



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Μελέτη και υλοποίηση συστημάτων ενδιάμεσου
λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless
Sensor Networks-WSNs)**

Ζαφειρία Ι. Διαμάντη

**Επιβλέποντες : Ευστάθιος Χατζηευθυμιάδης, Επίκουρος καθηγητής ΕΚΠΑ
Γεώργιος Αλυφαντής, Υποψήφιος Διδάκτωρ ΕΚΠΑ**

**ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2006**



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

**Μελέτη και υλοποίηση συστημάτων ενδιάμεσου
λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless
Sensor Networks-WSNs)**

Ζαφειρία Ι. Διαμάντη

**Επιβλέποντες : Ευστάθιος Χατζηευθυμιάδης, Επίκουρος καθηγητής ΕΚΠΑ
Γεώργιος Αλυφαντής, Υποψήφιος Διδάκτωρ ΕΚΠΑ**

**ΑΘΗΝΑ
ΙΟΥΛΙΟΣ 2006**

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks-WSNs)

Ζαφειρία Ι. Διαμάντη

A.M.: 018200000013

ΕΠΙΒΛΕΠΟΝΤΕΣ :

Ευστάθιος Χατζηευθυμιάδης, Επίκουρος καθηγητής ΕΚΠΑ

Γεώργιος Αλυφαντής, Υποψήφιος Διδάκτωρ ΕΚΠΑ

ΠΕΡΙΛΗΨΗ

Στην παρούσα πτυχιακή εργασία παρατίθενται πληροφορίες, σχετικά με τις εφαρμογές των ασύρματων δικτύων αισθητήρων, δίνοντας έτσι μια εικόνα των μελλοντικών στόχων της επιστημονικής κοινότητας, αναφορικά με τη νέα αυτή τεχνολογία, που αναμένεται να χρησιμοποιηθεί ευρέως στα επόμενα χρόνια.

Στη συνέχεια, παρουσιάζονται οι παράγοντες που αφορούν στη σχεδίαση ενός ασύρματου δικτύου αισθητήρων, ενώ περιγράφονται οι υπάρχουσες τεχνολογίες υλικού-λογισμικού(middleware) και ταυτόχρονα γίνεται μια αναφορά στα υπάρχοντα συστήματα ενδιάμεσου λογισμικού.

Ακολούθως, αναπτύσσεται μια πλατφόρμα μέρος της οποίας αποτελεί και ο RR Proxy. Η υλοποίηση του RR Proxy είναι το αντικείμενο της τρέχουσας πτυχιακής εργασίας, όπου και παρουσιάζεται αναλυτικά στη συνέχεια.

Τελειώνοντας, Θα ήθελα να ευχαριστήσω τον επιβλέποντα της εργασίας, Επίκουρο Καθηγητή Ευστάθιο Χατζηευθυμιάδη, μέλος του Εργαστηρίου Δικτύων Επικοινωνιών (CNL - Communication Networks Laboratory), του Τμήματος Πληροφορικής και Τηλεπικοινωνιών για τη δυνατότητα που μου έδωσε προκειμένου να ασχοληθώ με το συγκεκριμένο θέμα. Επίσης ευχαριστώ και τον Ερευνητή υποψήφιο Διδάκτορα Αλυφαντή Γεώργιο του οποίου η συμβολή ήταν καθοριστική για την ολοκλήρωση της εργασίας, όπως επίσης για την ανοχή και υπομονή που έδειξε, καθ' όλη τη διάρκεια της.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

• 1) ΕΙΣΑΓΩΓΗ	7
➤ 1.1) ΣΤΟΧΟΣ.....	8
➤ 1.2) ΟΡΓΑΝΩΣΗ ΤΗΣ ΠΤΥΧΙΑΚΗΣ ΕΡΓΑΣΙΑΣ	9
• 2) ΕΦΑΡΜΟΓΕΣ	11
➤ 2.1) ΣΤΡΑΤΙΩΤΙΚΕΣ	12
➤ 2.2) ΠΕΡΙΒΑΛΛΟΝΤΟΛΟΓΙΚΕΣ	14
➤ 2.3) ΣΤΟΝ ΚΛΑΔΟ ΥΓΕΙΑΣ	16
➤ 2.4) ΟΙΚΙΑΚΕΣ ΕΦΑΡΜΟΓΕΣ – ΟΙΚΙΑΚΟΣ ΑΥΤΟΜΑΤΙΣΜΟΣ	17
• 3) ΠΑΡΑΓΟΝΤΕΣ ΠΟΥ ΣΥΜΒΑΛΟΥΝ ΣΤΟ ΣΧΕΔΙΑΣΜΟ ΤΩΝ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ.....	20
➤ 3.1) ΑΝΕΚΤΙΚΟΤΗΤΑ ΣΕ ΣΦΑΛΜΑΤΑ	21
➤ 3.2) ΔΥΝΑΤΟΤΗΤΑ ΚΛΙΜΑΚΩΣΗΣ	21
➤ 3.3) ΚΟΣΤΟΣ ΠΑΡΑΓΩΓΗΣ	21
➤ 3.4) ΠΕΡΙΟΡΙΣΜΟΙ ΥΛΙΚΟΥ	22
➤ 3.5) ΤΟΠΟΛΟΓΙΑ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ	24
➤ 3.6) ΠΕΡΙΒΑΛΛΟΝ	26
➤ 3.7) ΜΕΣΑ ΜΕΤΑΔΟΣΗΣ	26
➤ 3.8) ΚΑΤΑΝΑΛΩΣΗ ΕΝΕΡΓΕΙΑΣ	28

- **4) ΥΠΑΡΧΟΥΣΕΣ ΤΕΧΝΟΛΟΓΙΕΣ MIDDLEWARE29**
 - 4.1) ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΕΝΔΙΑΜΕΣΟΥ ΛΟΓΙΣΜΙΚΟΥ (MIDDLEWARE)30
 - 4.2) ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΕΦΑΡΜΟΓΩΝ.33
 - 4.3) ΑΡΧΕΣ ΣΧΕΔΙΑΣΗΣ ΤΟΥ MIDDLEWARE.34
 - 4.4) ΚΑΤΗΓΟΡΙΕΣ MIDDLEWARE.36
 - 4.5) ΠΡΟΣΕΓΓΙΣΗ ΚΙΝΗΤΩΝ ΠΡΑΚΤΟΡΩΝ(MOBILE AGENTS).37
 - 4.5.1 SENSOR WARE37
 - 4.6) ΠΡΟΣΕΓΓΙΣΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ.38
 - 4.6.1 TinyDB39
 - 4.6.2 SINA.....39
 - 4.7) ΠΡΟΣΕΓΓΙΣΗ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ ΕΝΝΟΙΑ ΤΩΝ ΣΥΜΒΑΝΤΩΝ (EVENTS).40

- **5) ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ ΕΝΔΙΑΜΕΣΟΥ ΕΠΙΠΕΔΟΥ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΑΝΑΠΤΥΞΗΣ41**
 - 5.1) ΓΕΝΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ42
 - 5.2) Η UNIFIED SENSOR LANGUAGE (USL).....44
 - 5.3) Η ΑΙΤΗΣΗ (USL-REQUEST)46
 - 5.4) Η ΑΠΑΝΤΗΣΗ (USL-RESPONSE).....48
 - 5.5) ΤΟ SENSOR ABSTRACTION LAYER (SAL).....50
 - 5.6) RR Proxy (REQUEST-RESPONSE PROXY).....50
 - 5.7) WSN Driver.....51

• 6) ΑΝΑΛΥΣΗ ΥΛΟΠΟΙΗΣΗΣ	55
➤ 6.1) ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ	55
➤ 6.2) RequestPreprocessing	58
➤ 6.3) TCP_Select_Socket.....	59
➤ 6.4) WSN_Driver	59
➤ 6.5) Dispatcher.....	59
➤ 6.6) ResponseCreator	60
➤ 6.7) Αναλυτική Περιγραφή της Προώθησης και Επεξεργασίας ενός αριθμού αιτήσεων Requests.....	64
➤ 6.8) Ένα απλό παράδειγμα εφαρμογής του προγράμματος.....	68
• 7) ΣΥΝΟΨΗ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΞΕΛΙΞΗ	75
• 8) ΠΑΡΑΡΤΗΜΑ Α -ΛΕΠΤΟΜΕΡΕΙΕΣ ΕΚΤΕΛΕΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ	76
• 9) ΒΙΒΛΙΟΓΡΑΦΙΑ	78

ΚΕΦΑΛΑΙΟ 1

ΕΙΣΑΓΩΓΗ

Η πρόσφατη πρόοδος στην τεχνολογία των μικρό-ηλεκτρομηχανικών συστημάτων (micro-electro-mechanical systems - MEMS), των ασύρματων επικοινωνιών και των ψηφιακών ηλεκτρονικών μέσων, επέτρεψε την έρευνα και ανάπτυξη χαμηλού κόστους και κατανάλωσης ενέργειας πολύ-λειτουργικών κόμβων αισθητήρων, που είναι μικροί σε μέγεθος και μπορούν να επικοινωνούν σε κοντινές μεταξύ τους αποστάσεις βασιζόμενοι σε μια ad-hoc συμπεριφορά. Τέτοιου είδους κόμβοι χαρακτηρίζονται γενικά, από περιορισμένες υπολογιστικές και επικοινωνιακές δυνατότητες. Ένα σύνολο τέτοιων κόμβων-αισθητήρων που επικοινωνούν μεταξύ τους ασύρματα, συνιστούν ένα ασύρματο δίκτυο αισθητήρων (Wireless Sensor Network - WSN). Οι πιο εξελιγμένες εφαρμογές των δικτύων αυτών, αφορούν εφαρμογή σε εξειδικευμένες περιπτώσεις, όπως στην παρακολούθηση του περιβάλλοντος (στο habitat monitoring), στην παρακολούθηση της κυκλοφορίας, σε στρατιωτικά συστήματα, στο Supply Chain management, σε συστήματα υγείας και άλλα.

Ο προγραμματισμός τέτοιων ασύρματων δικτύων αισθητήρων, υλοποιείται συνήθως με τη χρήση ενός ενδιάμεσου λογισμικού (Middleware), το οποίο συνεργάζεται, κατά κανόνα, μόνο με συγκεκριμένες και καθορισμένες δομές υλικού (Hardware) για WSN. Αυτού του είδους το λογισμικό, εν αντιθέσει με τη συνήθη ερμηνεία του όρου, δεν αποσκοπεί στο να παρέχει μια γενική και υψηλού επιπέδου προγραμματιστική πλατφόρμα για ασύρματα δίκτυα αισθητήρων, αλλά ένα βασικό σύνολο εργαλείων και βιβλιοθηκών για τον χαμηλού επιπέδου χειρισμό των κόμβων αισθητήρων. Ο προγραμματισμός και η ρύθμιση του δικτύου αισθητήρων είναι εργασία δύσκολη και ευάλωτη σε λάθη, αφού αυτός που αναπτύσσει την εφαρμογή πρέπει στις περισσότερες περιπτώσεις να προγραμματίσει ξεχωριστούς κόμβους αισθητήρων, χρησιμοποιώντας χαμηλού επιπέδου προγραμματιστικές γλώσσες .

Όσο τα ασύρματα δίκτυα αισθητήρων χρησιμοποιούνται σε συγκεκριμένες εφαρμογές, αυτή η προγραμματιστική προσέγγιση είναι ευρέως αποδεκτή. Παρόλα αυτά,

προκειμένου να ενσωματωθούν οι υπάρχουσες WSN εφαρμογές σε άλλα υπολογιστικά σχήματα, κάποιες βελτιώσεις στο είδη υπάρχον ενδιάμεσο λογισμικό, είναι απαραίτητες. Στις μέρες μας ένα από τα πιο υποσχόμενα υπολογιστικά παραδείγματα είναι ο «διάχυτος υπολογισμός» (pervasive computing) ο οποίος ενσωματώνει έξυπνα δικτυακά περιβάλλοντα, στα οποία οι χρήστες μπορούν αδιάκοπα και προδραστικά να χρησιμοποιούν υπηρεσίες περιεχομένου. Προκειμένου να φέρουμε το «διάχυτο υπολογισμό» ένα βήμα πιο κοντά προς την υλοποίησή του, οι «διάχυτες εφαρμογές» θα πρέπει να μπορούν αδιάκοπα να έχουν πρόσβαση σε πληροφορίες πλαισίου (context), οι οποίες πηγάζουν από δομές αισθητήρων που βρίσκονται σε χαμηλότερο επίπεδο. Επιπρόσθετα, προκύπτει πως οι εφαρμογές δεν είναι απαραίτητο να «γνωρίζουν» τις διαφορετικές τεχνολογίες ενδιάμεσου λογισμικού που χρησιμοποιούνται, καθώς και τα χαρακτηριστικά τους. Επίσης, προκειμένου να γίνει ο προγραμματισμός, σε αυτά τα περιβάλλοντα, φιλικός ως προς τον προγραμματιστή, είναι απαραίτητο να χρησιμοποιηθούν κάποιες προγραμματιστικές αφαιρέσεις (abstractions) που απλοποιούν τη προγραμματιστική διαδικασία ενώ πρέπει παράλληλα το ενδιάμεσο λογισμικό να υποστηρίζει αυτές τις αφαιρέσεις.

1.1 Στόχος

Σύμφωνα με όσα προηγήθηκαν, γίνεται φανερό πως τα ασύρματα δίκτυα αισθητήρων, προκειμένου να υποστηρίξουν πλήρως μια εφαρμογή η οποία στηρίζεται σε μια μεγάλη ποικιλία χρηστών, διαφορετικών αναγκών, θα πρέπει ο σχεδιασμός τους να βασιστεί στις ακόλουθες αρχές:

1. Θα πρέπει να μην αποκαλύπτει τη διαφορετικότητα που προκύπτει από την επικοινωνία διαφορετικών δομών δικτύων αισθητήρων. Η διαφορετικότητα αυτών των δομών έγκειται στις τεχνολογίες των αισθητήρων του δικτύου και του ενδιάμεσου λογισμικού (middleware) που χρησιμοποιούνται. Αυτό καθίσταται απαραίτητο, καθώς θεωρείται απίθανο μια συγκεκριμένη δομή να ικανοποιήσει το μεγαλύτερο μέρος του τομέα των δικτύων αισθητήρων για τα επόμενα χρόνια και να γίνει πρότυπο.

2. Πρέπει να παρέχει ένα κατάλληλο προγραμματιστικό μοντέλο (ένα φιλικό προς τον προγραμματιστή σύνολο εντολών, ευκολίες επεξεργασίας πληροφορίας, καθώς και ρουτίνες επαναπροσδιορισμού και ελέγχου) που θα προωθήσει ακόμα περισσότερο την ανάπτυξη διάχυτων εφαρμογών. Πιστεύεται πως αυτές οι εφαρμογές θα αξιοποιήσουν πλήρως τις δυνατότητες των WSN.
3. Τέλος πρέπει να λαμβάνει υπόψη τις βασικές αρχές των ασύρματων δικτύων αισθητήρων (εξοικονόμηση ενέργειας, επεκτασιμότητα και ευρωστία).

Το σχήμα που αναπτύσσουμε βασίζεται ακριβώς σε αυτές τις αρχές, καθώς και στο ODBC/JDBC[2] πλαίσιο που χρησιμοποιείται ευρέως στον χώρο των βάσεων δεδομένων. Το δίκτυο αισθητήρων σαν σύνολο, αντιμετωπίζεται σαν μια πηγή πληροφοριών, ακριβώς όπως μια βάση δεδομένων στο κόσμο του ODBC/JDBC[2].

1.2 Οργάνωση της Πτυχιακής εργασίας

Στη δεύτερη ενότητα του τόμου αυτού παρατίθενται πληροφορίες, σχετικά με τις εφαρμογές των ασύρματων δικτύων αισθητήρων, δίνοντας έτσι μια εικόνα των μελλοντικών στόχων της επιστημονικής κοινότητας, αναφορικά με τη νέα αυτή τεχνολογία, που αναμένεται να χρησιμοποιηθεί ευρέως στα επόμενα χρόνια.

Στην Τρίτη ενότητα γίνεται αναφορά στους παράγοντες που αφορούν στη σχεδίαση ενός ασύρματου δικτύου αισθητήρων, ενώ στην τέταρτη περιγράφονται οι υπάρχοντες τεχνολογίες υλικού-λογισμικού(middleware) και ταυτόχρονα γίνεται μια αναφορά στα υπάρχοντα συστήματα ενδιάμεσου λογισμικού.

Η πέμπτη ενότητα αφορά την πλατφόρμα που αναπτύσσεται μέρος της οποίας αποτελεί και ο RR Proxy. Η υλοποίηση του RR Proxy είναι το αντικείμενο της τρέχουσας πτυχιακής εργασίας και περιγράφεται αναλυτικά στην έκτη ενότητα .

Το κομμάτι της πλατφόρμας που ονομάζεται RR Proxy οφείλει την επωνυμία του στο είδος των λειτουργιών που εκτελεί. Αναλυτικότερα, RR σημαίνει Request – Response και εκφράζει το σύνολο των αιτήσεων που επεξεργάζεται η συγκεκριμένη εφαρμογή. Οι

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

αιτήσεις έχουν χαρακτήρα Ερώτησης (Request) που ο χρήστης(API) στέλνει στην πλατφόρμα και αντιστοίχως Απάντησης (Response) από την πλατφόρμα ανάπτυξης στο χρήστη (API). Ακολούθως, με τον όρο Proxy, εκφράζεται η λειτουργικότητα του

ενδιάμεσου λογισμικού (middleware) που αναπτύσσεται στην παρούσα εργασία, το οποίο έχει αναλάβει το χειρισμό όλων των αιτήσεων που προωθούνται από το χρήστη προς τα ασύρματα δίκτυα αισθητήρων και αντίστροφα.

Στην έβδομη ενότητα παρουσιάζεται μια σύνοψη της εργασίας ,ενώ στο παράρτημα που ακολουθεί δίνεται μια σύντομη περιγραφή του τρεξίματος και της εξόδου του προγράμματος.

ΚΕΦΑΛΑΙΟ 2

ΕΦΑΡΜΟΓΕΣ ΤΩΝ ΑΣΥΡΜΑΤΩΝ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

Τα δίκτυα αισθητήρων μπορούν να αποτελούνται από πολλούς διαφορετικούς τύπους αισθητήρων όπως σεισμικούς, μαγνητικούς χαμηλού ρυθμού δειγματοληψίας, θερμικούς, οπτικούς, υπέρυθρους, ακουστικούς και ραντάρ, οι οποίοι είναι ικανοί να παρακολουθούν μια ευρεία ποικιλομορφία περιβαλλοντολογικών μεγεθών που περιλαμβάνουν τα ακόλουθα [16] [2]:

- Θερμοκρασία
- Υγρασία
- Κίνηση οχημάτων
- Συνθήκες φωτός.
- Πίεση.
- Διάρθρωση εδάφους.
- Επίπεδα θορύβου.
- Την παρουσία ή απουσία προκαθορισμένων ειδών αντικειμένων.
- Επίπεδα μηχανικής πίεσης σε προσκολλημένα αντικείμενα και
- Ταχύτητα, κατεύθυνση και μέγεθος ενός αντικειμένου.

Οι κόμβοι των ασύρματων δικτύων αισθητήρων μπορούν να χρησιμοποιηθούν για συνεχή ανίχνευση, ανίχνευση συμβάντων, ανίχνευση ταυτοτήτων γεγονότων, ανίχνευση θέσης και τοπικό έλεγχο μηχανισμών κίνησης. Η ιδέα της μικρό-ανίχνευσης και της ασύρματης σύνδεσης αυτών των κόμβων υπόσχεται πολλές νέες περιοχές εφαρμογών. Οι εφαρμογές των δικτύων αισθητήρων μπορούν να ομαδοποιηθούν σε στρατιωτικές, υγείας, περιβάλλοντος, οικιακές και εμπορικές. Είναι δυνατόν να επεκτείνουμε την ομαδοποίηση με περισσότερες κατηγορίες όπως εξερεύνηση του διαστήματος, χημική επεξεργασία, αντιμετώπιση καταστροφών κ.α.

2.1 Στρατιωτικές Εφαρμογές

Τα WSN μπορούν να αποτελέσουν ένα αναπόσπαστο κομμάτι των ελέγχων, των επικοινωνιών, της παρακολούθησης, της νοημοσύνης και της στόχευσης για μια στρατιωτική επιχείρηση. Η πολύ γρήγορη ανάπτυξη, η αυτοοργάνωση, και η ανεκτικότητα σε λάθη καθιστούν τα WSNs μια πολλά υποσχόμενη λύση για τις στρατιωτικές επιχειρήσεις του μέλλοντος. Αφού τέτοια δίκτυα βασίζονται στον πυκνό διασκορπισμό χαμηλού κόστους κόμβων, η καταστροφή ορισμένων από αυτούς από εχθρικές ενέργειες δεν επηρεάζει την στρατιωτική επιχείρηση όσο η καταστροφή ενός παραδοσιακού αισθητήρα, και κάνει τέτοια δίκτυα πιο εύχρηστα σε τέτοιες επιχειρήσεις. Μερικές από τις στρατιωτικές εφαρμογές αυτών των δικτύων, είναι η παρακολούθηση συμμαχικών δυνάμεων, εξοπλισμών και πυρομαχικών, η παρακολούθηση του πεδίου μάχης και ο συμπερασμός εχθρικών ενεργειών, η ανίχνευση στόχων, η καταγραφή ζημιών και τέλος η ανίχνευση πυρηνικής ή βιολογικής και χημικής επίθεσης.

Παρακολούθηση συμμαχικών δυνάμεων, εξοπλισμών και πυρομαχικών

Οι αρχηγοί και αυτοί που δίνουν εντολές, μπορούν κάθε στιγμή να παρακολουθούν τις κινήσεις φιλικών προς αυτούς στρατευμάτων, τη κατάσταση και την διαθεσιμότητα του εξοπλισμού και των πυρομαχικών σε ένα πεδίο μάχης, με τη χρήση δικτύων αισθητήρων. Κάθε στρατός, όχημα, εξοπλισμός καθώς και τα κρίσιμα πυρομαχικά μπορεί να φέρει μικρούς αισθητήρες που δίνουν αναφορά για την κατάστασή του. Αυτές οι αναφορές μαζεύονται σε κεντρικούς κόμβους και στέλνονται στους αρχηγούς του εκάστοτε στρατού. Τα δεδομένα μπορούν επίσης να προωθηθούν σε ανώτερα επίπεδα της στρατιωτικής ιεραρχίας ενώ συγκρίνονται με δεδομένα από άλλες μονάδες.

