

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

Τμήμα Μηχανικών Η/Υ και Πληροφορικής

Διπλωματική Εργασία

**Μελέτη, Σχεδιασμός & Ανάπτυξη Υβριδικών
Ασύρματων Δικτύων Αισθητήρων και Ενεργοποιητών**

Βασίλης Δ. Κανελλόπουλος
Α.Μ.: 2803

*Επιβλέπων: Πάυλος Σπυράκης, Καθηγητής
Συνεπιβλέπων: Δρ. Ιωάννης Χατζηγιαννάκης*

Πάτρα, Οκτώβριος 2009

Ευχαριστίες

Ολοκληρώνοντας την παρούσα εργασία, νιώθω την ανάγκη να ευχαριστήσω τους ανθρώπους που ήταν κοντά μου το διάστημα των σπουδών μου.

Αρχικά, θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Παύλο Σπυράκη, καθηγητή στο ΤΜΗΥΠ και διευθυντή του EAITY, για την ευκαιρία που μου έδωσε να γνωρίσω από κοντά και να ασχοληθώ με το αντικείμενο αυτό. Επίσης, θα ήθελα να ευχαριστήσω θερμά τον Δρ. Γιάννη Χατζηγιαννάκη, που ως συνεπιβλέπων με βοήθησε και με στήριξε για την περάτωση της εργασίας αυτής. Επιπλέον, θα ήθελα να τον ευχαριστήσω για την παροχή του απαραίτητου εξοπλισμού, ο οποίος χρειάστηκε για την ολοκλήρωση της παρούσας εργασίας.

Θα ήθελα να ευχαριστήσω, επίσης, τον Δρ. Γιώργο Μυλωνά και τους υπόλοιπους συναδέλφους που ασχολήθηκαν με παρόμοιο αντικείμενο στη διπλωματική τους εργασία, για τις ενδιαφέρουσες συζητήσεις που είχαμε στα πλαίσια των εβδομαδιαίων συναντήσεων μας.

Ένα μεγάλο ευχαριστώ θα ήθελα να δώσω στον κ. Ιωάννη Καλέμη, ερευνητή στο EAITY, για το βιβλιογραφικό υλικό που μου παραχώρησε για την εργασία αυτή.

Δεν θα μπορούσα να μην ευχαριστήσω τους συναδέλφους και φίλους Φραγκίσκο Μαλλιαρό, Νίκο Κύρτση, Χαράλαμπο Χειλά, Παναγιώτη Αθανασούλη και Σταμάτη Δραγουμάνο, πρώτα απ' όλα για την συνεργασία μας σε πολλές από τις υποχρεώσεις της σχολής, αλλά κυρίως για την φιλία που μας ένωσε και ελπίζω να διατηρηθεί για χρόνια.

Τέλος, θα ήθελα να ευχαριστήσω θερμά την οικογένειά μου, τον πατέρα μου Δημήτρη, την μητέρα Ασημίνα και την αδερφή μου Ειρήνη, καθώς και την αγαπημένη μου Κέλλυ που με στήριξαν στην απόφασή μου για συνέχιση των σπουδών μου στο ΤΜΗΥΠ. Σας ευχαριστώ για όλα όσα έχετε κάνει για μένα και ελπίζω κάποια στιγμή να τα ανταποδώσω.

Περιεχόμενα

Κατάλογος σχημάτων	vii
1 Εισαγωγή	3
1.1 Σκοπός και συνεισφορά της εργασίας	5
1.2 Διάρθρωση της εργασίας	6
2 Ασύρματα Δίκτυα Αισθητήρων και Ενεργοποιητών	7
2.1 Φυσικά χαρακτηριστικά των WSNs	8
2.2 Συντονισμός κόμβων	11
2.2.1 Συντονισμός αισθητήρα-ενεργοποιητή	11
2.2.2 Συντονισμός μεταξύ ενεργοποιητών	16
2.3 Πρωτόκολλα για τα WSNs	19
2.3.1 Διάσταση διαχείρισης	20
2.3.2 Διάσταση συντονισμού	20
2.3.3 Διάσταση επικοινωνίας	21
2.4 Ανακεφαλαίωση	23
3 Universal Plug and Play	25
3.1 Τι είναι το UPnP;	25
3.2 Ποιος καθορίζει το UPnP;	26
3.3 Συσκευές, υπηρεσίες και σημεία ελέγχου	27
3.3.1 Συσκευές	27
3.3.2 Υπηρεσίες	27
3.3.3 Σημεία ελέγχου	28

3.4	Πρωτόκολλα που χρησιμοποιούνται στο UPnP	28
3.4.1	Πρωτόκολλα καθορισμένα από το UPnP	29
3.4.2	TCP/IP	29
3.4.3	HTTP, HTTPU και HTTPMU	29
3.4.4	SSDP	29
3.4.5	GENA	30
3.4.6	SOAP	31
3.5	Πώς λειτουργεί το UPnP;	31
3.5.1	Διευθυνσιοδότηση	32
3.5.2	Ανακάλυψη συσκευών	32
3.5.3	Περιγραφή συσκευών	33
3.5.4	Έλεγχος συσκευών	33
3.5.5	Ειδοποίηση γεγονότων	34
3.5.6	Παρουσίαση συσκευών	35
4	UPnP και Ασύρματα Δίκτυα Αισθητήρων	37
4.1	BOSS – Bridge Of SensorS	38
4.2	ISNN – Intelligent Sensor Network Node	40
4.3	Sindrion	41
4.4	WSN σε CE συσκευές	43
4.5	Συμπεράσματα	45
5	picoPnP	47
5.1	Ελεγκτές, αισθητήρες και ενεργοποιητές	48
5.2	Λειτουργία του picoPnP	50
5.2.1	Ανακάλυψη συσκευών	51
5.2.2	Περιγραφή συσκευών	57
5.2.3	Συσχέτιση συσκευών	58
5.2.4	Γεγονότα και ενέργειες	65
5.3	Υλοποίηση συσκευών και υπηρεσιών	67
5.4	Ανακεφαλαίωση	68

6 Συμπεράσματα και Μελλοντικές Προεκτάσεις	71
6.1 Μελλοντικές Προεκτάσεις	72
Βιβλιογραφία	75
A' picoPnP – Τεχνικές λεπτομέρειες	79
A'.1 Screenshots από την εφαρμογή του ελεγκτή	79
A'.2 UML διαγράμματα των βασικών κλάσεων	80
A'.2.1 Συσκευές και Περιγραφή Συσκευής	80
A'.2.2 Υπηρεσίες και Περιγραφή Υπηρεσίας	81
A'.2.3 Διαχειριστής Ανακάλυψης (Discovery Manager)	82
A'.2.4 Διαχειριστής Περιγραφών (Description Manager)	83
A'.2.5 Διαχειριστής Συσχετίσεων (Association Manager)	84
A'.2.6 Διαχειριστής Συμβάντων	85
A'.3 Παραδείγματα υλοποίησης συσκευών και υπηρεσιών	86
A'.3.1 Αισθητήρας	86
A'.3.2 Ενεργοποιητής	88

Κατάλογος σχημάτων

2.1	Η αρχιτεκτονική ενός WSN σε φυσικό επίπεδο.	8
2.2	Ημι-Αυτοματοποιημένη και Αυτοματοποιημένη αρχιτεκτονική συ- ντονισμού κόμβων.	9
2.2.1	Ημι-Αυτοματοποιημένη	9
2.2.2	Αυτοματοποιημένη	9
2.3	Μπλοκ διαγράμματα αισθητήρα και ενεργοποιητή.	10
2.3.1	Αισθητήρας (Sensor)	10
2.3.2	Ενεργοποιητής (Actor)	10
2.4	Εξάλειψη πλεονασμού: ελάχιστο σύνολο ενεργοποιητών.	13
2.5	Επιλογή ενεργοποιητών για την πληροφόρηση συμβάντος.	14
2.5.1	Σύστημα πολλαπλών ενεργοποιητών.	14
2.5.2	Μοναδικός ενεργοποιητής.	14
2.6	Στοιβά πρωτοκόλλων για τα WSN.	20
3.1	Το λογότυπο UPnP.	26
3.2	Στοιβά πρωτοκόλλων που χρησιμοποιεί το UPnP.	28
4.1	Βασική διασύνδεση μέσω ενός BOSS κόμβου.	39
4.2	Η βασική δομή για την εγκαθίδρυση σύνδεσης μεταξύ ενός terminal και ενός Sindrion Transceiver.	42
4.3	Η αλυσίδα εργαλείων που χρησιμοποιείται για την μετατροπή του απλού προγραμματιστικού συμβολισμού σε bytecode.	44
5.1	Το περιεχόμενο ενός εκπεμπόμενου μηνύματος SEARCH. Το πεδίο για τον τύπο μετάδοσης τίθεται στην τιμή BROADCAST.	51

5.2	Το περιεχόμενο μηνύματος ADVERTISE , ως απάντησης σε ένα SEARCH μήνυμα. Το πεδίο για τον τύπο μετάδοσης τίθεται στην τιμή UNICAST .	52
5.4	Αναζήτηση συσκευών από τον ελεγκτή (Search).	53
5.3	Το περιεχόμενο μηνύματος ADVERTISE . Το πεδίο για τον τύπο μετάδοσης τίθεται στην τιμή BROADCAST .	53
5.5	Το περιεχόμενο ενός μηνύματος ADVERTISE_ACK .	54
5.6	Προβολή συσκευής στους ελεγκτές (Advertisement).	55
5.7	Το περιεχόμενο ενός μηνύματος HEARTBEAT .	55
5.8	Το περιεχόμενο ενός μηνύματος BYEBYE .	57
5.9	Τα μηνύματα που ανταλλάσσουν ελεγκτής και συσκευή για την ανάκτηση των υπηρεσιών.	58
5.9.1	Το περιεχόμενο ενός μηνύματος DESC_REQUEST .	58
5.9.2	Το περιεχόμενο ενός μηνύματος DESC_REPLY .	58
5.10	Διαδικασία ανάκτησης περιγραφής υπηρεσιών συσκευής από ελεγκτή.	59
5.11	Διαδικασία ανάθεσης συσχέτισης.	61
5.12	Τα μηνύματα για που ανταλλάσσονται κατά την ανάθεση και ανάκτηση συσχετίσεων.	62
5.12.1	Το περιεχόμενο ενός μηνύματος ASSOC_SET .	62
5.12.2	Το περιεχόμενο ενός μηνύματος ASSOC_SET_ACK .	62
5.12.3	Το περιεχόμενο ενός μηνύματος ASSOC_GET .	62
5.12.4	Το περιεχόμενο ενός μηνύματος ASSOC_REPLY .	62
5.13	Η διαδικασία ανάκτησης συσχετίσεων συσκευής από τον ελεγκτή.	64
5.14	Το περιεχόμενο ενός μηνύματος ASSOC_DEL .	65
5.15	Η διαδικασία διαγραφής συσχέτισης.	65
5.16	Το περιεχόμενο ενός μηνύματος EVENT_REPORT .	66
5.17	Διαδικασία ειδοποίησης γεγονότος για την εκτέλεση ενέργειας.	67
A'.1	Γραφικό περιβάλλον διαχείρισης δικτύου. Προβολή συσκευής.	79
A'.2	Γραφικό περιβάλλον διαχείρισης δικτύου. Δημιουργία συσχέτισης.	80
A'.3	UML Διάγραμμα κλάσεων συσκευής.	81
A'.4	UML Διάγραμμα κλάσεων υπηρεσίας.	82
A'.5	UML διάγραμμα κλάσεων για το Διαχειριστή Ανακάλυψης.	83
A'.6	UML διάγραμμα κλάσεων για το Διαχειριστή Περιγραφών.	83

A'.7 UML διάγραμμα κλάσεων για το Διαχειριστή Συσχετίσεων.	84
A'.8 UML διάγραμμα για την κλάση Association	85
A'.9 UML διάγραμμα κλάσεων για το Διαχειριστή Συμβάντων.	85
A'.10 UML διάγραμμα για την κλάση Event	86

Κεφάλαιο 1

Εισαγωγή

Από την πρώτη εμφάνιση των υπολογιστικών συστημάτων ως σήμερα, η βιομηχανία των υπολογιστών έχει περάσει, με ραγδαίο τρόπο, από διάφορες τεχνολογικές φάσεις. Από τους κεντρικούς υπερ-υπολογιστές (mainframe computers), που υπήρχαν ως την δεκαετία του '80 και όπου πολλοί χρήστες μοιράζονται έναν υπολογιστή, περάσαμε στους προσωπικούς υπολογιστές (PCs), όπου η οικειότητα που δημιουργείται μεταξύ του ανθρώπου και της μηχανής συνεχώς αυξάνεται. Στη συνέχεια, με την εμφάνιση των τεχνολογιών δικτύωσης και τη δυνατότητα δημιουργίας μεγάλων δικτύων υπολογιστών, με αποκορύφωση το Internet, τα υπολογιστικά συστήματα γίνονται όλο και περισσότερο ένα αναπόσπαστο κομμάτι της καθημερινότητας.

Με τη συνεχή ανάπτυξη της τεχνολογίας, όπως στην κατασκευή όλο και μικρότερων ολοκληρωμένων κυκλωμάτων, και την πρόοδο στις τηλεπικοινωνίες, προσφέρεται η δυνατότητα υλοποίησης συσκευών μικρών διαστάσεων, εξοπλισμένων με ισχυρούς μικροεπεξεργαστές. Συσκευές όπως τα κινητά τηλέφωνα, οι υπολογιστές παλάμης κ.ά. παρέχουν πλέον τη δυνατότητα σύνθετων και απαιτητικών υπολογισμών, όπως μεταξύ άλλων την επεξεργασία σημάτων και εικόνας και την ασύρματη δικτύωση. Η συνεχής αυτή πρόοδος οδηγεί σε μια τρίτη φάση την εξέλιξη των υπολογιστικών συστημάτων, την ιδέα των πανταχόθεν υπολογιστών (Ubiquitous Computing ή Ubicomp). Η ιδέα αυτή εμφανίζεται στα τέλη της δεκαετίας '80 από τον Mark Weiser¹ [Wei93] και πλέον αρχίζει να γίνεται πραγματικότητα. Στον κόσμο του ubicomp, οι υπολογιστές αποσύρονται από το προσκήνιο στο παρασκήνιο, όπου εργάζονται και εξυπηρετούν πολλούς ανθρώπους ταυτόχρονα, χωρίς όμως να γίνονται αντιληπτοί.

Τα τελευταία δέκα περίπου χρόνια, αυτή η εξέλιξη της τεχνολογίας οδήγησε και στη δημιουργία συσκευών μικρών διαστάσεων και χαμηλής κατανάλωσης ενέργειας, οι οποίες είναι εξοπλισμένες με ηλεκτρονικά κυκλώματα ικανά να «αισθάνονται» το

¹Ο Mark D. Weiser (1952-1999) εργαζόταν ως ανώτατο στέλεχος στο Xerox Palo Alto Research Center (Xerox PARC) και θεωρείται ο «πατέρας» του Ubiquitous Computing, καθιερώνοντας τον όρο το 1988.

φυσικό περιβάλλον και δύνανται να επικοινωνούν μεταξύ τους ασύρματα. Έτσι άρχισε να γεννάται η ιδέα των ασύρματων δικτύων αισθητήρων (Wireless Sensor Networks – WSN). Τέτοια δίκτυα επιτυγχάνουν την σύνδεση του «ψηφιακού κόσμου» των υπολογιστών με το φυσικό περιβάλλον. Τα δίκτυα αυτά συνιστούν μια ειδική κατηγορία δικτύων υπολογιστών και απαρτίζονται από ένα πλήθος κόμβων, εξοπλισμένων με αισθητήρες για την παρατήρηση του φυσικού περιβάλλοντος (θερμοκρασία, υγρασία, ένταση φωτός, ποιότητα αέρα κ.ά.). Η ιδέα λειτουργίας τους βασίζεται στην συνεργασία των κόμβων για την επίτευξη κάποιας εργασίας, όπου, αντί για τη συγκέντρωση και αποστολή των μετρούμενων δεδομένων ως έχουν, δύνανται, αξιοποιώντας τις επεξεργαστικές τους δυνατότητες, να επεξεργάζονται τα δεδομένα αυτά και να αποστέλλουν μόνο αυτά που είναι απαραίτητα από την εφαρμογή [ACYW02]. Το μεγάλος πλήθος των κόμβων και η συνεργασία τους σε ένα WSN καλύπτουν την μειωμένη επεξεργαστική τους δυνατότητα, υποστηρίζοντας την αξιόπιστη χρήση του δικτύου συνολικά. Λόγω της ενδεχόμενης αναξιόπιστίας τους, εξαιτίας του χαμηλού κόστους κατασκευής, η χρήση περισσότερων κόμβων για την επίτευξη κάποιας εργασίας εξασφαλίζει την ορθή επικοινωνία, την χαμηλότερη κατανάλωση ενέργειας και την ακριβή παρατήρηση ενός φαινομένου.

Οι εφαρμογές που έχουν αναπτυχθεί κατά καιρούς για τα ασύρματα δίκτυα αισθητήρων στοχεύουν κυρίως στην παρατήρηση, στην εξαγωγή δεδομένων και στην ανίχνευση συμβάντων μέσα στο περιβάλλον όπου αυτά αναπτύσσονται. Στις περισσότερες των περιπτώσεων αυτών, οι αποφάσεις για κάποια ενέργεια στο περιβάλλον, ανάλογη της παρατήρησης, γίνεται από τον άνθρωπο ή από κάποιο σύστημα λήψης αποφάσεων (DSS – Decision Support System)[PB07]. Το κενό αυτό έρχονται να καλύψουν τα Ασύρματα Δίκτυα Αισθητήρων και Ενεργοποιητών (Wireless Sensor and Actor Networks – WSAN)². Πρόκειται για ασύρματα δίκτυα, στα οποία εκτός από τους αισθητήρες λαμβάνει μέρος και ένα άλλο σύνολο κόμβων, οι ενεργοποιητές (actuators ή actors). Οι κόμβοι αυτοί είναι ικανοί να δρουν στο περιβάλλον. Συγκεκριμένα, τα WSAN αποτελούνται από ένα σύνολο κόμβων αισθητήρων και ενεργοποιητών, οι οποίοι συνδέονται μεταξύ τους ασύρματα, με σκοπό την επίτευξη καταναεμμένων εργασιών παρατήρησης του φυσικού κόσμου και δράσης σε αυτό[AK04]. Η προσθήκη των ενεργοποιητών στο δίκτυο επιτρέπει την άμεση λήψη αποφάσεων και την εκτέλεση ενεργειών, από τους κόμβους αυτούς, στο περιβάλλον. Αυτό έχει ως αποτέλεσμα τη μείωση της ανάγκης για ανθρώπινη παρέμβαση, η οποία πολλές φορές είναι αργή και επιρρεπής σε σφάλματα, ενώ παράλληλα προάγεται η ανάπτυξη εφαρμογών πραγματικού χρόνου.

Ξεκινώντας από στρατιωτικές και ερευνητικές εφαρμογές, τα WSAN έχουν αρχίσει να εφαρμόζονται σε πάρα πολλούς διαφορετικούς τομείς της ζωής του ανθρώπου, όπως η χρήση τους για έλεγχο και αυτοματισμό στο οικιακό και εργασιακό περιβάλλον, η μεταφορά και μετακίνηση, η παρακολούθηση του φυσικού περιβάλλοντος για την αντιμετώπιση εκτάκτων αναγκών, η παρακολούθηση και ασφάλεια

² Συχνά, στη βιβλιογραφία συναντούνται και με τον όρο SANET – Sensor and Actor NETWORKs

χώρων, στην εκπαίδευση, στην υγεία και σε πολλούς άλλους τομείς.

Η επέκταση αυτή αναβαθμίζει τη θέση των δικτύων αισθητήρων ως αναπόσπαστο κομμάτι του Ubicomp [BLRB05, SVB09] και αναδεικνύει νέους προβληματισμούς και προκλήσεις σχετικά με την διαχείριση τους. Ειδικότερα σε τομείς όπου η παρουσία εξειδικευμένων χρηστών είναι αδύνατη, όπως για παράδειγμα στο οικιακό περιβάλλον, με την παρουσία έξυπνων οικιακών εξοπλισμένων με αισθητήρες [PMG⁺00], η απουσία μιας ενιαίας αρχιτεκτονικής για την υλοποίηση και διαχείριση τους αποτελεί ένα από τα κρισιμότερα σημεία.

1.1 Σκοπός και συνεισφορά της εργασίας

Η ανάπτυξη εφαρμογών για τα WSN, αλλά και για τα WSN, αποτελεί μια επίπονη διαδικασία, η οποία εξαρτάται κατά κύριο λόγο από το πεδίο στο οποίο εφαρμόζονται. Οι πρώτες εφαρμογές που εμφανίστηκαν στόχευαν κυρίως στην υλοποίηση αλγορίθμων και πρωτοκόλλων για την δρομολόγηση, την αποδοτική κατανάλωση ενέργειας κ.τ.λ., σε αισθητήρες που ανήκουν σε ένα δίκτυο. Σε επόμενες εφαρμογές, αρχίζουν να προστίθενται επιπλέον υπηρεσίες και δυνατότητες για την δικτύωση περισσότερων του ενός δικτύων μεταξύ τους. Πλέον, οι εφαρμογές για τα WSN έχουν επεκταθεί αρκετά, καλύπτοντας πολλά διαφορετικά δίκτυα και αρχίζουν να αποκτούν μεγάλο ενδιαφέρον και σημασία.

Σε όλο το φάσμα των προσπαθειών για την υλοποίηση εφαρμογών, κάθε φορά χρησιμοποιούνται διαφορετικές τεχνικές, ανάλογα με τον διαθέσιμο εξοπλισμό και τον στόχο που εξυπηρετούν. Αυτό έχει ως αποτέλεσμα ο προγραμματισμός τέτοιων εφαρμογών να παραμένει δύσκολος, αφού δεν υπάρχει ένας απλός, ενιαίος τρόπος υλοποίησης [CMN07].

Λαμβάνοντας υπόψη την προαναφερθείσα δυσκολία υλοποίησης εφαρμογών, στη παρούσα εργασία προτείνεται μια αρχιτεκτονική, η οποία επιτρέπει την δημιουργία εφαρμογών για WSN και την παροχή των μέσων για τη διαχείριση του δικτύου.

Για το σχεδιασμό και την ανάπτυξη αυτής της αρχιτεκτονικής προηγήθηκε θεωρητική μελέτη πάνω στα WSN, όπου εξετάστηκαν σημαντικά θέματα, όπως ο συντονισμός των κόμβων, τα πρωτόκολλα επικοινωνίας και η ανάθεση εργασιών σε κόμβους του δικτύου, ενώ παράλληλα μελετήθηκε και η αρχιτεκτονική UPnP [UPn08], μία από τις πλέον καθιερωμένες, λόγω της απλότητας χρήσης στη δικτυακή σύνδεση συσκευών. Προηγούμενες εργασίες πάνω στην διασύνδεση δικτύων αισθητήρων και δικτύων που υλοποιούν το UPnP χρησιμοποιήθηκαν για να κατευθύνουν την προσέγγιση που ακολουθήθηκε στην προτεινόμενη αρχιτεκτονική.

Η έλλειψη μιας αρχιτεκτονικής που θα προσφέρει τις απαραίτητες δικτυακές λειτουργίες, χωρίς την ανάγκη πολύπλοκων χειρισμών, και παράλληλα ο τρόπος λειτουργίας του UPnP οδήγησαν στην ιδέα και υλοποίηση μιας αρχιτεκτονικής για

WSAN, η οποία θα παρέχει τις βασικές διαδικασίες σύνδεσης των κόμβων και την οργάνωση του δικτύου, καθώς και την παροχή εργαλείων για τη διαχείρισή του, προσφέροντας ταυτόχρονα την ευελιξία στη συγγραφή απλών προγραμμάτων για τους κόμβους του δικτύου.

1.2 Διάρθρωση της εργασίας

Προκειμένου να διαπραγματευθούμε κατά τον καλύτερο δυνατό τρόπο το αντικείμενο της συγκεκριμένης εργασίας ακολουθήθηκε μια συγκεκριμένη δομή, η οποία αντανακλά τους βασικούς άξονες της συλλογιστικής μας.

Στο Κεφάλαιο 2 μελετώνται τα Ασύρματα Δίκτυα Αισθητήρων και Ενεργοποιητών. Γίνεται μια αναφορά στη φύση των δικτύων αυτών και παρουσιάζονται οι σημαντικότερες προκλήσεις που αναδύονται και που καθορίζουν την υλοποίηση εφαρμογών για αυτά. Η απαίτηση, σε χρονικά κρίσιμες εφαρμογές, για αποκρίσεις πραγματικού χρόνου, ο συντονισμός των κόμβων για τον τρόπο λειτουργίας μεταξύ τους, αποσκοπώντας στην εκτέλεση κάποιας ενέργειας, η αρχιτεκτονική για την οργάνωση των κόμβων, η εξοικονόμηση ενέργειας, είναι μερικά από τα σημαντικά θέματα που εξετάζονται και απασχολούν την υλοποίηση τέτοιων δικτύων.

Στο Κεφάλαιο 3 παρουσιάζεται η αρχιτεκτονική Universal Plug and Play (UPnP), η οποία προσφέρει τη δυνατότητα αόρατης δικτύωσης μεταξύ συσκευών σε IP δίκτυα και την αυτόματη ανακάλυψή τους. Αρχικά γίνεται μια αναφορά στην προσφορά του UPnP στην δικτύωση συσκευών και στον τρόπο καθορισμού του. Έπειτα, περιγράφονται τα δομικά στοιχεία ενός δικτύου UPnP και αναφέρονται τα πρωτόκολλα και ο τρόπος με τον οποίο αυτά χρησιμοποιούνται από την αρχιτεκτονική αυτή. Τέλος, περιγράφεται η λειτουργία του UPnP στις έξι φάσεις που αυτό ορίζει.

Στο Κεφάλαιο 4 παρουσιάζεται η δυνατότητα διασύνδεσης των WSAN με UPnP δίκτυα. Γίνεται μια αναφορά και περιγραφή προϋπαρχουσών εργασιών πάνω σε αυτόν τον τομέα και εξάγονται κάποια συμπεράσματα.

Στο Κεφάλαιο 5 παρουσιάζεται η αρχιτεκτονική picoPnP, η οποία υλοποιήθηκε στα πλαίσια αυτής της εργασίας. Παρουσιάζονται οι συσκευές που συμμετέχουν σε ένα δίκτυο της εν λόγω αρχιτεκτονικής και αναλύονται οι λειτουργίες που παρέχει, όπως η ανακάλυψη συσκευών, η ανάκτηση της περιγραφής τους και η ανάθεση λειτουργιών σε αυτές από τον χρήστη.

Τέλος, στο Κεφάλαιο 6, γίνεται μια σύνοψη των συμπερασμάτων, τα οποία προκύπτουν από αυτήν την εργασία και παρουσιάζονται κάποιες μελλοντικές προεκτάσεις, οι οποίες μπορούν να βελτιώσουν την προτεινόμενη αρχιτεκτονική.

Κεφάλαιο 2

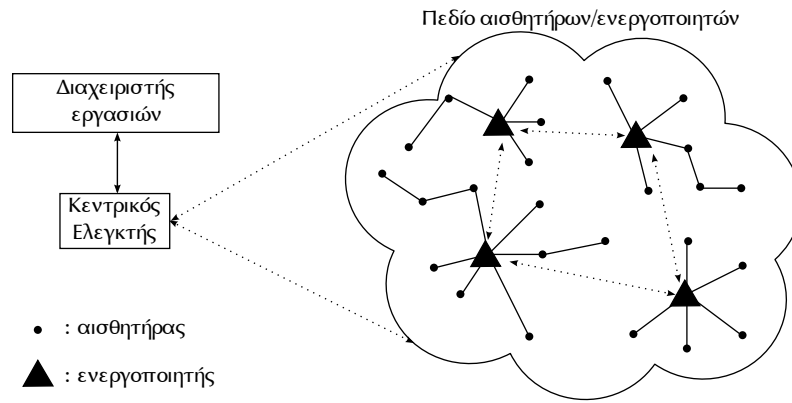
Ασύρματα Δίκτυα Αισθητήρων και Ενεργοποιητών

Τα ασύρματα δίκτυα αισθητήρων και ενεργοποιητών (WSAN – Wireless Sensor and Actor Networks) αναφέρονται σε ασύρματα δίκτυα –όπως μαρτυρά και η ονομασία τους–, τα οποία αποτελούνται από δύο τύπους κόμβων, τους αισθητήρες και τους ενεργοποιητές. Ο σκοπός τους είναι εκτέλεση κατανεμημένων εργασιών, τόσο για την παρατήρηση του φυσικού περιβάλλοντος, από τους αισθητήρες, όσο και για την επιτέλεση ενεργειών σ’ αυτό, από τους ενεργοποιητές.

Μπορούν να χρησιμοποιηθούν σε μια πληθώρα εφαρμογών, όπως σε συστήματα επιτήρησης σε πεδία μάχης, ελέγχου μικροκλίματος και αυτοματισμού σε κτήρια [Bdl⁺06, HMEZ⁺05], καθώς και σε περιβαλλοντικές παρατηρήσεις και αγροκαλλιέργειες [Fer07]. Παραδείγματος χάριν, στην περίπτωση πυρκαγιάς, οι αισθητήρες μπορούν να μεταδώσουν την ακριβή θέση και την ένταση της εστίας στους ενεργοποιητές κατάσβεσης, ώστε η φωτιά να αντιμετωπιστεί άμεσα, χωρίς να επεκταθεί. Αντίστοιχα, οι αισθητήρες κίνησης και φωτός σε ένα δωμάτιο είναι σε θέση να εξακριβώνουν την ύπαρξη κάποιου στο χώρο και στη συνέχεια να ειδοποιούν τους ενεργοποιητές να εκτελέσουν κάποιες προκαθορισμένες ενέργειες, βάσει των επιθυμιών του χρήστη.

Οι αισθητήρες είναι συσκευές χαμηλού κόστους και κατανάλωσης, με περιορισμένες δυνατότητες αίσθησης, υπολογιστικής ισχύος και ασύρματης επικοινωνίας. Οι ενεργοποιητές είναι συνήθως πιο σύνθετες συσκευές, υπολογιστικά δυνατότερες, μεγαλύτερης ισχύος ασύρματης επικοινωνίας και αυτονομίας. Έτσι, το πλήθος των αισθητήρων, που αναπτύσσονται σε μια περιοχή ενδιαφέροντος, είναι συνήθως της τάξης των εκατοντάδων, ή και χιλιάδων ακόμα, ενώ εκείνο των ενεργοποιητών είναι κατά πολύ μικρότερο, λόγω των υπέρτερων δυνατοτήτων τους, που καθιστούν δυνατή τη ευρύτερη δράση τους.

Τα WSANs έχουν δύο μοναδικά χαρακτηριστικά, τα οποία απουσιάζουν στα απλά ασύρματα δίκτυα αισθητήρων (WSN – Wireless Sensor Networks).



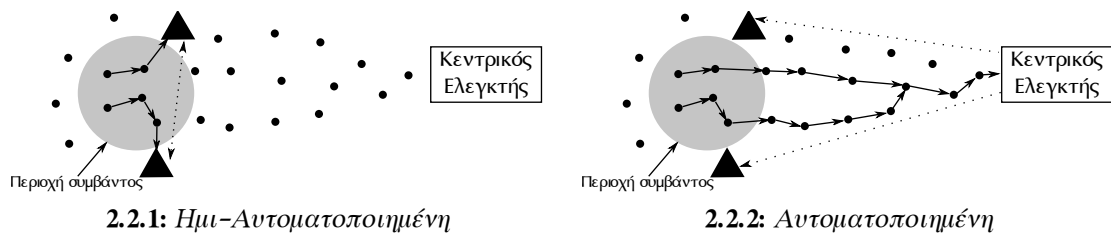
Σχήμα 2.1: Η αρχιτεκτονική ενός WSN σε φυσικό επίπεδο.

1. **Απαιτήσεις για πραγματικό χρόνο:** Σε μερικές εφαρμογές ο χρόνος απόκρισης σε κάποιο συμβάν, που αντιλαμβάνονται οι αισθητήρες, μπορεί να είναι κρίσιμος. Στο προαναφερθέν παράδειγμα της πυρκαγιάς, οι ενέργειες για την αντιμετώπιση της φωτιάς πρέπει να πραγματοποιηθούν άμεσα, μετά την εξακρίβωση της ύπαρξής της. Επιπλέον, τα δεδομένα που συλλέγουν, αλλά και αυτά που αποστέλλουν, οι αισθητήρες πρέπει να είναι έγκυρα κατά τη διάρκεια επιτέλεσης κάποιας ενέργειας από τους ενεργοποιητές.
2. **Συντονισμός:** Σε αντίθεση με τα δίκτυα αισθητήρων (WSN), όπου ο κεντρικός κόμβος (sink) εκτελεί τις εντολές για τα δεδομένα και τον συντονισμό, στα WSN υπάρχουν δύο νέα είδη συντονισμού. Πρώτον, αυτός μεταξύ αισθητήρα και ενεργοποιητή (sensor-actor coordination) και δεύτερον ο συντονισμός μεταξύ των ενεργοποιητών (actor-actor coordination). Συγκεκριμένα, ο πρώτος επιτρέπει την μετάδοση των γεγονότων από τους αισθητήρες στους ενεργοποιητές, ενώ ο τελευταίος αφορά τον συντονισμό μεταξύ των ενεργοποιητών, όταν αυτοί λάβουν πληροφορία για κάποιο γεγονός, ώστε να ληφθεί η καταλληλότερη απόφαση για δράση.

2.1 Φυσικά χαρακτηριστικά των WSNs

Σε ένα WSN, ο ρόλος των αισθητήρων είναι να συλλέγουν δεδομένα από το περιβάλλον, ενώ εκείνος των ενεργοποιητών είναι να δρουν ανάλογα. Έτσι οι κόμβοι διασκορπίζονται στο χώρο, ενώ υπάρχει και ένας κεντρικός κόμβος (sink), που επιτηρεί συνολικά το δίκτυο και επικοινωνεί από τη μια με το διαχειριστή εργασιών και από την άλλη με τους αισθητήρες και τους ενεργοποιητές (Σχήμα 2.1).

Όταν οι αισθητήρες ανιχνεύουν κάποιο φαινόμενο μπορούν είτε να μεταδώσουν τις μετρήσεις τους στους ενεργοποιητές, οι οποίοι επεξεργάζονται τα δεδομένα που λαμβάνουν και δρουν ανάλογα, είτε να δρομολογήσουν τα δεδομένα προς το sink, το οποίο στη συνέχεια ενημερώνει τους ενεργοποιητές με την κατάλληλη ενέργεια



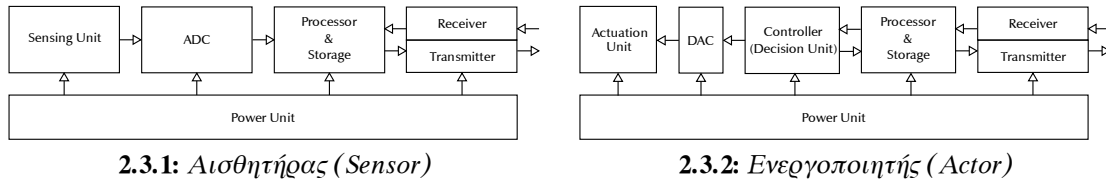
Σχήμα 2.2: Αρχιτεκτονικές για το συντονισμό κόμβων σε WSN. Όπου αισθητήρας και ενεργοποιητής.

που πρέπει αυτοί να εκτελέσουν. Η πρώτη περίπτωση, λόγω της απουσίας ενός κεντρικού ελεγκτή για το συντονισμό, ονομάζεται Αυτοματοποιημένη Αρχιτεκτονική (Automated Architecture), ενώ η δεύτερη Ημι-Αυτοματοποιημένη Αρχιτεκτονική (Semi-Automated Architecture), όπου ο κεντρικός ελεγκτής συλλέγει τα δεδομένα και συντονίζει τη δράση των ενεργοποιητών. Στο Σχήμα 2.2 παρουσιάζονται οι δύο αυτές αρχιτεκτονικές.

Ανάλογα με τον τύπο της εφαρμογής, χρησιμοποιείται μία εκ των δύο Αρχιτεκτονικών. Το πλεονέκτημα της Ημι-Αυτοματοποιημένης Αρχιτεκτονικής είναι ότι μοιάζει κατά πολύ με εκείνη που χρησιμοποιείται στα δίκτυα αισθητήρων [ACYW02], επομένως είναι δυνατή η χρήση των υπάρχοντων αλγορίθμων και πρωτοκόλλων για το συντονισμό και την επικοινωνία.

