

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ  
ΜΕΤΑΠΤΥΧΙΑΚΟ ΔΙΠΛΩΜΑ  
ΕΙΔΙΚΕΥΣΗΣ (MSc)  
στα ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ**

## **ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

**“Keyword-based Routing in IoT devices: Έλεγχος και  
Βελτιστοποίηση”**

**Ευάγγελος Λειβαδάρος  
P3341808**

**Επιβλέπων: Γεώργιος Ξυλωμένος, Καθηγητής**

**ΑΘΗΝΑ, ΟΚΤΩΒΡΙΟΣ 2020**

## Περίληψη

Σε μία εποχή όπου οι περισσότερες, εάν όχι όλες, «έξυπνες» συσκευές διασυνδέονται μεταξύ τους (Internet of Things), καθίσταται απαραίτητη η απρόσκοπτη τους επικοινωνία καθώς και η μεταξύ τους ανταλλαγή δεδομένων. Καθώς οι διασυνδεδεμένες αυτές συσκευές όλο και πληθαίνουν, τόσο στον εργασιακό όσο και στον οικιακό χώρο, οι παραδοσιακές μεθόδους επικοινωνίας, όπως για παράδειγμα η χρήση πρωτοκόλλων με διευθύνσεις IP, δυσχεραίνουν τις συνθήκες για την δημιουργία ιεραρχικών δομών δεδομένων. Αυτό έχει ως αποτέλεσμα, η επεξεργασία των παραγόμενων ή ήδη αποθηκευμένων αποτελεσμάτων, με σκοπό την εκμείωση χρήσιμων συμπερασμάτων, να καθίσταται χρονοβόρα και σε κάποιες περιπτώσεις μεγάλου παραγόμενου όγκου δεδομένων, αδύνατη.

Το KΙΟΤ πρωτόκολλο δρομολόγησης, δημιουργήθηκε με σκοπό την γρήγορη ανταλλαγή πληροφοριών, μεταξύ διασυνδεδεμένων IoT συσκευών ενός δικτύου. Εμπνευσμένο από το TagNet, ένα κεντρικό δίκτυο πληροφοριών με δυνατότητα κλιμάκωσης του βάσει ετικετών (αρχιτεκτονική ICN), το συγκεκριμένο πρωτόκολλο αποτελεί μία τροποποίηση της παραπάνω αναφερόμενης αρχιτεκτονικής με κύρια χαρακτηριστικά τη δημιουργία ενός μοναδικού δέντρου ιεραρχίας και τη χρήση ετικετών, για την εξυπηρέτηση ερωτημάτων μεταξύ των διασυνδεδεμένων κόμβων(συσκευών), θεσπίζοντας ταυτόχρονα τον απλούστερο κανόνα δρομολόγησης: κάθε κόμβος του δικτύου στέλνει δεδομένα πάντα στον «γονιό» του. Αυτό έχει ως αποτέλεσμα, σημαντική μείωση των χρόνων απόκρισης των κόμβων αλλά και αύξηση ταχύτητας της δρομολόγησης των πακέτων, μεταξύ του δικτύου.

Στα πλαίσια επέκτασης και βελτίωσης της υλοποίησης, το KΙΟΤ πρωτόκολλο υποστηρίζει πλέον την εκτέλεση βασικών πράξεων, στα αποθηκευμένα δεδομένα ενός ή περισσότερων κόμβων. Για παράδειγμα, σε ένα δίκτυο το οποίο δρομολογεί τα πακέτα του, με βάση το KΙΟΤ πρωτόκολλο, ο χρήστης είναι σε θέση ανάλογα με την αρχική παραμετροποίηση πριν την εκκίνηση του δικτύου, να βρει το μέγιστο, ελάχιστο ή μέσο όρο ενός αποθηκευμένου συνόλου δεδομένων σε κάποιο συγκεκριμένο κόμβο. Οι συγκεκριμένες λειτουργίες μπορούν να έχουν σημαντικές επιπτώσεις σε βιομηχανικούς αλλά και αστικούς χώρους, για την πρόληψη τυχόν ατυχημάτων και αποτροπή καταστροφής προϊόντων, με βάση τις μετρήσεις θερμοκρασίας, υγρασίας ή φωτισμού.

Σκοπός της παρούσας διπλωματικής εργασίας είναι ο έλεγχος της υφιστάμενης υλοποίησης και η βελτίωση αυτής, με την εύρεση και επίλυση τυχών λογικών λαθών (bugs), σε επίπεδο κώδικα. Η συγκεκριμένη υλοποίηση χρησιμοποιεί τις γλώσσες προγραμματισμού Java, για το επίπεδο της λειτουργικότητας και Python, για το κομμάτι του δικτύου. Δόθηκε ιδιαίτερη έμφαση στο σημείο απόδοσης ετικετών στους κόμβους, με πρωταρχικό στόχο τον έλεγχο υλοποίησης και δευτερεύοντος τον τυχόν ανασχεδιασμό του. Σημαντικές προκλήσεις αποτέλεσαν οι διάφορες εκτελέσεις ολόκληρης της διαδικασίας (end-to-end) καθώς η πολύπλοκη αρχιτεκτονική και η από κατασκευής δύσκολη αποσφαλμάτωση δυσχεραίνουν την απόφαση συμπερασμάτων.

# Περιεχόμενα

|   |    |
|---|----|
| <b>Εισαγωγή</b> .....                     | 4  |
| <b>Υπόβαθρο</b> .....                     | 5  |
| ICN - Information-Centric Networking..... | 5  |
| NDN - Named Data Networking .....         | 5  |
| KIOT - Keyword-Based IoT .....            | 6  |
| <b>Υλοποίηση</b> .....                    | 7  |
| Τύποι Κόμβων.....                         | 7  |
| Τύποι Πακέτων.....                        | 8  |
| Τύποι Μηνυμάτων .....                     | 9  |
| <b>Αλγόριθμος</b> .....                   | 10 |
| <b>Βελτιστοποίηση</b> .....               | 11 |
| <b>Παράδειγμα Εκτέλεσης</b> .....         | 12 |
| Create_Config_Files.....                  | 12 |
| Τοροex .....                              | 13 |
| <b>Βιβλιογραφικές Πηγές</b> .....         | 14 |