Παρακολούθηση του πεδίου μάχης

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

Τα κρίσιμα πεδία, μονοπάτια και στενά, μπορούν πολύ γρήγορα να καλυφθούν από αισθητήρες, ώστε να παρακολουθούνται στενά οι ενέργειες των αντίπαλων στρατευμάτων. Όσο οι επιχειρήσεις εξελίσσονται και ετοιμάζονται καινούργια

στρατηγικά σχέδια, καινούργιοι κόμβοι μπορούν να διασκορπιστούν για να ενισχύσουν τη παρακολούθηση.

Έλεγχος επίθεσης από αντίπαλα στρατεύματα

Τα δίκτυα αισθητήρων μπορούν να χρησιμοποιηθούν σε κρίσιμα πεδία με αποτέλεσμα να μπορούν να εξαχθούν πολύ χρήσιμες πληροφορίες σχετικά με το που, πως και πότε επιτίθενται τα αντίπαλα στρατεύματα, ώστε να μην υπάρξει περίπτωση αιφνιδιασμού.

Εύρεση στόχου

Με ειδική επεξεργασία των πληροφοριών που συλλέγονται από αυτά τα δίκτυα, μπορούν να εξαχθούν συμπεράσματα για την εύρεση του σωστού στόχου.

Καταγραφή ζημιών

Πριν και μετά από μια επίθεση, μπορούν να χρησιμοποιηθούν τα δίκτυα αισθητήρων για να μας δώσουν ποιοτικά και ποσοτικά συμπεράσματα αναφορικά με το μέγεθος της ζημιάς.

Διαπίστωση πυρηνικής ή βιολογικής ή χημικής επίθεσης

Σε περίπτωση χημικού ή βιολογικού πολέμου, η χρήση δικτύων αισθητήρων θα ήταν πολύ σημαντική για την γρήγορη ενημέρωση σχετικά με μια τέτοια επίθεση. Επίσης θα ενημέρωνε για τα στάδια της μόλυνσης σε διάφορες περιοχές και θα συνέβαλλε στην μείωση των θυμάτων τέτοιων επιθέσεων δραστικά. Επιπλέον δεν θα χρειαζόταν η

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)
αποστολή ειδικής ομάδας σε σημεία μολυσμένα από ραδιενέργεια για να κάνουν μετρήσεις, αφού αυτές θα μπορούν να γίνονται αυτόματα.

2.2 Περιβαλλοντολογικές Εφαρμογές

Μερικές περιβαλλοντολογικές εφαρμογές των ασύρματων δικτύων αισθητήρων περιλαμβάνουν την καταγραφή των μετακινήσεων πουλιών, μικρών ζώων και εντόμων, την παρακολούθηση περιβαλλοντολογικών συνθηκών που επηρεάζουν τη χλωρίδα και την πανίδα, τους υδροφόρους ορίζοντες, την ανίχνευση βιολογικών και χημικών στοιχείων, την ανίχνευση πυρκαγιών σε δασικές εκτάσεις, μετεωρολογικές ή γεωφυσικές έρευνες, την ανίχνευση πλημμύρων, τη χαρτογράφηση της βιοποικιλότητας του περιβάλλοντος καθώς και τη μελέτη της μόλυνσης του περιβάλλοντος.

Ανίχνευση πυρκαγιών

Από τη στιγμή που οι κόμβοι του δικτύου μπορούν να διασκορπιστούν τυχαία ή με βάση κάποια στρατηγική και με μεγάλη πυκνότητα μέσα σε κάποιο δάσος, οι αισθητήρες μπορούν να αποκαλύψουν την ακριβή εστία της φωτιάς πριν η διάδοσή της είναι ανεξέλεγκτη. Ένας μεγάλος αριθμός από αισθητήρες-κόμβοι μπορεί να χρησιμοποιηθεί μέσω ραδιοσυχνοτήτων / οπτικών συστημάτων. Επίσης, μπορούν να εξοπλιστούν με αποτελεσματικές μεθόδους για την εξοικονόμηση ενέργειας, όπως ηλιακές κυψέλες, επειδή οι αισθητήρες μπορεί να παραμείνουν χωρίς επιτήρηση για μήνες ή ακόμα και για χρόνια. Οι αισθητήρες θα αλληλεπιδρούν μεταξύ τους προκειμένου να πραγματοποιούν καταναεμημένες λειτουργίες αισθήσεων και προκειμένου να παρακάμψουν εμπόδια, όπως δέντρα και πέτρες που θα εμπόδιζαν την επικοινωνία μεταξύ τους.

Χαρτογράφηση της βιοποικιλότητας του περιβάλλοντος

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

Η χαρτογράφηση της βιοποικιλότητας του περιβάλλοντος προϋποθέτει ανεπτυγμένες προσεγγίσεις προκειμένου να συνδυάσουν τις πληροφορίες μεταξύ χωρικής και χρονικής κλίμακας. Η πρόοδος της τεχνολογίας στον τομέα της απομακρυσμένης αίσθησης και της αυτοματοποιημένης συλλογής δεδομένων έχει δώσει τη δυνατότητα για μεγαλύτερη χωρική και χρονική ανάλυση, με ένα γεωμετρικά φθίνον κόστος ανά

κόμβο. Παράλληλα με αυτή την τεχνολογική πρόοδο, οι αισθητήρες έχουν τη δυνατότητα να συνδέονται με το internet, το οποίο επιτρέπει σε απομακρυσμένους χρήστες να ελέγχουν, να παρακολουθούν και να παρατηρούν τη βιοποικιλότητα του περιβάλλοντος.

Παρόλο που οι δορυφορικοί και αεροπορικοί αισθητήρες είναι χρήσιμοι στην παρατήρηση μεγάλης κλίμακας βιοποικιλότητας, π.χ. χωρική πολυπλοκότητα κάποιων κυρίαρχων ειδών φυτών, δεν είναι αρκετά ικανοποιητικοί στην παρατήρηση μικρής κλίμακας βιοποικιλότητας που ουσιαστικά αποτελεί και το βασικό κομμάτι ενός οικοσυστήματος. Αυτό έχει ως αποτέλεσμα να υπάρχει ανάγκη για χρήση ασύρματου δικτύου αισθητήρων στην επιφάνεια του εδάφους για την πιο αποτελεσματική παρατήρηση της βιοποικιλότητας.

Ανίχνευση πλημμύρων

Ένα χαρακτηριστικό παράδειγμα ανίχνευσης πλημμύρων είναι το σύστημα ALERT [4] εφαρμόστηκε στις Ηνωμένες Πολιτείες της Αμερικής. Μερικοί τύποι αισθητήρων που χρησιμοποιήθηκαν στο σύστημα ALERT [4] είναι αισθητήρες βροχής, ύδατος και κλίματος. Αυτοί οι αισθητήρες παρέχουν πληροφορίες σε μία κεντρική βάση δεδομένων με έναν προδιαγεγραμμένο τρόπο. Μερικά ερευνητικά προγράμματα, όπως το COUGAR [5], εξετάζουν κατανεμημένες προσεγγίσεις αναφορικά με την αλληλεπίδραση με τους αισθητήρες σε ένα πεδίο αισθητήρων προκειμένου να παρέχουν άμεσες αλλά μακροχρόνιες επερωτήσεις.

Προβλέψεις χρήσιμες για τη γεωργία

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

Μερικά από τα πλεονεκτήματα των δικτύων αισθητήρων είναι η παρακολούθηση των επιπέδων του πόσιμου νερού, της διάβρωσης του εδάφους και της μόλυνσης του αέρα σε πραγματικό χρόνο.

2.3 Εφαρμογές στον κλάδο Υγείας

Μερικές από τις εφαρμογές των δικτύων αισθητήρων στο κλάδο της υγείας είναι η παροχή διεπαφών προς τους ανάπηρους, η συνεχής παρακολούθηση ασθενών, η διάγνωση, η σύσταση συγκεκριμένων φαρμάκων στα νοσοκομεία, η παρακολούθηση των κινήσεων και των εσωτερικών διεργασιών εντόμων και άλλων μικρών ζώων, η τηλεπαρακολούθηση δεδομένων της ανθρώπινης φυσιολογίας καθώς και ο εντοπισμός και η παρακολούθηση γιατρών και ασθενών σε ένα νοσοκομείο.

Τηλεπαρακολούθηση των δεδομένων της ανθρώπινης φυσιολογίας

Τα δεδομένα της φυσιολογίας που συλλέγονται από τους αισθητήρες μπορούν να αποθηκεύονται για ένα μεγάλο χρονικό διάστημα και να χρησιμοποιούνται για λόγους ιατρικής έρευνας. Οι αισθητήρες μπορούν επίσης να ανιχνεύουν τη συμπεριφορά ηλικιωμένων ανθρώπων, όπως για παράδειγμα κάποιο πέσιμο. Αυτοί οι μικροί κόμβοι δίνουν μια μεγάλη ελευθερία κινήσεων στους εξεταζόμενους ενώ παράλληλα δίνουν στους γιατρούς τη δυνατότητα να ανιχνεύουν προδιαγεγραμμένα για το άτομο συμπτώματα εγκαίρως. Επιπλέον εξασφαλίζουν μια υψηλότερης ποιότητας ζωή στον εξεταζόμενο, σε σύγκριση με την αντίστοιχη ενός κέντρου νοσηλείας.

Ανίχνευση και παρακολούθηση των κινήσεων των γιατρών και των ασθενών στο νοσοκομείο

Κάθε ασθενής έχει προσαρτημένους επάνω του μικρούς και ελαφρούς αισθητήρες. Κάθε αισθητήρας έχει το δικό του ξεχωριστό ρόλο. Για παράδειγμα ένας αισθητήρας

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) μπορεί να παρακολουθεί τους χτύπους της καρδιάς, ενώ κάποιος άλλος να ελέγχει την πίεση του αίματος. Οι γιατροί μπορούν επίσης να μεταφέρουν τέτοιους αισθητήρες, κάτι που επιτρέπει σε άλλους γιατρούς να τους εντοπίζουν μέσα στο νοσοκομείο.

Σύσταση συγκεκριμένων φαρμάκων στα νοσοκομεία

Αν ένας ασθενής φέρει κάποιους αισθητήρες, τότε είναι εύκολο να ανιχνευθούν οι αλλεργίες του και αντίστοιχα οι κατάλληλες φαρμακευτικές αγωγές για την αντιμετώπιση κάποιου προβλήματος υγείας, αποφεύγοντας έτσι την χορήγηση φαρμάκων στα οποία τελικά ο ασθενής είναι αλλεργικός.

2.4 Οικιακές Εφαρμογές

Οικιακός αυτοματισμός

Καθώς η τεχνολογία προοδεύει, έξυπνοι αισθητήρες μπορούν να τοποθετηθούν πάνω σε οικιακές συσκευές, όπως ηλεκτρικές σκούπες, φούρνοι μικροκυμάτων, ψυγεία κ.α. Αυτοί οι αισθητήρες μπορούν να επικοινωνούν μεταξύ τους, καθώς και με εξωτερικά δίκτυα μέσω Internet ή δορυφόρου. Έτσι ο χρήστης μπορεί πολύ εύκολα να χρησιμοποιεί και να ελέγχει αυτές τις συσκευές όντας απομακρυσμένος από αυτές.

Έξυπνο περιβάλλον

Ο σχεδιασμός του έξυπνου περιβάλλοντος μπορεί να έχει δύο διαφορετικές οπτικές. Την ανθρωποκεντρική και την τεχνοκεντρική. Στη περίπτωση της ανθρωποκεντρικής, ένα έξυπνο περιβάλλον πρέπει να προσαρμόζεται άριστα στις ανάγκες του χρήστη. Στη περίπτωση της τεχνοκεντρικής, πρέπει να αναπτυχθούν καινούργιες τεχνολογίες υλικού, δικτύων, καθώς και υπηρεσίες ενδιάμεσου λογισμικού. Ένα σενάριο του πως οι αισθητήρες μπορούν να χρησιμοποιηθούν για την ανάπτυξη ενός έξυπνου περιβάλλοντος είναι το εξής: Οι αισθητήρες εφάπτονται σε έπιπλα και συσκευές και

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) επικοινωνούν μεταξύ τους, καθώς και με έναν εξυπηρετητή (server) του δωματίου. Κάθε εξυπηρετητής δωματίου μπορεί να επικοινωνεί με άλλους παρόμοιους εξυπηρετητές και να μαθαίνει τι υπηρεσίες προσφέρουν (π.χ. scanning, printing και faxing).

Κάποιες επιπλέον εμπορικές εφαρμογές είναι η ανίχνευση αδυναμιών σε υλικά, η

δημιουργία εικονικών πληκτρολογίων, η κατηγοριοποίηση αγαθών, ο έλεγχος ποιότητας αγαθών, η κατασκευή έξυπνων χώρων σε γραφεία, ο έλεγχος του περιβάλλοντος σε κτίρια γραφείων, ο έλεγχος και η καθοδήγηση ρομπότ σε αυτοματοποιημένα κατασκευαστικά περιβάλλοντα, η κατασκευή διαδραστικών παιχνιδιών, η κατασκευή διαδραστικών μουσείων, ο έλεγχος διεργασιών στη βιομηχανία και η αυτοματοποίηση τους, η παρακολούθηση κατεστραμμένων περιοχών, η κατασκευή έξυπνων δομών με ενσωματωμένους αισθητήρες, η διάγνωση μηχανημάτων, η μετακίνηση, η ανίχνευση κλεμμένων αυτοκινήτων, ο εντοπισμός κινούμενων οχημάτων, καθώς και άλλες πολλές.

Έλεγχος περιβάλλοντος σε κτίρια γραφείων

Ο κλιματισμός και η θέρμανση στα περισσότερα κτίρια ελέγχεται από κάποιο κεντρικό σημείο. Συνεπώς η θερμοκρασία μέσα σε ένα δωμάτιο μπορεί να διαφέρει μερικούς βαθμούς. Για παράδειγμα αν υπάρχει μόνο ένα σώμα θέρμανσης και η κατανομή της θερμότητας δεν γίνεται ισομερώς, η μια πλευρά θα είναι πιο ζεστή από την άλλη. Ένα κατανεμημένο ασύρματο δίκτυο αισθητήρων θα μπορούσε να χρησιμοποιηθεί για να ελέγχει τη ροή του αέρα και της θερμοκρασίας σε διαφορετικά μέρη του δωματίου.

Κατασκευή διαδραστικών μουσείων

Μελλοντικά τα παιδιά θα μπορούν να αλληλεπιδρούν με αντικείμενα σε μουσεία προκειμένου να μάθουν περισσότερα για αυτά. Αυτά τα αντικείμενα θα μπορούν να ανταποκρίνονται στο άγγιγμα και το λόγο τους. Επίσης τα παιδιά θα μπορούν να συμμετέχουν σε πειράματα πραγματικού χρόνου που δικαιολογούν την αιτία και το αποτέλεσμα του πειράματος, το οποίο θα τους διδάξει πολλά για την επιστήμη.

Ανίχνευση και παρακολούθηση κλοπής αυτοκινήτων

Οι αισθητήρες που εφάπτονται σε ένα αυτοκίνητο μπορούν να προσδιορίσουν την ακριβή του θέση ανά πάσα στιγμή. Έτσι αν το αυτοκίνητο κλαπεί, θα είναι εύκολο για

την αστυνομία να εξακριβώσει γρήγορα και αποτελεσματικά που το έχουν μεταφέρει οι κλέφτες.

Κατηγοριοποίηση και έλεγχος λίστας αντικειμένων

Έστω μια αποθήκη στην οποία σε κάθε αντικείμενο εφάπτεται και ένας αισθητήρας. Ένας τελικός χρήστης μπορεί πολύ εύκολα να μάθει ποια αντικείμενα και πόσα περιέχονται στην αποθήκη αυτή. Αν θέλει να εισάγει μια νέα λίστα από τέτοια αντικείμενα, αρκεί να τους προσαρτήσει από ένα αισθητήρα

ΚΕΦΑΛΑΙΟ 3

ΠΑΡΑΓΟΝΤΕΣ ΠΟΥ ΣΥΜΒΑΛΟΥΝ ΣΤΟ ΣΧΕΔΙΑΣΜΟ ΤΩΝ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

3.1 ΑΝΤΟΧΗ ΣΕ ΣΦΑΛΜΑΤΑ

Είναι δυνατόν, ορισμένοι αισθητήριοι κόμβοι να μπλοκάρουν ή να αποτύχουν κατά τη λειτουργία τους, λόγω εξωτερικών περιβαλλοντολογικών παρεμβολών, έλλειψης ενέργειας ή φυσικής καταστροφής. Η αποτυχία ή καταστροφή (παροδική ή μόνιμη) μερικών αισθητήριων κόμβων δεν θα πρέπει να επηρεάζει τον συνολικό σκοπό του δικτύου αισθητήρων. Αυτό το θέμα αναφέρεται ως αξιοπιστία ή αντοχή σε σφάλματα. Η αντοχή σε σφάλματα είναι η δυνατότητα του δικτύου αισθητήρων να διατηρεί τη λειτουργικότητά του χωρίς διακοπές που να οφείλονται στις αποτυχίες των κόμβων του.

Οι αλγόριθμοι και τα πρωτόκολλα μπορούν να σχεδιαστούν, ώστε να εμπεριέχουν τα επίπεδα αντοχής σε λάθη που απαιτούνται από τα δίκτυα αισθητήρων. Αν το περιβάλλον στο οποίο πρόκειται να αναπτυχθεί ένα δίκτυο αισθητήρων δημιουργεί μικρές παρεμβολές τότε τα πρωτόκολλα μπορούν ανάλογα να είναι πιο ελαστικά. Για παράδειγμα, αν ένα δίκτυο αισθητήρων βρίσκεται εγκατεστημένο σε μια οικία προκειμένου να παρακολουθεί τα επίπεδα υγρασίας και θερμοκρασίας, η αντοχή σε σφάλματα μπορεί να είναι χαμηλή αφού τέτοιου είδους αισθητήριοι κόμβοι δεν καταστρέφονται και δεν επηρεάζονται εύκολα από το περιβάλλον. Αντιθέτως, σε ένα πεδίο μάχης, το δίκτυο αισθητήρων που θα εγκατασταθεί πρέπει να έχει μεγάλη αντοχή σε σφάλματα διότι είναι πολύ εύκολο να καταστραφούν αρκετοί κόμβοι του από εχθρικές επιχειρήσεις. Από τα παραπάνω διαπιστώνεται ότι η αντοχή σε σφάλματα εξαρτάται και από την εφαρμογή για την οποία προορίζεται το δίκτυο. Συνεπώς αυτό

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)
πρέπει να λαμβάνεται υπόψη στο σχεδιασμό του δικτύου αισθητήρων αλλά και των ίδιων των κόμβων

3.2 ΔΥΝΑΤΟΤΗΤΑ ΚΛΙΜΑΚΩΣΗΣ

Ο αριθμός των αισθητήριων κόμβων που έχουν αναπτυχθεί για την μελέτη ενός φαινομένου μπορεί να είναι της τάξης των εκατοντάδων ή χιλιάδων. Ανάλογα με την εφαρμογή, ο αριθμός αυτός μπορεί να φτάσει και την τάξη των εκατομμυρίων. Ό,τι πρωτόκολλο σχεδιαστεί θα πρέπει να μπορεί να χειριστεί αυτόν τον αριθμό των κόμβων. Πρέπει επίσης να χρησιμοποιήσουν την υψηλή πυκνότητα με την οποία εγκαθίστανται οι αισθητήριοι κόμβοι. Η πυκνότητα μπορεί να διαφέρει από μερικούς μέχρι εκατοντάδες κόμβους σε μια περιοχή η οποία μπορεί να είναι μικρότερη σε διάμετρο από 10m

Επιπλέον, ο αριθμός των κόμβων σε μια περιοχή μπορεί να χρησιμοποιηθεί για να δείξει την πυκνότητα των κόμβων. Η πυκνότητα αυτή εξαρτάται από την εφαρμογή για την οποία εγκαταστάθηκαν οι αισθητήριοι κόμβοι.

3.3 ΚΟΣΤΟΣ ΠΑΡΑΓΩΓΗΣ

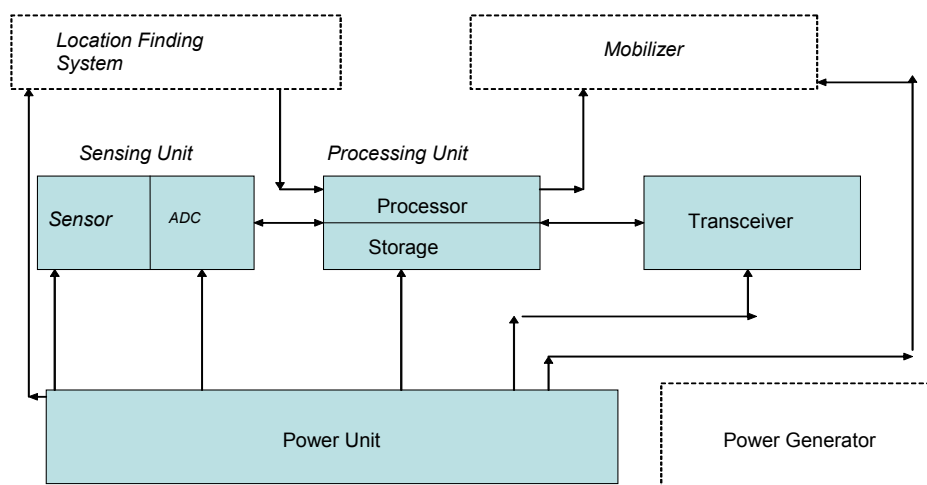
Αφού τα δίκτυα αισθητήρων αποτελούνται από ένα μεγάλο αριθμό κόμβων, το κόστος ενός μόνο αισθητήριου κόμβου είναι πολύ σημαντικό για ένα τέτοιο δίκτυο. Αν το κόστος του δικτύου είναι πιο ακριβό από το να εγκατασταθούν οι παραδοσιακοί αισθητήρες, τότε τα δίκτυα αισθητήρων δεν θα συμφέρουν οικονομικά. Αποτέλεσμα του παραπάνω είναι ότι το κόστος του κάθε αισθητήριου κόμβου πρέπει να είναι όσο το δυνατόν μικρότερο.

Το κόστος ενός ραδιοσυστήματος τεχνολογίας Bluetooth [2] , η οποία είναι μια χαμηλού κόστους συσκευή, είναι 10 φορές πιο ακριβή από την στοχευόμενη τιμή ενός αισθητήριου κόμβου. Επιπλέον ο αισθητήριοι κόμβος διαθέτει αισθητήρες και συσκευές επεξεργασίας. Επιπλέον αυτών, είναι δυνατόν να εξοπλιστεί με σύστημα εντοπισμού

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) θέσης με σύστημα κίνησης ή με σύστημα παραγωγής ενέργειας, ανάλογα με την εφαρμογή την οποία πρόκειται να εκτελέσει. Αποτέλεσμα όλων αυτών είναι ότι, η κατασκευή ενός αισθητήριου κόμβου με τόσο χαμηλό κόστος αποτελεί ένα πολύ ενδιαφέρον πρόβλημα.

