



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

Αρχιτεκτονική κατανομής πόρων υπολογιστικού
νέφους για την απομακρυσμένη εκτέλεση
εφαρμογών κινητών συσκευών

Διπλωματική Εργασία

του

Νικόλαου Α. Λατζώνη

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

Εργαστήριο Διαχείρισης και Βέλτιστου Σχεδιασμού Δικτύων Τηλεματικής
Αθήνα, Ιούλιος 2019



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων
Πληροφορικής
Εργαστήριο Διαχείρισης και Βέλτιστου Σχεδιασμού Δικτύων
Τηλεματικής

Αρχιτεκτονική κατανομής πόρων υπολογιστικού
νέφους για την απομακρυσμένη εκτέλεση
εφαρμογών κινητών συσκευών

Διπλωματική Εργασία

του

Νικόλαου Α. Λατζώνη

Επιβλέπων: Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 2^η Ιουλίου, 2019.

.....
Συμεών Παπαβασιλείου
Καθηγητής Ε.Μ.Π.

.....
Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π.

.....
Ιωάννα Ρουσσάκη
Επ. Καθηγήτρια Ε.Μ.Π.

Αθήνα, Ιούλιος 2019

.....
Νικόλαος Λατζώνης
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © – All rights reserved Νικόλαος Α. Λατζώνης, 2019.
Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Με την πρόοδο στην τεχνολογία των ασύρματων επικοινωνιών, όλο και περισσότεροι άνθρωποι εξαρτώνται από φορητές κινητές συσκευές για επιχειρήσεις, ψυχαγωγία και κοινωνικές αλληλεπιδράσεις. Μαζί με τη πρόοδο τεχνολογικά αναπτύσσονται ραγδαία και οι εφαρμογές οι οποίες ζητάνε όλο και πιο πολλούς υπολογιστικούς πόρους και δεδομένα. Προκειμένου να διατηρηθεί η υψηλή ποιότητα εξυπηρέτησης των χρηστών η εξ' αποστάσεως εκτέλεση εργασιών που απαιτούν μεγάλη υπολογιστική ισχύ σε ομάδες υπολογιστών γνωστούς ως υπολογιστικά νέφη, έγινε αναπόφευκτη. Ωστόσο, σήμερα θεωρείται ξεπερασμένη η ιδέα της αντιμετώπισης του υπολογιστικού νέφους ως απομονωμένα και απομακρυσμένα κέντρα δεδομένων, καθώς υπάρχουν σαφή οφέλη όταν αυτά είναι τοπικά προσβάσιμα και επίσης συνδέονται μεταξύ τους για να σχηματίσουν ένα δίκτυο (π.χ "έξυπνη" πόλη).

Η ανάγκη βελτιστοποίησης ενός τέτοιου δικτύου είναι ζωτικής σημασίας αφού μια κινητή συσκευή μπορεί να αποκτή πρόσβαση και να εκφορτώνει εργασίες μεγάλων δεδομένων σε ισχυρά υπολογιστικά κέντρα που είναι προσαρτημένα σε πολλά ασύρματα σημεία πρόσβασης, εξασφαλίζοντας έτσι ότι πληρούνται οι απαιτήσεις υψηλής ποιότητας εξυπηρέτησης των τελικών χρηστών. Ωστόσο, η συνδεσιμότητα κινητών συσκευών δεν είναι συνεχής, αλλά μάλλον σποραδική, με διαφορετικές δυνάμεις σήματος. Επιπλέον, η ετερογένεια των πόρων ενός υπολογιστικού κέντρου του υπολογιστικού νέφους και τα μεγάλα αιτήματα δεδομένων μιας εφαρμογής θέτουν πρόσθετες προκλήσεις για τη λήψη βέλτιστης απόφασης εκτέλεσης αυτής.

Σκοπός της αυτής της διπλωματικής εργασίας είναι η ανάπτυξη μίας αρχιτεκτονικής κατανομής πόρων του υπολογιστικού νέφους που λαμβάνει υπόψιν του την κινητικότητα των χρηστών, για εκτεταμένη εκτέλεση μεγάλων εφαρμογών. Επιπρόσθετα η αρχιτεκτονική αυτή προσομοιώνεται σε ένα αληθινό σύστημα στο οποίο υπάρχουν πολλοί κινητοί χρήστες (συνδεδεμένα στο τοπικό υπολογιστικό νέφος ή μη), το τοπικό υπολογιστικό νέφος που απαρτίζεται από ομάδες υπολογιστών με υψηλές προδιαγραφές και ένα απομακρυσμένο και απομονωμένο κέντρο εξυπηρέτησης (μη τοπικό υπολογιστικό νέφος). Η αρχιτεκτονική του συστήματος, η απόκριση του σε ένα αληθινό σύστημα παρουσιάζονται και αξιολογούνται καθώς επίσης μελετάται και η ενέργεια κατανάλωσης των χρηστών (υπολογιστικό και επικοινωνιακό κόστος), ο οποίος είναι κύριος στόχος ελαχιστοποίησης.

