



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΥΦΥΗΣ ΔΙΑΧΕΙΡΙΣΗ ΤΗΣ ΕΤΕΡΟΓΕΝΕΙΑΣ ΣΤΟ
ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ

ΠΑΠΑΘΑΝΑΣΙΟΥ ΔΗΜΗΤΡΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κων/νος Κολομβάτσος
Επικ. Καθηγητής Πανεπιστημίου Θεσσαλίας

Λαμία 30 Ιουλίου έτος 2022



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΕΥΦΥΗΣ ΔΙΑΧΕΙΡΙΣΗ ΤΗΣ ΕΤΕΡΟΓΕΝΕΙΑΣ ΣΤΟ
ΔΙΑΔΙΚΤΥΟ ΤΩΝ ΠΡΑΓΜΑΤΩΝ

ΠΑΠΑΘΑΝΑΣΙΟΥ ΔΗΜΗΤΡΙΟΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κων/νος Κολομβάτσος
Επικ. Καθηγητής Πανεπιστημίου Θεσσαλίας

Λαμία 30 Ιουλίου έτος 2022



UNIVERSITY OF
THESSALY

SCHOOL OF SCIENCE

DEPARTMENT OF INFORMATICS & TELECOMMUNICATIONS

SMART HETEROGENEITY MANAGEMENT
IN THE INTERNET OF THINGS

PAPATHANASIOU DIMITRIOS

FINAL THESIS

ADVISOR

Konstantinos Kolomvatsos
Assistant Professor of the University of Thessaly

Lamia July 30 year 2022

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις ⁽¹⁾, που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάστηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.
2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.
3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια
4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: 30/6/2022

Ω - Η Δηλ.



(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Το Διαδίκτυο των Πραγμάτων, ή Internet of Things - IoT, είναι ένα σύστημα αλληλένδετων υπολογιστικών συσκευών, μηχανικών και ψηφιακών μηχανών στα οποία παρέχονται μοναδικά αναγνωριστικά και η δυνατότητα μεταφοράς δεδομένων μέσω ενός δικτύου χωρίς την απαίτηση διαμεσολάβησης ανθρώπινης δραστηριότητας. Παρά την εξάπλωσή του, το IoT εξακολουθεί να βρίσκεται σε πρώιμο αναπτυξιακό στάδιο και έχει ακόμα μεγάλο περιθώριο έρευνας σε ποικίλα και μείζον θέματα. Ένα από αυτά είναι η διαχείριση της ετερογένειας μεταξύ των έξυπνων αντικειμένων και η διαχείριση των δεδομένων τους. Πολλές προσεγγίσεις έχουν προταθεί τα τελευταία χρόνια για το κομμάτι αυτό, με διαφορετικούς συμβιβασμούς σε παράγοντες όπως η ακρίβεια, η αξιοπιστία και η πολυπλοκότητα χρόνου [41]. Στην εργασία αυτή παρουσιάζεται ένας δενδροειδής αλγόριθμος για τη συγχώνευση ετικετών δικτύου που αλληλεπιδρά με μηνύματα JSON. Ως έξοδο ο αλγόριθμος παρέχει ένα ιεραρχικό φυλλογενετικό δέντρο, το οποίο παρουσιάζει τις συστάδες που δημιουργήθηκαν από τις ετικέτες του δικτύου. Για την υλοποίηση της αξιολόγησής του χρησιμοποιήθηκαν οι αλγόριθμοι μέτρησης απόστασης Hamming και Levenshtein ενώ για το δίκτυο χρησιμοποιήθηκε τοπολογία αστέρα που χρησιμοποιούσε πακέτα UDP για την αποστολή μηνυμάτων.

ABSTRACT

The Internet of Things - IoT, is a system of interconnected computing devices having the ability to transfer data over a network without the need of human involvement. Despite its widespread use, the IoT is still in its infancy and has great scope for research on a variety of major issues. One of them is the management of heterogeneity between smart objects and the aggregation of their data. Many approaches have been made in recent years to solve this problem, with different compromises on factors such as accuracy, the reliability and time complexity [41]. This work presents a tree-based algorithm for aggregating network tags that interact with JSON messages. As an output, the algorithm provides a hierarchical phylogenetic tree, which presents the clusters created by the network tags. Hamming and Levenshtein distance measurement algorithms were used to carry out its evaluation, while the network was composed as a star topology using UDP packets to send messages.

Table of Contents

ΠΕΡΙΛΗΨΗ	1
ABSTRACT	2
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ	2
1.1 ΟΡΙΣΜΟΣ ΤΟΥ ΙοΤ	2
1.2 Προκλήσεις του ΙοΤ	2
1.3 Ετερογένεια στο ΙοΤ	3
1.4 Συγχώνευση δεδομένων / Data aggregation	3
1.5 Τεχνικές συγχώνευση δεδομένων	4
1.5.α Συγχώνευση βασισμένη σε συσταδοποίηση	4
1.5.β Συγχώνευση βασισμένη σε αλυσίδες ή νήματα	5
1.5.γ Συγχώνευση βασισμένη σε δέντρο	5
1.5.δ Συγχώνευση βασισμένη σε πλέγμα	6
1.5.ε Συγχώνευση χωρίς δομή	6
ΚΕΦΑΛΑΙΟ 2: Βιβλιογραφική Επισκόπηση	7
2.1 Τρόποι διαχείρισης της ετερογένειας στο ΙοΤ	7
2.1.α Εφαρμογή του Cloud σε ετερογενή περιβάλλοντα	7
2.1.β Διαχείριση των δεδομένων σε ετερογενή περιβάλλοντα	8
2.1.γ Σάρωση και ανακάλυψη συσκευών σε ετερογενή δίκτυα	9
2.1.δ Σημσιολογική διαλειτουργικότητα σε ετερογενή δίκτυα	11
2.2 Συγχώνευση δεδομένων με δενδρική προσέγγιση	12
ΚΕΦΑΛΑΙΟ 3: ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΕΤΙΚΕΤΩΝ ΣΕ ΕΤΕΡΟΓΕΝΗ ΔΙΚΤΥΑ	15
3.1 Περιγραφή αλγορίθμου	15
3.1.α Υπολογισμός κοντινότερου ζευγαριού	15
3.1.β Αντιστοίχιση ετικετών με τον πίνακα αποστάσεων	16
3.1.γ Εισαγωγή κόμβων στο δέντρο	17
3.2 Παρουσίαση αλγορίθμου	18
ΚΕΦΑΛΑΙΟ 4: Αποτελέσματα αξιολόγησης του αλγορίθμου	20
4.1 ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΟΜΟΙΩΣΗΣ	20
4.2 Παραγωγή δέντρου με αλγορίθμους μέτρησης απόστασης	21
4.2.α Ταχύτητα δημιουργίας δέντρου	21
4.2.β Κατανάλωση ενέργειας	22
ΚΕΦΑΛΑΙΟ 5: Συμπεράσματα και μελλοντική έρευνα	28
ΒΙΒΛΙΟΓΡΑΦΙΑ	29

ΚΕΦΑΛΑΙΟ 1: Εισαγωγή

1.1 Ορισμός του IoT

Το Διαδίκτυο των Πραγμάτων, ή IoT, είναι ένα σύστημα αλληλένδετων υπολογιστικών συσκευών, μηχανικών και ψηφιακών μηχανών στα οποία παρέχονται μοναδικά αναγνωριστικά και η δυνατότητα μεταφοράς δεδομένων μέσω ενός δικτύου χωρίς την απαίτηση διαμεσολάβησης ανθρώπινης ενέργειας, δηλαδή αλληλεπίδρασης human-to-human ή human-to-machine. Ένα “πράγμα”, στο IoT μπορεί να είναι ένα άτομο με εμφύτευμα παρακολούθησης καρδιάς, ένα αυτοκίνητο με διάφορους αισθητήρες ένδειξης στον οδηγό και γενικότερα οποιοδήποτε άλλο φυσικό ή ανθρωπογενές αντικείμενο στο οποίο μπορεί να εκχωρηθεί μια διεύθυνση Πρωτοκόλλου Διαδικτύου (IP) και να μεταφέρει δεδομένα μέσω δικτύου.

Το IoT δίνει τη δυνατότητα στα ποικίλα αντικείμενα αυτά (πράγματα) να ελέγχονται απομακρυσμένα μέσω της εδραίωσης κατάλληλης δικτυακής υποδομής, δημιουργώντας με αυτό το τρόπο ευκαιρίες άμεσης ενσωμάτωσης του φυσικού κόσμου με τα υπολογιστικά συστήματα, προσφέροντας την αξιοποίηση των πλεονεκτημάτων τους, όπως βελτίωση της αποτελεσματικότητας, της ακρίβειας αλλά και τη μείωση του κόστους.

1.2 Προκλήσεις του IoT

Παρά την ευρεία εξάπλωσή του, το IoT εξακολουθεί να βρίσκεται σε πρώιμο αναπτυξιακό στάδιο και έχει μεγάλο περιθώριο έρευνας σε ποικίλα θέματα όπως η εδραίωση προτύπων, η επεκτασιμότητα, η ασφάλεια του συστήματος, η χρήση κοινής γλώσσας περιγραφής υπηρεσιών, η ανακάλυψη υπηρεσιών συγκεκριμένου τομέα, η ενσωμάτωσή του με ήδη υπάρχοντα συστήματα πληροφορικής και πολλά ακόμη [30].

Στο IoT, ένα ζωτικό ζήτημα είναι η διαχείριση της ετερογένειας μεταξύ των έξυπνων αντικειμένων. Σήμερα, ο αριθμός των διασυνδεδεμένων συσκευών έχει ήδη ξεπεράσει τον αριθμό των φυσικών χρηστών, και η ανάπτυξη του Διαδικτύου των Πραγμάτων θα κάνει αυτή τη διαφορά ακόμη μεγαλύτερη. Ταυτόχρονα, η ετερογένεια του αναπτυσσόμενου υλικού και το λογισμικό αυξάνεται με ραγδαίους ρυθμούς. Ενώ το κλασικό Διαδίκτυο μας φέρνει αντιμέτωπους με 2 ή 3 κυρίαρχα λειτουργικά συστήματα, τις οικογένειες CPU και τεχνολογίες δικτύου, οι συσκευές IoT χαρακτηρίζονται από μια πολύ μεγαλύτερη ποικιλία. Όσον αφορά την δικτυακή επικοινωνία, βλέπουμε, για παράδειγμα, τεχνολογίες όπως το GSM για επικοινωνία μεγάλης εμβέλειας, το IEEE 802.11 για τοπικά δίκτυα και πρωτόκολλα

βασισμένα στο IEEE 802.15.4 για συσκευές με περιορισμό ισχύος. Αυτή η ετερογένεια θέτει σημαντικές προκλήσεις στην ανάπτυξη λύσεων IoT [31].

1.3 Ετερογένεια στο IoT

Η ετερογένεια εξ ορισμού είναι η ανομοιότητα μεταξύ διαφόρων στοιχείων ενός συνόλου. Στο IoT η ετερογένεια αναφέρεται κυρίως στις διαφορές ανάμεσα στα “πράγματα” που χρησιμοποιούνται για να παράξουν, αποθηκεύσουν και να στείλουν δεδομένα. Οι διαφορές αυτές μπορεί να είναι από διαφορετικά πρωτόκολλα επικοινωνίας, τύπους δεδομένων, υπολογιστικές δυνατότητες μέχρι και λειτουργικές διαφορές ανάμεσα σε διαφορετικούς κατασκευαστές.

Το περιβάλλον του IoT είναι το πιο γνωστό και αντιπροσωπευτικό παράδειγμα του ζητήματος της ετερογένειας. Αυτό συμβαίνει επειδή περιέχει μια πληθώρα διαφορετικών, ως προς τη φύση τους, συσκευών. Ο κύριος στόχος του IoT είναι η δημιουργία ενός κοινού τρόπου για την αφαίρεση της ετερογένειας αυτών των συσκευών και την επίτευξη της βέλτιστης αξιοποίησης των δυνατοτήτων τους [29]. Η ετερογένεια μπορεί να σταθεί εμπόδιο και στη δημιουργία μιας καθολικής εικόνας (aggregation) ενός οικοσυστήματος IoT συσκευών, γεγονός που θα περιορίσει την ευχέρεια πραγματοποίησης στατιστικής ανάλυσης στο δίκτυο.

1.4 Συγχώνευση δεδομένων / Data aggregation

Η συγχώνευση δεδομένων είναι οποιαδήποτε διαδικασία κατά την οποία συλλέγονται πληροφορίες και εκφράζονται σε συνοπτική μορφή για σκοπούς όπως η στατιστική ανάλυση. Ένας κοινός σκοπός της συγχώνευσης είναι η λήψη περισσότερων πληροφοριών για συγκεκριμένες ομάδες με βάση συγκεκριμένες μεταβλητές όπως η ηλικία, το επάγγελμα ή το εισόδημα. Σε έναν αισθητήρα IoT οι μεταβλητές αυτές μπορεί να είναι η υπολογιστική δυνατότητά του, οι πληροφορίες που συλλέγει κ.α.