3.4 ΠΕΡΙΟΡΙΣΜΟΙ ΥΛΙΚΟΥ

Ένας κόμβος ενός ασύρματου δικτύου αισθητήρων, όπως φαίνεται και στην παρακάτω εικόνα (εικόνα 3.1), αποτελείται κατά βάση από τέσσερα τμήματα: 1) μια μονάδα αίσθησης, 2) μια μονάδα επεξεργασίας, 3) ένα πομποδέκτη και 4) μια μονάδα ενέργειας.



Εικόνα 3.1 : ΤΜΗΜΑΤΑ ΕΝΟΣ ΚΟΜΒΟΥ ΔΙΚΤΥΟΥ ΑΙΣΘΗΤΗΡΩΝ [2]

Ανάλογα με την εφαρμογή για την οποία προορίζεται μπορεί να διαθέτει επιπλέον τμήματα, όπως σύστημα εντοπισμού θέσης, μονάδα παραγωγής ενέργειας και μονάδα κίνησης. Η μονάδα αίσθησης συνήθως αποτελείται από δύο υπομονάδες: α) τους αισθητήρες και β) τους αναλογικό-ψηφιακούς μετατροπείς. Τα αναλογικά σήματα που παράγονται από τα αισθητήρια όργανα και βασίζονται στα παρατηρούμενα φαινόμενα μετατρέπονται σε ψηφιακά σήματα από τους αναλογικό-ψηφιακούς μετατροπείς και

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

κατόπιν μεταφέρονται στην μονάδα επεξεργασίας. Αυτή η μονάδα, που γενικά σχετίζεται με μια μικρή μονάδα αποθήκευσης, διαχειρίζεται τις διαδικασίες που κάνουν τον αισθητήριο κόμβο να συνεργάζεται με άλλους κόμβους για να φέρει εις πέρας τους προσδιορισμένους στόχους. Η μονάδα του πομποδέκτη συνδέει τον αισθητήριο κόμβο στο δίκτυο. Ένα από τα πιο σημαντικά τμήματα του αισθητήριου κόμβου είναι η μονάδα ενέργειας. Οι μονάδες ενέργειας είναι δυνατόν να υποστηρίζονται από μια μονάδα εξαγωγής και παραγωγής ενέργειας (scavenging energy) από το περιβάλλον όπως οι

ηλιακές κυψέλες. Υπάρχουν όμως και άλλες μικρότερες μονάδες, των οποίων η χρήση εξαρτάται από την εφαρμογή για την οποία χρησιμοποιούνται οι αισθητήριοι κόμβοι.

Οι περισσότερες από τις τεχνικές δρομολόγησης και οι εφαρμογές παρακολούθησης των δικτύων αισθητήρων απαιτούν την γνώση της θέσης με μεγάλη συνήθως ακρίβεια. Έτσι είναι σύνηθες για ένα αισθητήριο κόμβο να έχει προσαρτημένη και μια μονάδα εύρεσης θέσης. Μια μονάδα κίνησης είναι δυνατόν να χρησιμοποιηθεί όταν απαιτείται να κινηθούν οι αισθητήριοι κόμβοι προκειμένου να παρακολουθήσουν καλύτερα το παρατηρούμενο φαινόμενο.

Όλες αυτές οι μικρότερες μονάδες πρέπει να μπορούν να χωρέσουν σε ένα χώρο μεγέθους σπιρτόκουτου. Το απαιτούμενο μέγεθος μπορεί να απαιτείται να είναι μικρότερο από ένα κυβικό εκατοστό και να είναι αρκετά ελαφρύ για να παραμένει αιωρούμενο στον αέρα. Εκτός από το μέγεθος, υπάρχουν ακόμα πιο αυστηροί περιορισμοί για τους αισθητήριους κόμβους όπως:

- Πρέπει να καταναλώνουν εξαιρετικά χαμηλή ενέργεια.
- Πρέπει να λειτουργούν ακόμα και σε πολύ πυκνή χωρική τοποθέτηση.
- Πρέπει να έχουν χαμηλό κόστος παραγωγής και να είναι αναλώσιμοι.
- Πρέπει να είναι αυτόνομοι και να λειτουργούν χωρίς παρακολούθηση.
- Πρέπει να προσαρμόζονται στο περιβάλλον που θα λειτουργούν.

Αφού οι αισθητήριοι κόμβοι είναι συνήθως δύσχρηστοι σε ότι αφορά τις απαιτήσεις τους και τη λειτουργικότητά τους, η διάρκεια ζωής ενός δικτύου αισθητήρων εξαρτάται άμεσα από την διάρκεια ζωής των πηγών ενέργειας των κόμβων. Η ενέργεια είναι ένας σπάνιος πόρος του συστήματος εξαιτίας των περιορισμών του μεγέθους. Για το σύστημα του ολοκληρωμένου ασύρματου δικτύου αισθητήρων (Wireless Integrated

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) [24] η ολική ενέργεια που πρέπει να παρέχεται πρέπει να είναι μικρότερη των 30mA προκειμένου να έχει μεγάλη διάρκεια λειτουργίας. Οι κόμβοι στο παραπάνω σύστημα παίρνουν ενέργεια από μια τυπική μπαταρία Λιθίου (Li) τύπου νομίσματος (2.5 cm διάμετρος και 1 cm πάχος). Είναι δυνατόν να επεκτείνουμε την διάρκεια ζωής των δικτύων αισθητήρων χρησιμοποιώντας τεχνικές εξαγωγής και παραγωγής ενέργειας από το περιβάλλον.

3.5 ΤΟΠΟΛΟΓΙΑ ΔΙΚΤΥΩΝ ΑΙΣΘΗΤΗΡΩΝ

Ένας μεγάλος αριθμός δύσκολα προσβάσιμων και χωρίς παρακολούθηση αισθητήριων κόμβων, οι οποίοι είναι επιρρεπείς στις καταστροφές, κάνει την διατήρηση της τοπολογίας του δικτύου μια μεγάλη πρόκληση. Η πυκνότητα μπορεί να φθάνει και τους 20 κόμβους/m³, κάτι που δυσκολεύει ακόμα περισσότερο την διαχείριση της τοπολογίας. Στις ενότητες που ακολουθούν εξετάζουμε την διατήρηση της τοπολογίας του δικτύου αισθητήρων σε 3 φάσεις.

- Φάση πριν την εγκατάσταση και φάση επανατοποθέτησης επιπλέον κόμβων
- Φάση μετά την εγκατάσταση.
- Φάση τοποθέτησης επιπλέον κόμβων.

ΦΑΣΗ ΠΡΙΝ ΤΗΝ ΕΓΚΑΤΑΣΤΑΣΗ ΚΑΙ ΦΑΣΗ ΕΠΑΝΑΤΟΠΟΘΕΤΗΣΗΣ ΕΠΙΠΛΕΟΝ ΚΟΜΒΩΝ

Οι αισθητήριοι κόμβοι μπορούν να τοποθετηθούν μαζικά, είτε ένας-ένας, στο χώρο. Συγκεκριμένα, μπορούν να εγκατασταθούν με τους εξής τρόπους:

- Να ριφθούν από αεροπλάνο,
- Να βρίσκονται σε ένα βλήμα πυροβολικού (ή πύραυλο) το οποίο εκρήγνυται και τους διασπείρει στην περιοχή,
- Να ριφθούν με ένα καταπέλτη, π.χ. από το κατάστρωμα ενός πλοίου,
- Να τοποθετηθούν ένας – ένας από ένα άνθρωπο ή ένα ρομπότ.

Ο μεγάλος αριθμός των αισθητήρων καθώς και η χωρίς παρακολούθηση εγκατάστασή τους συνήθως περιλαμβάνει την τοποθέτησή τους σύμφωνα με ένα προσεχτικά μελετημένο σχέδιο. Παρ' όλα αυτά, η αρχική εγκατάσταση πρέπει να πληροί κριτήρια όπως η μείωση του κόστους της εγκατάστασης, η μείωση της ανάγκης για κάποια προμελετημένη οργάνωση ή σχεδιασμό, η αύξηση της ευελιξίας τοποθέτησης και τέλος η προώθηση της αυτό-οργάνωσης και της αντοχής σε σφάλματα

Φάση μετά την εγκατάσταση.

Μετά την εξάπλωση των αισθητήριων κόμβων, οι αλλαγές στην τοπολογία τους οφείλονται σε αλλαγές που σχετίζονται με τους ίδιους τους αισθητήριους κόμβους, όπως:

- Θέση,
- Δυνατότητα επικοινωνίας,
- Διαθέσιμη ενέργεια,
- Δυσλειτουργία,
- Λεπτομέρειες στο σκοπό για τον οποίο εγκαταστάθηκαν.

Οι τοπολογίες των δικτύων αισθητήρων υπόκεινται σε συχνές αλλαγές, ως αποτέλεσμα των γεγονότων που περιγράφονται παρακάτω:

- Η στατική εγκατάσταση των αισθητήριων κόμβων,
- Συχνές αποτυχίες λόγω έλλειψης ενέργειας ή καταστροφής,
- Κίνηση των αισθητήριων κόμβων,
- Δολιοφθορές.

Φάση τοποθέτησης επιπλέον κόμβων.

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

Επιπλέον κόμβοι είναι δυνατόν να εγκατασταθούν οποιαδήποτε χρονική στιγμή για να αντικαταστήσουν τους κόμβους που παρουσιάζουν δυσλειτουργίες ή λόγω αλλαγών στον αρχικό σκοπό για τον οποίο εγκαταστάθηκαν. Η πρόσθεση νέων κόμβων στο δίκτυο δημιουργεί την ανάγκη για επαναδιοργάνωση. Προκειμένου να αντιμετωπίσουμε τις συχνές αλλαγές στην τοπολογία ενός ασύρματου δικτύου αισθητήρων, το οποίο αποτελείται από ένα μεγάλο αριθμό κόμβων με μεγάλους περιορισμούς στην κατανάλωση ενέργειας, χρειαζόμαστε ειδικά σχεδιασμένα πρωτόκολλα δρομολόγησης.

3.6 ΠΕΡΙΒΑΛΛΟΝ

Οι αισθητήριοι κόμβοι τοποθετούνται , είτε πολύ κοντά στο υπό παρατήρηση φαινόμενο, είτε ακριβώς μέσα σε αυτό. Έτσι συνήθως εργάζονται χωρίς παρακολούθηση σε απομακρυσμένες γεωγραφικές περιοχές. Για παράδειγμα, είναι δυνατόν να εργάζονται:

- Στο εσωτερικό ενός μεγάλου μηχανήματος,
- Στα βάθη του ωκεανού,
- Μέσα σε ένα κυκλώνα,
- Στην επιφάνεια ενός ωκεανού στην διάρκεια μια καταιγίδας,
- Σε μια περιοχή μολυσμένη από ραδιενέργεια ή χημικές ουσίες,
- Στο πεδίο της μάχης, πίσω από τις γραμμές του εχθρού,
- Σε ένα σπίτι ή σε ένα μεγάλο κτίριο,
- Σε μια μεγάλη αποθήκη,
- Εμφυτευμένοι σε ζώα,
- Ενσωματωμένοι σε ταχέως κινούμενα οχήματα,
- Στα νερά ενός ποταμού.

Η παραπάνω λίστα μας δίνει μια περιγραφή των συνθηκών, υπό τις οποίες δουλεύουν οι κόμβοι ενός ασύρματου δικτύου αισθητήρων.

3.7 ΜΕΣΑ ΜΕΤΑΔΟΣΗΣ

Σ' ένα ασύρματο δίκτυο αισθητήρων, οι αισθητήριοι κόμβοι συνδέονται συνήθως μέσω του ασύρματου μέσου. Αυτού του είδους οι συνδέσεις μπορούν να πραγματοποιηθούν με τη βοήθεια ραδιοσυχνοτήτων, υπέρυθρων ή οπτικών μέσων. Προκειμένου να γίνει εφικτή μια παγκόσμια χρήση αυτών των δικτύων, το επιλεγμένο μέσο πρέπει να είναι διαθέσιμο παγκοσμίως.

Ευρέως χρησιμοποιούμενη συχνότητα στις ασύρματες ζεύξεις είναι η μπάντα ISM (Industrial Scientific Medical ISM Band), η οποία προσφέρεται και χωρίς άδεια χρήσης στις περισσότερες χώρες. Στην Ευρώπη χρησιμοποιούνται οι συχνότητες των 433 MHz και στην Β. Αμερική οι συχνότητες των 915 MHz.

Στις περισσότερες περιπτώσεις δικτύων αισθητήρων χρησιμοποιούμε την επικοινωνία η οποία γίνεται μέσω ραδιοσυχνοτήτων. Μεταξύ αυτών διακρίνουμε τις εξής περιπτώσεις :

- Το μAMPS το οποίο χρησιμοποιεί πομποδέκτη συμβατό με Bluetooth στα 2.4 GHz, με ένα ενσωματωμένο συνθέτη συχνοτήτων,
- Ένας αισθητήρας χαμηλής ενέργειας ο οποίος χρησιμοποιεί πομποδέκτη ραδιοσυχνότητας ενός καναλιού, το οποίο βρίσκεται σε συχνότητα λειτουργίας στα 916 MHz,
- Η αρχιτεκτονική WINS[11] χρησιμοποιεί ραδιοσυχνότητες μεταξύ των κόμβων.

Ένας άλλος τρόπος επικοινωνίας στα δίκτυα αισθητήρων είναι η χρήση υπέρυθρων, για την οποία δεν απαιτείται άδεια χρήσης, και επιπλέον είναι ανθεκτική στις παρεμβολές από ηλεκτρικές συσκευές. Το μοναδικό μειονέκτημα που εντοπίζουμε είναι η απαίτηση για οπτική επαφή των επικοινωνούντων συσκευών. Το μειονέκτημα αυτό αποτρέπει τη χρήση υπέρυθρων ως μέσο μετάδοσης στα ασύρματα δίκτυα αισθητήρων.

Οι συνήθεις απαιτήσεις των εφαρμογών για τις οποίες χρησιμοποιούνται τα δίκτυα αισθητήρων δημιουργούν μεγάλη πρόκληση στην επιλογή ενός μέσου μετάδοσης. Για παράδειγμα, σε εφαρμογές που μπορούν να είναι υποθαλάσσιες μπορεί να απαιτείται η

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

χρήση του νερού ως μέσου μετάδοσης. Επιπλέον, λόγω του ότι η κεραία ενός αισθητήρα μπορεί να μην έχει το απαιτούμενο ύψος, ή η ισχύ εκπομπής να είναι περιορισμένη, εκτός από την επιλογή του μέσου, μεγάλο ρόλο παίζει η χρήση ισχυρής κωδικοποίησης και η επιλογή της κατάλληλης συχνότητας προκειμένου να γίνει στο έπακρο εκμετάλλευση των χαρακτηριστικών του καναλιού.

3.8 ΚΑΤΑΝΑΛΩΣΗ ΕΝΕΡΓΕΙΑΣ

Δεδομένου ότι ο κάθε αισθητήριος κόμβος αποτελεί αυτοτελή μικροηλεκτρονική συσκευή, μπορεί να εφοδιαστεί με μια περιορισμένη πηγή ενέργειας (<0.5 Ah, 1.2V). Δεδομένου ότι η αντικατάσταση αυτής της πηγής ενέργειας είναι αδύνατη, η ζωή του αισθητήριου κόμβου εξαρτάται από αυτήν. Σε κάθε δίκτυο αισθητήρων ο κάθε κόμβος παίζει το ρόλο του αποστολέα και του δρομολογητή, επομένως εάν παρουσιαστεί κάποια βλάβη σε έναν από τους κόμβους του δικτύου, απαιτείται συνολική αναδιοργάνωση του δικτύου και επαναδρομολόγηση των μηνυμάτων. Η σωστή διαχείριση της ενέργειας των κόμβων παίζει μεγάλο ρόλο, και για αυτό το λόγο αποδίδεται σε τρεις λειτουργίες: 1) Επικοινωνία, 2) αίσθηση και 3) επεξεργασία δεδομένων.

3.8.1 ΕΠΙΚΟΙΝΩΝΙΑ

Η πιο απαιτητική λειτουργία από άποψη κατανάλωσης ενέργειας είναι η επικοινωνία. Συνήθως για τις μικρές αποστάσεις που λειτουργούν οι αισθητήριοι κόμβοι η κατανάλωση είναι ίδια κατά την εκπομπή και την λήψη. Βεβαίως, εκτός από αυτό, σοβαρό ρόλο παίζει και το άνοιγμα και κλείσιμο του κυκλώματος του πομποδέκτη.

3.8.2 ΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ

Η κατανάλωση ενέργειας κατά τη φάση της επεξεργασίας δεδομένων είναι μικρότερη από αυτή κατά τη φάση της επικοινωνίας. Συνεπώς, ο κόμβος κατά τη φάση της επεξεργασίας θα έχει ενσωματωμένο ένα κύκλωμα επεξεργασίας προκειμένου να επεξεργάζεται τα δεδομένα με απώτερο σκοπό την αποστολή λιγότερων πακέτων κατά τη φάση της επικοινωνίας

ΚΕΦΑΛΑΙΟ 4

ΥΠΑΡΧΟΥΣΕΣ ΤΕΧΝΟΛΟΓΙΕΣ MIDDLEWARE

Σε αυτό το κεφάλαιο θα γίνει αναφορά σε μερικά από τα πιο γνωστά υπάρχοντα συστήματα ενδιάμεσου λογισμικού που έχουν παρουσιαστεί τα τελευταία χρόνια για ασύρματα δίκτυα αισθητήρων. Οι περισσότερες σύγχρονες εφαρμογές είναι συσχετισμένες με συγκεκριμένες τεχνολογίες ενδιάμεσου λογισμικού ασύρματων δικτύων αισθητήρων.

Τα σημερινά δίκτυα ασύρματων αισθητήρων στοχεύουν σε εφαρμογές συλλογής δεδομένων και στις περισσότερες περιπτώσεις υποστηρίζουν τουλάχιστον μία τέτοια εφαρμογή ανά εγκατεστημένο δίκτυο. Γι' αυτό το λόγο μέχρι σήμερα ο σχεδιασμός των πρωτοκόλλων του δικτύου είναι άρρηκτα συνδεδεμένος με τις εφαρμογές που θα εκτελούνται σε αυτό. Τέτοιες όμως διαδικασίες είναι ad-hoc και επιβάλλουν άμεση αλληλεπίδραση με το ενσωματωμένο λειτουργικό σύστημα, ή ακόμα και το υλικό των κόμβων [12].

Όπως είναι προφανές κάτι δεν βοηθά στην κατασκευή αισθητήριων κόμβων οι οποίοι θα μπορούν να λειτουργήσουν όπως οι σημερινοί υπολογιστές, δηλαδή ανεξάρτητοι από το δίκτυο και τις εφαρμογές. Συνεπώς η εξάπλωση τους παραμένει μια διαδικασία δύσκολη και ακριβή. Θα πρέπει να δημιουργηθούν μέθοδοι σχεδιασμού εφαρμογών που να μη βασίζονται στα πρωτόκολλα και στο υλικό των δικτύων αισθητήρων. Επιπροσθέτως θα πρέπει να προβλεφθεί η περίπτωση όπου πολλαπλές εφαρμογές θα εκτελούνται ταυτόχρονα σε ένα δίκτυο αισθητήρων.

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

Από τα παραπάνω καθίσταται αναγκαία η ύπαρξη ενός ενδιάμεσου λογισμικού (middleware) το οποίο θα βρίσκεται μεταξύ των εφαρμογών και των λειτουργικών συστημάτων καθώς και του υλικού του δικτύου. Αυτό θα πρέπει να παρέχει :

- α) Τυποποιημένες υπηρεσίες συστήματος σε διάφορες εφαρμογές
- β) ένα περιβάλλον εκτέλεσης που θα υποστηρίζει και θα συντονίζει πολλαπλές εφαρμογές και
- γ) μηχανισμούς που θα επιτυγχάνουν την προσαρμόσιμη και αποτελεσματική χρησιμοποίηση των πόρων του εκάστοτε συστήματος.

Τέτοιο ενδιάμεσο λογισμικό είναι ιδιαίτερος χρήσιμο σε δίκτυα ασύρματων αισθητήρων που φιλοξενούν πολύπλοκες εφαρμογές με μεγάλο όγκο επεξεργασίας πληροφοριών και με αυστηρούς περιορισμούς στην κατανάλωση της ενέργειας.

Αν και έχουν γίνει αρκετές προσπάθειες για την ανάπτυξη διαφόρων πρωτοκόλλων επικοινωνίας και δρομολόγησης το θεμελιώδες πρόβλημα της αναγνώρισης, σχεδίασης και ανάπτυξης ενός κατάλληλου ενδιάμεσου λογισμικού (middleware) που θα εξυπηρετεί πλήρως τις ικανότητες και εφαρμογές των δικτύων αισθητήρων, δεν έχει λυθεί ακόμα. Τα παραδοσιακά καταμεμημένα ενδιάμεσα λογισμικά (distributed middleware) (όπως το DCOM, CORBA, PVM) είναι συνήθως «βαριά» σε κατανάλωση μνήμης και απαιτήσεων υπολογιστικής δύναμης, οπότε και καθίστανται ακατάλληλα για το περιβάλλον των δικτύων ασύρματων αισθητήρων. Για την περίπτωση μας αναγκαστικά ένας σχεδιασμός απλός, εύκολα υλοποιήσιμος, και «ελαφρύς» όσον αφορά στην κατανάλωση πόρων του συστήματος (ενέργεια, μνήμη, υπολογιστική ισχύς). Επιπλέον πρέπει να ανταποκρίνεται στα χαρακτηριστικά των δικτύων ασύρματων αισθητήρων που διαφέρουν από τα παραδοσιακά δίκτυα, όπως είναι η μη σχεδιασμένη από πριν

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) εγκατάσταση (ad-hoc deployment), η μη παρακολουθούμενη λειτουργία και η λειτουργία σε δυναμικά περιβάλλοντα.

4.1ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΟΥ ΕΝΔΙΑΜΕΣΟΥ ΛΟΓΙΣΜΙΚΟΥ.