Βέβαια, η Αυτοματοποιημένη Αρχιτεκτονική παρουσιάζει μεγαλύτερο ενδιαφέρον, εξαιτίας δύο κύριων χαρακτηριστικών της:

1. **Χαμηλή καθυστέρηση:** Τα δεδομένα, που συλλέγονται από τους αισθητήρες, μεταφέρονται απευθείας στους ενεργοποιητές και αφού η απόσταση μεταξύ τους μπορεί να είναι μικρή, η καθυστέρηση ελαχιστοποιείται.
2. **Μακροζωία δικτύου:** Στην Ημι-αυτοματοποιημένη Αρχιτεκτονική τα δεδομένα από τους αισθητήρες, όπως φαίνεται και στο Σχήμα 2.2.1, δρομολογούνται προς τον κεντρικό ελεγκτή, μέσω των κόμβων που απέχουν ένα hop απ' αυτόν. Οι κόμβοι αυτοί επιβαρύνονται σημαντικά με την μετάδοση αυτή και επιπλέον σε περίπτωση αποτυχίας τους, η συνδεσιμότητα με τον κεντρικό ελεγκτή μπορεί να χαθεί, αχρηστεύοντας όλο το δίκτυο. Βέβαια, με τη χρήση τεχνικών συνάθροισης μπορεί να επιτευχθεί η μείωση του φορτίου στους γειτονικούς κόμβους του κεντρικού ελεγκτή, όμως οι τελευταίοι εξακολουθούν να είναι κρίσιμοι για την λειτουργία του δικτύου συνολικά. Από την άλλη, στην Αυτοματοποιημένη Αρχιτεκτονική οι κόμβοι που απέχουν ένα hop από τους ενεργοποιητές πιθανόν να έχουν μεγαλύτερο φορτίο στην εμφάνιση κάποιου γεγονότος (Σχήμα 2.2.2). Βέβαια, στη περίπτωση αυτή είναι πιθανότερη η επικοινωνία με διαφορετικούς ενεργοποιητές, ανάλογα με το γεγονός, το οποίο έχει ως αποτέλεσμα να χρησιμοποιούνται και διαφορετικοί αισθητήρες κάθε φορά για τη μεταφορά των δεδομένων. Συνεπώς, το



Σχήμα 2.3: Μπλοκ διαγράμματα που παρουσιάζουν τις βαθμίδες του αισθητήρα και του ενεργοποιητή.

φορτίο για την μετάδοση των δεδομένων κατανέμεται σε περισσότερους κόμβους, επιτρέποντας τη μεγαλύτερη διάρκεια λειτουργίας τους.

Οι αισθητήρες είναι εξοπλισμένοι με μια μονάδα παροχής ενέργειας, υποσυστήματα επικοινωνίας (πομπός και δέκτης), μια μονάδα για την επεξεργασία και την αποθήκευση δεδομένων, ένα μετατροπέα αναλογικού σήματος σε ψηφιακό και τέλος μια μονάδα αίσθησης. Η τελευταία παρατηρεί το περιβάλλον και αντιλαμβάνεται τα φαινόμενα, όπως την μεταβολή της θερμοκρασίας. Το αναλογικό σήμα, που λαμβάνει ο αισθητήρας από το εξωτερικό περιβάλλον, μετατρέπεται σε ψηφιακό και επεξεργάζεται από την ανάλογη βαθμίδα. Τέλος, το κατάλληλο μήνυμα μεταδίδεται στους γειτονικούς ενεργοποιητές από την βαθμίδα του πομποδέκτη.

Οι ενεργοποιητές διαθέτουν τις βασικές βαθμίδες, όπως και οι αισθητήρες. Δηλαδή, μία μονάδα παροχής ενέργειας, υποσύστημα ασύρματης επικοινωνίας και μονάδα επεξεργασίας και αποθήκευσης δεδομένων. Επίσης, διαθέτει ένα μετατροπέα ψηφιακού σήματος σε αναλογικό, σε αντίθεση με τον μετατροπέα που χρησιμοποιούν οι αισθητήρες, και μία μονάδα ενεργοποίησης, αντί εκείνης της αίσθησης. Επιπλέον, ένας ενεργοποιητής διαθέτει και μία βαθμίδα για τη λήψη αποφάσεων. Η βαθμίδα αυτή αναλύει τα δεδομένα που μεταδίδονται από τους αισθητήρες και εξάγει τις ενέργειες που πρέπει να ακολουθηθούν. Πρόκειται για τις εντολές οι οποίες μετατρέπονται σε αναλογικά σήματα, τα οποία οδηγούν στη συνέχεια την μονάδα ενεργοποίησης.

Τα συστατικά στοιχεία των αισθητήρων και των ενεργοποιητών παρουσιάζονται στο Σχήμα 2.3.

Ανάλογα με την εφαρμογή, μπορεί να υπάρξουν κόμβοι οι οποίοι συνδυάζουν και τους δύο προαναφερθέντες τύπους. Εφόσον ένας τέτοιος κόμβος είναι σε θέση να αισθάνεται, αλλά και να δρα παράλληλα, διαθέτει ό, τι και ένας ενεργοποιητής, επομένως διαθέτει επιπλέον ένα μετατροπέα αναλογικού σε ψηφιακό σήμα και μια μονάδα αίσθησης. Τέτοια παραδείγματα κόμβων αποτελούν τα ρομπότ.

Εκτός από την επικοινωνία μεταξύ αισθητήρων και ενεργοποιητών, σε πολλές περιπτώσεις εφαρμογών η επικοινωνία ανάμεσα στους ενεργοποιητές είναι αναγκαία. Από την στιγμή που οι ενεργοποιητές είναι ισχυρότερες συσκευές, με ευρύτερη ακτίνα μετάδοσης, η επικοινωνία ανάμεσα τους είναι εφικτή σε μεγαλύτερες αποστάσεις, από εκείνη μεταξύ αισθητήρα και ενεργοποιητή.

Επίσης, η επικοινωνία μεταξύ των ενεργοποιητών, λόγω του χαλαρά κατανεμημένου και μικρού πλήθους τους, μοιάζει με εκείνη που εμφανίζεται στα ad-hoc δίκτυα. Επομένως τα WSN μπορούν να θεωρηθούν ως η ένωση των δικτύων αισθητήρων και δικτύων ad-hoc.

2.2 Συντονισμός κόμβων

Η λειτουργία ενός WSN βασίζεται κυρίως στον συντονισμό των κόμβων μεταξύ τους. Συγκεκριμένα, βασίζεται στο συντονισμό ανάμεσα στους αισθητήρες και τους ενεργοποιητές, ο οποίος έχει να κάνει με τις μεταδόσεις των συμβάντων από τους αισθητήρες προς τους ενεργοποιητές, αλλά και στο συντονισμό μεταξύ των ενεργοποιητών, που σχετίζεται με τις ενέργειες που πρέπει να γίνουν από αυτούς, όταν πληροφορηθούν για κάποιο συμβάν.

Στις ακόλουθες δύο ενότητες παρουσιάζονται τα χαρακτηριστικά και κάποια βασικά ζητήματα για το συντονισμό αισθητήρα-ενεργοποιητή και τον αντίστοιχο μεταξύ των ενεργοποιητών.

2.2.1 Συντονισμός αισθητήρα-ενεργοποιητή

Το σημαντικότερο χαρακτηριστικό, που πρέπει να παρέχει η επικοινωνία μεταξύ αισθητήρα και ενεργοποιητή, είναι η χαμηλή καθυστέρηση. Αυτό μπορεί να επιτευχθεί λόγω της εγγύτητας των αισθητήρων και των ενεργοποιητών. Υπάρχουν κάποια βασικά ζητήματα που απασχολούν το συντονισμό αισθητήρα-ενεργοποιητή.

Αρχικά, πρέπει να καθοριστούν οι βασικές απαιτήσεις για την επικοινωνία αισθητήρα-ενεργοποιητή, ανάλογα με την εφαρμογή. Μια από τις απαιτήσεις αυτές, η οποία ισχύει και στα δίκτυα αισθητήρων, είναι η χαμηλή κατανάλωση ενέργειας κατά την επικοινωνία μεταξύ τους. Η απαίτηση αυτή είναι συνήθως ανεξάρτητη της εφαρμογής και παίζει σημαντικό ρόλο για τη μακροζωία του δικτύου. Μια άλλη απαίτηση πιθανόν να είναι η απόκριση του δικτύου σε πραγματικό χρόνο. Όπως στο παράδειγμα της πυρκαγιάς, η καθυστέρηση στην επικοινωνία μεταξύ αισθητήρα και ενεργοποιητή πρέπει να είναι η μικρότερη δυνατή.

Ένα ακόμα ζήτημα, που απασχολεί στην επικοινωνία μεταξύ αισθητήρα και ενεργοποιητή, είναι τήρηση της χρονικής διάταξης των συμβάντων, τα οποία μεταδίδονται στους ενεργοποιητές. Για παράδειγμα, αν δύο αισθητήρες αναφέρουν δύο διαφορετικά συμβάντα σε έναν ή περισσότερους ενεργοποιητές, επικαλυπτόμενων περιοχών, τότε η αναφορά τους πρέπει να γίνει με την σειρά που αυτά συνέβησαν, ώστε οι ενέργειες που θα ακολουθήσουν να πραγματοποιηθούν με την ορθή σειρά.

Επίσης, σημαντικό ρόλο στο συντονισμό των αισθητήρων με τους ενεργοποιητές παίζει το πλήθος των πρώτων, όταν αναφέρουν όλοι ταυτόχρονα ένα γεγονός στους

τελευταίους. Αν η αναφορά ενός συμβάντος φθάνει σχεδόν ταυτόχρονα σε όλους τους εμπλεκόμενους ενεργοποιητές, πρέπει εξασφαλιστεί ότι οι απαιτούμενες ενέργειες θα γίνουν μόνο μια φορά στην ζητούμενη περιοχή.

Ο παραπάνω συγχρονισμός θα μπορούσε να γίνεται ανάμεσα στους ενεργοποιητές. Όμως, υπάρχουν εφαρμογές, στις οποίες είναι απαραίτητο η πληροφόρηση για κάποιο συμβάν να στέλνεται σε ένα σύνολο ενεργοποιητών, οι οποίοι βρίσκονται κοντά στο συμβάν, όταν αυτοί λαμβάνουν την πληροφόρηση και όχι κατά την ανίχνευσή του. Σε τέτοιες περιπτώσεις, οι αισθητήρες πρέπει να είναι σε θέση να παρακολουθούν την εξέλιξη του συμβάντος και να χρησιμοποιούν αυτή την πληροφορία για τον καθορισμό του κατάλληλου συνόλου ενεργοποιητών στο οποίο θα στείλουν τα δεδομένα.

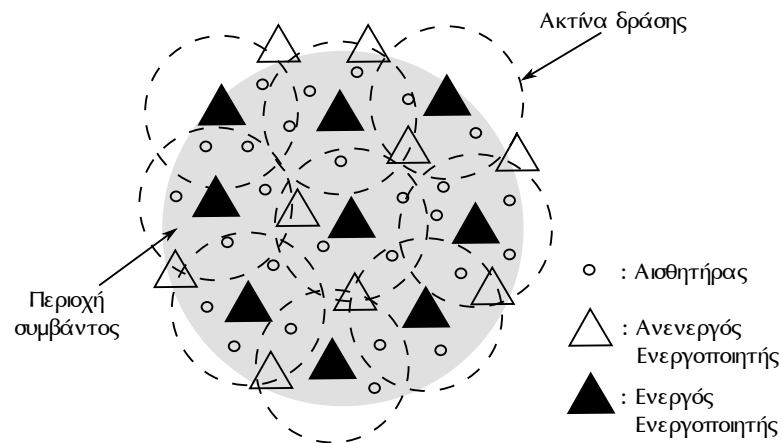
Συνεπώς, ένα πρωτόκολλο για την επικοινωνία αισθητήρα-ενεργοποιητή πρέπει:

- ▶ να παρέχει υπηρεσίες πραγματικού χρόνου, με καθορισμένα όρια καθυστέρησης ανάλογα με την εφαρμογή,
- ▶ να εξασφαλίζει τη χρονική διάταξη των διαφορετικών γεγονότων, όταν αυτά αναφέρονται στους ενεργοποιητές,
- ▶ να εξασφαλίζει την ενεργειακά αποδοτική επικοινωνία,
- ▶ να παρέχει συγχρονισμό ανάμεσα σε αισθητήρες που αναφέρουν το ίδιο γεγονός στους ίδιους ή διαφορετικούς ενεργοποιητές, με σκοπό τη διευκόλυνση δράσης σε όλη τη περιοχή μία μοναδική φορά,
- ▶ να παρακολουθεί την εξέλιξη των παρατηρήσιμων φαινομένων και να την αναφέρει σε διαφορετικό σύνολο ενεργοποιητών, όταν αυτά τα συμβάντα λαμβάνουν χώρα σε διαφορετικά μέρη.

Δύο ζητήματα που προκύπτουν από τα παραπάνω είναι ο καθορισμός του συνόλου των πηγών και προορισμών που συμμετέχουν στη μετάδοση των δεδομένων των αισθητήρων, καθώς και ο τύπος της μετάδοσης.

Για τον καθορισμό των συνόλων αισθητήρων και ενεργοποιητών υπάρχουν τέσσερις εναλλακτικές:

1. Επιλογή του ελάχιστου συνόλου ενεργοποιητών που καλύπτουν την περιοχή του συμβάντος.
2. Επιλογή των ελάχιστων αισθητήρων, οι οποίοι αναφέρουν το συμβάν.
3. Ένας συνδυασμός των δύο προηγούμενων.
4. Επιλογή όλων των αισθητήρων και των ενεργοποιητών που γειτνιάζουν της



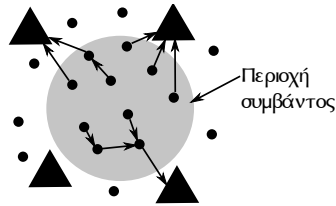
Σχήμα 2.4: Εξάλειψη πλεονασμού: ελάχιστο σύνολο ενεργοποιητών.

Οι τρεις πρώτες επιλογές αναφέρονται και ως το πρόβλημα *εξάλειψης πλεονασμού* (*redundancy elimination problem*). Σκοπός είναι η ελαχιστοποίηση της μέσης κατανάλωσης ενέργειας όλων των αισθητήρων και ενεργοποιητών, οι οποίοι βρίσκονται κοντά στην περιοχή συμβάντος.

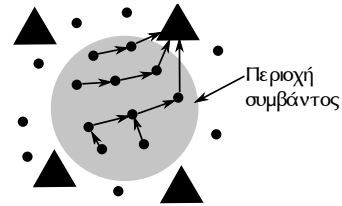
Για παράδειγμα, στο Σχήμα 2.4, από το σύνολο των 18 ενεργοποιητών μόνο οι 9 χρειάζονται για να καλύψουν την περιοχή γεγονότος, ενώ οι υπόλοιποι 9 δεν χρειάζεται να δράσουν. Στο ίδιο παράδειγμα μπορεί να είναι επιθυμητή η επιλογή και των ελάχιστων αισθητήρων για την κάλυψη της περιοχής. Βέβαια, σε μερικές περιπτώσεις εφαρμογών, μπορεί να χρειάζεται ο καθορισμός του επιπέδου πλεονασμού μεταξύ μιας ελάχιστης και μέγιστης τιμής.

Μια πιο αυστηρή απαίτηση, σε μερικές εφαρμογές, μπορεί να είναι η κάλυψη μιας περιοχής όχι μόνο από τους ελάχιστους ενεργοποιητές, αλλά και από αυτούς που η ακτίνα δράσης τους είναι *αμοιβαία αποκλειόμενη* (*mutually exclusive*). Παραδείγματος χάριν, σε ένα σύστημα που οι αισθητήρες παρακολουθούν την υγρασία του εδάφους και οι ενεργοποιητές αρδεύουν την περιοχή, θα πρέπει οι τελευταίοι, όχι μόνο να καλύπτουν όλη την περιοχή, αλλά και οι ακτίνες δράσης τους να μην αλληλοκαλύπτονται, ώστε να η άρδευση να γίνεται ομοιόμορφα.

Αν και η single-hop μετάδοση σε ένα WSN μπορεί να μην είναι αποδοτική, ή ακόμα και πραγματοποιήσιμη, λόγω της απόστασης των αισθητήρων από τον κεντρικό ελεγκτή, κάτι τέτοιο δεν είναι απαραίτητο ότι συμβαίνει σε ένα WSN. Η μικρή απόσταση των αισθητήρων από τους ενεργοποιητές μπορεί να εξασφαλίσει σε ορισμένες περιπτώσεις την single-hop μετάδοση. Συγκεκριμένα, η τοποθέτηση των ενεργοποιητών καθορίζει την αποτελεσματικότητα της single-hop μετάδοσης. Για παράδειγμα, αν η περιοχή συμβάντος είναι σχετικά μικρή και υπάρχει ένας ενεργοποιητής στο κέντρο της, τότε οι κόμβοι που βρίσκονται μακρύτερα θα επιβαρύνονται ενεργειακά λιγότερο. Από την άλλη, όταν η περιοχή συμβάντος είναι μεγαλύτερη ή αν οι ενεργοποιητές είναι τοποθετημένοι στις άκρες ή εκτός της περιοχής, τότε η multi-hop επικοινωνία μπορεί να είναι πιο αποδοτική έναντι της single-hop, εξαι-



2.5.1: Σύστημα πολλαπλών ενεργοποιητών.



2.5.2: Μοναδικός ενεργοποιητής.

Σχήμα 2.5: Επιλογή ενεργοποιητών για την πληροφόρηση συμβάντος. Όπου αισθητήρας και ενεργοποιητής.

τίας της μεγάλης απόστασης. Επομένως, ο τρόπος μετάδοσης καθορίζεται από την τοποθέτηση των ενεργοποιητών, στους οποίους οι αισθητήρες στέλνουν τα δεδομένα.

Επιλογή ενεργοποιητή

Επιπρόσθετα του τρόπου μετάδοσης και των απαιτήσεων της επικοινωνίας μεταξύ αισθητήρων και ενεργοποιητών, τίθεται το ερώτημα για το ποιοι ενεργοποιητές θα ενημερώνονται για κάποιο συμβάν. Στο ερώτημα αυτό υπάρχουν δύο περιπτώσεις επιλογής.

Όπως φαίνεται και στο Σχήμα 2.5.1 υπάρχει η δυνατότητα πολλαπλοί ενεργοποιητές να λαμβάνουν την πληροφόρηση για κάποιο συμβάν από τους αισθητήρες που το εντόπισαν. Η περίπτωση αυτή καλείται *σύστημα πολλαπλών ενεργοποιητών (Multi-Actor)* και κάθε αισθητήρας μπορεί ανεξαρτήτως να αποφασίζει σε ποιον ενεργοποιητή θα στείλει τα δεδομένα του. Βέβαια, η έλλειψη συντονισμού μεταξύ των αισθητήρων μπορεί να οδηγήσει στην επιλογή ακατάλληλων ενεργοποιητών, με αποτέλεσμα την συνολική αύξηση της κατανάλωσης ενέργειας από τους αισθητήρες.

Για να αποφευχθεί το φαινόμενο αυτό, οι αισθητήρες μπορούν να συντονίζονται μεταξύ τους και να σχηματίζουν συστάδες (clusters), όπου σε κάθε μία από αυτές θα υπάρχει ένας ενεργοποιητής που θα συλλέγει τα δεδομένα.

Οι συστάδες αυτές μπορούν να σχηματίζονται με τέτοιο τρόπο ώστε:

- ▶ ο χρόνος μετάδοσης ενός συμβάντος από τους αισθητήρες στους ενεργοποιητές να ελαχιστοποιείται, μιας και η μικρή καθυστέρηση είναι σημαντική για τα WSN,
- ▶ τα συμβάντα να μεταδίδονται μέσω των ελάχιστων ενεργειακά μονοπατιών και

- η ακτίνα δράσης των ενεργοποιητών να καλύπτει ολόκληρη την περιοχή συμβάντος.

Για παράδειγμα, στην εργασία [KM09] προτείνεται μια κατανεμημένη τοποθέτηση των ενεργοποιητών και ένας αλγόριθμος συσταδοποίησης, στον οποίο, ως κέντρα των συστάδων, χρησιμοποιούνται οι ενεργοποιητές και οι οποίοι τοποθετούνται με τέτοιο τρόπο ώστε η κάλυψη τους να μεγιστοποιεί τη συλλογή των δεδομένων, ενώ παράλληλα να ελαχιστοποιεί τους χρόνους δράσης των ενεργοποιητών. Στο [MF06] παρουσιάζεται το πρωτόκολλο ADP (Actuator Discovery Protocol) το οποίο παρέχει μία βέλτιστη προσάρτηση των αισθητήρων στους πλησιέστερους τους ενεργοποιητές, εγγυώντας τον συγχρονισμό και την διάταξη των γεγονότων, ενώ παράλληλα εξαλείφει τον ενδεχόμενο πλεονασμό ενεργειών από τους ενεργοποιητές. Ακόμα, στο [MPGA05] χρησιμοποιείται η τεχνική της συσταδοποίησης, με τις συστάδες να δημιουργούνται on-the-fly βάσει του συμβάντος, στοχεύοντας στη βέλτιστη και αξιόπιστη δράση, ελαχιστοποιώντας την κατανάλωση ενέργειας. Επίσης, στο [BMM06] προτείνεται ένα σχήμα συσταδοποίησης όπου κάθε ενεργοποιητής δημιουργεί μια περιοχή Voronoi στην οποία εμπεριέχονται οι πλησιέστεροι αισθητήρες, ενώ οι τελευταίοι ομαδοποιούνται σε συστάδα όταν αντιλαμβάνονται κάποιο συμβάν, εξασφαλίζοντας πως μόνο ο πλησιέστερος ενεργοποιητής θα πληροφορείται για το συμβάν αυτό.

Σε αντίθεση με το σύστημα πολλαπλών ενεργοποιητών, στο Σχήμα 2.5.2 παρουσιάζεται η περίπτωση όπου ένας μόνο ενεργοποιητής ενημερώνεται για το συμβάν. Η επιλογή ενός μοναδικού ενεργοποιητή για την ενημέρωση ονομάζεται *μοναδικός ενεργοποιητής (Single-Actor)* και μπορεί να θεωρηθεί ως μια ειδική περίπτωση του συστήματος πολλαπλών ενεργοποιητών. Μια από τις σημαντικότερες προκλήσεις εδώ είναι η επιλογή του μοναδικού ενεργοποιητή, στο οποίο οι αισθητήρες θα στέλνουν τα δεδομένα τους. Η επιλογή του καταλληλότερου ενεργοποιητή μπορεί να γίνει με βάση κριτήρια όπως:

- η απόσταση μεταξύ του ενεργοποιητή και της περιοχή συμβάντος να είναι μικρή, με στόχο τη μείωση των καθυστερήσεων και της κατανάλωσης ενέργειας,
- επιλογή του ελάχιστου ενεργειακά μονοπατιού από τους αισθητήρες προς τον ενεργοποιητή,
- η ακτίνα δράσης του ενεργοποιητή, ώστε οι κατάλληλες ενέργειες να πραγματοποιούνται στην περιοχή συμβάντος.

Να σημειωθεί πως το τελευταίο κριτήριο δεν εγγυάται την κάλυψη ολόκληρης της περιοχής συμβάντος από την ακτίνα δράσης του ενεργοποιητή. Επομένως, αντί να λαμβάνεται υπόψη η κατανάλωση ενέργειας και η καθυστέρηση, ίσως είναι προτιμότερο η επιλογή του κατάλληλου ενεργοποιητή να γίνεται με βάση τη θέση του,

ώστε να καλύπτει όλη την περιοχή συμβάντος. Σε αυτή την περίπτωση, ο ενεργοποιητής που λαμβάνει την πληροφόρηση για κάποιο συμβάν είναι σε θέση να ενεργήσει κατάλληλα, χωρίς την ανάγκη συντονισμού με άλλους ενεργοποιητές.

Στη περίπτωση του μοναδικού ενεργοποιητή, ο τελευταίος μπορεί να ενεργήσει άμεσα, αν η ακτίνα δράσης του και η ενεργειακή κατάστασή του τού το επιτρέπουν και η ενέργεια απαιτεί μόνο έναν από αυτούς. Στο ενδεχόμενο αυτό, η καθυστέρηση δεν παίζει καθοριστικό ρόλο και η απόκριση του δικτύου γίνεται μεγάλη. Όμως, αν ένας μόνο ενεργοποιητής δεν επαρκεί, ή αν αυτός δεν είναι ο κατάλληλος, λόγω θέσης ή ενεργειακής κατάστασης, τότε για την εκτέλεση της ενέργειας εκπέμπεται μια ανακοίνωση στους υπόλοιπους ενεργοποιητές. Βάσει των απαντήσεων στην ανακοίνωση αυτή από τους υπόλοιπους, μπορεί να επιλεγεί ένας ή περισσότεροι ενεργοποιητές για την εκτέλεση των κατάλληλων ενεργειών.

Η υπεροχή του συστήματος πολλαπλών ενεργοποιητών σε σχέση με το μοναδικό ενεργοποιητή έχει να κάνει με το γεγονός ότι το πρώτο παρέχει στους ενεργοποιητές τη δυνατότητα να εντοπίσουν το κέντρο του συμβάντος. Η ένταση ενός συμβάντος μπορεί να μην είναι ομοιόμορφα κατανομημένη σε ολόκληρη την περιοχή. Χρησιμοποιώντας διαφορετική ισχύ για την μετάδοση του συμβάντος από τους αισθητήρες προς κάθε ενεργοποιητή, οι τελευταίοι μπορούν να συγκρίνουν την ισχύ των λαμβανόμενων σημάτων μεταξύ τους και να εντοπίζουν πού είναι εντονότερο το συμβάν.

Το μειονέκτημα του συστήματος πολλαπλών ενεργοποιητών είναι ότι ο συντονισμός μεταξύ τους γίνεται ως επί το πλείστον με την ανταλλαγή μηνυμάτων μεταξύ τους, ενώ στη περίπτωση ενός μοναδικού ενεργοποιητή γίνεται με μηνύματα ανακοίνωσης. Ο κάθε ενεργοποιητής, από το σύστημα πολλαπλών ενεργοποιητών, διαθέτει μερική πληροφόρηση για το συμβάν και επομένως για να ενεργήσει κατάλληλα πρέπει να συντονιστούν μεταξύ τους. Αυτός ο συντονισμός μπορεί να οδηγήσει σε overhead στην επικοινωνία και σε καθυστερήσεις.

2.2.2 Συντονισμός μεταξύ ενεργοποιητών

Όπως έχει ήδη αναφερθεί, οι ενεργοποιητές επικοινωνούν και μεταξύ τους, εκτός από τους αισθητήρες. Όταν ένας ενεργοποιητής λαμβάνει τα δεδομένα ενός αισθητήρα και δεν μπορεί να ενεργήσει στη περιοχή του συμβάντος, είτε επειδή η ακτίνα δράσης του είναι μικρή, είτε λόγω χαμηλών ενεργειακών αποθεμάτων, τότε πρέπει να επικοινωνήσει με κάποιον άλλον και να του ανακοινώσει το συμβάν.

Επίσης, οι ενεργοποιητές πρέπει να επικοινωνήσουν μεταξύ τους όταν δεν επαρκεί ένας μόνο για την ενέργεια που πρέπει να εκτελεστεί. Από την άλλη, υπάρχουν περιπτώσεις που πολλαπλοί ενεργοποιητές λαμβάνουν πληροφόρηση για το ίδιο συμβάν, αλλά χρειάζεται η δράση μόνο από έναν. Τότε, πρέπει πάλι να επικοινωνήσουν μεταξύ τους, ώστε να αποφασιστεί ποιος από όλους θα ενεργήσει για το συμβάν.

Σε κάποιες εφαρμογές, αν απαιτούνται περισσότεροι του ενός ενεργοποιητές για την κάλυψη ολόκληρης της περιοχής συμβάντος, ίσως χρειάζεται να εξασφαλιστεί ότι δεν υπάρχει επικάλυψη των ακτίνων δράσης τους ή ότι αυτές είναι αμοιβαία αποκλειόμενες. Στην περίπτωση αυτή είναι αναγκαία πάλι η επικοινωνία μεταξύ τους, ώστε δράσουν ομοιόμορφα σε ολόκληρη την περιοχή συμβάντος.

Μια ακόμα περίπτωση, που απαιτείται η επικοινωνία μεταξύ των ενεργοποιητών, είναι όταν αυτοί λαμβάνουν πληροφόρηση για κάποιο συμβάν από πολλαπλούς αισθητήρες. Τότε, είναι πιθανό να πρέπει να εξασφαλιστεί ότι αυτοί θα δράσουν ταυτόχρονα. Αυτός ο συγχρονισμός στην εκτέλεση των ενεργειών απαιτείται σε εφαρμογές όπου η μερική δράση των ενεργοποιητών αλλάζει την κατάσταση του συμβάντος. Στην περίπτωση που πολλαπλά συμβάντα συμβαίνουν ταυτόχρονα, η ανάθεση κάποιας ενέργειας μπορεί να γίνει με την επικοινωνία μεταξύ των ενεργοποιητών. Επίσης, μπορεί να είναι επιθυμητό οι ενέργειες να εκτελούνται ακολουθιακά.

Όταν ένας ενεργοποιητής λάβει την πληροφόρηση για κάποιο συμβάν, το οποίο εξαπλώνεται σε περιοχές δράσης και άλλων ενεργοποιητών, τότε ο πρώτος μπορεί να ενημερώσει τους υπόλοιπους με τα δεδομένα του αισθητήρα ή με τη ενέργεια που πρέπει αυτοί να εκτελέσουν. Έτσι, δεν υπάρχει η ανάγκη να στείλουν οι αισθητήρες των περιοχών εξάπλωσης τα δεδομένα τους στους κοντινούς ενεργοποιητές. Αυτή η λύση είναι αντίστοιχη με την παρακολούθηση της εξέλιξης ενός συμβάντος από τους αισθητήρες που περιγράφηκε στην προηγούμενη ενότητα.

Από όλες τις παραπάνω περιπτώσεις προκύπτει η ανάγκη για τον συντονισμό μεταξύ των ενεργοποιητών, η οποία οδηγεί στο σημαντικό ερώτημα:

Ποια ενέργεια πρέπει να εκτελεστεί και από ποιους ενεργοποιητές;

Η ανάθεση των εργασιών στα WSNs μπορεί να εξεταστεί σε δύο άξονες. Ο πρώτος έχει να κάνει με το πλήθος των ενεργοποιητών που χρειάζονται για την επιτέλεση της εργασίας, ενώ ο δεύτερος με τη λήψη απόφασης για τη εργασία που πρέπει να εκτελεστεί.

Μια εργασία μπορεί να χρειάζεται μόνο ένα ενεργοποιητή για την εκτέλεσή της ή περισσότερους. Στην πρώτη περίπτωση πρόκειται για εργασία ενός ενεργοποιητή (*Single-Actor Task*), ενώ στη δεύτερη πρόκειται για μια εργασία πολλαπλών ενεργοποιητών (*Multi-Actor Task*).

Επίσης, σε ένα WSN υπάρχει η ανάγκη για την λήψη απόφασης για κάποια ενέργεια που πρέπει να πραγματοποιηθεί, ανάλογα με τα χαρακτηριστικά του συμβάντος. Η απόφαση αυτή μπορεί να ληφθεί είτε με συγκεντρωτικό τρόπο είτε με κατανεμημένο. Ο τελευταίος επιτρέπει στους ενεργοποιητές, που είναι κοντά, να συντονίζονται ώστε να παρέχουν έγκαιρες ενέργειες, ανεξάρτητα από το μέγεθος του δικτύου. Από την άλλη, η συγκεντρωτική μέθοδος, προσφέρει έναν πιο οργανωμένο

τρόπο για την λήψη απόφασης, αφού αυτή γίνεται από μόνο έναν ενεργοποιητή, ο οποίος μπορεί να είναι καλύτερα εξοπλισμένος.

Στην ενότητα που ακολουθεί, εξετάζονται κάποιες περιπτώσεις ανάθεσης εργασιών σε ενεργοποιητές, βάσει των δύο αξόνων, που μόλις περιγράφηκαν.

Ανάθεση εργασίας

Αν οι αισθητήρες αναφέρουν ένα συμβάν σε πολλαπλούς ενεργοποιητές¹ και η εργασία απαιτεί περισσότερους του ενός ενεργοποιητές, τότε στην κατανομημένη λήψη απόφασης οι ενεργοποιητές αυτοί «διαπραγματεύονται» μεταξύ τους και συντονίζονται τοπικά, επιλέγοντας τους πλέον κατάλληλους² για την επιτέλεση της εργασίας. Από την άλλη, στην περίπτωση συγκεντρωτικής απόφασης, τότε οι ενεργοποιητές μεταδίδουν, απ' ευθείας, τα δεδομένα για το συμβάν στον προκαθορισμένο ενεργοποιητή, ο οποίος βασισμένος στην γνώση που έχει από πριν για τις δυνατότητες των υπόλοιπων ενεργοποιητών, λαμβάνει την απόφαση για την επιλογή των κατάλληλων. Αφού γίνει η επιλογή αυτή, ο κεντρικός ενεργοποιητής ενημερώνει τους επιλεγμένους με τις ενέργειες που πρέπει να εκτελέσουν.

Οι επιλεγμένοι ενεργοποιητές, και στις δύο περιπτώσεις λήψης απόφασης, πιθανόν να μην αυτοί που έλαβαν από τους αισθητήρες την πληροφόρηση για το συμβάν, διότι ίσως να μην είναι οι κατάλληλοι για την επιτέλεση της εργασίας.

Στη περίπτωση που ένας, μόνο, ενεργοποιητής λαμβάνει την πληροφόρηση για κάποιο συμβάν³ και η εργασία απαιτεί περισσότερους από έναν ενεργοποιητές, πρέπει να προηγηθεί ο συντονισμός μεταξύ των ενεργοποιητών πριν αυτή εκτελεστεί. Ο συντονισμός αυτός είναι απαραίτητος, για την επιλογή των ενεργοποιητών και τον καθορισμό το σημείων στην περιοχή συμβάντος, που πρέπει αυτοί να δράσουν. Βέβαια, στην περίπτωση αυτή, αφού τα δεδομένα λαμβάνονται σε ένα μόνο ενεργοποιητή, αυτός μπορεί να λειτουργήσει ως κέντρο λήψης απόφασης και να εκπέμψει ένα μήνυμα ανακοίνωσης στους υπόλοιπους ενεργοποιητές, το οποίο περιέχει τις λεπτομέρειες για το συμβάν και την εργασία που πρέπει να εκτελεστεί. Από τις αποκρίσεις των ενεργοποιητών στην ανακοίνωση αυτή επιλέγει αυτούς που είναι κατάλληλοι για την επιτέλεση της εργασίας.

Αφού ανατεθεί η εργασία στους κατάλληλους ενεργοποιητές, αυτοί εκτελούν τις ενέργειες σε όλη την ακτίνα δράσης τους, οι οποίες συνολικά πρέπει να καλύπτουν όλη την περιοχή συμβάντος. Αν οι ακτίνες δράσης των ενεργοποιητών συνολικά υπερκαλύπτουν την περιοχή συμβάντος, μερικές ενέργειες θα εκτελεστούν εκτός

¹Πρόκειται για την περίπτωση συστήματος πολλαπλών ενεργοποιητών που αναφέρεται στην Ενότητα 2.2.1.

²Κατάλληλος ενεργοποιητής, στην περίπτωση αυτή, είναι εκείνος που βρίσκεται πιο κοντά στην περιοχή συμβάντος, ή διαθέτει μεγαλύτερο απόθεμα ενέργειας, ή εκείνος του οποίου ο χρόνος ολοκλήρωσης της ενέργειάς του είναι ο μικρότερος.

³Πρόκειται για την περίπτωση μοναδικού ενεργοποιητή που αναφέρεται στην Ενότητα 2.2.1.

αυτής. Το γεγονός αυτό οδηγεί σε σπατάλη πόρων των ενεργοποιητών, ενώ σε μερικές εφαρμογές μπορεί να είναι και καταστροφικό. Παρόμοια, όταν οι ακτίνες δράσεις μερικών ενεργοποιητών αλληλοκαλύπτονται και υπάρχει συγχρονισμένη δράση μπορεί να δημιουργήσει προβλήματα.

Συνεπώς, ανεξάρτητα από το πλήθος των ενεργοποιητών που λαμβάνουν δεδομένα από τους αισθητήρες για κάποιο συμβάν, ο κύριος στόχος μιας εργασίας πολλαπλών ενεργοποιητών είναι η επιλογή των κατάλληλων από αυτούς, οι οποίοι πληρούν τις απαιτήσεις της εκάστοτε εφαρμογής και των χαρακτηριστικών του συμβάντος.