Η κύρια λογική πίσω από τη συγχώνευση δεδομένων στο IoT, είναι ότι ελαχιστοποιεί την εξάντληση ενέργειας και το απαιτούμενο εύρος ζώνης δικτύου. Η χρήση διαφορετικών μεθόδων συγχώνευσης δεδομένων εξαλείφει τα περιττά ή επαναλαμβανόμενα δεδομένα. Αυτό μειώνει την κίνηση δικτύου ελαχιστοποιώντας σημαντικά τον αριθμό των πακέτων δεδομένων που αποστέλλονται. Οι κόμβοι αισθητήρων IoT μπορούν επίσης να εξαλείψουν πλεονασμούς στα δεδομένα που λαμβάνονται από γειτονικούς κόμβους πριν από τη μεταφορά των τελικών πακέτων δεδομένων.

1.5 Τεχνικές συγχώνευσης δεδομένων

Διάφορες τεχνικές συγχώνευσης έχουν διαμορφωθεί για να ικανοποιήσουν μια πληθώρα ζητημάτων. Η κάθε μια έχει τα δικά της πλεονεκτήματα και μειονεκτήματα. Αυτός είναι και ο λόγος που καθιστά σημαντική την επιλογή της καταλληλότερης τεχνικής. Οι δύο κατηγορίες τεχνικών είναι η **επίπεδη** και η **ιεραρχική** προσέγγιση συγχώνευσης δεδομένων [28].

Στα επίπεδα ασύρματα δίκτυα αισθητήρων, όλοι οι αισθητήρες παίζουν τον ίδιο ρόλο. Δηλαδή δεν υπάρχει κανένας ιεραρχικός διαχωρισμός. Κάθε κόμβος αισθητήρα εξυπηρετεί τον ίδιο σκοπό και όλοι οι κόμβοι του οικοσυστήματος IoT είναι ομότιμοι. Ένα μειονέκτημα των επίπεδων ασύρματων δικτύων αισθητήρων είναι ότι η συγκέντρωση δεδομένων λαμβάνει χώρα μόνο στην περιοχή του κόμβου συγκέντρωσης (sink node). Ως αποτέλεσμα, η καθυστέρηση δικτύου μπορεί να είναι υψηλή. Επίσης, εάν ο κόμβος απορρόφησης αποτύχει, αυτό επηρεάζει αρνητικά ολόκληρο το δίκτυο.

Με την ιεραρχική προσέγγιση των ασύρματων δικτύων αισθητήρων, υπάρχει μια ιεραρχία μεταξύ των επιμέρους κόμβων ανάλογα με τις δυνατότητές τους. Χονδρικά, οι κόμβοι χωρίζονται σε σταθμούς βάσης, κεφαλές συστάδων και κόμβους αισθητήρων. Οι κόμβοι αισθητήρων μέσα σε μια συγκεκριμένη συστάδα επικοινωνούν μεταξύ τους και στη συνέχεια επικοινωνούν με την κεφαλή της συστάδας (cluster head). Περισσότερη υπολογιστική ισχύς και αυξημένες δυνατότητες μετάδοσης δικτύου σημαίνουν λιγότερη διάρκεια ζωής της μπαταρίας. Έτσι, ένας από τους κύριους στόχους αυτής της μεθόδου δρομολόγησης είναι η επίτευξη καλύτερης ενεργειακής απόδοσης για τους αισθητήρες μέσα σε ένα σύμπλεγμα [27].

1.5.α Συγχώνευση βασισμένη σε συσταδοποίηση

Είναι μια ιεραρχική μέθοδος που ταιριάζει καλύτερα σε περιβάλλοντα αισθητήρων μεγάλης κλίμακας και περιορισμένης ενέργειας. Σε τέτοια σενάρια, δεν είναι αποτελεσματικό για τους αισθητήρες να μεταδίδουν τα δεδομένα IoT απευθείας στον κόμβο συγκέντρωσης (σταθμός βάσης). Αντίθετα, οι αισθητήρες μεταδίδουν δεδομένα σε έναν τοπικό συσσωρευτή, γνωστό και ως κεφαλή συστάδας (cluster head). Η κεφαλή του συμπλέγματος συγκεντρώνει δεδομένα από όλους τους αισθητήρες στη συστάδα του και τα μεταδίδει στον κόμβο συγκέντρωσης. Οι κεφαλές των συστάδων μπορούν να επικοινωνούν με τον κόμβο συγκέντρωσης απευθείας μέσω μεταδόσεων μεγάλης εμβέλειας.

Μπορούν επίσης να κάνουν multi-hopping μέσω άλλων κεφαλών συμπλέγματος. Τα τυπικά πρωτόκολλα σε αυτή τη τεχνική, περιλαμβάνουν ομαδοποιημένη διάχυση με

δυναμική συνάθροιση δεδομένων (CLUDDA), Χαμηλής Ενέργειας Προσαρμοστική Ιεραρχικοποιημένη Ομαδοποίηση (LEACH) και προσέγγιση Υβριδικής Ενεργειακά Αποτελεσματικής Κατανεμημένης Συσταδοποίησης (HEED) [27].

1.5.β Συγχώνευση βασισμένη σε αλυσίδες ή νήματα

Σε συγκεκριμένες περιπτώσεις, εάν το cluster head βρίσκεται πολύ μακριά από τους αισθητήρες, η επικοινωνία μεταξύ των αισθητήρων και της κεφαλής μπορεί να καταναλώσει υπερβολικές ποσότητες ενέργειας. Όταν συμβαίνει αυτό, είναι πιο αποτελεσματικό για τους αισθητήρες να μεταδίδουν δεδομένα μόνο στους πλησιέστερους γείτονές τους στο δίκτυο. Η συγχώνευση δεδομένων με βάση την αλυσίδα είναι μια ιεραρχική μέθοδος σύμφωνα με την οποία κάθε αισθητήρας μεταδίδει μόνο στον πλησιέστερο γείτονά του. Οι κόμβοι είναι συνήθως οργανωμένοι σε μια γραμμική αλυσίδα συγκέντρωσης δεδομένων.

Ο κόμβος που βρίσκεται πιο μακριά από το σταθμό βάσης ξεκινά το σχηματισμό αλυσίδας. Σε κάθε βήμα, ο πλησιέστερος γείτονας ενός κόμβου επιλέγεται ως ο διάδοχός του στην αλυσίδα. Έτσι, ένας κόμβος λαμβάνει δεδομένα από έναν από τους γείτονές του και τα συγχωνεύει με τα δικά του. Στη συνέχεια μεταδίδει τα συγχωνευμένα δεδομένα “πιο κάτω” στην αλυσίδα, δηλαδή στον επόμενο γείτονά του. Ένας κύριος ή αρχικός κόμβος, δουλεύει παρόμοια με το cluster head σε συγχώνευση συστάδων, μεταδίδει τα συγκεντρωτικά δεδομένα στο σταθμό βάσης. Ένα παράδειγμα chain based aggregation είναι το PEGASIS [27].

1.5.γ Συγχώνευση βασισμένη σε δέντρο

Σε αυτή τη τεχνική, τα δεδομένα συγκεντρώνονται μέσω της δημιουργίας ενός δέντρου συνάθροισης δεδομένων. Οι κόμβοι αισθητήρων είναι οργανωμένοι με τέτοιο τρόπο ώστε η συγχώνευση των δεδομένων να λαμβάνει χώρα σε ενδιάμεσους κόμβους κατά μήκος του δέντρου. Ο ριζικός κόμβος λαμβάνει μόνο μια ήδη δομημένη αναπαράσταση των δεδομένων. Αυτή η τεχνική συγκέντρωσης είναι κατάλληλη για εφαρμογές που απαιτούν συγκέντρωση δεδομένων εντός δικτύου. Μία από τις κύριες προκλήσεις της συνάθροισης βάσει δέντρων είναι η δημιουργία ενός ενεργειακά αποδοτικού δέντρου συγκέντρωσης δεδομένων που βελτιστοποιεί τη διάρκεια ζωής του δικτύου και ελαχιστοποιεί τον αριθμό των μεταδόσεων. Κατά μέσο όρο, οι μέθοδοι που βασίζονται σε δέντρα είναι γνωστό ότι έχουν υψηλή επιβάρυνση, υψηλή ενεργειακή ομοιομορφία, καθώς και μεγαλύτερη ευελιξία και επεκτασιμότητα σε σύγκριση με τις μεθόδους που βασίζονται σε συστάδες [27].

1.5.δ Συγχώνευση βασισμένη σε πλέγμα

Αυτή η μέθοδος βασίζεται στη διαίρεση της περιοχής ενός δικτύου αισθητήρων σε πολλά πλέγματα. Ένα σύνολο αισθητήρων λειτουργεί ως συγκεντρωτής δεδομένων σε προκαθορισμένες περιοχές του δικτύου αισθητήρων. Έτσι το δίκτυο έχει έναν συσσωρευτή δεδομένων που είναι σταθερός σε κάθε πλέγμα. Η συστοιχία (array) των αισθητήρων λειτουργεί ως συσσωρευτής εντός αυτής της συγκεκριμένης περιοχής του δικτύου αισθητήρων. Οι αισθητήρες σε αυτό το συγκεκριμένο πλέγμα μεταδίδουν τα δεδομένα απευθείας στον συσσωρευτή δεδομένων που συγκεντρώνει τα δεδομένα από όλους τους αισθητήρες IoT εντός του δικτύου. Στη συγχώνευση βάσει πλέγματος, οι μεμονωμένοι αισθητήρες σε ένα πλέγμα δεν επικοινωνούν μεταξύ τους. Η συγκέντρωση δεδομένων που βασίζεται σε πλέγμα έχει ως πλεονέκτημα την προσαρμοστικότητα της σε δυναμικές αλλαγές στο δίκτυο [27].

1.5.ε Συγχώνευση χωρίς δομή

Η συγκέντρωση δεδομένων χωρίς δομή δεν περιλαμβάνει κανένα είδος αρχιτεκτονικής. Η επικοινωνία πραγματοποιείται από οποιονδήποτε κόμβο σε οποιονδήποτε κόμβο εντός του δικτύου. Σε ορισμένες περιπτώσεις, όπως για παράδειγμα σε εφαρμογές που βασίζονται σε συμβάντα που διαφέρουν ανά περιοχή συμβάντος, η συνάθροιση χωρίς δομή είναι η προτιμώμενη προσέγγιση [27].

ΚΕΦΑΛΑΙΟ 2: Βιβλιογραφική Επισκόπηση

2.1 Τρόποι διαχείρισης της ετερογένειας στο IoT

Η ετερογένεια στο διαδίκτυο των πραγμάτων μπορεί να στοχευθεί και να “αντιμετωπιστεί” με διάφορους τρόπους και σε διάφορα επίπεδα του οικοσυστήματος, με σκοπό πάντα να εξομαλυνθούν λειτουργίες επικοινωνίας, διαχείρισης δεδομένων, βελτίωσης ασφάλειας και άλλα.

2.1.α Εφαρμογή του Cloud σε ετερογενή περιβάλλοντα

Με τη ραγδαία διάδοση του IoT όλο και περισσότερα δεδομένα θα παράγονται από συσκευές, τα οποία θα πρέπει να επεξεργαστούν και να αποθηκευτούν [2]. Η τεχνολογία του cloud δίνει όλο και περισσότερες δυνατότητες είτε σε μορφή πόρων είτε σε μορφή δεδομένων, ενώ προσφέρει και μια πληθώρα εφαρμογών που παρέχουν υπηρεσίες στο διαδίκτυο [20]. Για την ανάλυση των δεδομένων υπάρχουν πλατφόρμες στο cloud που παρέχουν υπηρεσίες μηχανικής μάθησης, με προ-εκπαιδευμένα μοντέλα που κατανοούν δεδομένα IoT. Πρακτικά το cloud αποτελεί μια από τις καλύτερες μεθόδους για τη διαχείριση του μεγάλου όγκου δεδομένων που παράγονται από το IoT καθώς προσφέρει μεγάλη επεκτασιμότητα, λογικό κόστος και χρηστικότητα [1].

Η χρήση της ακμής (edge) για την παράκαμψη των εμποδίων της χρήσης του cloud, όπως η απόσταση από τους αισθητήρες (sensors) και ο χρόνος αποστολής δεδομένων, και η χρήση του για την εύρεση του καλύτερου μοντέλου μηχανικής μάθησης είναι κάτι που προτάθηκε στο [1]. Το framework που ειπώθηκε, χρησιμοποιεί μια συνεργασία ανάμεσα στο cloud και στο edge κατά την οποία, το cloud θα πρέπει να αποφορτίσει τον υπολογιστικό φόρτο του edge (λόγω της πιθανής χαμηλής υπολογιστικής ισχύς του) και να μεταφέρει το σωστό μοντέλο ανάλογα με τα χαρακτηριστικά των δεδομένων. Παράλληλα το edge θα πρέπει να στέλνει όσο το δυνατόν λιγότερα δεδομένα στο cloud και να μπορέσει να χρησιμοποιήσει το αποτέλεσμα όσο πιο γρήγορα γίνεται. Ουσιαστικά το cloud συλλέγει και μετατρέπει τα δεδομένα σε κατάλληλη μορφή για επεξεργασία ενώ χρησιμοποιεί έναν model trainer, ο οποίος με τα δεδομένα αυτά δημιουργεί διάφορα μοντέλα, τα οποία αντιστοιχούν σε δυνατότητες των ακμών. Έπειτα ο model selector διαλέγει το καλύτερο μοντέλο, δηλαδή αυτό που ταιριάζει καλύτερα στις λειτουργίες της ακμής, και το προωθεί στο edge το οποίο το χρησιμοποιεί για τη πρόβλεψη και αντλεί τα αποτελέσματα. Ακόμα το μοντέλο

αποθηκεύεται στην ακμή για μετέπειτα χρήση του, μειώνοντας έτσι ακόμα περισσότερο το υπολογιστικό κόστος.