## Εισαγωγή

Το Διαδίκτυο των πραγμάτων είναι μια νέα αλλαγή παραδείγματος στον χώρο της πληροφορικής. Η φράση «Ίντερνετ των πραγμάτων», η οποία είναι επίσης σύντομα γνωστή ως IoT, δημιουργήθηκε από τις δύο λέξεις, δηλαδή η πρώτη λέξη είναι «Διαδίκτυο» και η δεύτερη λέξη «Πράγματα». Το Διαδίκτυο είναι ένα παγκόσμιο σύστημα διασυνδεδεμένων δικτύων υπολογιστών που χρησιμοποιούν την τυπική σουίτα πρωτοκόλλου Διαδικτύου (TCP / IP) για την εξυπηρέτηση δισεκατομμυρίων χρηστών παγκοσμίως. Είναι ένα δίκτυο δικτύων που αποτελείται από εκατομμύρια ιδιωτικά, δημόσια, ακαδημαϊκά, επιχειρηματικά και κυβερνητικά δίκτυα, τοπικού έως παγκόσμιου πεδίου, τα οποία συνδέονται με ένα ευρύ φάσμα ηλεκτρονικών, ασύρματων και οπτικών δικτύων τεχνολογίας [1].

Δεν υπάρχει διαθέσιμος μοναδικός ορισμός για το Internet of Things που να είναι αποδεκτός από την παγκόσμια κοινότητα των χρηστών. Στην πραγματικότητα, υπάρχουν πολλές διαφορετικές ομάδες, συμπεριλαμβανομένων ακαδημαϊκών, ερευνητών, επαγγελματιών, καινοτόμων, αναπτυξιακών και εταιρικών ανθρώπων που έχουν ορίσει τον όρο, αν και η αρχική του χρήση έχει αποδοθεί στον Kevin Ashton, εμπειρογνώμονα στην ψηφιακή καινοτομία. Αυτό που όλοι έχουν όλοι οι ορισμοί είναι η ιδέα ότι η πρώτη έκδοση του Διαδικτύου αφορούσε τα δεδομένα που δημιουργήθηκαν από ανθρώπους, ενώ η επόμενη έκδοση αφορά τα δεδομένα που δημιουργήθηκαν από πράγματα. Ο καλύτερος ορισμός για το Διαδίκτυο των πραγμάτων θα ήταν:

"Ένα ανοιχτό και ολοκληρωμένο δίκτυο ευφυών αντικειμένων που έχουν την ικανότητα να οργανώνουν αυτόματα, να μοιράζονται πληροφορίες, δεδομένα και πόρους, να αντιδρούν και να ενεργούν όταν αντιμετωπίζουν καταστάσεις και αλλαγές στο περιβάλλον"[2].

Το Διαδίκτυο των πραγμάτων ωριμάζει και συνεχίζει να είναι η τελευταία, πιο υποθετική ιδέα στον κόσμο της πληροφορικής. Κατά την τελευταία δεκαετία, ο όρος Ίντερνετ των πραγμάτων (IoT) έχει προσελκύσει την προσοχή προβάλλοντας το όραμα μιας παγκόσμιας υποδομής δικτύων φυσικών αντικειμένων, επιτρέποντας ανά πάσα στιγμή, σύνδεση οπουδήποτε για οτιδήποτε και όχι μόνο για κανένα [3]. Το Διαδίκτυο των πραγμάτων μπορεί επίσης να θεωρηθεί ως ένα παγκόσμιο δίκτυο που επιτρέπει την επικοινωνία μεταξύ ανθρώπου-προς-άνθρωπο, από άνθρωπο σε πράγματα και πράγματα-προς-πράγματα, που είναι οτιδήποτε στον κόσμο παρέχοντας μοναδική ταυτότητα σε κάθε αντικείμενο [4]. Το IoT περιγράφει έναν κόσμο όπου σχεδόν οτιδήποτε μπορεί να συνδεθεί και να επικοινωνεί με έξυπνο τρόπο που ποτέ άλλοτε. Οι περισσότεροι από εμάς σκέφτονται να «συνδεθούμε» σε όρους ηλεκτρονικών συσκευών όπως διακομιστές, υπολογιστές, tablet, τηλέφωνα και έξυπνα τηλέφωνα. Σε αυτό που ονομάζεται Διαδίκτυο των πραγμάτων, οι αισθητήρες και οι ενεργοποιητές που είναι ενσωματωμένοι σε φυσικά αντικείμενα — από οδούς έως βηματοδότες — συνδέονται μέσω ασύρματων και ασύρματων δικτύων, συχνά χρησιμοποιώντας την ίδια IP Internet που συνδέει το Διαδίκτυο. Αυτά τα δίκτυα δημιουργούν τεράστιους όγκους δεδομένων που ρέουν σε υπολογιστές για ανάλυση. Όταν τα αντικείμενα μπορούν να αισθανθούν το περιβάλλον και να επικοινωνήσουν, γίνονται εργαλεία για την κατανόηση της πολυπλοκότητας και την γρήγορη ανταπόκρισή τους. Αυτό που είναι επαναστατικό σε όλα αυτά είναι ότι αυτά τα φυσικά συστήματα πληροφοριών αρχίζουν τώρα να αναπτύσσονται και μερικά από αυτά λειτουργούν ακόμη και σε μεγάλο βαθμό χωρίς ανθρώπινη παρέμβαση. Το «Ίντερνετ των πραγμάτων» αναφέρεται στην κωδικοποίηση και τη δικτύωση καθημερινών αντικειμένων και πραγμάτων που τα καθιστούν ξεχωριστά αναγνώσιμα από τον υπολογιστή και ανιχνεύσιμα στο Διαδίκτυο [5] - [6].

