40] M. Kovatsch, S. Mayer, and B. Ostermaier. Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things. In *6th Int. Conf. on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS '12*.
- [41] L. Lefort, C. Henson, K. Taylor, P. Barnaghi, M. Compton, O. Corcho, R. García Castro, J. Graybeal, A. Herzog, K. Janowicz, H. Neuhaus, A. Nikolov, and K. Page. Semantic sensor network. In L. Lefort, C. Henson, and K. Taylor, editors, *W3C Incubator Group*, number June, page 80. W3C, 2011.