Ένας ακόμη τρόπος για την αποφόρτιση του cloud, αυτή τη φορά, στις επικοινωνίες συσκευών προτείνεται στο [3], όπου εφαρμόζεται η χρήση του fog computing μέσω μιας smart gateway. Το ενδιάμεσο επίπεδο (layer) του fog παρέχει προσωρινή αποθήκευση και ασφάλεια των δεδομένων, ενώ μπορεί ακόμα να τα προ επεξεργαστεί, σε συνδυασμό πάντα με την έξυπνη πύλη. Επιπλέον επειδή το fog είναι τοπικό, έχει επικοινωνία χαμηλής καθυστέρησης και καλύτερη επίγνωση του πλαισίου. Αυτό έχει ως αποτέλεσμα η ομίχλη να μπορεί να παρέχει ροή καλύτερης ποιότητας στους κόμβους.

Πέρα από την χρήση του cloud για επεξεργαστική δύναμη, μπορεί ακόμα να χρησιμοποιηθεί για την διευκόλυνση ενσωμάτωσης νέων συσκευών σε ένα οικοσύστημα IOT όπως αυτό του [4]. Σε αυτή τη περίπτωση το cloud χρησιμοποιείται κυρίως για την εύρεση οδηγών (drivers) για τις νέες συσκευές καθώς και για την εύρεση εφαρμογών μέσω ενός application store. Για να επιτευχθεί η λειτουργία αυτή, γίνεται χρήση ενός home gateway που συνδέεται με ένα cloud solution center στο οποίο βρίσκεται το application store, το οποίο παρέχει υπηρεσίες cloud. Όσον αφορά το κομμάτι των οδηγών, ένας deployment manager προωθεί τα χαρακτηριστικά της συσκευής στο cloud, με σκοπό να βρεθούν τα κατάλληλα αρχεία. Ως αποτέλεσμα όταν μια συσκευή είναι έτοιμη να ενταχθεί στο οικοσύστημα μέσω της διαδικασίας που προαναφέρθηκε, θα έχει έτοιμους προτεινόμενους οδηγούς καθώς και εφαρμογές για την βέλτιστη λειτουργία της.

2.1.β Διαχείριση των δεδομένων σε ετερογενή περιβάλλοντα

Παρόλες τις εξελίξεις στο IOT, η σύλληψη, αποθήκευση και επεξεργασία ετερογενών δεδομένων δεν έχει βελτιωθεί σε ικανοποιητικό βαθμό [5]. Το IoT αποτελεί μια κύρια πηγή μεγάλου όγκου δεδομένων, λόγω της τεράστιας ροής τους ανάμεσα στους δισεκατομμύρια αισθητήρες που το αποτελούν. Για να μπορέσουν τα δεδομένα αυτά να αξιοποιηθούν πρέπει να ληφθούν μέτρα για τη διαχείριση, αποθήκευση και την ασφάλειά τους.

Η αρχιτεκτονική που παρουσιάζεται στο [5], εμφανίζει τα τυπικά μοντέλα υπηρεσιών cloud. Το IaaS (Infrastructure as a Service), το οποίο περιέχει το υλικό (hardware) που χρειάζεται για να λειτουργήσει. Το PaaS/SDaaS (Platform as a Service/Sensor Data as a Service), όπου παρέχει πρόσβαση στα δεδομένα. Και τέλος SaaS (Software as a Service), όπου υπάρχουν οι εφαρμογές με τις οποίες τα δεδομένα επεξεργάζονται, δια μοιράζονται και

ελέγχονται. Οι παραπάνω υπηρεσίες διαχειρίζονται μέσω ενός SCN (Service Controlled Networking) middleware.

Σε παρόμοια βήματα βαδίζει και η πρόταση του [6], στην οποία εξετάζεται η εφαρμογή τριών επιπλέον επιπέδων στην αρχιτεκτονική. Αυτά είναι το Data Cleaning Layer για καθαρισμό των δεδομένων από τον θόρυβο. Το Event Processing Layer, στο οποίο τα “καθαρισμένα” δεδομένα θα προ επεξεργαστούν με σκοπό να βρεθεί η σημασιολογία τους. Ενώ τέλος υπάρχει το Data Storage and Analysis Layer στο οποίο τα δεδομένα θα αποθηκευτούν και θα επεξεργαστούν.

2.1.γ Σάρωση και ανακάλυψη συσκευών σε ετερογενή δίκτυα

Η μεγάλη διασυνδεσιμότητα είναι ένα από τα κύρια χαρακτηριστικά του IoT. Με τη νέα γενιά των κινητών τηλεφώνων, οι άνθρωποι έχουν συνηθίσει να είναι συνδεδεμένοι στο διαδίκτυο και να μπορούν να επικοινωνούν με καθέναν ή καθετί οποιαδήποτε ώρα και σε οποιοδήποτε χώρο [9]. Για αυτό το λόγο η χρήση και ζήτηση όλο και περισσότερων ηλεκτρονικών συσκευών δεν φαίνεται να σταματά στο σύντομο μέλλον. Η μεγάλη αύξηση των συσκευών αυτών θα μας ωθήσει αργά ή γρήγορα στην ανάγκη για καλύτερη υποδομή δικτύου, το οποίο θα αντέχει την κίνηση δεδομένων από τις συσκευές αυτές [7].

Στα [9] και [11] προτείνεται η εφαρμογή της οντολογίας, κατά την οποία όταν μια συσκευή εισαχθεί στο οικοσύστημα τότε με βάση τα χαρακτηριστικά της συσκευής θα της δίνεται μια σημασιολογία, ώστε έπειτα να μπορέσει να ενταχθεί σε μια κατηγορία. Η κατηγοριοποίηση αυτή μπορεί να γίνει με διάφορους τρόπους όπως για παράδειγμα στο [11] που χρησιμοποιείται ένας Naïve Bayesian Classifier.

Αρκετά παρόμοια είναι και η πρόταση του [7], στο οποίο γίνεται χρήση διαμόρφωσης προφίλ κάθε συσκευής του οικοσυστήματος με στόχο να συντεθεί μια πλειάδα συσκευών που μοιράζονται την ίδια χωροχρονική περιοχή. Το προφίλ αυτό περιέχει πληροφορίες της συσκευής, τη σχέση της με άλλες συσκευές στο οικοσύστημα, το μέρος που πρέπει να τοποθετηθεί η συσκευή και τέλος τις δυνατότητες της. Η διαχείριση των συσκευών αυτών γίνεται μέσω ενός πλήρως κεντροποιημένου (centralized) δικτύου SDIoT.

Όσον αφορά τα WSN (Wireless Sensor Networks), επειδή οι περισσότερες συσκευές λειτουργούν με μπαταρίες και υπάρχει περίπτωση να μην είναι διαθέσιμοι για εξερεύνηση όλοι οι κόμβοι του συστήματος, επιβάλλεται να χρησιμοποιηθεί μια άλλη προσέγγιση. Το

[16] αντιμετωπίζει αυτό το πρόβλημα χρησιμοποιώντας ένα τοπικό συντονισμό που δεν βασίζεται σε κεντρικό μητρώο. Αντί αυτού κάθε συσκευή έχει το δικό της μητρώο που περιέχει μια τοπική εικόνα των διαθέσιμων συσκευών. Κατά την εξερεύνηση πρώτα ελέγχεται το τοπικό μητρώο της συσκευής και αν δεν βρεθεί αποτέλεσμα ελέγχονται διαδοχικά όλες οι γειτονικές διαθέσιμες συσκευές.

Οι συγγραφείς του [25], χρησιμοποιούν μια τεχνική συστάδων για τη συλλογή των δεδομένων σε ετερογενή WSN δίκτυα. Στο δίκτυο οι κόμβοι υψηλής ενέργειας χρησιμοποιούνται ως Cluster Heads (CH), και δέχονται δεδομένα από όλους τους κόμβους της συστάδας και έπειτα προωθούν το μήνυμα στο Sink Node του συστήματος. Λόγω της αυξημένης κυκλοφορίας η ενέργεια (μπαταρία) του CH εξαντλείται γρήγορα. Για να αντιμετωπίσουν αυτό οι συγγραφείς εφάρμοσαν το Data Aggregation Layer το οποίο ελέγχει τη κυκλοφορία των μηνυμάτων και σε συνδυασμό με την εφαρμογή ενός TDMA slot σε όλους τους κόμβους διαχειρίζονται περιπτώσεις μεγάλης κίνησης. Σε παρόμοια βήματα βαδίζει και η πρόταση του [26], η οποία σε συνδυασμό με τις συστάδες χρησιμοποιεί και μια βελτιωμένη έκδοση του αλγόριθμου SEP, με την οποία οι κόμβοι της συστάδας αυτοανακηρύσσονται CH με βάση τα επίπεδα ενέργειάς τους. Αυτό πετυχαίνει μια ομοιόμορφη κατανομή της ενέργειας των κόμβων του δικτύου.

Τέλος στο [8] εξετάζεται μια ξεχωριστή μεθοδολογία αναγνώρισης πρωτοκόλλων όχι για εισαγωγή νέων συσκευών αλλά για σάρωση του δικτύου, η οποία εφαρμόζεται σε δίκτυα IEEE 802.11 και IEEE 802.15.4, και εκμεταλλεύεται το γεγονός ότι τα δίκτυα αυτά συμπεριφέρονται διαφορετικά όταν διευρύνονται σε γρήγορες ταχύτητες ή με μεγάλα πακέτα διερεύνησης με αποτέλεσμα μεγάλες RTT (Round Trip Time) καθυστερήσεις και υψηλές απώλειες. Η προσέγγιση στηρίζεται σε τρία βήματα. Αρχικά πραγματοποιείται μια αργή σάρωση σε όλη τη περιοχή και γίνεται καταμέτρηση των χρόνων RTT. Έπειτα ο αλγόριθμος σάρωσης που χρησιμοποιείται επιλέγει μερικούς κόμβους και εκτελεί αργή σάρωση με πακέτα μεγέθους 32 και 512 bytes, ενώ γίνεται ξανά μέτρηση του χρόνου RTT. Λόγω του μεγάλου χρόνου RTT στα πακέτα των 512 bytes η αργή σάρωση μπορεί να καταλάβει τους κόμβους του δικτύου IEEE 802.15.4. Στο τελευταίο βήμα ένα μέρος της διεύθυνσης που περιέχει πακέτα IEEE 802.15.4 ξανα σαρώνεται σε αργή ταχύτητα με μικρά πακέτα των 32 byte.

2.1.δ Σημασιολογική διαλειτουργικότητα σε ετερογενή δίκτυα

Η σημασιολογική διαλειτουργικότητα είναι ένα από τα πιο σημαντικά “ανοιχτά” ζητήματα στον τομέα της διαχείρισης της γνώσης [12]. Γενικότερα η διαλειτουργικότητα παραμένει ένα δύσκολο κομμάτι στους προγραμματιστές των συστημάτων IoT. Αυτό συμβαίνει λόγω της μεγάλης ετερογένειας μεταξύ των IoT συσκευών στα πρωτόκολλα επικοινωνίας, στη μορφή των δεδομένων καθώς και τη χρήση διαφορετικών τεχνολογιών. Προέκταση αυτών είναι και η έλλειψη εδραίωσης και εφαρμογής παγκόσμιων προτύπων [14]. Η σημασιολογική διαλειτουργικότητα επιτρέπει στις λύσεις να ανταλλάσσουν δεδομένα μεταξύ διεργασιών και εφαρμογών με έναν πιο τυποποιημένο τρόπο [19].

Για τη διευκόλυνση της διαλειτουργικότητας έχουν προταθεί διάφορες μέθοδοι. Όπως για παράδειγμα η πρόταση του [12], στην οποία εφαρμόζεται ένα framework το οποίο κάνει χρήση ενός επικοινωνιακού μοντέλου που ονομάζεται σημασιολογικό τρίγωνο, κατά το οποίο πράκτορες επικοινωνίας μοιράζονται αναφορές (πραγματικά αντικείμενα) και όχι έννοιες (νοητική εικόνα), διασφαλίζοντας έτσι μια αποτελεσματική σημασιολογική διαλειτουργικότητα όσον αφορά τη μετάδοση πληροφοριών.

Μια ακόμα πρόταση είναι η προσέγγιση με χρήση οντολογίας που χρησιμοποιείται στα [13],[15], [21]. Η οντολογία μπορεί να θεωρηθεί ως το μέσο ενοποίησης δεδομένων που προέρχονται από διαφορετικά στοιχεία του συστήματος σε μια καθολική κατανόηση, επιτυγχάνοντας έτσι διαλειτουργικότητα σε σημασιολογικό επίπεδο [13]. Με την εφαρμογή της οντολογίας οι έννοιες θα γίνουν σημασιολογικά συνδεδεμένες με τις αναφερόμενες τους πραγματικές οντότητες, και έπειτα με την επεξεργασία των εννοιών αυτών μπορούν να παραχθούν νέα γεγονότα. Τέλος με τη συσχέτιση των σχημάτων δεδομένων στην οντολογία, τα αποθηκευμένα δεδομένα εμπλουτίζονται με σημασιολογικές συμβάσεις που ισχύουν στον τομέα της εφαρμογής. Μια παρόμοια τεχνική κατηγοριοποίησης χρησιμοποιείται και στο [14], στο οποίο δεδομένα ενός ασθενή αναλύονται σημασιολογικά με σκοπό να κατηγοριοποιηθούν σε μια ασθένεια.