Τέλος μαζί με τα αποτελέσματα του πειράματος παρατίθενται κάποιες πιθανές βελτιστοποιήσεις καθώς επίσης στο κομμάτι της βιβλιογραφίας γίνονται και αναφορές άλλων αρχιτεκτονικών δρομολόγησης εργασιών για κινητές εφαρμογές μεγάλων δεδομένων.

Λέξεις Κλειδιά

Υπολογιστικό Κινητό Νέφος, Τοπικό Υπολογιστικό Νέφος, Απομακρυσμένο Υπολογιστικό Νέφος, Κινητές Συσκευές, Ενεργειακή Κατανάλωση, Χρόνος Εκτέλεσης, Βαθμός Συσχέτισης, Τυπική Απόκλιση, Αρχιτεκτονική, Κινητικότητα.

Abstract

With the advances in wireless communications technology, more and more people are dependent on mobile devices for business, entertainment and social interactions. Together with advances, technology is rapidly developing and more applications design that are demanding more and more computing resources and data. In order to maintain the high quality of service for users, the remote execution of tasks requiring high computing power in computer groups known as cloud computing, has become inevitable. However, the idea of dealing with cloud computing as isolated and remote data centers is now outdated, as there are clear benefits when they are locally accessible and also connected to one another to form a network (*e.g smart cities*).

The need to optimize such a network is vital since a mobile device can access and unload large data tasks on powerful servers that are attached to multiple wireless access points, thereby ensuring that end-user service quality requirements are met. However, the connectivity of mobile devices is not continuous but rather sporadic, with different signal strengths. In addition, the heterogeneity of a cloud computing center resources and the large data requests of an application pose additional challenges for making the best decision to execute it..

The aim of this diploma thesis is to develop a cloud computing resource allocation architecture that takes into account the mobility of users for extended large applications. Additionally, this architecture is simulated in a real system in which there are many mobile users *connected to the local cloud computing or not*, the local cloud computing comprised of high-end computer groups and a remote and isolated service center (*non-local cloud*). The architecture of the system, its response to a real system is presented and evaluated, as well as the energy consumption of the users (*cost of processing, battery, communication*), which is the main objective of minimizing.

Finally, along with the results of the experiment, some possible optimizations are presented, as well as references to other architectural routing tasks for large data mobile applications.

Keywords

Mobile Cloud Computing, Cloud, Cloudlet, Mobile Devices, Energy Consumption, Execution Time, Associativity, Standard Deviation, Architecture, Mobility.

Ευχαριστίες

Με την παρούσα διπλωματική εργασία ολοκληρώνονται οι σπουδές μου στην Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου.

Αρχικά θα ήθελα να ευχαριστήσω τον καθηγητή κύριο Συμεών Παπαβασιλείου για την επίβλεψη της διπλωματικής μου εργασίας καθώς επίσης ένα μεγάλο ευχαριστώ στους μεταδιδακτορικούς ερευνητές του εργαστηρίου για τη καθοδήγηση τους και για την υλοποίηση της παρούσας διπλωματικής εργασίας.

Παράλληλα θέλω να ευχαριστήσω θερμά τους φίλους μου και συνάδελφους μου με τους οποίους ξεκινήσαμε μαζί και επισφραγίζουμε το τέλος των σπουδών μας πάλι μαζί. Ένα ιδιαίτερο ευχαριστώ οφείλω στα άτομα του εργαστηρίου για την παρέα τους, την βοήθεια τους και την φαντασία που επιδείκνυαν στα διαλείμματα αυτής της διπλωματικής.

Τέλος οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου, που παρά τις όποιες δυσκολίες, με στήριξε και βοήθησε στο να φτάσω εδώ που είμαι σήμερα.

Νικόλαος Α. Λατζώνης

Περιεχόμενα

Περίληψη	5
Abstract	7
Ευχαριστίες	9
Περιεχόμενα	12
Κατάλογος Σχημάτων	14
Κατάλογος Πινάκων	15
1 Εισαγωγή	17
1.1 Υπολογιστικό Κινητό Νέφος	18
1.2 Τοπικό Υπολογιστικό Νέφος	18
1.3 Πρόβλημα Επιλογής Βέλτιστου Τοπικού Υπολογιστικού Νέφους . .	20
1.3.1 Στόχοι Βελτιστοποίησης Προβλήματος	20
1.4 Αντικείμενο της Διπλωματικής	20
1.4.1 Συνεισφορά	21
1.5 Οργάνωση Του Τόμου	21
2 Βιβλιογραφία	22
2.1 Βασικοί Στόχοι Αρχιτεκτονικών Δομών Υπολογιστικού Κινητού Νέφους	22
2.2 Το Πρόβλημα Της Απομακρυσμένης Εκτέλεσης Κώδικα Εφαρμογής	24
2.2.1 Απομακρυσμένη Εκτέλεση Κώδικα Εφαρμογής στον Master Cloud	24
2.2.2 Απομακρυσμένη Εκτέλεση Κώδικα Εφαρμογής στα Cloudlets	27
3 Σχεδιασμός Συστήματος Και Ενεργειακά Μοντέλα	30
3.1 Αρχιτεκτονική Δομή Υπολογιστικού Κινητού Νέφους	30
3.1.1 Ορισμός Υπολογιστικού Κινητού Νέφους	30
3.1.2 Οργάνωση Δομής Υπολογιστικού Κινητού Νέφους	31
3.1.3 Εφαρμογές Υπολογιστικού Κινητού Νέφους	31
3.1.4 Πλεονεκτήματα Και Οφέλη Υπολογιστικού Κινητού Νέφους .	32
3.2 Σχεδίαση Αρχιτεκτονικής Συστήματος(Mobi-Het)	33