Το ενδιάμεσο λογισμικό είναι συνήθως λογισμικό που ενώνει τους παροχείς υπηρεσιών (service suppliers) και τους χρήστες των υπηρεσιών αυτών (service consumers). Παροχέας υπηρεσιών θεωρείται οποιοσδήποτε τύπος δικτυακού κόμβου (υλικό ή λογισμικό) που μπορεί να προσφέρει υπηρεσίες (π.χ. εκτυπωτές, αισθητήρες, βάσεις δεδομένων και εφαρμογές). Αντίστοιχα, χρήστης υπηρεσιών είναι οποιοσδήποτε τύπος

δικτυακού κόμβου που απαιτεί υπηρεσίες από ένα παροχέα. Οι εφαρμογές επιπλέον μπορούν να συμπεριφέρονται και σαν παροχείς υπηρεσιών αλλά και σαν χρήστες των παρεχομένων υπηρεσιών ταυτόχρονα [13].

Χαρακτηριστικά θα μπορούσαμε να αναφέρουμε πως το ενδιάμεσο λογισμικό χαρακτηρίζεται από τα παρακάτω: παρουσιάζει τα εξής χαρακτηριστικά:

- **Ανεξαρτησία από το δίκτυο**

Το ενδιάμεσο λογισμικό (middleware) συνήθως λειτουργεί σαν μια γέφυρα μεταξύ πολλαπλών δικτυακών εφαρμογών, ή πρέπει να προσαρμοστεί σε πολλαπλά υποκείμενα δίκτυα. Αυτό περικλείει την ικανότητα να συνδέονται ενσύρματες δικτυακές τεχνολογίες, όπως τοπικά δίκτυα Ethernet και ATM δίκτυα κορμού μεταξύ τους ή με άλλα ασύρματα δίκτυα όπως Bluetooth δίκτυα, 802.11 ή υπέρυθρα ασύρματα δίκτυα. Πέρα από όλα αυτά ένα ενδιάμεσο λογισμικό για να είναι ευέλικτο θα πρέπει να είναι ανεξάρτητο από το δίκτυο στο οποίο εφαρμόζεται.

- **Εύρεση θέσης και δρομολόγηση**

Σε πολλά συστήματα, ειδικά αν πρόκειται για κινητά συστήματα, το ενδιάμεσο λογισμικό απαιτεί γνώση της θέσης του συστήματος και την ύπαρξη μηχανισμού για επικοινωνία

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) μέσω διαδοχικών κόμβων(multiple network hops). Αντί να ενσωματωθούν τέτοιες υπηρεσίες στην κάθε εφαρμογή θα ήταν καλύτερο να ενσωματωθούν στο ενδιάμεσο λογισμικό. Αν και τα πρωτόκολλα ανίχνευσης της υπηρεσίας μπορούν να προσφέρουν λειτουργίες εύρεσης θέσης, μπορεί να μην έχουν επαρκή πληροφορία για τα κατώτερα επίπεδα του δικτύου προκειμένου να πάρουν τις βέλτιστες αποφάσεις. Γι' αυτό, το ενδιάμεσο λογισμικό πρέπει να μπορεί να εκμεταλλεύεται την πληροφορία από τα κατώτερα επίπεδα του δικτύου για εύρεση της θέσης και δρομολόγηση όταν απαιτούνται τέτοιοι έλεγχοι.

- **Χρονοπρογραμματισμός**

Μερικά περιβάλλοντα έχουν αυστηρούς περιορισμούς σε λειτουργίες που μπορούν να εκτελεστούν σε μια συγκεκριμένη χρονική στιγμή. Το ενδιάμεσο λογισμικό μπορεί να αποφασίσει για την χρονική σειρά των αλληλεπιδράσεων, βάσει προτεραιοτήτων ή περιορισμών του εύρους ζώνης. Για παράδειγμα, αν μια υπηρεσία πρόκειται να διακοπεί (π.χ. παροχή υπηρεσίας σε κινητό το οποίο κινείται εκτός εμβελείας), τότε η δοσοληψία που την αφορά θα πρέπει είτε να ολοκληρωθεί είτε να μεταφερθεί σε άλλη υπηρεσία που ταιριάζει με τους περιορισμούς της αρχικής. Αυτές οι αλληλεπιδράσεις μπορούν να χρόνο-προγραμματιστούν με υψηλή προτεραιότητα και πιθανώς να τους αποδοθεί περισσότερο εύρος ζώνης. Παρόμοια ζητήματα χρόνο-προγραμματισμού τίθενται στο grid computing όπου το ενδιάμεσο λογισμικό πρέπει να καθορίζει σειρά εκτέλεσης των διεργασιών στους επεξεργαστές.

- **Διαλειτουργικότητα**

Σε μερικά συστήματα το ενδιάμεσο λογισμικό δίνει έμφαση στην ανάγκη σύνδεσης μεταξύ πολλαπλών γλωσσών και /ή μεταξύ πλατφόρμων ενδιάμεσου λογισμικού. Το CORBA μπορεί να συνδεθεί με τις περισσότερες κοινές εφαρμογές που βασίζονται σε διαφορετικές γλώσσες προγραμματισμού, αλλά πρέπει να υπάρχει ένα αντικείμενο CORBA, το οποίο μερικές φορές είναι δύσκολο να υλοποιηθεί σε μικρά συστήματα.

Υπάρχει γενικότερος προβληματισμός αν θα είναι επιτυχές το παραπάνω σε ένα μεγάλης κλίμακας δίκτυο όπως το Διαδίκτυο. Θα πρέπει ταυτόχρονα με την διαλειτουργικότητα να ζυγιστεί προσεκτικά το κόστος της, ειδικά όταν μιλάμε για ενσωματωμένα (embedded) συστήματα. Για ένα οποιοδήποτε σύστημα, είναι αναγκαία η χρήση μιας σημασιολογικής γλώσσας (markup language) όπως είναι η XML ή οποιασδήποτε άλλης η οποία παρέχει σημασιολογική ανεξαρτησία (semantics independence), για να έχουμε εγγυημένη διαλειτουργικότητα (guarantee interoperability).

4.2 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΤΩΝ ΕΦΑΡΜΟΓΩΝ

Αν και οι εφαρμογές των δικτύων αισθητήρων καλύπτουν ένα μεγάλο εύρος δραστηριοτήτων μπορούν να χαρακτηριστούν από αρκετά κοινά χαρακτηριστικά. Συνήθως οι αισθητήρες λειτουργούν με μπαταρίες και εγκαθίστανται σε μια περιοχή προκειμένου να παρακολουθήσουν διάφορα φαινόμενα. Προκειμένου να εξοικονομήσουν ενέργεια, οι αισθητήρες μπορούν να έχουν διάφορες καταστάσεις λειτουργίας, και συνεπώς διαφορετική ενεργειακή κατανάλωση [12].

Οι μηχανισμοί που αλλάζουν τις καταστάσεις λειτουργίας ενός αισθητήρα καλούνται «μοχλοί» του συστήματος (system “knobs”). Αρχικά, προκειμένου να εξοικονομήσουν ενέργεια οι αισθητήρες μπορούν να βρίσκονται σε κατάσταση «ύπνου», εκτός από μερικούς οι οποίοι αναλαμβάνουν το ρόλο της «φύλαξης». Όταν οι ενεργοί (αυτοί που δεν βρίσκονται σε κατάσταση «ύπνου») αισθητήριοι κόμβοι διαπιστώσουν την ύπαρξη κάποιου ανιχνεύσιμου φαινομένου, είναι υπεύθυνοι να ενεργοποιήσουν τον απαιτούμενο αριθμό κόμβων προκειμένου να ανιχνευθεί καλύτερα το φαινόμενο αυτό.

Προκειμένου να εκπληρωθεί η παραπάνω διαδικασία πρέπει να ληφθούν υπόψη κάποια ζητήματα, όπως :

- Η δημιουργία μηχανισμών οι οποίοι θα θέτουν σε λειτουργία επιλεκτικά κάποιους αισθητήριους κόμβους, οι οποίοι βρίσκονται σε κατάλληλες θέσεις γύρω από την περιοχή του εκάστοτε στόχου και ταυτόχρονα έχουν αρκετή εναπομένουσα ενέργεια.
- Η συνεργασία μεταξύ των ενεργοποιημένων κόμβων για να κατανέμουν το συνολικό επεξεργαστικό φόρτο, να συγκεράσουν (aggregate) τα αποτελέσματα και τέλος να δρομολογήσουν το τελικό αποτέλεσμα στο σταθμό βάσης.
- Τυχόν απαιτήσεις ποιότητας υπηρεσίας που θα πρέπει να ικανοποιηθούν. Πιο συγκεκριμένα, πρόκειται για απαιτήσεις οι οποίες είναι σχετικές με την ακρίβεια της απόφασης, και οι οποίες καθορίζουν την απαιτούμενη ποσότητα της ενέργειας καθώς και των πόρων για υπολογισμούς, επικοινωνίες, και ανίχνευση.
- Η διαχείριση του δικτύου, που μπορεί να γίνει αρκετά πολύπλοκη λόγω της ανομοιογένειας των αισθητήριων κόμβων (στο δίκτυο μπορεί να υπάρχουν κόμβοι με διαφορετικά επίπεδα ενέργειας, δυνατότητες επεξεργασίας και ανίχνευσης).

Τα παραπάνω ζητήματα δεν μπορούν να παραβλεφθούν, αν αναλογιστούμε α) σε κάποιες περιπτώσεις, τα ad-hoc χαρακτηριστικά του δικτύου, β) την περιορισμένη ενέργεια και τις δυνατότητες υπολογιστικής ισχύος, επικοινωνίας και ανίχνευσης των αισθητήρων, γ) τις πιθανές μεταβολές στις συνθήκες του συστήματος και του περιβάλλοντος και δ) το γεγονός ότι οι αισθητήριοι κόμβοι λειτουργούν στο περιβάλλον χωρίς κάποια ανθρώπινη παρακολούθηση. Προκειμένου να περιοριστεί η πολυπλοκότητα για τους σχεδιαστές των εφαρμογών, το ενδιάμεσο λογισμικό πρέπει να παρέχει ένα περιβάλλον εκτέλεσης που θα λαμβάνει υπόψη τα παραπάνω χαρακτηριστικά και περιορισμούς. Επίσης για να εγγυηθεί η σωστή λειτουργία πολλαπλών εφαρμογών κάτω από τους αυστηρούς ενεργειακούς περιορισμούς, το ενδιάμεσο λογισμικό πρέπει να είναι ικανό να «θυσιάζει» αποτελεσματικά την ποιότητα υπηρεσίας μίας εφαρμογής προς όφελος κάποιας άλλης και αντίστροφα.

4.3 ΑΡΧΕΣ ΣΧΕΔΙΑΣΗΣ ΤΟΥ ΕΝΔΙΑΜΕΣΟΥ ΛΟΓΙΣΜΙΚΟΥ (MIDDLEWARE)

Προκειμένου το ενδιάμεσο λογισμικό να είναι σε θέση να ανταποκριθεί στα χαρακτηριστικά των δικτύων ασύρματων αισθητήρων θα πρέπει κι αυτό με τη σειρά του να ακολουθεί κάποιες βασικές αρχές[12] [14], όπως οι παρακάτω :

α) Δεδομένο-κεντρικοί Μηχανισμοί (Data-centric mechanisms): Το ενδιάμεσο λογισμικό πρέπει να παρέχει δεδομένο-κεντρικούς μηχανισμούς για την επεξεργασία των δεδομένων και τις ερωτήσεις στο δίκτυο. Λόγω της απλότητας, ευελιξίας, και ευρωστίας (robustness) οι αρχιτεκτονικές που βασίζονται στην ομαδοποίηση έχουν ευρεία χρήση στο σχεδιασμό και την υλοποίηση των πρωτοκόλλων του δικτύου.

β) Γνώση της εφαρμογής (application knowledge): Μπορεί να χρησιμοποιηθεί προκειμένου να συνδέσει το σχεδιασμό και την υλοποίηση του λογισμικού. Είναι, λοιπόν, σημαντικό να ενοποιηθεί η γνώση της εφαρμογής με τις προσφερόμενες υπηρεσίες από

το ενδιάμεσο λογισμικό. Όμως είναι δυνατόν, λόγω της αναγκαιότητας για υποστήριξη και βελτιστοποίηση ενός μεγάλου αριθμού εφαρμογών, να γίνεται μια ανταλλαγή μεταξύ του βαθμού υποστήριξης μιας συγκεκριμένης εφαρμογής και της γενικότητας του ενδιάμεσου λογισμικού. Μια εφαρμόσιμη πρακτική είναι να ενσωματώνονται τα μοναδικά χαρακτηριστικά της εφαρμογής μέσα στον κώδικά της, ο οποίος θα μπορεί να διερμηνευτεί από το ενδιάμεσο λογισμικό. Αποτέλεσμα είναι οι πληροφορίες αυτές για τα χαρακτηριστικά και τις δυνατότητες που έχει η εφαρμογή να μπορούν αξιοποιηθούν και να χρησιμοποιηθούν για να κατευθύνουν τις λειτουργίες του ενδιάμεσου λογισμικού.

γ) Τοπικοί αλγόριθμοι (localized algorithms): θα πρέπει να χρησιμοποιούνται ώστε να επιτυγχάνουν συλλογικά ένα επιθυμητό μαζικό στόχο, ενώ ταυτόχρονα να προσφέρουν επεκτασιμότητα και ευρωστία στο σύστημα. Δεδομένου ότι η ομαδοποιημένη αρχιτεκτονική κάνει την αλληλεπίδραση των αισθητήριων κόμβων τοπική και ως εκ τούτου και το συντονισμό και έλεγχο των επιπλέον απαιτήσεων σε μια συγκεκριμένη περιοχή, είναι δικαιολογημένο να θεωρείται η κάθε ομάδα σαν μια βασική οντότητα του ενδιάμεσου λογισμικού. Συμπερασματικά, το ενδιάμεσο λογισμικό δρα σαν ένα καταμεμημένο λογισμικό αποτελούμενο από πολλαπλές ομάδες.

δ) Αλγόριθμοι Προσαρμόσιμης Πιστότητας (Adaptive fidelity algorithms): Αυτοί επιτρέπουν το αντιστάθμισμα μεταξύ της ποιότητας του αποτελέσματος και της χρήσης πόρων του συστήματος, γι' αυτό και πρόκειται για μια αρχή σχεδιασμού η οποία συμβάλει καθοριστικά στην αποτελεσματική χρήση των πόρων. Μια ιδανική περίπτωση

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

είναι η εφαρμογή να μπορεί να διαλέξει μεταξύ πολλών διαφορετικών αλγορίθμων οι οποίοι επιλύουν το ίδιο πρόβλημα με διαφορετικές απαιτήσεις στην ποιότητα του αποτελέσματος και στην χρήση πόρων των κόμβων του δικτύου

ε) Ελαφρύ στη λειτουργία (lightweight): Αφού οι διαθέσιμες πηγές ενέργειας των αισθητήριων κόμβων είναι μικρές, ένα άλλο χαρακτηριστικό που πρέπει να διαθέτει το ενδιάμεσο λογισμικό είναι να είναι «ελαφρύ» σε απαιτήσεις υπολογισμών και επικοινωνίας. Το παραπάνω χαρακτηριστικό απαιτεί την χρήση απλών και αποτελεσματικών μεθόδων για λιγότερο βέλτιστες λύσεις.

στ) Λόγω των περιορισμένων πόρων, είναι πολύ πιθανό ότι δεν θα μπορούν να ικανοποιηθούν ταυτόχρονα οι απαιτήσεις απόδοσης των εκτελέσιμων εφαρμογών. Γι' αυτό και είναι αναγκαίο το ενδιάμεσο λογισμικό να μπορεί να διαπραγματεύεται την ποιότητα υπηρεσίας μεταξύ των εφαρμογών.

Στα συμβατικά υπολογιστικά συστήματα, κάθε συσκευή έχει και ένα κάτοχο, ο οποίος είναι υπεύθυνος για την ρύθμιση, τη συντήρηση και τον χειρισμό της σε περίπτωση λάθους ή άλλης αστοχίας. Σε αντίθεση, τα δίκτυα ασύρματων αισθητήρων πρέπει να μπορούν να λειτουργούν χωρίς παρακολούθηση, γεγονός που σημαίνει ότι το ενδιάμεσο λογισμικό πρέπει να παρέχει επίπεδα υποστήριξης για αυτόματη ρύθμιση και χειρισμό λαθών. Άλλο θέμα αποτελεί η χρήση των δικτύων ασύρματων αισθητήρων για την παρακολούθηση γεγονότων του πραγματικού κόσμου με αποτέλεσμα ο χρόνος και η θέση να παίζουν πολύ σημαντικό ρόλο. Επίσης μερικές εφαρμογές μπορεί να έχουν απαιτήσεις για λειτουργία και εξαγωγή αποτελεσμάτων σε πραγματικό χρόνο. Αυτό έχει ως αποτέλεσμα το ενδιάμεσο λογισμικό να πρέπει να έχει από κατασκευής του, ενσωματωμένα τα χαρακτηριστικά για τη διαχείριση της έννοιας του χρόνου και της θέσης.

4.4 ΚΑΤΗΓΟΡΙΕΣ ΕΝΔΙΑΜΕΣΟΥ ΛΟΓΙΣΜΙΚΟΥ

Σύμφωνα με το [14] μπορούμε να ομαδοποιήσουμε τις υπάρχουσες προσεγγίσεις σε ό,τι αφορά το ενδιάμεσο λογισμικό στις επόμενες τρεις κατηγορίες:

- Κινητών πρακτόρων (Mobile Agents),.

Η κατηγορία αυτή προσεγγίζει το δίκτυο εμπνευσμένη από τους κινητούς πράκτορες και τον κινητό κώδικα (mobile code) και λειτουργεί εισάγοντας ένα πρόγραμμα στο δίκτυο. Το πρόγραμμα αυτό μπορεί να συγκεντρώσει δεδομένα από τους κόμβους, να διαδώσει αντίγραφα του εαυτού του στο δίκτυο και να επικοινωνεί με αυτά.

- Βάσεων Δεδομένων

Η κατηγορία αυτή χειρίζεται το δίκτυο σαν μια κατανεμημένη βάση δεδομένων όπου οι χρήστες μπορούν να θέτουν ερωτήματα τύπου SQL προκειμένου το δίκτυο να εκτελέσει κάποια αισθητήρια λειτουργία.

- Συμβάντων (Events)

Η κατηγορία αυτή βασίζεται στην έννοια των συμβάντων. Σύμφωνα με αυτή τη προσέγγιση οι εφαρμογές επικεντρώνουν το ενδιαφέρον τους σε αλλαγές της κατάστασης των αντικειμένων που παρακολουθούν στον πραγματικό

κόσμο δηλαδή στα συμβάντα που λαμβάνουν χώρα. Όταν ένας αισθητήρας αντιληφθεί ένα τέτοιο γεγονός, στέλνει μια ειδοποίηση συμβάντος προς την εφαρμογή. Η εφαρμογή μπορεί όμως να θέσει περιορισμούς στις ειδοποιήσεις θέτοντας κάποια κριτήρια τα οποία πρέπει να πληρούνται προκειμένου οι αισθητήρες να στείλουν τέτοιου είδους ειδοποιήσεις.

4.5 ΠΡΟΣΕΓΓΙΣΗ ΚΙΝΗΤΩΝ ΠΡΑΚΤΟΡΩΝ

Στα περισσότερα δίκτυα θεωρείται ότι οι προτεινόμενοι αλγόριθμοι, είναι ενσωματωμένοι στην μνήμη του κάθε κόμβου. Σε μερικές πλατφόρμες, όπως το TinyOS, ο σχεδιαστής εφαρμογών χρησιμοποιεί ένα λειτουργικό σύστημα με επίπεδα στον κάθε κόμβο, το οποίο έχει πολλά πλεονεκτήματα καθώς προσφέρει συναρμολογισμότητα, πολυδιεργασία, και μία αφαιρετική άποψη του υλικού. Ακόμα και σε αυτήν την περίπτωση ο σχεδιαστής πρέπει να δημιουργήσει ένα εκτελέσιμο πρόγραμμα το οποίο θα το φορτώσει χειροκίνητα σε κάθε ένα κόμβο.

Μια προσέγγιση, η οποία κερδίζει μεγάλη απήχηση τελευταία, είναι η προσέγγιση του ενεργού αισθητήρα («active sensor» approach). Ο όρος χρησιμοποιείται για να περιγράψει μια ομάδα πλαισίων (frameworks) τα οποία προσπαθούν να

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

προγραμματίσουν τους αισθητήρες με ένα συνηθισμένο τρόπο. Ο τρόπος αυτός είναι παρόμοιος με αυτόν που τα πλαίσια (frameworks) προγραμματίζουν τους κόμβους ενός οποιουδήποτε δικτύου. Η διαφορά ανάμεσα στα δύο είναι το γεγονός ότι στο πρώτο δίκτυο πρέπει να προγραμματίσουμε να αντιδρά μόνο όταν λαμβάνει ένα πακέτο δεδομένων, ενώ στα ασύρματα δίκτυα αισθητήρων έχουμε αντίδραση σε πολλών ειδών γεγονότα, όπως αίσθησης, δικτύου, και εκπνοής χρόνου.

4.5.1 SENSOR WARE

Η αρχιτεκτονική του SensorWare [15] βασίζεται σε ένα περιβάλλον εκτέλεσης (run-time) με δυνατότητα εγγραφής μικρών προγραμμάτων (scriptable). Το περιβάλλον αυτό έχει βελτιστοποιηθεί για χρήση σε αισθητήριους κόμβους με περιορισμένη ενέργεια και μνήμη. Είναι συμβατό για ένα ή περισσότερα απλά, συμπυκνμένα, και ανεξάρτητα

πλατφόρμας σενάρια (script) ελέγχου αισθητήριων κόμβων. Οι πόροι για ανίχνευση, επικοινωνία και επεξεργασία σήματος ενός κόμβου, εκτίθενται στα προγράμματα ελέγχου που οργανώνουν την ροή των δεδομένων ώστε να δημιουργήσουν τα συνηθισμένα πρωτοκολλά και στοιβές επεξεργασίας σήματος. Το ενδιάμεσο λογισμικό πρέπει επίσης να προάγει την δημιουργία καταναμημένων προδραστικών αλγορίθμων που βασίζονται στη γλώσσα προγραμματισμού που περιγράφηκε παραπάνω. Όταν λέμε προδραστικό αλγόριθμο εννοούμε ένα αυτόνομο και καταναμημένο αλγόριθμο. Γι' αυτό το λόγο τα σενάρια (scripts) γίνονται «κινητά» χρησιμοποιώντας ειδικές εντολές και οδηγίες. Ένα τέτοιο script μπορεί να αναπαραχθεί (replicate) ή να μεταναστεύσει (migrate) τον κώδικά και τα δεδομένα του σε άλλους κόμβους, επηρεάζοντας άμεσα την συμπεριφορά τους. Αυτή τη διαδικασία θα την ονομάζουμε «αναπαραγωγή» (“population”).