Στο [24], γίνεται η παρουσίαση ενός μοντέλου που λειτουργεί με τη χρήση της εμπιστοσύνης. Η εμπιστοσύνη ορίζεται από παράγοντες όπως η φήμη, η εμπειρία και η γνώση REK (Reputation Experience Knowledge). Όταν ζητηθεί μια υπηρεσία τότε ο χειριστής ελέγχει τις άδειες της υπηρεσίας. Εκεί όσες λειτουργίες δεν πληρούν ένα συγκεκριμένο ποσοστό εμπιστοσύνης δεν αδειοδοτούνται. Όταν καθοριστεί η άδεια ο

χειριστής θα ελέγξει τη διαθεσιμότητα της υπηρεσίας, και σε περίπτωση που αυτή είναι, ελέγχεται και το ιστορικό επικοινωνίας με σκοπό να βρεθεί η τιμή εμπιστοσύνης της σύνδεσης. Τέλος γίνεται επαναξιολόγηση της τιμής εμπιστοσύνης και ανανεώνονται οι τιμές.

Τέλος, υπάρχει και η περίπτωση του [16], στο οποίο η σημασιολογία απευθύνεται κυρίως σε χαρακτηριστικά υπηρεσιών με σκοπό να διερευνάται αν κάποια υπηρεσία μπορεί να υλοποιηθεί από το WSN.

2.2 Συγχώνευση δεδομένων με δενδρική προσέγγιση

Στο [32] έχει γίνει η πρόταση μιας αρχιτεκτονικής, που αναζητά υπηρεσίες IoT μέσω ιεραρχικών κόμβων που περιέχουν έξυπνους χώρους όπου οι συσκευές εκθέτουν τις υπηρεσίες τους. Ο μηχανισμός αναζήτησης χρησιμοποιεί ομαδοποίηση και συγκέντρωση πληροφοριών που βασίζονται σε μια ψευδό-μετρική. Η μέθοδος ανακαλύπτει όλες τις υπηρεσίες που απαντούν σε ένα αίτημα και επιφέρει καλή κλιμάκωση, όσον αφορά το κόστος σε σχέση με άλλες προσεγγίσεις. Επιπλέον, περιορίζει τον αριθμό των κόμβων στους οποίους αποστέλλεται το αίτημα βάσει της περιγραφής του. Ως αποτέλεσμα η πιθανότητα πλημμύρας (flooding) όλων των κόμβων μειώνεται. Το μειονέκτημα είναι ότι είναι υπολογιστικά απαιτητικό σε ένα δυναμικό περιβάλλον.

Μια επιπλέον πρόταση έγινε στο [34], όπου παρουσιάστηκε ένας τρόπος κατασκευής δέντρου ευρετηρίου υψηλής απόδοσης, στο οποίο εδραιώνονται οι αισθητήρες σε μια ιεραρχική δενδρική δομή. Στη δομή αυτή οι κόμβοι αισθητήρων βρίσκονται σε λοξή κατανομή. Οι κύριες συνεισφορές της μεθόδου είναι ο συνδυασμός γειτονικών υπο περιοχών όπου οι αποστάσεις προώθησης μηνυμάτων μεταξύ δύο υπο περιοχών, που θα ενωθούν, είναι οι μικρότερες. Αυτό έχει ως αποτέλεσμα τα ερωτήματα που θα χρησιμοποιούν το δέντρο να καταναλώνουν μικρότερη ενέργεια. Παρόλα αυτά η μέθοδος αυτή δε λαμβάνει υπόψη της την ετερογένεια των κόμβων.

Επιπρόσθετα, οι Ramachandran, G.S. et. al. (2016) [33] πρότειναν το HitchHiker ως μοντέλο δέσμευσης στοιχείων για την υποστήριξη συγχώνευσης δεδομένων πολλαπλού άλματος και διαχείριση των δεδομένων αυτών στο IoT. Σε αυτόν τον μηχανισμό, οι σύνδεσμοι ταξινομούνται σε υψηλής ή χαμηλής προτεραιότητας. Οι δεσμεύσεις των στοιχείων παρέχουν τα metadata, ώστε να δημιουργηθεί η συγχώνευση πολλαπλού άλματος των δεδομένων. Ένας κεντρικός μετα-διαχειριστής (meta-manager) χρησιμοποιείται για την ανακάλυψη του μονοπατιού στο δίκτυο πολλαπλών αλμάτων, με σκοπό να υποστηρίξει τη δρομολόγηση κυκλοφορίας χαμηλής προτεραιότητας. Επιπλέον, αυτός ο μηχανισμός έχει

ορισμένα οφέλη όπως η κατανάλωση ενέργειας και η βελτίωση της καθυστέρησης καθώς και η ανοχή στην απώλεια δεδομένων (data loss). Όμως και εδώ δεν λαμβάνεται υπόψη η ετερογένεια των κόμβων του δικτύου.

Στο [35] προτείνεται ένα μοντέλο στο οποίο τα δεδομένα φιλτράρονται χρησιμοποιώντας μια πληθώρα παραμέτρων και έπειτα στέλνονται μαζί με μια fitting function η οποία θα συσχετίζει όλες τις παραμέτρους μεταξύ τους. Αυτό έχει ως σκοπό, να μην χρειάζεται να σταλούν όλα τα δεδομένα από τον συσσωρευτή (aggregator) στον sink αλλά να στέλνεται μόνο ένα μέρος των δεδομένων μαζί με τη συνάρτηση που εκφράζει όλες τις παραμέτρους. Έτσι τα υπόλοιπα δεδομένα μπορούν να υπολογιστούν στον sink με τη χρήση της συνάρτησης. Το μοντέλο δοκιμάστηκε χρησιμοποιώντας ένα Bayesian Belief Network και έδειξε μεγάλα ποσοστά ακρίβειας.

Στο [37], παρουσιάζεται ένα Ενεργειακά Αποδοτικό Σχέδιο Δρομολόγησης (EERS) για ασύρματα δίκτυα αισθητήρων, το οποίο περιέχει ιεραρχικό πρωτόκολλο δρομολόγησης. Το πρωτόκολλο αυτό χωρίζεται σε δύο φάσεις. Τη φάση της δημιουργίας των συστάδων και τη φάση μετάδοσης των δεδομένων. Κατά τη συσταδοποίηση οι κόμβοι ομαδοποιούνται σε διαφορετικές συστάδες, όπου η κάθε μια έχει μια κεφαλή (cluster-head). Η κεφαλή αυτή είναι υπεύθυνη για τη συγχώνευση των δεδομένων που έχουν συλλεχθεί από τους αισθητήρες. Η δημιουργία των συστάδων γίνεται με τη χρήση του αλγορίθμου LEACH. Έπειτα ένα δενδρικό δίκτυο διαμορφώνεται χρησιμοποιώντας όλες τις κεφαλές των συστάδων. Για να υπολογιστεί η εμβέλεια των κόμβων γίνεται η χρήση ενός MST (Minimum Spanning Tree). Κατά τη μετάδοση των δεδομένων η κεφαλή μεταδίδει το token της σε όλους τους γείτονές της επαναληπτικά μέχρι αυτό να επιστρέψει σε αυτή. Όταν το token επιστραφεί η κεφαλή συλλέγει όλη της πληροφορία από τον κόμβο όπως το αναγνωριστικό του και τη τοπολογία του. Τότε η κεφαλή εκτελεί τον αλγόριθμο SHORT με σκοπό να δημιουργήσει ζευγάρια επικοινωνίας.

Μια ακόμη πρόταση γίνεται από το [36], η οποία βασίζεται στην κατηγορία περιοδικού ανά άλμα (per-hop), μοντέλου χρονισμού συγκέντρωσης δεδομένων κατά μήκος κάθε υπο-δέντρου. Στην κατηγορία περιοδικών per-hop μοντέλων χρονισμού, ένας κόμβος στέλνει το συγκεντρωτικό πακέτο μόλις αυτό ακούσει από όλα τα παιδιά του. Έτσι και σε αυτή τη προσέγγιση η ρίζα θα στείλει το τελικό αυτό μήνυμα, το οποίο θα περιέχει τους συντελεστές. Κάθε κόμβος στο δίκτυο έχει ένα μοναδικό αναγνωριστικό id ενώ ακόμα περιέχει πληροφορίες τοποθεσίας μέσω τριγωνισμού. Η δενδρική μορφή επιτυγχάνεται μέσω πολλαπλών διαχωρισμών δέντρων που βασίζονται σε χαρακτηριστικά των κόμβων (query trees) χρησιμοποιώντας τον αλγόριθμο QT_FORM που αναπτύχθηκε από τους συγγραφείς.

Ουσιαστικά κάθε κόμβος συγκεκριμένου επιπέδου μεταδίδει ένα πακέτο τύπου BEACON στους υπόλοιπους κόμβους του ίδιου επιπέδου. Ένας κόμβος μεγαλύτερου επιπέδου επιλέγει τυχαία έναν κόμβο ως τον parent node των υπόλοιπων στέλνοντάς του έναν μήνυμα PROBE. Ο παραλήπτης του PROBE περιμένει με τη σειρά του να παραλάβει αντίστοιχα μηνύματα PROBE από τα παιδιά του ώστε να επιλέξει 2 για παιδιά του. Αφού ο πατέρας επιλέξει τα παιδιά, τους στέλνει ένα μήνυμα JOIN ώστε να τα εισάγει στο δέντρο. Τα αποτελέσματα έδειξαν ότι η κυκλοφοριακή συμφόρηση μειώθηκε σημαντικά σε σχέση με αντίστοιχες άλλες προσεγγίσεις.

ΚΕΦΑΛΑΙΟ 3: Αλγόριθμος διαχείρισης ετικετών σε ετερογενή δίκτυα

3.1 Περιγραφή αλγορίθμου

Πολλές αρχιτεκτονικές συστημάτων IoT, βασίζονται στην επικοινωνία των συσκευών με ανταλλαγή μηνυμάτων JSON, ή παρόμοιων προτύπων ετικέτας-τιμής [2],[9],[10],[24],[42]. Παρόλο που η χρήση του προτύπου χρησιμοποιείται, μαζί με άλλες τεχνολογίες, για να περιορίσει την ετερογένεια στο επικοινωνιακό κομμάτι του δικτύου, δεν γίνεται περαιτέρω επεξεργασία για την ομαδοποίηση και παρουσίαση των δεδομένων του δικτύου σε κατανοητή μορφή.

Ο αλγόριθμος που σχεδιάσαμε στοχεύει στην δημιουργία ενός φυλογενετικού δέντρου το οποίο θα περιέχει στη ρίζα του ένα υπερσύνολο όλων των ετικετών των αισθητήρων του δικτύου του. Το δέντρο θα χτίζεται από κάτω προς τα πάνω. Πιο χαμηλά θα βρίσκονται τα ξεχωριστά ζευγάρια ετικετών που έχουν τη μικρότερη απόσταση μεταξύ τους. Η δημιουργία του θα πραγματοποιείται στο κόμβο Sink και με αυτό το τρόπο ο κόμβος θα μπορεί να έχει μια πλήρη σχεσιακή εικόνα των αισθητήρων του δικτύου. Τα δεδομένα που παράγονται από τους αισθητήρες θα αποθηκεύονται σε μια λίστα η οποία θα μπορεί να χρησιμοποιηθεί αργότερα για παραχθούν στατιστικά για τη λειτουργία του δικτύου αλλά και να αποθηκευτούν σε μια βάση δεδομένων. Ο αλγόριθμος υποθέτει ότι ο κόμβος sink θα λαμβάνει ένα μήνυμα μορφής JSON όπως στα άρθρα [2],[9],[10],[24],[42] ή γενικότερα μήνυμα το οποίο θα περιέχει την ετικέτα που θα προσδιορίζει τι ανιχνεύτηκε και την ανιχνεύσιμη τιμή.