3.3	Βασικές Έννοιες Για Την Αναπαράσταση Δομικών Στοιχείων Της Αρχιτεκτονικής Mobi-Het	35
3.4	Ενεργειακά Μοντέλα Κινητής Συσκευής	37
3.4.1	Ενεργειακή Κατανάλωση Υπολογιστικής Μονάδας Επεξεργασίας(CPU)	37
3.4.2	Ενεργειακή Κατανάλωση Ασύρματου Δικτύου WiFi	38
3.4.3	Ενεργειακή Κατανάλωση Δικτύου 3G	41
4	Περιγραφή Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het)	42
4.1	Παρουσίαση Προβλήματος Δυναμικής Δρομολόγησης	42
4.2	Επίλυση Προβλήματος επιλεκτικής κατανομής πόρων σε εφαρμογές .	43
4.2.1	Προσδιορισμός Βαθμού Συσχέτισης	44
4.2.2	Υπολογισμός Τυπικής Απόκλισης Συστήματος	45
4.2.3	Υπολογισμός Χρόνου Εκτέλεσης Εφαρμογής	46
4.3	Γενική Σύνοψη Αλγορίθμου	47
5	Πειραματική Αξιολόγηση Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het)	48
5.1	Λεπτομέρειες Υλοποίησης	48
5.2	Πειραματική Αξιολόγηση	51
5.2.1	Πείραμα Α:Επισκόπηση Αποτελεσμάτων Κινητής Συσκευής και Λειτουργία Μοντέλου Δυναμικής Δρομολόγησης (Mobi-Het)	52
5.2.2	Πείραμα Β:Σύγκριση Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het) με Μοντέλο Τυχαίας Δρομολόγησης Με Μεταβλητή Παράμετρο Την Ακτίνα Τοπικού Υπολογιστικού Νέφους	56
5.2.3	Πείραμα Γ:Σύγκριση Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het) με Μοντέλο Τυχαίας Δρομολόγησης Με Μεταβλητή Παράμετρο Τον Αριθμό Των Κινητών Συσκευών . . .	60
5.2.4	Πείραμα Δ:Σύγκριση Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het) με Μοντέλο Τυχαίας Δρομολόγησης Με Μεταβλητή Παράμετρο Τον Βαθμό Κινητικότητας Των Κινητών Συσκευών	64
6	Συμπεράσματα και Μελλοντική Εξέλιξη Εργασίας	69
6.1	Σύνοψη συμπερασμάτων αξιολόγησης	69
6.2	Μελλοντικές Επεκτάσεις	70
	Βιβλιογραφία	73

Κατάλογος Σχημάτων

1.1	Περιβάλλον εκτέλεσης εφαρμογών κινητών συσκευών σε "έξυπνες" πόλεις (smart cities).	18
3.1	Αρχιτεκτονική Δομή Υπολογιστικού Κινητού Νέφους	31
3.2	Αρχιτεκτονική σχεδίασης Mobi-Het	35
3.3	Διάγραμμα Κλάσεων Του Υπολογιστικού Νέφους.	37
3.4	Ενεργειακό μοντέλο σε συνάρτηση του RSSI και της παραμέτρου β_1	40
3.5	Κατανάλωση ενέργειας σε συνάρτηση με το χρόνο	41
4.1	Προσδιορισμός του βαθμού συσχετισμού.	45
5.1	Ενεργειακές Καταναλώσεις Κινητής Συσκευής Με Αλγόριθμο Mobi-Het	53
5.2	Κάτοψη Κίνησης Κινητής Συσκευής Εντός Πεδίου (grid)	53
5.3	Συνολική Μέση Απόκριση Εργασιών Με Αλγόριθμο Δυναμικής Δρομολόγησης Mobi-Het	54
5.4	Μέσος Χρόνος Απόκρισης Εφαρμογής, Αλγορίθμου Mobi-Het, Μοντέλου Πρόβλεψης Mobi-Het Και Μοντέλου Πρόβλεψης Τυχαίας Κατανομής Εργασιών.	55
5.5	Μέση Τιμή Associativity, Μοντέλου Mobi-Het Και Τυχαίας Κατανομής Εργασιών.	55
5.6	Μέση Τυπική Απόκλιση Ανά Χρονικό Διάστημα Εκφόρτωσης Εφαρμογής Σε Cloudlet, Μοντέλου Mobi-Het Και Τυχαίας Κατανομής Εργασιών.	55
5.7	Συνολικός Χρόνος Απόκρισης Σε Συνάρτηση Του Radius Scale	57
5.8	Μέση Τυπική Απόκλιση Συνολικού Φορτίου Των Cloudlets Ανά Χρονικό Διάστημα Δημιουργίας Μιας Εφαρμογής Σε Συνάρτηση Του Radius Scale.	58
5.9	Συνολικά Απορριφθέντα Αιτήματα Εφαρμογών Σε Συνάρτηση Του Radius Scale.	59
5.10	Συνολική Κατανάλωση Ενέργειας Κινητών Συσκευών	59
5.11	Συνολικά Ληφθέντα Αιτήματα Εφαρμογών Από Τον Master Cloud Σε Συνάρτηση Του Radius Scale.	60
5.12	Συνολικός Μέσος Χρόνος Απόκρισης Ανά Εφαρμογή Σε Συνάρτηση Του Αριθμού των Κινητών Συσκευών.	61