Πέρα από τις προσεγγίσεις που αναφέρθηκαν πιο πάνω, υπάρχουν και άλλες παρόμοιες εκδοχές ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων. Για παράδειγμα, το Impala [15] εκμεταλλεύεται τις τεχνικές κινητού κώδικα, προκειμένου να αλλάζει τη λειτουργικότητα του ενδιάμεσου λογισμικού που εκτελείται σε κάποιο απομακρυσμένο αισθητήρα. Το MiLAN [16] είναι μια ακόμη προσέγγιση ενδιάμεσου λογισμικού που επιτρέπει τη δυναμική ρύθμιση του δικτύου (π.χ. εύρεση και οργάνωση

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) των πόρων του δικτύου) προκειμένου να εξασφαλίσει εγγύηση καλής ποιότητας υπηρεσιών στις εφαρμογές.

4.6 ΠΡΟΣΕΓΓΙΣΗ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

Σύμφωνα με αυτήν την προσέγγιση, χειριζόμαστε τα δίκτυα αισθητήρων σαν μια κατακεντρωμένη βάση δεδομένων, στην οποία οι χρήστες μπορούν να θέτουν ερωτήματα τύπου SQL, προκειμένου το δίκτυο να εκτελέσει κάποια συγκεκριμένη λειτουργία. Τέτοιου τύπου ενδιάμεσο λογισμικό είναι το TinyDB [3] και το SINA [6] (*Sensor Information Networking Architecture and Applications*). Αυτές οι προσεγγίσεις χρησιμοποιούν κατακεντρωμένο έλεγχο και προσφέρουν επεξεργασία των δεδομένων από τους αισθητήρες προτού μεταδοθούν, επιτυγχάνοντας μείωση του όγκου

δεδομένων προς μετάδοση. Με απλά λόγια, αυτό σημαίνει εξοικονόμηση ενέργειας, αφού η επεξεργασία καταναλώνει ελάχιστη ενέργεια σε σχέση με την επικοινωνία.

4.6.1 TinyDB

Το TinyDB, όπως και το Cougar αντιμετωπίζουν το δίκτυο σαν μια κατακεντρωμένη βάση δεδομένων. Θεωρούν την ύπαρξη ενός εικονικού πίνακα (Sensors) μιας βάσης δεδομένων, η κάθε στήλη του οποίου είναι και ένας συγκεκριμένος υποστηριζόμενος τύπος (π.χ., Θερμοκρασία, Υγρασία). Οι τιμές κάθε αισθητήρα καταγράφονται σε μια γραμμή του πίνακα αυτού. Οι χρήστες επιλέγουν τα δεδομένα που επιθυμούν μέσα από συγκεκριμένες επερωτήσεις, παρόμοιες με αυτές της SQL (για την ακρίβεια είναι ένα υποσύνολό τους με κάποιους περιορισμούς, (π.χ. δεν υποστηρίζεται το λογικό “or”) και κάποιες επεκτάσεις που επιτρέπουν σε μια επερώτηση να είναι περιοδικά επαναλαμβανόμενη). Και το TinyDB και το Cougar χρησιμοποιούν μια αποκεντρωμένη προσέγγιση, όπου ο κάθε κόμβος έχει το δικό του επεξεργαστή επερωτήσεων που επεξεργάζεται και αποστέλλει τα δεδομένα που συλλέγει κατευθείαν στο χρήστη. Παρόλο που το TinyDB θεωρείται μια “de-facto” λύση, η SQL-like φύση των επερωτήσεων που δέχεται δεν είναι πάντοτε ευέλικτη. Επίσης δεν υποστηρίζει τον “on-

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) the-fly” ορισμό συμβάντων, αλλά μόνο ορισμένα από πριν συμβάντα τα οποία έχουν προγραμματιστεί στο υλικό του αισθητήρα κατά τη διάρκεια της μεταγλώττισης.

4.6.2 SINA

Το SINA είναι ένα ενδιάμεσο λογισμικό που επιτρέπει σε εφαρμογές των αισθητήρων να πραγματοποιούν επερωτήσεις και εργασίες, να συλλέγουν απαντήσεις και αποτελέσματα και να παρακολουθούν αλλαγές μέσα στο δίκτυο. Τα βασικά χαρακτηριστικά του περιλαμβάνουν την ιεραρχική ομαδοποίηση των κόμβων των αισθητήρων, την ονοματοδοσία βάσει μεταβλητών των κόμβων, καθώς και ένα παράδειγμα οργάνωσης των δεδομένων στους κόμβους. Το SINA χρησιμοποιεί SQL-Like επερωτήσεις καθώς και SQTL (Sensor Query and Tasking Language) διαδικαστικά scripts. Οι SQL-Like επερωτήσεις χρησιμοποιούν τα προαναφερθέντα χαρακτηριστικά

για να εκτελέσουν απλές εργασίες επερωτήσεων και παρακολούθησης, ενώ για πιο προχωρημένες εφαρμογές η SQTL παίζει το ρόλο της διεπαφής μεταξύ των αισθητήρων και του SINA [6]. Ένα SQTL μήνυμα που περιέχει ένα script, βρίσκεται μέσα σε ένα XML-Like SQTL Wrapper και προορίζεται ώστε να διαβαστεί και να εκτελεστεί από ένα περιβάλλον εκτέλεσης των αισθητήρων (Sensor Execution Environment - SEE), που τρέχει σε κάθε αισθητήρα. Εν αντιθέσει με το TinyDB [3] και το Cougar [5], επειδή παρέχει μια εναλλακτική scripting διεπαφή, θεωρείται πιο ευέλικτη προσέγγιση. Παρόλα αυτά ο ουσιαστικός προγραμματισμός των εργασιών μπορεί να αποδειχθεί αρκετά δύσκολος.

4.7 ΠΡΟΣΕΓΓΙΣΗ ΒΑΣΙΣΜΕΝΗ ΣΤΗΝ ΕΝΝΟΙΑ ΤΩΝ ΣΥΜΒΑΝΤΩΝ (EVENTS)

Μια άλλη προσέγγιση στο ενδιάμεσο λογισμικό για δίκτυα αισθητήρων βασίζεται στην έννοια των συμβάντων. Σύμφωνα με αυτή, η εφαρμογή δηλώνει το ενδιαφέρον της για συγκεκριμένες αλλαγές της κατάστασης των παρακολουθούμενων γεγονότων (βασικά συμβάντα) στον πραγματικό κόσμο. Μόλις ένα τέτοιο συμβάν ανιχνευθεί, ο αισθητήριος κόμβος, που το ανίχνευσε, στέλνει μια ειδοποίηση (event notification) προς την εφαρμογή που έχει εκδηλώσει το ανάλογο ενδιαφέρον. Μπορεί να καθοριστεί από την

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) εφαρμογή και ένα συγκεκριμένο σχήμα συμβάντων (pattern of events), έτσι ώστε να έχουμε ειδοποίηση μόνο αν λάβει χώρα η συγκεκριμένη σειρά γεγονότων.

Το DSWare[10] είναι μια λύση που βασίζεται αποκλειστικά στην ιδέα των συμβάντων.

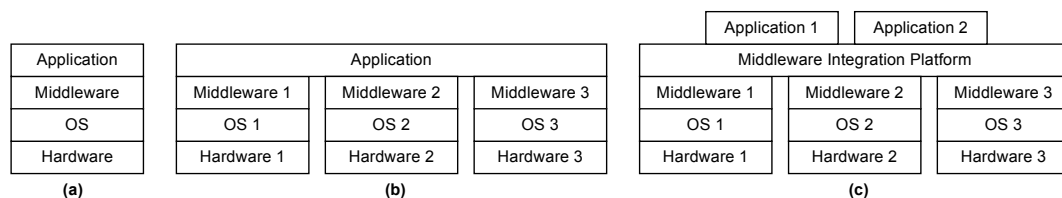
ΚΕΦΑΛΑΙΟ 5

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ ΕΝΔΙΑΜΕΣΟΥ ΛΟΓΙΣΜΙΚΟΥ

Όλα τα συστήματα ενδιάμεσου λογισμικού (δες Κεφ. 4) ακολουθούν διαφορετικές προγραμματιστικές προσεγγίσεις, π.χ. database προσέγγιση, agent-based προσέγγιση και event-based προσέγγιση, και διαφέρουν μεταξύ τους όσον αφορά στην ευκολία χρήσης τους, την εκφραστικότητα, την επεκτασιμότητα και το overhead. Αυτό που αξίζει να τονιστεί είναι πως αυτός που αναπτύσσει μία εφαρμογή είναι στενά εξαρτημένος από την προσέγγιση που χρησιμοποιεί το ενδιάμεσο λογισμικό(middleware) και ως αποτέλεσμα πρέπει να γνωρίζει τα συγκεκριμένα χαρακτηριστικά, τις δυνατότητες και τους περιορισμούς της, ενώ πρέπει παράλληλα να γνωρίζει τις συγκεκριμένες προγραμματιστικές διεπαφές.

Προκειμένου η ανάπτυξη εφαρμογών βασισμένων στα ασύρματα δίκτυα αισθητήρων να γίνει πραγματικότητα, πρέπει αυτός που αναπτύσσει τις εφαρμογές να μην χρειάζεται να γνωρίζει τις τεχνολογίες ενδιάμεσου λογισμικού που χρησιμοποιούνται καθώς και τα χαρακτηριστικά τους. Προκειμένου να επιτευχθεί αυτός ο στόχος, είναι απαραίτητο να ακολουθηθεί μία προσέγγιση σύμφωνα με την αρχιτεκτονική του σχήματος 5.1(c). Εν

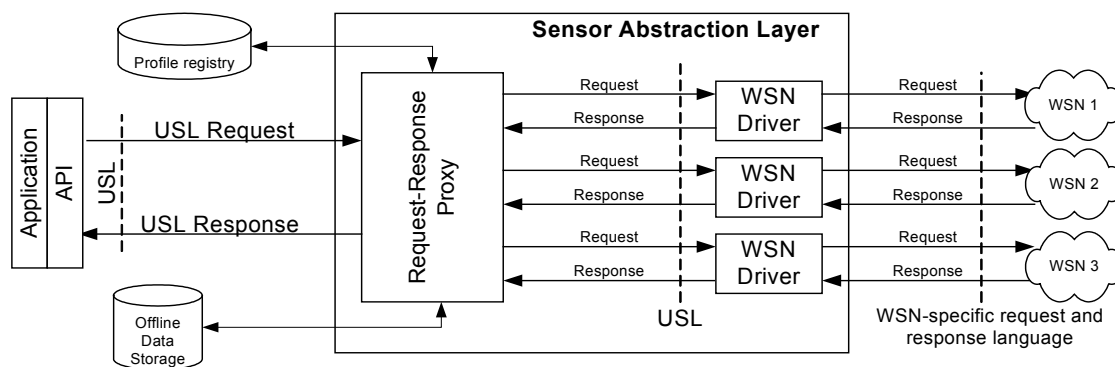
Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)
 αντιθέσει με τις προσεγγίσεις που φαίνονται στα σχήματα 5.1(a) και 5.1(b), οι εφαρμογές δεν γνωρίζουν τις χρησιμοποιούμενες τεχνολογίες και έχουν πρόσβαση στα δεδομένα των αισθητήρων με μια ενιαία λογική.



Σχήμα 5.1: Σενάρια χρήσης των WSN. (a) Μια απλή εφαρμογή λειτουργεί για ένα συγκεκριμένο WSN (WSN-specific application). Οι περισσότερες σημερινές εφαρμογές ακολουθούν αυτήν την προσέγγιση. (b) Μια απλή εφαρμογή χρησιμοποιεί διάφορα WSNs. Η εφαρμογή πρέπει να είναι ικανή να κρύβει την υποκείμενη ετερογένεια. (c) Διαφορετικές εφαρμογές χρησιμοποιούν διάφορα WSN's με ενοποιημένο τρόπο, με την χρήση μιας πλατφόρμας ολοκλήρωσης ενδιάμεσου λογισμικού. Οι εφαρμογές δεν γνωρίζουν τις χρησιμοποιούμενες τεχνολογίες στα υποκείμενα δίκτυα.

5.1 ΓΕΝΙΚΗ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΗΣ ΠΛΑΤΦΟΡΜΑΣ

Σε αυτήν την ενότητα περιγράφεται πως οι στόχοι που αναφέρθηκαν προηγουμένως μπορούν να πραγματοποιηθούν. Προκειμένου να καλυφθεί η ετερογένεια μεταξύ των διαφόρων υλοποιήσεων των δικτύων αισθητήρων, πρέπει να παρεμβληθεί ένα ακόμα επίπεδο το οποίο θα διευκολύνει την επικοινωνία εφαρμογών συλλογής δεδομένων με το δίκτυο αισθητήρων.



Σχήμα 5.2: Η αρχιτεκτονική της πλατφόρμας [1]

Αυτό το επίπεδο θα χρησιμεύει σαν ένα επίπεδο αφαίρεσης, κρύβοντας τις τεχνικές λεπτομέρειες της υλοποίησης του δικτύου από τις εφαρμογές που θα το χρησιμοποιούν. Αιτήσεις και απαντήσεις από διάφορα ετερογενή δίκτυα θα χειρίζονται με ενιαίο τρόπο, με τη βοήθεια μιας καλά ορισμένης γλώσσας. Τη γλώσσα αυτή την αποκαλούμε USL (Unified Sensor Language). Οι εφαρμογές θα πρέπει να γνωρίζουν τη γλώσσα αυτή και να είναι συμβατές με αυτή. Οι αιτήσεις που θα φτάνουν στο Sensor Abstraction Layer (SAL), θα αναλύονται, θα μεταφράζονται σε συγκεκριμένες WSN γλώσσες και θα προωθούνται στα αντίστοιχα υποκείμενα δίκτυα. Αντιστρόφως, οι απαντήσεις, έχοντας μια συγκεκριμένη δομή, θα συλλέγονται, θα μεταφράζονται σε USL και στη συνέχεια θα προωθούνται σε αυτόν που έκανε την αίτηση. Το SAL, λοιπόν, μπορεί να θεωρηθεί ο κορμός όλου του συστήματος. Επιλέγουμε την XML γλώσσα για την αναπαράσταση της USL, καθώς είναι μια τυποποιημένη και ευρέως χρησιμοποιούμενη μεταγλώσσα, ικανή να περιγράφει δεδομένα με δομημένο τρόπο. Επιπλέον ο αυτοματοποιημένος

μετασχηματισμός και η επεξεργασία της σύνταξης που χρησιμοποιεί η XML προσδίδουν στην USL τα χαρακτηριστικά που επιθυμούμε.

Αναφορικά με το στόχο της παροχής στους προγραμματιστές ενός υψηλού επιπέδου προγραμματιστικού μοντέλου, πρέπει να χρησιμοποιηθεί μια ενιαία προγραμματιστική διεπαφή, όπως δείχνει και το σχήμα 5.2. Αυτό θα αποτελεί και το API (Application Programming Interface) που θα χρησιμοποιούν οι χρήστες, έχοντας έτσι μια πληθώρα εκφραστικών και ευέλικτων εντολών, προκειμένου να πραγματοποιούν αιτήσεις προς το δίκτυο. Αυτό το API θα αντιστοιχεί σε διαφορετικά επίπεδα αφαίρεσης του περιβάλλοντος που θα μπορούσε ο προγραμματιστής να χρησιμοποιήσει βασιζόμενος στις συγκεκριμένες ανάγκες του (π.χ. μια χαμηλού επιπέδου έκδοση του API θα προσέδιδε στον προγραμματιστή μεγάλη ευελιξία, αλλά θα απαιτούσε πιο προσεκτικό προγραμματισμό, ενώ μια υψηλού επιπέδου έκδοσή του, μπορεί μεν να είχε περιορισμένη ευελιξία, που θα επέτρεπε δε ευκολότερο προγραμματισμό εφαρμογών).

Η σχεδίασή του API βασίζεται σε δύο οντότητες: 1) περιοχές (locations) και 2) συσκευές (devices). Ο προγραμματιστής αντιλαμβάνεται το περιβάλλον σαν έναν συνδυασμό από διαφορετικές περιοχές και συσκευές και βάσει αυτών κατευθύνει τις επερωτήσεις του προς το δίκτυο. Αντιθέτως με τις περιοχές, οι συσκευές δεν θεωρούνται στάσιμες, αλλά

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) μπορούν να είναι φορητές (π.χ. PDA) έχοντας παράλληλα δυνατότητες αίσθησης. Άλλες οντότητες, όπως άνθρωποι, ρομπότ, αυτοκίνητα κ.α. μπορούν να μεταφέρουν αυτές τις συσκευές. Με αυτές τις δυνατότητες, οι εφαρμογές μπορούν να ζητούν πληροφορίες από τους αισθητήρες που βρίσκονται σε συγκεκριμένες περιοχές, ή που σχετίζονται με συγκεκριμένες οντότητες.

Όπως φαίνεται στο σχήμα 5.2, το σύστημα υποστηρίζεται επίσης από τα «Profile Registry» και «Offline - Data Storage». Το πρώτο αποτελείται από διάφορους πίνακες βάσης δεδομένων και αρχεία ρυθμίσεων και κρατάει πληροφορίες για τις ρυθμίσεις, τις δυνατότητες υποστήριξης και τα χαρακτηριστικά του συγκεκριμένου δικτύου αισθητήρων με το οποίο σχετίζεται (π.χ. τους τύπους δεδομένων που υποστηρίζουν οι αισθητήρες, όπως θερμοκρασία, υγρασία κ.α., διάφορες μονάδες μέτρησης όπως Celsius, Fahrenheit, Volt κ.α., βασικές συναρτήσεις που μετασχηματίζουν ακατέργαστα δεδομένα από τους αισθητήρες σε πιο χρήσιμες μορφές όπως μετασχηματισμός από Volt σε θερμοκρασία καθώς και πληροφορίες σχετικά με συγκεκριμένες WSN διεπαφές).

Όλη αυτή η πληροφορία είναι χρήσιμη και βοηθάει στο να επιλεγούν τα WSN που ικανοποιούν τα χαρακτηριστικά συγκεκριμένων αιτημάτων. Επιπλέον το Profile Registry αποθηκεύει πληροφορίες σχετικά με τις φορητές συσκευές που αναφέρθηκαν παραπάνω. Όταν υπάρχει κάποια αλλαγή στο σύστημα, ο διαχειριστής του συστήματος πρέπει να ενημερώνει τις ρυθμίσεις του Profile Registry κατάλληλα.

Το ODS βασίζεται σε ένα κοινό σύστημα βάσεων δεδομένων (π.χ. μια σχεσιακή βάση) και είναι υπεύθυνο για την συγκέντρωση και την αποθήκευση όλων των απαντήσεων που περνούν από το SAL, προκειμένου να δώσουν τη ευκαιρία στους χρήστες που ενδιαφέρονται για τα ιστορικά στοιχεία, καθώς και στατιστικές ιδιότητες του φαινομένου που παρακολουθούν να αποκτήσουν αυτήν τη δυνατότητα. Η χρήση του ODS θα επιτρέψει την εφαρμογή πολύπλοκης επεξεργασίας πληροφορίας (π.χ. post-processing για εξόρυξη δεδομένων και caching) που αποτελεί ενεργό πεδίο έρευνας αυτήν την εποχή.

5.2 Η UNIFIED SENSOR LANGUAGE (USL)

Το SAL, όπως περιγράφηκε στη προηγούμενη ενότητα, βασίζεται στην USL, μια γενικής χρήσης γλώσσα για την επικοινωνία με τα δίκτυα αισθητήρων. Αυτή η γλώσσα έχει οριστεί, μετά από εκτεταμένη ανάλυση απαιτήσεων (απαιτήσεις που αφορούν τόσο στην αλληλεπίδραση μεταξύ του προγραμματιστή με το περιβάλλον των αισθητήρων, όσο και στην λειτουργικότητα των διαθέσιμων ενδιάμεσων λογισμικών για ασύρματα δίκτυα αισθητήρων). Η ανάλυση αυτή οδήγησε στα ακόλουθα απαραίτητα χαρακτηριστικά της γλώσσας προκειμένου να ικανοποιούνται οι απαιτήσεις των χρηστών:

1. Υποστήριξη για *σύγχρονες επερωτήσεις* που παίρνουν τα επιθυμητά δεδομένα, είτε από το WSN, είτε από το ODS και επιστρέφουν τις απαντήσεις σε πραγματικό χρόνο.
2. Υποστήριξη για *προγραμματισμό βάσει συμβάντων*. Πολλές εφαρμογές επίγνωσης πλαισίου (context-aware) πρέπει να πυροδοτούν διάφορες ενέργειες,

όταν κάποια συμβάντα λάβουν χώρα. Οι χρήστες πρέπει να μπορούν να εγκαθιστούν listeners και χειριστές (handlers) τέτοιων συμβάντων, π.χ. σε περίπτωση που η θερμοκρασία είναι αρκετά υψηλή (π.χ. μεγαλύτερη από 42 βαθμούς Κελσίου) μέσα σε ένα δωμάτιο και η φωτεινότητα είναι επίσης αυξημένη, τότε θα πρέπει να σημάνει συναγερμός για φωτιά.

3. Υποστήριξη για *περιοδική παρακολούθηση* των τιμών των αισθητήρων π.χ. μέτρηση της θερμοκρασίας κάθε 5 λεπτά, ξεκινώντας από τη Δευτέρα 2 Ιουλίου στις 15:00 μ.μ. και σταματώντας ακριβώς μια βδομάδα μετά.
4. Εύκολη και δυναμική αλλαγή των τύπων των μεγεθών που μετρούν οι αισθητήρες, καθώς και των συναρτήσεων που υποστηρίζουν. Καθώς νέοι αισθητήρες μπορεί να χρησιμοποιηθούν, ο προγραμματιστής πρέπει να έχει τη δυνατότητα να συμπεριλάβει τη νέα αυτή λειτουργικότητά στη λογική του προγράμματός του. Επιπλέον, σε περίπτωση σφάλματος στους αισθητήρες, κάτι που συμβαίνει αρκετά συχνά, πρέπει να υπάρχει ειδικός χειρισμός του λάθους που να συνοδεύεται με σαφή περιγραφή.