3.1.α Υπολογισμός κοντινότερου ζευγαριού

Ο τρόπος εύρεσης των πιο κοντινών ζευγαριών ετικετών είναι ισχυρά εμπνευσμένος από την τεχνική Feng-Doolittle multiple alignment [43], που χρησιμοποιείται για την αντιστοίχιση αλληλουχιών νουκλεοτιδίων. Οι ετικέτες συγκρίνονται μεταξύ τους μέσω αλγορίθμων σύγκρισης συμβολοσειρών (στο πείραμα χρησιμοποιήθηκαν οι hamming και levenshtein) και οι τιμές των διαφορών τους αποθηκεύονται σε ένα διαγώνιο πίνακα [43]. Επειτα βρίσκουμε το κοντινότερο ζευγάρι, δηλαδή αυτό που έχει τη μικρότερη τιμή, και τις συντεταγμένες του στο πίνακα. Αν χρησιμοποιούσαμε την απόσταση hamming για τον υπολογισμό του πίνακα διαφορών θα είχαμε:

$$distance[n][n] = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} hamming(tag_i, tag_j)$$

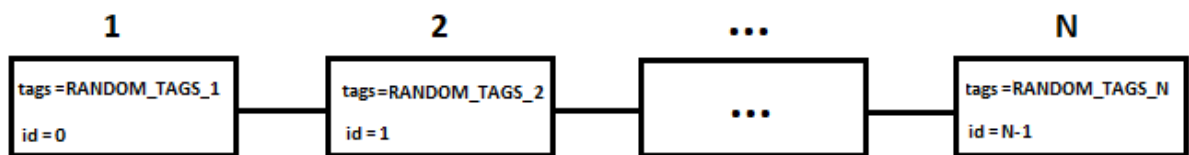
το οποίο παράγει τον πίνακα:

$$\begin{bmatrix} 0 & d_{01} & d_{02} & \dots & d_{0n} \\ 0 & 0 & d_{12} & \dots & d_{1n} \\ \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & d_{(n-1)n} \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}$$

Χρησιμοποιούμε μια τροποποιημένη εκδοχή της σειράς για να μετατρέψουμε τον πίνακα σε άνω τριγωνικό αντί για διαγώνιο, ώστε να εξαλείψουμε τον υπολογισμό επαναλαμβανόμενης πληροφορίας στο πίνακα και να εξοικονομήσουμε έτσι περισσότερο χρόνο και πόρους από τους είδη χαμηλούς σε ισχύ κόμβους. Επειδή οι συσκευές IoT είναι σε μεγάλο βαθμό χαμηλής υπολογιστικής δυνατότητας, η αποφυγή των αχρείαστων υπολογισμών απόστασης είναι αδήριτη ανάγκη.

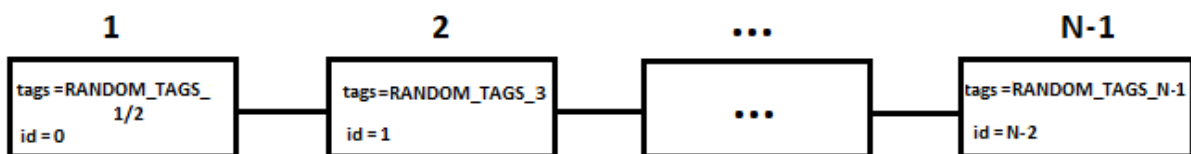
3.1.β Αντιστοίχιση ετικετών με τον πίνακα αποστάσεων

Οι ετικέτες αποθηκεύονται σε μια λίστα μαζί με ένα αναγνωριστικό id το οποίο κυμαίνεται στις τιμές 0 - (N-1), όπου N ο αριθμός ετικετών στη λίστα. Ο αριθμός id χρησιμοποιείται για την αντιστοίχιση των ετικετών στον πίνακα διαφορών. Δηλαδή η ετικέτα με id = 0, θα βρίσκεται στη γραμμή 0 και στη στήλη 0 και αντίστοιχα η ετικέτα με id = n-1 θα βρίσκεται στη στήλη και γραμμή n-1.



Εικόνα 1. Λίστα με ετικέτες πριν τη συγχώνευση

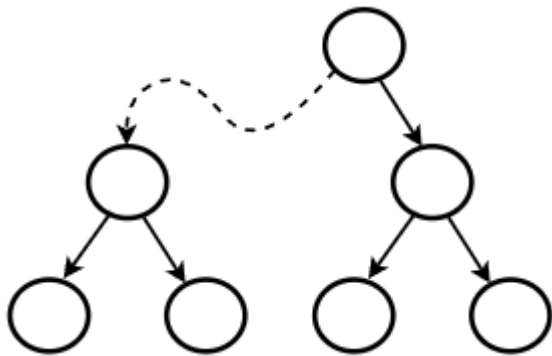
Όταν γίνει η συγχώνευση ετικετών διαγράφονται από τη λίστα οι ετικέτες και τη θέση τους παίρνει ένας νέος κόμβος με τον συνδυασμό των ετικετών.



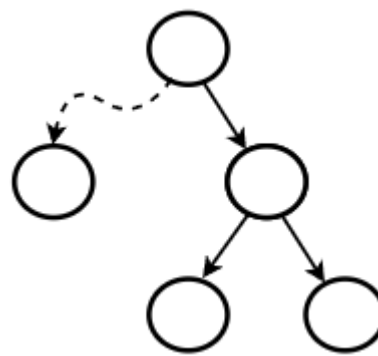
Εικόνα 2. Λίστα με ετικέτες μετά τη συγχώνευση

3.1.γ Εισαγωγή κόμβων στο δέντρο

Οι ετικέτες του κοντινότερου ζευγαριού εισάγονται ως φύλλα στο δέντρο, ενώ η ένωσή τους εισάγεται ως ο πατέρας τους. Σε περίπτωση που στο κοντινότερο ζευγάρι δεν περιέχεται ο συνδυασμός ετικετών της ρίζας, δημιουργείται ένα καινούργιο υποδέντρο με τις ετικέτες αυτές. Η ρίζα του υποδέντρου αυτού συνδυάζεται με τη ρίζα του ήδη υπάρχοντος δέντρου για να δημιουργήσουν τη νέα ρίζα του δέντρου όπως φαίνεται στην Εικόνα 3.



Εικόνα 3. Ενσωμάτωση υποδέντρου κοντινότερου ζευγαριού στο είδη υπάρχον δέντρο.



Εικόνα 4. Ενσωμάτωση κοντινότερου κόμβου στο δέντρο

Η διαδικασία συνεχίζεται μέχρι να υπάρχει μόνο ένας κόμβος στη λίστα ετικετών. Τότε έχουμε παράξει το τελικό υποσύνολο των ετικετών το οποίο υπάρχει και στη ρίζα του δέντρου. Το τελικό δέντρο μπορεί να διαφέρει ανάλογα με ποια απόσταση θα χρησιμοποιήσουμε

3.2 Παρουσίαση αλγορίθμου

Ο αλγόριθμος που υλοποιεί τη κατασκευή του δέντρου παρουσιάζεται παρακάτω.

Algorithm 1 Tree generation

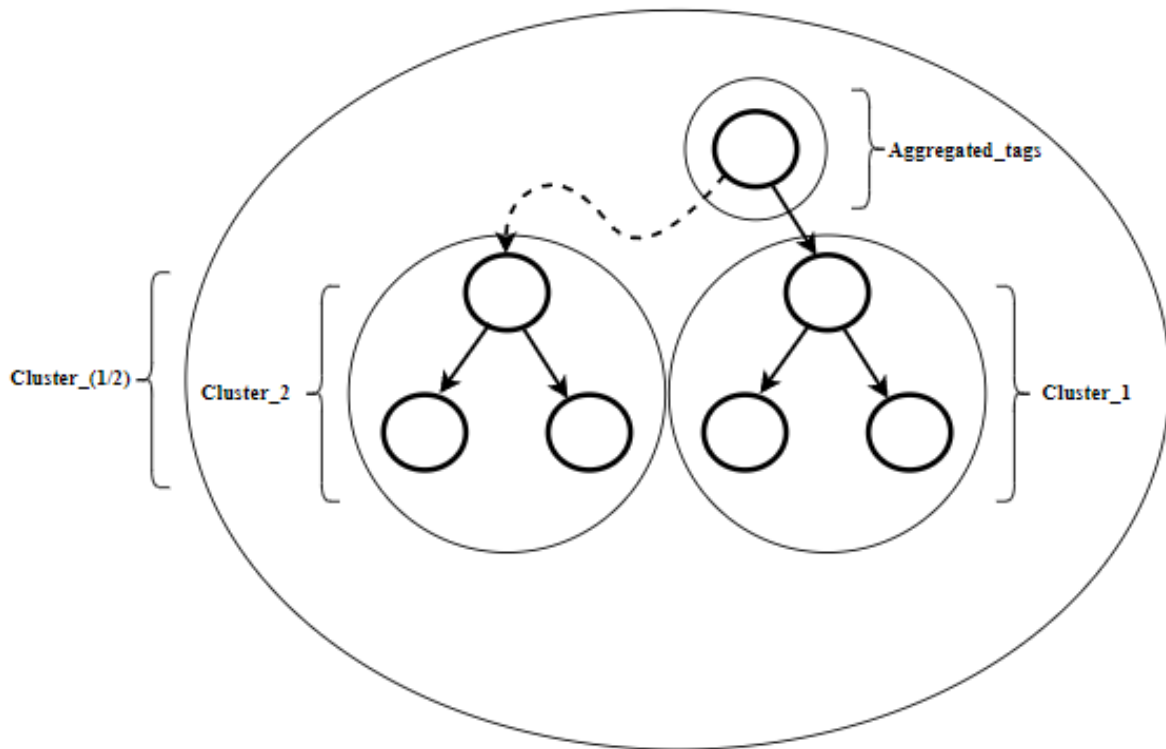
Input: $Tags_1 \dots Tags_N$

Output: *Pointer to phylogenetic tree*

```
1: function TREEGENERATION( $N, Tags$  [ [ ] ])
2:   tagsList  $\leftarrow$  NULL
3:   treeHead  $\leftarrow$  NULL
4:   insertInList(&tagsList, Tags [ [ ] ])
5:   differences[n][n]  $\leftarrow$  {0}
6:   minVal  $\leftarrow$  FLOAT_MAX
7:   while tableSize  $\neq$  1 do
8:     minVal  $\leftarrow$  FLOAT_MAX
9:     minRow  $\leftarrow$  -1
10:    minCol  $\leftarrow$  -1
11:    calculate_differences(tagsList, differences)
12:    find_min_val(&minVal, &minRow, &minCol)
13:    firstCoupleTag  $\leftarrow$  find_tag_with_id(tagsList, *minRow)
14:    secondCoupleTag  $\leftarrow$  find_tag_with_id(tagsList, *minCol)
15:    combinationOfTags  $\leftarrow$  combine(firstCoupleTag, secondCoupleTag)
16:    parent  $\leftarrow$  create_tree_node(combinationOfTags)
17:    firstTagExists  $\leftarrow$  search_tree(treeHead, firstCoupleTags)
18:    secondTagExists  $\leftarrow$  search_tree(treeHead, secondCoupleTags)
19:    if treeHead == NULL then
20:      insertRight(parent, firstCoupleTag)
21:      insertLeft(parent, secondCoupleTag)
22:      treeHead  $\leftarrow$  parent
23:    else if firstTagExists AND secondTagExists then
24:      pass /* do nothing */
25:    else if firstTagExists then
26:      insertLeft(parent, secondCoupleTag)
27:      insertParentRight(parent, treeHead)
28:      treeHead  $\leftarrow$  parent
29:    else if secondTagExists then
30:      insertLeft(parent, firstCoupleTag)
31:      insertParentRight(parent, treeHead)
32:      treeHead  $\leftarrow$  parent
33:    else
34:      insertLeft(parent, secondCoupleTag)
35:      insertRight(parent, firstCoupleTag)
36:      superCombinationOfTags  $\leftarrow$  combine(parent  $\rightarrow$  tags, treeHead  $\rightarrow$  tags)
37:      superParent  $\leftarrow$  create_tree_node(superCombinationOfTags)
38:      insertParentLeft(superParent, parent);
39:      insertParentRight(superParent, treeHead);
40:      treeHead  $\leftarrow$  superParent
41:    end if
42:    delete_from_list(tagsList, minCol)
43:    delete_from_list(tagsList, minRow)
44:    insertInList(&tagsList, combinationOfTags)
45:    update_indexes(tagsList)
46:    tableSize  $\leftarrow$  -1
47:  end while
48:  return treeHead
49: end function
```

Εικόνα 5. Αλγόριθμος υλοποίησης φυλογενετικού δέντρου.

Κάθε κόμβος του δέντρου είναι ουσιαστικά μια ετικέτα ή ένα σύνολο ομαδοποιημένων ετικετών. Η ιεραρχία των κόμβων δείχνει ότι κάθε υποδέντρο αποτελεί τη δική του συστάδα με τα πιο όμοια σε αυτή ζευγάρια να βρίσκονται κοντινότερα στη ρίζα της.



Εικόνα 6. Συστάδες που προκύπτουν κατά τη δημιουργία του δέντρου.

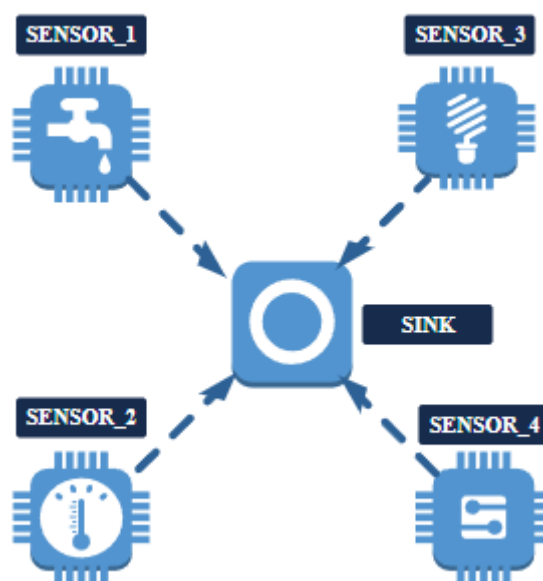
ΚΕΦΑΛΑΙΟ 4: Αποτελέσματα αξιολόγησης του αλγορίθμου

Στο κεφάλαιο αυτό θα αναλύσουμε το περιβάλλον προσομοίωσης και τα αποτελέσματα από την εκτέλεση του αλγορίθμου.