5.13 Μέση Τυπική Απόκλιση Συνολικού Φορτίου Των Cloudlets Ανά Χρονικό Διάστημα Δημιουργίας Μιας Εφαρμογής Σε Συνάρτηση Του Αριθμού των Κινητών Συσκευών.	62
5.14 Συνολικά Απορριφθέντα Αιτήματα Εφαρμογών Σε Συνάρτηση Του Αριθμού των Κινητών Συσκευών	63
5.15 Συνολική Κατανάλωση Ενέργειας Κινητών Συσκευών	63
5.16 Συνολικά Ληφθέντα Αιτήματα Εφαρμογών Από Τον Master Cloud .	64
5.17 Συνολικός Μέσος Χρόνος Απόκρισης Ανά Εφαρμογή Σε Συνάρτηση Της Μέσης Ταχύτητας Κινητικότητας Των Κινητών Συσκευών. . . .	65
5.18 Μέση Τυπική Απόκλιση Συνολικού Φορτίου Των Cloudlets Ανά Χρονικό Διάστημα Δημιουργίας Μιας Εφαρμογής Σε Συνάρτηση Της Μέσης Κινητικότητας Των Κινητών Συσκευών	66
5.19 Συνολικά Απορριφθέντα Αιτήματα Εφαρμογών Σε Συνάρτηση Της Μέσης Ταχύτητας Κινητικότητας Των Κινητών Συσκευών.	67
5.20 Συνολική Κατανάλωση Ενέργειας Κινητών Συσκευών Σε Συνάρτηση Της Μέσης Ταχύτητας Κινητικότητας Των Κινητών Συσκευών. . . .	67
5.21 Συνολικά Ληφθέντα Αιτήματα Εφαρμογών Από Τον Master Cloud Σε Συνάρτηση Της Μέσης Ταχύτητας Κινητικότητας Των Κινητών Συσκευών.	68

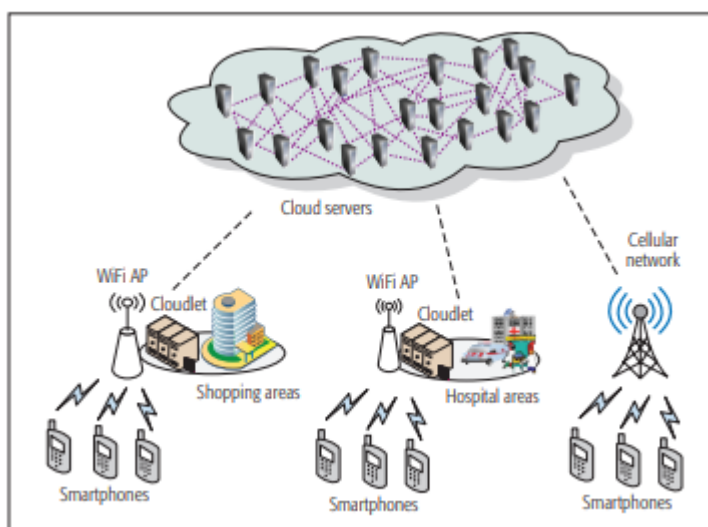
Κατάλογος Πινάκων

3.1	Πίνακας Κατανομής Τιμών β_i^{freq} , β_i^{idle} με βάση τη Συχνότητα (freq) λειτουργίας του επεξεργαστή	38
3.2	RSSI,Ενεργειακό μοντέλο,SSE,R-Square,RMSE	40
4.1	Αντίστοιχη RSSI με τον ρυθμό μετάδοσης	47