# Υπόβαθρο

## ICN - Information-Centric Networking

Η πληροφορικό-κεντρική δικτύωση (ICN) είναι μία αρχιτεκτονική, η οποία στοχεύει και δίνει έμφαση στο περιεχόμενο των αποθηκευμένων αντικειμένων στους κόμβους ή τους δρομολογητές του δικτύου καθώς και στην άμεση πρόσβαση του περιεχομένου από οποιαδήποτε αρχιτεκτονική διαφορετική από την τρέχουσα, όπου βρίσκεται αποθηκευμένο και προσβάσιμο από τους κεντρικούς υπολογιστές, όπως διακομιστές [7]. Η αρχιτεκτονική προορίζεται να επιτρέψει την αποθήκευση στο δίκτυο, για προσωρινή αποθήκευση περιεχομένων, αποσύνδεση ενός αποστολέα και δέκτη περιεχομένου και επίσης για ενεργοποίηση πολλαπλών μερών επικοινωνία μέσω αναπαραγωγής. Ο κύριος σκοπός είναι να υπάρχει μια αξιόπιστη διανομή περιεχομένου μέσω μιας αποτελεσματικής πλατφόρμας επικοινωνίας και μία παροχή υπηρεσιών που διατίθεται σε ειδικά συστήματα όπως ιδιόκτητα δίκτυα διανομής περιεχομένου και peer-to επικαλύψεις peer (P2P) [8]. Η ιδέα πίσω από το ICN είναι να αναπτυχθεί ένα δίκτυο που θα ερμηνεύει, επεξεργάζεται και παραδίδει πληροφορίες περιεχομένου αυτόματα και ανεξάρτητα από τη φυσική του θέση. Η συγκεκριμένη αρχιτεκτονική βρίσκεται στο προσκήνιο της επιστήμης της Πληροφορικής τα τελευταία χρόνια. Διάφορες ερευνητικές πρωτοβουλίες στοχεύουν στον συγκεκριμένο αναδυόμενο ερευνητικό χώρο, με στόχο την μετάβαση από τις τρέχον αρχιτεκτονικές δικτύων, που έχουν δημιουργηθεί και σχεδιαστεί με βάση αρχών επικοινωνίας μεταξύ των κόμβων (end-to-end hosts), σε αρχιτεκτονικές οι οποίες δραστηριοποιούνται γύρω από τα δεδομένα του δικτύου, με σκοπό την βέλτιστη και πληροφορικό-κεντρική δρομολόγηση.

## NDN - Named Data Networking

Η Δικτύωση Ονοματισμένων Δεδομένων (NDN) είναι μία αρχιτεκτονική προσέγγιση η οποία ασπάζεται τις βασικές αρχές της ICN. Η συγκεκριμένη αρχιτεκτονική εκμεταλλεύεται το πλεονέκτημα ενός καθολικού κεντρικού επίπεδο με μια απλή διεπαφή επικοινωνίας παρόμοια με αυτήν που προτείνεται για το Πρωτόκολλο Ελέγχου Μεταφοράς και Διαδικτύου (TCP/IP) [9]. Η αρχιτεκτονική NDN δανείζεται ορισμένες ιδιότητες από το TCP/IP σουίτα. Για παράδειγμα, η δρομολόγηση αποσυνδέεται από την προώθηση (forwarding) αλλά λειτουργεί παράλληλα (τα πακέτα προωθούνται ενώ οι πίνακες δρομολόγησης συνεχίζουν να εξελίσσονται με την πάροδο του χρόνου) με τα τρέχοντα δίκτυα IP. Ωστόσο, η επικοινωνία NDN καθοδηγείται από δέκτη. Ένας καταναλωτής πρόθυμος για λήψη δεδομένων στέλνει ένα πακέτο ενδιαφέροντος, καθορίζοντας το όνομα (αναγνωριστικό) του επιθυμητού περιεχομένου. Ένας δρομολογητής καταγράφει τη διεπαφή αιτήσεων και στη συνέχεια την προωθεί σύμφωνα με τη βάση του ενδιαφέροντος προώθησης (FIB), η οποία είναι ένας πίνακας που αποθηκεύει διαδρομές που συμπληρώνονται από ένα πρωτόκολλο δρομολόγησης βάσει ονόματος. Όταν το πακέτο ενδιαφέροντος φτάσει σε έναν κόμβο που έχει (αλλά όχι απαραίτητα αυτόν που δημιούργησε) τα ζητούμενα δεδομένα, ένα πακέτο δεδομένων αποστέλλεται πίσω. Ένα πακέτο δεδομένων φέρει το όνομα του ζητούμενου περιεχομένου και το ίδιο περιεχόμενο καθώς και τα δύο υπογράφονται με το κλειδί του παραγωγού. Το πακέτο δεδομένων ακολουθεί τη διαδρομή που δημιούργησε το πακέτο ενδιαφερόντων κατά την επιστροφή του στον καταναλωτή. Ούτε τα δεδομένα ούτε το

ενδιαφέρον περιέχουν οποιεσδήποτε πληροφορίες (διευθύνσεις IP) που σχετίζονται με τους κεντρικούς υπολογιστές που διασχίζουν. Τα πακέτα ενδιαφέροντος δρομολογούνται προς τους παραγωγούς δεδομένων με βάση το όνομα που μεταφέρεται στο πακέτο. Τέλος, τα πακέτα δεδομένων επιστρέφονται με βάση τις δηλωμένες πληροφορίες που συλλέγονται από το πακέτο ενδιαφέροντος σε κάθε δρομολογητή [10].

## KIOT - Keyword-Based IoT

Μία περισσότερο ευέλικτη προσέγγιση της Δικτύωσης Ονοματισμένων Δεδομένων (NDN), είναι η αυτή της KIOT, η οποία και χρησιμοποιείται στην συγκεκριμένη υλοποίηση. Χρησιμοποιεί παρόμοια λογική ονοματοδοσίας με αυτή της τεχνολογίας TagNet, μια αρχιτεκτονική ICN σχεδιασμένη να υποστηρίξει «push and pull» ροές με δύο ανεξάρτητες μεθόδους διεύθυνσης και παράδοσης: μία με περιγραφές περιεχομένων και μία με εντοπισμό κεντρικών κόμβων [11]. Η συγκεκριμένη ονοματοδοσία ωστόσο, διαφοροποιείται στην σχέση αντιστοίχισης, η οποία πλέον ορίζεται ως μια σχέση υπερσυνόλου μεταξύ ενός συνόλου λέξεων-κλειδιών και των κόμβων. Η δρομολόγηση λαμβάνει χώρα σε ένα μοναδικό δέντρο κόμβων, όπου ο κόμβος ρίζας είναι ο σύνδεσμος προς το δέντρο δρομολόγησης. Ακόμα, όλα τα αιτήματα προς το δέντρο προέρχονται από τον κόμβο ρίζας και οι τελικές απαντήσεις ταξιδεύουν μέχρι να φτάσουν σε αυτόν. Συνεπώς κάθε κόμβος οφείλει να γνωρίζει μόνο τις συνδέσεις που οδηγούν προς τον κόμβο ρίζας για να κατευθύνει τις απαντήσεις. Για τον εντοπισμό κάθε αντικειμένου και του περιεχομένου του, χρησιμοποιείται ένα Bloom Filter το οποίο υποδεικνύει τις λέξεις – κλειδιά που σχετίζονται με αυτό. Στην συνέχεια σε κάθε κόμβο του δέντρου διατηρείται ένας κατάλογος με τους συνδέσμους προς τους κόμβους του κατώτερου επιπέδου, δηλαδή μία λίστα με τα Bloom Filters η οποία δείχνει ποια αντικείμενα είναι διαθέσιμα μέσω ενός συνδέσμου. Σε αντίθεση με το TagNet, ένα αίτημα στη συγκεκριμένη αρχιτεκτονική ταιριάζει με ένα αντικείμενο περιεχομένου όταν το Bloom Filter του είναι υποσύνολο του Bloom Filter του αντικειμένου. Αυτό επιτρέπει τη συγχώνευση των Bloom Filters προς την κατεύθυνση του κόμβου ρίζας, με το κόστος της δημιουργίας false positives[12].