Για μια εργασία μοναδικού ενεργοποιητή, ο βασικός στόχος είναι η επιλογή εκείνου, ο οποίος είναι κατάλληλος για την εκτέλεση της εργασίας. Συγκεκριμένα, αν χρησιμοποιείται ένα σύστημα πολλαπλών ενεργοποιητών, ο συντονισμός για μια εργασία ενός ενεργοποιητή μπορεί να θεωρηθεί ως μια ειδική περίπτωση της εργασίας πολλαπλών ενεργοποιητών, ανεξαρτήτως του τρόπου λήψης απόφασης (κατανεμημένη ή συγκεντρωτική). Βέβαια, αν ένας ενεργοποιητής πληροφορηθεί για κάποιο συμβάν (περίπτωση μοναδικού ενεργοποιητή), τότε προκύπτει το ερώτημα αν είναι αυτός που θα επιτελέσει την εργασία ή θα αποφασίσει για την επιλογή κάποιου άλλου.

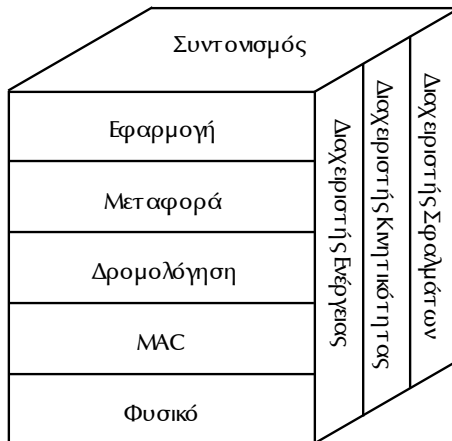
Στην περίπτωση εφαρμογής, μη ανεκτικής σε καθυστερήσεις και όταν η επικοινωνία μεταξύ αισθητήρα και ενεργοποιητή είναι αργή, τότε αν ο τελευταίος τηρεί τις βασικές απαιτήσεις για την εκτέλεση της εργασίας είναι προτιμότερο να δράσει άμεσα. Σε αυτό το ενδεχόμενο, η εργασία μπορεί να μην εκτελεστεί από τον καταλληλότερο ενεργοποιητή, αλλά εξασφαλίζεται η άμεση απόκριση στο συμβάν, πράγμα το οποίο απαιτεί η εφαρμογή.

Από την άλλη, αν η καθυστέρηση επικοινωνίας μεταξύ αισθητήρα και ενεργοποιητή είναι μικρή, ή ο τελευταίος δεν είναι κατάλληλος για την εκτέλεση της εργασίας, τότε πρέπει να επικοινωνήσει με το κόμβο που παίρνει τις αποφάσεις, στην περίπτωση συγκεντρωτικής λήψης αποφάσεων, ή να εκπέμψει μία ανακοίνωση για την κατανεμημένη λήψη απόφασης, όπως περιγράφηκε προηγουμένως. Και στις δύο αυτές περιπτώσεις θα ενημερωθεί ο κατάλληλος ενεργοποιητής για την εκτέλεση της εργασίας.

2.3 Πρωτόκολλα για τα WSNs

Στο [AK04] προτείνεται η στοίβα πρωτοκόλλων του Σχήματος 2.6, η οποία αποτελείται από τρεις διαστάσεις (planes), επικοινωνίας (Communication), συντονισμού (Coordination) και διαχείρισης (Management).

Η διάσταση επικοινωνίας επιτρέπει την ανταλλαγή πληροφορίας μεταξύ των κόμβων του δικτύου. Τα δεδομένα που λαμβάνονται από τον κόμβο στη διάσταση αυτή



Σχήμα 2.6: Στοιβά πρωτοκόλλων για τα WSN.

μεταφέρονται σε αυτή του συντονισμού, η οποία αποφασίζει το τρόπο με τον οποίο ενεργεί ο κόμβος στα δεδομένα αυτά.

Η διάσταση διαχείρισης είναι υπεύθυνη για την παρακολούθηση και τον έλεγχο του αισθητήρα ή του ενεργοποιητή, εξασφαλίζοντας την ορθή λειτουργία. Η τελευταία παρέχει επίσης και την απαραίτητη πληροφορία στη διάσταση συντονισμού για την λήψη αποφάσεων.

Στις ενότητες που ακολουθούν αναλύονται οι απαιτήσεις και τα χαρακτηριστικά της κάθε διάστασης για τους δύο

τύπους συντονισμού, που έχουν αναφερθεί, αισθητήρα-ενεργοποιητή και ενεργοποιητή-ενεργοποιητή.

2.3.1 Διάσταση διαχείρισης

Οι λειτουργίες που εκτελούνται από τη διάσταση διαχείρισης μπορούν να ομαδοποιηθούν στις ακόλουθες κατηγορίες:

- Η **διαχείριση ενέργειας** ρυθμίζει το πώς ένας κόμβος χρησιμοποιεί την ενέργειά του. Για παράδειγμα, όταν το επίπεδο ισχύος ενός κόμβου είναι χαμηλό, το επίπεδο αυτό ενημερώνει τη διάσταση συντονισμού, ώστε ο κόμβος να μη συμμετέχει σε λειτουργίες αίσθησης, αναμετάδοσης ή την εκτέλεση ενεργειών.
- Η **διαχείριση κινητικότητας** ανιχνεύει και καταγράφει τις κινήσεις του κόμβου, ώστε η συνεκτικότητα του δικτύου να παραμένει συνεχώς.
- Η **διαχείριση σφαλμάτων** αναφέρεται στην ανίχνευση και την επίλυση των προβλημάτων στον κόμβο. Για παράδειγμα, όταν η ευαισθησία κάποιας μονάδας αίσθησης μειωθεί, ενημερώνει τη διάσταση συντονισμού για αυτή την κατάσταση.

2.3.2 Διάσταση συντονισμού

Η διάσταση συντονισμού καθορίζει το τρόπο με τον οποίο ένας κόμβος «συμπεριφέρεται» ανάλογα με τα δεδομένα που αυτός λαμβάνει από τις διαστάσεις επικοινωνίας και διαχείρισης. Ένας αισθητήρας, αφού αντιληφθεί ένα συμβάν, επικοινωνεί

με τους υπόλοιπους στέλνοντας τις μετρήσεις τους. Κάθε αισθητήρας, που λαμβάνει τα δεδομένα αυτά, τα μεταφέρει στη διάσταση συντονισμού, ώστε να λάβει κάποια απόφαση. Με αυτό τον τρόπο, οι αισθητήρες είναι ικανοί να συντονίζονται μεταξύ τους σε ένα υψηλότερο επίπεδο. Επιπλέον, μπορεί να απαιτείται ο συντονισμός ανάμεσα στους αισθητήρες, με σκοπό είτε τον καθορισμό εκείνων που δεν θα στείλουν τα δεδομένα τους, είτε για multi-hop δρομολόγηση, είτε την συνάθροιση δεδομένων, αλλά και για τον πιο σημαντικό λόγο, την επιλογή των ενεργοποιητών για την αποστολή πληροφοριών.

Η ύπαρξη διάστασης συντονισμού είναι κρίσιμης σημασίας για τους ενεργοποιητές, περισσότερο από ότι για τους αισθητήρες, αφού αυτοί συνεργάζονται μεταξύ τους, ώστε να επιτελέσουν τις ενέργειές τους. Όταν παρουσιάζεται κάποιο συμβάν, ο κοινός στόχος όλων των ενεργοποιητών είναι να εκτελέσουν την ανάλογη ενέργεια. Επομένως, οι «κοινωνικές» ικανότητες και η δυνατότητα διαπραγμάτευσης είναι απαραίτητες για ένα WSN, ώστε να εξασφαλίζεται η συνεπής συμπεριφορά των ενεργοποιητών.

2.3.3 Διάσταση επικοινωνίας

Η διάσταση επικοινωνίας λαμβάνει εντολές⁴ από τη διάσταση συντονισμού και σύμφωνα με αυτές παρέχει τη σύνδεση με άλλους κόμβους χρησιμοποιώντας πρωτόκολλα επικοινωνίας. Συγκεκριμένα, η διάσταση επικοινωνίας μεταχειρίζεται το φυσικό κανάλι, τη πρόσβαση του κόμβου στο μέσο (MAC), την επιλογή των μονοπατιών για τη δρομολόγηση και την μεταφορά πακέτων από τον ένα κόμβο στον άλλον.

Στις ενότητες που ακολουθούν γίνεται μια παρουσίαση των απαιτήσεων που πρέπει να τηρούνται από τα επίπεδα μεταφοράς, MAC και δρομολόγησης.

Επίπεδο μεταφοράς

Εκτός των λειτουργιών που προσφέρουν τα υπάρχοντα πρωτόκολλα, ένα πρωτόκολλο μεταφοράς για τα WSNs πρέπει να πληροί τις απαιτήσεις για εφαρμογές πραγματικού χρόνου. Αρκετά πρωτόκολλα μεταφοράς έχουν αναπτυχθεί για ad-hoc δίκτυα και WSNs. Όμως, δύσκολα υλοποιείται ένα πρωτόκολλο μεταφοράς που εξασφαλίζει αξιοπιστία και χαμηλές καθυστερήσεις. Για παράδειγμα, αν παρατηρηθεί χαμηλή αξιοπιστία στην επικοινωνία αισθητήρα-ενεργοποιητή, τότε το πρωτόκολλο μεταφοράς πρέπει να ενημερώσει για την ύπαρξη αναξιόπιστων δεδομένων και τους γειτονικούς ενεργοποιητές. Εφόσον οι επικοινωνίες αισθητήρα-ενεργοποιητή και ενεργοποιητή-ενεργοποιητή υφίστανται συνεχώς στα

⁴Οι εντολές αυτές έχουν να κάνουν με τις αποφάσεις που έλαβε η διάσταση συντονισμού για το πώς θα συμπεριφερθεί ο κόμβος.

WSAN, η χρήση ενός πρωτοκόλλου μεταφοράς που καλύπτει και τους δύο αυτούς τύπους είναι αναγκαία.

Επίπεδο δρομολόγησης

Όπως έχει ήδη αναφερθεί, η επιλογή ενός ενεργοποιητή από έναν αισθητήρα για την αποστολή των δεδομένων αποτελεί πρόκληση. Όμως, αφού αυτή η επιλογή πραγματοποιηθεί, τα δεδομένα από τον αισθητήρα πρέπει να δρομολογηθούν στο επιλεγμένο ενεργοποιητή με έναν αποδοτικό τρόπο. Για παράδειγμα, ενώ τα δεδομένα μεταδίδονται προς τον ενεργοποιητή, μέσω των υπόλοιπων κόμβων, μπορούν να συναθροισθούν και με άλλα από τους ενδιαμέσους κόμβους, ώστε να επιτευχθεί μεγαλύτερη αποδοτικότητα. Επιπροσθέτως, η επιλογή του μονοπατιού για την δρομολόγηση των δεδομένων πρέπει να γίνεται με τέτοιο τρόπο ώστε να εξασφαλίζεται η χαμηλή καθυστέρηση⁵. Ακόμα, το πρωτόκολλο δρομολόγησης πρέπει να υποστηρίζει τη ρύθμιση προτεραιοτήτων, εξασφαλίζοντας την ταχύτερη μετάδοση κρίσιμων δεδομένων.

Κάποια πρωτόκολλα δρομολόγησης για ad-hoc δίκτυα, όπως το DSR, το AODV [PBRD03] και το OLSR, μπορούν να χρησιμοποιηθούν, αρκεί να μπορούν να υποστηρίξουν τις απαιτήσεις για πραγματικό χρόνο.

Επίπεδο MAC

Για να είναι δυνατή η μετάδοση της πληροφορίας από ένα μεγάλο πλήθος αισθητήρων στους ενεργοποιητές αποδοτικά, η ύπαρξη ενός MAC πρωτοκόλλου είναι αναγκαία. Επιπλέον, σε μερικές εφαρμογές, όπου οι ενεργοποιητές είναι κινούμενοι και αλλάζουν περιοχές, το πρωτόκολλο MAC πρέπει να φροντίζει την συνδεσιμότητά τους με το υπόλοιπο δίκτυο.

Τα κλασσικά, βασισμένα στην σύνδεση (connection-based), πρωτόκολλα δεν είναι κατάλληλα για εφαρμογές πραγματικού χρόνου, αφού η πρόσβαση στο κανάλι απαιτεί την ύπαρξη handshaking, το οποίο αυξάνει το χρόνο μετάδοσης των δεδομένων.

Το πρωτόκολλο IEEE 802.15.4 [The03] έχει σχεδιαστεί ώστε να προσφέρει αξιόπιστη, χαμηλής κατανάλωσης επικοινωνία, για τη χρήση του σε ασύρματα δίκτυα αισθητήρων, αλλά μπορεί να χρησιμοποιηθεί και στα WSANs [BdI⁺06]. Το IEEE 802.15.4 καθορίζει το φυσικό επίπεδο και το επίπεδο MAC, για μεταδόσεις χαμηλού ρυθμού και ασύρματη σύνδεση με σταθερές, κινούμενες και φορητές συσκευές, χαμηλής κατανάλωσης ενέργειας, οι οποίες λειτουργούν στον *προσωπικό χώρο* (POS – Personal Operating Space) των 10m.

⁵Ειδικά στις περιπτώσεις που υπάρχουν χρονικές προθεσμίες για την παράδοση δεδομένων, ώστε αυτά να θεωρούνται έγκυρα.

2.4 Ανακεφαλαίωση

Η αποτελεσματικότητα που μπορεί να εγγυώνται τα ασύρματα δίκτυα αισθητήρων (WSNs) μπορεί να διαταραχθεί με την προσθήκη των ενεργοποιητών στο δίκτυο. Όταν στο πεδίο που αναπτύσσονται οι αισθητήρες εμφανίζονται και οι ενεργοποιητές, γεννούνται νέα ζητήματα καθώς επεκτείνονται οι επιλογές λειτουργίας. Εκτός από την παρατήρηση του περιβάλλοντος χώρου από τους αισθητήρες, δημιουργούνται πλέον οι επιλογές για την δράση σε αυτόν, καθώς και τη λήψη αποφάσεων για την επιλογή της κατάλληλης δράσης βάσει της παρατήρησης.

Κατά την υλοποίηση ενός WSN είναι επιθυμητό, στις περισσότερες των περιπτώσεων, να εξασφαλίζονται οι απαιτήσεις που προκύπτουν από την συνύπαρξη των δύο διαφορετικών ομάδων κόμβων, που περιγράφηκαν στο Κεφάλαιο αυτό. Το ζήτημα του συντονισμού μεταξύ αισθητήρων και ενεργοποιητών ενός WSN αποτελεί ένα μεγάλο πεδίο για έρευνα αλγορίθμων που εξασφαλίζουν το συγχρονισμό και την εξάλειψη πλεοναζόντων ενεργειών από τους ενεργοποιητές. Επιπλέον, για τον συντονισμό μεταξύ των ενεργοποιητών προκύπτει η ανάγκη για τον καθορισμό ενός ενιαίου πλαισίου λειτουργίας, το οποίο θα εκμεταλλεύεται την διαφορετικότητα της κάθε εφαρμογής και θα επιλέγει τον καταλληλότερο (βέλτιστο) τρόπο για την πραγματοποίηση της αποδοτικότερης συνεργασίας μεταξύ των ενεργοποιητών. Ακόμα, λαμβάνοντας υπ' όψιν την ανάγκη για την ελαχιστοποίηση της κατανάλωσης ενέργειας και την ανίχνευση, ή ακόμα και την αποτροπή, ύπαρξης σφαλμάτων, είναι απαραίτητη η ύπαρξη ενός πλαισίου λειτουργίας που θα λαμβάνει υπ' όψιν και τις τρεις διαστάσεις, συντονισμού, επικοινωνίας και διαχείρισης.

Τέλος, στις περισσότερες εφαρμογές WSN η κρισιμότητα της απόκρισης σε πραγματικό χρόνο είναι απαραίτητη. Προκύπτει, επομένως, η ανάγκη για πρωτόκολλα επικοινωνίας συντονισμού μεταξύ αισθητήρων και ενεργοποιητών, αλλά και μεταξύ των τελευταίων, που θα εξασφαλίζουν την απαίτηση αυτή.

Κεφάλαιο 3

Universal Plug and Play

Με την εμφάνιση του Device Plug and Play (PnP) στα λειτουργικά συστήματα, η σύνδεση και η ρύθμιση περιφερειακών συσκευών σε ένα υπολογιστή έγινε σημαντικά πιο εύκολη. Το Universal Plug and Play (UPnP) επεκτείνει αυτή την απλότητα σε ολόκληρα δίκτυα από συσκευές και υπηρεσίες, καθιστώντας ικανή την αναζήτηση και τον έλεγχο τους.

3.1 Τι είναι το UPnP;

Το UPnP είναι παραπάνω από μια απλή επέκταση του PnP για τις περιφερειακές συσκευές. Είναι σχεδιασμένο ώστε να υποστηρίζει zero-configuration, αόρατη δικτύωση και αυτόματη ανακάλυψη για πολλές συσκευές από ένα μεγάλο εύρος κατασκευαστών.

Με το UPnP, μια συσκευή μπορεί με δυναμικό τρόπο να συνδέεται στο δίκτυο, να αποκτά μια διεύθυνση IP, να εκχωρεί τις δυνατότητές της και να ενημερώνεται για τις δυνατότητες των άλλων συσκευών, και όλα αυτά αυτόματα. Οι συσκευές στη συνέχεια μπορούν να επικοινωνούν απ' ευθείας μεταξύ τους σχηματίζοντας έτσι μια peer-to-peer δικτύωση. Οι ποικιλία των διαφορετικών τύπων συσκευών, οι οποίες μπορούν να επωφεληθούν από την UPnP δικτύωση είναι μεγάλη. Στις συσκευές αυτές περιλαμβάνονται «έξυπνες» οικιακές συσκευές, ασύρματες συσκευές και υπολογιστές οποιασδήποτε μορφής.

Τα πεδία εφαρμογής του UPnP μπορούν να είναι πολλά, όπως για παράδειγμα στο αυτοματισμό του σπιτιού, στην audio/video ψυχαγωγία, τις οικιακές μιας κουζίνας, τα κινητά δίκτυα, δίκτυα εγγύτητας κ.ά..

Το UPnP χρησιμοποιεί το TCP/IP και τα πρωτόκολλα του Internet, επιτρέποντας την αδιάλειπτη προσαρμογή στα υπάρχοντα δίκτυα. Η χρήση αυτών των στάνταρντ πρωτοκόλλων επιτρέπει στο UPnP να επωφεληθεί τόσο από πλευράς εμπειρίας όσο και γνώσης, και καθιστά τη δια-λειτουργικότητα εσωτερικό χαρακτηριστικό.

Επειδή το UPnP είναι μία κατανεμημένη, ανοικτή αρχιτεκτονική δικτύωσης, η οποία καθορίζεται από τα πρωτόκολλα που χρησιμοποιεί, είναι ανεξάρτητη του λειτουργικού συστήματος, της γλώσσας προγραμματισμού και του φυσικού μέσου, όπως ακριβώς είναι και το Internet. Δεν προσδιορίζει τα APIs τα οποία θα χρησιμοποιήσουν οι εφαρμογές, επιτρέποντας στους σχεδιαστές και κατασκευαστές των λειτουργικών συστημάτων να δημιουργήσουν τα δικά τους, ανάλογα με τις ανάγκες των πελατών τους.

3.2 Ποιος καθορίζει το UPnP;

Το Universal Plug and Play Forum καθορίζει τις Περιγραφές (Descriptions) για τις UPnP συσκευές και υπηρεσίες, σύμφωνα με μια κοινή αρχιτεκτονική συσκευής. Το UPnP Forum σχηματίστηκε τον Οκτώβριο του 1999 και πλέον αποτελεί μια σύμπραξη περισσότερων των οκτακοσίων κατασκευαστών προσωπικών υπολογιστών, ηλεκτρονικών συσκευών οικιακού αυτοματισμού και ασφάλειας, οικοσκευών, δικτυακών προϊόντων υπολογιστών, κινητών συσκευών κ.ά..

Έχει ως στόχο να επιτρέψει την εύκολη και ισχυρή διασύνδεση μεταξύ των προϊόντων αυτών και παράλληλα να απλοποιήσει την υλοποίηση των δικτύων σε οικιακά και εργασιακά περιβάλλοντα. Παράλληλα, επιδιώκει την ανάπτυξη προτύπων για την περιγραφή συσκευών και πρωτοκόλλων, τα οποία βασίζονται σε XML σχήματα και τα οποία εξασφαλίζουν την δια-λειτουργικότητα των συσκευών σε κλιμακούμενα δικτυακά περιβάλλοντα.



Σχήμα 3.1: Το λογότυπο UPnP.

Το UPnP Implements Corporation (UIC) αποτελείται από εταιρείες διαφόρων κλάδων της βιομηχανίας, οι οποίες είναι μέλη του UPnP Forum και οι οποίες προωθούν τη θέσπιση ενιαίων τεχνικών προδιαγραφών για τη διασύνδεση, τον έλεγχο και την πιστοποίηση των εν λόγω συσκευών. Το UIC αναπτύσσει και διαχειρίζεται τις διαδικασίες δοκιμαστικού ελέγχου και πιστοποίησης,

καθώς επίσης και το πρόγραμμα λογοτύπου (UPnP Logo Program). Παράλληλα, παρέχει πληροφορίες σχετικές με την πιστοποίηση των UPnP συσκευών στα μέλη του, αλλά και σε οποιονδήποτε ενδιαφερόμενο. Η πιστοποίηση μιας συσκευής, ως *UPnP συμβατή*, είναι μια διαδικασία ανοικτή σε οποιονδήποτε κατασκευαστή, αρκεί αυτός να είναι μέλος του UPnP Forum και του UIC, να έχει καταβάλει το καθορισμένο χρηματικό ποσό στο UIC και οι συσκευές του υποστηρίζουν αρχιτεκτονική UPnP.

Το UPnP Forum έχει συστήσει επιτροπές εργασίας, ειδικευμένες σε συγκεκριμένους τομείς, οι οποίες ευθύνονται για τη δημιουργία προτεινόμενων προτύπων για τις

συσκευές και τη κατασκευή εφαρμογών ελέγχου και δοκιμής. Οι κατασκευαστές UPnP συσκευών μπορούν να σχεδιάσουν τα προϊόντα τους με τη βεβαιότητα της δια-λειτουργικότητας και τα οφέλη της κοινής πνευματικής ιδιοκτησίας και του προγράμματος λογοτύπου. Οι κατασκευαστές εκτός του προγράμματος λογοτύπου μπορούν επίσης να δημιουργήσουν προϊόντα, τα οποία τηρούν τις προδιαγραφές της αρχιτεκτονικής UPnP χωρίς τον επίσημο καθορισμό προδιαγραφών.

3.3 Συσκευές, υπηρεσίες και σημεία ελέγχου

Τα βασικά συστατικά μέρη ενός UPnP δικτύου είναι τρία· οι συσκευές, οι υπηρεσίες και τα σημεία ελέγχου. Στις ακόλουθες παραγράφους γίνεται μια σύντομη περιγραφή για αυτά.

3.3.1 Συσκευές

Μια UPnP συσκευή περιέχει υπηρεσίες και εμφωλευμένες συσκευές. Διαφορετικές κατηγορίες συσκευών σχετίζονται με διαφορετικά σύνολα υπηρεσιών και ενσωματωμένων συσκευών. Συνεπώς, διαφορετικές επιτροπές εργασίας του UPnP Forum καθορίζουν το σύνολο των υπηρεσιών που παρέχει κάποια συγκεκριμένη συσκευή. Όλη αυτή η πληροφορία καταγράφεται σε ένα XML αρχείο, περιγραφή της συσκευής, το οποίο πρέπει να διαθέτει η συσκευή. Εκτός των υπηρεσιών που παρέχονται, στο αρχείο αυτό καταγράφονται και άλλα χαρακτηριστικά της συσκευής, όπως το όνομα και το εικονίδιο που την περιγράφει.

3.3.2 Υπηρεσίες

Η μικρότερη μονάδα ελέγχου σε ένα UPnP δίκτυο είναι η υπηρεσία. Μια υπηρεσία εκθέτει ενέργειες (actions) και μοντελοποιεί την κατάστασή της με τη χρήση μεταβλητών. Παρόμοια με την περιγραφή της συσκευής, η πληροφορία αυτή είναι μέρος ενός XML αρχείου, το οποίο καθορίζεται από το UPnP Forum. Ένας δείκτης (URL) για την περιγραφή κάθε υπηρεσίας, που παρέχει μια συσκευή, περιέχεται στο αρχείο περιγραφής της συσκευής. Μια συσκευή μπορεί να παρέχει περισσότερες της μιας υπηρεσίες.

Μια υπηρεσία UPnP συσκευής αποτελείται από έναν πίνακα κατάστασης και δύο εξυπηρετητές, ελέγχου και γεγονότων (events). Ο πίνακας κατάστασης μοντελοποιεί την κατάσταση της υπηρεσίας, μέσω των ανάλογων μεταβλητών, τις οποίες και ενημερώνει όταν αυτή αλλάζει. Ο εξυπηρετητής ελέγχου λαμβάνει αιτήσεις για την εκτέλεση ενεργειών και τις εκτελεί. Παράλληλα, ενημερώνει για την νέα κατάσταση τον προαναφερθέντα πίνακα και επιστέφει στο αιτούντα τα αποτελέσματα

της εκτέλεσης. Ο εξυπηρετητής γεγονότων ανακοινώνει τα γεγονότα στους ενδιαφερόμενους συνδρομητές της υπηρεσίας, όταν η κατάσταση της τελευταίας αλλάξει.

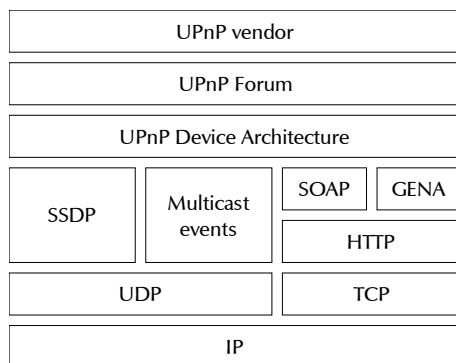
3.3.3 Σημεία ελέγχου

Ένα σημείο ελέγχου σε ένα δίκτυο UPnP είναι ένας ελεγκτής, ικανός να εντοπίζει και να ελέγχει τις άλλες συσκευές στο δίκτυο. Αφού ανακαλύψει τις συσκευές που υπάρχουν σε ένα δίκτυο, ένα σημείο ελέγχου μπορεί να:

- ▶ ανακτά τη περιγραφή μια συσκευής και την λίστα των υπηρεσιών που αυτή παρέχει,
- ▶ ανακτά τις περιγραφές των υπηρεσιών από τις συσκευές που το ενδιαφέρουν,
- ▶ να επικαλείται την εκτέλεση ενεργειών ελέγχοντας τις υπηρεσίες και
- ▶ να εγγράφεται ως συνδρομητής σε υπηρεσίες και κάθε φορά που αλλάζει η κατάσταση κάποιας από αυτές, λαμβάνει ένα μήνυμα από τον εξυπηρετητή γεγονότων της.

3.4 Πρωτόκολλα που χρησιμοποιούνται στο UPnP

Η αρχιτεκτονική του UPnP περιλαμβάνει προϋπάρχουσες τεχνολογίες που χρησιμοποιούν πρωτόκολλα όπως τα IP, TCP, UDP, HTTP και την XML.



Σχήμα 3.2: Στοιβά πρωτοκόλλων που χρησιμοποιεί το UPnP.

Η χρήση των Internet πρωτοκόλλων εξασφαλίζει την λειτουργία σε οποιοδήποτε φυσικό μέσο και μεταξύ διαφορετικών κατασκευαστών. Επιπλέον, η χρήση των πρωτοκόλλων αυτών προσφέρει τη δυνατότητα συνεργασίας με άλλα προϋπάρχοντα δίκτυα, όπως το Internet και πολλά intranets, τα οποία υπάρχουν σε οικιακούς και εργασιακούς χώρους.

Ακόμα, με κατάλληλη γεφύρωση η UDA μπορεί να συμπεριλάβει και μέσα, τα οποία δεν είναι σε θέση να χρησιμο-

ποιήσουν IP πρωτόκολλα λόγω κόστους ή τεχνολογίας.

Στο Σχήμα 3.2 παρουσιάζεται η στοιβά πρωτοκόλλων που χρησιμοποιεί η UPnP αρχιτεκτονική. Η χρήση μερικών από των πρωτοκόλλων αυτών στο UPnP περιγράφεται στη συνέχεια.

3.4.1 Πρωτόκολλα καθορισμένα από το UPnP

Οι κατασκευαστές των UPnP συσκευών, οι UPnP Forum επιτροπές εργασίας και η UPnP αρχιτεκτονική συσκευής καθορίζουν τα υψηλότερα επίπεδα της στοίβας και χρησιμοποιούνται για την υλοποίηση του UPnP. Βασιζόμενες στην αρχιτεκτονική της συσκευής, οι επιτροπές εργασίας καθορίζουν την πληροφορία για συγκεκριμένους τύπους συσκευών. Στη συνέχεια οι κατασκευαστές συσκευών καθορίζουν τα δεδομένα για τα προϊόντα τους, όπως το μοντέλο και το όνομα της συσκευής, τα URLs για τα Web sites τους κ.ά..

3.4.2 TCP/IP

Η TCP/IP στοίβα πρωτοκόλλων αποτελεί την βάση, πάνω στην οποία λειτουργούν τα υπόλοιπα πρωτόκολλα. Χρησιμοποιώντας αυτή τη δοκιμασμένη και επικρατούσα σουίτα πρωτοκόλλων, το UPnP επιτρέπει την δυνατότητα λειτουργίας σε πληθώρα φυσικών μέσων, ενώ παράλληλα εξασφαλίζει την δια-λειτουργικότητα μεταξύ των προϊόντων πολλών κατασκευαστών.

Οι UPnP συσκευές μπορούν να χρησιμοποιήσουν πολλά από τα πρωτόκολλα της στοίβας TCP/IP, όπως το TCP, UDP, IGMP, ARP και το IP, καθώς επίσης και υπηρεσίες όπως τη DHCP και DNS. Από τη στιγμή που το TCP/IP είναι από τα πλέον χρησιμοποιούμενα πρωτόκολλα, η κατασκευή μιας UPnP συσκευής γίνεται με σχετικά απλό και εύκολο τρόπο.

3.4.3 HTTP, HTTPU και HTTPMU

Το HTTP [FGM⁺99] είναι ένα από τα κυριότερα μέρη της δικτύωσης UPnP. Υπεύθυνο για την μεγάλη επιτυχία του Internet, το HTTP αποτελεί τη βάση σε όλες της πτυχές του UPnP.

Το HTTPU και το HTTPMU αποτελούν παραλλαγές του HTTP οι οποίες χρησιμοποιούν το UDP/IP για την μεταφορά των μηνυμάτων, αντί του TCP/IP που χρησιμοποιεί η τελευταία. Αυτά τα πρωτόκολλα χρησιμοποιούνται από το SSDP (Simple Service Discovery Protocol), το οποίο περιγράφεται παρακάτω. Η βασική μορφή των μηνυμάτων που χρησιμοποιούν τα πρωτόκολλα αυτά είναι όμοια με αυτή του HTTP, ενώ χρησιμοποιούνται για μετάδοση multicast, ή όταν απαιτείται η γρήγορη μετάδοση, και παράλληλα δεν υπάρχουν περιορισμοί για την αξιοπιστία.

3.4.4 SSDP

Το SSDP [Yar99] είναι ένα πρωτόκολλο σχεδιασμένο από τη Microsoft και την Hewlett-Packard και αποτελεί το βασικό πρωτόκολλο ανακάλυψης για το UPnP.

Το πρωτόκολλο αυτό καθορίζει τον τρόπο με τον οποίο ανιχνεύονται οι δικτυακές υπηρεσίες. Χρησιμοποιεί τα HTTPU και HTTPMU και καθορίζει τις μεθόδους με τις οποίες τα σημεία ελέγχου εντοπίζουν τους πόρους του δικτύου, για τους οποίους ενδιαφέρονται, και πώς οι συσκευές ανακοινώνουν την διαθεσιμότητά τους στο δίκτυο. Καθορίζοντας αμφότερα τα αιτήματα αναζήτησης και τις ανακοινώσεις παρουσίας, το SSDP εξαλείφει το overhead, το οποίο θα απαιτούνταν αν χρησιμοποιούνταν μόνο ο ένας από τους δύο αυτούς μηχανισμούς. Ως αποτέλεσμα, κάθε σημείο ελέγχου στο δίκτυο έχει την πλήρη γνώση για την κατάσταση του δικτύου, ελαχιστοποιώντας την κίνηση σ' αυτό.

Αμφότερα τα σημεία ελέγχου και οι συσκευές χρησιμοποιούν το SSDP. Ένα σημείο ελέγχου, όταν ενεργοποιείται μπορεί να στείλει ένα αίτημα αναζήτησης, ώστε να ανιχνεύσει τις συσκευές και τις υπηρεσίες που υπάρχουν στο δίκτυο. Έχει επίσης τη δυνατότητα να περιορίσει την αναζήτηση σε συσκευές ή υπηρεσίες συγκεκριμένου τύπου, ή ακόμα να αναζητήσει κάποια συγκεκριμένη συσκευή.

Οι συσκευές UPnP λαμβάνουν τις αιτήσεις των σημείων ελέγχου σε ένα καθορισμένο multicast port. Με τη λήψη ενός μηνύματος, η συσκευή εξετάζει τα κριτήρια αναζήτησης και αν συμπίπτουν με τα χαρακτηριστικά της τότε απαντά με ένα unicast SSDP μήνυμα στο σημείο ελέγχου.

Με παρόμοιο τρόπο, όταν μια συσκευή συνδέεται στο δίκτυο στέλνει πολλαπλές SSDP ανακοινώσεις για την παρουσία της, προβάλλοντας παράλληλα και τις υπηρεσίες που υποστηρίζει.

Οι ανακοινώσεις παρουσίας και οι απαντήσεις στα αιτήματα αναζήτησης περιέχουν ένα δείκτη στο αρχείο περιγραφής της συσκευής, το οποίο περιέχει πληροφορία σχετική με τα χαρακτηριστικά της συσκευής και τις υπηρεσίες που αυτή υποστηρίζει.

Επιπλέον των δυνατοτήτων ανακάλυψης, το SSDP παρέχει επίσης το τρόπο ώστε μια συσκευή και οι υπηρεσίες που σχετίζονται με αυτή να μπορούν να αποσυνδεθούν από το δίκτυο ομαλά, ενώ θέτει χρονικά όρια σ' αυτές, επιτρέποντας την αυτο-ίαση του δικτύου σε περίπτωση σφάλματος. Για την επίτευξη του τελευταίου οι συσκευές στέλνουν τις ανακοινώσεις παρουσίας τους περιοδικά, ώστε αυτές να είναι πάντα έγκυρες.

3.4.5 GENA

Η Generic Event Notification Architecture (GENA) [CA98] δημιουργήθηκε με στόχο να προσφέρει τη δυνατότητα αποστολής και λήψης ανακοινώσεων χρησιμοποιώντας το HTTP πάνω από TCP/IP και multicast UDP. Χρησιμοποιεί το μοντέλο συνδρομητή/εκδότη (subscriber/publisher), επιτρέποντας την ανίχνευση και μετάδοση γεγονότων.

Στο UPnP, το GENA χρησιμοποιείται για την δημιουργία των ανακοινώσεων παρουσίας συσκευών οι οποίες στέλνονται με το SSDP. Παρέχει επίσης τη δυνατότητα σήμανσης των αλλαγών στην κατάσταση κάποιας υπηρεσίας για την ειδοποίηση κάποιου γεγονότος. Ένα σημείο ελέγχου, ενδιαφερόμενο στο να λαμβάνει τις ειδοποιήσεις για κάποιο γεγονός, μπορεί να γίνεται συνδρομητής σε κάποια συσκευή, στέλνοντας ένα αίτημα εγγραφής με την υπηρεσία που το ενδιαφέρει, τη τοποθεσία στην οποία θα στέλνονται οι ειδοποιήσεις και την χρονική στιγμή της συνδρομής. Οι συνδρομές αυτές πρέπει να ενημερώνονται περιοδικά ώστε το σημείο ελέγχου να συνεχίζει να λαμβάνει τις ειδοποιήσεις, ενώ παρέχεται επίσης και η δυνατότητα ακύρωσης κάποιας συνδρομής.