Κεφάλαιο 1

Εισαγωγή

Σήμερα, οι "έξυπνες" πόλεις (smart cities) αναδεικνύονται ως προτεραιότητα της τεχνολογικής καινοτομίας για την έρευνα και την ανάπτυξη σε ολόκληρο τον κόσμο. Γενικά, μια "έξυπνη" πόλη είναι μια αποτελεσματική ενσωμάτωση των φυσικών, ψηφιακών και ανθρώπινων συστημάτων για να προσφέρει ένα βιώσιμο και ευνοούμενο μέλλον στους πολίτες της [41]. Ωστόσο, όλο και περισσότεροι άνθρωποι εξαρτώνται από ψηφιακές κινητές συσκευές οι οποίες συνήθως υστερούν σε επεξεργαστική ισχύ και μπαταρία με αποτέλεσμα η απομακρυσμένη εκτέλεση εργασιών/εφαρμογών να είναι αναπόφευκτη. Αυτό οδηγεί σε πολλούς κάθε φορά συνδεδεμένους χρήστες οι οποίοι πρέπει να διαχειριστούν με σωστό και αποδοτικό τρόπο από το υπολογιστικό νέφος. Επομένως, σημαντικό ζήτημα για υπηρεσίες είναι η πρόσβαση σε πραγματικό χρόνο, η ανταλλαγή, η αποθήκευση, η επεξεργασία και η ανάλυση τέτοιων μεγάλων δεδομένων οπουδήποτε και ανά πάσα στιγμή χωρίς προβλήματα [39]. Σε αυτό το σενάριο, εισέρχεται η έννοια του Mobile Cloud Computing (MCC) [11] που μπορεί να διαδραματίσει σημαντικό ρόλο. Οι εφαρμογές MCC όπως οι μετρήσεις οδικής κυκλοφορίας, η παρακολούθηση της ατμοσφαιρικής ρύπανσης, τα συστήματα υγειονομικής περίθαλψης και ασφάλειας, η έξυπνη εκπαίδευση και πολλά άλλα είναι κατάλληλα παραδείγματα εισόδου για απομακρυσμένη εκτέλεση σε smart cities. Επιτρέπει στις κινητές συσκευές να εκτελούν μεγάλες εργασίες δεδομένων με τη βοήθεια ισχυρών απομακρυσμένων υπολογιστικών κέντρων που λέγονται σύννεφα (clouds). Ωστόσο, η εκτέλεση μεγάλου κώδικα εφαρμογής δεδομένων σε ένα απομακρυσμένο cloud οδηγεί συχνά σε κακή απόδοση λόγω της υψηλής καθυστέρησης και του χαμηλού εύρους ζώνης, καθώς ο πραγματικός πάροχος βρίσκεται συχνά μακριά από τον πελάτη. Ως εκ τούτου, μια πρόσφατη τάση είναι η χρήση του cloudlet, ενός μικρότερου υπολογιστικού κέντρου στο οποίο ο χρήστης συνδέεται σε αυτό μέσω ενός σημείου πρόσβασης τοπικού δικτύου, για σκοπούς εκτέλεσης απομακρυσμένου κώδικα μεγάλης ταχύτητας μεγάλων εφαρμογών δεδομένων [15, 22], ιδίως σε περιοχές όπως νοσοκομεία, εκπαιδευτικά ιδρύματα, εμπορικές περιοχές, τερματικά αεροδρομίων, λεωφορεία, σιδηροδρομικοί σταθμοί κ.λ.π. Τα cloudlet είναι πυκνά αναπτυγμένα μέσα σε ένα περιβάλλον MCC, όπως φαίνεται στο Σχήμα 1.1. Συνήθως, ένα cloudlet έχει πολύ λιγότερη υπολογιστική ικανότητα σε σύγκριση με ένα κύριο απομακρυσμένο σύννεφο και οι μεγάλες εφαρμογές δεδομένων μεγάλου βάρους που δεν μπορούν να εκτελεστούν αποστέλλονται στο κύριο σύννεφο [6].



Σχήμα 1.1: Περιβάλλον εκτέλεσης εφαρμογών κινητών συσκευών σε "έξυπνες" πόλεις (smart cities).

1.1 Υπολογιστικό Κινητό Νέφος

Οι πλατφόρμες Mobile Cloud Computing (MCC) [11] αποσκοπούν στην υπέρβαση των περιορισμών πόρων των κινητών συσκευών με τη χρήση πόρων που διατίθενται σε κατακεντρωμένα περιβάλλοντα που λέγονται σύννεφα (clouds). Ο στόχος είναι να εκφορτωθούν οι υπολογιστικές εργασίες και οι εργασίες μεγάλων δεδομένων από κινητές συσκευές φτωχές σε πόρους σε κόμβους σύννεφων. Ωστόσο η κινητικότητα των χρηστών εισάγει νέες δυσκολίες στην εξασφάλιση ποιότητας υπηρεσίας των εφαρμογών. Τόσο ο αριθμός όσο και η συνολική κινητικότητα των χρηστών αυξάνονται με αποτέλεσμα μειωμένη αξιοπιστία. Καθώς ένας χρήστης κινείται στο χώρο, η φυσική απόσταση του με το cloud δημιουργεί πρόσθετες καθυστερήσεις. Ομοίως, οι μικρές ζώνες εμβέλειας σε δίκτυα WiFi, η γρήγορη κίνηση του χρήστη μπορεί να προκαλέσει αύξηση των ανεπιτυχών αιτημάτων εφαρμογών για απομακρυσμένη εκτέλεση. Με άλλα λόγια, η κινητικότητα των χρηστών, αν δεν αντιμετωπιστεί σωστά, μπορεί να οδηγήσει σε μη αποτελεσματική αξιοποίηση των πόρων και τελικά σε μειωμένη ποιότητα υπηρεσίας της εφαρμογής και αύξηση της συνολικής καθυστέρησης του χρήστη.

