

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ  
ΕΙΔΙΚΕΥΣΗΣ (MSc)  
στην ΑΝΑΠΤΥΞΗ & ΑΣΦΑΛΕΙΑ  
ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**“Δρομολόγηση με βάση λέξεις-κλειδιά  
στο Διαδίκτυο των Πραγμάτων”**

**Κωνσταντίνος Δούμπας**

**p3312203**

**ΑΘΗΝΑ, ΙΟΥΛΙΟΣ 2024**

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

**SCHOOL OF INFORMATION  
SCIENCE & TECHNOLOGY  
MASTER OF SCIENCE (MSc) in  
INFORMATION SYSTEMS  
DEVELOPMENT AND SECURITY**

**MSc THESIS**

**“Keyword-based Routing for IoT (KIOT)”**

**Konstantinos Doumpas**

**p3312203**

**ATHENS, JULY 2024**

## Περιεχόμενα

Περίληψη .....	5
Abstract .....	6
Εισαγωγή .....	7
Περιγραφή ΚΙΟΤ .....	9
Στρατηγικές Λειτουργίας .....	9
Naive .....	9
Minimum Transfer .....	9
Είδη Κόμβων .....	10
Τύποι Πακέτων .....	10
Λειτουργία .....	12
Υλοποίηση και Βελτιώσεις .....	16
Guest Node .....	16
Στρατηγική Minimum-Transfer .....	16
Αναμονή αποτελεσμάτων .....	17
Έξοδος κόμβου από την τοπολογία .....	18
Υπολογισμός RTT για την εύρεση κόμβου .....	18
Ρυθμίσεις Bloom Filters .....	18
Logging .....	18
Βασικές Δομές-Components ΚΙΟΤ .....	19
Τύποι Μηνυμάτων .....	20
Παραμετροποίηση Κόμβων .....	20
Επεξεργασία δεδομένων .....	21
Πειράματα .....	22
Συμπεράσματα .....	25
Βιβλιογραφία .....	26

## Κατάλογος Εικόνων

Εικόνα 1. Είδη Στρατηγικών ΚΙΟΤ .....	9
Εικόνα 2. Αποστολή μηνυμάτων MSG_PARENT .....	12
Εικόνα 3. Αποστολή μηνυμάτων MSG_ADVERTISEMENT .....	12
Εικόνα 4. Εισαγωγή αιτήματος στο σύστημα .....	13
Εικόνα 5. Απάντηση προς Client .....	14
Εικόνα 6. Έξοδος κόμβου παιδιού .....	15
Εικόνα 7. Έξοδος γονικού κόμβου .....	15

Εικόνα 8. Minimum Transfer: υπολογισμός Average τιμής.....	17
Εικόνα 9. Αποτελέσματα πειραμάτων.....	23
Εικόνα 10. Μέση χρήση μνήμης .....	23

## **Κατάλογος Πινάκων**

Πίνακας 1. Βασικές δομές ΚΙΟΤ.....	19
Πίνακας 2. Τύποι μηνυμάτων .....	20
Πίνακας 3. Παράμετροι κόμβου .....	21
Πίνακας 4. Λειτουργίες στα δεδομένα προς απάντηση .....	21

## Περίληψη

Ο όρος Διαδίκτυο των Πραγμάτων (Internet of Things, IoT) σημαίνει τη σύνδεση οποιασδήποτε συσκευής στο Internet, αποτελώντας ουσιαστικά ένα όραμα για τη διασύνδεση όλων των "αντικειμένων" στον κόσμο. Η έννοια "Things" (αντικείμενα) αναφέρεται σε μία ευρεία ποικιλία συσκευών εντελώς διαφορετικών μεταξύ τους, όπως αυτοκίνητα με ενσωματωμένους αισθητήρες, κάμερες, κλιματιστικά, φώτα, συστήματα ασφαλείας, smartwatches, ακόμα και αυτοκίνητα των οποίων οι περίπλοκοι αισθητήρες εντοπίζουν αντικείμενα στην πορεία τους.

Η Πληροφοριοκεντρική Δικτύωση (Information-Centric Networking, ICN) είναι μια πολλά υποσχόμενη προσέγγιση για το IoT, καθώς η πρόσβαση στο περιεχόμενο στο ICN βασίζεται σε ονόματα δεδομένων και όχι σε διευθύνσεις IP. Η ονομαστική δικτύωση δεδομένων (Named Data Networking, NDN) είναι μια αρχιτεκτονική που μπορεί να ονομάσει μεμονωμένες τιμές δεδομένων, όπως όλες οι τιμές θερμοκρασίας σε ένα κτίριο, καθώς και ομάδες τιμών, χρησιμοποιώντας ένα σχήμα ιεραρχικής ονομασίας. Ωστόσο, δεν μπορεί να υποστηρίξει ταυτόχρονα πολλές ιεραρχίες ονομάτων για τα ίδια δεδομένα με οικονομικό τρόπο, επομένως τα ερωτήματα για τα δεδομένα έχουν τον περιορισμό να ακολουθούν μία συγκεκριμένη σειρά ομαδοποίησης.

Η δρομολόγηση με βάση λέξεις-κλειδιά (keywords) για το IoT (KIOT) είναι ένα πρωτόκολλο που βασίζεται στην προσέγγιση ICN, και συγκεκριμένα στην αρχιτεκτονική NDN, η οποία βασίζεται στην αποστολή μηνύματος Ενδιαφέροντος (Interest) για ονόματα δεδομένων και στη λήψη των αντίστοιχων μηνυμάτων Δεδομένων (Data). Το σκεπτικό του KIOT είναι να χρησιμοποιεί σύνολα λέξεων-κλειδιών (keywords) τόσο για τις τιμές των ετικετών (Τροίας, Οροφος 1, Εργαστήριο 1, θερμοκρασία), όσο και για να εκφράζει αιτήματα για δεδομένα, π.χ. (Οροφος1, θερμοκρασία). Ο στόχος είναι ένα ερώτημα με συγκεκριμένες λέξεις-κλειδιά να εντοπίζει και να επιστρέφει όλες τις τιμές δεδομένων που χαρακτηρίζονται από ένα υπερσύνολο των κλειδιών αυτών.

Σκοπός της παρούσας εργασίας είναι η υλοποίηση της στρατηγικής υπολογισμού Minimum-Transfer (πέραν της Naive), η προσθήκη λειτουργιών ανάκαμψης του συστήματος σε περίπτωση εξόδου ενός κόμβου από το σύστημα και η γενικότερη βελτίωση και αντιμετώπιση προβλημάτων της υπάρχουσας υλοποίησης, για παράδειγμα, στην αναμονή ενός αγνώστου αριθμού απαντήσεων.

## **Abstract**

The Internet of Things (IoT) is the concept of connecting any device with an on and off switch to the Internet, essentially a vision for interconnecting all of the world's "things." The concept "Things" refers to a wide variety of different things, such as cars with built-in sensors, cameras, air conditioners, lights, security systems, smartwatches, and even cars whose complex sensors detect objects in their path.

Information-Centric Networking (ICN) is a promising approach for the Internet of Things, since accessing content in ICN is based on data names, rather than on IP addresses. Named-Data Networking (NDN) is an architecture that can name individual data values, such as all the temperature values in a building, as well as groups of values, using a hierarchical naming scheme. However, it cannot support multiple hierarchies of names in an economic manner, thus queries for data are restricted to follow one specific grouping order.

Keyword-based routing for IoT (KIOT) is a protocol based on the ICN approach, specifically in the NDN architecture, which is based on sending Interests for named data and receiving the corresponding Data. The rationale of KIOT is to use unordered sets of keywords, both for tagging values (Troias, Floor 1, Lab 1, temperature), and to express requests for data, for example, (Floor 1, temperature). The goal is for a keyword query to find and return all data values that are characterized by a superset of those keywords.

This thesis focuses on the implementation of the Minimum-Transfer strategy (beyond Naive), the addition of system recovery functions in case of a node exit from the system, and the general improvement and troubleshooting of the existing implementation, for example, managing an unknown number of responses for a simple keyword query.

## Εισαγωγή

Το Internet of Things (IoT) είναι μια έννοια που αφορά τα καθημερινά αντικείμενα, από βιομηχανικά μηχανήματα έως φορητές συσκευές που χρησιμοποιούν ενσωματωμένους αισθητήρες για τη συλλογή δεδομένων και τον έλεγχο τους εντός του δικτύου. Το IoT γίνεται πραγματικότητα και μπορεί να εφαρμοστεί σε πολλά σενάρια, όπως έξυπνα σπίτια, εφαρμογές υγειονομικής περίθαλψης, Συστήματα Διαχείρισης Κτιρίων και άλλα. Με βάση το Cloud έχουν προταθεί λύσεις για εφαρμογές πλούσιες σε περιεχόμενο, που επιχειρούν να εξυπηρετήσουν τις συνεχώς αυξανόμενες ανάγκες των χρηστών για ενημέρωση. Ωστόσο, αυτές οι λύσεις που βασίζονται στο Cloud οδηγούν σε υψηλές τιμές Round-Trip Time (RTT) και εξαρτώνται από αβέβαιη συνδεσιμότητα δικτύου.

Το Information Centric Networking (ICN) έχει προταθεί ως μια πολλά υποσχόμενη λύση για IoT σενάρια. Η κύρια ιδέα του ICN είναι να ονομάζει δεδομένα και όχι κεντρικούς υπολογιστές. Με αυτόν τον τρόπο οι χρήστες έχουν άμεσα πρόσβαση σε πληροφορίες, σε επίπεδο δικτύου, χωρίς να χρειάζεται να γνωρίζουν τον κεντρικό υπολογιστή στον οποίο πρέπει να συνδεθούν προκειμένου να λάβουν αυτές τις πληροφορίες.

Αυτή η αρχιτεκτονική επιδιώκει να δημιουργήσει δίκτυα ικανά να διανέμουν περιεχόμενο στο τελικούς χρήστες εστιάζοντας στις ίδιες τις πληροφορίες και όχι στα τελικά σημεία. Αυτό στοχεύει στη μείωση κυκλοφορίας, αποθηκεύοντας τις πληροφορίες πιο κοντά στους χρήστες. Η κύρια ιδέα πίσω από τέτοια δίκτυα είναι πως όταν ζητείται μια συγκεκριμένη πληροφορία, ο χρήστης δεν θα ονομάσει το τελικό σημείο στο οποίο αναμένεται να βρίσκονται αυτές οι πληροφορίες, αλλά θα ζητείται το περιεχόμενο απευθείας προσδιορίζοντας μια περιγραφή του περιεχομένου αυτού. Στη συνέχεια, το δίκτυο θα φροντίσει για τη δρομολόγηση του αιτήματος με τέτοιο τρόπο ώστε ο χρήστης να λάβει ένα αντίγραφο του περιεχομένου.