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

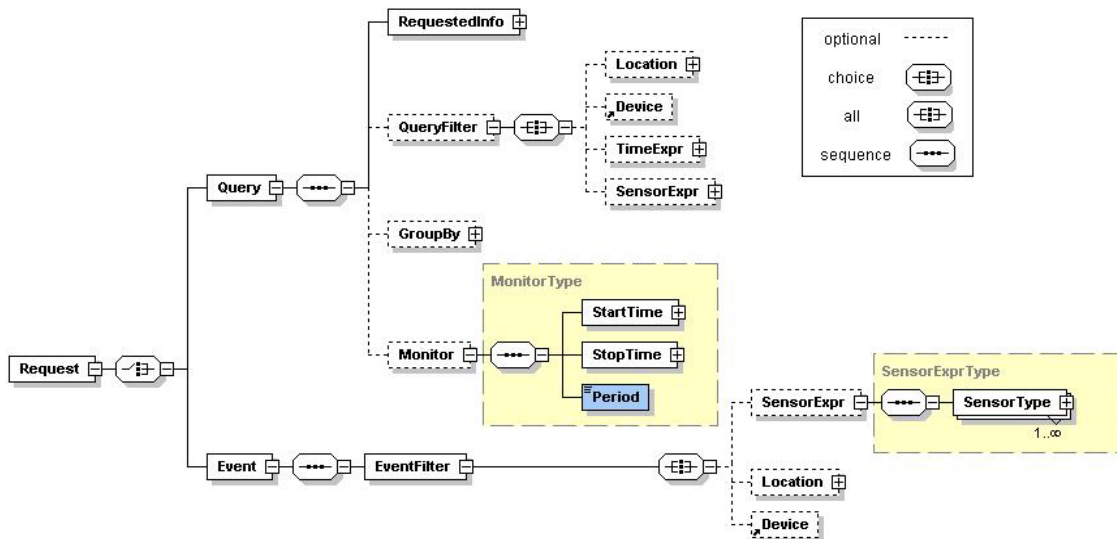
Το αποτέλεσμα από την ανάλυση απαιτήσεων είναι ο προσδιορισμός της USL γλώσσας που βασίζεται σε XML.

Δεδομένης της πρόσφατης απήχησης που έχει η database-like επεξεργασία πληροφορίας για δεδομένα δικτύων αισθητήρων, κάποιος θα μπορούσε να υποστηρίξει πως μια γλώσσα βασισμένη σε SQL θα ήταν πιο κατάλληλη. Παρόλα αυτά, η σχεσιακή φύση των γλωσσών αυτών, δημιουργεί κάποιους περιορισμούς, αναφορικά με τον γενικό χειρισμό των WSN υποδομών, καθώς εξαρτάται από τη μορφή των σχεσιακών επερωτήσεων (select - from - where) και δεν υποστηρίζει τη δηλωτική (declarative) και βασισμένη σε συμβάντα (event-driven) λογική που θέλουμε να παρέχεται στο χρήστη. Παράλληλα, η USL μπορεί να βελτιωθεί, με την πρόσθεση διαχειριστικής λειτουργικότητας, κάτι που θα καθυστερούσε πάρα πολύ με τη χρήση μιας γραμματικής βασισμένης σε SQL.

Η USL ορίζει δύο βασικές οντότητες, την αίτηση και την απάντηση. Αφηρημένες εκδοχές των XML Schemas αυτών των δύο οντοτήτων περιγράφονται στις επόμενες ενότητες.

5.3 Η ΑΙΤΗΣΗ (USL- REQUEST)

Η USL αίτηση αντιπροσωπεύει τις αιτήσεις (σύγχρονες, event-driven και περιοδικές) του χρήστη προς τα δίκτυα αισθητήρων ή το ODS. Το στοιχείο "Request" που βρίσκεται στη ρίζα του δένδρου περιέχει μια μοναδική μεταβλητή "ID". Αυτή η μεταβλητή προσδιορίζει μοναδικά ένα αίτημα του χρήστη.



Σχήμα 5.3: Η USL-REQUEST Αίτηση [1]

Στη περίπτωση της σύγχρονης επερώτησης, ο χρήστης προσδιορίζει τις πληροφορίες για τις οποίες ενδιαφέρεται, μέσα στο στοιχείο “ RequestedInfo ”. Αυτή η πληροφορία μπορεί να περιλαμβάνει το ενδιαφέρον για μια ανάγνωση από έναν ή περισσότερους αισθητήρες, περιορισμούς στην τοποθεσία, τη χρονική στιγμή ή τη διάρκεια ενός συγκεκριμένου φαινομένου, το ID της συσκευής που ικανοποιεί κάποια κριτήρια, ή ένα συνδυασμό των παραπάνω. Φυσικά δεν είναι όλοι οι συνδυασμοί έγκυροι, γι’ αυτό πρέπει να διενεργείται σημασιολογικός έλεγχος στις αιτήσεις. Οι περιορισμοί του “Request” περιγράφονται στο στοιχείο “QueryFilter”. Σε αυτό το στοιχείο, κάποιος μπορεί να δηλώσει συνθήκες χρόνου (στοιχείο “TimeExpr”), συνθήκες σχετικά με τα μετρούμενα από τους αισθητήρες μεγέθη (στοιχείο “SensorExpr”), καθώς και να περιορίσει την επερώτηση σε μια συγκεκριμένη περιοχή ή σε μια συγκεκριμένη συσκευή

(στοιχεία “Location” και “Device” αντίστοιχα). Το “SensorExprType” (βλέπε σχήμα 5.3), είναι η XML αναπαράσταση της παρακάτω BNF γραμματικής [9]:

Πίνακας 5.1: Το SensorExpression κομμάτι της αίτησης

SensorExpr = [Function,] SensorType, Conditional, Value;


```
Function = 'tempAverage' | 'tempMinimum' | 'tempCount' | 'tempMaximum' |  
'spatialAverage' | 'spatialSum';  
SensorType = 'Temperature' | 'Humidity' | 'Acceleration';  
Conditional = 'Greater' | 'Less' | 'Equals' | 'WithinRange';  
Value = Alphanumeric | RealNumber | Integer;
```

Φυσικά οι ορισμοί των “Function” και “SensorType” δεν είναι ολοκληρωμένοι. Οι δυνατές τιμές τους προσδιορίζονται σε ένα ξεχωριστό XML Schema. Το στοιχείο “TimeExpr” έχει παρόμοια σύνταξη.

Επιπλέον υπάρχει το στοιχείο “GroupBy”, το οποίο μπορεί να ταξινομή τις απαντήσεις με τρόπο αντίστοιχο με το γνωστό GroupBy της SQL, ενώ η σύνταξη του είναι παρόμοια με του “RequestedInfo”. Το στοιχείο “Monitor”, στη περίπτωση που υπάρχει, δηλώνει πως η επερώτηση θα πρέπει να εκτελεστεί περιοδικά όπως περιγράφεται από τα στοιχεία “StartTime”, “StopTime” και “Period”. Από εδώ και πέρα, θα αναφερόμαστε στις περιοδικές επερωτήσεις χρησιμοποιώντας τον όρο monitors.

Από την άλλη, στη περίπτωση της χρήσης event-driven προγραμματισμού, το μόνο υποστοιχείο του Request, είναι το Event. Αυτό το στοιχείο περιλαμβάνει μια σειρά από συνθήκες (στοιχείο EventFilter), που όταν ικανοποιούνται, πυροδοτούν κάποια events στα ανώτερα στρώματα του ενδιάμεσου λογισμικού. Αυτά τα ανώτερα επίπεδα, έχουν εγκαταστήσει από πριν listeners για αυτά τα events, και (μέσω της οντότητας USL Response), διενεργούν κάποιες προσδιορισμένες από πριν ενέργειες.

Το στοιχείο “EventFilter” μοιάζει πάρα πολύ με το “QueryFilter”, εκτός από το γεγονός ότι δεν περιέχει καθόλου συνθήκες χρόνου. Αυτό είναι λογικό για αυτήν την πρώτη έκδοση της USL, καθώς συμβάντα που γενικεύονται από συνθήκες χρόνου, μπορούν να εφαρμοστούν μόνο με στόχο το ODS (πάνω σε ιστορικά δεδομένα) και

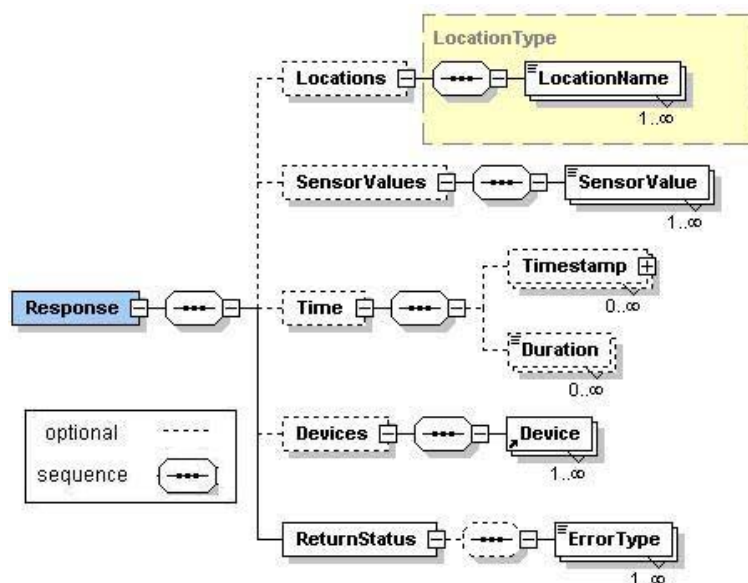
περιλαμβάνουν πολύπλοκες τεχνικές επεξεργασίας πληροφορίας. Καθώς υπάρχει μεγάλη ερευνητική δραστηριότητα σε αυτές τις περιοχές, μια μελλοντική έκδοση της USL, θα μπορούσε να συμπεριλαμβάνει συμβάντα βασισμένα στο χρόνο (π.χ. σε περίπτωση που η θερμοκρασία αλλάζει σε ένα δωμάτιο με υπολογιστές με ρυθμό +3 βαθμούς Κελσίου την ώρα, ενημέρωσε τον θυρωρό προκειμένου να ενεργοποιήσει τον κλιματισμό).

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

Το USL request, εκτός από το να παρέχει τη δυνατότητα της εγκατάστασης event listeners και τη περιγραφή των επερωτήσεων, υποστηρίζει επίσης την κατάργηση ήδη εγκατεστημένων event listeners και monitors. Γι' αυτό το λόγο, τα στοιχεία "Event" και "Monitor", περιλαμβάνουν Boolean μεταβλητές abort. Όταν ένας χρήστης καταργεί ένα event ή ένα monitor, το γνωστό του ID περνάει στη μεταβλητή ID του Request στοιχείου, και η συγκεκριμένη μεταβλητή abort τίθεται σε true, ενώ όλα τα υπόλοιπα στοιχεία είναι απόντα ή κενά.

5.4 Η Απάντηση (USL-RESPONSE)

Η USL απάντηση είναι αρκετά πιο απλή από τη USL αίτηση. Το στοιχείο που βρίσκεται στη ρίζα ("Response"), περιλαμβάνει επίσης μια μεταβλητή "ID" και πάντα περιέχει το στοιχείο "ReturnStatus". Η τιμή του ID ταυτίζεται με την τιμή του ID της αντίστοιχης USL αίτησης.



Σχήμα 5.3: Η USL Απάντηση (RESPONSE) [1]

Αν η μεταβλητή "error" αυτού του στοιχείου πάρει τη τιμή true, τότε κάποιο λάθος συνέβη μέσα στο SAL ή το δίκτυο αισθητήρων και έτσι ο τύπος του (στοιχείο "ErrorType") επιστρέφεται στο API προκειμένου να δημιουργηθεί κάποιο exception (εξάιρεση) στην εφαρμογή. Σε περίπτωση που η αίτηση δεν περιέχει κάποιο σφάλμα, τα

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

απαιτούμενα δεδομένα (στο RequestedInfo) επιστρέφονται στην οντότητα που ζήτησε τα δεδομένα. Εναλλακτικά, σε περίπτωση που η αίτηση εγκατέστησε κάποιο event, τότε όλα τα στοιχεία εκτός από το "ReturnStatus" είναι απόντα (η απάντηση είναι ισοδύναμη με ένα flag που δείχνει αν το event έλαβε χώρα).

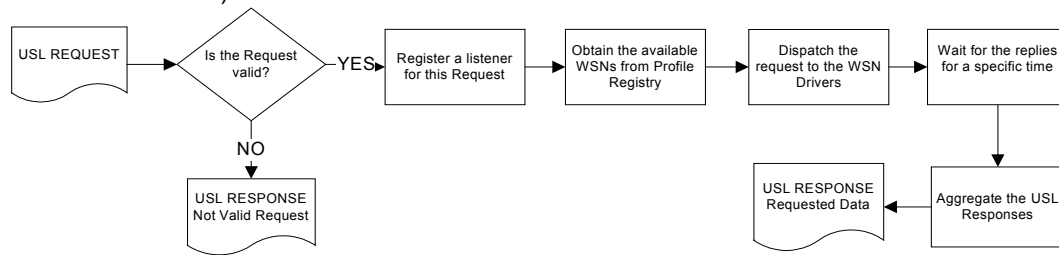
Κάποιες από τις παραμέτρους στα προαναφερθέντα USL στοιχεία και μεταβλητές, μπορεί να διαφέρουν αναλόγως την περίπτωση. Αυτοί είναι οι τύποι των αισθητήρων (π.χ. αισθητήρες θερμοκρασίας, φωτεινότητας, κτλ), οι συναρτήσεις που μπορούν να εφαρμοστούν σε αυτούς (π.χ. TempAverage, δηλαδή Temporal Average), και οι τύποι ένδειξης λάθους (π.χ. TIMEOUT). Όλα αυτά απαριθμούνται σε ένα ξεχωριστό XML Schema στο οποίο αναφέρονται τα Request και Response Schema, με σκοπό να εφαρμόσουν κάποιους περιορισμούς κατά τη διάρκεια του XML validation των USL αιτήσεων και απαντήσεων από τον RR Proxy(παραγρ. 5.6). Αυτό το ξεχωριστό XML Schema μπορεί να θεωρηθεί σαν ένα κομμάτι του Profile Registry, επειδή ενημερώνεται κάθε φορά που αλλάζουν οι ρυθμίσεις των υποκείμενων δικτύων αισθητήρων.

Ο σχηματισμός του USL request μπορεί να πραγματοποιηθεί από το API. Αντιστρόφως, η USL απάντηση γεμίζει με τα δεδομένα που συλλέχθηκαν από το δίκτυο ή από την ένδειξη κάποιου λάθους και τελικώς επιστρέφεται στο χρήστη.

5.5 To Sensor Abstraction Layer (SAL)

Όπως αναφέρθηκε και προηγουμένως, το SAL είναι το πιο σημαντικό κομμάτι του συστήματος και αποτελείται από τις ακόλουθες λειτουργικές οντότητες:

- Τον RR Proxy (Request – Response Proxy)
- Το WSN Driver (έναν ή περισσότερους, αναλόγως με το πλήθος των διαφορετικών δικτύων αισθητήρων που είναι διαθέσιμα).



Σχήμα 5.4: Η προώθηση της αίτησης προς τον RR Proxy [1]

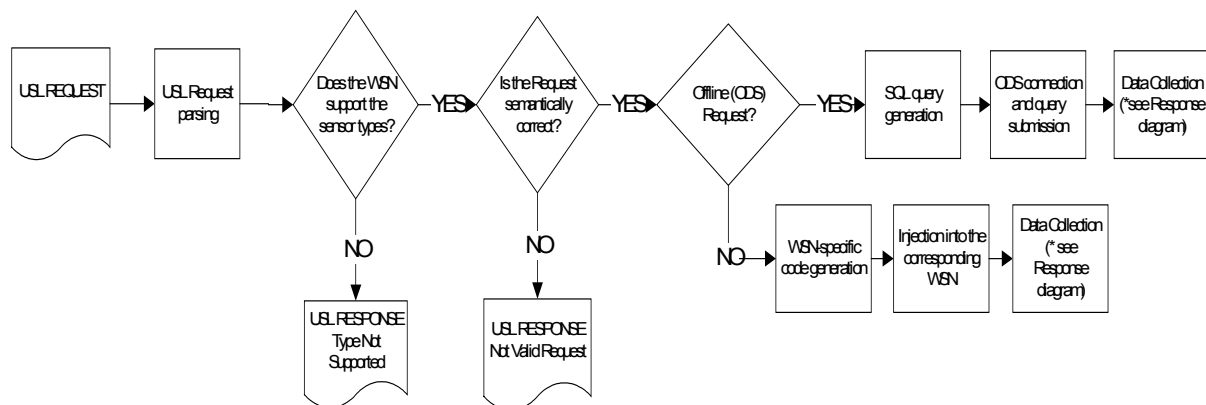
5.6 RR Proxy (Request – Response Proxy)

Ο RR Proxy, όπως φαίνεται και στο σχήμα 5.4, λαμβάνει τις USL αιτήσεις και διενεργεί μια πρώτου επιπέδου επεξεργασία, που περιλαμβάνει τα ακόλουθα βήματα:

- Αρχικά ελέγχει αν η αίτηση είναι σωστή γραμματικά (αν το XML document που χρησιμοποιείται για την αίτηση είναι έγκυρο αναφορικά με το XML Schema που χρησιμοποιείται. Αν ο έλεγχος ορθότητας είναι θετικός, η επεξεργασία της αίτησης συνεχίζει, αλλιώς ένα μήνυμα λάθους (Not Valid Request) επιστρέφεται στον χρήστη.
- Μετά από ένα πετυχημένο έλεγχο συντακτικής ορθότητας, ο RR Proxy, συλλέγει πληροφορίες για τα διαθέσιμα δίκτυα αισθητήρων από το Profile Registry. Αν τα διαθέσιμα δίκτυα αισθητήρων είναι περισσότερα από ένα, τότε ο RR Proxy πρέπει να διανείμει την αίτηση στους αντίστοιχους WSN Drivers. Επίσης ο RR Proxy εγκαθιστά έναν listener για την αίτηση, το οποίο σχετίζεται με ένα μοναδικό request ID που ανατέθηκε στην αίτηση από το API του χρήστη. Επίσης διαθέτει έναν σχετιζόμενο Timer που λήγει μετά από μια συγκεκριμένη χρονική περίοδο. Ο listener συγκεντρώνει τις απαντήσεις που σχετίζονται με ένα συγκεκριμένο request ID και στη συνέχεια επιστρέφει τα δεδομένα στον χρήστη. Σε περίπτωση που δεν επιστραφεί τίποτα από τα υποκείμενα δίκτυα αισθητήρων, μέχρι να λήξει ένα συγκεκριμένο χρονικό διάστημα, ο listener επιστρέφει στον χρήστη ένα μήνυμα λάθους (Request Timeout).

5.7 WSN Driver

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)
 Όπως φαίνεται στο σχήμα 5.5, ο WSN Driver επεξεργάζεται αιτήσεις που λαμβάνει από τον RR Proxy, ακολουθώντας τα παρακάτω βήματα:



Σχήμα 5.5: Η προώθηση της αίτησης προς το δίκτυο αισθητήρων [1]

1. USL Request Parsing

Ο WSN Driver ελέγχει συντακτικά αν τα δεδομένα που περιέχονται στη USL αίτηση είναι σωστά δομημένα και έγκυρα.

2. Έλεγχος Υποστήριξης αιτούμενων στοιχείων

Στην αίτηση περιέχονται διάφορες πληροφορίες που περιλαμβάνουν τους τύπους, τις συναρτήσεις και τις περιοχές που υποστηρίζονται από το δίκτυο με το οποίο συνεργάζεται ο WSN Driver. Αφού γίνει το parsing της USL αίτησης, ο WSN Driver πρέπει να ελέγξει αν το ασύρματο δίκτυο αισθητήρων με το οποίο συνεργάζεται μπορεί να υποστηρίξει την αίτηση. Αρχικά εξάγει μια λίστα με τύπους αισθητήρων (π.χ.

Μελέτη και υλοποίηση συστημάτων ενδιαμέσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) θερμοκρασία) που βρίσκονται στο στοιχείο RequestedInfo. Το ίδιο κάνει με τους τύπους και τις περιοχές που βρίσκονται στο στοιχείο QueryFilter ή στο στοιχείο EventFilter. Στη συνέχεια συμβουλευτείται το Profile Registry Σε περίπτωση που κάποιος από τους τύπους ή τις περιοχές δεν υποστηρίζεται από το δίκτυο, η επεξεργασία τερματίζεται και επιστρέφεται κατάλληλο μήνυμα στον RR Proxy που καταδεικνύει το σφάλμα. Αν υποστηρίζονται όλα τα επιθυμητοί τύποι δεδομένων, τότε η επεξεργασία προχωράει στο στάδιο της σημασιολογικής ανάλυσης.

3. Σημασιολογική Ανάλυση

Σε αυτό το στάδιο, ο WSN Driver πραγματοποιεί σημασιολογικούς ελέγχους στα δεδομένα που περιέχονται στην αίτηση που έλαβε. Σε περίπτωση ένδειξης σημασιολογικού σφάλματος, η επεξεργασία σταματά και αποστέλλεται κατάλληλο μήνυμα στον RR Proxy που καταδεικνύει το σφάλμα. Αν όχι, τότε η επεξεργασία προχωράει στο επόμενο στάδιο (αναλόγως αν θέλουμε offline ή real -time επεξεργασία).

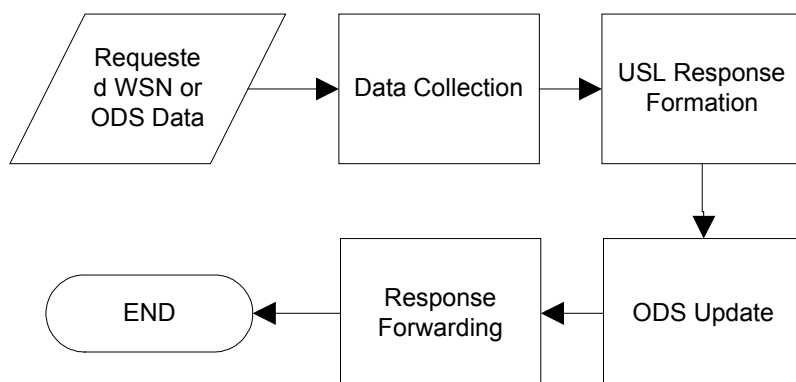
4. Δρομολόγηση της Αίτησης

Αυτό το στάδιο βασίζεται στην απόφαση που πάρθηκε στο αμέσως προηγούμενο σχετικά με το αν επιθυμούμε offline (π.χ. τότε η θερμοκρασία ξεπέρασε τους 50 βαθμούς Κελσίου) ή πραγματικού χρόνου (real - time) επεξεργασία (π.χ., Ποιά είναι η μέση τιμή της υγρασίας από όλους τους αισθητήρες). Στη πρώτη περίπτωση, το request μεταφράζεται σε μια SQL επερώτηση, αλλιώς σε ειδικό κώδικά για WSN.