1.2 Τοπικό Υπολογιστικό Νέφος

Ο όρος τοπικό υπολογιστικό νέφος ή πιο απλά cloudlet αναφέρεται σε ένα στρώμα που συνδέει τις κινητές συσκευές με τα υπολογιστικά κέντρα στο MCC [26]. Η έννοια του cloudlet απευθύνεται σε μία αξιόπιστη υποδομή που μπορεί να είναι ένας υπολογιστής ή ένα σύμπλεγμα ψηφιακής υποδομής με λογισμικό που παρέχει στους χρήστες cloud μια γρήγορη απάντηση στα αιτήματα εφαρμογών τους. Οι χρήστες

του cloud για κινητά στέλνουν τις αιτήσεις για υπηρεσίες μέσω του εικονικού μηχανήματος (VM) που είναι προσαρτημένο στις εφαρμογές πελάτη και οι αιτήσεις θα προσεγγίσουν αποτελεσματικά το κοντινότερο cloudlet. Η μέθοδος πρόσβασης γίνεται μέσω των σημείων πρόσβασης στο ασύρματο δίκτυο (WiFi). Πριν από την υποβολή των αιτημάτων εξυπηρέτησης στο cloudlet, οι περισσότεροι υπολογισμοί σχετικά με τις κατανομές πόρων πραγματοποιούνται μέσω δυναμικό προγραμματισμό (dynamic programming) και για αυτό το λόγο τα cloudlets λέγονται και αλλιώς dynamic cloudlet (DCL). Συνήθως, τα DCLs έχουν σχεδιαστεί για να ολοκληρώσουν τουλάχιστον τρεις αποστολές, οι οποίες περιλαμβάνουν τα εξής:

1. Χρησιμοποιώντας δυναμικό προγραμματισμό για τον προσδιορισμό και την πρόβλεψη φυσικών μηχανών για τρέχουσα και μελλοντική χρήση.
2. Υπολογισμοί απλών εφαρμογών και γρήγορη ανταπόκριση στους τελικούς χρήστες.
3. Πρόβλεψη για το εάν οι χρήστες θα πρέπει να μεταβούν σε άλλα cloudlets ή στο Master Cloud σε πραγματικό χρόνο.

Τα Cloudlets εφαρμόζουν δυναμικό προγραμματισμό για να προσδιορίσουν ποιοι νέοι διακομιστές πρέπει να συνδεθούν με απώτερο σκοπό την εξοικονόμηση ενέργειας του χρήστη [12]. Επιπλέον, σύμφωνα με τις διάφορες απαιτήσεις υπηρεσιών, υπάρχουν δύο τύποι ανάπτυξης dynamic cloudlet(DCL) που καθορίζονται ως εξής:

- Private dynamic cloudlets: Αυτή η ανάπτυξη υποστηρίζει έναν ενιαίο σύνολο υπηρεσιών του cloud με σκοπό την αύξηση της απόδοσης των υπηρεσιών και τη μείωση της κατανάλωσης ενέργειας. Για παράδειγμα, ιδιωτικά δυναμικά νέφη μπορούν να χρησιμοποιηθούν για υπηρεσίες παγκόσμιου συστήματος θέσης (GPS) που παρέχονται από έναν πάροχο υπηρεσιών.
- Public dynamic cloudlets: Δημόσια πρόσβαση σε υπηρεσίες Mobile Cloud Computing μέσω πολλών dynamic cloudlets, οι οποίες ενδέχεται να υποστηρίζουν πολλούς παρόχους υπηρεσιών. Δίνουμε ένα παράδειγμα ότι μια δυναμική υπηρεσία αναζήτησης μπορεί να παραδοθεί με την ανάπτυξη δημόσιων DCLs, δεδομένου ότι οι υπηρεσίες ίσως υποστηρίζονται από πολλούς παρόχους υπηρεσιών που ποικίλουν στα δεδομένα που προσφέρουν.

Στα πλαίσια της παρούσας διπλωματικής εργασίας τον δυναμικό προγραμματισμό μέσω του αλγορίθμου δρομολόγησης που θα παρουσιάσουμε, τον υλοποιεί το απομακρυσμένο σύννεφο (Master Cloud) για όλα τα cloudlets, το οποίο χαρακτηρίζεται από υψηλές προδιαγραφές (υπολογιστικούς και αποθηκευτικούς πόρους).

1.3 Πρόβλημα Επιλογής Βέλτιστου Τοπικού Υπολογιστικού Νέφους

Σε ένα τέτοιο περιβάλλον, λόγω της κινητικότητας των κινητών συσκευών και των ετερογένειων που υπάρχουν τόσο στους πόρους των cloudlet όσο και στα αιτήματα εφαρμογών μεγάλων δεδομένων, η κατανομή των πόρων για αιτήσεις εκφόρτωσης εφαρμογών σε έναν από τους πολλά διαθέσιμα cloudlets βρίσκεται σε μεγάλη προσπάθεια. Επιπλέον, η ισχύς σήματος WiFi από διαφορετικές θέσεις του κινητού ποικίλλει σημαντικά με την πάροδο του χρόνου, γεγονός που προκαλεί μεγάλη πίεση σε εφαρμογές που απαιτούν μεγάλες ποσότητες δεδομένων για εκφόρτωση.