Μια προσέγγιση του ICN είναι το Named Data Networking (NDN), μια αρχιτεκτονική που βασίζεται σε ιεραρχικά ονόματα, όπου κάθε πληροφορία χωρίζεται σε κομμάτια που αποτελούν τη βασική διευθυνσιοδοτούμενη μονάδα. Ένα παράδειγμα είναι "ΟΠΑ/Τροίας/Οροφος1/Εργαστήριο 1/θερμοκρασία", που αναφέρεται στη θερμοκρασία του εργαστηρίου 1 στον πρώτο όροφο στο κτήριο του ΟΠΑ στην Τροίας. Ακριβώς όπως τα προθέματα IP, τα ονόματα των NDN μπορούν να συγκεντρωθούν σε πίνακες προώθησης με σκοπό τη δρομολόγηση αιτημάτων χρηστών. Στα συστήματα NDN, ένας χρήστης που θέλει να πάρει κάποιες πληροφορίες πρέπει να το ζητήσει ρητά.

Στο NDN υπάρχουν δύο είδη πακέτων:

1. **Interest packets:** Αυτά τα πακέτα φέρουν αίτημα για συγκεκριμένο περιεχόμενο (π.χ., για την ανάκτηση της θερμοκρασίας στέλνεται ένα πακέτο ενδιαφέροντος για το όνομα "ΟΠΑ/Τροίας/Οροφος 1/Εργαστήριο 1/θερμοκρασία"). Στην ουσία, ο χρήστης κάνει ένα ερώτημα σχετικά με τις πληροφορίες που θέλει.
2. **Data packets:** Αφού σταλεί το interest packet, και αφού βρεθεί το περιεχόμενο της πληροφορίας που ζητήθηκε, το data packet αποστέλλεται σαν απάντηση πίσω στον χρήστη.

Η παρούσα εργασία ασχολείται με το KIOT, μια προσέγγιση στη δρομολόγηση βασισμένη σε λέξεις-κλειδιά, η οποία λειτουργεί πάνω από το NDN. Το KIOT έχει εμπνευστεί από το TagNet, μια αρχιτεκτονική ICN που χρησιμοποιεί ετικέτες για να ονομάσει αντικείμενα περιεχομένου σε παγκόσμιο επίπεδο.

Σε σχέση με το TagNet, το KIOT χρησιμοποιεί ένα πιο ευέλικτο σύστημα ονομασίας, το οποίο είναι πιο κατάλληλο για ένα δίκτυο ICN που παράγει προσωρινές πληροφορίες σε μεγάλη κλίμακα. Ο στόχος του συστήματος είναι η επεξεργασία ενός συνόλου δεδομένων IoT τοπικά, πριν από την αποστολή ενός αποτελέσματος στο Internet, χρησιμοποιώντας ένα ευέλικτο και κλιμακωτό σχήμα ονομασίας με βάση λέξεις-κλειδιά.

Η δρομολόγηση πραγματοποιείται σε ένα μόνο δέντρο, με τη ρίζα του να είναι ο δρομολογητής συνόρων, ο οποίος συνδέει ένα απλό δίκτυο NDN με το δίκτυο που χρησιμοποιεί το KIOT. Επιπλέον, γίνεται η υπόθεση πως όλα τα αιτήματα περιεχομένου προέρχονται από τη ρίζα και επομένως όλες οι τελικές απαντήσεις πρέπει να επιστραφούν στη ρίζα. Αυτό σημαίνει ότι κάθε κόμβος χρειάζεται να γνωρίζει μόνο ποια από τις συνδέσεις του οδηγεί στη ρίζα για να επιστρέψει εκεί τα αποτελέσματα. Για την εύρεση του περιεχομένου, κάθε στοιχείο περιεχομένου φέρει ετικέτα με ένα Bloom-filter που υποδεικνύει τις λέξεις-κλειδιά στις οποίες συσχετίζεται και κάθε κόμβος στο δέντρο διατηρεί μια δήλωση για κάθε ένα κόμβο-παιδί, δηλαδή μια λίστα Bloom-filters, αναφέροντας ποια στοιχεία περιεχομένου είναι διαθέσιμα μέσω αυτού του συνδέσμου. Σε αντίθεση με το TagNet, ένα αίτημα ενδιαφέροντος ταιριάζει με ένα στοιχείο περιεχομένου αν το Bloom-filter του αιτήματος είναι ένα υποσύνολο του Bloom-filter του στοιχείου (για παράδειγμα το {εργαστήριο, θερμοκρασία} είναι υποσύνολο του {Τροιάς, Όροφος 0, εργαστήριο, θερμοκρασία, υγρασία}).

Στο KIOT, κάθε κόμβος στο δίκτυο έχει αποθηκευμένες 3 δομές δεδομένων:

1. **Forwarding Information Base (FIB):** Χρησιμοποιείται για την προώθηση πακέτων ενδιαφέροντος προς μια εκ των πηγών που διαθέτουν το περιεχόμενο που ζητά ο χρήστης. Πιο αναλυτικά, το FIB συσχετίζει κάθε πρόθεμα ονόματος/ετικέτας σε ένα σύνολο προορισμών. Ο κόμβος χρησιμοποιεί το FIB για να προωθήσει ένα πακέτο ενδιαφέροντος σε μια διεπαφή άλλου κόμβου εάν ταιριάζει με ένα συγκεκριμένο πρόθεμα. Ακριβώς όπως ένας τυπικός δρομολογητής IP διευθύνσεων, η προώθηση λειτουργεί χρησιμοποιώντας αντιστοίχιση μακρύτερου προθέματος μεταξύ του ονόματος στο πακέτο ενδιαφέροντος και τα προθέματα που είναι αποθηκευμένα στο FIB.
2. **Pending Interest Data (PID):** Αποθηκεύει προσωρινά την κατάσταση των αιτημάτων ενδιαφέροντος που προωθήθηκαν κάποια στιγμή στο παρελθόν, αλλά δεν έχουν σταλεί σαν απάντηση ακόμα πίσω στο χρήστη, μαζί με τις απαντήσεις από τους κόμβους στους οποίους προωθήθηκε το αίτημα. Καθώς, το πακέτο μετακινείται μεταξύ των κόμβων, μια νέα εγγραφή δημιουργείται στο PID του κόμβου. Οι απαντήσεις που είναι αποθηκευμένες στο PID θα σταλούν αργότερα σαν απάντηση στο χρήστη. Για να συμβεί αυτό, το PID πρέπει να αποθηκεύσει το αναγνωριστικό του πακέτου ενδιαφέροντος και την διεπαφή από την οποία στάλθηκε το πακέτο ενδιαφέροντος.
3. **Bloom Filter (BF):** Αποτελεί μια δομή δεδομένων που χρησιμοποιείται για τον αποδοτικό έλεγχο του αν ένα στοιχείο ανήκει σε ένα σύνολο. Βασίζεται σε ένα πίνακα με bits (είναι αρχικοποιημένος με κάθε bit να έχει την τιμή 0) και πολλαπλές συναρτήσεις κατακερματισμού (hash functions). Όταν προστίθεται ένα στοιχείο, οι τιμές από τις hash functions ορίζουν συγκεκριμένες θέσεις στον πίνακα με τα bits όπου θα αλλάξουν σε 1. Για τον έλεγχο αν ένα στοιχείο ανήκει στο σύνολο, ελέγχονται αν οι θέσεις αυτές είναι ισούνται με 1. Αν οποιαδήποτε από αυτές δεν ισούται με 1, το στοιχείο δεν ανήκει στο σύνολο.



# Περιγραφή ΚΙΟΤ

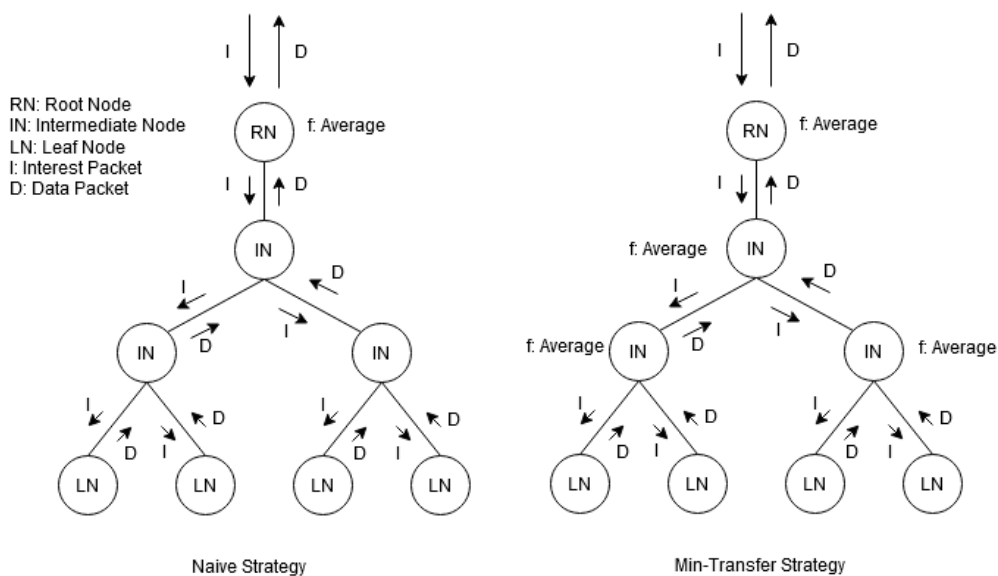
## Στρατηγικές Λειτουργίας

### Naive

Στα συστήματα που βασίζονται στο Cloud, όλα τα δεδομένα που απαιτούνται για την εκτέλεση μιας λειτουργίας, όπως η Μέση Τιμή μιας ομάδας μετρήσεων, πρέπει να ανακτηθούν ατομικά από τους κόμβους IoT σε έναν διακομιστή cloud, ώστε να μπορούν να επεξεργαστούν. Στην αρχιτεκτονική του ΚΙΟΤ, ο προφανής τόπος για την εκτέλεση μιας τέτοιας λειτουργίας θα ήταν ο δρομολογητής συνόρων (border router), δεδομένου ότι όλα τα δεδομένα συγκλίνουν εκεί σύμφωνα με το σχήμα δρομολόγησής μας. Επιπλέον, αυτό επιτρέπει την επιστροφή ενός μόνο κομματιού δεδομένων στο μέρος του δικτύου που είναι συμβατό με NDN, αντί για ένα σύνολο τιμών. Επομένως, η πρώτη στρατηγική που ονομάζεται **Naive** είναι η πιο απλή: επεξεργασία όλων των δεδομένων στον ριζικό κόμβο (root node) του τομέα IoT.

### Minimum Transfer

Σε ένα απλό παράδειγμα ενός IoT συστήματος, όπως ένας Πανεπιστημιακός χώρος, υπάρχουν παραπάνω από ένα επίπεδα στο δένδρο δρομολόγησης, για παράδειγμα περιοχή, κτήριο, όροφος, αισθητήρας, με πολλούς αισθητήρες να συνδέονται με ένα gateway κόμβο, μετά με ένα controller περιοχής και ούτω καθεξής μέχρι τον ριζικό κόμβο. Επομένως, περισσότερη αξία έχει οι υπολογισμοί να γίνονται πιο κοντά στην πηγή δεδομένων, για την αποφυγή συμφόρησης μεγάλων όγκων δεδομένων στον δρομολογητή συνόρων και την επιβάρυνσή του με την επεξεργασία τους και τον υπολογισμό της εκάστοτε λειτουργίας. Έτσι, η δεύτερη στρατηγική ονομάζεται **Minimum Transfer**: επεξεργασία των δεδομένων στο χαμηλότερο επίπεδο διακλάδωσης των κόμβων, κοντά στους αισθητήρες.