5. Εισαγωγή της Αίτησης στο Δίκτυο

Αν όλα τα προηγούμενα στάδια έχουν ολοκληρωθεί με επιτυχία, τότε η αίτηση θα παραδοθεί στο υποκείμενο δίκτυο αισθητήρων ή στο ODS και ο WSN Driver θα

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs) περιμένει μέχρι να λάβει απάντηση.Τα βήματα τα οποία ακολουθούνται για το σχηματισμό της απάντησης, όπως φαίνονται και στο σχήμα 5.6, είναι τα ακόλουθα:



Σχήμα 5.6: Η προώθηση της αίτησης προς το δίκτυο αισθητήρων [1]

1. Συλλογή δεδομένων και σχηματισμός της απάντησης

Ο WSN Driver λαμβάνει τα δεδομένα από το δίκτυο ή από τη βάση και στη συνέχεια τα ενσωματώνει σε μια USL απάντηση, σύμφωνα με τη δομή που περιγράφηκε προηγουμένως.

2. Ενημέρωση της βάσης (ODS)

Πριν επιστραφεί η απάντηση στον RR Proxy, πρέπει να ενημερώνεται η βάση (μόνο στη περίπτωση που η αίτηση περιελάμβανε real-time επεξεργασία, που προωθήθηκε δηλαδή στο δίκτυο). Αυτή η εγγραφή στη βάση, θα περιλαμβάνει την απάντηση που επιστράφηκε στον RR Proxy, το ID της αίτησης, καθώς και άλλες πρόσθετες πληροφορίες (π.χ. Timestamp, location, device κ.α.). Με την καταγραφή αυτών των στοιχείων, μπορεί κάποιος αργότερα να πραγματοποιήσει στατιστική ή κάθε άλλου είδους επεξεργασία σε ιστορικά δεδομένα μετρήσεων.

3. Προώθηση της απάντησης

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

Η USL απάντηση προωθείται στον RR Proxy. Αυτό που αξίζει να σημειωθεί, είναι πως η επικοινωνία μεταξύ του RR Proxy και των WSN Drivers μέσα στο SAL, βασίζεται και αυτή στην USL. Αυτό επιτρέπει στο να χειριζόμαστε το SAL σε ένα ανεξάρτητο γλώσσας επίπεδο (language neutral). Επίσης, με την προτεινόμενη αρχιτεκτονική, ο κατασκευαστής του WSN, δεδομένου ενός αντίστοιχου Driver για το WSN του, (ένα συστατικό που να μπορεί να μετατρέπει USL σε κώδικα για WSN και αντίστροφα), θα μπορεί να ενσωματώνει το WSN του στην συγκεκριμένη αρχιτεκτονική και να το κάνει συμβατό με ήδη υπάρχουσες εφαρμογές (υπό αυτή την σκοπιά προέκυψε και ο όρος Driver, παραπέμποντας στους Drivers που χρησιμοποιούνται από εκτυπωτές, περιφερειακά και άλλες συσκευές). Για την πραγματική εφαρμογή της παραπάνω δομής, ο κατασκευαστής του WSN πρέπει να ενημερώσει επίσης το Profile Registry, με πληροφορίες που αφορούν στις δυνατότητες του WSN και συναρτήσεις, για την μετατροπή των ακατέργαστων δεδομένων των αισθητήρων, σε κοινά αποδεκτές μορφές (π.χ. Volt σε βαθμούς Κελσίου).

ΚΕΦΑΛΑΙΟ 6

ΑΝΑΛΥΣΗ ΥΛΟΠΟΙΗΣΗΣ

Σε αυτό το κεφάλαιο θα παρουσιαστεί το κομμάτι της πτυχιακής εργασίας που σχετίζεται με την υλοποίηση. Πρόκειται για το τμήμα RR Proxy [1], κομμάτι της συνολικής αρχιτεκτονικής του συστήματος, που αποτελεί βασικό τμήμα του SAL (Sensor Abstraction Layer) [1].

Μία σύντομη περιγραφή της λειτουργικότητάς του έχει ως εξής:

- Λαμβάνει USL αιτήσεις από το API
- Επεξεργάζεται τις αιτήσεις και ακολούθως τις προωθεί στους κατάλληλους WSN Drivers

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

- Δέχεται τα Responses από τους κατάλληλους WSN Drivers
- Τα επεξεργάζεται και παράγει ένα συνολικό USL Response

Η υλοποίηση πραγματοποιήθηκε στη γλώσσα προγραμματισμού Java και το περιβάλλον υλοποίησης που χρησιμοποιήθηκε είναι ο Jbuilder X.

6.1. ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ

Σε αυτήν την παράγραφο, περιγράφουμε πιο αναλυτικά τη λειτουργικότητα του RR Proxy. Παραλαμβάνει από κάποιο χρήστη-API μία αίτηση η οποία είναι δομημένη σε XML μορφή. Την επεξεργάζεται ώστε να εξάγει κάποια συμπεράσματα σχετικά με το τι ζητάει ο χρήστης- API και αν αυτά που ζητάει έχουν νόημα και υποστηρίζονται από ένα ή περισσότερα υποκείμενα δίκτυα αισθητήρων, την προωθεί στους αντίστοιχους WSN Driver. Κατόπιν συλλέγει τις απαντήσεις από τους WSN Driver, τις επεξεργάζεται και τις συνενώνει-δομεί σε μια ενιαία απάντηση η οποία και προωθείται ολοκληρωμένη πλέον προς το χρήστη- API.

Πιο συγκεκριμένα απαιτείται να ικανοποιούνται τα εξής ως προς την αίτηση που προωθείται προς τους WSN Driver:

- Χρήση μιας κοινά αποδεκτής γλώσσας για τη μορφοποίηση της αίτησης και της απάντησης. Για τη συγκεκριμένη απαίτηση έχει επιλεγεί η USL που είναι βασισμένη στην XML. Αυτή η γλώσσα παρέχει μια σειρά από δυνατότητες (όπως η χρήση περιορισμών, ορίων κ.α.)
- Η αίτηση η οποία καταφθάνει στον RR Proxy, αναλύεται σημασιολογικά. Συγκεκριμένα, ελέγχονται τα μεγέθη που υποστηρίζει (Temperature, Humidity κλπ) και αναλόγως προωθείται στους υποκείμενους WSN Driver που τα υποστηρίζουν.
- Δυνατότητα εξυπηρέτησης πολλών αιτήσεων ταυτόχρονα ώστε να μην υπάρχουν άσκοπες καθυστερήσεις
- Το σύστημα προωθεί την αίτηση προς τους υποκείμενους WSN Driver, οι οποίοι μπορεί να είναι εγκατεστημένοι σε διαφορετικές μηχανές. Η

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

επικοινωνία, λοιπόν, επιτυγχάνεται μέσω TCP/IP σε συγκεκριμένη πόρτες και διευθύνσεις που «ακούν» οι συγκεκριμένοι WSN Driver.

- Μεταφερσιμότητα (Portability). Για το λόγο αυτό έχει επιλεγεί η αντικειμενοστραφής γλώσσα προγραμματισμού Java.

Όπως φαίνεται και στο σχηματικό διάγραμμα του συστήματος (σχήμα 6.1), ο RR Proxy αποτελείται από διαφορετικές κλάσεις. Οι βασικές είναι οι εξής

Request Preprocessing: Αναμένει τις αιτήσεις

από το χρήστη –API , τις συγκεντρώνει και

τις προωθεί αναλόγως προς τους υποκείμενους WSN Driver

TCP_Select_Socket:

Πραγματοποιεί τη σύνδεση με τους

υποκείμενους WSN Driver. Αποτελεί υλοποίηση ενός TCP Select Socket, χρησιμοποιήθηκε γιατί επιτρέπει ασύγχρονη επικοινωνία και αποφυγή αδιεξόδων.

Dispatcher:

Συλλέγει όλες τις απαντήσεις από τους

υποκείμενους WSN Driver. Τις κατηγοριοποιεί ανάλογα με το ID τους και τις προωθεί στην κλάση Response Creator.

Response Creator:

Συγκεντρώνει όλες τις αιτήσεις από την κλάση

Dispatcher που αντιστοιχούν σε αυτόν (δηλαδή σε ένα συγκεκριμένο ID) και

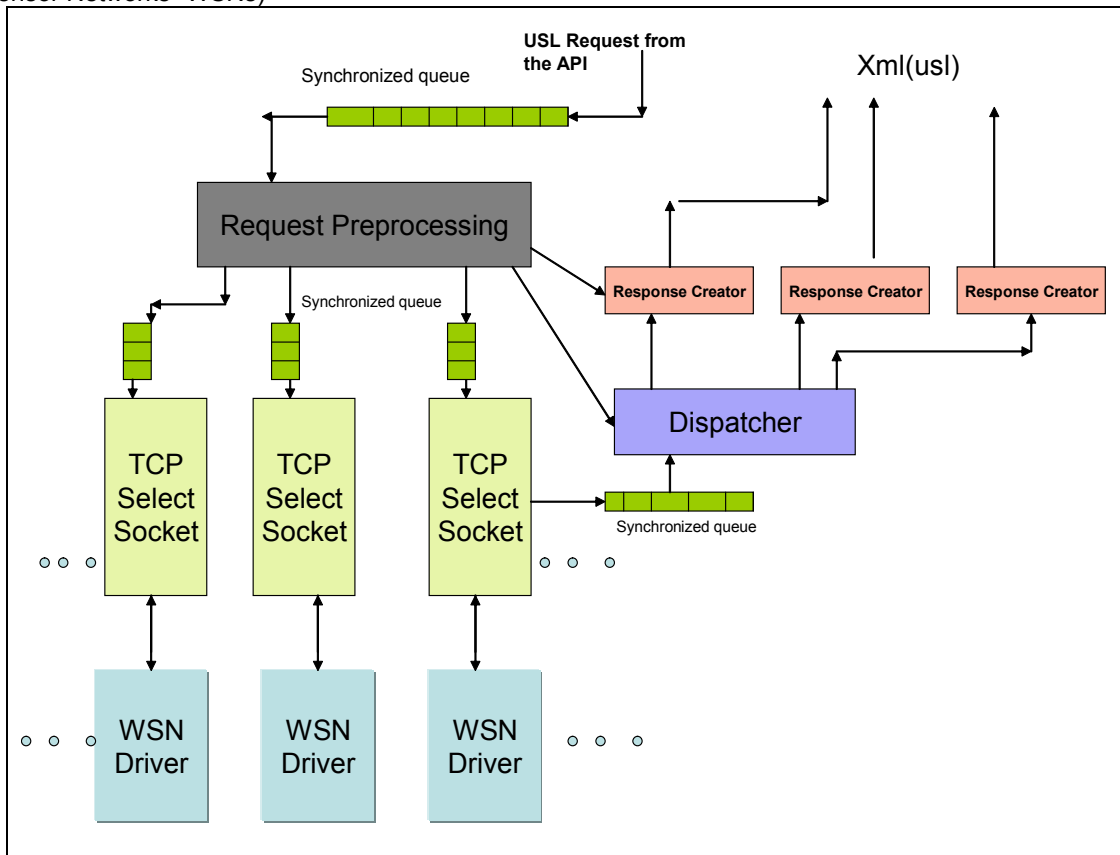
ακολουθώς τις συνενώνει σε μια ενιαία

XML απάντηση.

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

Παράλληλα, χρησιμοποιείται ένα σύνολο κλάσεων οι οποίες υλοποιούν τη λογική του συστήματος. Αυτές είναι οι παρακάτω:

- 1) **All_Basic_Classes**, η οποία περιέχει όλες τις συναρτήσεις επεξεργασίας και ανάλυσης των XML αιτήσεων που προωθούνται στα διαφορετικά επίπεδα της αρχιτεκτονικής,
- 2) **Sens_Type**, η οποία ελέγχει ποια δίκτυα υποστηρίζουν μία αίτηση με βάση τα ζητούμενα μεγέθη και
- 3) **CheckTime**, η οποία ελέγχει αν υπάρχουν αιτήσεις που δεν έχουν εξυπηρετηθεί μέσα στο προκαθορισμένο χρονικό περιθώριο, δηλαδή ο μέγιστος χρόνος αναμονής έχει εξαντληθεί (time out). Σε αυτήν την περίπτωση, ο *Response Creator* που σχετίζεται με τη συγκεκριμένη αίτηση καλείται να ολοκληρώσει τη διαδικασία, απαντώντας άμεσα με ό,τι απαντήσεις έχει συγκεντρώσει, είτε επιστρέφοντας ένα σχετικό μήνυμα σφάλματος.



Σχήμα 6.1 Διάγραμμα κλάσεων της αρχιτεκτονικής του RR Proxy

6.2) Request Preprocessing

Η κλάση αυτή αποτελεί το σημείο επικοινωνίας του API με τον RR Proxy. Πιο συγκεκριμένα αναμένει αιτήσεις από το API, τις οποίες υποβάλλει σε ένα πρώτο στάδιο επεξεργασίας και εν συνεχεία προωθεί στην κλάση TCP_Select_Socket. Πιο συγκεκριμένα, προωθεί τις κατάλληλες αιτήσεις στις ουρές που βρίσκονται στο TCP_Select_Socket. Παράλληλα δημιουργεί και τις κλάσεις Response Creator για κάθε αίτηση (ένας Response Creator για κάθε αίτηση).

Κάθε αίτηση επεξεργάζεται ως προς τα μεγέθη που υποστηρίζει (Temperature, Humidity, κτλ). Τα μεγέθη της συγκρίνονται με τα μεγέθη που υποστηρίζει η εκάστοτε ουρά της κλάσης TCP_Select_Socket (Temperature, Humidity, κτλ). Εάν όλα τα μεγέθη

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

της αίτησης υποστηρίζονται, τότε η συγκεκριμένη αίτηση προωθείται προς αυτήν την ουρά. Τα μεγέθη που υποστηρίζει η εκάστοτε ουρά εκφράζουν ουσιαστικά τα μεγέθη που υποστηρίζουν τα υποκείμενα δίκτυα (κλάση WSN Driver) στο πιο κάτω επίπεδο (βλέπε σχήμα 6.1) .

6.3) TCP_Select_Socket

Η συγκεκριμένη κλάση δέχεται αιτήσεις από την κλάση RequestPreprocessing και στη συνέχεια τις προωθεί στους υποκείμενους WSN Drivers. Συγκεκριμένα, η κλάση αυτή είναι υπεύθυνη να κάνει εγκατάσταση της σύνδεσης μέσω socket με τους WSN_Drivers. Στη συνέχεια στέλνει τα Requests που δέχεται από την κλάση Request Preprocessing στους WSN_Drivers και περιμένει την απάντησή τους. Κατόπιν, προωθεί τις απαντήσεις στον Dispatcher.

6.4) WSN_Driver

Ο ρόλος του WSN Driver είναι ουσιαστικά να παραλαμβάνει τις αιτήσεις που αποστέλλονται από την κλάση TCP_Select_Socket και στη συνέχεια να επιστρέφει τις αντίστοιχες απαντήσεις, αφού επικοινωνήσει με το υποκείμενο δίκτυο αισθητήρων. Η υλοποίηση του, σύμφωνα με τις προδιαγραφές που έχουν συζητηθεί πιο πάνω (παράγραφο 6.1), δεν αποτελεί αντικείμενο της παρούσας εργασίας. Εδώ υλοποιήθηκε μία κλάση που προσομοιώνει τη λογική του WSN Driver, ώστε να μπορεί να δοκιμαστεί η ορθή λειτουργία του RR Proxy.

6.5)Dispatcher

Στην κλάση αυτή συγκεντρώνονται όλες οι αιτήσεις από κάθε WSN_Driver και αναλόγως διανέμονται στους αντίστοιχους Response_Creator. Η διαλογή πραγματοποιείται βάσει του ID που φέρει κάθε απάντηση και είναι το ίδιο με το ID της αντίστοιχης αίτησης (και Response creator). Ελέγχεται το ID του response και

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)
αναλόγως προωθείται προς τον κατάλληλο Response creator ο οποίος χαρακτηρίζεται από το αντίστοιχο ID.

Ο Dispatcher ουσιαστικά κάνει τη διανομή των απάντησης προς τους αντίστοιχους Response creator προκειμένου να συγκεντρωθούν όλες οι απαντήσεις και να προωθηθούν συνοψισμένες σε μία απάντηση προς το API.

6.6)Response Creator

Καθώς ο ρόλος της προτεινόμενης πλατφόρμας είναι να καλύψει την ετερογένεια μεταξύ διαφορετικών δικτύων αισθητήρων, το σενάριο ταυτόχρονης ύπαρξης περισσότερων του ενός δικτύου, είναι κάτι το αναμενόμενο. Αυτό έχει ως αποτέλεσμα ότι μία αίτηση αφού «κλωνοποιηθεί» και προωθηθεί στα υπάρχοντα δίκτυα, θα παράγει ένα ισάριθμο πλήθος απαντήσεων. Αυτές οι απαντήσεις θα πρέπει με κάποιο τρόπο να ομαδοποιηθούν, ώστε τελικά να παραδοθεί μία ενιαία απάντηση. Ο ρόλος της κλάσης Response Creator ουσιαστικά βασίζεται στη δημιουργία μιας συνολικής απάντησης σε ό,τι αφορά τις διαφορετικές αιτήσεις που καταφθάνουν στον RR Proxy.

Οι απαντήσεις ομαδοποιούνται σε μια ενιαία απάντηση η οποία έχει τα εξής χαρακτηριστικά:

- Κοινό ID με όλες τις απαντήσεις που ομαδοποιούνται
- Όλα τα μεγέθη που υποστηρίζει η κάθε απάντηση
- Το πεδίο ReturnStatus σε κάθε απάντηση θα πρέπει να είναι False προκειμένου να συνενωθούν σε μία συνολική απάντηση.

Μπορούμε ενδεικτικά να αναφέρουμε ως παράδειγμα συνολικής απάντησης αυτό του σχήματος 6.2. Οι αιτήσεις που κατέφτασαν στο Response Creator είναι αυτές των σχημάτων 6.2.α , 6.2.β, 6.2.γ. Ενώ παρατηρούμε ότι συνενώθηκαν μόνο αυτές όπου το πεδίο ReturnStatus είναι False

```
<?xml version="1.0" encoding="UTF-8"?>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents and
Settings\Zetaki\ptyxiakh\dom4jfiles\WSNResponse.xsd" ID="403">
  <SensorValues>
    <SensorValue function="NoFunction" type="Temperature">15</SensorValue>
    <SensorValue function="NoFunction" type="Temperature">25</SensorValue>
  </SensorValues>
  <ReturnStatus error="false">
    <ErrorType>TIMEOUT</ErrorType>
  </ReturnStatus>
</Response>
```

Σχήμα 6.2 Παράδειγμα Συνολικής απάντησης

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents and
Settings\Zetaki\ptyxiakh\dom4jfiles\WSNResponse.xsd" ID="403">
  <SensorValues>
    <SensorValue function="NoFunction" type="Temperature">15</SensorValue>
  </SensorValues>
  <ReturnStatus error="false">
    <ErrorType>TIMEOUT</ErrorType>
  </ReturnStatus>
</Response>
```

Σχήμα 6.2.α Παράδειγμα απάντησης


```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)--
>
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents and
Settings\Zetaki\ptyxiakh\dom4jfiles\WSNResponse.xsd" ID="403">
  <SensorValues>
    <SensorValue function="NoFunction"
type="Temperature">29</SensorValue>
  </SensorValues>
  <ReturnStatus error="true">
    <ErrorType>TIMEOUT</ErrorType>
  </ReturnStatus>
</Response>
```

Σχήμα 6.2.β Παράδειγμα απάντησης

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents and
Settings\Zetaki\ptyxiakh\dom4jfiles\WSNResponse.xsd" ID="403">
  <SensorValues>
    <SensorValue function="NoFunction" type="Temperature">25</SensorValue>
  </SensorValues>
  <ReturnStatus error="false">
    <ErrorType>TIMEOUT</ErrorType>
  </ReturnStatus>
</Response>
```

Σχήμα 6.2.γ Παράδειγμα απάντησης

6.7) Αναλυτικό Παράδειγμα Προώθησης και Επεξεργασίας αιτήσεων-Requests

Σε αυτήν την ενότητα, παρουσιάζεται, μέσω παραδειγμάτων /δοκιμών που διεξάχθηκαν, οι λειτουργίες επεξεργασίας και προώθησης των αιτήσεων και απαντήσεων στο σύστημα. Υποθέτουμε ότι καταφθάνουν στο σύστημα 3 USL αιτήσεις από το API, σε XML μορφή. Οι αιτήσεις αυτές, προωθούνται στο Synchronized Queue του RR proxy (όπως φαίνεται στο σχήμα 6.1), και από εκεί λαμβάνονται σταδιακά η μία μετά την άλλη από την κλάση Request Preprocessing. Μία τυπική USL αίτηση έχει μορφή όπως αυτή παρουσιάζεται στο σχήμα 6.3 .