## Υλοποίηση

Η τοπολογία της KΙΟΤ αρχιτεκτονικής έχει την μορφή δέντρου (tree topology). Παρακάτω περιγράφονται τα βασικά συστατικά στοιχεία της υλοποίησης: τα διαφορετικά είδη των κόμβων, των πακέτων και των μηνυμάτων, καθώς και ο αλγόριθμος εκτέλεσης της διαδικασίας δρομολόγησης.

### Τύποι Κόμβων

- **Leaf Nodes:** Είναι οι κόμβοι, οι οποίοι αποτελούν τα άκρα του δέντρου. Κάθε κόμβος-φύλλο συνδέεται με έναν γονέα, για τον οποίο γνωρίζει μόνο την διεύθυνση IP, παράγει ονοματισμένα δεδομένα και περιέχει πληροφορίες όπως τις λέξεις κλειδιά που εξυπηρετούν, οι οποίες ονομάζονται BF-Keyword. Το σύνολο λέξεων-κλειδιών που εξυπηρετεί κάθε κόμβος ονομάζεται BF-Catalog και αποθηκεύεται στους κόμβους-φύλλα του δέντρου.
- **Interior Nodes:** Είναι οι ενδιάμεσοι κόμβοι του δέντρου με ανάμεσα στην ρίζα και τα φύλλα του δέντρου και είναι υπεύθυνοι για την επικοινωνία του κόμβου-ρίζα με τους κόμβους-φύλλων. Αξίζει να σημειωθεί πως στο δέντρο μπορεί να έχει πολλά επίπεδα και κάθε μεσαίος κόμβος πολλά κόμβους-παιδιά. Οι συγκεκριμένοι κόμβοι έχουν ένα πίνακα δρομολόγησης, ο οποίος αποτελείται από δύο στήλες: η μία στήλη έχει αποθηκευμένους του εισερχόμενους Bloom Filter καταλόγους (BF-Catalog) από τους κόμβους-γονείς ενώ η δεύτερη έχει αποθηκευμένες τις διευθύνσεις IP των κόμβων-παιδιών, για τα οποία ανήκει ο εκάστοτε Bloom Filter κατάλογος. Γνωρίζουν επίσης και την διεύθυνση IP των κόμβων-γονέων τους.
- **Root Node:** Ο συγκεκριμένος κόμβος έχει τα ίδια χαρακτηριστικά με έναν μεσαίο κόμβο, χωρίς ωστόσο να έχει κανένα κόμβο γονέα, εφόσον είναι και ο αρχικός κόμβος του δικτύου. Τα ερωτήματα της εφαρμογής γίνονται σε αυτόν τον κόμβο καθώς και τα αποτελέσματα απάντησης κάθε άλλου κόμβου καταλήγουν σε αυτόν. Αξίζει να σημειωθεί ότι ο συγκεκριμένος κόμβος δεν παράγει ερωτήματα προς τους υπόλοιπους κόμβους αλλά προωθεί τα ερωτήματα που τυχόν δέχεται από άλλες εφαρμογές ή σημεία εισαγωγής ερωτημάτων ενός δικτύου. Τέλος, ο συγκεκριμένος κόμβος κρατάει μνήμη της σειράς των ερωτημάτων που έγιναν στον συγκεκριμένο δίκτυο, σκοπό την σωστή αντιστοίχιση των εκάστοτε απαντήσεων με των αντίστοιχων ερωτημάτων.

Το σύστημα είναι αυτό-οργανωμένο, απαιτώντας οι κόμβοι να γνωρίζουν μόνο το επίπεδο 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. Για παράδειγμα, εάν σε ένα κτίριο τον ρόλο των κόμβων-φύλλων τον είχαν αισθητήρες θερμοκρασίας, οι οποίοι θα περιγράφονταν με την λέξη κλειδί 01000010, και αισθητήρες υγρασίας, οι οποίοι περιγράφονταν με την λέξη κλειδί 00100100, η τιμή του BF-Catalog θα ήταν 01100110, όπου αποτελεί την ένωση (λογικό OR) των δύο παραπάνω τιμών των λέξεων-κλειδιών.