Εικόνα 1. Είδη Στρατηγικών ΚΙΟΤ.

## Είδη Κόμβων

Στο KΙΟΤ η (εικονική) τοπολογία του δικτύου έχει πάντα δενδρική μορφή, επομένως κάθε κόμβος εντάσσεται σε έναν από τους ακόλουθους τρεις τύπους κόμβων:

1. **Leaf Node:** Ο τύπος αυτός αντιπροσωπεύει τους κόμβους που βρίσκονται στο κατώτατο επίπεδο της τοπολογίας. Κάθε περίπτωση ενός κόμβου φύλλων συνδέεται με ένα όνομα και ένα σύνολο λέξεων-κλειδιών που περιγράφουν τις πληροφορίες που προέρχονται από τους αισθητήρες. Κάθε αισθητήρας περιγράφεται σε ένα Bloom Filter που ονομάζεται BF-Keyword. Η ένωση αυτών των λέξεων-κλειδιών BF οδηγεί στον κατάλογο BF, ο οποίος περιγράφει τα περιεχόμενα ενός κόμβου φύλλων. Για παράδειγμα, ένας κόμβος φύλλων μπορεί να έχει πληροφορίες σχετικά με τη θερμοκρασία (λέξη-κλειδί BF: 01001000) και την υγρασία (BF λέξη-κλειδί: 01000001), οπότε ο κατάλογός του BF θα είναι η ένωση αυτών των δύο με τιμή 01001001.
2. **Intermediate Node:** Ο τύπος αυτός αντιπροσωπεύει τους κόμβους του δένδρου που βρίσκονται ανάμεσα στη ρίζα και τα παιδιά (μπορούν να υπάρχουν πολλά επίπεδα). Αυτοί οι κόμβοι κατέχουν έναν πίνακα δρομολόγησης με δύο στήλες. Η πρώτη στήλη αποθηκεύει τους εισερχόμενους καταλόγους BF από τους θυγατρικούς κόμβους, ενώ η δεξιά αποθηκεύει τη διεύθυνση IP από τον κόμβο στον οποίο ανήκει ο κατάλογος BF (θα μπορούσε να είναι και διεύθυνση Ethernet).
3. **Root Node:** Ο τύπος κόμβου είναι ο ίδιος με έναν μεσαίο κόμβο, χωρίς το ανώτερο τελικό σημείο, αφού οι κόμβοι αυτοί είναι το ανώτερο επίπεδο του δικτύου. Τα ερωτήματα υποβάλλονται σε αυτό τον κόμβο, και εκεί επιστρέφουν οι απαντήσεις.

Το σύστημα είναι αυτο-οργανωμένο, απαιτώντας οι κόμβοι να γνωρίζουν μόνο το επίπεδό τους στο δέντρο κατά την αρχικοποίηση, με τις συνδέσεις των δέντρων να επιλέγονται δυναμικά. Οι κόμβοι φύλλων φυσικά πρέπει επίσης να γνωρίζουν τις λέξεις-κλειδιά BF για τους συνδεδεμένους αισθητήρες.

## Τύποι Πακέτων

Το KΙΟΤ χρησιμοποιεί τέσσερις τύπους πακέτων.

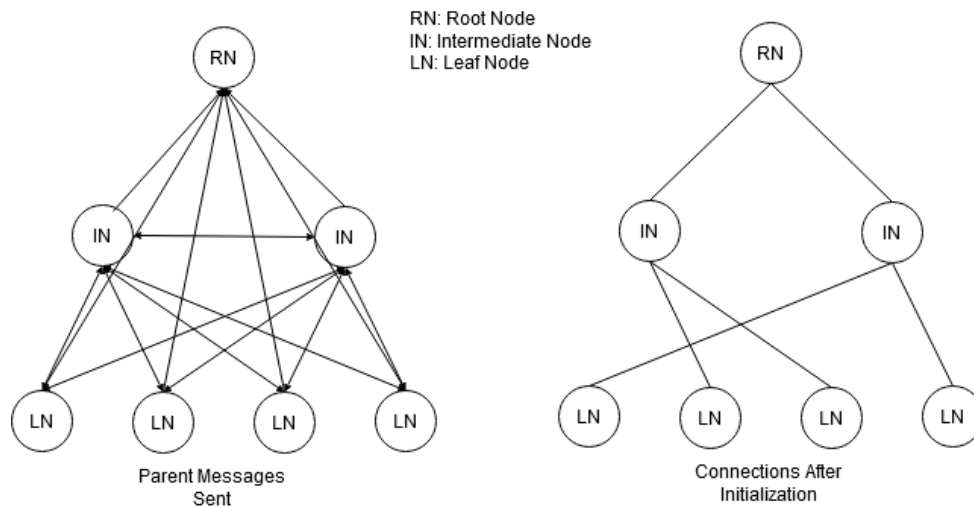
- **Parent Packet:** Κατά την αρχικοποίηση του συστήματος, κάθε κόμβος πρέπει να βρει τον γονέα του για να μπορεί να επικοινωνεί μαζί του και να στέλνει τα κατάλληλα δεδομένα. Το μόνο που πρέπει να γνωρίζει είναι το επίπεδο στο οποίο βρίσκεται μέσα στο δένδρο. Επομένως, στην αρχή κάθε κόμβος στέλνει ένα μήνυμα εκπομπής στο δίκτυο (σε ένα ασύρματο δίκτυο, αυτοί θα είναι οι κόμβοι εντός του εύρους εκπομπής) και όλοι οι κόμβοι που λαμβάνουν αυτό το μήνυμα μπορούν να αποφασίσουν εάν είναι δυνητικοί γονείς του ζητούμενου κόμβου. Οι δυνητικοί γονείς είναι εκείνοι οι κόμβοι που βρίσκονται στο επόμενο (ανώτερο) επίπεδο του αιτούντος. Η επιλογή του κόμβου πατέρα ανάμεσα στους δυνητικούς κόμβους που έχουν απαντήσει στο Parent Message κρίνεται με μια εκ των ακόλουθων πολιτικών:
  1. **Load Balancing:** Αυτή η πολιτική βασίζεται στον αριθμό των παιδιών των δυνητικών γονέων. Όταν ένας κόμβος στέλνει το μήνυμα εκπομπής, οι δυνητικοί γονείς θα απαντήσουν. Αυτή η απάντηση θα περιέχει τον αριθμό των παιδιών τους. Έτσι, ο κόμβος με τα λιγότερα παιδιά θα επιλεγεί ως γονέας του αιτούντος κόμβου. Εάν όλοι οι δυνητικοί γονείς έχουν τον ίδιο αριθμό παιδιών, τότε θα επιλεγεί ένας τυχαία.
  2. **Delay Minimization:** Αυτή η πολιτική βασίζεται στο Round Trip Time (RTT) μεταξύ κόμβων. Σε αυτήν την περίπτωση, όταν ο αιτών κόμβος λαμβάνει την απάντηση από δυνητικούς γονείς, λαμβάνει επίσης δεδομένα σχετικά με το RTT. Ο κόμβος με χαμηλότερη τιμή RTT θα είναι ο γονέας του ζητούμενου κόμβου.

- **Advertisement Packet:** Αυτή η ενέργεια λαμβάνει χώρα μόλις ένας κόμβος-φύλλο αρχίσει να λειτουργεί. Κάθε κόμβος (από κάτω προς τα πάνω) στέλνει λέξεις-κλειδιά που διαφημίζουν τις πληροφορίες που προσφέρει, αποθηκευμένες σε Bloom Filters, στον μητρικό του κόμβο. Κάθε μητρικός κόμβος διατηρεί έναν πίνακα προώθησης (FIB) που αποθηκεύει αρχεία των πληροφοριών (BF καταλόγους) που προέρχονται από τους θυγατρικούς κόμβους και τις διευθύνσεις των κόμβων αυτών. Ο κόμβος που λαμβάνει το Advertisement Packet θα πραγματοποιήσει μια ένωση μεταξύ του ληφθέντος Bloom Filter και του τοπικού Bloom Filter και θα το διαφημίσει στον δικό του μητρικό κόμβο, αν υπάρχει. Ο μητρικός κόμβος θα κάνει το ίδιο με τη σειρά του.
- **Interest Packet:** Κάθε φορά που εμφανίζεται ένα πακέτο Interest στη ρίζα, η διαδικασία είναι η ακόλουθη. Το αίτημα ενδιαφέροντος (π.χ. Τροίας, υγρασία) αποστέλλεται στον κόμβο ρίζας. Ο ριζικός κόμβος ελέγχει αν το περιεχόμενο που διαθέτει περιλαμβάνει το περιεχόμενο του ερωτήματος ή όχι, ελέγχοντας τον κατάλογο BF. Στη συνέχεια, ελέγχει εάν το αίτημα ερώτησης είναι ένα υποσύνολο οποιουδήποτε από τους καταλόγους BF του FIB της ρίζας. Εάν αυτό είναι αληθινό, τότε το αίτημα ερωτήματος προωθείται στους θυγατρικούς κόμβους (ένα ή περισσότερα) των αντίστοιχων αποτελεσμάτων. Δεδομένου ότι οι διευθύνσεις IP αποθηκεύονται, ο γονικός κόμβος ξέρει πού να στείλει το αίτημα ενδιαφέροντος, σύμφωνα με την αντιστοιχία του καταλόγου BF. Αυτό συμβαίνει μέχρι να φτάσει το αίτημα στους κόμβους των φύλλων όπου αποθηκεύονται οι πραγματικές πληροφορίες.
- **Data Packet:** Ο τύπος του πακέτου αντιστοιχεί στην απάντηση του ερωτήματος που εισήλθε στο σύστημα. Ανάλογα με τη στρατηγική λειτουργίας του συστήματος, το περιεχόμενο του πακέτου θα σταλεί προς την κορυφή του δένδρου είτε επεξεργασμένο από τους προηγούμενους κόμβους (minimum-transfer στρατηγική) είτε για επεξεργασία από τον κόμβο (naive στρατηγική) και, στη συνέχεια, θα σταλεί ως απάντηση στον client που έθεσε το ερώτημα (Εικόνα 1).

## Λειτουργία

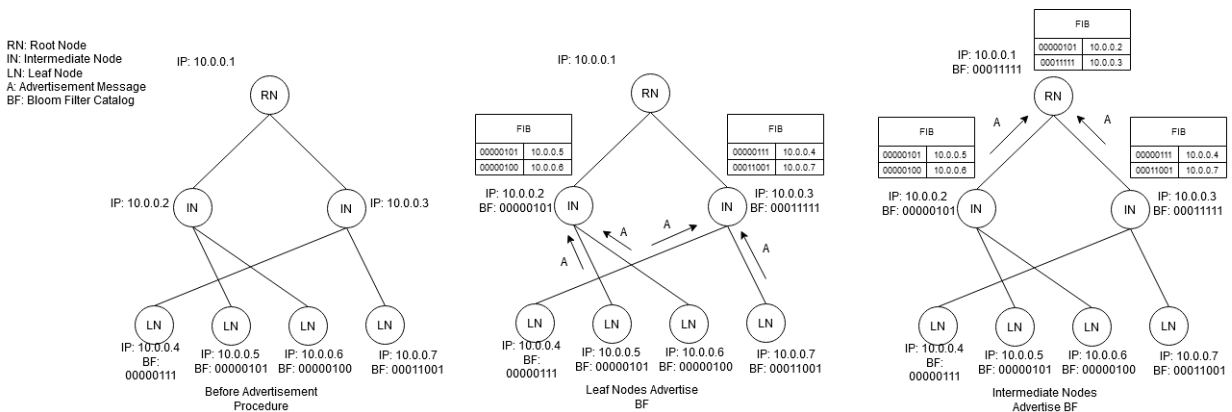
Παρακάτω περιγράφεται ο κύκλος ζωής ενός κόμβου μέσα σε μία KIoT τοπολογία, από την αρχικοποίηση του μέχρι τη λήψη ενός αιτήματος και την αποστολή της απάντησης του.