3.4.6 SOAP

To Simple Object Access Protocol (SOAP) [Dev00] καθορίζει τη συνδυασμένη χρήση της XML και του HTTP με σκοπό την απομακρυσμένη κλήση διαδικασιών (RPC – Remote Procedure Calls). Το SOAP έχει καθιερωθεί ως πρωτόκολλο για την RPC επικοινωνία στο Internet. Χρησιμοποιώντας την υπάρχουσα δομή του Internet, λειτουργεί αποδοτικά και αποτελεσματικά με firewalls και proxies. Επίσης, μπορεί να χρησιμοποιήσει το Secure Socket Layer (SSL) προσφέροντας μεγαλύτερη ασφάλεια, αλλά και τις ευκολίες διαχείρισης σύνδεσης του HTTP, καθιστώντας την καταναεμμένη επικοινωνία μέσω Internet τόσο απλή, όσο και η προσπέλαση των ιστοσελίδων.

Όπως και στη περίπτωση της RPC, το UPnP χρησιμοποιεί το SOAP με σκοπό το έλεγχο των συσκευών. Τα σημεία ελέγχου στέλνουν κατάλληλα μηνύματα προς τις συσκευές και αυτές επιστρέφουν τα αποτελέσματα ή τα πιθανά σφάλματα της εκτέλεσης κάποιας ενέργειας πίσω στα σημεία ελέγχου.

Κάθε μήνυμα ελέγχου αποτελεί ένα SOAP μήνυμα, το οποίο περιέχει την προς εκτέλεση ενέργεια μαζί με το σύνολο των παραμέτρων της. Η απόκριση της εκτέλεσης είναι και αυτή ένα μήνυμα SOAP, που περιέχει την κατάσταση της υπηρεσίας, η οποία εκτέλεσε την ενέργεια, την επιστρεφόμενη τιμή και οποιαδήποτε άλλη παράμετρο χρειάζεται.

3.5 Πώς λειτουργεί το UPnP;

Το UPnP είναι υπεύθυνο στο να παρέχει υποστήριξη στην επικοινωνία μεταξύ των σημείων ελέγχου και των συσκευών. Το μέσο δικτύωσης, το TCP/IP και το HTTP παρέχουν τη βασική δικτύωση και διευθυνσιοδότηση που χρειάζεται. Πάνω από αυτά τα ανοικτά, καθιερωμένα Internet πρωτόκολλα, το UPnP καθορίζει ένα σύνολο από HTTP εξυπηρετητές, οι οποίοι χειρίζονται την ανακάλυψη, την περιγραφή, τον έλεγχο, την ειδοποίηση γεγονότων και την παρουσίαση των συσκευών.

Η UPnP αρχιτεκτονική συσκευής (UDA – UPnP Device Architecture) [UPnP08] ορίζει το σχήμα – περίγραμμα (template) για τη δημιουργία των συσκευών και των υπηρεσιών. Μεμονωμένες επιτροπές εργασίας, βάσει του περιγράμματος αυτού, καθορίζουν τους διάφορους τύπους συσκευών και υπηρεσιών και στη συνέχεια σχηματίζουν ένα περίγραμμα για κάθε έναν από τους τύπους αυτούς. Στο τέλος, ο κατασκευαστής της συσκευής συμπληρώνει το περίγραμμα αυτό με τη πληροφορία που σχετίζεται με το προϊόν του. Η πληροφορία συνήθως περιέχει το όνομα και το μοντέλο της συσκευής, το σειριακό αριθμό της, το όνομα του κατασκευαστή και κάποιο URL για την περιγραφή της. Τα δεδομένα αυτά ενθυλακώνονται στα UPnP πρωτόκολλα σε ένα αρχείο XML.

Η απαιτούμενη πληροφορία για το UPnP εισάγεται σε όλα τα μηνύματα, πριν αυτά μορφοποιηθούν χρησιμοποιώντας κάποιο από τα SSDP, GENA και SOAP πρωτόκολλα και στη συνέχεια παραδίδονται μέσω του HTTP, HTTPU ή HTTPMU.

Στις επόμενες υποενότητες περιγράφονται εν συντομία τα βήματα της UPnP δικτύωσης. Για περισσότερες τεχνικές λεπτομέρειες ο αναγνώστης παραπέμπεται στο [UPnP08], όπου αναλύονται σε περισσότερο βάθος τα βήματα λειτουργίας του UPnP.

3.5.1 Διευθυνσιοδότηση

Τη βάση για τη δικτύωση UPnP αποτελεί η σουίτα πρωτοκόλλων TCP/IP και πιο συγκεκριμένα η διευθυνσιοδότηση IP. Κάθε συσκευή ή σημείο ελέγχου διαθέτει ένα Dynamic Host Configuration Protocol (DHCP) [Dro97] client και όταν συνδέεται για πρώτη φορά στο δίκτυο, αναζητά κάποιο DHCP server, ώστε να τους ανατεθεί μια IP διεύθυνση. Αν στο δίκτυο δεν υπάρχει κάποιος DHCP server, τότε οι συσκευές και οι ελεγκτές χρησιμοποιούν το πρωτόκολλο Auto-IP για να αποκτήσουν διεύθυνση IP.

Εν συντομία, το Auto-IP [S. 05] καθορίζει το τρόπο με τον οποίο μία συσκευή ή ένα σημείο ελέγχου μπορεί να επιλέξει μία διεύθυνση IP από ένα δεσμευμένο σύνολο διευθύνσεων, καθώς επίσης και τον τρόπο με τον οποίο μπορεί να μετακινείται ανάμεσα σε οργανωμένα και ανοργάνωτα δίκτυα.

Στην περίπτωση που υπάρχει DHCP server και επιπλέον η συσκευή ή το σημείο ελέγχου διαθέτει κάποιο domain name, τότε χρησιμοποιεί το όνομα αυτό για τις επόμενες δικτυακές λειτουργίες. Σε οποιαδήποτε άλλη περίπτωση χρησιμοποιεί την διεύθυνση IP.

3.5.2 Ανακάλυψη συσκευών

Από τη στιγμή που οι συσκευές προσαρτηθούν στο δίκτυο και διευθυνσιοδοτηθούν κανονικά, ξεκινά η διαδικασία ανακάλυψης. Το SSDP είναι το πρωτόκολλο που αναλαμβάνει την διαδικασία αυτή. Το SSDP επιτρέπει στις συσκευές, όταν αυτές

συνδέονται στο δίκτυο, να ανακοινώνουν τις υπηρεσίες τους στα σημεία ελέγχου. Από την άλλη, όταν συνδέεται ένα σημείο ελέγχου, το SSDP αναλαμβάνει να εντοπίσει τις συσκευές και τις υπηρεσίες ενδιαφέροντος, που υπάρχουν στο δίκτυο.

Η διαδικασία και στις δύο περιπτώσεις γίνεται με την ανταλλαγή μηνυμάτων ανακάλυψης (discovery messages), στα οποία περιέχονται κάποια κρίσιμα, βασικά χαρακτηριστικά, που έχουν να κάνουν με την συσκευή ή κάποια από τις υπηρεσίες της, όπως για παράδειγμα, ο τύπος της συσκευής, κάποιο αναγνωριστικό και ίσως κάποιος δείκτης για το XML αρχείο περιγραφής της συσκευής.

3.5.3 Περιγραφή συσκευών

Το επόμενο βήμα στη δικτύωση UPnP είναι η περιγραφή. Αφού ένα σημείο ελέγχου έχει ανακαλύψει μια συσκευή, γνωρίζει λίγα γι αυτή. Για να αποκομίσει περισσότερες πληροφορίες για τη συσκευή αυτή και τις δυνατότητές της, ώστε να επιτευχθεί η αλληλεπίδραση τους, ο ελεγκτής πρέπει να αποκτήσει την περιγραφή της συσκευής. Η περιγραφή αυτή είναι διαθέσιμη από το URL της συσκευής, το οποίο γνωστοποιείται στον ελεγκτή στο προηγούμενο βήμα, κατά την ανακάλυψη της συσκευής.

Οι συσκευές, όπως έχει ήδη αναφερθεί, μπορούν να περιέχουν άλλες λογικές συσκευές και υπηρεσίες. Η περιγραφή UPnP για μια συσκευή εκφράζεται σε XML και περιέχει πληροφορίες από τον κατασκευαστή, όπως το μοντέλο και τον σειριακό αριθμό της συσκευής, το όνομα και URLs για τα Web sites του κατασκευαστή κ.ά.. Επίσης, η περιγραφή περιλαμβάνει μία λίστα με τις ενσωματωμένες συσκευές και υπηρεσίες, καθώς και τα URLs για τον έλεγχο, την ειδοποίηση και την παρουσίαση των τελευταίων.

3.5.4 Έλεγχος συσκευών

Έπειτα από την ανάκτηση της περιγραφής της συσκευής, ένα σημείο ελέγχου είναι σε θέση, γνωρίζοντας πλέον τα απαραίτητα, να ελέγξει την λειτουργία της συσκευής αυτής. Για να μάθει περισσότερα για μια υπηρεσία, το σημείο ελέγχου πρέπει να ανακτήσει μια λεπτομερέστερη περιγραφή για την κάθε υπηρεσία της συσκευής. Η περιγραφή μιας υπηρεσίας εκφράζεται και αυτή σε XML και περιλαμβάνει την λίστα με τις εντολές (ή ενέργειες), αλλά και τις παραμέτρους για κάθε μία από αυτές, καθώς επίσης και το πώς αυτή αποκρίνεται. Επιπλέον, η περιγραφή μιας υπηρεσίας περιέχει και μια λίστα από μεταβλητές. Αυτές οι μεταβλητές μοντελοποιούν την κατάσταση της υπηρεσίας κατά την διάρκεια λειτουργίας της και περιγράφονται βάσει του τύπου δεδομένων τους, το εύρος τιμών και τα χαρακτηριστικά του γεγονότος το οποίο αντιπροσωπεύουν.

Ο έλεγχος μιας συσκευής από ένα σημείο ελέγχου επιτυγχάνεται με την αποστολή ενός μηνύματος από το τελευταίο προς την υπηρεσία της συσκευής. Το κατάλληλο

αυτό μήνυμα ελέγχου στέλνεται στο URL ελέγχου της υπηρεσίας, το οποίο παρέχεται στο σημείο ελέγχου με την περιγραφή της συσκευής. Τα μηνύματα ελέγχου εκφράζονται σε XML χρησιμοποιώντας το SOAP. Ως απάντηση σε ένα μήνυμα ελέγχου, η υπηρεσία επιστρέφει τις ανάλογες με την εκτέλεση της εντολής τιμές ή κάποιο μήνυμα σφάλματος, στην περίπτωση που σημειωθεί κάποια αποτυχία.

3.5.5 Ειδοποίηση γεγονότων

Όπως προαναφέρθηκε, η περιγραφή μιας UPnP υπηρεσίας περιλαμβάνει μια λίστα με τις ενέργειες στις οποίες η υπηρεσία αυτή απαντά και τη λίστα με τις μεταβλητές, οι οποίες μοντελοποιούν την κατάσταση της υπηρεσίας κατά τη διάρκεια λειτουργίας της. Η υπηρεσία δημοσιεύει ενημερώσεις όταν αλλάξει κάποια από τις μεταβλητές αυτές και τα σημεία ελέγχου μπορούν να εγγραφούν συνδρομητές ώστε να λαμβάνουν αυτή την πληροφορία.

Η υπηρεσία δημοσιεύει τις ενημερώσεις στέλνοντας μηνύματα ειδοποίησης. Τα μηνύματα ειδοποίησης περιέχουν τα ονόματα ενός ή περισσότερων μεταβλητών κατάστασης και την τρέχουσα τιμή τους και εκφράζονται με XML χρησιμοποιώντας το GENA.

Όταν ένα σημείο ελέγχου επιθυμεί να ειδοποιείται, γίνεται συνδρομητής στην υπηρεσία, που το ενδιαφέρει και λαμβάνει ένα ειδικό αρχικό μήνυμα ειδοποίησης. Αυτό το ειδικό μήνυμα περιέχει τα ονόματα και τις τιμές όλων των μεταβλητών, επιτρέποντας στο σημείο ελέγχου να αρχικοποιήσει το δικό του μοντέλο για την κατάσταση της υπηρεσίας.

Για να είναι εφικτή η υποστήριξη ειδοποίησης πολλαπλών σημείων ελέγχου, η διαδικασία ειδοποίησης είναι σχεδιασμένη με τέτοιο τρόπο ώστε να ενημερώνει όλα τα σημεία ελέγχου για τα αποτελέσματα εκτέλεσης κάποιας ενέργειας. Επομένως, όλοι οι συνδρομητές λαμβάνουν μηνύματα ειδοποίησης για όλες τις μεταβλητές που άλλαξαν ανεξαρτήτως του λόγου της αλλαγής.

Το UPnP υποστηρίζει και τη multicast ειδοποίηση. Με την χρήση multicast τα σημεία ελέγχου μπορούν να παρακολουθούν τις αλλαγές κατάστασης των υπηρεσιών χωρίς την υποχρεωτική συνδρομή σε αυτές. Αυτός ο τρόπος ειδοποίησης είναι πολύ χρήσιμος όταν γεγονότα, τα οποία δεν είναι σχετικά με συγκεκριμένες UPnP αλληλεπιδράσεις, πρέπει να παραδοθούν στα σημεία ελέγχου, ώστε να ενημερωθούν οι χρήστες. Επίσης, αυτός ο τρόπος μετάδοσης είναι χρήσιμος, όταν πολλές συσκευές πρέπει να ειδοποιηθούν περισσότερους από έναν ελεγκτές. Βέβαια, το multicast θεωρείται αναξιόπιστο, αφού βασίζεται στο πρωτόκολλο UDP. Για να μειωθεί η πιθανότητα εσφαλμένης μετάδοσης, οι ειδοποιήσεις μπορούν να μεταδίδονται περισσότερες φορές.

3.5.6 Παρουσίαση συσκευών

Αν μια συσκευή διαθέτει ένα URL για την παρουσίασή της, ένα σημείο ελέγχου μπορεί να ανακτήσει μία σελίδα από αυτό το URL και να τη φορτώσει σε ένα browser. Ανάλογα με τις δυνατότητες της σελίδας αυτής και της συσκευής, μπορεί να επιτρέπεται στο χρήστη ο έλεγχος και η παρατήρηση της κατάστασης της συσκευής.

Κεφάλαιο 4

UPnP και Ασύρματα Δίκτυα Αισθητήρων

Καθώς η χρήση των ασύρματων δικτύων αισθητήρων αυξάνεται συνεχώς και καθώς εισέρχονται όλο και περισσότερο στην καθημερινότητα, προκύπτει η ανάγκη για την εύρεση και δημιουργία απλών τρόπων διαχείρισής τους. Οι χρήστες των δικτύων αυτών βρίσκονται αντιμέτωποι με προβλήματα, που σχετίζονται με τη ρύθμιση και τον έλεγχο ενός μεγάλου πλήθους συσκευών (αισθητήρων), με αποτέλεσμα να απαιτείται ειδική γνώση και χρόνος, ώστε να είναι εφικτή η διαχείρισή τους. Εφόσον δεν έχει καθοριστεί κάποια στάνταρντ τεχνολογία για τη διαχείριση των δικτύων αισθητήρων, η δημιουργία ενός γενικού εργαλείου για το σκοπό αυτό είναι πλέον επιτακτική.

Μια επιτυχής υλοποίηση μιας εφαρμογής ελέγχου δικτύων αισθητήρων απαιτεί μια συμφωνία όχι μόνο στο επίπεδο δικτύωσης αλλά και σε εκείνο που έχει να κάνει με την σημασιολογία της συσκευής και της λειτουργίας της. Αυτό σημαίνει πως πρέπει να δημιουργηθούν κάποιοι στάνταρντ μηχανισμοί για την ανακάλυψη, περιγραφή λειτουργίας των κόμβων (αισθητήρων και ενεργοποιητών), καθώς και μεθόδων για την πρόσβαση στις λειτουργίες αυτών. Αυτά τα θέματα καλύπτονται από μια middleware πλατφόρμα. Όσες τέτοιες πλατφόρμες έχουν σχεδιαστεί ως τώρα, η σχεδίαση τους έχει γίνει έχοντας ως στόχο τους υπολογιστές. Επομένως, η επεξεργασία των πρωτοκόλλων που αυτές ορίζουν είναι ενεργειακά απαιτητική ή ακόμα και αδύνατη από τις συσκευές χαμηλού κόστους όπως είναι οι κόμβοι των αισθητήρων. Μια middleware πλατφόρμα για την αδιάλειπτη αλληλεπίδραση των κόμβων πρέπει να υποστηρίζει την ad-hoc δικτύωση, χωρίς την ανάγκη χειροκίνητης εγκατάστασης και να είναι αρκετά ευέλικτη, ώστε να ενσωματώνεται στις χαμηλού κόστους συσκευές αισθητήρων.

Η CORBA [cor] είναι μία καθολική αρχιτεκτονική, η οποία επιτρέπει σε μια client εφαρμογή να καλεί συναρτήσεις ενός απομακρυσμένου server. Ο πυρήνας ενός συστήματος CORBA (ORB – Object Request Broker) μεσολαβεί για την επικοινωνία μεταξύ client και server. Όμως η αρχιτεκτονική αυτή αποτελεί μια πολύ δύσκολη πλατφόρμα για την εφαρμογή της σε συσκευές αισθητήρων.

Το JXTA [jxt] στοχεύει στην απλοποίηση της ρύθμισης και λειτουργίας των peer-to-peer δικτύων. Παρέχει τους μηχανισμούς για την ανακάλυψη υπηρεσιών και ένα πλαίσιο από XML πρωτόκολλα για τους κόμβους του δικτύου. Είναι σχεδιασμένο για απλά και μικρά συστήματα, αν και το XML parsing απαιτεί κάποια υπολογιστική ισχύ. Βασικά, το JXTA προσφέρει την δυνατότητα μεταφοράς δεδομένων και εκτελέσιμου κώδικα μεταξύ των συστημάτων, αλλά υστερεί σε στάνταρντς για την εκτέλεση κώδικα.

Η UPnP αρχιτεκτονική, επιτρέποντας τη zero-configuration δικτύωση, την αυτόματη ανακάλυψη και τον έλεγχο συσκευών, θα αποτελούσε μια καλή λύση για το πρόβλημα διαχείρισης των δικτύων αισθητήρων. Όμως, η τεχνολογία αυτή δεν είναι κατάλληλη για την υλοποίησή της σε μικρές συσκευές, όπως είναι αυτές των αισθητήρων. Πρώτον, η λειτουργία της βασίζεται στο TCP/IP, το οποίο δεν υλοποιείται σχεδόν από καμία ασύρματη συσκευή τέτοιου είδους. Δεύτερον, οι απαιτήσεις σε υπολογιστική ισχύ –και κατά συνέπεια σε κατανάλωση ενέργειας– για την λειτουργία μιας τέτοιας αρχιτεκτονικής είναι μεγάλες, και αφού οι συσκευές των αισθητήρων είναι περιορισμένων υπολογιστικών δυνατοτήτων και ενέργειας, κάτι τέτοιο είναι πρακτικά μη υλοποιήσιμο.

Για την επίλυση του προβλήματος της διαχείρισης των δικτύων αισθητήρων, έχουν προταθεί κατά καιρούς πολλά μοντέλα, τα οποία συνδυάζουν την ευκολία της χρήσης του UPnP και τα δίκτυα αισθητήρων. Τα περισσότερα βασίζονται στην ιδέα της γεφύρωσης των διαφορετικών δικτύων, μέσω ειδικών κόμβων που αναλαμβάνουν την διαχείριση ενός δικτύου αισθητήρων από UPnP σημεία ελέγχου. Η ιδέα της γεφύρωσης μη συμβατών δικτύων με UPnP δίκτυα είναι κάτι το οποίο προτείνεται και από το UPnP Forum, όταν το κόστος ή τεχνολογία των συσκευών δεν επιτρέπει την υλοποίηση του UPnP στις συσκευές αυτές [UPn08, σελ. 1].

Στις ενότητες που ακολουθούν πραγματοποιείται μια σύντομη περιγραφή εργασιών που έχουν γίνει κατά το παρελθόν, οι οποίες έχουν ως στόχο την παροχή υπηρεσιών UPnP από τα δίκτυα αισθητήρων.

4.1 BOSS – Bridge Of SensorS

Στο [KLSS05] προτείνεται μια κεντριοποιημένη αρχιτεκτονική η οποία βασίζεται στο UPnP για τη διαχείριση των δικτύων αισθητήρων. Όμως το UPnP μπορεί να εκτελείται μόνο σε συσκευές ισχυρότερων επεξεργαστικά και με περισσότερη μνήμη, καθιστώντας τους κόμβους ενός WSN ανίκανους να επεξεργαστούν το πρωτόκολλο αυτό. Για το λόγο αυτό οι συγγραφείς υλοποιούν ένα ειδικό κόμβο τον οποίο καλούν Bridge Of SensorS (BOSS), ο οποίος είναι υπεύθυνος για την γεφύρωση μεταξύ ενός WSN με το δίκτυο UPnP. Το προτεινόμενο σύστημα αποτελείται από τρία βασικά μέρη: ένα UPnP σημείο ελέγχου, τον κόμβο BOSS και τους κόμβους του WSN. Το σημείο ελέγχου αποτελεί μια υπολογιστικά ισχυρή συσκευή, η οποία

διαθέτει τους απαιτούμενους πόρους ώστε να εκτελεί το UPnP και να δύναται να διαχειρίζεται το WSN μέσω του BOSS. Ένα σημείο ελέγχου μπορεί να είναι ένα PC, PDA ή notebook. Ο κόμβος BOSS είναι ένα σταθμός βάσης ο οποίος λειτουργεί ως διαμεσολαβητής μεταξύ των κόμβων του WSN και του σημείου ελέγχου.

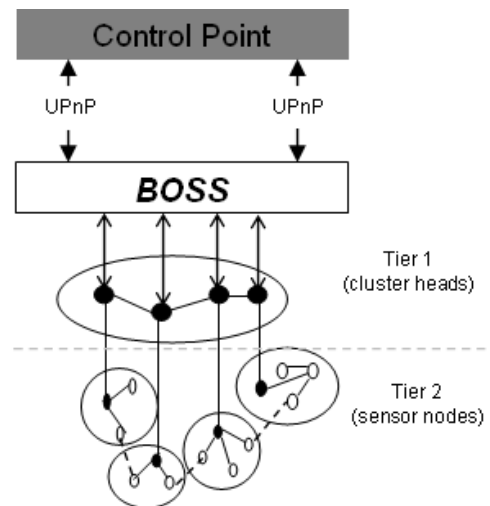
Το σημείο ελέγχου και ο κόμβος BOSS επικοινωνούν μεταξύ τους χρησιμοποιώντας το UPnP πρωτόκολλο, ενώ η επικοινωνία μεταξύ των κόμβων του WSN και του BOSS πραγματοποιείται μέσω του πρωτοκόλλου που ορίζουν οι πρώτοι.

Ο κόμβος BOSS προσφέρει τρεις λειτουργίες. Πρώτον, χρησιμοποιείται για την μεταφορά των UPnP μηνυμάτων μεταξύ των αισθητήρων και του σημείου ελέγχου. Δεύτερον, εφόσον το UPnP πρωτόκολλο χρησιμοποιεί την XML για την δόμηση των μηνυμάτων, και αυτή διαφέρει από τη μορφή των μηνυμάτων από τους αισθητήρες, ο BOSS αναλαμβάνει την μετάφραση των μηνυμάτων αυτών. Τέλος, συγκεντρώνει πληροφορία σχετική με την διαχείριση του δικτύου από τους αισθητήρες. Η πληροφορία αυτή χρησιμοποιείται ώστε να παρέχονται στο σημείο ελέγχου επιπλέον υπηρεσίες που αφορούν το δίκτυο συνολικά. Ο κόμβος BOSS αποθηκεύει την πληροφορία αυτή και εκτελεί όλες τις απαραίτητες λειτουργίες για τη διαχείριση του WSN, χωρίς να επιβαρύνει του κόμβους του τελευταίου με απαιτητικού υπολογισμού.

Ένας κόμβος BOSS διαθέτει ένα πίνακα με τις υπηρεσίες κάθε τύπου κόμβου και ένα πίνακα με τις υπηρεσίες που παρέχει αυτός. Επίσης, διαθέτει ένα διαχειριστή υπηρεσιών, ο οποίος αναλαμβάνει τον εντοπισμό των υπηρεσιών που παρέχονται από τους αισθητήρες καθώς και την αποστολή των UPnP περιγραφών τους στα σημεία ελέγχου. Διαθέτει ακόμα δυο διαχειριστές για τον έλεγχο και την ειδοποίηση γεγονότων, αντίστοιχα.

Το σημείο ελέγχου μπορεί να ορίσει για ποια συμβάντα που γίνονται αντιληπτά από τους κόμβους του WSN ενδιαφέρεται. Ο ενδιάμεσος κόμβος αναλαμβάνει να επεξεργαστεί τα αναφερόμενα συμβάντα από τους αισθητήρες και ειδοποιεί ανάλογα το σημείο ελέγχου.

Οι υπηρεσίες δικτύου που προσφέρει το BOSS περιλαμβάνουν τη βασική πληροφόρηση για το δίκτυο, τον εντοπισμό, συγχρονισμό των κόμβων και την διαχείριση ενέργειας. Ο BOSS έχει τη δυνατότητα να ανακτά τη πληροφορία για την κατάσταση του δικτύου από τους αισθητήρες, καθώς και την Περιγραφή Συσκευής των



Σχήμα 4.1: Βασική διασύνδεση μέσω ενός BOSS κόμβου. Το Σχήμα εμφανίζεται στο [LDCO07].

αισθητήρων, το πλήθος τους και την τοπολογία του δικτύου.

Η υπηρεσία εντοπισμού παρέχει πληροφορία σχετική με την τοποθέτηση κάθε αισθητήρα στο δίκτυο, ενώ η υπηρεσία συγχρονισμού είναι υπεύθυνη για τον συγχρονισμό των ρολογιών των κόμβων του WSN. Για παράδειγμα, όταν ένας αισθητήρας προστίθεται στο δίκτυο στέλνει ένα μήνυμα στο BOSS και αυτός στη συνέχεια ενημερώνει το σημείο ελέγχου. Έπειτα, ο BOSS συγχρονίζει τον νέο κόμβο με τους υπόλοιπους του δικτύου.

Η υπηρεσία διαχείρισης ενέργειας επιτρέπει στο χρήστη να διαχειρίζεται την κατανάλωση ενέργειας κάθε αισθητήρα, ελέγχοντας την κατάσταση της μπαταρίας τους και αλλάζοντας το τρόπο λειτουργίας τους στοχεύοντας στην εξοικονόμηση ενέργειας.

Επειδή δεν είναι αποδοτικό να δημιουργούνται περιγραφές για κάθε αισθητήρα στο δίκτυο και όλες αυτές, στη συνέχεια, να μεταφέρονται στο σημείο ελέγχου, στη αρχιτεκτονική αυτή οι περιγραφές χωρίζονται σε δύο μέρη. Το πρώτο αφορά τον κόμβο BOSS και χρησιμοποιεί το μηχανισμό του UPnP, ενώ το δεύτερο αφορά τους κόμβους των αισθητήρων και χρησιμοποιεί μια νέα μέθοδο για τη δημιουργία της περιγραφής. Συγκεκριμένα, αποφεύγεται η περιγραφή κόμβων όμοιου τύπου. Ο BOSS κόμβος δημιουργεί τις περιγραφές για κάθε τύπο αισθητήρα, σχηματίζοντας έτσι ομάδες από αισθητήρες του ίδιου τύπου. Το κομμάτι που αφορά την περιγραφή του κόμβου BOSS περιέχει τις επιπλέον υπηρεσίες που αφορούν συνολικά το δίκτυο αισθητήρων και παρέχει ο κόμβος αυτός. Με αυτόν τον τρόπο επιτυγχάνεται ο περιορισμός στο μέγεθος των αρχείων περιγραφής που πρέπει να σταλούν από τον BOSS στο σημείο ελέγχου.

Ένας κόμβος BOSS διαθέτει ένα πίνακα με τις υπηρεσίες κάθε τύπου κόμβου και ένα πίνακα με τις υπηρεσίες που παρέχει αυτός. Επίσης, διαθέτει ένα διαχειριστή υπηρεσιών, ο οποίος αναλαμβάνει τον εντοπισμό των υπηρεσιών που παρέχονται από τους αισθητήρες καθώς και την αποστολή των UPnP περιγραφών τους στα σημεία ελέγχου. Διαθέτει ακόμα δυο διαχειριστές για τον έλεγχο και την ειδοποίηση γεγονότων, αντίστοιχα.

4.2 ISNN – Intelligent Sensor Network Node

Επίσης, μια παρόμοια λύση γεφύρωσης των δύο δικτύων παρουσιάζεται και στο [MP07]. Οι συγγραφείς προτείνουν την κατασκευή ενός κόμβου, τον οποίο ονομάζουν ISNN (Intelligent Sensor Network Node), τόσο ως προς το hardware όσο και ως προς το software, βασισμένοι στην αρχή του co-design [ND04].

Για την υλοποίηση του ISNN χρησιμοποιούν τον μικροεπεξεργαστή της Atmel, Atmega 128, οποίος συνδέεται με μια ασύρματη βαθμίδα για την επικοινωνία με το δίκτυο των αισθητήρων, ενώ παράλληλα, μέσω σειριακού interface, επιτυγχά-

νεται η σύνδεση με κάποιον υπολογιστή. Επίσης, ο κόμβος αυτός διαθέτει και μια βαθμίδα flash μνήμης για την αποθήκευση των δεδομένων.

Για το software του ISNN, γίνεται αρχικά ένας διαχωρισμός των εργασιών σε αυτές που εκτελούνται από εφαρμογές και αφορούν τις μετρήσεις του αισθητήρα, σε άλλες που αφορούν την ασύρματη επικοινωνία της συσκευής και τέλος σε εκείνες που έχουν να κάνουν με το συνολικό δίκτυο. Έχοντας ορίσει αυτό το διαχωρισμό, η υλοποίηση της πλατφόρμας λογισμικού μπορεί να γίνει σε επίπεδα. Από την μία μεριά επομένως υπάρχει το λογισμικό μεταξύ του λειτουργικού συστήματος και του αισθητήρα, ενώ από την άλλη η κατανεμημένη εφαρμογή, η οποία αλληλεπιδρά με το δίκτυο.

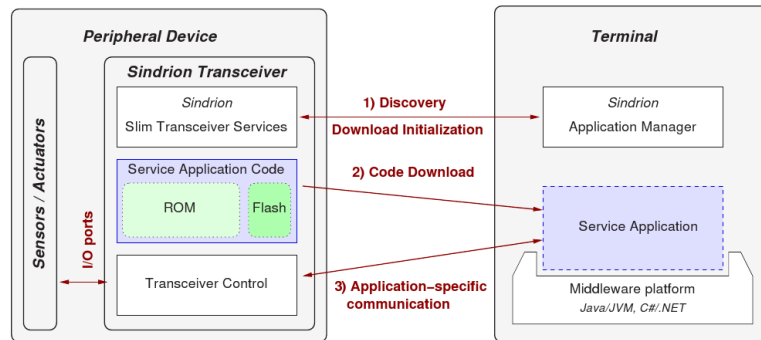
Όπως αναφέρθηκε, ένας ISNN κόμβος αποτελεί τη γέφυρα μεταξύ των δύο δικτύων. Από την πλευρά του UPnP δικτύου ο κόμβος αυτός αντιμετωπίζεται ως μια συσκευή UPnP και επικοινωνεί με τα σημεία ελέγχου χρησιμοποιώντας το UPnP. Από την άλλη, οι μη συμβατές UPnP συσκευές των αισθητήρων, επικοινωνούν με τον ISNN μέσω του ιδιόκτητου πρωτοκόλλου. Η λειτουργία του ISNN έχει ως εξής: όταν ένας κόμβος-αισθητήρας προστεθεί στο δίκτυο, στέλνει μια ανακοίνωση για την υπηρεσία του στον ISNN. Ο τελευταίος τοποθετεί την ανακοίνωση αυτή σε ένα πίνακα με τις υπηρεσίες των non-UPnP κόμβων και στη συνέχεια ενημερώνει το σημείο ελέγχου, με χρήση της UPnP ειδοποίησης, για την εισαγωγή του νέου αισθητήρα. Το σημείο ελέγχου καλώντας τη *GetSensorDescription()*, την οποία παρέχει ο ISNN, μπορεί να πληροφορηθεί με την περιγραφή του αισθητήρα. Η περιγραφή αυτή αποτελεί μια τυπική UPnP περιγραφή σε XML και δημιουργείται από τον ISNN.

Αντίστοιχα, όταν συνδέεται ένα σημείο ελέγχου στο UPnP δίκτυο, μπορεί να ζητήσει τα κείμενα των περιγραφών για τους αισθητήρες από το ISNN μέσω της ανακάλυψης και της περιγραφής του UPnP.

4.3 Sindrion

Μία άλλη προσέγγιση, η οποία δεν υλοποιεί γεφύρωση ανάμεσα στα διαφορετικά δίκτυα, παρουσιάζεται στο [GSS⁺04]. Οι συγγραφείς έχουν αναπτύξει μια κατανεμημένη αρχιτεκτονική συστήματος η οποία ονομάζεται Sindrion. Η βασική ιδέα του είναι να αφαιρέσει από τους κόμβους του WSN την εκτέλεση των πολύπλοκων υπολογισμών και να τους αναθέσει σε ειδικά σχεδιασμένους κόμβους, οι οποίοι καλούνται *terminals*. Το Sindrion υποστηρίζει την ασύρματη επικοινωνία και την αυτόνομη διαλειτουργικότητα των κόμβων αυτών.

Βάση του Sindrion αποτελεί η εγκαθίδρυση της ασύρματης επικοινωνίας μεταξύ των κόμβων του WSN και των *terminals*. Σκοπός της σύνδεσης αυτής είναι να μεταφέρει την υπολογιστικά απαιτητική επεξεργασία δεδομένων από τους αισθητήρες στα *terminals*. Για την επίτευξη αυτού, οι συσκευές εξοπλίζονται με «μικρούς



Σχήμα 4.2: Η βασική δομή για την εγκαθίδρυση σύνδεσης μεταξύ ενός terminal και ενός Sindrion Transceiver.

έξυπνους» πομποδέκτες RF οι οποίοι ονομάζονται Sindrion Transceivers. Οι συσκευές είναι είτε αισθητήρες είτε ενεργοποιητές, οι οποίες είναι περιορισμένων υπολογιστικών δυνατοτήτων και συνδέονται με τους Sindrion Transceivers μέσω γραμμών ελέγχου.

Ένα terminal διαθέτει και αυτό ένα RF πομποδέκτη, συμβατό με τον Sindrion Transceiver. Η επεξεργασία των δεδομένων και ο έλεγχος πραγματοποιείται από τα terminals, τα οποία, υποθετικά, διαθέτουν απεριόριστη επεξεργαστική ισχύ και μνήμη σε σύγκριση με τους Sindrion Transceivers.

Στο Σχήμα 4.2 παρουσιάζεται η διαδικασία για τη σύνδεση μεταξύ ενός terminal και ενός Sindrion Transceiver, ο οποίος ήταν άγνωστος ως τώρα. Η διαδικασία που ακολουθείται έχει ως εξής:

1. **Ανακάλυψη (Discovery):** Τα δύο άκρα εντοπίζουν το ένα το άλλο στη φάση αυτή. Στο σημείο αυτό χρησιμοποιείται το αντίστοιχο πρωτόκολλο της αρχιτεκτονικής UPnP.
2. **Μεταφόρτωση κώδικα:** Αν το terminal δεν διαθέτει ακόμα την εφαρμογή για τον έλεγχο τους συγκεκριμένου Sindrion Transceiver, ο τελευταίος αποστέλλει την εφαρμογή αυτή στο terminal. Για τον κώδικα της εφαρμογής προτιμάται η χρήση μιας middleware πλατφόρμας όπως είναι η Java Virtual Machine. Αυτό εξασφαλίζει την ανεξαρτησία από την πλατφόρμα και επιτρέπει την αδιάλειπτη ενσωμάτωση σε μια πληθώρα terminals. Επιπλέον, η χρήση υψηλού επιπέδου γλωσσών προγραμματισμού διευκολύνει την ανάπτυξη των εφαρμογών.
3. **Επικοινωνία βάσει της εφαρμογής:** Το επόμενο στάδιο επικοινωνίας πραγματοποιείται μεταξύ της μεταφορτωμένης από πριν εφαρμογής και του ελεγκτή του Sindrion Transceiver. Η επικοινωνία αυτή εξαρτάται από την κάθε εφαρμογή και δεν χρειάζεται να καθορίζεται από κάποιο στάνταρντ.