Για τη διεξαγωγή του πειράματος, δημιουργήσαμε ένα περιβάλλον χρησιμοποιώντας τον προσομοιωτή του Cooja Contiki, με σκοπό να αξιολογήσουμε την απόδοση του αλγορίθμου στους άξονες της χρονικής εκτέλεσης και της ενέργειας που ο κόμβος θα σπαταλήσει για να δημιουργήσει το δέντρο καθώς και για να διαχειριστεί τα δεδομένα που του εισέρχονται. Ο αλγόριθμος αναπτύχθηκε σε γλώσσα C.

4.1 Περιβάλλον προσομοίωσης

Όπως αναφέρθηκε με τη χρήση του προσομοιωτή contiki δημιουργήθηκε ένας χώρος όπου περιείχε αντίστοιχα 4,8,13 κόμβους (εξαιρούμενου του κόμβου sink) που αποστέλλουν διαφορετικά δεδομένα χρησιμοποιώντας μορφή παρόμοια της JSON. Χρησιμοποιήθηκε μορφολογία αστέρα για τη τοποθέτηση των αισθητήρων στο δίκτυο. Τα δεδομένα που παράγονται από τους κόμβους είναι τυχαία και στέλνονται μέσω του best effort local area signal που παρέχει η βιβλιοθήκη Rime του contiki. Το πρωτόκολλο που χρησιμοποιείται από τη βιβλιοθήκη αυτή είναι το UDP, ενώ γίνεται ακόμα χρήση του rime-stack το οποίο αποτελεί μια ελαφριά και πολυεπίπεδη στοίβα επικοινωνίας για τα δίκτυα αισθητήρων.



Εικόνα 7. Δίκτυο τοπολογίας αστέρα

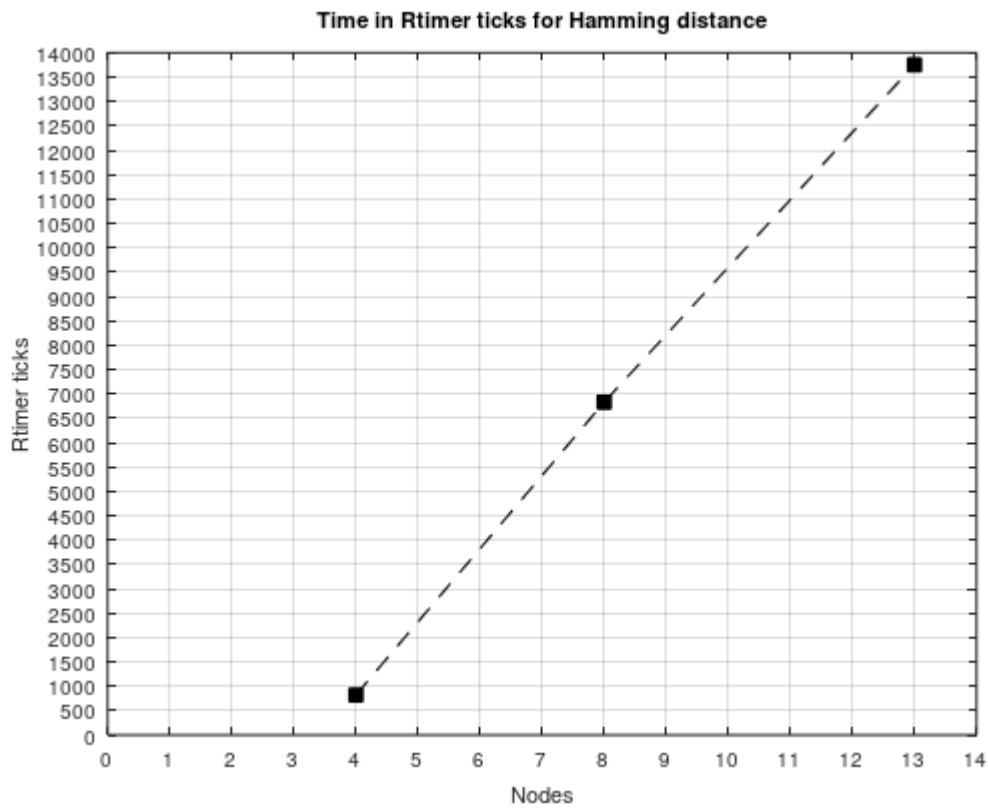
4.2 Παραγωγή δέντρου με αλγορίθμους μέτρησης απόστασης

Το τελικό δέντρο που θα παραχθεί μπορεί να έχει διαφορετική μορφή ανάλογα τον αλγόριθμο που θα χρησιμοποιηθεί για να μετρηθούν οι αποστάσεις μεταξύ των ετικετών των κόμβων. Η χρήση συγκεκριμένου αλγορίθμου μέτρησης απόστασης μπορεί να χρησιμοποιηθεί ώστε να παραχθεί η επιθυμητή μορφή δέντρου ανάλογα τη χρήση του.

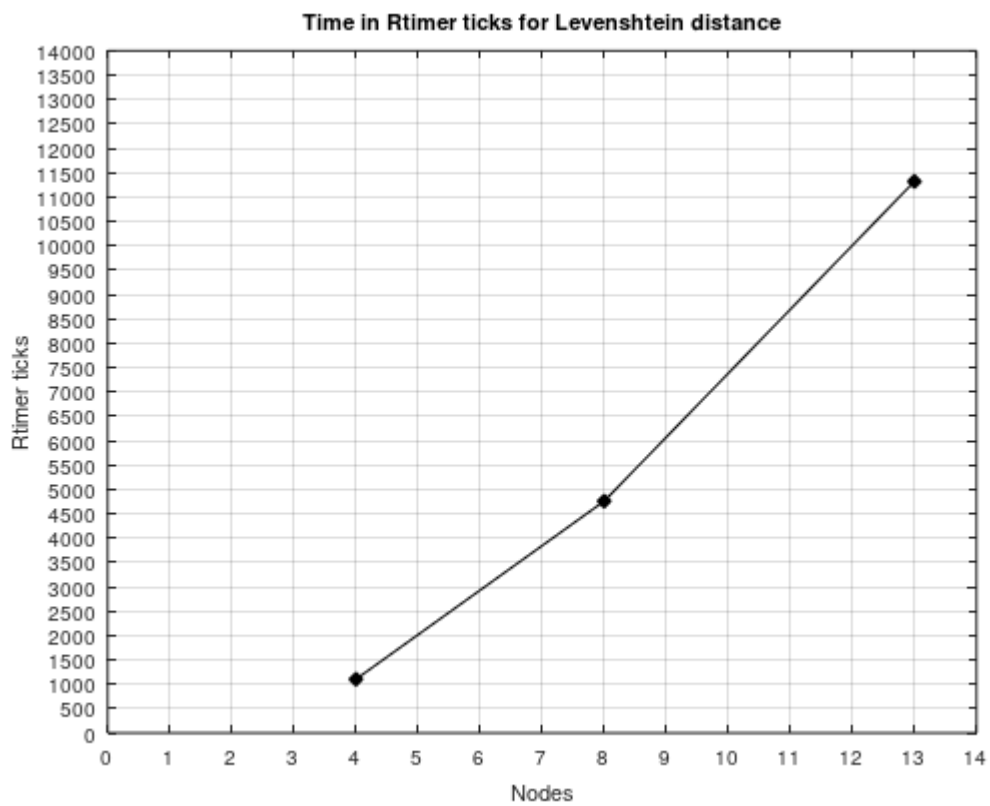
4.2.α Ταχύτητα δημιουργίας δέντρου

Στο περιβάλλον που δημιουργήσαμε πραγματοποιήσαμε μετρήσεις, χρησιμοποιώντας τις αποστάσεις hamming και levenshtein. Η απόσταση hamming έχει τροποποιηθεί, ώστε να λαμβάνει την απόκλιση του μεγέθους των συμβολοσειρών ως διαφορά, σε περίπτωση πάντα που οι συμβολοσειρές είναι διαφορετικού μήκους.

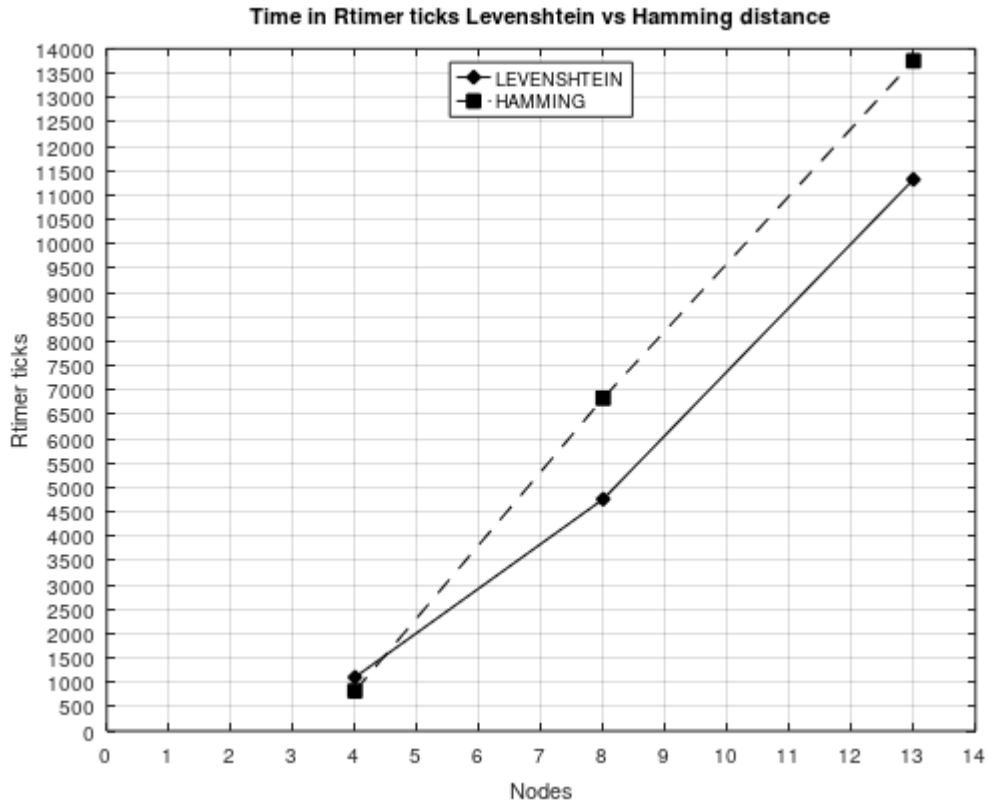
Για τη μέτρηση του χρόνου, χρησιμοποιήθηκε η βιβλιοθήκη του contiki-ng Rtimer. Η βιβλιοθήκη αυτή παρέχει μηχανισμούς προγραμματισμού (scheduling) σε πραγματικό χρόνο, συνήθως για εφαρμογές όπου μια απόκριση σε ένα εξωτερικό συμβάν είναι εξαιρετικά ευαίσθητη στο χρόνο. Τα Rtimers χρησιμοποιούν τον απόλυτο χρόνο ρολογιού του συστήματος ως αναφορά και πιο συγκεκριμένα σε ένα δευτερόλεπτο γίνονται 32678 rtimer ticks [38]. Οι μετρήσεις έδειξαν ότι η απόσταση hamming ήταν πιο αποδοτική από την levenshtein στη παραγωγή του δέντρου όταν αυτό περιείχε λιγότερους από 4 κόμβους. Σε αντίθεση η levenshtein εμφάνισε πολύ μικρότερους χρόνους όταν το δίκτυο είχε πάνω από 5 κόμβους και συγκεκριμένα 8 και 13.



Εικόνα 8. Χρόνος δημιουργίας δέντρου σε Rtimer ticks χρησιμοποιώντας hamming distance



Εικόνα 9. Χρόνος δημιουργίας δέντρου σε Rtimer ticks χρησιμοποιώντας levenshtein distance



Εικόνα 10. Σύγκριση χρόνων δημιουργίας ανάλογα τον αλγόριθμο απόστασης

4.2.β Κατανάλωση ενέργειας

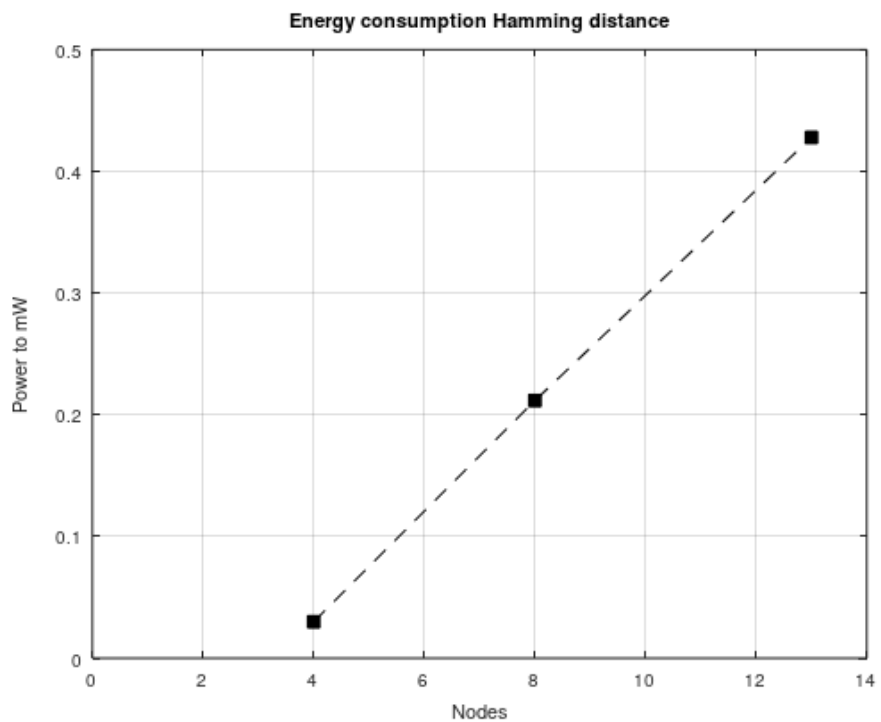
Μετρήσεις έγιναν ακόμα και για την κατανάλωση της ενέργειας του κόμβου sink για να τη παραγωγή του δέντρου, χρησιμοποιώντας τις δύο αυτές μετρικές διαφορών.