1. Στη φάση αρχικοποίησης του συστήματος, κάθε κόμβος στέλνει ένα μήνυμα εκπομπής MSG\_PARENT για να αφήσει τους άλλους κόμβους να γνωρίζουν ότι βρίσκεται στην αναζήτηση γονικού κόμβου. Οι κόμβοι στο ανώτερο επίπεδο θα απαντήσουν στον αιτούντα με το MSG\_PARENT\_ACK. Ανάλογα με την πολιτική που εφαρμόζεται στο σύστημα, ο κόμβος θα επιλέξει είτε τον κόμβο με τα λιγότερα παιδιά είτε με τον κόμβο με το μικρότερο RTT, στέλνοντας ένα MSG\_PARENT\_CHILD στον επιλεγμένο γονικό κόμβο του, έτσι ώστε να μπορεί να αυξήσει τα παιδιά του.



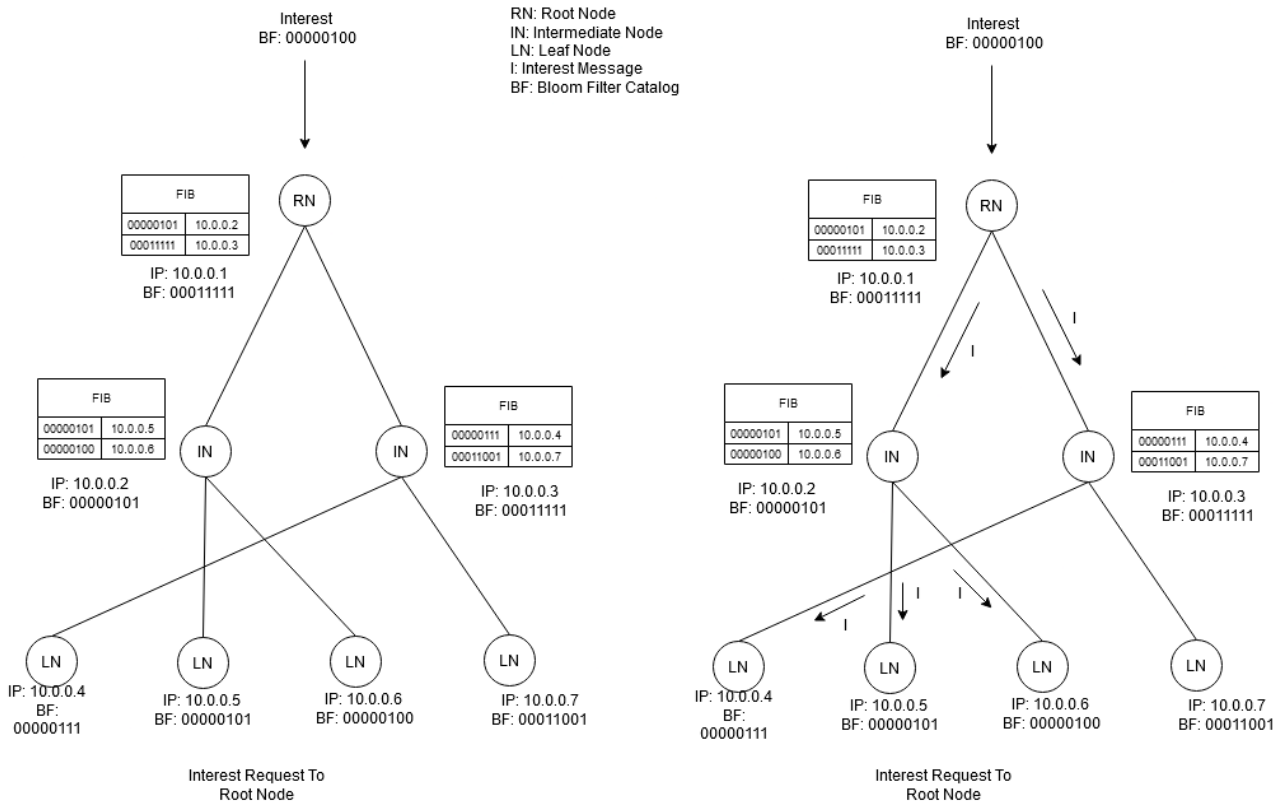
Εικόνα 2. Αποστολή μηνυμάτων MSG\_PARENT

2. Το επόμενο βήμα είναι η διαφήμιση του BF καταλόγου από τους κόμβους του κατώτερου επιπέδου για την πληροφορία του αισθητήρα που υποστηρίζουν (θερμοκρασία, υγρασία, πίεση κ.λπ.). Όταν όλοι οι κόμβοι βρουν τους γονείς τους, περνάνε σε αυτή τη φάση της διαφήμισης. Οι κόμβοι φύλλων στέλνουν MSG\_ADVERTISEMENT στους γονείς τους με τις πληροφορίες τους. Τώρα, αυτοί οι κόμβοι στέλνουν τις συγκεντρωτικές πληροφορίες στους γονείς τους και αυτή η διαδικασία επαναλαμβάνεται μέχρι να φτάσουμε στον κόμβο ρίζας. Κάθε μήνυμα MSG\_ADVERTISEMENT επιβεβαιώνεται από ένα MSG\_ADVERTISEMENT\_ACK, για να γνωρίζει ο κόμβος αν το μήνυμα λήφθηκε από τον γονικό κόμβο με επιτυχία.



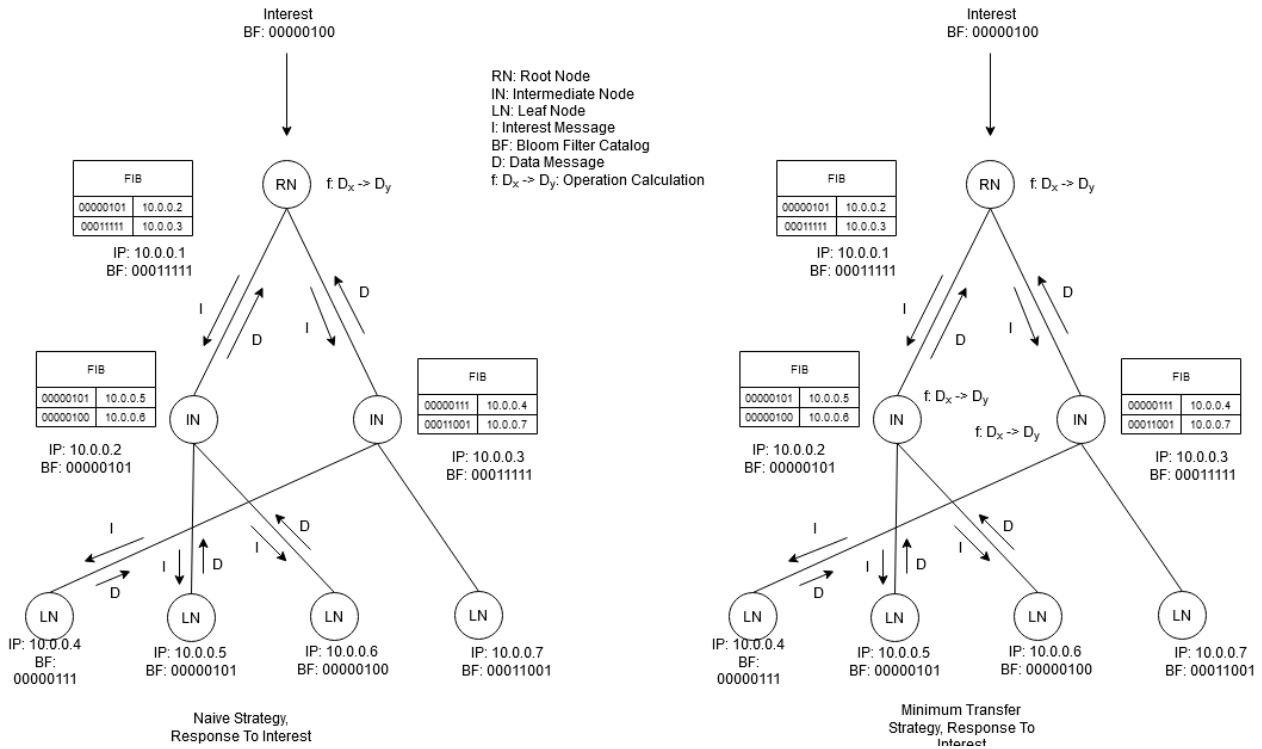
Εικόνα 3. Αποστολή μηνυμάτων MSG\_ADVERTISEMENT

3. Σε αυτό το σημείο το σύστημα είναι έτοιμο. Όλοι οι κόμβοι έχουν δημιουργήσει τους πίνακες FIB, γνωρίζουν τους γονικούς κόμβους τους και είναι έτοιμοι να για εισερχόμενα μηνύματα. Ένας client μπορεί να ζητήσει πληροφορίες τώρα στέλνοντας ένα MSG\_INTEREST με το είδος των πληροφοριών (π.χ. ΟΠΑ, Ευελπίδων, υγρασία) που θέλει στον κόμβο ρίζα. Ο κόμβος ελέγχει το FIB για να στείλει με τη σειρά του MSG\_INTEREST μήνυμα στο κατάλληλο κόμβο (ή κόμβους) του κατώτερου επιπέδου. Οι κόμβοι αυτοί με τη σειρά τους ελέγχουν το FIB τους για να κατευθύνουν το πακέτο στο κατώτερο επίπεδο μέχρι το μήνυμα να φτάσει στους κόμβους του κατώτατου επιπέδου (κόμβοι φύλλων), από όπου και θα ξεκινήσει και η διαδικασία απάντησης στο εισερχόμενο αίτημα του client.



Εικόνα 4. Εισαγωγή αιτήματος στο σύστημα

4. Όταν το αίτημα φτάσει τους κόμβους στο κατώτατο επίπεδο (κόμβοι φύλλων), ένα μήνυμα MSG\_DATA με το περιεχόμενο δρομολογείται από το κάτω προς τα πάνω επίπεδα (κόμβος σε γονέα) μέχρι να φτάσει το περιεχόμενο στη ρίζα. Στην περίπτωση που το σύστημα εφαρμόζει Naïve στρατηγική, το περιεχόμενο όλων των MSG\_DATA μηνυμάτων θα φτάσει χωρίς επεξεργασία στον κόμβο ρίζας, όπου και θα γίνει η επεξεργασία πριν την αποστολή της απάντησης προς τον client. Στην περίπτωση που το σύστημα εφαρμόζει Minimum Transfer στρατηγική, το περιεχόμενο των MSG\_DATA μηνυμάτων θα υποστεί επεξεργασία από τους κόμβους των ενδιάμεσων επιπέδων μεταξύ του κατώτατου και του ανώτερου επιπέδου πριν φτάσει στον ριζικό κόμβο όπου και θα εφαρμοστεί η τελική επεξεργασία (αν κρίνεται απαραίτητη) πριν σταλεί σαν απάντηση προς τον client μειώνοντας τη έτσι τη συμφόρηση μηνυμάτων στον ριζικό κόμβο και την ανάγκη για επαρκή υπολογιστική ισχύ.