## Τύποι Πακέτων

- **Parent packets:** Όλοι οι κόμβοι, εξαιρουμένου του κόμβου ρίζα, πρέπει να βρουν έναν κατάλληλο γονέα για να μπορούν να επικοινωνούν μαζί του, με σκοπό να στέλνουν τα απαραίτητα δεδομένα. Η συγκεκριμένη ενέργεια λαμβάνει χώρα στην φάση της αρχικοποίησης του συστήματος. Το μόνο που είναι απαραίτητο να γνωρίζουμε είναι το επίπεδο διαμονής των κόμβων. Αυτό σημαίνει ότι κάθε κόμβος στέλνει αρχικά ένα μήνυμα εκπομπής στο δίκτυο, στο οποίο βρίσκεται, και έπειτα όλοι οι κόμβοι που λαμβάνουν το μήνυμα αυτό είναι ικανοί να αποφασίσουν εάν είναι ή όχι, δυνητικοί γονείς του επίμαχου αυτού κόμβου. Οι δυνητικοί γονείς είναι εκείνοι οι κόμβοι, οι οποίοι βρίσκονται στο ανώτερο επίπεδο του αιτούντος κόμβου. Στην υλοποίηση αυτή, υπάρχουν δύο συγκεκριμένες πολιτικές που ακολουθούνται με σκοπό να βρεθούν αυτοί οι ανώτεροι σύνδεσμοι (κόμβοι-γονείς):
  - **Load Balancing:** Η συγκεκριμένη πολιτική στηρίζεται στον αριθμό των παιδιών των δυνητικών κόμβων-γονέων. Όταν ένας κόμβος στέλνει την εκπομπή, οι δυνητικοί γονείς θα απαντήσουν. Η απάντηση αυτή περιέχει τον αριθμό των κόμβων-παιδιών. Ο κόμβος με τα λιγότερα παιδιά θα γίνει ο κόμβος-γονέας, του αιτούντος κόμβου. Στην περίπτωση που όλοι οι μελλοντικοί γονείς έχουν τον ίδιο αριθμό παιδιών, τότε επιλέγεται τυχαία.
  - **Delay Minimization:** Η πολιτική αυτή είναι βασισμένη στο Round Trip Time (RTT) μεταξύ κόμβων. Στην περίπτωση αυτή, όταν ένας κόμβος αιτείται να λάβει απάντηση από δυνητικούς γονείς, θα λάβει επίσης και δεδομένα σχετικά με το RTT. Ο κόμβος με την μικρότερη RTT τιμή, θα είναι ο κόμβος-γονέας του αιτούντος κόμβου.
- **Advertisement packets:** Η συγκεκριμένη ενέργεια γίνεται τη στιγμή όπου ένας κόμβος-φύλλο ξεκινήσει την διαδικασία. Οι λέξεις-κλειδιά, όπου διαφημίζουν τις προσφερόμενες πληροφορίες, στέλνονται από κάθε κόμβος (bottom-to-top), και οποίες αποθηκεύονται στον πίνακα δρομολόγησης του κόμβου-γονέα του. Αυτός, με τη σειρά του, αποθηκεύει τη διεύθυνση και τις πληροφορίες, οι οποίες προέρχονται από ανώτερους κόμβους-γονείς. Ο εκάστοτε κόμβος που θα λάβει το διαφημιστικό αυτό μήνυμα, εκτελεί μια συνένωση (λογικό OR) μεταξύ του BF που έλαβε και του δικού του BF και το διαφημίζει στον δικό του κόμβο-γονέα, εάν υπάρχει. Στην συνέχεια, ο κόμβος γονέας θα ακολουθήσει την ίδια διαδικασία.
- **Interest packets:** Ακολουθείται η παρακάτω διαδικασία για κάθε αίτημα ενδιαφέροντος. Γίνεται αποστολή του αιτήματος στον κόμβο-ρίζα. Ο κόμβος-ρίζα, ελέγχει την ύπαρξη του ονόματος του, στο ερώτημα, εξετάζοντας τον κατάλογο BF. Εν συνεχεία, γίνεται έλεγχος εάν το αίτημα είναι υποσύνολο κανενός από τους καταλόγους BF του FIB πίνακα, του κόμβου-ρίζα. Εάν κάτι τέτοιο ισχύει, τότε το αίτημα ερωτήματος προωθείται έναν ή περισσότερους κόμβους ανωτέρων επιπέδων, των αντίστοιχων αποτελεσμάτων. Με δεδομένο την αποθήκευση των διευθύνσεων IP, ο κόμβος-γονέας γνωρίζει πού να σταλεί το αίτημα, με βάση τα δεδομένα του καταλόγου BF. Αυτό επαναλαμβάνεται μέχρι το αίτημα στους, να φτάσει στους κόμβους-φύλλα, όπου οι πραγματικές πληροφορίες είναι αποθηκευμένες.



- **Data packets:** Τα συγκεκριμένα πακέτα είναι τα αποτελέσματα ένωσης του αντίστοιχου ερωτήματος. Τη στιγμή που θα υπάρξει πρόσβαση σε κατάλληλους κόμβους και έπειτα από την ανάκτηση των πληροφοριών, τα αποτελέσματα θα ξανά σταλούν σε ανώτερα επίπεδα του δέντρου έως ότου καταλήξουν στον κόμβο-ρίζα.

## Τύποι Μηνυμάτων

Για την καλύτερη κατανόηση της λειτουργίας του συστήματος, θα πρέπει να περιγραφεί η διαδικασία αρχικοποίησης αλλά και αποστολής ενός αιτήματος, καθώς και η διαδικασία λήψης μιας απάντησης, η οποία αποτελεί αναπόσπαστο κομμάτι της ολόκληρης της λειτουργικότητας.