Η πολυπλοκότητα των σταδίων ελέγχου (control) και ειδοποίησης γεγονότων (eventing) του UPnP αποτρέπει την συνολική υλοποίηση της UPnP Αρχιτεκτονικής Συσκευής τους σε συσκευές χαμηλού κόστους, όπως είναι οι αισθητήρες. Επομένως, κάθε Sindrion Transceiver λειτουργεί ως βασική UPnP συσκευή. Μια βασική UPnP συσκευή υποστηρίζει την ανακάλυψη, περιγραφή και παρουσίαση, λειτουργίες που δεν τόσο είναι απαιτητικές όσο ο έλεγχος και η ειδοποίηση γεγονότος. Συνεπώς, ένας Sindrion Transceiver μπορεί να συνδεθεί άμεσα σε ένα δίκτυο UPnP.

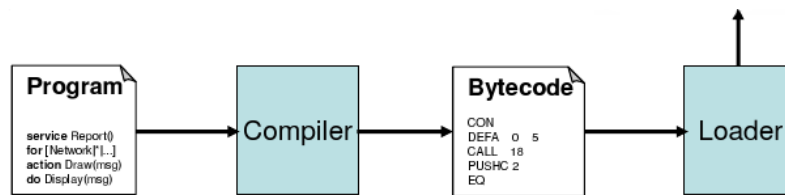
Ακολουθώντας την προσέγγιση του Sindrion η σελίδα παρουσίασης περιέχει ένα Java Applet μέσω του οποίου επιτυγχάνεται ο έλεγχος του Sindrion Transceiver και των περιφερειακών του, ενώ τα δεδομένα για τον έλεγχο και την ειδοποίηση γεγονότων μπορούν να μεταφερθούν στο terminal για την επεξεργασία τους. Για τον λόγο αυτό ένας Sindrion Διαχειριστής Εφαρμογής εκτελείται στα terminals ώστε να εντοπίζει τους Sindrion Transceivers, να μεταφορτώνει τον κώδικα της εφαρμογής τους και στη συνέχεια να την εκτελεί. Η εφαρμογή αυτή λειτουργεί ως UPnP proxy για τον Transceiver αυτόν και αναλαμβάνει την μετάφραση των πρωτοκόλλων που αφορούν την εφαρμογή στα αντίστοιχα UPnP XML μηνύματα που αφορούν τον έλεγχο και την ειδοποίηση γεγονότων.

4.4 WSN σε CE συσκευές

Στο [BLV09] παρουσιάζεται μια εφαρμογή που έχει ως στόχο να συνδέσει τις CE (Consumer Electronics) συσκευές με τα ασύρματα δίκτυα αισθητήρων αποσκοπώντας στην ανίχνευση δραστηριότητας στο οικιακό περιβάλλον για την εξυπηρέτηση σε θέματα υγείας. Οι συγγραφείς εστιάζουν στα μηνύματα που ανταλλάσσονται μεταξύ των αισθητήρων. Καθορίζουν μια κωδικοποίηση για τα μηνύματα αυτά βασισμένη στα γεγονότα, και η οποία είναι συγκρίσιμη με εκείνη που χρησιμοποιείται σε στάνταρντ πρωτόκολλα όπως το UPnP και το OSGi[osg], λαμβάνοντας υπ' όψιν τους περιορισμούς λόγω των «μικρών» αισθητήρων.

Το σύστημα που προτείνεται εδώ αποτελείται από έναν προγραμματιστικό συμβολισμό ο οποίος υποστηρίζεται από έναν compiler, έναν loader και το run-time σύστημα (Σχήμα 4.3). Ο προγραμματισμός των συσκευών γίνεται από το χρήστη γράφοντας ένα πρόγραμμα προσανατολισμένο στην υπηρεσία (service-oriented) για ένα πλήθος συσκευών. Για κάθε υπηρεσία ο χρήστης υποδεικνύει μέσω μια διεύθυνσης (content based address – CBA) υπό ποιες συνθήκες η συσκευή θα πρέπει να εγκαταστήσει και να εκτελέσει την εφαρμογή. Ο compiler εξάγει τις υπηρεσίες και τις μεταφράζει σε εντολές οι οποίες μεταγλωττίζονται από το σύστημα εκτέλεσης των συσκευών. Ο loader αποστέλλει τη CBA και τις μεταφρασμένες υπηρεσίες στο δίκτυο αισθητήρων. Η μορφή των μηνυμάτων αυτών είναι ανεξάρτητη του hardware των αισθητήρων.

Επιπρόσθετα, το πρόγραμμα καθορίζει τις συνδέσεις μεταξύ των υπηρεσιών με την



Σχήμα 4.3: Η αλυσίδα εργαλείων που χρησιμοποιείται για την μετατροπή του απλού προγραμματιστικού συμβολισμού σε *bytecode*.

μορφή συνδρομών (subscriptions). Μια συνδρομή συνδέει τυπικά την υπηρεσία που «παράγει» τα δεδομένα με την υπηρεσία που τα «καταναλώνει» και καθορίζει μια σειρά από παραμέτρους, όπως ο χρονισμός, η προτεραιότητα και η αξιοπιστία. Τα άκρα μιας τέτοιας σύνδεσης καθορίζονται από CBAs, ώστε το πρόγραμμα να μην εξαρτάται από πληροφορίες που έχουν να κάνουν με τις συσκευές. Οι συνδρομές πραγματοποιούνται από κάποια τρίτη οντότητα, η οποία μπορεί να συνδέσει μια υπηρεσία «παραγωγό» με μια «καταναλωτή». Αυτή η οντότητα μπορεί να είναι ο loader ο οποίος εγκαθιστά μια εφαρμογή ή απλά μια εφαρμογή εγκατεστημένη σε κάποια CE συσκευή, η οποία είναι ικανή να στέλνει μηνύματα στο δίκτυο των αισθητήρων.

Για το σύστημα αυτό αναπτύχθηκε ένα «ελαφρύς» compiler ο οποίος σχεδιάστηκε για το δίκτυο αισθητήρων. Εκτός από απλούς υπολογισμούς υποστηρίζει το χειρισμό βασικών εντολών για τη προσανατολισμένη σε υπηρεσίες γλώσσα προγραμματισμού. Ο compiler αυτός υλοποιήθηκε στους αισθητήρες σε C, επικοινωνώντας πάνω από IEEE 802.15.4, και σε H/Y και PDAs σε Python, επικοινωνώντας πάνω από UDP. Χρησιμοποιώντας μια εφαρμογή γεφύρωσης γίνεται εφικτή η επικοινωνία μεταξύ των αισθητήρων και των H/Y και PDAs με χρήση broadcasting. Η broadcast μετάδοση χρησιμοποιείται για την εγκατάσταση του προγράμματος από τον loader και περιστασιακά από τις συνδεδεμένες υπηρεσίες.

Οι κόμβοι μπορεί να είναι πραγματικοί ή προσομοιωμένοι. Οι τελευταίοι μπορούν να έχουν πρόσβαση σε πιο προχωρημένες δυνατότητες που προσφέρονται από το περιβάλλον που τους φιλοξενεί, όπως για παράδειγμα χώρο αποθήκευσης, προηγμένες επεξεργαστικές δυνατότητες, γραφική απεικόνιση κ.ά.. Η χρήση των επιπλέον αυτών χαρακτηριστικών πραγματοποιείται μέσω κλήσεων συστήματος (system calls). Οι κλήσεις συστήματος είναι προσβάσιμες από την ιδεατή μηχανή που εκτελείται στον κόμβο και η οποία δίνει σε όλους τους κόμβους την δυνατότητα πρόσβασης στις επιπρόσθετες λειτουργίες μέσω ενός μηχανισμού δειγματοληψίας των γεννητριών γεγονότος. Με την σύνδεση των κλήσεων συστήματος με UPnP υπηρεσίες προσφέρεται η δυνατότητα πρόσβασης μέσω ενός gateway.

4.5 Συμπεράσματα

Από τις εργασίες που παρουσιάστηκαν παραπάνω, οι οποίες αποτελούν ένα μέρος των προσπαθειών για την δυνατότητα παροχής UPnP δυνατοτήτων στα ασύρματα δίκτυα αισθητήρων, αναδεικνύεται η δυσκολία που ενέχεται σε ένα τέτοιο εγχείρημα. Οι δύο πρώτες προσεγγίσεις, BOSS και ISNN, βασίζονται στην γεφύρωση των δικτύων αισθητήρων με IP δίκτυα, τα οποία παρέχουν δυνατότητες UPnP. Μια προσέγγιση αυτού του τύπου, εκτός από την απαίτηση για την υλοποίηση ειδικών κόμβων, τόσο σε υλικό όσο και στη σχεδίαση του λογισμικού τους, αποτελούν κεντρικοποιημένες λύσεις. Η πρόσβαση στο δίκτυο των αισθητήρων βασίζεται αποκλειστικά από το κόμβο που γεφυρώνει τα δύο δίκτυα. Επομένως ο κόμβος αυτός πρέπει να σχεδιαστεί με μεγάλη προσοχή, ώστε να εξασφαλιστεί η αξιόπιστη λειτουργία συνολικά. Επίσης, σε μια τέτοια υλοποίηση το μέρος της επικοινωνίας μεταξύ των αισθητήρων και του κόμβου γεφύρωσης εξαρτάται αποκλειστικά από το πρωτόκολλο που χρησιμοποιείται από τους πρώτους και λαμβάνεται υπόψη κατά την υλοποίηση. Επομένως, για τη χρήση τέτοιων κόμβων γεφύρωσης σε άλλου τύπου δίκτυα, που χρησιμοποιούν κάποιο άλλο πρωτόκολλο επικοινωνίας απαιτεί την εκ νέου σχεδίαση και υλοποίηση τους.

Στο Sindrion ακολουθείται μια διαφορετική προσέγγιση για την παροχή UPnP λειτουργιών στο δίκτυο αισθητήρων από εκείνη της γεφύρωσης. Και εδώ απαιτείται ο σχεδιασμός και η υλοποίηση όλων των κόμβων του δικτύου, αφού ακόμα και οι συσκευές των αισθητήρων υποστηρίζουν IP πρωτόκολλα ώστε να υποστηρίζουν άμεσα τις βασικότερες λειτουργίες του UPnP. Βασικά, στην αρχιτεκτονική αυτή οι αισθητήρες και οι ενεργοποιητές δεν ενσωματώνονται με τις συσκευές ασύρματης επικοινωνίας, αλλά συνδέονται με τους Transceivers.

Τέλος, στη τέταρτη εργασία, εκτός από την διασύνδεση ενός δικτύου με φορητές συσκευές που χρησιμοποιούν δίκτυα IP, προσφέρεται και η δυνατότητα συγγραφής προγραμμάτων για τους αισθητήρες, χρησιμοποιώντας μια απλή γλώσσα προγραμματισμού ειδικά σχεδιασμένη για το σκοπό αυτό. Το γεγονός αυτό προσφέρει μεγάλη ευελιξία κατά δημιουργία των εφαρμογών για τους κόμβους. Και εδώ χρησιμοποιείται ένα είδος γεφύρωσης του δικτύου αισθητήρων με το υπόλοιπο δίκτυο, αλλά η παροχή UPnP υπηρεσιών πραγματοποιείται μέσω κόμβου gateway.

Κεφάλαιο 5

picoPnP

Το picoPnP αποτελεί μια προσπάθεια υλοποίησης μιας πλατφόρμας διαχείρισης και δημιουργίας εφαρμογών για αισθητήρες και ενεργοποιητές με στόχο τον αυτοματισμό κάποιων λειτουργιών σε εσωτερικούς χώρους, όπως το οικιακό περιβάλλον και το περιβάλλον εργασίας σε γραφεία. Σκοπός είναι η παροχή απλών εργαλείων στο χρήστη ώστε να διαχειρίζεται και να χρησιμοποιεί με συννεργατικό τρόπο τους αισθητήρες και τους ενεργοποιητές και παράλληλα να προσφέρει όλο το σύνολο των απαιτούμενων λειτουργιών, αφήνοντας στο προγραμματιστή μόνο τη υλοποίηση των τελικών εφαρμογών που εκτελούνται από τους αισθητήρες και τους ενεργοποιητές.

Η πλατφόρμα αυτή δημιουργήθηκε αποκλειστικά για την χρήση στα Sun SPOT (Small Programmable Object Technology) και είναι υλοποιημένη σε JAVA χρησιμοποιώντας το αντίστοιχο SDK. Το picoPnP προσπαθεί να διευκολύνει το χρήστη, αλλά και τον προγραμματιστή, στη δημιουργία απλών εφαρμογών, χρησιμοποιώντας τον ελάχιστο κώδικα JAVA, προσφέροντας όλες τις υπόλοιπες απαραίτητες λειτουργίες. Το “pico” προέρχεται από το γεγονός ότι προορίζεται για μικρές συσκευές, όπως είναι τα Sun SPOT, και το PnP από το γνωστό “Plug and Play” λόγω της εξάλειψης της ανάγκης για τον προγραμματισμό από το χρήστη όλων των βασικών λειτουργιών και αφήνοντάς του μόνο την υλοποίηση της εφαρμογής του κάθε κόμβου.

Η βασική ιδέα του picoPnP προέρχεται από το UPnP. Αν και δεν πρόκειται για κάποιο “universal” πρότυπο, όπως είναι το UPnP, δανειζεται πολλά από την φιλοσοφία λειτουργίας του τελευταίου. Από τεχνολογικής άποψης το picoPnP δεν μοιάζει καθόλου στο UPnP, αφού το δεύτερο προορίζεται αποκλειστικά σε δίκτυα IP και κάνει χρήση πολλών πρωτοκόλλων, τα οποία είναι δύσκολα –ως και καθόλου, σε μερικές περιπτώσεις– εφαρμόσιμα από τις μικρές συσκευές των Sun SPOT. Αντίθετα το picoPnP χρησιμοποιεί τα πρωτόκολλα επικοινωνίας που παρέχονται από τα Sun SPOT.

Πριν την περιγραφή και την ανάλυση της λειτουργίας του picoPnP, είναι απαραί-

τητο να γίνει μια σύντομη αναφορά στα Sun SPOT, στο τρόπο λειτουργίας τους και στο τι παρέχουν ως συσκευές. Τα Sun SPOT είναι μικρές, ασύρματες συσκευές, που λειτουργούν με μπαταρία, οι οποίες δημιουργήθηκαν στα Sun Labs [sun]. Μπορούν να χρησιμοποιηθούν σε ένα ευρύ φάσμα εφαρμογών, όπως την παρατήρηση του περιβάλλοντος, τη ρομποτική κ.ά.. Είναι εξοπλισμένα με μια ειδικά σχεδιασμένη JAVA Virtual Machine, που ονομάζεται Squawk, ικανή στο χειρίζεται πολλαπλές εφαρμογές ταυτόχρονα, ενώ δεν απαιτεί την ύπαρξη κάποιου λειτουργικού συστήματος για την λειτουργία της. Είναι συσκευές προγραμματιζόμενες και διαθέτουν παράλληλα κάποιους βασικούς αισθητήρες, όπως έντασης φωτός, θερμοκρασίας και επιταχυνσιόμετρο. Επίσης διαθέτουν οκτώ πολυχρωματικά LED, 2 πλήκτρα, 5 ακροδέκτες ψηφιακής εισόδου/εξόδου, 6 ακροδέκτες αναλογικής εισόδου και 4 ακροδέκτες ψηφιακής εξόδου. Για την ασύρματη επικοινωνία τους χρησιμοποιούν το πρότυπο IEEE 802.15.4 και διαθέτουν MAC διευθύνσεις. Επίσης, υλοποιούν και πρωτόκολλα δρομολόγησης για Mesh δίκτυα, όπως το AODV (Ad-hoc On demand Distance Vector) και το LQRP (Link Quality Routing Protocol), ενώ προσφέρουν και τη δυνατότητα για multi-hop broadcasting. Τέλος, να αναφερθεί πως διαθέτουν 4MB μνήμης flash, για την μόνιμη αποθήκευση δεδομένων. Με όλα τα παραπάνω και με την ευκολία που παρέχει ο προγραμματισμός τους σε JAVA, η χρήση τους σε ένα μεγάλο εύρος εφαρμογών είναι προφανής.

5.1 Ελεγκτές, αισθητήρες και ενεργοποιητές

Στο picoPnP υπάρχουν τρεις τύποι κόμβων: οι ελεγκτές, οι αισθητήρες και οι ενεργοποιητές. Οι δύο τελευταίοι θα αναφέρονται και ως συσκευές. Οι ελεγκτές είναι εκείνοι οι κόμβοι του δικτύου, οι οποίοι προσφέρουν στο χρήστη ένα σημείο από το οποίο αυτός μπορεί να ορίζει την λειτουργία των υπόλοιπων κόμβων. Οι συσκευές (αισθητήρες και ενεργοποιητές) είναι οι κόμβοι που παρέχουν κάποιες υπηρεσίες στο χρήστη, ο οποίος μπορεί να συνδυάζει τη λειτουργία τους χρησιμοποιώντας έναν ελεγκτή.

Όπως ήδη έχει αναφερθεί, οι αισθητήρες προορίζονται για την παρατήρηση του περιβάλλοντος, ενώ οι ενεργοποιητές για την εκτέλεση καθορισμένων ενεργειών σ' αυτό. Αμφότερες οι συσκευές αυτές περιγράφονται από κάποια βασικά χαρακτηριστικά. Συγκεκριμένα, τα χαρακτηριστικά που έχουν χρησιμοποιηθεί είναι:

1. **Το όνομα της συσκευής**, το οποίο δίνεται στην συσκευή κατά τον προγραμματισμό της εφαρμογής της, ώστε να παρέχεται η ευκολία αναγνώρισής της από το χρήστη. Το όνομα δεν είναι απαραίτητο να είναι διαφορετικό για κάθε συσκευή, αν και προτείνεται για την εύκολη διαφοροποίησή της από τις υπόλοιπες.
2. **Η διεύθυνση της συσκευής**, η οποία είναι η 64bit MAC διεύθυνση που διαθέτει κάθε SPOT και είναι μοναδική.

3. **Ο τύπος της συσκευής**, ο οποίος καθορίζει το αν πρόκειται για αισθητήρα ή ενεργοποιητή.
4. **Το πλήθος των hops**, το οποίο χρησιμοποιεί η συσκευή για την εκπομπή των στοιχείων της, όταν προστίθεται στο δίκτυο (αναφέρεται παρακάτω με λεπτομέρεια).
5. **Ο χρόνος ζωής** μιας συσκευής, που αποτελεί το χρονικό διάστημα μέσα στο οποίο η συσκευή μπορεί να θεωρείται ενεργή, χωρίς να έχει επικοινωνήσει με τον ελεγκτή. Με το πέρας της χρονικής αυτής διάρκειας η συσκευή στέλνει στους διαθέσιμους ελεγκτές ένα μήνυμα για την ανανέωση του χρόνου αυτού (ο μηχανισμός αυτός αναφέρεται παρακάτω με λεπτομέρεια).
6. **Η στάθμη της μπαταρίας** της συσκευής, σε ποσοστό επί τοις εκατό.

Εκτός από τα παραπάνω χαρακτηριστικά, κάθε συσκευή διαθέτει και κάποιες λειτουργίες τις οποίες παρέχει στο δίκτυο και καλούνται *υπηρεσίες*. Η κάθε υπηρεσία διαθέτει κάποιες προκαθορισμένες, κατά τον προγραμματισμό της συσκευής, τιμές ως *παραμέτρους*. Κάθε συσκευή μπορεί να παρέχει περισσότερες της μιας υπηρεσίες και κάθε υπηρεσία μπορεί να διαθέτει αρκετές παραμέτρους. Το πλήθος των τελευταίων έχει να κάνει με την φύση της υπηρεσίας. Για παράδειγμα ένας αισθητήρας με όνομα **Light Sensor 1**, που παρατηρεί την ένταση του φωτός, παρέχει την υπηρεσία **Light Meter**, η οποία διαθέτει τις παραμέτρους **LOW**, **MEDIUM**, **HIGH**. Ή ακόμα, ο ενεργοποιητής **Ambient Actuator 1**, που ελέγχει τον ηλεκτρικό φωτισμό και τον εξαερισμό κάποιου χώρου, παρέχει τις υπηρεσίες **Fan Operation**, με παραμέτρους **FAST**, **SLOW** και **OFF**, και **Light Operation**, με παραμέτρους **OFF**, **10%**, ..., **100%**.

Αναλυτικότερα, κάθε υπηρεσία χαρακτηρίζεται από τα εξής στοιχεία:

1. **Το ID**, που είναι το μοναδικό αναγνωριστικό για κάθε υπηρεσία στη συγκεκριμένη συσκευή.
2. **Το όνομα της υπηρεσίας**, το οποίο και αυτό πρέπει να είναι μοναδικό ώστε να διαφοροποιείται από τις άλλες στο χρήστη.
3. **Πληροφορίες** σχετικές με την υπηρεσία, ώστε να επεξηγείται στο χρήστη τι παρέχει.
4. **Πίνακας με τις παραμέτρους** που διαθέτει η υπηρεσία. Ο πίνακας αυτός περιέχει το αναγνωριστικό κάθε παραμέτρου και το όνομά της. Οι παράμετροι κάθε υπηρεσίας προκαθορίζονται από τον χρήστη κατά την υλοποίηση της εφαρμογής της συσκευής.

Τα πρώτα έξι στοιχεία περιγραφής μιας συσκευής αποτελούν την *βασική περιγραφή* της (basic description), ενώ προσθέτοντας και τις υπηρεσίες της προκύπτει η

εκτενής περιγραφή της συσκευής (extended description). Ο χρήστης μπορεί να υλοποιεί την δική του συσκευή, ορίζοντας τα εν λόγω στοιχεία, συσκευής και υπηρεσιών, και παρέχοντας μόνο τον κώδικα που εκτελείται από τις υπηρεσίες. Τα υπόλοιπα τα αναλαμβάνει το *picoPnP*, δηλαδή τον τρόπο επικοινωνίας με τις υπόλοιπες συσκευές και ελεγκτές και τον τρόπο συσχέτισης της συσκευής αυτής με άλλες. Η υλοποίηση των συσκευών και των υπηρεσιών αναφέρεται σε επόμενη ενότητα.

Σκοπός του *picoPnP* είναι να προσφέρει στο χρήστη, μέσω μια απλής γραφικής διεπαφής που εκτελείται στον ελεγκτή, τη δυνατότητα να συνδυάζει τις υπηρεσίες που προσφέρουν οι συσκευές¹. Στο προηγούμενο παράδειγμα, ο χρήστης είναι σε θέση να συνδυάσει την υπηρεσία *Light Meter* του *Light Sensor 1* με την υπηρεσία *Light Operation* του *Ambient Actuator 1* ως εξής: όταν το επίπεδο του φωτός, που διαβάσει η *Light Meter* υπηρεσία, βρίσκεται στο *LOW*, τότε θέτει το φωτισμό μέσω της *Light Operation* στο *100%* και όταν η *Light Meter* διαβάσει επίπεδο φωτός *HIGH*, τότε η *Light Operation* θέτει το φωτισμό στο *OFF*.

Ένας ελεγκτής, όπως έχει ήδη αναφερθεί, είναι το μέσο από το οποίο ο χρήστης μπορεί να συσχετίζει τις υπηρεσίες των συσκευών, συνδυάζοντας τις παραμέτρους τους. Πρόκειται για μια εφαρμογή που εκτελείται σε κάποιο *H/Y*, στον οποίο είναι συνδεδεμένο ένα *Basestation SPOT*. Το τελευταίο είναι ένα *SPOT*, το οποίο δεν διαθέτει μπαταρία ούτε περιφερειακά, όπως αυτά που χρησιμοποιούνται από τους αισθητήρες και τους ενεργοποιητές. Λειτουργεί μόνο όταν συνδέεται στη *USB* θύρα κάποιου *H/Y* και προσφέρει την δυνατότητα ασύρματης επικοινωνίας του *H/Y* με τα υπόλοιπα *SPOT*. Η εφαρμογή προσφέρει μια γραφική διεπαφή (*GUI*) μέσω της οποίας ο χρήστης μπορεί να βλέπει τις διαθέσιμες συσκευές και να ορίζει συνεργασίες μεταξύ τους. Να σημειωθεί πως το *picoPnP* επιτρέπει τη δυνατότητα ταυτόχρονης σύνδεσης περισσότερων του ενός ελεγκτών στο δίκτυο. Στο Παράρτημα Α.1 παρουσιάζεται το γραφικό περιβάλλον διαχείρισης που εκτελείται στους ελεγκτές.

5.2 Λειτουργία του *picoPnP*

Το *picoPnP* παρέχει κάποιες βασικές λειτουργίες, όπου μερικές μοιάζουν κατά πολύ με τις αντίστοιχες του *UPnP*. Προσφέρει ένα μηχανισμό *ανακάλυψης των συσκευών* από τους ελεγκτές, ένα μηχανισμό για την *ανάκτηση της περιγραφής* και των υπηρεσιών των συσκευών από τους ελεγκτές, καθώς επίσης και το μηχανισμό για την *συσχέτιση των υπηρεσιών* των συσκευών μεταξύ τους. Τέλος, παρέχει το τρόπο με τον οποίο επικοινωνούν μεταξύ τους συσχετισμένες συσκευές για την *επίτευξη* κάποιας *συνεργασίας*. Όπως είναι ήδη φανερό, πολλές από αυτές σχετίζονται με εκείνες που υπάρχουν και στο *UPnP*. Βέβαια ο τρόπος λειτουργίας τους και τα πρωτόκολλα που χρησιμοποιούν δεν είναι τα ίδια.

¹Συγκεκριμένα, η συσχέτιση των υπηρεσιών γίνεται μεταξύ συσκευών διαφορετικού τύπου. Δηλαδή, μπορεί να συσχετιστεί μια υπηρεσία ενός αισθητήρα με μια ενός ενεργοποιητή.

Με εξαίρεση την πρώτη φάση της διευθυνσιοδότησης και τις λειτουργίες ελέγχου και παρουσίασης, που υπάρχουν στο UPnP, οι υπόλοιπες φάσεις ανακάλυψης, περιγραφής και ειδοποίησης γεγονότων, υπάρχουν και στο *picoPnP*. Εδώ δεν χρειάζεται η ύπαρξη κάποιου μηχανισμού διευθυνσιοδότησης για τους κόμβους του δικτύου, αφού ο καθένας διαθέτει μια μοναδική διεύθυνση MAC και η αναφορά σ' αυτούς μπορεί να γίνει μέσω αυτής. Επίσης, δεν έχει νόημα ο μηχανισμός της παρουσίας των συσκευών, όπως αυτή εννοείται στο UPnP, αφού κάτι τέτοιο προϋποθέτει την ύπαρξη κάποιου Web server που θα εκτελείται στις συσκευές των αισθητήρων και των ενεργοποιητών, συν το γεγονός ότι ένα είδος παρουσίασης των συσκευών γίνεται ήδη στους ελεγκτές. Τα υπόλοιπα στάδια, που είναι κατά κάποιο τρόπο κοινά με αυτά του UPnP, δεν παρέχουν ακριβώς τις ίδιες δυνατότητες, αλλά είναι απλουστευμένα ώστε να μπορούν να εκτελούνται με αξιοπιστία στους «μικρούς» κόμβους. Στις ακόλουθες ενότητες παρουσιάζονται αναλυτικά οι μηχανισμοί που προσφέρονται από το *picoPnP* για την συνολική λειτουργία του δικτύου και την επίτευξη του σκοπού του, ο οποίος είναι η ανάθεση εργασιών στις συσκευές.

5.2.1 Ανακάλυψη συσκευών

Η ανακάλυψη των συσκευών είναι το πρώτο βήμα στο *picoPnP*. Μέσω της ανακάλυψης των συσκευών, οι ελεγκτές εντοπίζουν τους διαθέσιμους αισθητήρες και ενεργοποιητές στο δίκτυο. Επίσης, επιτρέπει στους ελεγκτές να αποκτήσουν στη συνέχεια μια λεπτομερέστερη γνώση για τις συσκευές και τις υπηρεσίες που αυτές παρέχουν.

Όταν μια συσκευή ενεργοποιείται πρέπει να εκπέμψει ένα μήνυμα προβάλλοντας την περιγραφή της στους διαθέσιμους ελεγκτές. Οι τελευταίοι «ακούν» στο προκαθορισμένο port για τα μηνύματα αυτά και ενημερώνονται για την ύπαρξη κάποιας νέας συσκευής στο δίκτυο. Από την άλλη, όταν ενεργοποιείται ένα ελεγκτής, εκπέμπει στο δίκτυο ένα μήνυμα αναζήτησης για τον εντοπισμό των διαθέσιμων συσκευών σε αυτό. Οι συσκευές που λαμβάνουν το μήνυμα αυτό, από το προκαθορισμένο port, απαντούν στον ελεγκτή που το έστειλε με ένα μήνυμα με το οποίο, προβάλλουν τα χαρακτηριστικά τους.

```
SEARCH message {
    byte      messageType = SEARCH
    boolean   transmissionType = BROADCAST
}
```

Σχήμα 5.1: Το περιεχόμενο ενός εκπεμπόμενου μηνύματος **SEARCH**. Το πεδίο για τον τύπο μετάδοσης τίθεται στην τιμή **BROADCAST**.

Επίσης, το πρωτόκολλο αυτό παρέχει και ένα μηχανισμό με τον οποίο ένας ελεγκτής ενημερώνεται συνεχώς για την κατάσταση των συσκευών και την διαθεσιμότητά τους. Επιπλέον εξασφαλίζει την ομαλή αποχώρηση από το δίκτυο τόσο στις

συσκευές όσο και στους ελεγκτές.

Η μελέτη του πρωτοκόλλου ανακάλυψης παρουσιάζεται σε τρεις φάσεις. Πρώτα περιγράφονται οι λειτουργίες που συμβαίνουν κατά την αναζήτηση των συσκευών από τους ελεγκτές. Έπειτα αναλύεται η διαδικασία προβολής μια συσκευής που εισέρχεται στο δίκτυο στους ενεργούς ελεγκτές και τέλος αναφέρονται οι μηχανισμοί που εξασφαλίζουν την ομαλή λειτουργία κατά την αποχώρηση κάποιου κόμβου και την συνεχή ενημέρωση από τις συσκευές προς τους ελεγκτές για την κατάστασή τους και την διαθεσιμότητά τους.

Αναζήτηση συσκευών από τους ελεγκτές

```

ADVERTISE message {

    byte      messageType = ADVERTISE

    boolean   transmissionType = UNICAST

    int       dataLength = το μέγεθος των
                           δεδομένων που αποστέλλονται

    byte[]    data = τα δεδομένα που περιέ-
                     χουν τα βασικά στοιχεία της
                     συσκευής

}

```

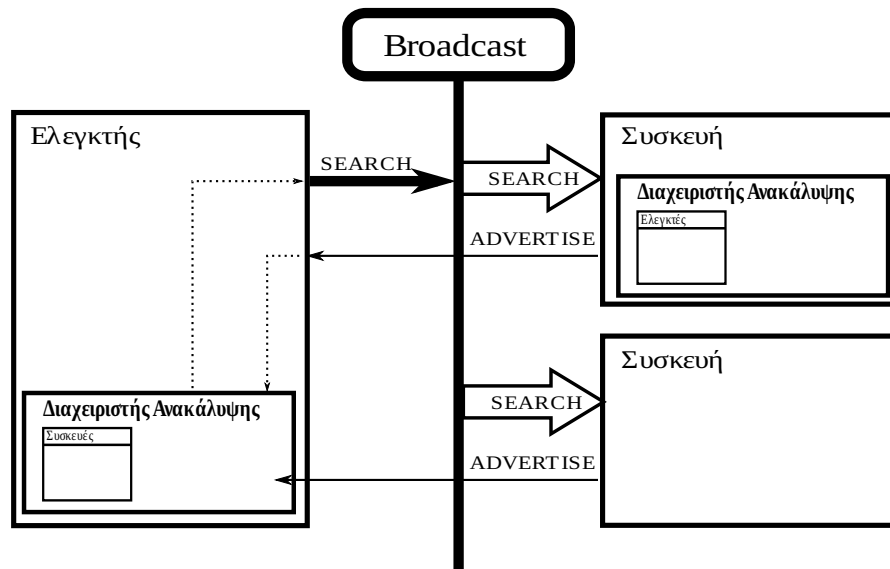
Σχήμα 5.2: Το περιεχόμενο μηνύματος *ADVERTISE*, ως απάντησης σε ένα *SEARCH* μήνυμα. Το πεδίο για τον τύπο μετάδοσης τίθεται στην τιμή *UNICAST*.

Όπως ήδη αναφέρθηκε, όταν ενεργοποιείται ένας ελεγκτής εκπέμπει ένα μήνυμα αναζήτησης στο δίκτυο. Αρχικά, το μήνυμα αυτό εκπέμπεται σε εύρος ενός hop, ανακαλύπτοντας έτσι τις άμεσα γειτονικές συσκευές του. Βέβαια, στο χρήστη παρέχεται η δυνατότητα να αναζητήσει συσκευές και σε μεγαλύτερο εύρος, μεταβάλλοντας το πλήθος των hops, από την εφαρμογή διεπαφής. Ένα μήνυμα αναζήτησης που εκπέμπεται από έναν ελεγκτή ονομάζεται *SEARCH* και περιέχει τα πεδία που παρουσιάζονται στο Σχήμα 5.1. Το πρώτο πεδίο χαρακτηρίζει τον τύπο του μηνύματος, ενώ το δεύτερο χαρακτηρίζει τον τρόπο μετάδοσης. Το πρωτόκολλο ανα-

κάλυψης υποστηρίζει την αποστολή μηνύματος αναζήτησης και προς συγκεκριμένη διεύθυνση, δηλαδή με unicast μετάδοση², επομένως το δεύτερο πεδίο χρησιμοποιείται ώστε να διαχωρίζονται τα μηνύματα που εκπέμπονται από εκείνα που στέλνονται σε συγκεκριμένες διευθύνσεις.

Μια συσκευή που λαμβάνει ένα *SEARCH* μήνυμα, αντιλαμβάνεται την ύπαρξη κάποιου ελεγκτή. Τοποθετεί την διεύθυνση του ελεγκτή αυτού σε ένα πίνακα μαζί με το timestamp της επικοινωνίας αυτής και στη συνέχεια απαντάει με ένα μήνυμα *ADVERTISE* ενημερώνοντας τον αποστολέα με τα στοιχεία της. Τα περιεχόμενα ενός *ADVERTISE* μηνύματος παρουσιάζονται στο Σχήμα 5.2. Το πρώτο πεδίο χαρακτηρίζει τον τύπο του μηνύματος, ενώ το δεύτερο πεδίο, που χαρακτηρίζει τον τρόπο μετάδοσης του μηνύματος, τίθεται σε unicast, αφού η συσκευή το αποστέλλει

²Η unicast μετάδοση γίνεται με την χρήση των πρωτοκόλλων δρομολόγησης που παρέχονται από τα SPOT. Στην εφαρμογή αυτή χρησιμοποιείται το πρωτόκολλο LQRP.



Σχήμα 5.4: Αναζήτηση συσκευών από τον ελεγκτή (Search). Ο ελεγκτής εκπέμπει το SEARCH μήνυμα. Οι συσκευές που λαμβάνουν το μήνυμα αυτό τοποθετούν τη διεύθυνση του ελεγκτή στον πίνακα και απαντούν με ένα ADVERTISE μήνυμα το οποίο περιέχει τη βασική περιγραφή της συσκευής. Ο ελεγκτής λαμβάνει το ADVERTISE και προσθέτει τη περιγραφή της συσκευής στον πίνακα συσκευών.

στη διεύθυνση του ελεγκτή από τον οποίο έλαβε το SEARCH μήνυμα. Το μήνυμα αυτό περιέχει επίσης και τη βασική περιγραφή της συσκευής, σε ένα byte array. Βέβαια πριν το byte array, τοποθετείται ένα πεδίο ακόμα στο μήνυμα το οποίο αναφέρεται στο μήκος του array με την περιγραφή, ώστε ο παραλήπτης να γνωρίζει πόσα bytes αποτελούν χρήσιμη πληροφορία.

Ο ελεγκτής λαμβάνει στη συνέχεια τα ADVERTISE μηνύματα από τις συσκευές που απαντούν, διαβάζει τα περιεχόμενά του καθενός, στα οποία περιέχεται η βασική περιγραφή της κάθε συσκευής και την αποθηκεύει σε ένα πίνακα. Παράλληλα, σε έναν άλλο πίνακα αντιστοιχεί τη διεύθυνση της κάθε συσκευής με την χρονική στιγμή που έλαβε το μήνυμα (timestamp). Η τελευταία αυτή λειτουργία χρησιμεύει για τη μελλοντική εξακρίβωση λειτουργίας της συσκευής και αναλύεται παρακάτω.