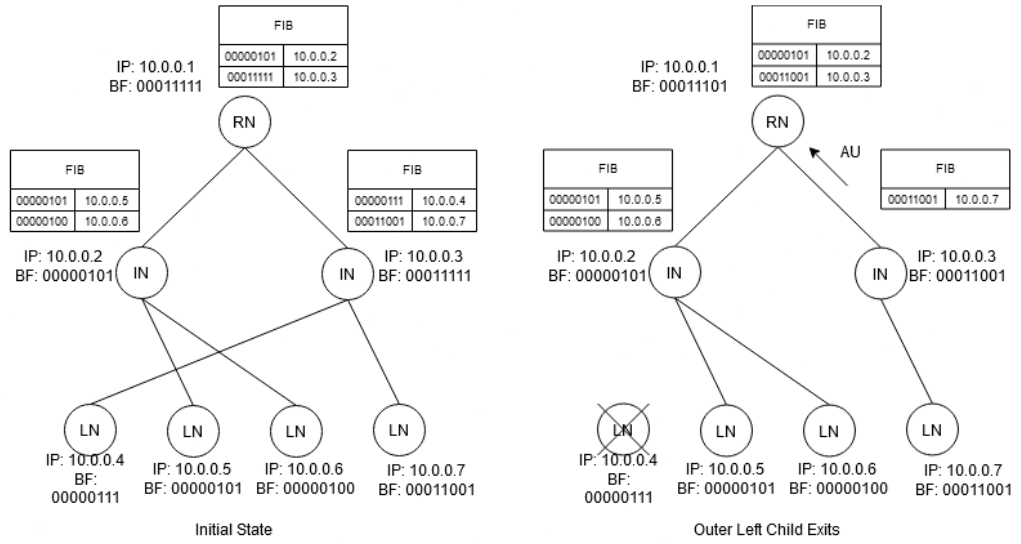


Εικόνα 5. Απάντηση προς Client

5. Στην περίπτωση που ένας κόμβος εξέλθει από το σύστημα, οι υπόλοιποι κόμβοι θα πρέπει να ενημερώσουν την κατάσταση των BF καταλόγων τους αλλά και τους FIB πίνακες τους με σκοπό τη σωστή αντιμετώπιση σε μελλοντικά εισερχόμενα μηνύματα. Κάθε κόμβος είναι υλοποιημένος να αντιμετωπίσει τις εξής περιπτώσεις:

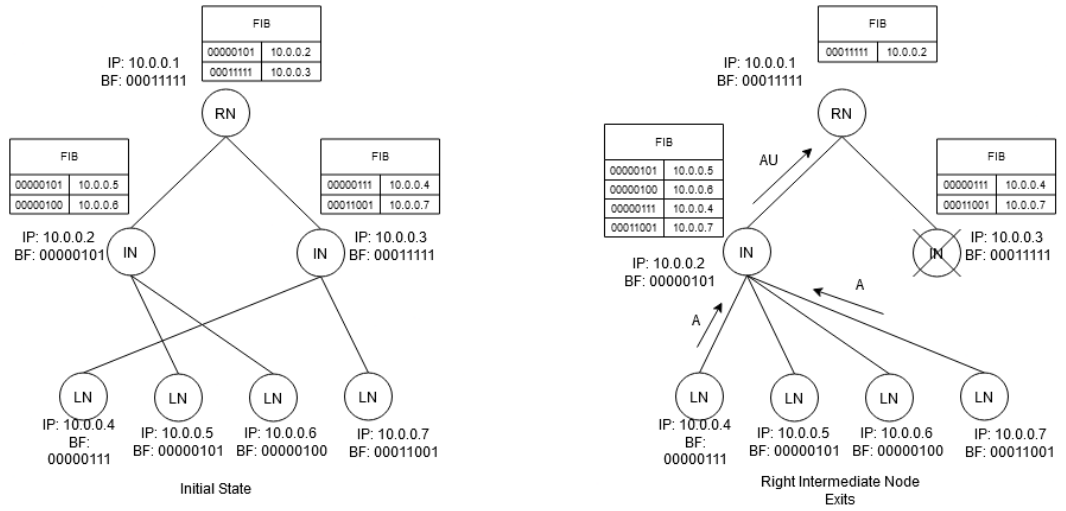
- **Έξοδος κόμβου παιδιού.** Κάθε κόμβος ελέγχει ανά τακτά χρονικά διαστήματα αν όλα τα παιδιά του είναι ενεργά. Στην περίπτωση που ένας κόμβος-παιδί έχει εξέλθει από το σύστημα τότε ενημερώνει το BF κατάλογο του και τα περιεχόμενα του τοπικού FIB πίνακα και αποστέλλει ένα MSG\_ADVERTISEMENT\_UPDATE μήνυμα στον γονικό κόμβο του με σκοπό να ενημερώσει και εκείνος με τη σειρά του τα τοπικά δεδομένα. Ο γονικός κόμβος απαντάει με MSG\_ADVERTISEMENT\_ACK μήνυμα για να ενημερώσει για επιτυχή λήψη του μηνύματος. Η διαδικασία για ενημέρωση των τοπικών BF καταλόγων και των περιεχομένων του FIB πίνακα γίνεται σε όλα τα επίπεδα της τοπολογίας μέχρι τον κόμβο ρίζα.
- **Έξοδος γονικού κόμβου.** Κάθε κόμβος ελέγχει ανά τακτά χρονικά διαστήματα αν ο γονικός κόμβος είναι ενεργός. Στην περίπτωση που ο κόμβος έχει εξέλθει από το σύστημα τότε ο κόμβος παιδί ξεκινάει από τη αρχή την διαδικασία εύρεσης γονικού κόμβου (Βήμα 1).

RN: Root Node  
 IN: Intermediate Node  
 LN: Leaf Node  
 A: Advertisement Message  
 AU: Advertisement Update Message  
 BF: Bloom Filter Catalog



Εικόνα 6. Έξοδος κόμβου παιδιού

RN: Root Node  
 IN: Intermediate Node  
 LN: Leaf Node  
 A: Advertisement Message  
 AU: Advertisement Update Message  
 BF: Bloom Filter Catalog



Εικόνα 7. Έξοδος γονικού κόμβου

## Υλοποίηση και Βελτιώσεις

Στην ενότητα αυτή περιγράφεται η τρέχουσα υλοποίηση του KΙOT, με έμφαση στις βελτιώσεις που έγιναν στα πλαίσια της παρούσας εργασίας. Η υλοποίηση αποτελείται από δύο ενότητες κώδικα, την KΙOT και την Guest. Η KΙOT είναι η υλοποίηση ενός κόμβου του δικτύου (ρίζας, φύλλου, ενδιάμεσου) ενώ η Guest είναι η υλοποίηση ενός κόμβου πελάτη που συνδέεται με τον κόμβο ρίζας για υποβολή ερωτημάτων. Η υλοποίηση τόσο του KΙOT όσο και του Guest node, έγινε με τη χρήση της γλώσσας προγραμματισμού Java. Για τη δημιουργία εκτελέσιμων αρχείων που μπορούν να εκτελούνται χωρίς το JVM σε λειτουργικά συστήματα βασισμένα σε Linux, χρησιμοποιήθηκε το GraalVM, το οποίο μετατρέπει τον bytecode της Java σε binary εκτελέσιμο. Στη συνέχεια, περιγράφονται οι αλλαγές-βελτιώσεις που υλοποιήθηκαν στο KΙOT συγκριτικά με προηγούμενες εκδόσεις του.

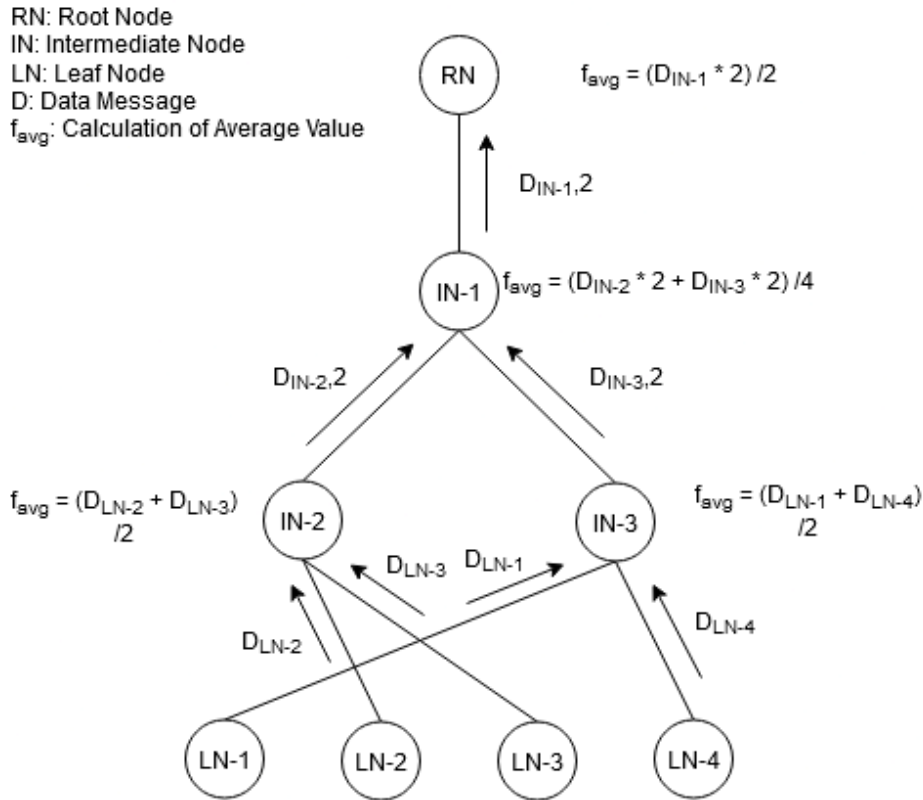
### Guest Node

Ο ρόλος του Guest κόμβου είναι να δημιουργεί Interest πακέτα προς το ριζικό κόμβο της τοπολογίας του KΙOT. Ο κόμβος διαβάζει τις λέξεις κλειδιά που απαρτίζουν το αίτημα μαζί με την λειτουργία (πίνακας 4) που πρέπει να εφαρμοστεί στα δεδομένα που περιμένει σαν απάντηση, από ένα αρχείο CSV που έχει οριστεί στο configuration του. Για τη δημιουργία του Interest πακέτου, οι λέξεις-κλειδιά που διαβάζονται από το αρχείο CSV μετατρέπονται σε δομή Bloom-Filter. Ο Guest κόμβος αποτελεί μια απλουστευμένη μορφή του KΙOT κόμβου, χρησιμοποιώντας πλήθος από δομές-components όπως αυτές έχουν υλοποιηθεί στο KΙOT.