- A.** Στη φάση αρχικοποίησης του συστήματός μας, κάθε κόμβος στέλνει ένα μήνυμα εκπομπής MSG\_PARENT για να αφήσει τους άλλους κόμβους να γνωρίζουν ότι ψάχνει γονέα. Οι κόμβοι στο ανώτερο επίπεδο θα απαντήσουν στον αιτούντα με το MSG\_PARENT\_ACK. Επιλέγοντας την πολιτική εξισορρόπησης φορτίου (Load Balancing), η οποία περιγράφεται παραπάνω, ο επίμαχος κόμβος θα επιλέξει έναν άλλο κόμβο με τα λιγότερα παιδιά, στέλνοντας ένα μήνυμα MSG\_PARENT\_CHILD στον επιλεγμένο γονικό του κόμβο, με σκοπό να είναι ικανός να αυξήσει τους κόμβους-παιδιά του. Σε περίπτωση ελαχιστοποίησης καθυστέρησης, υπολογίζουμε τους χρόνους RTT όλων των πιθανών γονέων και επιλέγουμε το κόμβο με τον συντομότερο χρόνο.
- B.** Όταν όλοι οι κόμβοι βρουν τους γονείς τους, περνάμε σε αυτή τη φάση της «διαφήμισης». Με την έννοια «διαφήμιση», εννοούμε την εκπομπή του περιεχομένου των αισθητήρων (φωτισμός, υγρασία, θερμοκρασία κ.α), από τους κόμβους-φύλλα. Σε αυτό το βήμα, οι κόμβοι φύλλων στέλνουν MSG\_ADVERTISEMENT μηνύματα στους γονείς τους, με τα δεδομένα τους. Τώρα, οι κόμβοι γονείς στέλνουν τις συγκεντρωτικές πληροφορίες στους γονείς τους και αυτή η διαδικασία επαναλαμβάνεται μέχρι να καταλήξουμε στον κόμβο-ρίζα. Κάθε μήνυμα MSG\_ADVERTISEMENT επιβεβαιώνεται από ένα MSG\_ADVERTISEMENT\_ACK μήνυμα αντιστοίχως, με σκοπό να γνωρίζουμε εάν το περιεχόμενο μετακινήθηκε σωστά, από κόμβο σε κόμβο.
- C.** Στο σημείο αυτό, το στήσιμο του συστήματος έχει ολοκληρωθεί. Όλοι οι πίνακες FIB είναι δημιουργημένοι, όλοι οι κόμβοι γνωρίζουν τους γονείς τους και είναι έτοιμοι να επικοινωνήσουν μεταξύ τους. Ένας φιλοξενούμενος κόμβος τώρα, μπορεί να ζητήσει δεδομένα στέλνοντας ένα μήνυμα MSG\_INTEREST με το είδος των πληροφοριών που επιθυμεί, για παράδειγμα (δεύτερος όροφος, Αργυρούπολη, υγρασία), στο κόμβο ρίζα. Ο κόμβος-ρίζα ελέγχει τον FIB πίνακα, με σκοπό να στείλει το μήνυμα αυτό στο κατάλληλο κόμβο-παιδί ή κόμβους-παιδιά ανάλογα με τους καταλόγους εξυπηρέτησης. Οι κόμβοι-παιδιά στη συνέχεια, ελέγχουν τον FIB με σκοπό να κατευθύνουν το πακέτο στους αντίστοιχους κόμβους-παιδιά τους, μέχρι να φτάσουμε στο κάτω μέρος του δέντρου (κόμβοι φύλλων) για να λάβουμε τα δεδομένα που ζητά ο φιλοξενούμενος κόμβος.

- D. Φτάνοντας στους κόμβους-φύλλα του δέντρου, ένα μήνυμα MSG\_DATA με το ζητούμενο περιεχόμενο δρομολογείται από τα κατώτερα επίπεδα προς τα ανώτερα στρώματα (κόμβος-παιδί σε κόμβο-γονέα) μέχρι το συγκεκριμένο περιεχόμενο να φτάσει στη ρίζα.

## Αλγόριθμος

Η συγκεκριμένη υλοποίηση έχει γίνει με την χρήση δύο βασικών γλωσσών προγραμματισμού: την Python, για την διαχείριση δικτύου και την Java, για τον αλγόριθμο. Παρακάτω αναλύονται, επιγραμματικά (high level analysis) οι διαδικασίες που ακολουθούνται στους δύο αυτούς κώδικες.

Για την εκτέλεση των πειραμάτων και όλης της διαδικασίας, χρησιμοποιείται ο προσομοιωτής δικτύου Mininet, ο οποίος περιγράφεται αναλυτικότερα παρακάτω. Ο συγκεκριμένος προσομοιωτής δικτύου έχει μία διεπαφή χρήστη (API), δημιουργημένη σε Python, η οποία μας επιτρέπει να το εκτελούμε τοπικά στον υπολογιστή μας. Το Mininet ξεκινά τη λειτουργία του με τη δημιουργία της τοπολογίας του δικτύου και την ανάθεση τοπικών εντολών κελύφους σε αυτό. Για να οριστεί το δίκτυο πρέπει να δημιουργηθούν τα αρχεία ιδιοτήτων, με ονομασία *createConfigFiles.py*. Με την βοήθεια αυτού του αρχείου, μπορούμε να ορίσουμε στο δίκτυο τη τοπολογία αστέρα με όλους τους κόμβους να συνδέονται στον ίδιο δρομολογητή. Ακόμα, μπορούμε να ορίζουμε το ύψος του δέντρου που θέλουμε να προσομοιώσουμε και να χειριστούν οι KΙΟΤ κόμβοι. Η διαδικασία (function) *get\_nodes* ορίζει πόσους κόμβους θα έχει το κάθε επίπεδο και στη συνέχεια αν είναι κόμβος-φύλλο του προσθέτει τις ετικέτες.

Με την εκκίνηση του προγράμματος και την εκτέλεση της διαδικασίας δημιουργείται το αντικείμενο τοπολογίας, μέσω του python αρχείου πειραμάτων *toroex.py*. Το αντικείμενο της τοπολογίας έχει τις εξής ιδιότητες: τον αριθμό, τις ιδιότητες, τις καθυστερήσεις, τη λίστα και τον δρομολογητή στον οποίο θα συνδεθούν οι κόμβοι.

Τέλος πρέπει να οριστούν οι συνδέσεις μεταξύ δρομολογητών και κόμβων. Αφού ολοκληρωθεί η δημιουργία της τοπολογίας, αναθέτουμε στους κόμβους την εκτέλεση των προγραμμάτων στο παρασκήνιο με τον ορισμό ενός σύνθετου λεκτικού. το οποίο αποτελείται από το πρόγραμμα που θα εκτελεστεί, το αρχείο ρυθμίσεων για του προγράμματος αυτού, τον ειδικό χαρακτήρα «>» για την έξοδο του αποτελέσματος της κονσόλας σε αρχείο χαρακτήρων (text) και τον ειδικό χαρακτήρα «&» (ampersand) για την εκτέλεση στο παρασκήνιο.