Από την παραπάνω διαδικασία εξασφαλίζονται τα εξής: οι ελεγκτές γνωρίζουν τη βασική περιγραφή για κάθε μια διαθέσιμη συσκευή στο δίκτυο και οι τελευταίες γνωρίζουν τις διευθύνσεις των ενεργών ελεγκτών στο δίκτυο. Στο Σχήμα 5.4 φαίνε-

```

ADVERTISE message {

    byte      messageType = ADVERTISE

    boolean   transmissionType = BROADCAST

    int       dataLength = το μέγεθος των
                        δεδομένων που αποστέλλονται

    byte[]    data = τα δεδομένα που περιέ-
                    χουν τα βασικά στοιχεία της
                    συσκευής

}

```

Σχήμα 5.3: Το περιεχόμενο μηνύματος ADVERTISE. Το πεδίο για τον τύπο μετάδοσης τίθεται στην τιμή BROADCAST.

ται σχηματικά η διαδικασία αυτή.

Προβολή συσκευής στους ελεγκτές

Αντίστοιχα με την αναζήτηση των συσκευών από τους ελεγκτές, όταν μια συσκευή ενεργοποιείται στο δίκτυο τότε πρέπει να ανακαλύψει τους διαθέσιμους ελεγκτές και να τους ενημερώσει με την βασική της περιγραφή. Κατά την ενεργοποίηση μια συσκευής, λοιπόν, εκπέμπεται στο δίκτυο ένα μήνυμα **ADVERTISE**. Το πλήθος των hops, που γίνεται η εκπομπή αυτή καθορίζεται από την κάθε συσκευή κατά τον προγραμματισμό της από τον χρήστη. Το μήνυμα αυτό είναι το ίδιο όπως και στην περίπτωση απάντησης σε ένα **SEARCH** μήνυμα, μόνο που η τιμή του πεδίου για τον τύπο μετάδοσης έχει την τιμή που αντιστοιχεί σε broadcast (Σχήμα 5.3).

```
ADVERTISE_ACK message {

    byte      messageType = ADVERTISE_ACK

    boolean   transmissionType = UNICAST

}
```

Σχήμα 5.5: Το περιεχόμενο ενός μηνύματος **ADVERTISE_ACK**.

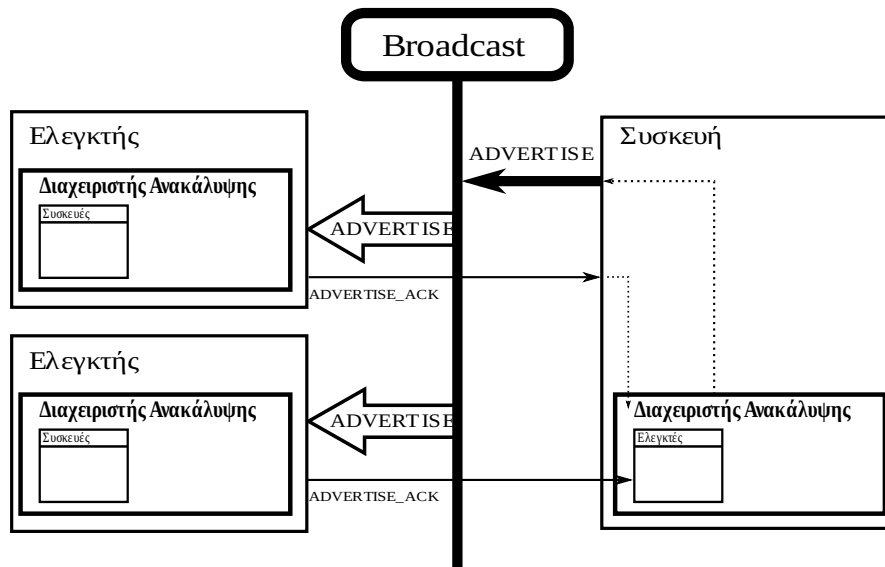
Ο ελεγκτής, που λαμβάνει το παραπάνω μήνυμα, αναγνωρίζει πως πρόκειται για κάποια νέα συσκευή που μόλις εισήλθε στο δίκτυο. Αυτό διαπιστώνεται εύκολα από το πεδίο τύπου μετάδοσης, που έχει την τιμή που αντιστοιχεί σε broadcast μετάδοση. Διαβάζει τα περιεχόμενα του μηνύματος και τοποθετεί την περιγραφή της συσκευής στο πίνακα που κρατά τις πληροφορίες για όλες τις διαθέσιμες συσκευές στο δί-

κτυο. Επίσης τοποθετεί και την χρονική στιγμή που έλαβε το μήνυμα στον ανάλογο πίνακα. Στην συνέχεια ο ελεγκτής πρέπει να ενημερώσει με τη σειρά του την συσκευή για την ύπαρξή του. Στέλνει, επομένως, ένα μήνυμα επιβεβαίωσης σε αυτήν το οποίο δεν περιέχει κάποια πληροφορία, απλώς χρησιμοποιείται ώστε η συσκευή να γνωρίσει την διεύθυνση του ελεγκτή (Σχήμα 5.5).

Η συσκευή λαμβάνει τα μηνύματα **ADVERTISE_ACK** από τους ενεργούς ελεγκτές και τοποθετεί την διεύθυνσή τους σε ένα πίνακα μαζί με τη χρονική στιγμή που έλαβαν το μήνυμα. Η διαδικασία αυτή παρουσιάζεται στο Σχήμα 5.6. Έτσι, οι ελεγκτές γνωρίζουν την βασική περιγραφή της νέας συσκευής και αυτή, με τη σειρά της, την ύπαρξη των διαθέσιμων ελεγκτών.

Διατήρηση σύνδεσης και ομαλή έξοδος από το δίκτυο

Το πρωτόκολλο ανακάλυψης, παράλληλα με την ανακάλυψη των συσκευών, διαθέτει και μηχανισμούς που εξασφαλίζουν την ομαλή λειτουργία του δικτύου. Οι μηχανισμοί αυτοί παρακολουθούν την διατήρηση της σύνδεσης με τις συσκευές και επιπλέον ενημερώνουν τους ελεγκτές όταν μια συσκευή αποχωρεί από το δίκτυο.



Σχήμα 5.6: Προβολή συσκευής στους ελεγκτές (*Advertisement*). Η συσκευή εκπέμπει το *ADVERTISE* μήνυμα το οποίο περιέχει τη βασική της περιγραφή. Οι ελεγκτές που λαμβάνουν το μήνυμα αυτό τοποθετούν την περιγραφή της συσκευής στο πίνακα συσκευών και απαντούν με ένα μήνυμα επιβεβαίωσης, ώστε να εντοπιστούν και από την συσκευή.

Όπως έχει αναφερθεί, σε κάθε συσκευή, κατά τον προγραμματισμό της, δηλώνεται από το χρήστη ο μέγιστος χρόνος ζωής της. Ο χρόνος αυτός χρησιμοποιείται για να παρέχεται η δυνατότητα στον ελεγκτή να ελέγχει αν οι συσκευές είναι ενεργές κατά τη διάρκεια λειτουργίας του. Συγκεκριμένα, κάθε συσκευή στέλνει περιοδικά ένα μήνυμα ενημέρωσης σε κάθε ελεγκτή που γνωρίζει. Η περίοδος αποστολής είναι ο χρόνος ζωής της συσκευής και το μήνυμα καλείται **HEARTBEAT**.

Ένα **HEARTBEAT** μήνυμα από την συσκευή προς τους ελεγκτές περιέχει τη στάθμη της μπαταρίας της συσκευής. Αντίθετα, σε ένα αντίστοιχο μήνυμα από τον ελεγκτή προς τη συσκευή στο πεδίο αυτό η τιμή τίθεται σε -1, αφού δεν έχει νόημα το πεδίο αυτό για τον ελεγκτή, καθώς δε διαθέτει μπαταρία. Το περιεχόμενο ενός μηνύματος **HEARTBEAT** παρουσιάζεται στο Σχήμα 5.7.

```

HEARTBEAT message {

    byte      messageType = HEARTBEAT

    boolean   transmissionType = UNICAST

    int       batteryStatus = η στάθμη της
                           μπαταρίας της συσκευής [0-
                           100]

}

```

Σχήμα 5.7: Το περιεχόμενο ενός μηνύματος **HEARTBEAT**.

Κάθε ελεγκτής που λαμβάνει ένα τέτοιο μήνυμα ανανεώνει το timestamp της συσκευής στον πίνακα των timestamps, το επίπεδο της μπαταρίας για τη συσκευή αυτή και απαντά σ' αυτή με ένα όμοιο μήνυμα. Ομοίως, η συσκευή που λαμβάνει την απάντηση ανανεώνει με τη σειρά της το timestamp του ελεγκτή. Στέλνοντας την

στάθμη της μπαταρίας, ο χρήστης έχει τη δυνατότητα να γνωρίζει την κατάσταση της συσκευής και να τη λάβει υπ' όψιν του κατά την ανάθεση κάποιας εργασίας σε αυτή.

Οι συσκευές ελέγχουν περιοδικά τα timestamps των ελεγκτών, εντοπίζουν αυτούς των οποίων δεν έχει ανανεωθεί το timestamp και τους διαγράφουν από τον πίνακα διευθύνσεων. Το κριτήριο ελέγχου για τα timestamps είναι:

$$ts_{now} - ts_{con} > maxage_{dev} + d, \quad (5.1)$$

όπου ts_{now} είναι η τρέχουσα χρονική στιγμή, ts_{con} το timestamp του ελεγκτή, $maxage_{dev}$ ο χρόνος ζωής της συσκευής και d ένα χρονικό περιθώριο του οποίου η τιμή κυμαίνεται περίπου στα 2s και ο σκοπός ύπαρξής του είναι στο να καλύψει το χρόνο για την αποστολή και λήψη των μηνυμάτων HEARTBEAT. Εφόσον η αποστολή των μηνυμάτων αυτών ξεκινά από τις συσκευές κάθε φορά με περίοδο ίση το χρόνο ζωής τους και οι ελεγκτές απλώς απαντούν στα μηνύματα αυτά, στην ουσία οι συσκευές λαμβάνουν τα μηνύματα HEARTBEAT χονδρικά με την ίδια περίοδο που τα στέλνουν. Μια μικρή διαφορά μπορεί να οφείλεται στο χρόνο μετάδοσης.

Οι ελεγκτές, των οποίων το timestamp ικανοποιεί τη σχέση 5.1, θεωρούνται ανενεργοί και διαγράφονται από τον πίνακα των διευθύνσεων. Η διαδικασία ελέγχου πραγματοποιείται κάθε φορά πριν η συσκευή στείλει τα HEARTBEAT μηνύματα στους ελεγκτές.

Από την άλλη, οι ελεγκτές ελέγχουν και αυτοί τα timestamps των συσκευών, ώστε να εντοπίσουν αυτές που είναι ανενεργές (δεν έχουν ανανεώσει το timestamp τους). Ο έλεγχος εδώ γίνεται κάθε 1s, αφού κάθε συσκευή μπορεί να στέλνει με διαφορετική περίοδο τα HEARTBEAT μηνύματα. Ο χρόνος αυτός επιλέγεται σκοπίμως μικρός, ώστε να γίνονται άμεσα αντιληπτές οι ανενεργές συσκευές. Ούτως ή άλλως, η συχνή εκτέλεση του ελέγχου δεν επιβαρύνει τους ελεγκτές, αφού δεν υπάρχουν περιορισμοί στην κατανάλωση ενέργειας, εφόσον δεν διαθέτουν μπαταρία και τροφοδοτούνται άμεσα από τον H/Y. Η σχέση που πρέπει να ισχύει ώστε να θεωρηθεί μια συσκευή ανενεργή είναι:

$$ts_{now} - ts_{dev} > maxage_{dev} + d, \quad (5.2)$$

όπου ts_{now} είναι η τρέχουσα χρονική στιγμή, ts_{dev} το timestamp της συσκευής, $maxage_{dev}$ ο χρόνος ζωής της συσκευής και d το χρονικό περιθώριο που περιγράφηκε παραπάνω. Όταν ισχύει η σχέση αυτή για το timestamp κάποιας συσκευής, τότε αυτή αφαιρείται από τον πίνακα των συσκευών και διαγράφεται και από τον πίνακα των timestamps.

Επίσης, αν ένας ελεγκτής λάβει ένα μήνυμα HEARTBEAT από κάποια συσκευή, η οποία όμως δεν υπάρχει στον πίνακα συσκευών, τότε σημαίνει πως υπάρχει κάποια συσκευή η οποία τον γνωρίζει αλλά αυτός δεν διαθέτει την περιγραφή της. Σε

μια τέτοια περίπτωση ο ελεγκτής στέλνει ένα μήνυμα αναζήτησης **SEARCH**³ στη συσκευή αυτή ζητώντας την περιγραφή της. Η διαδικασία που ακολουθεί είναι όμοια με αυτή που περιγράφεται παραπάνω.

Τέλος, κάθε συσκευή διαθέτει ένα μηχανισμό με τον οποίο παρατηρεί την κατάσταση της μπαταρίας της και το πότε επανεκκινεί ή τερματίζει τη λειτουργία της ενημερώνοντας τους ελεγκτές ότι αποχωρεί από το δίκτυο. Συγκεκριμένα, όταν η συσκευή διαπιστώσει χαμηλή στάθμη στη μπαταρία ή ο χρήστης την τερματίζει ή επανεκκινεί, τότε εκπέμπει ένα μήνυμα «αποχαιρετισμού» στο δίκτυο με το οποίο ενημερώνονται οι ελεγκτές ότι η συσκευή αποσυνδέεται από το δίκτυο. Το μήνυμα αυτό ονομάζεται **BYEBYE** και παρουσιάζεται στο Σχήμα 5.8 και οι ελεγκτές που το λαμβάνουν αφαιρούν τη συσκευή αυτή από τον πίνακα συσκευών. Παρόμοια, όταν ο χρήστης τερματίζει την εφαρμογή του ελεγκτή, αυτός εκπέμπει στο δίκτυο ένα **BYEBYE** μήνυμα ώστε να ενημερώσει τις συσκευές⁴ με την αποχώρησή του.

```

BYEBYE message {

    byte      messageType = BYEBYE

    boolean   transmissionType = BROADCAST

}

```

Σχήμα 5.8: Το περιεχόμενο ενός μηνύματος **BYEBYE**.

5.2.2 Περιγραφή συσκευών

Το αμέσως επόμενο βήμα, μετά από την ανακάλυψη των συσκευών από τους ελεγκτές, είναι οι τελευταίοι να αποκτήσουν τις πληροφορίες για τις υπηρεσίες που αυτές παρέχουν, ώστε να είναι διαθέσιμες στο χρήστη. Αφού ο ελεγκτής ανακαλύψει μια συσκευή, γνωρίζει μόνο τη βασική περιγραφή της. Για να πληροφορηθεί για τις υπηρεσίες που παρέχει η συσκευή αυτή πρέπει να επικοινωνήσει μαζί της ζητώντας τις πληροφορίες για αυτές. Τη διαδικασία αυτή την αναλαμβάνει το πρωτόκολλο περιγραφής. Το πρωτόκολλο αυτό χρησιμοποιεί ένα προκαθορισμένο port, διαφορετικό από εκείνο του πρωτοκόλλου ανακάλυψης.

Ένας ελεγκτής, για να ζητήσει τις πληροφορίες για τις υπηρεσίες μιας συσκευής, στέλνει ένα μήνυμα σ' αυτή μέσω τους πρωτοκόλλου περιγραφής. Το μήνυμα αυτό καλείται **DESC_REQUEST**. Μια συσκευή που λαμβάνει αυτό το μήνυμα απαντάει με ένα μήνυμα **DESC_REPLY**, το οποίο περιέχει την πληροφορία για κάθε μία από τις υπηρεσίες της. Τα μηνύματα αυτά παρουσιάζονται στα Σχήματα 5.9.1 και 5.9.2.

³Το μήνυμα αυτό είναι όμοιο με αυτό του Σχήματος 5.1, μόνο που στο πεδίο που δηλώνει τον τύπο μετάδοσης τίθεται η τιμή που αντιστοιχεί σε unicast μετάδοση.

⁴Αν και το μήνυμα αυτό θα μπορούσε να αποστέλλεται σε κάθε γνωστή διεύθυνση, κάτι τέτοιο καθυστερεί πολύ τον τερματισμό της συσκευής ή του ελεγκτή, επομένως χρησιμοποιείται broadcast μετάδοση. Η αναξιοπιστία της broadcast μετάδοσης, εδώ δεν παίζει πολύ σημαντικό ρόλο, αφού κάποια στιγμή η συσκευή ή ο ελεγκτής που απενεργοποιείται θα αφαιρεθεί μη έχοντας ανανεώσει την παρουσία του.

```
DESC_REQUEST message {

    byte      messageType = DESC_REQUEST

}
```

5.9.1: Το περιεχόμενο ενός μηνύματος *DESC_REQUEST*.

```
DESC_REPLY message {

    byte      messageType = DESC_REPLY

    int       totalOfServices = το πλήθος
                        των υπηρεσιών που αποστέλλο-
                        νται
                        Για κάθε υπηρεσία{

        int     serviceDataLength = το μέ-
                        γεθος των δεδομένων για την
                        τρέχουσα υπηρεσία

        byte[]  serviceData = τα δεδομένα
                        της τρέχουσας υπηρεσίας
                        }

}
```

5.9.2: Το περιεχόμενο ενός μηνύματος *DESC_REPLY*.

Σχήμα 5.9: Τα μηνύματα που ανταλλάσσουν ελεγκτής και συσκευή για την ανάκτηση των υπηρεσιών.

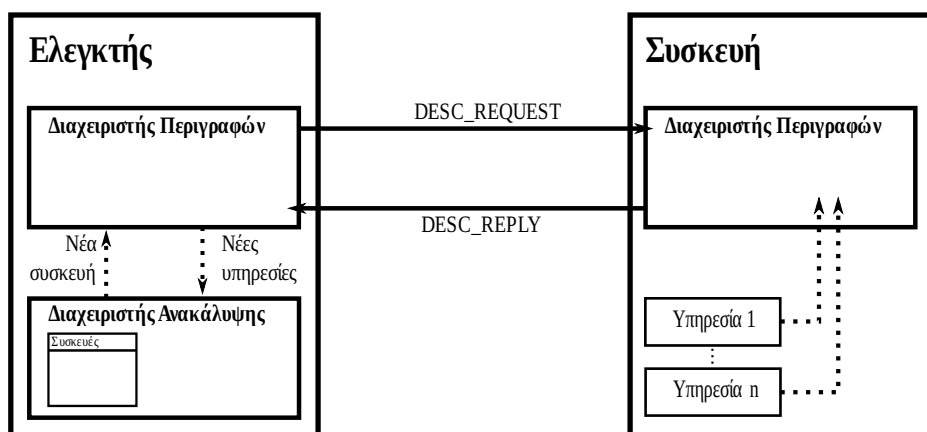
κτής. Όταν εντοπιστεί μια νέα συσκευή, ο Διαχειριστής του πρωτοκόλλου Ανακάλυψης ενημερώνει αυτόν της Περιγραφής, ο οποίος στη συνέχεια στέλνει το *DESC_REQUEST* μήνυμα στην συσκευή. Η συσκευή λαμβάνει το μήνυμα και απαντά με το *DESC_REPLY*, το οποίο περιέχει τις υπηρεσίες τις. Ο ελεγκτής της λαμβάνει τις υπηρεσίες και ενημερώνει την περιγραφή της συσκευής στον πίνακα συσκευών του διαχειριστή ανακάλυψης.

5.2.3 Συσχέτιση συσκευών

Η συσχέτιση των συσκευών, για την επιτέλεση κάποιας συνεργασίας μεταξύ τους, είναι μία από τις σημαντικότερες λειτουργίες του picoPnP. Είναι αυτή που επιτρέπει στον χρήστη να θέσει απλούς κανόνες για τις συσκευές ώστε να επιτευχθεί ο

Όπως φαίνεται το μήνυμα *DESC_REQUEST* δεν περιέχει κάποια επιπλέον πληροφορία. Αντίθετα, το μήνυμα *DESC_REPLY* περιέχει τα δεδομένα για την κάθε υπηρεσία που παρέχει η συσκευή. Αρχικά στο μήνυμα γράφεται το πλήθος των υπηρεσιών που αποστέλλονται και στη συνέχεια για κάθε μία από τις υπηρεσίες γράφεται το μέγεθος των δεδομένων της και έπειτα τα δεδομένα της σε byte array. Τα δεδομένα που χαρακτηρίζουν μια υπηρεσία είναι αυτά που αναφέρονται στη σελίδα 49. Ο ελεγκτής που ζήτησε τις υπηρεσίες από τη συσκευή λαμβάνει το μήνυμα απάντησης και διαβάζοντας τα πεδία του δημιουργεί την περιγραφή για κάθε μία από τις υπηρεσίες. Στη συνέχεια προσθέτει τις υπηρεσίες στη βασική περιγραφή της συσκευής, που είναι αποθηκευμένη στον πίνακα με τις συσκευές και τον οποίον χειρίζεται το πρωτόκολλο ανακάλυψης. Πλέον, για την συσκευή αυτή, διαθέτει την εκτενή περιγραφή της και είναι σε θέση να προβάλει τις υπηρεσίες στο χρήστη.

Στο Σχήμα 5.10 απεικονίζεται η διαδικασία για την ανάκτηση των υπηρεσιών των συσκευών από τον ελεγ-



Σχήμα 5.10: Διαδικασία ανάκτησης περιγραφής υπηρεσιών συσκευής από ελεγκτή. Όταν ο ελεγκτής πληροφορείται για την ύπαρξη μιας νέας συσκευής, ο Διαχειριστής Ανακάλυψης ενημερώνει το Διαχειριστή Περιγραφών με τη διεύθυνση της εν λόγω συσκευής. Ο τελευταίος επικοινωνεί με την συσκευή και ζητά τις υπηρεσίες της. Εκείνη απαντά με αυτές και ο Διαχειριστής Περιγραφών τις λαμβάνει και τις προωθεί στον Διαχειριστή Ανακάλυψης ώστε να ενημερώσει την περιγραφή της συσκευής αυτής.

αυτοματισμός λειτουργιών που επιθυμεί. Η ανάθεση των συσχετίσεων μεταξύ των συσκευών γίνεται από τους ελεγκτές, απ' όπου ο χρήστης έχει πρόσβαση στο δίκτυο. Εφόσον ένας ελεγκτής έχει εντοπίσει τις διαθέσιμες συσκευές στο δίκτυο και έχει αποκτήσει τις περιγραφές τους και τις υπηρεσίες τους, τότε μέσω αυτού και με κατάλληλες ενέργειες μπορεί να επιτευχθεί η συσχέτιση των υπηρεσιών των συσκευών.

Μια συσχέτιση μεταξύ δύο συσκευών περιγράφεται από τα εξής:

1. Τη MAC διεύθυνση αισθητήρα.
2. Τη MAC διεύθυνση ενεργοποιητή.
3. Το ID υπηρεσίας αισθητήρα.
4. Το ID υπηρεσίας ενεργοποιητή.
5. Πίνακας αντιστοίχισης παραμέτρων υπηρεσίας αισθητήρα με παραμέτρους εκείνης του ενεργοποιητή.
6. Η χρονική στιγμή της δημιουργίας της (timestamp).
7. Η κατάστασή της. Μια συσχέτιση μπορεί να είναι ενεργή ή ανενεργή, σε περιπτώσεις σφάλματος.

Τα τέσσερα πρώτα στοιχεία αποτελούν ένα συνδυασμό που χαρακτηρίζει μοναδικά τη κάθε συσχέτιση, ενώ για κάθε μία μπορούν να αντιστοιχίζονται περισσότερες από μία παράμετροι των δύο υπηρεσιών. Η αντιστοίχιση είναι «ένα προς

ένα», δηλαδή, μια συγκεκριμένη παράμετρος της υπηρεσίας του αισθητήρα μπορεί να αντιστοιχηθεί σε μία μόνο παράμετρο της υπηρεσίας του ενεργοποιητή. Με τον περιορισμό αυτό, αποφεύγονται οι αντιστοιχίες παραμέτρων που οδηγούν σε αντικρουόμενες ενέργειες.

Στο προηγούμενο παράδειγμα με τους **Light Sensor 1** και **Ambient Actuator 1**, μια πιθανή συσχέτισή τους είναι για την υπηρεσία **Light Monitor** του πρώτου, όταν η τιμή της έντασης φωτός είναι **LOW** τότε να θέτει την τιμή της υπηρεσίας **Light Operation** στην τιμή **100%** και αντίστοιχα όταν η τιμή της πρώτης είναι **HIGH** να τίθεται η τιμή της δεύτερης σε **OFF**.

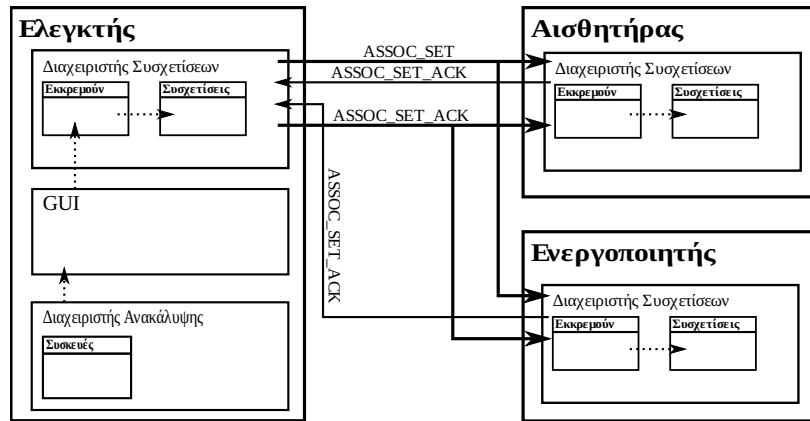
Ένας ελεγκτής πρέπει να παρέχει στο χρήστη τις εξής λειτουργίες, σχετικές με τις συσχετίσεις: *ανάθεση, ανανέωση, διαγραφή και ανάκτηση* συσχετίσεων. Το *picoPnP* υποστηρίζει τις λειτουργίες αυτές μέσω ενός Διαχειριστή Συσχετίσεων. Κάθε μία από αυτές αναλύεται στις επόμενες ενότητες.

Ανάθεση συσχέτισης στις συσκευές

Για την συσχέτιση δύο συσκευών, ο χρήστης, μέσω του γραφικού περιβάλλοντος του ελεγκτή, επιλέγει τον αισθητήρα και τον ενεργοποιητή που επιθυμεί να συσχετίσει, στην συνέχεια επιλέγει τις υπηρεσίες, μία από τον κάθε ένα, και τέλος αντιστοιχίζει τις παραμέτρους που επιθυμεί για την κάθε υπηρεσία (Σχήμα Α΄.2).

Ο ελεγκτής είναι υπεύθυνος να στείλει την συσχέτιση και στις δύο εμπλεκόμενες συσκευές. Επίσης πρέπει να εξασφαλίσει ότι και οι δύο συσκευές έλαβαν την συσχέτιση.

Ο Διαχειριστής Συσχετίσεων, που υπάρχει και στους ελεγκτές, αλλά και στις συσκευές, διαθέτει ένα κύριο πίνακα που αποθηκεύει τις συσχετίσεις και έναν βοηθητικό πίνακα που τοποθετούνται οι νέες συσχετίσεις, πριν αυτές «εγκριθούν». Αρχικά, δημιουργείται μια νέα συσχέτιση από τον ελεγκτή, τίθεται «ανενεργή» και τοποθετείται στο βοηθητικό πίνακα. Έπειτα, η συσχέτιση αυτή αποστέλλεται στις συσκευές με ένα μήνυμα **ASSOC_SET**, το οποίο παρουσιάζεται στο Σχήμα 5.12.1. Οι συσκευές λαμβάνουν το μήνυμα αυτό, διαβάζουν τα δεδομένα της συσχέτισης και την τοποθετούν στο βοηθητικό τους πίνακα. Η κάθε συσκευή στέλνει ένα μήνυμα επιβεβαίωσης **ASSOC_SET_ACK** το οποίο περιέχει το **timestamp** της συσχέτισης. Ο ελεγκτής όταν λάβει την επιβεβαίωση και από τις δύο συσκευές, εντοπίζει την συσχέτιση με βάση το **timestamp** αυτό και θέτει την συσχέτιση σε κατάσταση «ενεργή». Έπειτα, την μεταφέρει από το βοηθητικό πίνακα στον κύριο. Στη συνέχεια αποστέλλει ένα όμοιο μήνυμα επιβεβαίωσης στις συσκευές, ώστε να θέσουν ενεργή τη συσχέτιση και αυτές. Οι συσκευές λαμβάνουν την επιβεβαίωση αναζητούν από το βοηθητικό πίνακα την συσχέτιση με το **timestamp** που έλαβαν και την διεύθυνση του αποστολέα (ελεγκτή) και την ενεργοποιούν μεταφέροντάς την από τον βοηθητικό πίνακα στον κύριο. Μάλιστα, οι συσκευές αποθηκεύουν τη συσχέτιση αυτή και στη μόνιμη μνήμη **flash** που διαθέτουν τα **SPOT** ώστε να είναι διαθέσιμη



Σχήμα 5.11: Διαδικασία ανάθεσης συσχέτισης. Ο ελεγκτής στέλνει τη συσχέτιση και στις δύο συσκευές και περιμένει την επιβεβαίωση και από τις δύο. Όταν λάβει τις επιβεβαιώσεις, τότε θέτει την συσχέτιση ενεργή και επιβεβαιώνει με τη σειρά του τις συσκευές ώστε να ενεργοποιηθούν και εκείνες τη συσχέτιση.

ακόμα όταν κάποια συσκευή επανεκκινεί τη λειτουργία της. Με τον τρόπο αυτό ο χρήστης δεν χρειάζεται να θέτει τη συσχέτιση κάθε φορά που κάποια συσκευή τερματίζει τη λειτουργία της και ενεργοποιείται ξανά.

Αν για κάποιο λόγο ο ελεγκτής δεν λάβει την επιβεβαίωση και από τις δύο εμπλεκόμενες συσκευές τότε δεν στέλνει την επιβεβαίωσή του και η συσχέτιση παραμένει ανενεργή. Επομένως, με το τρόπο αυτό, της διπλής επιβεβαίωσης, εξασφαλίζεται πως και οι δύο συσκευές (αισθητήρας και ενεργοποιητής) ενημερώνονται για τη νέα συσχέτιση και μόνο τότε αυτή ενεργοποιείται. Στο Σχήμα 5.11 απεικονίζεται η διαδικασία που μόλις περιγράφηκε.

Στο σημείο αυτό, πρέπει να γίνει μια εξήγηση για τη χρήση του timestamp μιας συσχέτισης στη διαδικασία επιβεβαίωσης κατά την ανάθεση. Για τον ελεγκτή, που δημιουργεί τη συσχέτιση, το timestamp αυτής είναι μοναδικό, αφού τη δεδομένη χρονική στιγμή ο συγκεκριμένος μπορεί να δημιουργήσει μόνο μία. Αντίθετα, για μια συσκευή το timestamp αυτό δεν αποτελεί μοναδικό στοιχείο για τη συσχέτιση, γιατί μπορεί να υπάρχει μια άλλη, από άλλον ελεγκτή, η οποία να διαθέτει το ίδιο timestamp. Η πιθανότητα αυτή, αν και είναι μικρή, μπορεί να υπάρξει λόγω του ότι δεν υπάρχει κάποιος μηχανισμός εξασφάλισης συγχρονισμού των ελεγκτών. Επομένως, για μια συσκευή, η αναγνώριση μιας συσχέτισης κατά τη διάρκεια της ανάθεσης γίνεται με βάση το timestamp και την MAC διεύθυνση του ελεγκτή και συνεπώς στα μηνύματα ASSOC_SET_ACK το timestamp μπορεί να χρησιμοποιηθεί για την αναφορά στη συσχέτιση, μέχρι αυτή να ενεργοποιηθεί. Η τακτική επιβεβαίωσης με τη χρήση του timestamp μιας συσχέτισης υιοθετείται κυρίως με σκοπό την αποφυγή αποστολής μεγάλων σε μήκος μηνυμάτων⁵ και στο να βελτιώσει το χρόνο που χρειάζεται για να εντοπιστεί μια συσχέτιση που εκκρεμεί στον αντίστοιχο πίνακα.

⁵ Αντί για τέσσερα πεδία που χαρακτηρίζουν μοναδικά μια συσχέτιση, χρησιμοποιείται μόνο το timestamp.

ASSOC_SET message {

```

byte      messageType = ASSOC_SET

int       assocDataLength = το μέγεθος
                        δεδομένων που αποστέλλονται

byte[]    assocData = τα δεδομένα της
                        συσχέτισης
}

```

5.12.1: Το περιεχόμενο ενός μηνύματος ASSOC_SET.

ASSOC_GET message {

```

byte      messageType = ASSOC_GET
}

```

5.12.3: Το περιεχόμενο ενός μηνύματος ASSOC_GET.

ASSOC_SET_ACK message {

```

byte      messageType = ASSOC_SET_ACK

long      assocTimestamp = το timestamp
                        δημιουργίας της συσχέ-
                        τισης
}

```

5.12.2: Το περιεχόμενο ενός μηνύματος ASSOC_SET_ACK.

ASSOC_REPLY message {

```

byte      messageType = ASSOC_REPLY

int       totalOfAssociations = το
                        πλήθος των συσχετίσεων που
                        αποστέλλονται

        Για κάθε συσχέτιση{

int       assocDataLength = το μέγεθος
                        δεδομένων για την τρέχουσα
                        συσχέτιση

byte[]    assocData = τα δεδομένα της
                        τρέχουσας συσχέτισης

        }

}

```

5.12.4: Το περιεχόμενο ενός μηνύματος ASSOC_REPLY.

Σχήμα 5.12: Τα μηνύματα για που ανταλλάσσονται κατά την ανάθεση και ανάκτηση συσχετίσεων. Τα μηνύματα των Σχημάτων 5.12.1 και 5.12.2 αφορούν στη διαδικασία ανάθεσης, ενώ εκείνα στα Σχήματα 5.12.3 και 5.12.4 στη διαδικασία ανάκτησης.

Ανανέωση παραμέτρων υπάρχουσας συσχέτισης

Η ανανέωση μιας υπάρχουσας συσχέτισης συσκευών είναι η διαδικασία με την οποία ο χρήστης μπορεί να αλλάξει την αντιστοίχιση των παραμέτρων της. Μέσω της γραφικής διεπαφής στον ελεγκτή, ο χρήστης μπορεί να προβάλλει οποιαδήποτε υπάρχουσα συσχέτιση και να επεξεργαστεί την αντιστοίχιση των παραμέτρων των συσχετισμένων υπηρεσιών. Το να αλλάξει κάποια από τις υπηρεσίες των συσκευών δεν έχει νόημα, αφού πρόκειται για κάποια άλλη συσχέτιση, επομένως έχει νόημα μόνο η αλλαγή της αντιστοίχισης των παραμέτρων αυτών.

Η διαδικασία ανανέωσης των παραμέτρων μιας συσχέτισης είναι όμοια με εκείνη της ανάθεσης. Στην πραγματικότητα με την αποστολή μια ανανεωμένης συσχέτισης στις συσκευές, διαγράφεται η προηγούμενη και τοποθετείται στη θέση της μια νέα.

Ανάκτηση συσχετίσεων στους ελεγκτές

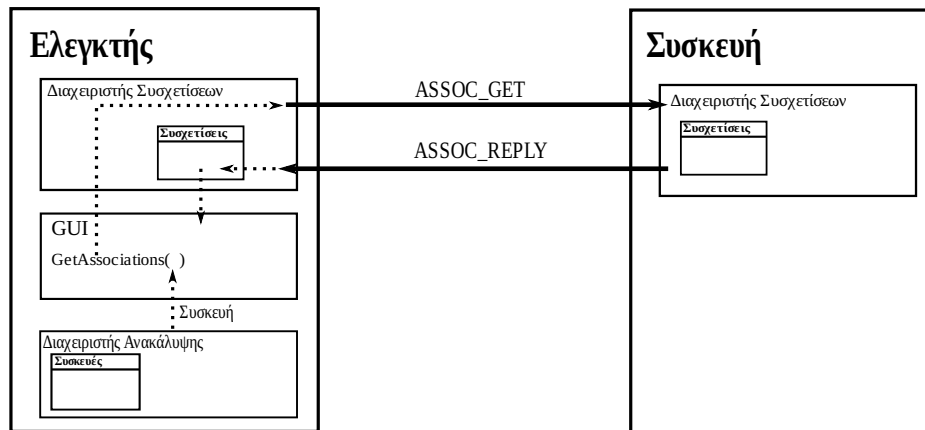
Κάθε φορά που συνδέεται ένας ελεγκτής στο δίκτυο, εκτός από την ανακάλυψη των συσκευών και την ανάκτηση των περιγραφών τους, πρέπει να ενημερώνεται και για τις υπάρχουσες συσχετίσεις κάθε συσκευής, ώστε να τις παρέχει στο χρήστη. Σε αντίθεση με την περιγραφή μιας συσκευής, η οποία δεν αλλάζει κατά την λειτουργία των κόμβων, οι συσχετίσεις μπορεί να μεταβάλλονται κατά τη διάρκεια λειτουργίας του ελεγκτή. Η μεταβολή αυτή μπορεί να είναι κάποια ανάθεση, ανανέωση ή διαγραφή συσχέτισης, που γίνεται από άλλους ελεγκτές, συνδεδεμένους στο δίκτυο.