### Στρατηγική Minimum-Transfer

Σε προηγούμενες εκδόσεις του KΙOT ήταν υλοποιημένη μόνο η Naive στρατηγική. Με την υλοποίηση της Minimum στρατηγικής Transfer, οι ενδιάμεσοι κόμβοι επιβαρύνονται με την επεξεργασία των απαντήσεων όπως που προέρχονται από τους κόμβους παιδιά, ενώ παράλληλα μειώνεται ο αριθμός των υπολογισμών που χρειάζονται να γίνουν στον ριζικό κόμβο καθώς και το μέγεθος και πλήθος δεδομένων που στέλνονται προς αυτόν. Ιδιαίτερη προσοχή χρειάστηκε να δοθεί στον υπολογισμό της μέσης τιμής των αποτελεσμάτων που προέρχονται από τους κόμβους-παιδιά για την στάθμιση των αποτελεσμάτων που αποστέλλονται προς τον χρήστη. Πιο συγκεκριμένα, μαζί με την μέση τιμή κάθε ενδιάμεσος κόμβος έστελνε και τον αριθμό των κόμβων-παιδιών από τα οποία έλαβε απάντηση, με σκοπό να συμπεριληφθεί αυτός ο αριθμός στον υπολογισμό της μέσης τιμής από τον γονικό του κόμβο. Η αποστολή του αριθμού των κόμβων-παιδιών, από τα οποία ένας κόμβος έλαβε απάντηση, εφαρμόζεται μόνο την περίπτωση που πρέπει να υπολογιστεί η μέση τιμή των αποτελεσμάτων. Για τις υπόλοιπες λειτουργίες αποστέλλονται μόνο τα αποτελέσματα από τους κόμβους παιδιά προς τους γονείς τους. Η συγκεκριμένη υλοποίηση επεξηγείται καλύτερα στην ακόλουθη εικόνα.





Εικόνα 8. Minimum Transfer: υπολογισμός Average τιμής

## Αναμονή αποτελεσμάτων

Σε προηγούμενες εκδόσεις του KIOT, ο Guest μαζί με το MSG\_INTEREST έστειλε ένα αριθμό ως ελάχιστο αριθμό απαντήσεων που χρειαζόταν να λάβει ο ριζικός κόμβος από το υπόλοιπο σύστημα για στείλει την απάντηση πίσω στον client. Αυτό οφείλεται στο ότι δεν γνωρίζουμε από την αρχή πόσες απαντήσεις θα παραχθούν σε απάντηση ενός αιτήματος. Ένα χρονικό περιθώριο είχε οριστεί στον ριζικό κόμβο για να στείλει την απάντηση πίσω στον Guest σε περίπτωση που δεν είχε λάβει επαρκή αριθμό απαντήσεων από το υπόλοιπο σύστημα, έχοντας υπόψιν πώς οι απαντήσεις προωθούνταν από τα κατώτερα επίπεδα προς τη ρίζα χωρίς περαιτέρω καθυστέρηση.

Στη συγκεκριμένη έκδοση του KIOT τόσο ο ριζικός κόμβος όσο και οι ενδιάμεσοι κόμβοι κρατάνε σαν πληροφορία τον αριθμό των κόμβων-παιδιά στους οποίους προώθησαν το μήνυμα καθώς και από πόσους κόμβους έλαβαν MSG\_DATA μήνυμα. Στην περίπτωση της Naïve στρατηγικής αν οι ενδιάμεσοι κόμβοι έχουν λάβει τις απαντήσεις από όλους τους κόμβους-παιδιά στους οποίους προώθησαν το αίτημα τότε αποστέλλουν ένα επιπλέον μήνυμα MSG\_DATA\_ACK προς τον ριζικό κόμβων ενημερώνοντας τον για την λήψη όλων των αναμενόμενων απαντήσεων. Στην περίπτωση της Minimum Transfer στρατηγικής, όπου η επεξεργασία των απαντήσεων γίνεται και στους ενδιάμεσους κόμβους, κάθε κόμβος αποστέλλει την επεξεργασμένη απάντηση του όταν έχει λάβει όλες τις αναμενόμενες απαντήσεις από τους κόμβους παιδιά.

Όταν ο κόμβος δεν έχει λάβει απάντηση από όλους τους κόμβους παιδιά, τότε χρησιμοποιείται ένα δυναμικά ορισμένο χρονικό περιθώριο πριν γίνει η προώθηση της απάντησης είτε στους κόμβους ανώτερου επιπέδου είτε στον client. Έτσι, μειώνεται ο συνολικός χρόνος αποστολής των αποτελεσμάτων από την στιγμή που λήφθηκε το Interest μήνυμα, ενώ παράλληλα μεγιστοποιείται το σύνολο των αποτελεσμάτων που στέλνονται σαν απάντηση προς τον χρήστη. Το χρονικό αυτό περιθώριο υπολογίζεται σε κάθε κόμβο με βάση το επίπεδο στο οποίο ανήκει ο κόμβος και τον συνολικό αριθμό των επιπέδων στην τοπολογία του συστήματος  $Threshold = \frac{(Topology Levels - Node Level + 1) * 5000}{Topology Levels}$ .

## Έξοδος κόμβου από την τοπολογία

Για τον έλεγχο της περίπτωσης που ένας κόμβος έχει εξέλθει από το σύστημα, υλοποιήθηκαν κάποια scheduled tasks, που εκτελούνται ανά τακτά χρονικά διαστήματα, στέλνοντας ICMP echo αιτήματα αν ο κόμβος έχει privileged δικαιώματα, αλλιώς γίνεται προσπάθεια εγκαθίδρυσης μιας TCP σύνδεσης στην πόρτα 7 του κόμβου (μέσω της συνάρτησης InetAddress.isReachable που παρέχεται από την Java). Για την αντιμετώπιση της περίπτωσης που ένας κόμβος έχασε τον γονικό του κόμβο και στο σύστημα δεν υπάρχει άλλος κόμβος στο ίδιο επίπεδο με τον κόμβο πατέρα, οι κόμβοι που βρίσκονται στο ακριβώς επόμενο επίπεδο από αυτό του πατέρα, μπορούν να απαντήσουν σε MSG\_PARENT μηνύματα όταν λάβουν 3 συνεχόμενα MSG\_PARENT μηνύματα εντός 1 λεπτού από τον ίδιο κόμβο. Έτσι, όλοι οι κόμβοι θα είναι προσβάσιμοι χωρίς καμία απώλεια πλην του κόμβου που εξήλθε.

## Υπολογισμός RTT για την εύρεση κόμβου

Για τον υπολογισμό του RTT κατά την επιλογή κόμβου πατέρα από έναν άλλο κόμβο, ο κόμβος στέλνει 3 ICMP echo αιτήματα αν ο κόμβος έχει privileged δικαιώματα, αλλιώς γίνεται προσπάθεια εγκαθίδρυσης 3 TCP συνδέσεων στην πόρτα 7 (μέσω της συνάρτησης InetAddress.isReachable που παρέχεται από την Java) σε κάθε δυνητικό κόμβο πατέρα, από όπου υπολογίζεται η μέση RTT τιμή και επιλέγεται ο κόμβος με την μικρότερη μέση τιμή.

## Ρυθμίσεις Bloom Filters

Το Bloom Filter είναι μια δομή δεδομένων που χρησιμοποιείται για τον γρήγορο έλεγχο της ύπαρξης στοιχείων σε ένα σύνολο με την πιθανότητα ύπαρξης false-positives αποτελεσμάτων, αλλά χωρίς false-negatives. Η αρχικοποίηση του Bloom Filter γίνεται ορίζοντας το μέγεθος του πίνακα με Bits που χρησιμοποιείται από το Bloom Filter καθώς και το ανώτατο όριο ετικετών που μπορεί να αποθηκεύσει ο κατάλογος. Το μέγεθος του BF καταλόγου κάθε κόμβου έχει σταθερό μέγεθος 160 Bits ενώ ο μέγιστος αριθμός από ονόματα/ετικέτες που μπορούν να καταχωρηθούν στον κατάλογο είναι 20. Ο αριθμός των συναρτήσεων κατακερματισμού που θα χρησιμοποιηθούν υπολογίζεται δυναμικά από το μέγεθος του BF καταλόγου και τον μέγιστο αριθμό στοιχείων που μπορούν να αποθηκευτούν με βάση τον τύπο

$num\_of\_hashes = \lceil \frac{BitSet\ Size}{Max\ Entries} \ln(2) \rceil$ , ελαχιστοποιώντας έτσι την πιθανότητα εμφάνισης false-positives αποτελεσμάτων. Τέλος η συνάρτηση κατακερματισμού που χρησιμοποιείται είναι ο αλγόριθμος MD5.

## Logging

Σημαντικό κομμάτι για την ανάπτυξη και στον έλεγχο σωστής λειτουργίας της εφαρμογής αποτελεί η αποτύπωση των επικοινωνιών που πραγματοποιεί ο κάθε κόμβος με σκοπό τον εντοπισμό των πακέτων που φτάνουν και φεύγουν προς άλλους καθώς και των σφαλμάτων που μπορεί να προκύψουν κατά τη διάρκεια εκτέλεσης της εφαρμογής.

Το σύστημα υποστηρίζει δύο διαφορετικούς τύπους καταγραφής:

- Console: τα logs αποτυπώνονται στην κονσόλα εκτέλεσης (stdout)
- Αρχεία καταγραφής (Log files): τα logs καταγράφονται σε αρχεία για κάθε κόμβο του συστήματος σύμφωνα με το όνομα που έχει οριστεί στην παραμετροποίηση.

Τα μηνύματα καταγραφής ακολουθούν την εξής μορφή

< Timestamp: yyyy – MM – dd HH: mm: ss. SSS > < Level: Info | Error > < Component > < Message >

## Βασικές Δομές-Components KIOT

Στη συνέχεια περιγράφονται οι βασικές δομές, όπως έχουν υλοποιηθεί και χρησιμοποιούνται στο KIOT.