Για να υπολογιστεί η ενέργεια χρησιμοποιήθηκαν τα ticks της cpu όταν αυτή ήταν σε κανονική λειτουργία τα οποία εισήχθησαν στον ακόλουθο μαθηματικό τύπο:

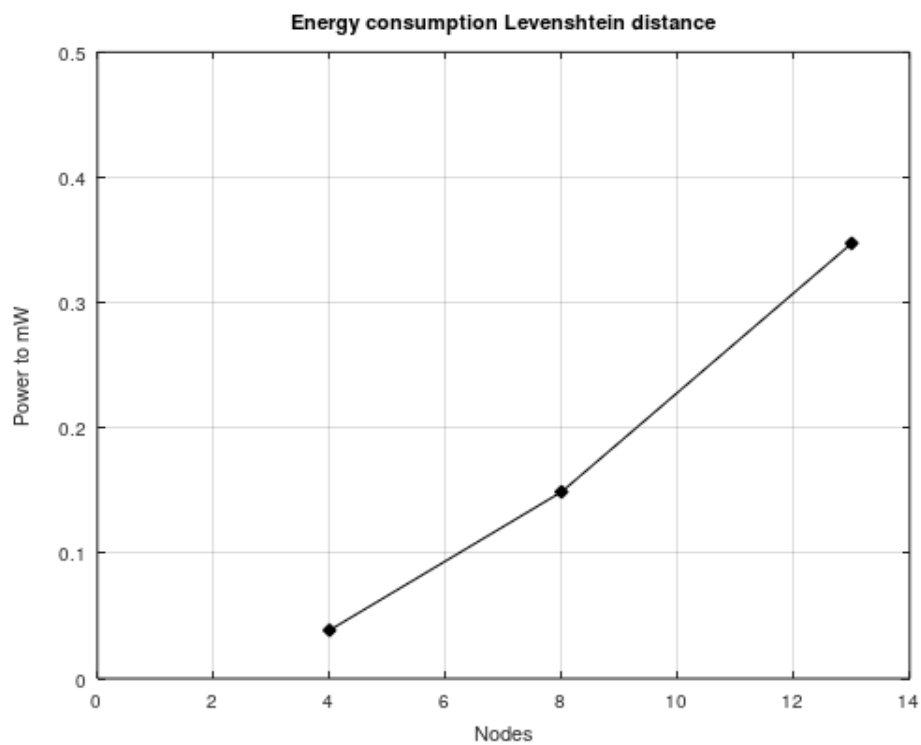
$$Power = \frac{ACTIVE\ CPU\ TICKS * CURRENT * VOLTAGE}{RTIMER\ SECOND}$$

Όπου το Rtimer Second είναι 32768, το Current και Voltage, σύμφωνα με το documentation του contiki για τα motes που χρησιμοποιήθηκαν είναι 0.33 Amber και 3 Volt αντίστοιχα [40]. Η βιβλιοθήκη που χρησιμοποιήθηκε για να γίνουν η μετρήσεις των Active CPU Ticks είναι η energest, η οποία παρέχει τρόπους μέτρησης και παρακολούθησης της λειτουργίας των εξαρτημάτων [39].

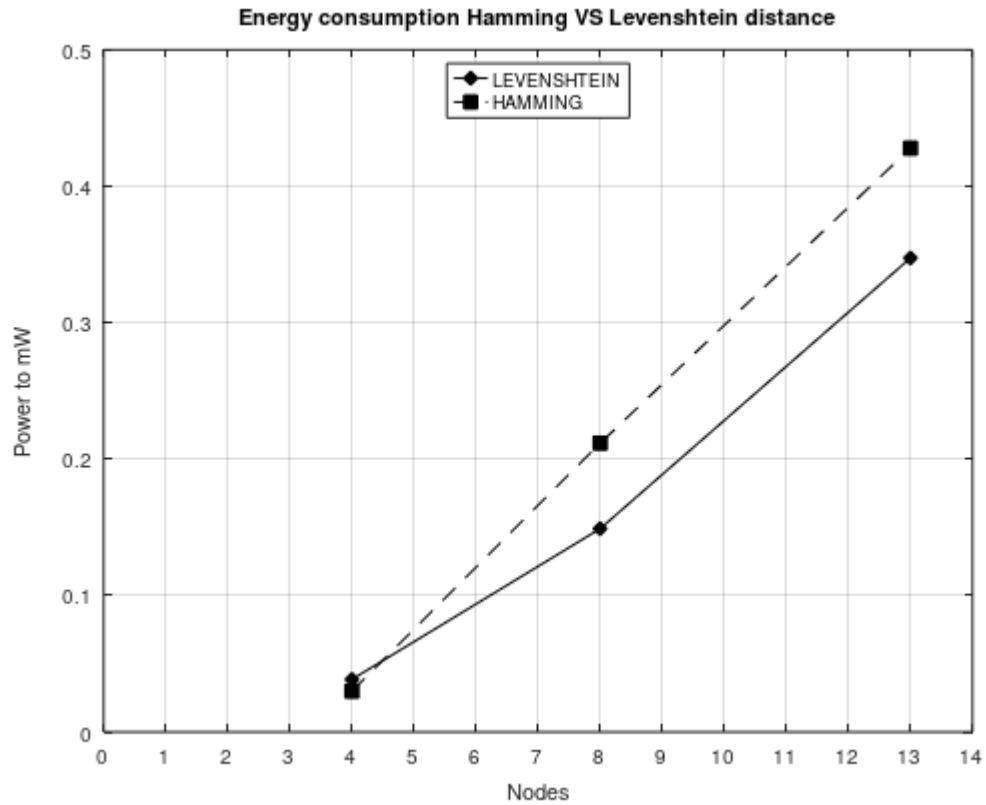
Και εδώ οι μετρήσεις έδειξαν σχεδόν πανομοιότυπη εικόνα με τις χρονικές, αφού χρησιμοποιώντας την απόσταση hamming για 4 nodes η κατανάλωση της ενέργειας ήταν μικρότερη σε σχέση με την levenshtein, ενώ όσο οι κόμβοι αυξάνονται η απόσταση levenshtein παρουσιάζει το πλεονέκτημα.



Εικόνα 11. Ενέργεια σε mW για τη δημιουργία του δέντρου σε Rtimer ticks χρησιμοποιώντας hamming distance

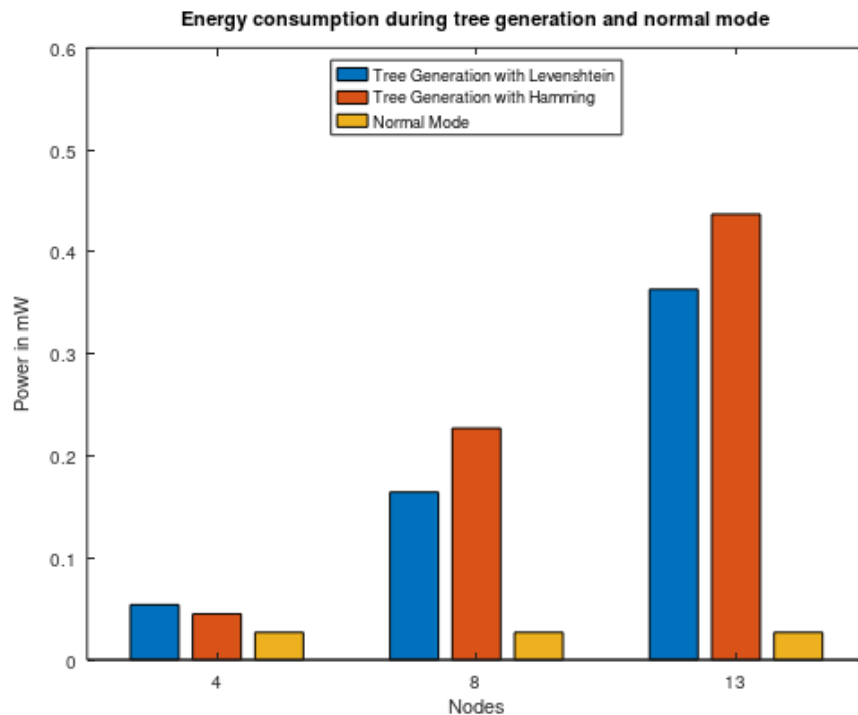


Εικόνα 12. Ενέργεια σε mW για τη δημιουργία του δέντρου σε Rtimer ticks χρησιμοποιώντας levenshtein distance



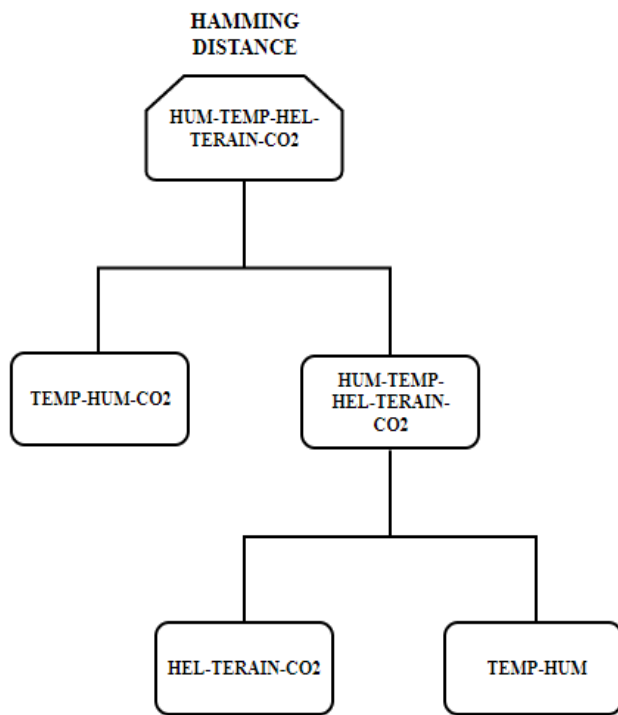
Εικόνα 13. Σύγκριση κατανάλωσης ενέργειας ανάμεσα στις 2 αποστάσεις

Σε κανονική λειτουργία, δηλαδή όταν ο sink έχει ήδη δημιουργήσει το δέντρο και οι πόροι του κατανέμονται μόνο στη διαχείριση των μηνυμάτων και της αποθήκευσης των τιμών, των μηνυμάτων αυτών, ο κόμβος καταναλώνει περίπου την ίδια ενέργεια ανεξάρτητα το πλήθος κόμβων που υπήρχαν στο δίκτυο.

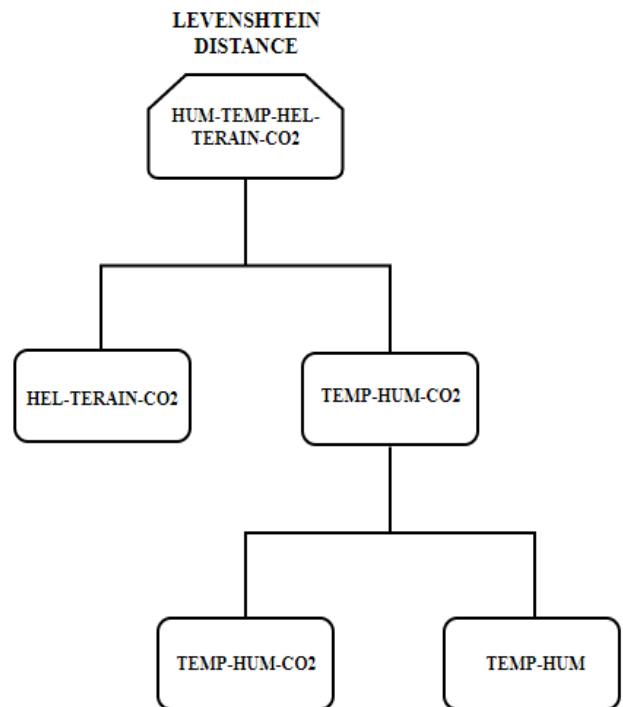


Εικόνα 14. Κατανάλωση ενέργειας του κόμβου κατά τη δημιουργία του δέντρου, ανάλογα με τον αλγόριθμο απόστασης, και σε κανονική λειτουργία

Η ενεργειακή και χρονική διαφορά συνιστά τη χρήση της απόστασης Levenshtein για μεγαλύτερου πλήθους δίκτυα, παρόλα αυτά η δομή του δέντρου που θα παραχθεί είναι διαφορετική ανάλογα την μετρική αποστάσεων που θα χρησιμοποιηθεί. Αν θέλουμε να εστιάσουμε σε μια συγκεκριμένη παραγωγή θα πρέπει να χρησιμοποιήσουμε την ανάλογη μετρική απόστασης συμβολοσειρών. Τις περισσότερες φορές το δέντρο θα έχει την ίδια δομή αλλά αυτό δε σημαίνει ότι το αποτέλεσμα θα είναι συνέχεια το ίδιο. Στις εικόνες 15,16 δίνεται ένα παράδειγμα ετικετών που δείχνει την παραπάνω δήλωση. Αν θέσουμε ως είσοδο τις ετικέτες “TEMP-HUM-CO2”, “HUM-TEMP” και “HEL-TERAIN-CO2” θα έχουμε παρακάτω αποτελέσματα. Αξίζει να σημειωθεί ότι το παράδειγμα είναι φτιαγμένο για να τονίσει την διαφορετική αυτή παραγωγή.