1.3.1 Στόχοι Βελτιστοποίησης Προβλήματος

Όπως έχει ήδη προαναφερθεί κάθε τοπικό υπολογιστικό νέφος μπορεί να διαφέρει το ένα με το άλλο με τα βασικά μελετούμενα χαρακτηριστικά να είναι τα εξής:

- υπολογιστικοί πόροι του συστήματος,
- το εύρος εμβέλειας και η ισχύ του σήματος δικτύου WiFi ,
- το υπάρχον φόρτο εργασίας

Στην ουσία πρόκειται για τρεις ασυσχέτιστοι παράγοντες που στόχος είναι μέσω του αλγορίθμου που θα παρουσιάσουμε να ληφθούν υπόψη στην επιλογή σύζευξης κινητού με cloudlet για απομακρυσμένη εκτέλεση εφαρμογών ώστε να έχουμε τη μέγιστη ποιότητα υπηρεσίας από πλευράς του χρήστη σε όσο γίνεται μικρότερο χρονικό διάστημα απόκρισης. Η επιλογή πρέπει να γίνεται δυναμικά και με όσον το λιγότερο δυνατή εφικτή καθυστέρηση.

1.4 Αντικείμενο της Διπλωματικής

Σε αυτό το άρθρο, προτείνουμε ένα βέλτιστο μοντέλο κατανομής των πόρων υπολογιστικού νέφους για απομακρυσμένη εκτέλεση εργασιών μεγάλων δεδομένων, που ονομάζεται Mobi-Het [10], το οποίο επιλέγει το βέλτιστο τοπικό υπολογιστικό νέφος (δηλαδή τα μικρά νέφη) σε συνάρτηση με τη κινητικότητα των χρηστών, τις απαιτήσεις υπηρεσίας της εφαρμογής, την ποιότητα του σήματος σύνδεσης ασύρματου δικτύου WiFi και τη κατανομή του τρέχοντος φόρτου εργασίας στα αντίστοιχα cloudlets. Αυτή η επιλογή εκτελείται από τον με τη βοήθεια ορισμένων πληροφοριών από κινητές συσκευές. Διατυπώνουμε το πρόβλημα της κατανομής των καταλληλότερων πόρων για την εκτέλεση αιτημάτων εφαρμογών μεγάλων δεδομένων από κινητές συσκευές ανά πάσα στιγμή ως ένα πρόβλημα πολλαπλών στόχων μεικτού ακέραιου γραμμικού προγραμματισμού (MILP). Θεωρούμε την ελεύθερη κίνηση των κινητών

συσκευών με μεταβλητή ταχύτητα και κατεύθυνση με βάση ένα τυχαίο μοντέλο απόφασης. Αντιμετωπίζουμε την ετερογένεια των cloudlets και τις απαιτήσεις κάθε εφαρμογής για μια ισορροπημένη κατανομή του συνολικού φόρτου εργασίας προκειμένου να προβλέψουμε τον χρόνο απομακρυσμένης εκτέλεσης που θα μας βοηθήσει στη συνέχεια να εφαρμόσουμε τον αλγόριθμο δυναμικής δρομολόγησης που θα παρουσιάσουμε. Στο υπόλοιπο άρθρο, περιγράφουμε πρώτα την προτεινόμενη αρχιτεκτονική Mobi-Het και στη συνέχεια ακολουθούμε με περιγραφές των βασικών συστατικών της. Στη συνέχεια, το σύστημα Mobi-Het αξιολογείται μέσω της υλοποίησης δοκιμών μέσα σε ένα πραγματικό σύστημα προσομοίωσης και τα πειραματικά αποτελέσματα δείχνουν ότι το σύστημα Mobi-Het ξεπερνάει κατά πολύ από τον αλγόριθμο της τυχαίας επιλογής με τον οποίο και τον συγκρίνουμε. Επίσης παρατίθενται ενεργειακά μοντέλα, τρόποι υπολογισμού ενέργειας υπολογιστικής μονάδας (CPU) [42], ασύρματου δικτύου WiFi [33] και δικτύου 3G [14] που μας δείχνουν αν μη τι άλλο σαν μία επιπρόσθετη πληροφορία για το εάν μία εφαρμογή εκτελέστηκε είτε τοπικά στη συσκευή είτε σε κάποιο cloudlet είτε στον Master Cloud.