Δομή	Περιγραφή
BloomFilter	Δομή που χρησιμοποιείται για την μετατροπή ενός συνόλου ετικετών σε Bloom Filter, καθώς και για αποθήκευση του BF καταλόγου του κόμβου, μαζί με όλες τις απαραίτητες λειτουργίες του (σύγκριση BF καταλόγων, συγχώνευση BF καταλόγων κ.α.).
FIB	Αποθηκεύει τον BF κατάλογο και τη διεπαφή (IP Διεύθυνση, πόρτα) όλων των κόμβων-παιδιών ενός ριζικού ή ενδιάμεσου κόμβου.
InterestParameters	Πληροφορίες του πακέτου Interest όπως αυτό στάλθηκε από τον κόμβο Guest (διεπαφή κόμβου Guest, λειτουργία που πρέπει να εφαρμοστεί στα αποτελέσματα).
MsgParentCache	Αποθηκεύει τον αριθμό των μηνυμάτων MSG_PARENT που στάλθηκαν από ένα κόμβο που δεν ανήκει στο ακριβώς επόμενο επίπεδο της τοπολογίας.
PendingInterestData	Αποθηκεύει τα δεδομένα που δεν έχουν σταλεί ακόμα σαν απάντηση σε ένα Interest μήνυμα.
Receiver	Βασική δομή ενός κόμβου KIOT που δέχεται πακέτα όπως αποστέλλονται από άλλους κόμβους στο σύστημα προς επεξεργασία.
Sender	Βασική δομή ενός κόμβου KIOT για την αποστολή πακέτων προς άλλους κόμβους στο σύστημα.
Configuration	Δομή για την ανάγνωση του αρχείου που περιέχει τις ρυθμίσεις-παραμέτρους ενός κόμβου.
Constants	Δομή με αμετάβλητες ρυθμίσεις-παραμέτρους ενός κόμβου.
Processor	Δομή που επεξεργάζεται τα δεδομένα που έχουν συλλεχθεί για ένα Interest πακέτο πριν σταλούν σαν απάντηση στον γονικό κόμβο ή στον χρήστη.
SendMessageTimer	Χρησιμοποιείται για την επιλογή του γονικού κόμβου ανάλογα με την πολιτική που εφαρμόζεται στον κόμβο (RTT, Load-Balancing).
RequestTimerTask	Χρησιμοποιείται την προώθηση όλων των δεδομένων που συλλέχθηκαν από ένα κόμβο για ένα μήνυμα Interest είτε στους κόμβους ανώτερου επιπέδου είτε στον client στις περιπτώσεις που έχει λήξει το δυναμικά ορισμένο χρονικό περιθώριο που έχει στη διάθεση του ο κόμβος.
ChildTimerTask	Χρησιμοποιείται για τον έλεγχο ανά τακτά χρονικά διαστήματα αν κάποιος κόμβος παιδί έχει εξέλθει από το σύστημα.
ClearMsgParentCacheTask	Χρησιμοποιείται για τον καθαρισμό του MsgParentCache ανά τακτά χρονικά διαστήματα.
ParentTimerTask	Χρησιμοποιείται για τον έλεγχο ανά τακτά χρονικά διαστήματα αν ο κόμβος γονέας έχει εξέλθει από το σύστημα.

Πίνακας 1. Βασικές δομές KIOT

## Τύποι Μηνυμάτων

Στη συνέχεια αναφέρονται όλοι οι τύποι μηνυμάτων που χρησιμοποιούνται από τους κόμβους μέσα σε ένα σύστημα KIOT.

Μήνυμα	Περιγραφή	Δεδομένα
MSG_ADVERTISEMENT (ID: 1)	Μήνυμα διαφήμισης BF καταλόγου	BF κατάλογος
MSG_ADVERTISEMENT_ACK (ID: 2)	Μήνυμα επιβεβαίωσης λήψης μηνύματος MSG_ADVERTISEMENT	
MSG_ADVERTISEMENT_UPDATE (ID: 3)	Μήνυμα διαφήμισης BF καταλόγου για επανυπολογισμό από τον γονικό κόμβο.	BF κατάλογος
MSG_INTEREST (ID: 4)	Αιτήματος για δεδομένα από client προς κόμβο ρίζα	BF κατάλογος, Αναγνωριστικό αιτήματος (Μόνο μεταξύ κόμβων εντός συστήματος)
MSG_DATA (ID: 5)	Δεδομένα προς απάντηση σε αίτημα από client	Αναγνωριστικό αιτήματος, Δεδομένα Απάντησης, Αριθμός παιδιών που απάντησαν (Μόνο σε Min-Transfer Στρατηγική)
MSG_DATA_ACK (ID: 6)	Μήνυμα επιβεβαίωσης προς τον γονικό κόμβων λήψης όλων των μηνυμάτων MSG_DATA από τους κόμβους παιδιά.	Αναγνωριστικό αιτήματος
MSG_PARENT (ID: 7)	Αρχικό μήνυμα εκπομπής για την εύρεση γονικού κόμβου	Επίπεδο στην τοπολογία
MSG_PARENT_ACK (ID: 8)	Μήνυμα επιβεβαίωσης από δυνητικό γονικό κόμβο	Αριθμός παιδιών
MSG_PARENT_CHILD (ID: 9)	Μήνυμα επιβεβαίωσης επιλογής γονέα προς γονικό κόμβο	

Πίνακας 2. Τύποι μηνυμάτων

## Παραμετροποίηση Κόμβων

Στη συνέχεια παρουσιάζεται το configuration ενός κόμβου, το οποίο είναι κρίσιμο για τη σωστή λειτουργία και απόδοση του συστήματος. Το configuration περιλαμβάνει τις παραμέτρους που καθορίζουν τη συμπεριφορά και τις δυνατότητες του κόμβου. Ακολουθεί ο πίνακας με τις λεπτομέρειες του configuration κάθε κόμβου.

Παράμετρος	Περιγραφή
NodeID	Αναγνωριστικό κόμβου
Level	Επίπεδο στο οποίο ανήκει ο κόμβος
Type	Τύπος κόμβου (0 για ριζικό, 1 για ενδιάμεσο, 2 για κόμβο-φύλλο)

OutputFile	Τοποθεσία αρχείου καταγραφής (log file)
LoggingMode	Λειτουργία καταγραφής (0 για stdout, stderr, 1 για αρχείο, 2 για 1 και 2)
BroadcastIP	IP διεύθυνση εκπομπής
TotalLevels	Συνολικός αριθμός επιπέδων στο δίκτυο
Strategy	Στρατηγική που θα εφαρμοστεί (NAIVE ή MIN-TRANSFER)
Tags	Κατάλογος ετικετών (Μόνο σε κόμβους φύλλα)

Πίνακας 3. Παράμετροι κόμβου

## Επεξεργασία δεδομένων

Οι λειτουργίες που υποστηρίζει το σύστημα στα δεδομένα που αποτελούν την απάντηση σε αιτήματα από clients τόσο κατά την στρατηγική naïve όσο και κατά την στρατηγική minimum transfer είναι τα ακόλουθα.

Λειτουργία	Επεξήγηση
NOTHING (ID: 0)	Τα δεδομένα αποστέλλονται χωρίς καμία επεξεργασία προς τον client
AVG (ID: 1)	Αποστέλλεται η μέση τιμή από τα δεδομένα που προέκυψαν
MAX (ID: 2)	Αποστέλλεται η μέγιστη τιμή από τα δεδομένα που προέκυψαν
MIN (ID: 3)	Αποστέλλεται η μικρότερη τιμή από τα δεδομένα που προέκυψαν

Πίνακας 4. Λειτουργίες στα δεδομένα προς απάντηση

## Πειράματα

Σε αυτή την ενότητα παρουσιάζονται τα αποτελέσματα από μερικά απλά πειράματα που αξιοποιούν και τις δύο στρατηγικές υπολογισμού του δικτύου, Naive και Minimum Transfer, για διαφορετικό αριθμό κόμβων μέσα στο δίκτυο.

Η εκτέλεση των πειραμάτων έγινε με την χρήση του Mininet για την δημιουργία ενός εικονικού δικτύου με κοινό δρομολογητή για όλους τους κόμβους. Επιπλέον, έγινε χρήση του Docker για την εκτέλεση πειραμάτων μικρής κλίμακας ώστε να είναι εφικτή η αναπαραγωγή του σεναρίου όπου ένας κόμβος εξέρχεται από το σύστημα και εισέρχεται ύστερα από κάποιο χρονικό διάστημα, σενάριο που δεν ήταν δυνατόν να αναπαραχθεί μέσω του Command Line Interface (CLI) που παρέχει το Mininet.

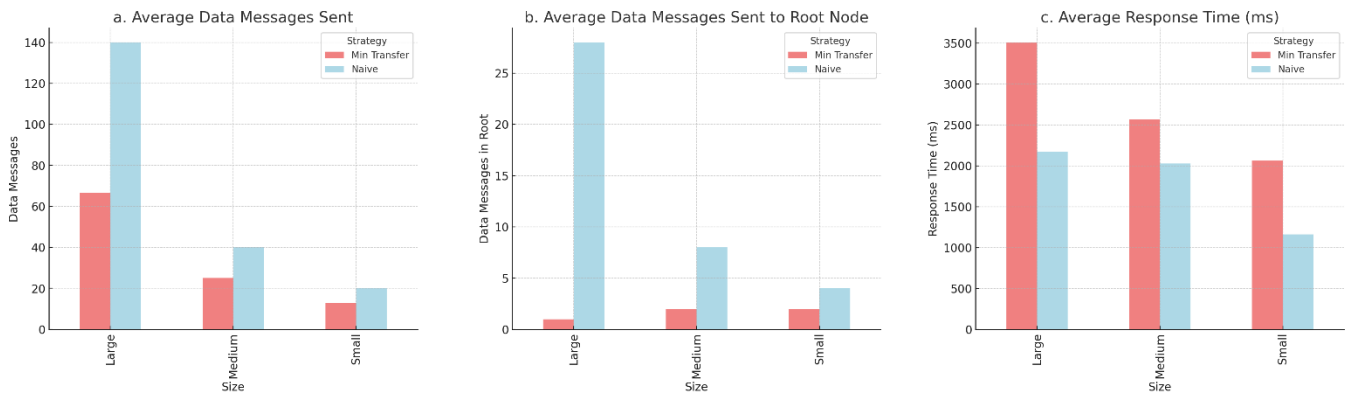
Πιο συγκεκριμένα, τα σενάρια έτρεξαν για μικρό, μεσαίο και μεγάλο αριθμό κόμβων με 9, 35 και 101 κόμβους αντίστοιχα. Η τοπολογία του δικτύου περιλαμβάνει 6 επίπεδα με την υπόθεση ότι υπάρχει πάντα ένας ριζικός κόμβος (πρώτο επίπεδο) που δέχεται αιτήματα από χρήστες (τον κόμβο Guest). Η επιλογή του αριθμού κόμβων σε κάθε επίπεδο κατά την αρχικοποίηση της τοπολογίας γινόταν τυχαία, με τον αριθμό αυτό να βρίσκεται μέσα σε ένα προκαθορισμένο εύρος ανάλογα με το εκάστοτε επίπεδο. Για παράδειγμα για το προτελευταίο επίπεδο τα όρια είχαν καθοριστεί σε [1,2] για μικρά σενάρια, [1,4] για μεσαία σενάρια και [2,4] για μεγάλα σενάρια. Ο συνολικός αριθμός των κόμβων της τοπολογίας πρόκυπτε πολλαπλασιάζοντας τους επιλεγμένους αριθμούς κόμβων κάθε επιπέδου. Για παράδειγμα αν οι επιλεγμένοι αριθμοί είναι 1, 2, 4, 4, 1 για τα επίπεδα 1, 2, 3, 4, 5 αντίστοιχα, ο συνολικός αριθμός κόμβων είναι  $1*2*4*4*1 + 1$  (για τον ριζικό κόμβο).