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2 U (http://www.xmlspy.com) by Ba-k (ZonaWarez.com) -->
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents and
Settings\Zetaki\ptyxiakh\dom4jfiles\WSNRequest.xsd" ID="400">
  <Query>
    <RequestedInfo location="false" time-instance="false" device="true" sensor="true"
time-duration="false">
      <SensorList>
        <SensorType function="NoFunction" type="Temperature"/>
        <SensorType function="NoFunction" type="Humidity"/>
      </SensorList>
    </RequestedInfo>
    <QueryFilter>
      <SensorExpr>
        <SensorType function="NoFunction" type="Light-Intensity">
          <Greater value="38"/>
        </SensorType>
        <SensorType function="NoFunction" type="Temperature">
          <Less value="50"/>
        </SensorType>
      </SensorExpr>
    </QueryFilter>
  </Query>
</Request>
```

Σχήμα 6.3 Παράδειγμα USL Αίτησης

Η κλάση Request Preprocessing εξετάζει την αίτηση που καταφθάνει. Συγκεκριμένα ελέγχει τα πεδία

```
<SensorList>  
    <SensorType function="NoFunction" type="Temperature"/>  
    <SensorType function="NoFunction" type="Humidity"/>  
</SensorList>
```

όπου δηλώνονται τα μεγέθη που ζητά η αίτηση. Στο παράδειγμα-σχήμα 6.3 έχουμε μία αίτηση η οποία ζητά Temperature και Humidity, ενώ από τα πεδία

```
<SensorType function="NoFunction" type="Light-Intensity">  
    <Greater value="38"/>  
</SensorType>  
<SensorType function="NoFunction" type="Temperature">  
    <Less value="50"/>  
</SensorType>
```

συμπεραίνουμε πως η "Light-Intensity" απαιτούμε να έχει τιμή μεγαλύτερη από 38 , <Greater value="38"/> και η "Temperature" τιμή μικρότερη από 50 <Less value="50"/>.

Επομένως καταλήγουμε συνολικά στο συμπέρασμα πως τα απαιτούμενα μεγέθη που πρέπει να μπορούν να χειριστούν τα υποκείμενα δίκτυα είναι Temperature, Humidity, και Light-Intensity. Η κλάση Request Preprocessing με κατάλληλη επεξεργασία της αίτησης, αποφαινεται ποια είναι τα δίκτυα αυτά, και τελικά προωθεί την αίτηση μόνο στους αντίστοιχους WSN Driver (μέσω της αντίστοιχης κλάσης TCP Select Socket).

Οι υποκείμενες κλάσεις WSN_Driver, αφού λάβουν την αίτηση, επικοινωνούν με το υποκείμενο δίκτυο αισθητήρων και μόλις λάβουν τα δεδομένα που αφορούν την συγκεκριμένη αίτηση, σχηματίζουν την απάντηση σε USL μορφή. Η USL απάντηση στέλνεται εν συνεχεία στην κλάση TCP_Select_Socket.

Μια τυπική απάντηση-Response που καταφθάνει στις κλάσεις TCP_Select_Socket έχει τη μορφή που φαίνεται στο σχήμα 6.4

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents and
Settings\Zetaki\ptyxiakh\dom4jfiles\WSNResponse.xsd" ID="400">
  <SensorValues>
    <SensorValue function="NoFunction" type="Temperature">25</SensorValue>
    <SensorValue function="NoFunction" type="Humidity">66</SensorValue>
  </SensorValues>
  <ReturnStatus error="false">
    <ErrorType>TIMEOUT</ErrorType>
  </ReturnStatus>
</Response>
```

Σχήμα 6.4 Παράδειγμα USL Απάντησης

Η κλάση TCP Select Socket, όταν λάβει μία USL απάντηση, την προωθεί στο Dispatcher. Ακολούθως ο Dispatcher, βάσει του ID που φέρει η απάντηση (ίδιο με την αντίστοιχη αίτηση), την προωθεί στον Response Creator που είναι υπεύθυνος για την αίτηση αυτή¹. Ο Response Creator αφού συγκεντρώσει όλες τις απαντήσεις που

¹ Όπως έχει συζητηθεί και παραπάνω, ένας Response Creator δημιουργείται την στιγμή που καταφθάνει μία νέα αίτηση στον RR proxy και είναι υπεύθυνος για αυτήν και μόνον αυτήν την αίτηση.

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

αφορούν τη συγκεκριμένη αίτηση, συνενώνει τα δεδομένα των απαντήσεων σε μια ολοκληρωμένη απάντηση.

Πρέπει να σημειωθεί ότι κατά τη διάρκεια της διαδικασίας, υπάρχει ο χρόνος του συστήματος ο οποίος ξεκινά με την άφιξη της αίτησης στον RR Proxy. Όταν ξεπεραστεί κάποιο χρονικό διάστημα το οποίο μπορεί να καθοριστεί από τον χρήστη, τότε οι διαδικασίες επισπεύδονται και όσες απαντήσεις έχουν συγκεντρωθεί στον Response Creator συνενώνονται σε μια, έστω και αν δεν είναι όλες όσες θα έπρεπε να έχουν συγκεντρωθεί.

6.8 Ένα απλό παράδειγμα εφαρμογής του προγράμματος

Για να εκτελέσουμε το πρόγραμμα χρειαζόμαστε 4 παράθυρα command line (γραμμής εντολών), ένα για κάθε WSN Driver.

Με το που εγκαθίσταται η σύνδεση, οι WSN Drivers βγάζουν ένα μήνυμα αποδοχής της σύνδεσης το οποίο έχει ως εξής :

Accepted connection request from Socket [addr=/127.0.0.1,port=1042,localport=5023]

Το μήνυμα αυτό παρουσιάζει τα στοιχεία σύνδεσης του υπάρχοντος socket μεταξύ TCP Select Socket και WSN Driver. Δεδομένου ότι στη παρούσα εφαρμογή έχουμε 4 TCP Select Socket και 4 WSN Driver, συνακόλουθα και τα παράθυρα σύνδεσης είναι 4 και τα μηνύματα αντιστοίχως έχουν ως εξής:

Accepted connection request from Socket[addr=/127.0.0.1,port=1045,localport=5024]

Accepted connection request from Socket [addr=/127.0.0.1,port=1048,localport=5025]

Accepted connection request from Socket [addr=/127.0.0.1,port=1051,localport=5026]

Ουσιαστικά μεταξύ TCP Select Socket και WSN Driver εγκαθίσταται επικοινωνία μέσω Socket. Στο ρόλο του Client βρίσκεται ο TCP Select Socket και στο ρόλο του Server βρίσκεται ο WSN Driver.

Έτσι, κάθε φορά που εγκαθίσταται η επικοινωνία ο TCP Select Socket βγάζει ένα μήνυμα της μορφής :

```
Server Found
&&&
5023*****

Server Found
&&&
5024*****

Server Found
&&&
5025*****

Server Found
&&&
5026*****
```

Όπου υποδεικνύεται ότι έχει εγκατασταθεί η σύνδεση μεταξύ Client- Server (TCP Select Socket - WSN Driver). Το μήνυμα αυτό παρουσιάζεται 4 φορές, όσα δηλαδή και τα ζεύγη TCP Select Socket - WSN Driver που έχουμε στην εφαρμογή μας. Στη συνέχεια παρουσιάζονται τα μηνύματα που υποδεικνύουν ότι ο TCP Select Socket διαβάζει από την ουρά και τα στέλνει στον WSN Driver. Τα μηνύματα έχουν ως εξής:

Από τον Client (TCP Select Socket) :

This Request is supported from the WSN(Wireless sensor Network):1

By IP_address:127.0.0.1

By Port:5023

I am writing to the BUFFER1:\$\$\$\$\$\$\$\$\$<?xml version="1.0" encoding="UTF-8"?>

<!-- edited with XML Spy v4.2 U (<http://www.xmlspy.com>) by Ba-k (ZonaWarez.com) -->

<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:noNamespaceSchemaLocation="C:\Documents and

Settings\Zetaki\ptyxiakh\dom4jfiles\WSNRequest.xsd" ID="400"><Query>

<RequestedInfo location="false" time-instance="false" device="true" sensor="true" time-duration="false">

<SensorList>

<SensorType function="NoFunction" type="Temperature"/>

<SensorType function="NoFunction" type="Humidity"/>

</SensorList>

</RequestedInfo>

<QueryFilter>

<SensorExpr>

<SensorType function="NoFunction" type="Light-Intensity">

<Greater value="38"/>

</SensorType>

<SensorType function="NoFunction" type="Temperature">

<Less value="50"/>

</SensorType>

</SensorExpr>

</QueryFilter>

</Query>

</Request>

Πιο πάνω βλέπουμε ένα παράδειγμα μίας αίτησης USL, η οποία υποστηρίζεται μόνο από το δίκτυο (Wireless Sensor Network)1.

Συνεχίζοντας, αφού ο TCP Select Socket στείλει στον WSN Driver το παραπάνω μήνυμα. Τότε στην οθόνη του WSN Driver βλέπουμε τα εξής

Από τον Server (WSN Driver):

```
I read from the client: <?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XML Spy v4.2 U (http://www.xmlspy.com) by Ba-k (ZonaWarez.com)
-->
<Request xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\DocumentsandSettings\Zetaki\ptyxiakh\dom4jfiles
\WSNRequest.xsd" ID="400">
  <Query>
    <RequestedInfo location="false" time-instance="false" device="true"
sensor="true" time-duration="false">
      <SensorList>
        <SensorTypefunction="NoFunction" type="Temperature"/>
        <SensorTypefunction="NoFunction" type="Humidity"/>
      </SensorList>
    </RequestedInfo>
    <QueryFilter>
      <SensorExpr>
        <SensorType function="NoFunction" type="Light-Intensity">
          <Greater value="38"/>
        </SensorType>
        <SensorTypefunction="NoFunction" type="Temperature">
          <Less value="50"/>
        </SensorType>
      </SensorExpr>
    </QueryFilter>
  </Query>
</Request>
```

```
</SensorExpr>
</QueryFilter>
</Query>
</Request>
```

The ID of this Request is:::400

The Client receives : ++++++++<?xml version="1.0" encoding="UTF-8"?>

<!--Sample XML file generated by XML Spy v4.2 U (<http://www.xmlspy.com>)-->

```
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\DocumentsandSettings\Zetaki\ptyxiakh\dom4jfiles
\WSNResponse.xsd" ID="400">
```

```
<SensorValues>
  <SensorValuefunction="NoFunction" type="Temperature">25</Sensor
Value>
<SensorValue function="NoFunction" type="Humidity">66</SensorValue>
</SensorValues>
<ResponseStatus error="false">
  <ErrorType>TIMEOUT</ErrorType>
</ResponseStatus>
</Response>
```

Και η USL απάντηση που στέλνει ο WSN Driver ανταποκρίνεται στην αίτηση με ID=400.

Συνεχίζοντας, ο TCP Select Socket διαβάζει από το Socket την απάντηση, με αποτέλεσμα να δούμε τα εξής στην οθόνη:

Απάντηση Client (TCP Select Socket)

```
I Read from the buffer *+++++++<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents and
Settings\Zetaki\ptyxiakh\dom4jfiles\WSNResponse.xsd" ID="400">
  <SensorValues>
    <SensorValue function="NoFunction"
type="Temperature">25</SensorValue>
    <SensorValue function="NoFunction"
type="Humidity">66</SensorValue>
  </SensorValues>
  <ReturnStatus error="false">
    <ErrorType>TIMEOUT</ErrorType>
  </ReturnStatus>
</Response>
```

*

```
I am writing at the QUEUE :<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<Response xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents and
Settings\Zetaki\ptyxiakh\dom4jfiles\WSNResponse.xsd" ID="400">
  <SensorValues>
    <SensorValue function="NoFunction"
type="Temperature">25</SensorValue>
    <SensorValue function="NoFunction"
type="Humidity">66</SensorValue>
  </SensorValues>
```

```
<ResponseStatus error="false">
    <ErrorType>TIMEOUT</ErrorType>
</ResponseStatus>
</Response>
```

Στην ουρά(Queue), η οποία συνδέει τον TCP Select Socket με τον Dispatcher, καταλήγουν όλες οι απαντήσεις που στέλνουν οι υπάρχοντες WSN Driver. Όταν μία απάντηση φτάσει στον Dispatcher το παρακάτω μήνυμα εμφανίζεται:

Λήψη απάντησης στον Dispatcher

```
Got message: in Dispatcher <?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.2 U (http://www.xmlspy.com)-->
<Response      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Documents      and
Settings\Zetaki\ptyxiakh\dom4jfiles\WSNResponse.xsd" ID="400">
<SensorValues>
<SensorValue function="NoFunction" type="Temperature">25</SensorValue>
<SensorValue function="NoFunction" type="Humidity">66</SensorValue>
</SensorValues>
<ResponseStatus error="false">
    <ErrorType>TIMEOUT</ErrorType>
</ResponseStatus>
</Response>
```

Ακολουθώς ο Dispatcher προωθεί τις απαντήσεις στον κατάλληλο Response Creator, ο οποίος είναι υπεύθυνος να συνενώνει σε μια ενιαία USL-Response. Όπως έχει αναφερθεί και πιο πάνω, μία ενιαία απάντηση σχηματίζεται όταν συγκεντρωθούν όλες οι αναμενόμενες απαντήσεις πριν εκπνεύσει ένα συγκεκριμένο χρονικό περιθώριο, ή εάν αυτό το περιθώριο ξεπεραστεί, οπότε αποστέλλεται μία ενιαία απάντηση με δεδομένα μόνο από όσες απαντήσεις έχουν συγκεντρωθεί μέχρι εκείνη τη στιγμή.

ΚΕΦΑΛΑΙΟ 7

ΣΥΝΟΨΗ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΞΕΛΙΞΗ

Σε αυτήν την εργασία μελετήθηκε η ανάπτυξη μιας προγραμματιστικής πλατφόρμας ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων. Το βασικό αντικείμενο μελέτης αυτής της αρχιτεκτονικής είναι να καλύψει την ετερογένεια μεταξύ διαφορετικών τύπων και δομών δικτύων αισθητήρων.

Η ετερογένεια των δικτύων αισθητήρων βασίζεται στη διαφορετικότητά τους ως προς τον τύπο των αισθητήρων που τα απαρτίζουν, τα μεγέθη που μετρούν, ή και τις διαφορετικές πλατφόρμες υλικού και λογισμικού που χρησιμοποιούνται.

Επίσης, η πλατφόρμα έχει σχεδιαστεί με τέτοιο τρόπο, ώστε να προβάλλει ένα φιλικό προς τον προγραμματιστή εφαρμογών, τρόπο χειρισμού και επικοινωνίας, ενθαρρύνοντας με αυτό τον τρόπο την ανάπτυξη παρόμοιων εφαρμογών στο μέλλον.

Για την ανάπτυξη μιας τέτοιας πλατφόρμας, χρησιμοποιήθηκε το προγραμματιστικό περιβάλλον εργασίας του Jbuilder X και η γλώσσα προγραμματισμού Java. Η λειτουργικότητα της εφαρμογής στηρίζεται στη συνένωση των διαφορετικών απαντήσεων-Responses που καταφθάνουν στο σύστημα από τα υποκείμενα δίκτυα και οι οποίες αντιστοιχούν στις διαφορετικές αιτήσεις- Requests που στέλνονται στο σύστημα από το χρήστη.

Στα μελλοντικά σχέδια της εργασίας, βρίσκεται η μελέτη και η διαχείριση των πληροφοριών και των δεδομένων που συλλέγονται από ένα ασύρματο δίκτυο αισθητήρων. Πιο συγκεκριμένα, μελετάται η ανάπτυξη και ο σχεδιασμός μιας πλατφόρμας η οποία θα είναι υπεύθυνη για τη σωστή αποθήκευση του όγκου των δεδομένων και της επεξεργασίας τους.

ΚΕΦΑΛΑΙΟ 8

ΠΑΡΑΡΤΗΜΑ Α- ΛΕΠΤΟΜΕΡΕΙΕΣ ΕΚΤΕΛΕΣΗΣ ΤΗΣ ΕΦΑΡΜΟΓΗΣ

Για να εκτελεστεί το υλοποιημένο λογισμικό (RR proxy), είναι αναγκαίο να προηγηθεί κατάλληλη προετοιμασία του συστήματος. Όπως έχει περιγραφεί στα προηγούμενα κεφάλαια, θεωρούμε ότι υπάρχει ένα πλήθος διαθέσιμων δικτύων αισθητήρων και αντίστοιχα ένας WSN Driver για κάθε ένα από αυτά. Ο WSN Driver είναι υπεύθυνος να διερμηνεύει αιτήσεις και απαντήσεις μεταξύ του υποκείμενου δικτύου και του RR proxy. Καθώς ένας WSN Driver μπορεί να είναι εγκατεστημένος σε κάποιο μηχάνημα, άλλο από αυτό που είναι εγκατεστημένος ο RR proxy, η επικοινωνία μεταξύ τους επιτυγχάνεται μέσω TCP/IP. Υπενθυμίζουμε ότι η επικοινωνία γίνεται με τη βοήθεια sockets (client-server μοντέλο), όπου server είναι ο WSN Driver και client ο RR proxy.

Απαραίτητο για την επικοινωνία μεταξύ RR proxy και WSN Driver είναι η γνώση από τον πρώτο της διεύθυνσης του δεύτερου, ώστε να μπορεί να εγκαθιδρυθεί η TCP σύνθεση, μέσω της οποίας θα ανταλλάσσονται τα μηνύματα (αιτήσεις και απαντήσεις). Η TCP/IP διεύθυνση μίας δικτυακής εφαρμογής ορίζεται με τον αριθμό θύρας (port number) που «ακούει» η εφαρμογή και την IP διεύθυνση του μηχανήματος που την φιλοξενεί. Αυτή η πληροφορία πρέπει, λοιπόν, να είναι γνωστή στον RR proxy.

Όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, σημαντικό για την πλατφόρμα είναι να γνωρίζει τις δυνατότητες των υποκείμενων δικτύων, με βασικότερο το ποια μεγέθη είναι ικανό κάθε δίκτυο να μετρήσει, ώστε να προωθούνται ορθά οι αιτήσεις που καταφθάνουν από τους χρήστες. Όλες αυτές οι πληροφορίες (TCP/IP διεύθυνση των WSN Driver, δυνατότητες υποκείμενων δικτύων κτλ), σύμφωνα με τον σχεδιασμό της πλατφόρμας είναι αποθηκευμένες σε μία βοηθητική οντότητα, το Profile Registry. Στα πλαίσια της εργασίας αυτής δεν υλοποιήθηκε το Profile Registry στην πλήρη μορφή του, αλλά αντί αυτού υλοποιήθηκε μία απλοποιημένη δομή. Συγκεκριμένα, υλοποιήθηκε η κλάση Global_wsn στην οποία στατικά τίθενται όλες οι προαναφερθείσες πληροφορίες. Εκεί, δηλαδή, ορίζονται οι TCP/IP διευθύνσεις των υπαρχόντων WSN Driver, καθώς και τα μεγέθη που υποστηρίζουν τα υποκείμενα σε αυτούς δίκτυα αισθητήρων.

Πριν εκτελεστεί ο RR proxy, θα πρέπει να τεθούν οι σωστές τιμές στις μεταβλητές που υπάρχουν στην κλάση Global_wsn. Αφού γίνει αυτό, πρέπει να γίνει compile η κλάση,

ώστε να τεθούν σε ισχύ οι αλλαγές. Επίσης, θεωρούμε ότι θα πρέπει οι WSN Driver που μας ενδιαφέρουν να τρέχουν ήδη, περιμένοντας αιτήσεις σύνδεσης. Αλλιώς, ο RR proxy δε θα μπορέσει να κάνει τη σύνδεση και η διαδικασία θα αποτύχει.

Στο παράδειγμα που έχει περιγραφεί και παραπάνω, θεωρήθηκε ότι υπάρχουν τέσσερις WSN Drivers. Χωρίς βλάβη της γενικότητας, οι WSN Driver αυτοί είναι εγκατεστημένοι στο ίδιο μηχάνημα με τον RR proxy. Οι πληροφορίες αυτές είναι καταχωρημένες στην κλάση Global_wsn. Πρώτα, ανοίγουμε τέσσερα παράθυρα γραμμής εντολής, σε καθένα από τα οποία θα εκτελεστεί ένας WSN Driver. Η εκτέλεση γίνεται πληκτρολογώντας το path που βρίσκονται όλες οι κλάσεις που χρησιμοποιεί και τελικώς την κλάση WSN Driver ακολουθούμενη από το Port που χρησιμοποιεί ο συγκεκριμένος WSN Driver (στο συγκεκριμένο παράδειγμα 5023, 5024, 5025 και 5026). Ο RR proxy εκτελείται μέσω του Jbuilder (ή και από παράθυρο γραμμής εντολών), όπου και τρέχουμε την κλάση Dispatcher χωρίς ορίσματα.

ΚΕΦΑΛΑΙΟ 9

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] Tilemahos Hasiotis, George Alyfantis, Vassileios Tsetsos, Odysseas Sekkas, Stathes Hadjiefthymiades. “*Sensation: A Middleware Integration Platform for Pervasive Applications in Wireless Sensor Networks*”
- [2] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “*A Survey on Sensor Networks*”, IEEE Communications Magazine, August 2002
- [3] K. Martinez, J. K. Hart, and R. Ong, “*Sensor Network Applications*”, IEEE Computer, August 2004
- [4] S. R. Madden, “*The Design and Evaluation of a Query Processing Architecture for Sensor Networks*”, Ph.D. Thesis. UC Berkeley. Fall, 2003 (TinyDB)
- [5] P. Bonnet, J. E. Gehrke, and P. Seshadri, “*Querying the Physical World*”, IEEE Personal Communications, vol. 7, no. 5, pp 10-15, Oct. 2000. (Cougar)
- [6] C.-C. Shen, Ch. Srisathapornphat, and Ch. Jaikaeo, “*Sensor Information Networking Architecture and Applications*”, IEEE Personal Communications, August 2001 (SINA)
- [7] T. L. and M. Martonosi. “*Impala: A Middleware System for Managing Autonomic, Parallel Sensor Systems*”. ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, June 2003
- [8] W. Heinzelman, A. Murphy, H. Carvalho and M. Perillo, “*Middleware to Support Sensor Network Applications*”, IEEE Network Magazine Special Issue. Jan. 2004 (MiLAN)

- [9] Sh. Li A1, S. H. Son A1, J. A. Stankovic, "*Event Detection Services Using Data Service Middleware in Distributed Sensor Networks*", Lecture Notes in Computer Science, Springer-Verlag Heidelberg, Volume 2634 / 2003

- [10] D.H.Crocker, "*Standard for the format of ARPA Internet text messages*", STD11, RFC 822, UDEL, August 1982

- [11] G.J. Pottie, W.J. Kaiser, "Wireless integrated network sensors", Communications of the ACM 43 (5) (2000) 551-558.

- [12] Yang Yu, Bhaskar Krishnamachari, and Viktor K. Prasanna, "Issues in Designing Middleware for Wireless Sensor Networks", IEEE Network Magazine Special Issue, Vol. 18, Issue 1, pp. 15-21, January 2004

- [13] Hervaldo S. Carvalho, Amy L. Murphy, Wendi B. Heinzelman, Claudionor J. N. Coelho, "Network-Based Distributed Systems Middleware", in Proceedings of the 1st International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC'03) co-located with Middleware 2003, Rio de Janeiro, Brazil, July 2003.

- [14] Kay Römer, "Programming Paradigms and Middleware for Sensor Networks", GI/ITG Workshop on Sensor Networks, pp. 49-54, Karlsruhe, Germany, February 2004.

- [15] A. Boulis, C.C. Han, and M. B. Srivastava, "Design and Implementation of a Framework for Efficient and Programmable Sensor Networks", In MobiSys, San Francisco, CA, 2003.

- [16] D. Estrin, R. Govindan, J. Heidemann, S. Kumar, "Next century

Μελέτη και υλοποίηση συστημάτων ενδιάμεσου λογισμικού για ασύρματα δίκτυα αισθητήρων(Wireless Sensor Networks- WSNs)

challenges: scalable coordination in sensor networks”, ACM MobiCom'99, Washington, USA, 1999, pp. 263-270.