## Βελτιστοποίηση

Όπως αναφέρθηκε και παραπάνω, για την προσομοίωση του δικτύου KIoT, χρησιμοποιείται το λογισμικό Mininet. Το Mininet είναι ένας προσομοιωτής δικτύου, ο οποίος δημιουργεί ένα ρεαλιστικό εικονικό δίκτυο, που εκτελεί εφαρμογές σε επίπεδο kernel, σε ένα μόνο εικονικό μηχάνημα (VM). Το συγκεκριμένο λογισμικό αναλαμβάνει να δημιουργεί το σύνολο των κόμβων που απαρτίζουν το δίκτυο εκτελώντας ένα instance της εφαρμογής και τέλος εκτελεί ένα στιγμιότυπο της εφαρμογής. Ένα σύνολο από scripts σε Python χρησιμοποιείται για την παραμετροποίηση των κόμβων και την εκκίνηση λειτουργίας του δικτύου. Η υλοποίηση της εφαρμογής βασίζεται σε πολύνηματική αρχιτεκτονική η οποία αποτελείται κατά βάση από δύο νήματα που αναλαμβάνουν το ρόλο λήψης και αποστολής πακέτων. Επιπρόσθετα, έχει υλοποιηθεί ένα ακόμα νήμα το οποίο χρησιμοποιείται για την κατανάλωση και επεξεργασία των δεδομένων[13]. Η συγκεκριμένη υλοποίηση προσδίδει ένα μεγάλο αριθμό πλεονεκτημάτων για την υποστήριξη του πρωτοκόλλου KIoT, όπως είναι η υποστήριξη πολλαπλών παράλληλων λειτουργιών σε ένα κόμβο χωρίς την ύπαρξη καθυστερήσεων και η ασύγχρονη εκτέλεση ταυτόχρονων ενεργειών για την απάντηση αιτημάτων από εξωτερικούς clients. Επιπρόσθετα, η χρήση του λογισμικού Mininet επιτρέπει την παράλληλη εκτέλεση πολλών στιγμιότυπων της εφαρμογής δημιουργώντας ένα εικονικό δίκτυο, το οποίο θα ήταν δύσκολο να δοκιμαστεί με πραγματικά μηχανήματα. Μπορούν επίσης να δοκιμαστούν διαφορετικές τοπολογίες και χαρακτηριστικά δικτύου[14].

Από την άλλη πλευρά, το μεγαλύτερο μειονέκτημα της συγκεκριμένης υλοποίησης είναι η δυσκολία στην αποσφαλμάτωση και ελέγχου της ορθής λειτουργίας του συστήματος καθώς και των αποτελεσμάτων του. Ακόμα, η αναγκαία μεταφορά των εκτελέσιμων αρχείων στο περιβάλλον του Mininet δημιουργεί ένα επιπλέον φορτίο δυσκολίας καθώς θα πρέπει τα περιβάλλοντα ανάπτυξης και εκτέλεσης να είναι συγχρονισμένα ως προς τις εκδόσεις λογισμικού που χρησιμοποιούν (π.χ. έκδοση Java). Δηλαδή, ο προγραμματιστής και η εικονική μηχανή του Mininet πρέπει να χρησιμοποιούν τις ίδιες βιβλιοθήκες. Για την αντιμετώπιση της συγκεκριμένης δυσκολίας αρχικά αντικαταστάθηκε το εικονικό δίκτυο Mininet με την εκτέλεση του συνόλου των εφαρμογών σε ένα μόνο μηχάνημα, στο οποίο πραγματοποιήθηκε και η ανάπτυξη. Για αυτό το σκοπό χρησιμοποιήθηκαν οι δυνατότητες που προσφέρει το IDE IntelliJ με τη δημιουργία πολλαπλών παραμετροποιήσεων εκτέλεσης της ίδιας εφαρμογής και στη συνέχεια την ομαδοποίησή τους σε ένα group.

Με στόχο την επίλυση των παραπάνω προβλημάτων, η συγκεκριμένη πτυχιακή εργασία εστιάστηκε στην δημιουργία και παραμετροποίηση ενός εικονικού μηχανήματος (VM), με εγκατεστημένα τα αναγκαία προγράμματα (Mininet, IntelliJ IDE), την σωστή έκδοση Java που χρειάζεται η εφαρμογή για την εκτέλεσή της καθώς και έτοιμη την δομή των φακέλων, τα αντικείμενα του κώδικα (compiled objects) και αρχεία οδηγιών προς τον προγραμματιστή για την ορθή χρήση της εφαρμογής, όπως εντολές για την εκτέλεση της εφαρμογής μέσω της γραμμής εντολών.

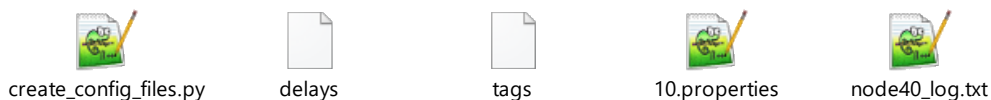
## Παράδειγμα Εκτέλεσης

### Create\_Config\_Files

Όπως προαναφέρθηκε, η υλοποίηση του KΙΟΤ εκτελείτε σε δύο βήματα. Το πρώτο βήμα εκτέλεσης αφορά την αρχικοποίηση του δικτύου με την τοποθέτηση των κόμβων, τον ορισμό των ετικετών (tags) των δεδομένων εξυπηρέτησης καθώς και την δημιουργία εικονικών καθυστερήσεων του δικτύου. Αυτό γίνεται εφικτό με την εκτέλεση του Python προγράμματος *create\_config\_files*. Το συγκεκριμένο αρχείο περιέχει τις εξής διαδικασίες (functions):

- **clear\_dir** → Σε περίπτωση που δοθεί μονοπάτι για φάκελο που ήδη υπάρχει, η συγκεκριμένη διαδικασία είναι υπεύθυνη για να διαγράψει το μονοπάτι που δόθηκε, ως όρισμα στην γραμμή εντολών.
- **get\_nodes** → Η συγκεκριμένη διαδικασία είναι υπεύθυνη να δημιουργήσει όλους τους κόμβους της τοπολογίας (log & properties αρχεία).
- **set\_paths\_and\_tags** → Στη διαδικασία αυτή, υπάρχει η δομή του αρχείου ετικετών (tags).
- **create\_config\_files** → Περιέχει την διαμόρφωση των εκάστοτε αρχείων των κόμβων του δικτύου, ανάλογα την θέση και τον ρόλο τους στο δέντρο. (node#.properties)
- **get\_names** → Δεν χρησιμοποιείται πια.
- **create\_delays\_file** → Η διαδικασία αυτή, δημιουργεί τις τεχνικές καθυστερήσεις του δικτύου, με σκοπό την ενεργοποίηση της Load Balancing πολιτικής (delays).
- **create\_tags\_file** → Με βάση την τοποθεσία που δόθηκε αρχικά για την δημιουργία των αρχείων, η συγκεκριμένη διαδικασία είναι υπεύθυνη να δημιουργήσει το φυσικό αρχείο των ετικετών της υλοποίησης (tags).