Κάθε κόμβος-φύλλο περιλάμβανε διαφορετικό, άλλα ίδιου μεγέθους, σύνολο από ετικέτες (tags). Τα σύνολα των ετικετών που υπάρχουν διαθέσιμα προς επιλογή από τους κόμβους φύλλα ορίζονταν δυναμικά ανάλογα με τις προαναφερθείς επιλογές του αριθμού κόμβου σε κάθε επίπεδο, με κάθε ετικέτα να περιέχει ένα αναγνωριστικό του επιπέδου στο οποίο αναφέρεται (b για κτήριο, f για όροφο, w για πτέρυγα, r για δωμάτιο και s για αισθητήρα) ακολουθούμενο από έναν αύξοντα αριθμό. Έτσι στο σενάριο όπου οι αριθμοί κόμβων που είχαν επιλεγθεί ήταν 1,1,1,2,2 για τα επίπεδα 1 έως 5 αντίστοιχα, τα διαθέσιμα σύνολα ετικετών θα ήταν  $\{b_0, f_0, w_0, r_0, s_0\}$ ,  $\{b_0, f_0, w_0, r_0, s_1\}$ ,  $\{b_0, f_0, w_0, r_1, s_0\}$  και  $\{b_0, f_0, w_0, r_1, s_1\}$  με κάθε κόμβο-φύλλο να έχει στη διάθεση του σύνολα ετικετών που δεν έχουν επιλεγθεί από άλλους κόμβους. Οι ετικέτες περιέγραφαν πάντα το μονοπάτι προς το αισθητήρα και ήταν της μορφής  $b_0, f_0, w_0, r_0, s_0$ , όπου  $b_0$  κτήριο 0,  $f_0$  όροφος 0,  $w_0$  πτέρυγα 0,  $r_0$  δωμάτιο 0 και  $s_0$  αισθητήρας 0. Να σημειωθεί πως το αίτημα ενδιαφέροντος για το μικρό αριθμό κόμβων αποσκοπούσε σε απάντηση από όλους τους κόμβους-φύλλα, ενώ το αίτημα ενδιαφέροντος για το μεσαίο και μεγάλο αριθμό κόμβων αποσκοπούσε σε απάντηση τουλάχιστον από τους μισούς κόμβους-φύλλα.

Για την εκτέλεση των πειραμάτων, χρησιμοποιούνται τρία scripts σε Python:

- **Setup:** Δημιουργία των configurations των κόμβων και των συνόλων ετικετών για τους κόμβους παιδιά, με είσοδο τις ακόλουθες παραμέτρους: Logging Mode, Broadcast IP, η στρατηγική που θα εφαρμοστεί, ο αριθμός των αιτημάτων ενδιαφέροντος που θα σταλούν στον ριζικό κόμβο και το μέγεθος του πειράματος.
- **Topoex:** Εκτέλεση πειράματος χρησιμοποιώντας Mininet, με είσοδο την επιλογή αν οι κόμβοι θα τρέξουν με χρήση Java ή natively.
- **Topoex Docker:** Εκτέλεση του πειράματος χρησιμοποιώντας Docker, με είσοδο την επιλογή αν οι κόμβοι θα τρέξουν με χρήση Java ή natively.

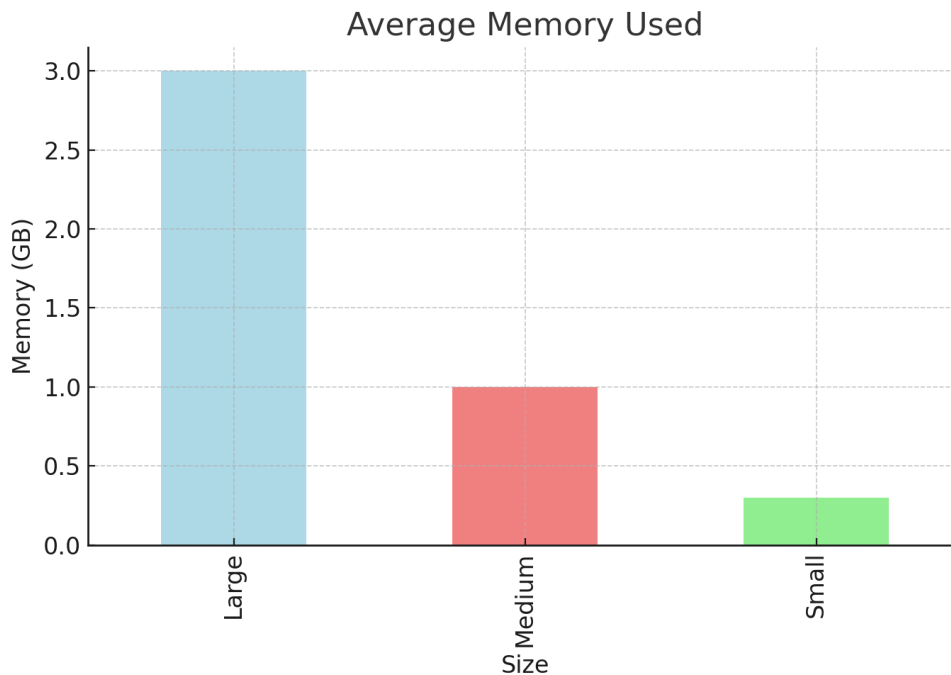
Στην συνέχεια παρουσιάζονται τα αποτελέσματα όπως προέκυψαν από την εκτέλεση των πειραμάτων.



Εικόνα 9. Αποτελέσματα πειραμάτων

Στην εικόνα 9α παρουσιάζεται η μέση τιμή του αριθμού μηνυμάτων MSG\_DATA που στάλθηκαν στο δίκτυο, όπου παρατηρείται πως κατά την Minimum Transfer στρατηγική επιτυγχάνεται η μείωση της αποστολής πακέτων μέσα στο δίκτυο συγκριτικά με την Naive στρατηγική. Στην εικόνα 9b παρουσιάζεται η μέση τιμή του πλήθους των μηνυμάτων MSG\_DATA που στάλθηκαν προς τον ριζικό κόμβο, δείχνοντας πως μειώνεται δραστικά η συμφόρηση μηνυμάτων που λαμβάνονται από τον ριζικό κόμβο κατά την Minimum Transfer στρατηγική. Τέλος στην εικόνα 9c παρουσιάζεται η μέση τιμή της απόκρισης σε αίτημα ενδιαφέροντος από τη στιγμή που το αίτημα λαμβάνεται από τον ριζικό κόμβο μέχρι την στιγμή που η απάντηση στο αίτημα λαμβάνεται από τον client κόμβο, με την Naive στρατηγική να παρουσιάζει μικρότερο χρόνο απόκρισης συγκριτικά με την Minimum Transfer στρατηγική.

Ακόμη παρουσιάζεται η μέση τιμή της συνολικής χρήσης μνήμης για την εκτέλεση των πειραμάτων για τον διαφορετικό αριθμό κόμβων σε κάθε πείραμα.



Εικόνα 10. Μέση χρήση μνήμης

Συμπερασματικά, η επιλογή της στρατηγικής που θα εφαρμοστεί στο σύστημα, είτε Naive είτε Minimum Transfer, εξαρτάται από τη συμφόρηση του δικτύου και την υπολογιστική ισχύ της υποδομής που θα φιλοξενήσει το σύστημα. Η στρατηγική Minimum Transfer είναι πιο κατάλληλη για δίκτυα με υψηλή συμφόρηση και επαρκή υπολογιστική ισχύ τόσο στον ριζικό κόμβο όσο και στους ενδιάμεσους κόμβους. Αντίθετα, η στρατηγική Naive είναι πιο κατάλληλη όταν μόνο ο ριζικός κόμβος διαθέτει επαρκή υπολογιστική ισχύ και ο χρόνος απόκρισης αποτελεί κρίσιμο παράγοντα.



## Συμπεράσματα

Η υιοθέτηση του ICN και των εφαρμογών του, όπως το NDN, προσφέρει σημαντικά πλεονεκτήματα για τα σενάρια IoT, αντιμετωπίζοντας προκλήσεις που σχετίζονται με την παραδοσιακή αρχιτεκτονική βασισμένη στο Cloud. Με τη χρήση ενός συστήματος ονομασίας που επικεντρώνεται στις πληροφορίες αντί για τις συσκευές, το ICN επιτρέπει άμεση και αποδοτική πρόσβαση σε περιεχόμενο, μειώνοντας τη συμφόρηση του δικτύου και εξασφαλίζοντας μεγαλύτερη αξιοπιστία ακόμα και σε συνθήκες αβέβαιης συνδεσιμότητας.

Η προσέγγιση του KIOT, με τη χρήση ετικετών και Bloom Filters, προσφέρει ευελιξία και δυνατότητα προσθήκης νέων κόμβων σε συνθήκες μεγάλου όγκου δεδομένων, καθιστώντας το ICN κατάλληλο για τη διαχείριση μεγάλων και ποικιλόμορφων συνόλων δεδομένων IoT σε τοπικά δίκτυα πριν αυτά διανεμηθούν στο Internet. Αυτές οι εξελίξεις δείχνουν έναν πολλά υποσχόμενο δρόμο για την αποτελεσματική ενσωμάτωση του IoT στην καθημερινή ζωή.

Η παρούσα εργασία επικεντρώθηκε στην εισαγωγή της Minimum Transfer στρατηγικής στην αρχιτεκτονική του KIOT, η οποία αποσκοπεί στη μείωση της κυκλοφορίας δεδομένων και την εξασφάλιση αποτελεσματικής χρήσης του δικτύου και τη σύγκριση της με την προϋπάρχουσα Naive στρατηγική. Επιπλέον, ασχολήθηκε με την εισαγωγή λύσεων για γρήγορη και αποτελεσματική αντιμετώπιση σε περίπτωση που ένας κόμβος βγει από το σύστημα, διασφαλίζοντας την αδιάλειπτη λειτουργία του δικτύου.

Υπάρχουν ωστόσο, περιθώρια για βελτίωση, όπως η προσαρμογή του μεγέθους των Bloom Filters καταλόγων στις περιπτώσεις μεγάλου αριθμού κόμβων στο σύστημα, καθώς και η χρήση lightweight κρυπτογραφίας για την ασφαλή ανταλλαγή πακέτων μεταξύ των κόμβων με περιορισμένη υπολογιστική ισχύ.

## Βιβλιογραφία

1. M. Papalini, A. Carzaniga, K. Khazaei, and A. L. Wolf. Scalable routing for tag-based information-centric networking. In Proceedings of the ACM Conference on Information-Centric Networking, pages 17–26, September 2014.
2. M. Amadeo, C. Campolo, A. Iera, and A. Molinaro. Named data networking for IoT: An architectural perspective. In Proceedings of the 2014 European Conference on Networks and Communications (EuCNC), pages 1–5, June 2014.
3. S. S. Adhatarao, J. Chen, M. Arumaithurai, X. Fu, and K. K. Ramakrishnan. Comparison of naming schema in ICN. In Proceedings of the IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), pages 1–6, June 2016.
4. O. Ascigil, S. Rene, G. Xylomenos, I. Psaras and G. Pavlou. A Keyword-based ICN-IoT Platform. In Proceedings of the ACM Conference on Information-Centric Networking, pages 22-28, September 2017.
5. S. Tarkoma, C. E. Rothenberg and E. Lagerspetz. Theory and Practice of Bloom Filters for Distributed Systems. IEEE Communications Surveys & Tutorials 14.1, pages 131–155, 2012.
6. Xylomenos, G., Zafeiratos, E. and Prokopakis, M. Keyword-Based Information Retrieval For The IoT, ACM/IEEE HoTWoT Workshop, 2019.