Εικόνα 15. Δέντρο που παράχθηκε από hamming distance



Εικόνα 16. Δέντρο που παράχθηκε από levenshtein distance

ΚΕΦΑΛΑΙΟ 5: Συμπεράσματα και μελλοντική έρευνα

Σε αυτή την εργασία παρουσιάζουμε έναν αλγόριθμο συγχώνευσης ετερογενούς δεδομένων, ο οποίος σχεδιάζει ένα φύλλο γενετικό δέντρο με τις ετικέτες των αισθητήρων του δικτύου. Η υλοποίηση χρησιμοποιεί μεθόδους εύρεσης ελάχιστης απόστασης, όπως η hamming και levenshtein distance, των ετικετών και τη δημιουργία συστάδων με βάση αυτές. Με τη δημιουργία του δέντρου πέρα από τη συγχώνευση των δεδομένων επιτυγχάνουμε και μια ιεραρχικά ομαδοποιημένη εικόνα των συλλεγόμενων δεδομένων του δικτύου. Η εικόνα αυτή μπορεί να χρησιμοποιηθεί ακόμα για αναδιοργάνωση των κόμβων του δικτύου, σε μια δενδρική τοπολογία για καλύτερη ενεργειακή κατανάλωση των κόμβων του δικτύου. Οι μετρήσεις ενέργειας και χρόνου υλοποίησης έδειξαν ότι ο αλγόριθμος είναι απαιτητικός σε υπολογιστική δύναμη γεγονός που δημιουργεί περιορισμούς στην επιλογή του aggregator node σε ένα κλασσικό, χαμηλής δύναμης δίκτυο IoT. Σε μελλοντική έρευνα θα διερευνηθεί περαιτέρω βελτιστοποίηση της πολυπλοκότητας του αλγορίθμου και χρήση περισσότερων μετρικών μέτρησης απόστασης. Τέλος θα εξεταστεί το ενδεχόμενο μεταφοράς του μοντέλου συγχώνευσης στο cloud όπου ο περιορισμός των πόρων δεν θα αποτελεί περιοριστικός παραγοντας. Ακόμα είναι ορθότερο, ως πρακτική, η συγχώνευση να γίνεται στο cloud παρά σε κάποιο sink ή gateway του δικτύου. Όλα αυτά μπορούν να βελτιώσουν την απόδοση και πρακτικότητα του αλγορίθμου, παρόλα αυτά θα χρειαστεί να υλοποιηθεί η έρευνα για να προβούμε σε συμπεράσματα.

BIBΛΙΟΓΡΑΦΙΑ

- [1] Moon, J., Kum, S., & Lee, S. (2019). A heterogeneous IoT data analysis framework with collaboration of edge-cloud computing: Focusing on indoor PM10 and PM2.5 status prediction. *Sensors*, 19(14), 3038.
- [2] Zhou, J., Leppanen, T., Harjula, E., Ylianttila, M., Ojala, T., Yu, C., ... & Yang, L. T. (2013, June). Cloudthings: A common architecture for integrating the internet of things with cloud computing. In *Proceedings of the 2013 IEEE 17th international conference on computer supported cooperative work in design (CSCWD)* (pp. 651-657). IEEE.
- [3] Aazam, M., & Huh, E. N. (2014, August). Fog computing and smart gateway based communication for cloud of things. In *2014 International Conference on Future Internet of Things and Cloud* (pp. 464-470). IEEE.
- [4] Kim, J. E., Boulos, G., Yackovich, J., Barth, T., Beckel, C., & Mosse, D. (2012, June). Seamless integration of heterogeneous devices and access control in smart homes. In *2012 Eighth International Conference on Intelligent Environments* (pp.206-213). IEEE.
- [5] Sowe, S. K., Kimata, T., Dong, M., & Zettsu, K. (2014, July). Managing heterogeneous sensor data on a big data platform: IoT services for data-intensive science. In *2014 IEEE 38th International Computer Software and Applications Conference Workshops* (pp. 295-300). IEEE.
- [6] Ma, M., Wang, P., & Chu, C. H. (2013, August). Data management for internet of things: Challenges, approaches and opportunities. In *2013 IEEE International conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing* (pp. 1144-1151). IEEE.
- [7] Galán-Jiménez, J., Berrocal, J., Garcia-Alonso, J., Canal-Velasco, J. C., & Murillo, J. M. (2017). Coordinating heterogeneous IoT devices by means of the centralized vision of the SDN controller.
- [8] Metongnon, L., & Sadre, R. (2018). Fast and efficient probing of heterogeneous IoT networks. *International Journal of Network Management*, 28(1), e1997.

- [9] Jara, A. J., Lopez, P., Fernandez, D., Castillo, J. F., Zamora, M. A., & Skarmeta, A. F. (2014). Mobile discovery: discovering and interacting with the world through the internet of things. *Personal and ubiquitous computing*, 18(2), 323-338.
- [10] Vargas, D. C. Y., & Salvador, C. E. P. (2016). Smart IoT gateway for heterogeneous devices interoperability. *IEEE Latin America Transactions*, 14(8), 3900-3906.
- [11] Tooba, M., Muhammad, A., & Martinez-Enriquez, A. M. (2018, October). Smart Solution for Heterogeneous Device Interoperability in IoT. In *2018 Seventeenth Mexican International Conference on Artificial Intelligence (MICAI)* (pp. 70-75). IEEE.
- [12] Amato, F., Mazzeo, A., Moscato, V., & Picariello, A. (2013, March). A framework for semantic interoperability over the cloud. In *2013 27th International Conference on Advanced Information Networking and Applications Workshops* (pp. 1259-1264). IEEE.
- [13] Bui, V., Brandt, P., Liu, H., Basten, T., & Lukkien, J. (2014, September). Semantic interoperability in body area sensor networks and applications. In *Proceedings of the 9th International Conference on Body Area Networks* (pp. 210-216).
- [14] Jabbar, S., Ullah, F., Khalid, S., Khan, M., & Han, K. (2017). Semantic interoperability in heterogeneous IoT infrastructure for healthcare. *Wireless Communications and Mobile Computing*, 2017.
- [15] Brandt, P., Basten, T., Stuiik, S., Bui, V., de Clercq, P., Pires, L. F., & van Sinderen, M. (2013, April). Semantic interoperability in sensor applications making sense of sensor data. In *2013 IEEE Symposium on Computational Intelligence in Healthcare and e-health (CICARE)* (pp. 34-41). IEEE.
- [16] Fok, C. L., Roman, G. C., & Lu, C. (2012). Servilla: a flexible service provisioning middleware for heterogeneous sensor networks. *Science of Computer Programming*, 77(6), 663-684.
- [17] Gama, K., Touseau, L., & Donsez, D. (2012). Combining heterogeneous service technologies for building an Internet of Things middleware. *Computer Communications*, 35(4), 405-417.

- [18] Jung, M., Weidinger, J., Kastner, W., & Olivieri, A. (2013, March). Building automation and smart cities: An integration approach based on a service-oriented architecture. In *2013 27th International Conference on Advanced Information Networking and Applications Workshops* (pp. 1361-1367). IEEE.
- [19] Avila, K., Sanmartin, P., Jabba, D., & Jimeno, M. (2017). Applications based on service-oriented architecture (SOA) in the field of home healthcare. *Sensors*, *17*(8), 1703.
- [20] Lan, L., Wang, B., Zhang, L., Shi, R., & Li, F. (2015). An Event-driven Service-oriented Architecture for the Internet of Things Service Execution. *International Journal of Online Engineering*, *11*(2).
- [21] Hong, Y. (2012, November). A resource-oriented middleware framework for heterogeneous internet of things. In *2012 international conference on cloud and service computing* (pp. 12-16). IEEE.
- [22] Razzaque, M. A., Milojevic-Jevric, M., Palade, A., & Clarke, S. (2015). Middleware for internet of things: a survey. *IEEE Internet of things journal*, *3*(1), 70-95.
- [23] GĂITAN, N. C., GĂITAN, V. G., PENTIUC, Ș. G., Ungurean, I., & Dodi, E. (2010). Middleware based model of heterogeneous systems for SCADA distributed applications. *Development and Application Systems*, *44*.
- [24] Abbasi, M. A., Memon, Z. A., Durrani, N. M., Haider, W., Laeeq, K., & Mallah, G. A. (2021). A multi-layer trust-based middleware framework for handling interoperability issues in heterogeneous IoTs. *Cluster Computing*, 1-28.
- [25] Krishnan, A. M., & Kumar, P. G. (2016). An effective clustering approach with data aggregation using multiple mobile sinks for heterogeneous WSN. *Wireless Personal Communications*, *90*(2), 423-434.
- [26] Aderohunmu, F. A., & Deng, J. D. (2009). An enhanced stable election protocol (sep) for clustered heterogeneous wsn. *Department of Information Science, University of Otago, New Zealand*.
- [27] *IOT data aggregation methods in WSN Landscapes: Unlocking more value*. Record Evolution. (2022, February 25). Retrieved May 11, 2022, from

<https://www.record-evolution.de/en/blog/iot-data-aggregation-methods-in-wireless-sensor-landscapes-on-making-data-valuable/>

[28] Rajagopalan, R., & Varshney, P. K. (2006). Data aggregation techniques in sensor networks: A survey.

[29] Ali, Z. H., Ali, H. A., & Badawy, M. M. (2015). Internet of Things (IoT): definitions, challenges and recent research directions. *International Journal of Computer Applications*, 128(1), 37-47.

[30] Jabbar, S., Ullah, F., Khalid, S., Khan, M., & Han, K. (2017). Semantic interoperability in heterogeneous IoT infrastructure for healthcare. *Wireless Communications and Mobile Computing*, 2017.

[31] Metongnon, L., & Sadre, R. (2018). Fast and efficient probing of heterogeneous IoT networks. *International Journal of Network Management*, 28(1), e1997.

[32] Fredj, S. B., Boussard, M., Kofman, D., & Noirie, L. (2013, August). A scalable IoT service search based on clustering and aggregation. In *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing* (pp. 403-410). IEEE.

[33] Ramachandran, G. S., Proença, J., Daniels, W., Pickavet, M., Staessens, D., Huygens, C., ... & Hughes, D. (2016). Hitch Hiker 2.0: a binding model with flexible data aggregation for the Internet-of-Things. *Journal of Internet Services and Applications*, 7(1), 1-15.

[34] Zhou, Z., Tang, J., Zhang, L. J., Ning, K., & Wang, Q. (2014). EGF-tree: an energy-efficient index tree for facilitating multi-region query aggregation in the internet of things. *Personal and ubiquitous computing*, 18(4), 951-966.

[35] Atoui, I., Ahmad, A., Medlej, M., Makhoul, A., Tawbe, S., & Hijazi, A. (2016, April). Tree-based data aggregation approach in wireless sensor network using fitting functions. In *2016 Sixth international conference on digital information processing and communications (ICDIPC)* (pp. 146-150). IEEE.

- [36] Banerjee, T., Chowdhury, K., & Agrawal, D. P. (2005, July). Tree based data aggregation in sensor networks using polynomial regression. In *2005 7th International Conference on Information Fusion* (Vol. 2, pp. 8-pp). IEEE.
- [37] Liang, C. K., Huang, Y. J., & Lin, J. D. (2008, March). An energy efficient routing scheme in wireless sensor networks. In *22nd International Conference on Advanced Information Networking and Applications-Workshops (aina workshops 2008)* (pp. 916-921). IEEE.
- [38] *Timers*. Timers - Contiki. (n.d.). Retrieved May 11, 2022, from <https://anrg.usc.edu/contiki/index.php/Timers>
- [39] Contiki-Ng. (n.d.). *Documentation: Energest · Contiki-ng/contiki-ng wiki*. GitHub. Retrieved May 11, 2022, from <https://github.com/contiki-ng/contiki-ng/wiki/Documentation:-Energest>
- [40] Sonhan. (n.d.). *Sonhan/contiki: My contiki code*. GitHub. Retrieved May 11, 2022, from <https://github.com/sonhan/contiki>
- [41] Jesus, P., Baquero, C., & Almeida, P. S. (2014). A survey of distributed data aggregation algorithms. *IEEE Communications Surveys & Tutorials*, 17(1), 381-404.
- [42] Anon, F., Navarathinarasah, V., Hoang, M., & Lung, C. H. (2014, September). Building a framework for internet of things and cloud computing. In *2014 IEEE International Conference on Internet of Things (iThings), and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)* (pp. 132-139). IEEE.
- [43] Durbin, R., Eddy, S., Krogh, A., & Mitchison, G. (1998). Multiple sequence alignment methods. In *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids* (pp. 135-160). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511790492.007