Για την ανάκτηση των υπαρχουσών συσχετίσεων από κάποια συσκευή, ο ελεγκτής στέλνει ένα μήνυμα **ASSOC_GET** στην συσκευή ζητώντας να του επιστραφούν οι συσχετίσεις στις οποίες ανήκει αυτή. Η συσκευή που λαμβάνει το μήνυμα αυτό απαντά στον ελεγκτή με το μήνυμα **ASSOC_REPLY** στο οποίο περιέχονται οι ζητούμενες συσχετίσεις. Τα μηνύματα αυτά παρουσιάζονται στα Σχήματα 5.12.3 και 5.12.4 αντίστοιχα.

Ο ελεγκτής, αφού λάβει τις εν λόγω συσχετίσεις, τις τοποθετεί στο πίνακα συσχετίσεων, ανανεώνοντας τυχόν παλαιότερες, και ενημερώνει τον χρήστη. Η διαδικασία που μόλις περιγράφηκε απεικονίζεται στο Σχήμα 5.13.

Διαγραφή συσχέτισης

Ο χρήστης, στα πλαίσια της πλήρους διαχείρισης των συσχετίσεων, έχει τη δυνατότητα να διαγράφει υπάρχουσες συσχετίσεις συσκευών, καταργώντας με αυτόν τον τρόπο τη συνεργασία μεταξύ δύο συσκευών. Για τη διαγραφή μιας συσχέτισης στέλνεται από τον ελεγκτή ένα μήνυμα **ASSOC_DEL** στις δύο συσκευές, το οποίο περιέχει τα τέσσερα βασικά στοιχεία που χαρακτηρίζουν μια συσχέτιση. Το μήνυμα



Σχήμα 5.13: Η διαδικασία ανάκτησης συσχετίσεων συσκευής από τον ελεγκτή. Ο χρήστης ζητά τις συσχετίσεις μιας συσκευής. Ο Διαχειριστής Συσχετίσεων στέλνει ένα **ASSOC_GET** μήνυμα προς τη συσκευή. Εκείνη απαντά με ένα **ASSOC_REPLY** μήνυμα το οποίο περιέχει τις συσχετίσεις της. Στη συνέχεια, ο Διαχειριστής Συσχετίσεων του ελεγκτή τοποθετεί τις λαμβανόμενες συσχετίσεις στο αντίστοιχο πίνακα και ενημερώνει το χρήστη με αυτές.

αυτό παρουσιάζεται στο Σχήμα 5.14. Οι συσκευές που λαμβάνουν το μήνυμα διαγράφουν την συσχέτιση από τον πίνακα συσχετίσεων, βάσει των στοιχείων που περιέχονται σε αυτό. Εφόσον μια συσχέτιση διαγραφεί, παύει να ισχύει και οι συσκευές που συμμετείχαν σε αυτή δεν συνεργάζονται πλέον για την συγκεκριμένη ενέργεια. Στο Σχήμα 5.15 απεικονίζεται η προαναφερθείσα διαδικασία.

Μόνιμη αποθήκευση των συσχετίσεων στις συσκευές

Κάθε συσκευή έχει την δυνατότητα να αποθηκεύει τις συσχετίσεις στην μνήμη flash που διαθέτει, ώστε να μην χρειάζεται ο χρήστης να τις θέτει εκ νέου σε περίπτωση μεταβολής της κατάστασης λειτουργίας κάποιας συσκευής. Για την αποθήκευση των συσχετίσεων κάθε συσκευή διαθέτει ένα Διαχειριστή Αποθήκευσης, ο οποίος αναλαμβάνει την εισαγωγή, διαγραφή και ανάκτηση από την μόνιμη μνήμη.

Όταν ο Διαχειριστής Συσχετίσεων μιας συσκευής λάβει μια νέα συσχέτιση από κάποιον ελεγκτή, ενημερώνει τον Διαχειριστή Αποθήκευσης με την νέα αυτή συσχέτιση και εκείνος την αποθηκεύει στην μνήμη της συσκευής.

Κατ' αντίστοιχο τρόπο, όταν ο χρήστης διαγράφει μια συσχέτιση, οι εμπλεκόμενες συσκευές, λαμβάνοντας το μήνυμα διαγραφής, διαγράφουν την συσχέτιση αυτή και από την μόνιμη μνήμη τους. Τέλος, κατά την εκκίνηση λειτουργίας, μια συσκευή ανακτά από την μόνιμη μνήμη της τις υπάρχουσες συσχετίσεις και τις τοποθετεί στο πίνακα συσχετίσεων που διατηρεί ο Διαχειριστής Συσχετίσεων, ώστε να τίθενται σε ισχύ και να είναι διαθέσιμες στους ελεγκτές.

```

ASSOC_DEL message {

    byte      messageType = ASSOC_DEL

    long      sensorAddress = η διεύθυνση
                του αισθητήρα

    long      actorAddress = η διεύθυνση
                του ενεργοποιητή

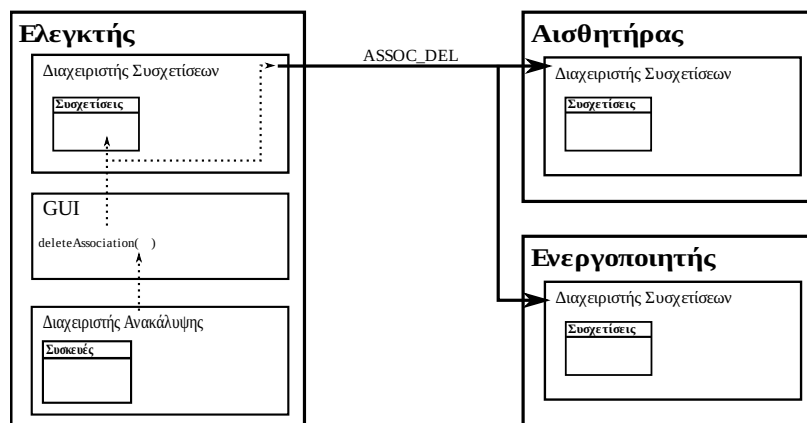
    int       sensorServiceID = το ID της
                υπηρεσίας του αισθητήρα

    int       actorServiceID = το ID της
                υπηρεσίας του ενεργοποιητή

}

```

Σχήμα 5.14: Το περιεχόμενο ενός μηνύματος ASSOC_DEL



Σχήμα 5.15: Η διαδικασία διαγραφής συσχέτισης. Ο χρήστης μέσω τους GUI επιλέγει τη συσχέτιση που αφορά κάποια συσκευή και επιθυμεί να διαγράψει. Στην συνέχεια ο Διαχειριστής Συσχετίσεων στέλνει ένα ASSOC_DEL μήνυμα προς τις δύο συσκευές που μετέχουν στη συσχέτιση αυτή. Οι δύο συσκευές που λαμβάνουν το μήνυμα αφαιρούν την συσχέτιση από τον πίνακα συσχετίσεων και την διαγράφουν και από την μόνιμη μνήμη τους.

5.2.4 Γεγονότα και ενέργειες

Όπως ήδη έχει αναφερθεί, ο σκοπός της εν λόγω αρχιτεκτονικής είναι η ανάθεση, από τον χρήστη, μιας από κοινού εργασίας στους αισθητήρες και τους ενεργοποιητές του δικτύου. Ο χρήστης, συσχετίζοντας δύο συσκευές, έχει θέσει τον τρόπο συνεργασίας μεταξύ τους. Στη συνέχεια, οι δύο συσχετιζόμενες συσκευές, επιτελούν τις αναμενόμενες λειτουργίες χωρίς την ανάγκη παρέμβασης του χρήστη. Ο τρόπος συνεργασίας μεταξύ των συσχετιζόμενων, αισθητήρα και ενεργοποιητή, γίνεται με την αναφορά ενός συμβάντος από τον πρώτο στον τελευταίο, το οποίο ικανοποιεί τα κριτήρια κάποιας συσχέτισης. Συγκεκριμένα, τα κριτήρια αυτά ορίζονται

```

EVENT_REPORT message {

    byte      messageType = EVENT_REPORT

    int       eventDataLength = το μέγεθος
                        των δεδομένων για το αναφε-
                        ρόμενο Event

    byte[]    eventData = τα δεδομένα του
                        Event

}

```

Σχήμα 5.16: Το περιεχόμενο ενός μηνύματος *EVENT_REPORT*.

από τις αντιστοιχίσεις των παραμέτρων της συσχέτισης των δύο εμπλεκόμενων υπηρεσιών των συσκευών.

Όταν ένας αισθητήρας αντιληφθεί κάποιο συμβάν, μέσω της ανάλογης υπηρεσίας, ελέγχει τον πίνακα συσχετίσεων του για ενδεχόμενες συσχετίσεις της υπηρεσίας αυτής με κάποια(ες) από τις υπηρεσίες των ενεργοποιητών. Αφού εντοπίσει τους ενεργοποιητές που σχετίζονται με την υπηρεσία που αντιλαμβάνεται το συμβάν, τότε στέλνει ένα μήνυμα προς αυτούς ώστε να ενεργήσουν

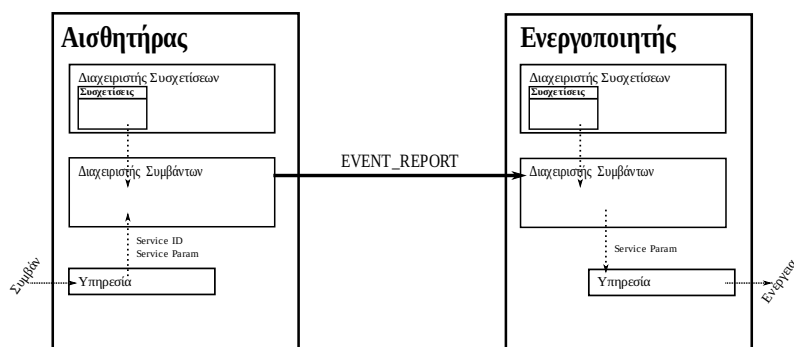
ανάλογα.

Για να εξασφαλιστεί ότι η ενέργεια που επακολουθεί του γεγονότος είναι σωστά ορισμένη και δεν υπάρχει ασυνέπεια μεταξύ των συσχετίσεων των δύο συσκευών, δεν στέλνεται από τον αισθητήρα στον ενεργοποιητή μόνο η «εντολή» για την εκτέλεση κάποιας ενέργειας. Συγκεκριμένα, ο αισθητήρας στέλνει όλα τα στοιχεία της συσχέτισής του που ικανοποιούνται από το συμβάν.

Αναλυτικότερα η διαδικασία ειδοποίησης συμβάντος έχει ως εξής: Κάθε συσκευή διαθέτει ένα Διαχειριστή Συμβάντων. Αυτός είναι υπεύθυνος για την αποστολή των μηνύματος ειδοποίησης σε περίπτωση κάποιου συμβάντος και αντίστοιχα για να λαμβάνει τέτοια μηνύματα. Μια υπηρεσία που αντιλαμβάνεται κάποιο συμβάν, ελέγχει τις συσχετιζόμενες με αυτήν υπηρεσίες των ενεργοποιητών, από τον πίνακα συσχετίσεων. Αφού εντοπίσει κάποια, τότε ελέγχει αν η τρέχουσα παράμετρος της, που αναλογεί στο συμβάν, σχετίζεται με κάποια από τις παραμέτρους της υπηρεσίας του ενεργοποιητή. Αν βρεθεί αντιστοίχιση παραμέτρων, τότε δημιουργεί ένα “Event”. Ένα Event χαρακτηρίζεται από τα εξής πεδία:

1. το ID της υπηρεσίας του αισθητήρα που αντιλαμβάνεται το συμβάν.
2. το ID της υπηρεσίας του ενεργοποιητή που σχετίζεται με εκείνη του αισθητήρα που αντιλαμβάνεται το συμβάν.
3. την παράμετρο της υπηρεσίας του αισθητήρα που αντιστοιχεί στο συμβάν.
4. την παράμετρο της υπηρεσίας του ενεργοποιητή με την οποία πρέπει να εκτελεστεί.
5. τη χρονική στιγμή που το συμβάν λαμβάνει χώρα (timestamp).

Το Event αυτό στέλνεται στον ενεργοποιητή, μέσω του Διαχειριστή Συμβάντων με ένα μήνυμα *EVENT_REPORT* (Σχήμα 5.16). Ο Διαχειριστής Συμβάντων του ενεργοποιητή λαμβάνει το μήνυμα αυτό και ελέγχει αν ο αισθητήρας που το έστειλε



Σχήμα 5.17: Διαδικασία ειδοποίησης γεγονότος για την εκτέλεση ενέργειας. Ο Διαχειριστής Γεγονότων του αισθητήρα λαμβάνει την ειδοποίηση για κάποιο συμβάν από την υπηρεσία, ελέγχει τις συσχετίσεις για την υπηρεσία αυτή και ειδοποιεί τον ενεργοποιητή που σχετίζεται με την υπηρεσία αυτή. Ο ενεργοποιητής λαμβάνει την ειδοποίηση, ελέγχει αν συσχετίζεται κάποια από τις υπηρεσίες τους με κάποια από εκείνες του αισθητήρα και αν υπάρχει αντιστοίχιση για την τιμή που λαμβάνει τότε καλεί την αρμόδια υπηρεσία με την κατάλληλη παράμετρο, ώστε να ενεργήσει. Ανάλογα με το αποτέλεσμα της εκτέλεσης απαντά στον αισθητήρα.

υπάρχει στον πίνακα των συσχετίσεων της συσκευής. Έπειτα ελέγχει αν η υπηρεσία που αναφέρει το συμβάν σχετίζεται με κάποια από τις υπηρεσίες του και αν αυτή συμπίπτει με εκείνη που περιέχεται στο Event. Τέλος, εξετάζει αν υπάρχει η αντιστοίχιση των παραμέτρων που στέλνεται με το Event στην συσχέτιση. Αν όλα τα παραπάνω ικανοποιούνται, ο Διαχειριστής Συμβάντων καλεί την υπηρεσία του ενεργοποιητή με την ανάλογη παράμετρο προς εκτέλεση, ενώ σε αντίθετη περίπτωση δεν εκτελεί καμία ενέργεια, αφού δεν υπάρχει συνέπεια ανάμεσα στις συσχετίσεις του αισθητήρα και των δικών του. Με τον τρόπο αυτό αποφεύγονται τυχόν ανεπιθύμητες ενέργειες που μπορεί να προκύψουν από την εκτέλεση της υπηρεσίας του ενεργοποιητή. Στο Σχήμα 5.17 απεικονίζεται η διαδικασία αναφοράς ενός συμβάντος από έναν αισθητήρα σε έναν ενεργοποιητή.

5.3 Υλοποίηση συσκευών και υπηρεσιών

Μέχρι τώρα έγινε μια περιγραφή των μηχανισμών που παρέχει το picoPnP στο χρήστη για την λειτουργία και την συνεργασία κόμβων του δικτύου. Όμως, όπως αναφέρθηκε στην αρχή του Κεφαλαίου, το picoPnP εκτός από την παροχή των μηχανισμών αυτών, δίνει επίσης τη δυνατότητα στο χρήστη, με απλό και μικρό σε μέγεθος κώδικα Java, να δημιουργήσει τις συσκευές και τις υπηρεσίες που αυτές θα παρέχουν.

Το picoPnP προσφέρει δύο abstract κλάσεις για την υλοποίηση ενός αισθητήρα ή ενός ενεργοποιητή. Οι κλάσεις αυτές, **Sensor** και **Actor** αντίστοιχα, μπορούν να επεκταθούν από τον χρήστη δημιουργώντας τη συσκευή που επιθυμεί. Το μόνο που

χρειάζεται να δηλωθεί στις υλοποιήσεις των κλάσεων αυτών είναι τα στοιχεία που συνθέτουν την περιγραφή της συσκευής. Συγκεκριμένα χρειάζεται να δηλωθούν το όνομα, το πλήθος των hops που αυτή εκπέμπει τη περιγραφή της και ο χρόνος ζωής της συσκευής. Τα υπόλοιπα στοιχεία δεν χρειάζεται να δηλωθούν, αφού τα δύο είναι δεδομένα και παρέχονται από τα Sun SPOT (MAC διεύθυνση και επίπεδο μπαταρίας), ενώ ο τύπος της συσκευής τίθεται ανάλογα από την κλάση κληρονομείται (**Sensor** ή **Actor**).

Επίσης, εκτός από τη βασική περιγραφή της συσκευής, πρέπει να δηλωθούν και οι υπηρεσίες που αυτή παρέχει. Ο χρήστης έχει στη διάθεσή του άλλες δύο abstract κλάσεις για την υλοποίηση των υπηρεσιών. Η πρώτη ονομάζεται **Sense** και κληρονομείται για την υλοποίηση των υπηρεσιών ενός αισθητήρα, ενώ η δεύτερη καλείται **Action** και χρησιμοποιείται για την υλοποίηση των υπηρεσιών ενός ενεργοποιητή. Εκτός από τα στοιχεία για την υπηρεσία που πρέπει να δηλωθούν, οι κλάσεις αυτές υποχρεώνουν την υλοποίηση κάποιων abstract μεθόδων. Ειδικότερα, η κλάση **Sense** απαιτεί την υλοποίηση της μεθόδου `start()`, η οποία εκκινεί την υπηρεσία και η **Action** απαιτεί την υλοποίηση της `act(int param)` για την εκτέλεση της ενέργειας ανάλογα με την παράμετρο `param` στον ενεργοποιητή. Τέλος, η υπηρεσία του αισθητήρα που αντιλαμβάνεται κάποιο συμβάν μπορεί να ενημερώσει του συσχετιζόμενους ενεργοποιητές με το **Event** καλώντας την μέθοδο `notifyActors(int param)` που κληρονομείται από την κλάση **Sense**. Η μέθοδος αυτή αναλαμβάνει την εύρεση των συσχετιζόμενων ενεργοποιητών και την αποστολή του **Event** μέσω του Διαχειριστή Συμβάντων.

Ο χρήστης προσθέτει τις υπηρεσίες που ορίζει στην συσκευή που υλοποιεί, και αφού προγραμματιστεί το Sun SPOT, είναι έτοιμη για λειτουργία.

Στο Παράρτημα Α'3 παρουσιάζεται μια σειρά απλών προγραμμάτων σε Java για την υλοποίηση συσκευών και υπηρεσιών.

5.4 Ανακεφαλαίωση

Η αρχιτεκτονική *picoPnP* αποτελεί μια πρόταση για την υλοποίηση εφαρμογών σε WSN με απλές ενέργειες και προσφέρει τις βασικές λειτουργίες για την δικτύωση των αισθητήρων και των ενεργοποιητών, με σκοπό την συσχέτισή τους για την επιτέλεση κάποιας συνεργασίας. Παράλληλα, παρέχει μια γραφική διεπαφή, μέσω της οποίας ο χρήστης δύναται να διαχειρίζεται τους αισθητήρες και τους ενεργοποιητές του δικτύου, δημιουργώντας συσχετίσεις των συσκευών. Επιπλέον, δίνει τη δυνατότητα στο χρήστη για την συγγραφή απλών εφαρμογών για τους κόμβους του δικτύου, χρησιμοποιώντας ελάχιστο κώδικα Java ME.

Στο Κεφάλαιο αυτό αρχικά ορίστηκαν οι τύποι των κόμβων που υπάρχουν σε ένα δίκτυο που υλοποιεί την αρχιτεκτονική αυτή καθώς και τα βασικά στοιχεία που περιγράφουν μια συσκευή, μια υπηρεσία. Έγινε μια αναλυτική περιγραφή του πρωτο-

κόλλου ανακάλυψης συσκευών που χρησιμοποιείται καθώς και του πρωτοκόλλου περιγραφής, με το οποίο ο χρήστης πληροφορείται για τις υπηρεσίες που παρέχει μια συσκευή. Έπειτα περιγράφηκε πώς ορίζεται μια συσχέτιση μεταξύ ενός αισθητήρα και ενός ενεργοποιητή, καθώς και το πρωτόκολλο συσχέτισης που θέτει την συσχέτιση αυτή στις εμπλεκόμενες συσκευές. Στη συνέχεια αναφέρθηκε και αναλύθηκε ο τρόπος με τον οποίο λειτουργούν δυο συσχετιζόμενες συσκευές, δηλαδή όταν προκύψει κάποιο συμβάν και το αντιληφθεί ο αισθητήρας πώς ενημερώνεται ο συσχετιζόμενος ενεργοποιητής ώστε να δράσει ανάλογα με το συμβάν. Τέλος, περιγράφηκε ο τρόπος με τον οποίο ο χρήστης μπορεί να αναπτύξει τις δικές του εφαρμογές για τους αισθητήρες και ενεργοποιητές.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντικές Προεκτάσεις

Όπως έχει ήδη προαναφερθεί στην Εισαγωγή, οι έως τώρα υλοποιήσεις και προσεγγίσεις των WSN βρίσκονται ως επί το πλείστον σε στάδιο έρευνας και απευθύνονται σε εξειδικευμένους χρήστες. Οι έως τώρα πλατφόρμες για την διαχείριση και τον έλεγχο δικτύων λαμβάνουν υπ' όψιν τη χρήση τους σε δίκτυα υπολογιστών. Αντίθετα, για τα ασύρματα δίκτυα αισθητήρων υπάρχουν πολλές προτάσεις κατά καιρούς, αλλά η καθεμιά τους προσεγγίζει τις απαιτήσεις της κάθε εφαρμογής. Αυτό έχει ως αποτέλεσμα να μην έχει προκύψει κάτι καθολικό στην διαχείριση και τον έλεγχο των εν λόγω δικτύων και το σημαντικότερο ότι απαιτούνται ειδικές γνώσεις για την δημιουργία, λειτουργία και διαχείριση εφαρμογών πάνω σε WSN.

Πρόθεσή μας λοιπόν, με την συγκεκριμένη πρόταση, είναι να καλύψουμε αυτή την ανάγκη που έχει αναδειχθεί, προτείνοντας ένα σύστημα διαχείρισης ενός WSN το οποίο θα επιτρέπει, ακόμα και σε μη εξειδικευμένους χρήστες, τον έλεγχο και την λειτουργία ενός τέτοιου δικτύου, στοχεύοντας στην επιτέλεση συντονισμένων ενεργειών.

Για την υλοποίηση της προτεινόμενης πλατφόρμας, προηγήθηκε η μελέτη πάνω στα Ασύρματα Δίκτυα Αισθητήρων και Ενεργοποιητών, ώστε να παρουσιαστούν οι προκλήσεις που προκύπτουν από την προσθήκη των ενεργοποιητών στο δίκτυο. Τονίστηκε η σημαντικότητα για την απόκριση σε πραγματικό χρόνο που ενέχεται σε εφαρμογές γι αυτά τα δίκτυα, διάφοροι τρόποι με τους οποίους μπορούν να οργανωθούν και να συντονιστούν οι κόμβοι, με σκοπό την λήψη της σωστής απόφασης για κάποια ενέργεια, βασισμένη στις παρατηρήσεις των αισθητήρων, από τους ενεργοποιητές. Αναδείχθηκαν τα θέματα που προκύπτουν σχετικά με την επιλογή του κατάλληλου ενεργοποιητή για την εκτέλεση κάποιας εργασίας όσον αφορά την περιοχή δράσης του, το ενεργειακό απόθεμά του και την τήρηση των απαιτήσεων για ενέργεια σε πραγματικό χρόνο.

Μελετήθηκε επίσης το πρωτόκολλο UPnP, το οποίο αποτελεί μια διαδεδομένη πρόταση για την αδιάλειπτη και αόρατη δικτύωση συσκευών. Η τεχνολογία αυτή κα-

τέχει μεγάλο μέρος στην δικτύωση IP συσκευών και έχει μεγάλη απήχηση σε εμπορικές εφαρμογές. Το γεγονός αυτό οφείλεται στην ευκολία που προσφέρει στην περιγραφή των συσκευών και των υπηρεσιών που αυτές προσφέρουν, χρησιμοποιώντας διαδεδομένα πρωτόκολλα όπως η XML, καθώς επίσης και στην διασύνδεσή τους μέσω κυρίαρχων πρωτοκόλλων. Επίσης, περιγράφηκε ο τρόπος λειτουργίας του πρωτοκόλλου UPnP που ενέπνευσε στην ιδέα για την υλοποίηση της προτεινόμενης πλατφόρμας στην εργασία αυτή.

Επιπλέον, παρουσιάστηκαν κάποιες εργασίες στις οποίες γίνεται η προσπάθεια για την σύνδεση των ασύρματων δικτύων αισθητήρων με δίκτυα UPnP, με σκοπό την υλοποίηση τρόπων για την διαχείριση τους. Στην πλειονότητα των εργασιών που εξετάστηκαν προτείνεται η γεφύρωση των δύο δικτύων μέσω ειδικών κόμβων. Οι κόμβοι αυτοί αναλαμβάνουν την μετάφραση των μηνυμάτων καθώς μεταφέρονται από τον έναν τύπο δικτύου στον άλλον και παράλληλα την παροχή κάποιων ειδικών λειτουργιών που αποσκοπούν στον έλεγχο και τον περιορισμό της συνολικής κατανάλωσης ενέργειας του δικτύου αισθητήρων. Επίσης, υπήρξαν και προτάσεις για κόμβους που υλοποιούν τις βασικές λειτουργίες του UPnP –αυτές που δεν απαιτούν αρκετή επεξεργαστική ισχύ– στους οποίους συνδέονται αισθητήρες και ενεργοποιητές. Η τακτική αυτή προσφέρει την άμεση υλοποίηση του UPnP σε ένα δίκτυο αισθητήρων και ενεργοποιητών, δίνοντας άμεσα πρόσβαση σε αυτό, μέσω συσκευών, όπως PDAs και H/Y. Ακόμα, παρουσιάστηκε μέσω των εργασιών αυτών η ανάγκη για την δημιουργία μια απλής συμβολικής γλώσσας προγραμματισμού η οποία θα διευκολύνει πολύ στην συγγραφή προγραμμάτων για τη λειτουργία των κόμβων.

Στην συνέχεια, και λαμβάνοντας υπ' όψιν όλα τα προαναφερθέντα, παρουσιάστηκε η αρχιτεκτονική picoPnP η οποία έχει ως στόχο την υλοποίηση εφαρμογών για WSN και την διαχείριση τους, κυρίως για αυτοματισμούς σε κλειστούς χώρους. Η βασική ιδέα της προτεινόμενης αρχιτεκτονικής στηρίζεται εν μέρει στη φιλοσοφία του UPnP, χωρίζοντας τις διάφορες λειτουργίες σε στάδια ανάλογα με αυτά, και στην ευχέρεια που δίνει στο χρήστη με την συγγραφή απλών προγραμμάτων για τους κόμβους, να υλοποιεί τις δικές τους εφαρμογές. Παρουσιάστηκε αναλυτικά ο τρόπος με τον οποίον κατηγοριοποιούνται και περιγράφονται οι κόμβοι στο δίκτυο και οι υπηρεσίες τους. Αναλύθηκαν οι διάφορες παρεχόμενες λειτουργίες που προσφέρει η εν λόγω αρχιτεκτονική και δόθηκε σημασία στον συσχετισμό των αισθητήρων με τους ενεργοποιητές για την επίτευξη κάποιας συνεργασίας.

6.1 Μελλοντικές Προεκτάσεις

Κατά τη διάρκεια ενασχόλησης με παρούσα εργασία διαπιστώθηκε η δυσκολία του εγχειρήματος. Η προτεινόμενη αρχιτεκτονική αποτελεί μια αρχική προσέγγιση για την δημιουργία και διαχείριση εφαρμογών σε WSN. Προσφέρει τις απαραίτητες

λειτουργίες για την ανακάλυψη, την περιγραφή και συνεργασία των κόμβων του δικτύου. Υπάρχουν, όμως, αρκετές λειτουργίες που απουσιάζουν, οι οποίες θα βελτίωναν αρκετά τις υπάρχουσες, αλλά και άλλες που θα επέκτειναν τις δυνατότητες της εν λόγω αρχιτεκτονικής. Στην ενότητα αυτή αναφέρονται συνοπτικά κάποιες από αυτές τις βελτιώσεις και επεκτάσεις.

Οι επεκτάσεις που μπορούν να διευρύνουν τις λειτουργίες του picoPnP θα παρουσιαστούν σε τρεις άξονες. Ο πρώτος έχει να κάνει με την οργάνωση των κόμβων στο δίκτυο. Στην παρούσα φάση η υποδομή του δικτύου στηρίζεται αποκλειστικά στις δυνατότητες της δρομολόγησης και της multi-hop εκπομπής μηνυμάτων που προσφέρουν τα Sun SPOTs. Μια σημαντική βελτίωση, που δεν θα στηρίζεται στην multi-hop εκπομπή μηνυμάτων, η οποία θεωρείται αναξιόπιστη, θα είναι η οργάνωση των κόμβων σε συστάδες με κέντρα τους ενεργοποιητές. Αυτό θα αποτρέψει την εκπομπή των μηνυμάτων σε μεγάλο βάθος στο δίκτυο και θα την περιορίσει στους άμεσα γειτονικούς κόμβους. Για τη δυνατότητα αυτή υπάρχουν αρκετές προτάσεις σε άλλες εργασίες, όπως μερικές από αυτές που αναφέρθηκαν εν συντομία στο Κεφάλαιο 2 στη σελίδα 15. Μερικά από τα πλεονεκτήματα μιας τέτοιας οργάνωσης είναι ο περιορισμός της συνολικής κατανάλωσης ενέργειας στο δίκτυο, η αξιοπιστία στην μετάδοση και η ελαχιστοποίηση των καθυστερήσεων.

Επιπλέον, στη ειδοποίηση των ενεργοποιητών για τα συμβάντα από τους αισθητήρες, ο μηχανισμός για την εξακρίβωση της ορθής ενέργειας από τους πρώτους μπορεί να ενισχυθεί με μία απάντηση προς τους ενεργοποιητές, η οποία θα περιλαμβάνει το αποτέλεσμα της ενέργειας, ώστε να σημειώνονται οι ενδεχόμενες αποτυχίες και να ενημερώνεται ο χρήστης, όταν συνδέεται μέσω ενός ελεγκτή. Η τακτική αυτή θα αυξήσει την αξιοπιστία του συστήματος.

Στο θέμα της διαχείρισης του δικτύου, μια σημαντική προσθήκη στην παρούσα αρχιτεκτονική είναι η δυνατότητα συσχέτισης περισσότερων των δύο κόμβων για την επίτευξη κάποιας λειτουργίας. Μια τέτοια δυνατότητα διευρύνει σημαντικά τις επιλογές του χρήστη, ορίζοντας με περισσότερη λεπτομέρεια τις αυτοματοποιημένες λειτουργίες. Για παράδειγμα, ο χρήστης θα μπορεί να συσχετίζει τις τιμές από δύο ή και περισσότερους αισθητήρες με έναν ενεργοποιητή, πετυχαίνοντας μεγαλύτερη ακρίβεια για την ενέργεια που επιθυμεί. Στην περίπτωση αυτή πρέπει να οριστεί μια συμβολική γλώσσα, με την οποία ο χρήστης θα θέτει τους κανόνες για τις συσχετιζόμενες συσκευές και τις υπηρεσίες τους. Οι κανόνες θα πρέπει να ορίζονται με τον απλούστερο δυνατό τρόπο, ώστε να αποφεύγεται η σύγχυση.

Ακόμα, στο picoPnP μπορεί να προστεθεί ο άμεσος έλεγχος και η παρατήρηση των κόμβων από το περιβάλλον διαχείρισης. Συγκεκριμένα, ο χρήστης θα είναι σε θέση να παρατηρεί τις μετρήσεις των αισθητήρων και να ελέγχει τις υπηρεσίες των ενεργοποιητών.

Επίσης, στη προτεινόμενη αρχιτεκτονική μπορεί να προστεθεί η δυνατότητα για την διατήρηση ιστορικού συμβάντων και των ενεργειών που εκτελέστηκαν. Ο μηχανισμός αυτός συνδυάζεται με εκείνον που αναφέρθηκε παραπάνω, σχετικά με

την αναφορά για το αποτέλεσμα της δράσης σε κάποιο συμβάν.

Τέλος, όσον αφορά την υλοποίηση των προγραμμάτων για τους κόμβους, η παρούσα διαδικασία μπορεί να επεκταθεί ακόμα περισσότερο, ενεργοποιώντας τον ασύρματο προγραμματισμό των συσκευών (Over The Air). Ο χρήστης θα μπορεί να υλοποιεί την εφαρμογή του για τον κόμβο και να την στέλνει, μέσω του περιβάλλοντος διαχείρισης, σε αυτόν, διευκολύνοντας σημαντικά τον προγραμματισμό τους.

Βιβλιογραφία

- [ACYW02] I. F. Akyildiz, E. Cayirci, Sankarasubramaniam Y., and Su W. Wireless Sensor Networks: A Survey. *Computer Networks (Elsevier) Journal*, 38:393–422, March 2002.
- [AK04] I. F. Akyildiz and I. H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2:351–367, 2004.
- [BdI⁺06] C. Blondia, F. de Greve, G. Impens, I. Moerman, Jeroen Avonts, K. Lamont, M. Mertens, W. Vandenberghe, and B. Latré. A System Architecture for Wireless Building Automation. In *15th IST Mobile & Wireless Communications Summit 2006*, June 2006.
- [BLRB05] Waylon Brunette, Jonathan Lester, Adam D. Rea, and Gaetano Borriello. Some sensor network elements for ubiquitous computing. In *IPSN '05: Fourth International Symposium on Information Processing in Sensor Networks*, pages 388–392, 2005.
- [BLV09] Remi Bosman, Johan Lukkien, and Richard Verhoeven. Integrating wireless sensor networks with ce devices for health care activity tracking in the home environment. *Computers in Education, International Conference on*, 0:1–2, 2009.
- [BMM06] F. Bouhafs, M. Merabti, and H. Mokhtar. A Coordination Protocol for Wireless Sensor and Actor Networks. In *PGNet '06: 7th Annual PostGraduate Symposium on the convergence of telecommunications, networking and broadcasting*, 2006.
- [CA98] J. Cohen and S. Aggarwal. General Event Notification Architecture Base. <http://tools.ietf.org/html/draft-cohen-gena-p-base-01>, July 1998.
- [CMN07] I. Chatzigiannakis, G. Mylonas, and S. Nikoletseas. 50 ways to build your application: A Survey of Middleware and Systems for Wireless Sensor Networks. In *ETFA '07: IEEE Conference on Emerging Technologies and Factory Automation, 2007.*, pages 466–473, September 2007.