Παρακάτω φαίνονται παραδείγματα των προαναφερόμενων αρχείων, τα οποία δημιουργούνται κατά την εκτέλεση του πρώτου βήματος της υλοποίησης, με σκοπό την καλύτερη κατανόηση της εφαρμογής:



Αξίζει να σημειωθεί πως υπάρχει αντιστοιχία των αρχείων «.properties» και «\_log», που ωστόσο δεν είναι τόσο εμφανής. Για να καταλάβουμε ποιο «.properties» αρχείο αντιστοιχεί σε ποιο «\_log» αρχείο, αρκεί να ανοίξουμε το «.properties». Η πληροφορία αναγράφεται στην μεταβλητή «Output file:».

## Τοποex

Εφόσον ολοκληρωθεί το πρώτο βήμα της διαδικασίας του ΚΙΟΤ, δηλαδή η δημιουργία της τοπολογίας του δικτύου, ο χρήστης βρίσκεται σε θέση να εκτελέσει την εφαρμογή. Δηλαδή, το δίκτυο πλέον είναι ικανό να απαντήσει σε οποιοδήποτε ερώτημα ενός χρήστη, με βάση τις ετικέτες που έχουν ανατεθεί στους κόμβους του. Για να επιτευχθεί ο συγκεκριμένος στόχος, πρέπει αρχικά να συμπληρώσουμε το αρχείο «`guest_config.properties`», το οποίο περιέχει τα δεδομένα της εκτέλεσης του ερωτήματος, όπως για παράδειγμα την IP του εικονικού συστήματος (VM) καθώς και την ετικέτα που επιθυμεί ο χρήστης να ελέγξει.



guest\_config.properties

Εφόσον τα απαραίτητα δεδομένα συμπληρωθούν, στο παραπάνω αρχείο, θα πρέπει ο χρήστης να εκτελέσει το python πρόγραμμα «`topoex.py`», στην γραμμή εντολών, με όρισμα την τοποθεσία φακέλου στην οποία έχουν δημιουργηθεί τα αρχεία «`.properties`» και «`_log`», από το προηγούμενο βήμα της εκτέλεσης. Το συγκεκριμένο αρχείο περιέχει τις εξής διαδικασίες:

- **runExperiment** → Η συγκεκριμένη διαδικασία είναι υπεύθυνη για την δρομολόγηση των πακέτων της εφαρμογής ΚΙΟΤ, σε περίπτωση που εξετάζονται οι κόμβοι του δικτύου (KIoT.jar) αλλά και σε περίπτωση όπου γίνεται κάποιο ερώτημα στο δίκτυο και ο χρήστης περιμένει τα αποτελέσματα της εφαρμογής (Guest.jar).
- **\_\_init\_\_** → Στη διαδικασία αυτή γίνονται όλες οι απαραίτητες αρχικοποιήσεις του δικτύου, πριν από την εκτέλεση των ερωτημάτων, όπως για παράδειγμα την IP του οικοδεσπότη (host IP) αλλά και την εύρεση των αρχείων καθυστέρησης (delays) και ετικετών (tags), του προηγούμενου βήματος.



guest\_config.txt



topoex.py

Κατά την ολοκλήρωση της διαδικασίας εκμείευσης των αποτελεσμάτων του ερωτήματος του χρήστη, δημιουργείται το αρχείο «`guest_config`», το οποίο περιέχει αναλυτικά όλα τα ερωτήματα (queries) που εκτελέστηκαν. Αξίζει να σημειωθεί, πως η αναλυτική επεξήγηση του κώδικα των εκτελέσιμων προγραμμάτων σε Java (KIoT & Guest jars) έχει ολοκληρωθεί από την πτυχιακή εργασία του κ. Σκορδούλη.

## Βιβλιογραφικές Πηγές

- [1] Nunberg, G. (2012) The Advent of the Internet: 12th April, Courses.
- [2] Internet of things: A Literature Review. Naved Alam, Prashant Vats, Neha Kashyap.
- [3] Kosmatos, E.A., Tselikas, N.D. and Boucouvalas, A.C. (2011) Integrating RFIDs and Smart Objects into a Unified Internet of Things Architecture. *Advances in Internet of Things: Scientific Research*, 1, 5-12. <http://dx.doi.org/10.4236/ait.2011.11002>
- [4] Aggarwal, R. and Lal Das, M. (2012) RFID Security in the Context of “Internet of Things”. *First International Conference on Security of Internet of Things, Kerala*, 17-19 August 2012, 51-56. <http://dx.doi.org/10.1145/2490428.2490435>
- [5] Biddlecombe, E. (2009) UN Predicts “Internet of Things”. Retrieved July 6.
- [6] Reinhardt, A. (2004) A Machine-to-Machine Internet of Things.
- [7] Alcardo Alex Barakabitze, Tan Xiaoheng, Guo Tan, A Survey on Naming, Name Resolution and Data Routing in Information Centric Networking (ICN).
- [8] G. Carofiglio, G. Morabito, L. Muscariello, I. Solis, M. Varvello, From content delivery today to information centric networking, *Computer Networks* 57 (2013) 3116–3127, July 2013.
- [9] L. Zhang, D. Estrin, J. Burke, V. Jacobson, J. Thornton, D. Smetters, B. Zhang, G. Tsudik, D. Massey, C. Papadopoulos, et al., Named Data Networking (NDN) Project, Tech. Rep., PARC, Tech. Report ndn0001, 2010.
- [10] Luo, S., Zhong, S., & Lei, K. (2018). IP/NDN: A multi-level translation and migration mechanism. *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*.
- [11] Papalini, M., Khazaei, K., Carzaniga, A., & Rogora, D. (2016). High Throughput Forwarding for ICN with Descriptors and Locators. *Proceedings of the 2016 Symposium on Architectures for Networking and Communications Systems*.
- [12] Xylomenos, G., Zafeiratos, E. and Prokopakis, M., “Keyword-Based Information Retrieval For The IoT,” *ACM/IEEE HoTWoT Workshop*, 2019.
- [13] Stylianopoulos G., “KIOT: Εκτέλεση συναρτήσεων και βελτιστοποίηση υλοποίησης σε Java”.
- [14] Skordoulis D., “Προγραμματισμός συστήματος δρομολόγησης με βάση τις λέξεις-κλειδιά στο Διαδίκτυο των Πραγμάτων”.