- [cor] Object Management Group, Inc. <http://www.corba.org>.
- [Dev00] DevelopMentor Don Box and IBM David Ehnebuske and Microsoft Gopal Kakivaya, Andrew Layman, Henrik Frystyk Nielsen, Satish Thatte and Lotus Development Corp. Noah Mendelsohn and UserLand Software Inc. Dave Winer. Simple Object Access Protocol (SOAP) 1.1. <http://www.w3.org/TR/2000/NOTE-SOAP-20000508>, May 2000. W3C Note.
- [Dro97] R. Droms. Dynamic Host Configuration Protocol. RFC 2131 (Draft Standard), March 1997. Updated by RFCs 3396, 4361, 5494.
- [Fer07] Pedro Sanchez Cristina Vicente-Chicote Barbara Alvarez Andres Iborra Fernando Losilla. A WSN Solution for Irrigation Control from a Model Driven Perspective. In *Wireless Sensor and Actor Networks*, volume 248 of *IFIP International Federation for Information Processing*, pages 35–46. Springer Boston, 2007.
- [FGM⁺99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. Updated by RFC 2817.
- [GSS⁺04] Yvonne Gsottberger, Xiaolei Shi, Guido Stromberg, Thomas F. Sturm, and Werner Weber. Embedding low-cost wireless sensors into universal plug and play environments. In *EWSN*, pages 291–306, 2004.
- [HMEZ⁺05] Sumi Helal, William Mann, Hicham El-Zabadani, Jeffrey King, Youssef Kaddoura, and Erwin Jansen. The Gator Tech Smart House: A Programmable Pervasive Space. *Computer*, 38(3):50–60, March 2005.
- [jxt] JXTATM Community Project. <http://www.jxta.org>.
- [KLSS05] Daeyoung Kim, Kangwoo Lee, Jongwoo Sung, and Hyungjoo Song. UPnP-Based Sensor Network Management Architecture. In *ICMU '05: International Conference on Mobile Computing and Ubiquitous Networking*, 2005.
- [KM09] Fatih Senel Kemal Akkaya and Brian McLaughlan. Clustering of wireless sensor and actor networks based on sensor distribution and connectivity. *Journal of Parallel and Distributed Computing*, 69:573–587, 2009.
- [LDCO07] W. L. Lee, A. Datta, and R. Cardell-Oliver. , chapter Network management in wireless sensor networks. *Handbook of Mobile Ad Hoc and Pervasive Communications*. American Scientific Publishers, 2007.

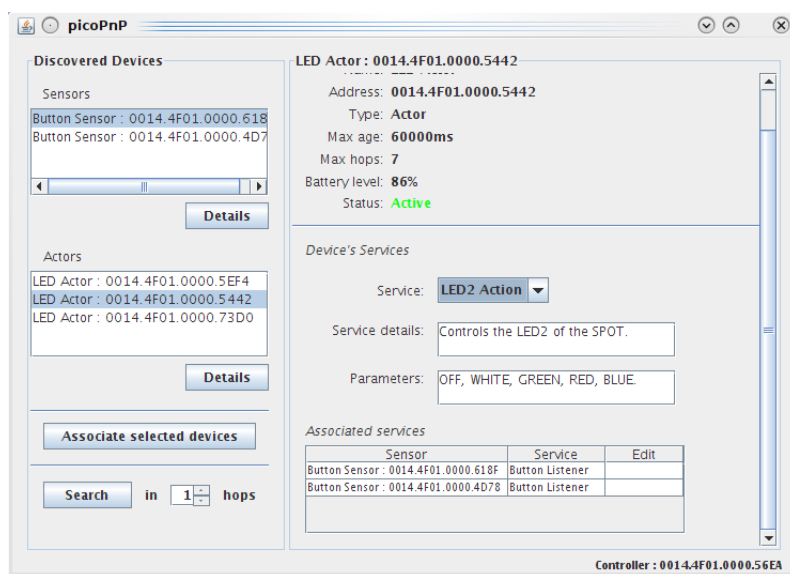
- [MF06] Muhammad Farukh Munir and Fethi Filali. A novel self organizing framework for sanets. In *EW '06: 12th European Wireless Conference*, 2006.
- [MP07] Maximilian Nicolae Radu Dobrescu Matei Dobrescu and Dan Popescu. Embedding wireless sensors in UPnP services networks. *International Journal of Communications*, 1(2):62–67, 2007.
- [MPGA05] Tommaso Melodia, Dario Pompili, Vehbi C. Gungor, and Ian F. Akyildiz. A distributed coordination framework for wireless sensor and actor networks. In *MobiHoc '05: The ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 99–110, 2005.
- [ND04] A. Nisbet and S. Dobson. A Systems architecture for sensor networks based on hardware/software co-design. In *1st IFIP Workshop on Automatic Communications*. Springer Verlag, 2004.
- [osg] Open Service Gateway initiative (OSGi), OSGi alliance. <http://www.osgi.org>.
- [PB07] Daniel Jacobi Pablo Ezequiel Guerrero and Alejandro Buchmann. Workflow Support for Wireless Sensor and Actor Networks. In *DMSN '07: 4th International Workshop on Data Management for Sensor Networks*, pages 31–36, 2007.
- [PBRD03] C. Perkins, E. Belding-Royer, and S. Das. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561, July 2003.
- [PMG⁺00] D. C. Petriu, D. Makrakis, V. Z. Groza, E. M. Petriu, and N .D. Georganas. Sensor-based information appliances. *IEEE Instrumentation and Measurement Magazine*, 3(4):31–35, 2000.
- [S. 05] B. Aboba E. Guttman S. Cheshire. Dynamic Configuration of IPv4 Link-Local Addresses. RFC 3927 (Draft Standard), May 2005.
- [sun] Sun SPOT World. <http://www.sunspotworld.com>.
- [SVB09] Martin Strohbach, Jesús Bernat Vercher, and Martin Bauer. A case for ims - harvesting the power of ubiquitous sensor networks. *Vehicular Technology Magazine, IEEE*, 4(1):57–64, March 2009.
- [The03] The Institute of Electrical and Electronics Engineers, Inc. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Press*, October 2003.

- [UPn08] UPnP Forum. UPnP Device Architecture 1.1. <http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf>, October 2008.
- [Wei93] Mark Weiser. Some Computer Science Issues in Ubiquitous Computing. *Communications of the ACM*, 36(7):75–84, July 1993.
- [Yar99] Ting Cai Paul Leach Ye Gu Shivaun Albright Yaron Y. Goland. Simple Service Discovery Protocol/1.0 Operating without an Arbiter. <http://tools.ietf.org/id/draft-cai-ssdp-v1-03.txt>, October 1999. Expired April 2000.

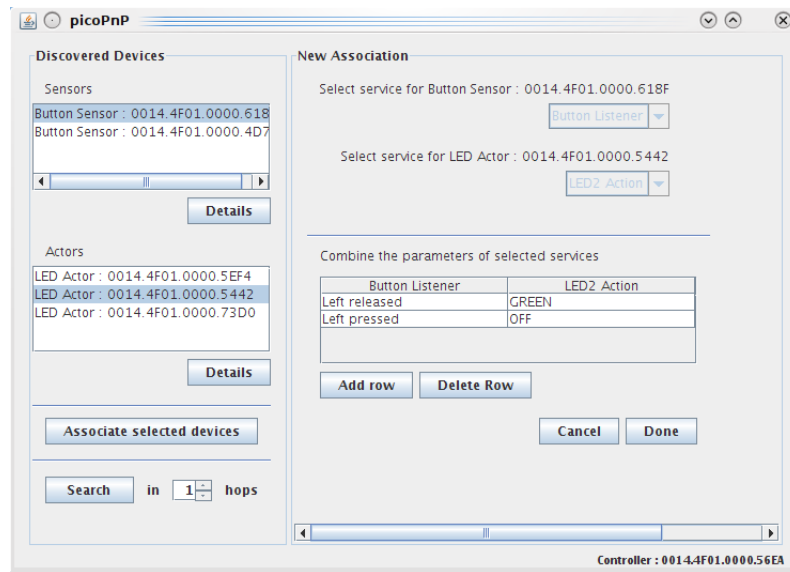
Παράρτημα Α'

picoPnP – Τεχνικές λεπτομέρειες

Α'.1 Screenshots από την εφαρμογή του ελεγκτή



Σχήμα Α'.1: Γραφικό περιβάλλον διαχείρισης δικτύου. Προβολή συσκευής. Στα αριστερά εμφανίζεται η λίστα με τις διαθέσιμες συσκευές, κατηγοριοποιημένες ανάλογα με τον τύπο τους. Επιλέγοντας έναν αισθητήρα ή έναν ενεργοποιητή και πατώντας το αντίστοιχο κουμπί “Details” παρουσιάζονται στο δεξιό οι λεπτομέρειες σχετικά με την επιλεγμένη συσκευή. Στο πάνω μέρος εμφανίζεται η βασική περιγραφή της συσκευής. Ακολουθεί μια λίστα με τις υπηρεσίες που αυτή παρέχει. Για κάθε υπηρεσία που επιλέγεται παρουσιάζεται τα σύντομα σχόλια για αυτήν, καθώς και οι παράμετροί της. Τέλος, στο κάτω μέρος παρουσιάζονται οι συσχετίσεις της προβαλλόμενης συσκευής.



Σχήμα Α'.2: Γραφικό περιβάλλον διαχείρισης δικτύου. Δημιουργία συσχέτισης. Επιλέγοντας έναν αισθητήρα και έναν ενεργοποιητή από τις λίστες στα αριστερά και πατώντας το κουμπί “Associate selected devices” εμφανίζεται στα δεξιά η δυνατότητα για την συσχέτιση τους. Αρχικά επιλέγονται οι υπηρεσίες από τις δύο συσκευές και στην συνέχεια αντιστοιχίζονται οι παράμετροι τους. Με το πάτημα του κουμπιού “Done” η συσχέτιση αποστέλλεται στις συσκευές.

Α'.2 UML διαγράμματα των βασικών κλάσεων

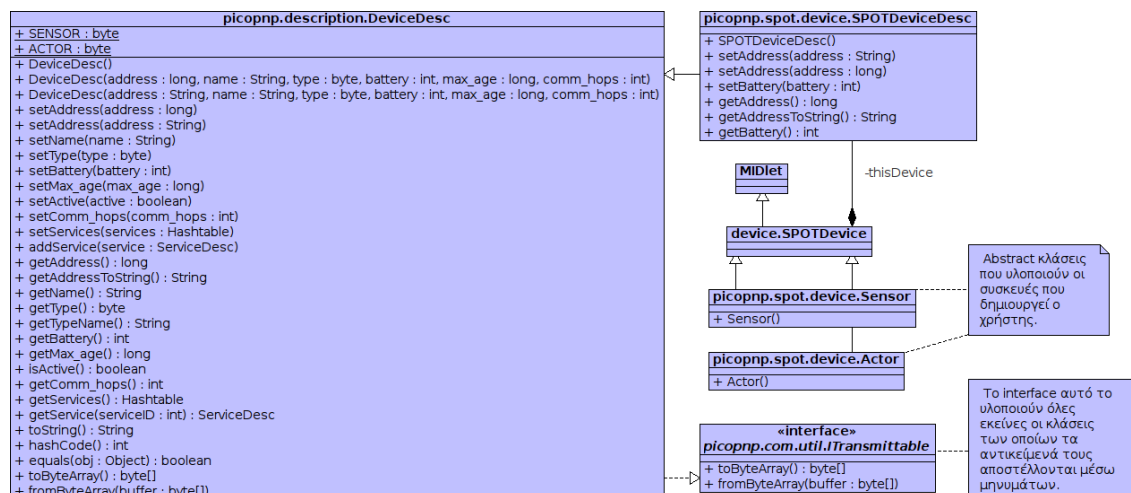
Α'.2.1 Συσκευές και Περιγραφή Συσκευής

Στο Σχήμα Α'.3 παρουσιάζονται οι κλάσεις που σχετίζονται με τις συσκευές και την περιγραφή τους. Η κλάση **DeviceDesc** είναι η κλάση που περιέχει τα στοιχεία που περιγράφουν μια συσκευή. Υλοποιεί κάποιες μεθόδους για την ανάθεση και ανάκτηση των τιμών (Set και Get μέθοδοι). Επιπλέον, η κλάση αυτή υλοποιεί τη διεπαφή **ITransmittable** η οποία υποχρεώνει το καθορισμό των μεθόδων `toByteArray():byte[]` και `fromByteArray(buffer:byte[])`.

Η κλάση **SPOTDevice** είναι μια abstract κλάση η οποία επεκτείνει της κλάση **MIDlet**. Η τελευταία είναι η βασική κλάση της εφαρμογής που εκτελείται από τα Sun SPOTs. Η **SPOTDevice** διαθέτει μια περιγραφή για την συσκευή την **SPOTDeviceDesc** η οποία είναι μια επέκταση της **DeviceDesc** και περιγράφει την τρέχουσα συσκευή που εκτελεί την εφαρμογή. Η κλάση **SPOTDevice** κληρονομείται από άλλες δύο abstract κλάσεις, **Sensor** και **Actor**, των οποίων οι υλοποιήσεις αποτελούν τις τελικές εφαρμογές για έναν αισθητήρα ή έναν ενεργοποιητή, αντίστοιχα.

Η διεπαφή **ITransmittable** χρησιμοποιείται από τις κλάσεις των οποίων τα αντικείμενα αποστέλλονται με datagrams. Μια κλάση που υλοποιεί την διεπαφή αυτή ορίζει τις μεθόδους `toByteArray():byte[]` και `fromByteArray(buffer:byte[])`. Μέσω των

μεθόδων αυτών ορίζεται ποια από τα στοιχεία του αντικειμένου θα εξαχθούν σε ένα byte array ή θα εισαχθούν από ένα byte array, αντίστοιχα. Αυτή η διαδικασία διευκολύνει κατά την αποστολή των στοιχείων, που αφορούν το αντικείμενο, στέλνοντας μόνο το byte array και κατά την λήψη όταν δημιουργείται το αντικείμενο από το byte array που λήφθηκε.

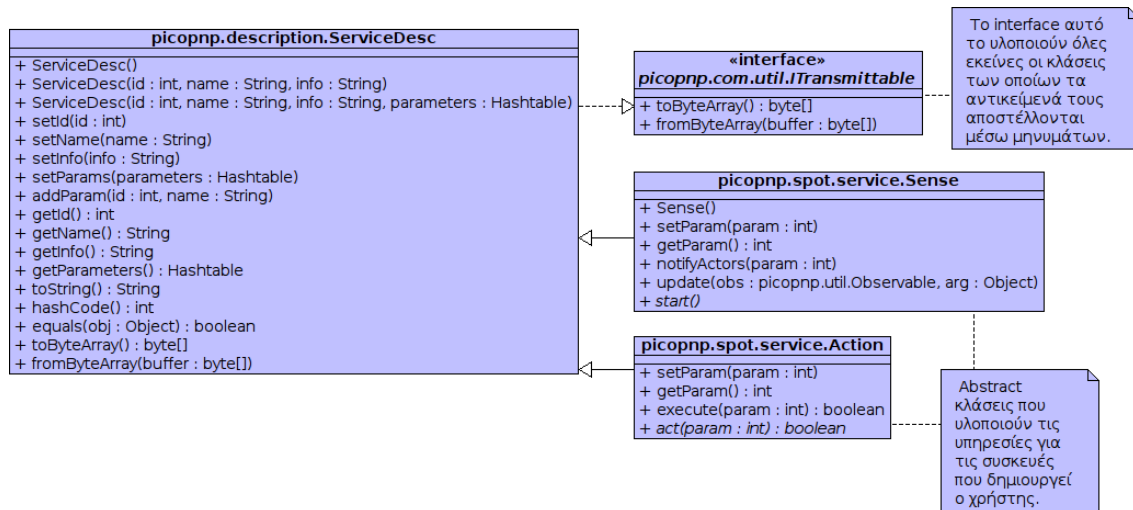


Σχήμα Α'.3: UML Διάγραμμα κλάσεων συσκευής.

Α'.2.2 Υπηρεσίες και Περιγραφή Υπηρεσίας

Στο Σχήμα Α'.4 παρουσιάζεται το UML διάγραμμα κλάσεων που σχετίζονται με τις υπηρεσίες των συσκευών. Η περιγραφή μια υπηρεσίας από την κλάση `ServiceDesc`, όπου ορίζονται όλα τα στοιχεία που χαρακτηρίζουν μια υπηρεσία, κληρονομείται από τις abstract κλάσεις `Sense` και `Action`. Οι υλοποιήσεις των τελευταίων αποτελούν τις υπηρεσίες για τους αισθητήρες και ενεργοποιητές, αντίστοιχα. Η κλάση `Sense` παρέχει την μέθοδο `notifyActors(param:int)` με την οποία ειδοποιούνται οι συσχετιζόμενοι ενεργοποιητές ανάλογα με το συμβάν που παρατηρήθηκε από αυτήν. Η κλάση `Action` υποχρεώνει την υλοποίηση της μεθόδου `act(param:int):boolean` η οποία αναλαμβάνει την εκτέλεση της ενέργειας ανάλογα με την τιμή της μεταβλητής `param`.

Η `ServiceDesc` υλοποιεί την διεπαφή `ITransmittable` για την αποστολή των υπηρεσιών μέσω datagrams.

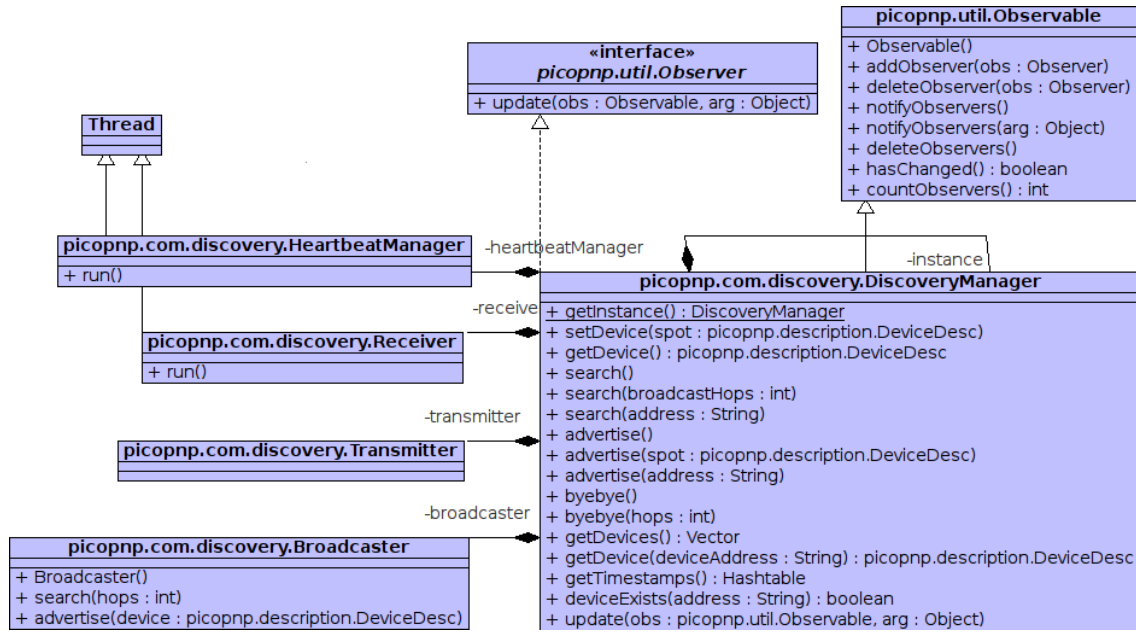


Σχήμα Α'.4: UML Διάγραμμα κλάσεων υπηρεσίας.

Α'.2.3 Διαχειριστής Ανακάλυψης (Discovery Manager)

Στο Σχήμα Α'.5 παρουσιάζεται ο Διαχειριστής Ανακάλυψης. Η κλάση **DiscoveryManager** δημιουργεί ένα singleton αντικείμενο του διαχειριστή αυτού. Ο τελευταίος παρέχει τις βασικές μεθόδους, όπως την αναζήτηση και την προβολή συσκευών σε διάφορες παραλλαγές ανάλογα με το αν πρόκειται για broadcast ή unicast μετάδοση. Διαθέτει τον **Transmitter** για την αποστολή μηνυμάτων προς συγκεκριμένη διεύθυνση, τον **Receiver** ο οποίος εκτελείται σε ξεχωριστό Thread και λαμβάνει τα μηνύματα που αφορούν την ανακάλυψη συσκευών, τον **Broadcaster** οποίος αναλαμβάνει την εκπομπή των μηνυμάτων στο δίκτυο και τον **HeartbeatManager**, ο οποίος εκτελείται και αυτός σε ξεχωριστό Thread και είναι υπεύθυνος για την ανανέωση και τον έλεγχο των συσκευών, ώστε να θεωρούνται ενεργές.

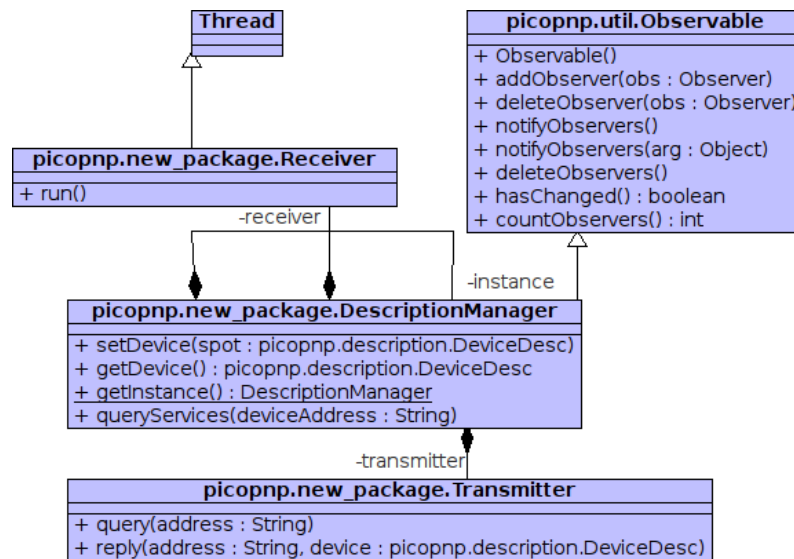
Ο Διαχειριστής Ανακάλυψης χρησιμοποιεί το port 100 για την επικοινωνία μεταξύ των κόμβων.



Σχήμα Α'.5: UML διάγραμμα κλάσεων για το Διαχειριστή Ανακάλυψης.

Α'.2.4 Διαχειριστής Περιγραφών (Description Manager)

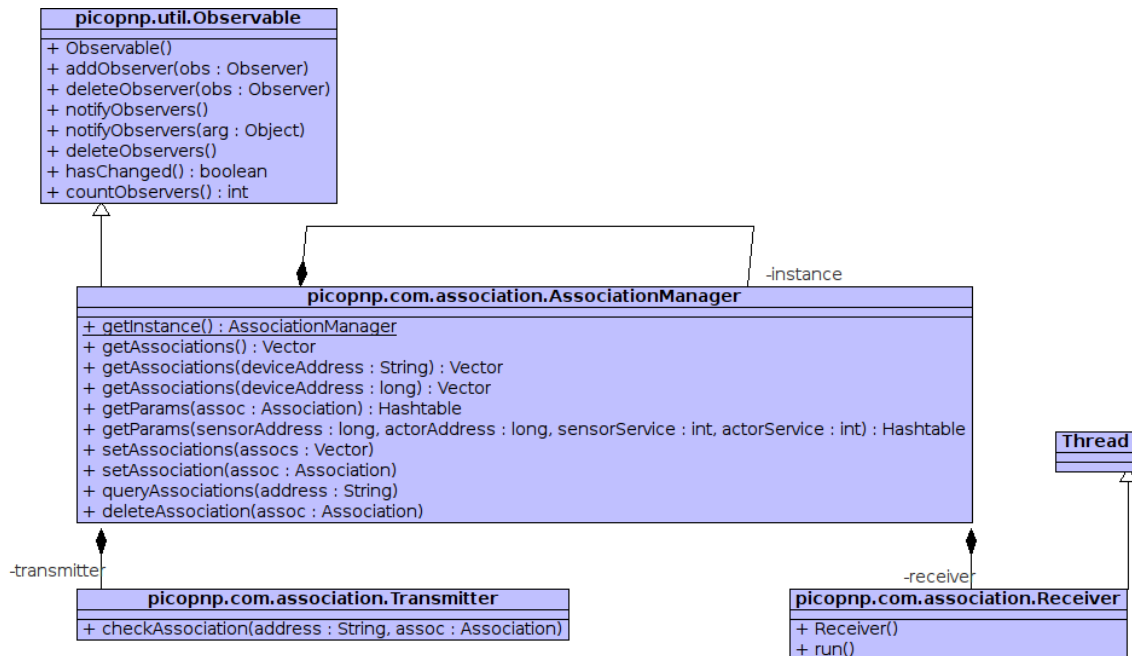
Στο Σχήμα Α'.6 παρουσιάζονται οι κλάσεις που αφορούν τον Διαχειριστή Περιγραφών. Ο Διαχειριστής Περιγραφών υλοποιείται από την κλάση **DescriptionManager** και διαθέτει έναν **Transmitter** και έναν **Receiver** για την αποστολή και λήψη των μηνυμάτων, αντίστοιχα. Το port που χρησιμοποιείται για την επικοινωνία είναι το 101.



Σχήμα Α'.6: UML διάγραμμα κλάσεων για το Διαχειριστή Περιγραφών.

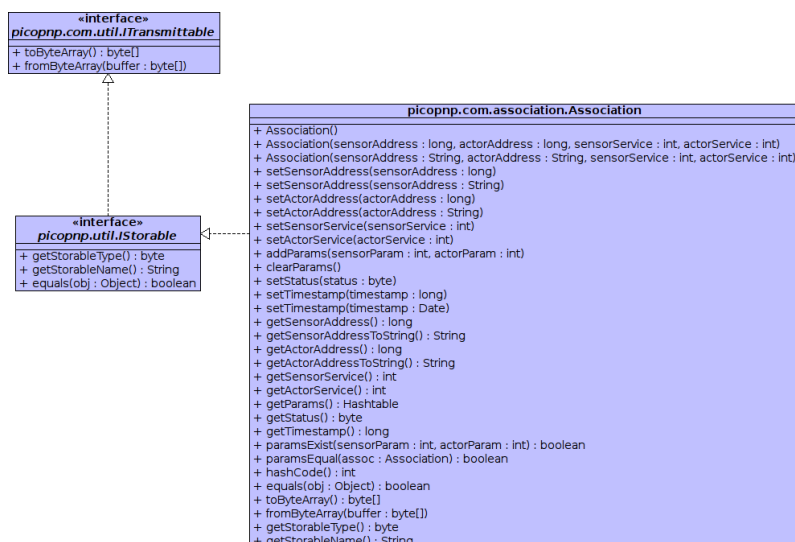
Α'.2.5 Διαχειριστής Συσχετίσεων (Association Manager)

Στο Σχήμα Α'.7 παρουσιάζονται οι κλάσεις του Διαχειριστή Συσχετίσεων. Ο Διαχειριστής Συσχετίσεων, που υλοποιείται από την κλάση **AssociationManager**, διαθέτει έναν **Transmitter** και έναν **Receiver** για την αποστολή των μηνυμάτων που αφορούν τις συσχετίσεις. Το port που χρησιμοποιείται για την επικοινωνία είναι το 102.



Σχήμα Α'.7: UML διάγραμμα κλάσεων για το Διαχειριστή Συσχετίσεων.

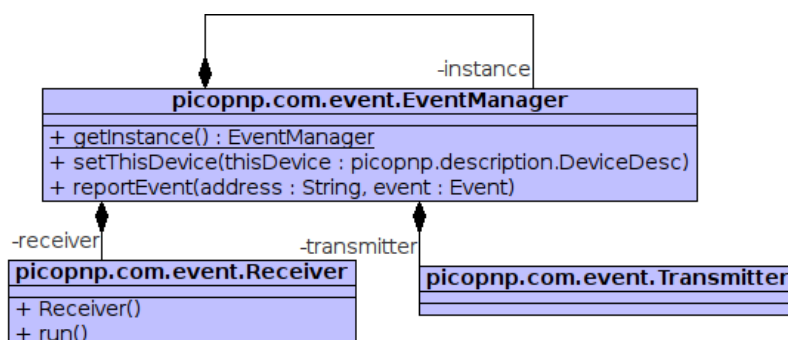
Στο Σχήμα 85 απεικονίζεται η κλάση **Association** που υλοποιεί την συσχέτιση μεταξύ ενός αισθητήρα και ενός ενεργοποιητή. Η κλάση αυτή υλοποιεί τη διεπαφή **IStorable**, η οποία επεκτείνει την **ITransmittable**, και χρησιμοποιείται για την αποθήκευση του αντικειμένου **Association** στη μνήμη flash των Sun SPOTs.



Σχήμα Α'.8: UML διάγραμμα για την κλάση Association.

Α'.2.6 Διαχειριστής Συμβάντων

Στο Σχήμα 85 παρουσιάζεται το διάγραμμα κλάσεων του Διαχειριστή Συμβάντων, **EventManager**, που είναι υπεύθυνος για την ειδοποίηση των ενεργοποιητών όταν παρατηρηθεί κάποιο συμβάν από τους αισθητήρες. Διαθέτει έναν **Transmitter** και έναν **Receiver** για την αποστολή των μηνυμάτων που αφορούν τα συμβάντα. Το port που χρησιμοποιείται για την επικοινωνία είναι το 103.



Σχήμα Α'.9: UML διάγραμμα κλάσεων για το Διαχειριστή Συμβάντων.

Στο Σχήμα 86 παρουσιάζεται η κλάση **Event** που χρησιμοποιείται για την ειδοποίηση ενός ενεργοποιητή από έναν αισθητήρα όταν ο τελευταίος αντιληφθεί ένα συμβάν που ταιριάζει με κάποια από τις συσχετίσεις του.



Σχήμα Α'.10: UML διάγραμμα για την κλάση *Event*.

Α'.3 Παραδείγματα υλοποίησης συσκευών και υπηρεσιών

Α'.3.1 Αισθητήρας

Το ακόλουθο πρόγραμμα υλοποιεί έναν αισθητήρα που παρακολουθεί την κατάσταση των πλήκτρων του Sun SPOT.

```

package picopnp.spot.applications.buttonsensor;

import picopnp.spot.device.Sensor;

/**
 * Η κλάση αυτή υλοποιεί ένα αισθητήρα ο οποίος παρακολουθεί
 * την κατάσταση των δύο buttons που παρέχει το SPOT
 */
// Η κλάση ButtonSensor κληρονομεί την Abstract κλάση Sensor
public class ButtonSensor extends Sensor {
    // Υλοποιεί την abstract μέθοδο της Sensor
    protected void startDevice() {
        // Καθορισμός των στοιχείων για την περιγραφή της συσκευής
        // το όνομα του αισθητήρα
        getSPOTDevice().setName("Button Sensor");
        // ο χρόνος ζωής τίθεται στα 5 λεπτά
        getSPOTDevice().setMax_age(300000);
        // το πλήθος των hops για την εκπομπή της περιγραφής
        getSPOTDevice().setComm_hops(5);

        // Προσθήκη γρήγορης στον αισθητήρα
        getSPOTDevice().addService(new ButtonListener());
    }
}
  
```

Ο κώδικας που ακολουθεί υλοποιεί την υπηρεσία για τον παραπάνω αισθητήρα.


```
package picopnp.spot.applications.multisensor;
import com.sun.spot.sensorboard.EDemoBoard;
import com.sun.spot.sensorboard.peripheral.ISwitch;
import com.sun.spot.sensorboard.peripheral.ISwitchListener;
import picopnp.spot.service.Sense;

/**
 * H klash ayth ylopoiei thn yphresia toy ais8hthra poy parakoloy8ei
 * thn katastash twn buttons toy SPOT. Ylopoiei to interface
 * poy parexetai apo to SPOT SDK ISwitchListener to opoio parakolou8ei
 * ta buttons
 */
public class ButtonListener extends Sense implements ISwitchListener {

    // oi parexomenes parametroi
    private final static int LEFT_PRESSED = 0;
    private final static int RIGHT_PRESSED = 1;
    private final static int LEFT_RELEASED = 2;
    private final static int RIGHT_RELEASED = 3;
    // ta buttons
    private ISwitch sw1;
    private ISwitch sw2;

    /**
     * Constructor 8etei ta stoixeia ths yphresias
     */
    public ButtonListener() {
        // ta buttons toy spot
        sw1 = EDemoBoard.getInstance().getSwitches()[EDemoBoard.SW1];
        sw2 = EDemoBoard.getInstance().getSwitches()[EDemoBoard.SW2];

        // Ka8orismos ths perigrafhs ths yphresias
        // to id
        setId(0);
        // to onoma
        setName("Button Listener");
        // oi plhrofories sxetika me ayth
        setInfo("Informs when a button on the
            SPOT is pressed or released.");

        // pros8hkh twn parexomenwn parametrwvn apo thn
        // yphresia
        addParam(LEFT_PRESSED, "Left pressed");
        addParam(RIGHT_PRESSED, "Right pressed");
        addParam(LEFT_RELEASED, "Left released");
        addParam(RIGHT_RELEASED, "Right released");

        // 8ese ton switch listener
        sw1.addISwitchListener(this);
        sw2.addISwitchListener(this);
    }

    /**
     * Ylopoiei me8odo tou ISwitchLister.
     * Ekteleitai otan piestei kapoio
     * apo ta buttons tou SPOT.
     */
    public void switchPressed(ISwitch sw) {
        if(sw == sw1)
            // an htan to aristero button
            // eidopoihse tou actors gia
            // ayto
            notifyActors(LEFT_PRESSED);
        else if(sw == sw2){
```

```

        // alliws eidopoihse toys actors
        // gia to de3i button
        notifyActors(RIGHT_PRESSED);
    }
}

/**
 * Ylopoiei me8odo tou ISwitchLister.
 * Ekteleitai otan kapoio apo ta buttons
 * ginei released
 */
public void switchReleased(ISwitch sw) {
    if(sw == sw1)
        // an htan to aristero button
        // eidopoihse tou actors gia
        // ayto
        notifyActors(LEFT_RELEASED);
    else if(sw == sw2){
        // alliws eidopoihse toys actors
        // gia to de3i button
        notifyActors(RIGHT_RELEASED);
    }
}

/**
 * Ylopoiei thn abstract me8odo ths Sense
 */
public void start() {
    // den yparxei kati na arxikopoih8ei
}
}

```

A'.3.2 Ενεργοποιητής

Το πρόγραμμα που ακολουθεί υλοποιεί έναν ενεργοποιητή ο οποίος ανάβει το LED 1 του Sun SPOT σε κάποιο από τρία διαθέσιμα χρώματα ή το σβήνει.

```

package picopnp.spot.applications.ledsensor;

import picopnp.spot.device.Actor;

/**
 * H klash ayth ylopoiei enan energopoihth o opoios
 * elegxei th leitourgia tou LED 1 tou spot. Mporei
 * na to anapsei se kokkino, prasino h mple h na to
 * sbhsei.
 */
// H klash LEDActor klhronomei thn Abstract klassh Actor
public class LEDActor extends Actor {
    // Ylopoiei thn abstract me8odo ths Actor
    protected void startDevice() {
        // Ka8orismos twn stoxeiwn gia thn perigrafh ths syskeyhs
        // to onoma toy ais8hthra
        getSPOTDevice().setName("LED Actor");
        // o xronos zwhs ti8etai sto 1 lepto
        getSPOTDevice().setMax_age(60000);
        // to plh8os twn hops gia thn ekpomph ths perigraphs
        getSPOTDevice().setComm_hops(7);

        // Pros8hkh yphresias ston ais8hthra
        getSPOTDevice().addService(new LED1Action());
    }
}

```

```
}
}
```

Ο ακόλουθος κώδικας υλοποιεί την υπηρεσία για τον έλεγχο του LED 1 του παραπάνω ενεργοποιητή.

```
package picopnp.spot.applications.ledactor;

import com.sun.spot.sensorboard.EDemoBoard;
import com.sun.spot.sensorboard.peripheral.ITriColorLED;
import com.sun.spot.sensorboard.peripheral.LEDColor;
import picopnp.spot.service.Action;

/**
 * Η κλάση αυτή υλοποιεί την υπηρεσία του ενεργοποιητή
 * που ελέγχει τη λειτουργία του LED1
 */
public class LED1Action extends Action {
    // to LED1 από το demo board
    private ITriColorLED led = EDemoBoard.getInstance().getLEDs()[0];

    /**
     * Constructor δίνει τα στοιχεία της υπηρεσίας
     */
    public LED1Action() {
        // το id της υπηρεσίας
        setId(0);
        // το όνομα της υπηρεσίας
        setName("LED1 Action");
        // πληροφορίες για την υπηρεσία
        setInfo("Controls the LED1 of the SPOT.");

        // προσθήκη των παρεχόμενων παραμέτρων
        addParam(0, "BLUE");
        addParam(1, "RED");
        addParam(2, "GREEN");
        addParam(3, "OFF");
    }

    /**
     * Υλοποιεί την abstract μέθοδο της κλάσης Action
     * η οποία εκτελεί την υπηρεσία.
     */
    public boolean act(int param) {
        // ελέγξε την παραμέτρο και ενεργήσε ανάλογα
        switch(param){
            case 0:
                led.setOff();
                led.setColor(LEDColor.BLUE);
                led.setOn();
                break;
            case 1:
                led.setOff();
                led.setColor(LEDColor.RED);
                led.setOn();
                break;
            case 2:
                led.setOff();
                led.setColor(LEDColor.GREEN);
                led.setOn();
                break;
            case 3:
                led.setOff();
                break;
        }
    }
}
```

```
        default:  
            return false;  
    }  
    return true;  
}  
}
```