1.4.1 Συνεισφορά

Τα πειραματικά αποτελέσματα δείχνουν ότι ο αλγόριθμος είναι μία γρήγορη και κλιμακωτή λύση στο πρόβλημα ελαχιστοποίησης της απόκρισης εφαρμογών. Όμως σε συνθήκες όπου ο αριθμός των χρηστών είναι αρκετά μεγάλος (μεγαλύτερο από το σημείο κορεσμού του συστήματος) ή η μέση κινητικότητα των χρηστών είναι υψηλή ή η αλληλοκάλυψη των πεδίων WiFi των cloudlets είναι μικρή, ο αλγόριθμος δεν ανταποκρίνεται το ίδιο ικανοποιητικά όπως φαίνεται στα αποτελέσματα. Η πολυπλοκότητα του αλγορίθμου εξαρτάται κατά κύριο λόγο από τον αριθμό των χρηστών μέσα στο σύστημα, τον αριθμό των cloudlets αλλά από την επιφάνεια αλληλοκάλυψης μεταξύ αυτών. Επομένως με κατάλληλες τιμές εισόδου ο αλγόριθμος ανταποκρίνεται σε πολύ ικανοποιητικό βαθμό και χαμηλή σε κόστος υπολογιστική πολυπλοκότητα.

1.5 Οργάνωση Του Τόμου

Η παρούσα διπλωματική εργασία είναι οργανωμένη σε έξι κεφάλαια. Οι εργασίες σχετικές με το αντικείμενο της διπλωματικής και παρόμοιες προσεγγίσεις παρουσιάζονται στο Κεφάλαιο 2. Στο Κεφάλαιο 3 παρουσιάζεται το θεωρητικό υπόβαθρο πάνω στο Mobile Cloud Computing, καθώς και οι βασικές έννοιες και τεχνικές που χρησιμοποιήσαν στη διπλωματική εργασία. Στο Κεφάλαιο 4 παρουσιάζεται εκτενώς ο αλγόριθμος δυναμικής δρομολόγησης Εργασιών (Mobi-Het). Στο Κεφάλαιο 5 παρουσιάζουμε τα πειραματικά αποτελέσματα των προσομοιώσεων, ανάλυση αυτών και συμπεράσματα. Το Κεφάλαιο 6 παρέχει ιδέες και κατευθύνσεις για την προοπτική μελλοντικής εξέλιξης της παρούσας διπλωματικής εργασίας.

Κεφάλαιο 2

Βιβλιογραφία

2.1 Βασικοί Στόχοι Αρχιτεκτονικών Δομών Υπολογιστικού Κινητού Νέφους

Κάθε σχεδιαζόμενη αρχιτεκτονική δρομολόγησης εργασίας κινητής συσκευής για απομακρυσμένη εκτέλεση σε cloud ή cloudlet οφείλει να διασφαλίσει την υψηλή ποιότητα εξυπηρέτησης του χρήστη, η οποία αποτελεί μια πολύ σημαντική μέτρηση που πρέπει να ληφθεί υπόψιν και χωρίζεται σε τέσσερις κατηγορίες σύμφωνα με το άρθρο [40]:

1. **Συνδεσιμότητα:**

Σε ένα ετερογενές δίκτυο ομίχλης (Fog Computing), η αναμετάδοση, η διαμερίση και η ομαδοποίηση αιτημάτων εφαρμογών για απομακρυσμένη εκτέλεση, παρέχουν νέες ευκαιρίες για μείωση του κόστους, μείωση των δεδομένων επεξεργασίας και επέκταση της συνδεσιμότητας. Για παράδειγμα, ένα ασύρματο δίκτυο αισθητήρων μπορεί να χωριστεί σε πολλά συμπλέγματα κόμβων ομίχλης, πλούσιους σε υπολογιστικούς και αποθηκευτικούς πόρους. Το άρθρο [38] προτείνει μια στρατηγική σύνδεσης που όχι μόνο επιτυγχάνει μια ελάχιστη απόδοση, αλλά στοχεύει και στην αποτελεσματικότητα των υπολογιστικών γενικών εξόδων. Ομοίως, η επιλογή του κόμβου ομίχλης από τον τελικό χρήστη θα επηρεάσει σε μεγάλο βαθμό την απόδοση. Μπορούμε δυναμικά να επιλέξουμε ένα υποσύνολο κόμβων ομίχλης ως κόμβους αναμετάδοσης για σκοπούς βελτιστοποίησης της μέγιστης διαθεσιμότητας υπηρεσιών ομίχλης για μια συγκεκριμένη περιοχή ή για έναν μόνο χρήστη, με περιορισμούς όπως η καθυστέρηση, η απόδοση, η συνδεσιμότητα και η κατανάλωση ενέργειας.

2. **Αξιοπιστία:**

Η αξιοπιστία μπορεί να βελτιωθεί μέσω περιοδικών επιθεωρήσεων προς επανάληψη μετά από αποτυχία, αναδιάταξη αποτυχημένων εργασιών ή αναπαραγωγή για εκμετάλλευση παράλληλης εκτέλεσης. Στο άρθρο [23] εξετάζεται η απαίτηση αξιοπιστίας της συστοιχίας υπολογιστικών στοιχείων (clustering computing),

του δικτύου, του cloud και του δικτύου αισθητήρων. Ωστόσο, ο έλεγχος και η αναδιάταξη ενδέχεται να μην ταιριάζουν με το ιδιαίτερα δυναμικό περιβάλλον υπολογιστικής ομίχλης, καθώς θα υπάρχουν λανθάνουσες καταστάσεις οι οποίες δεν μπορούν να προσαρμοστούν. Η αναπαραγωγή εργασιών για παράλληλη εκτέλεση φαίνεται πιο ελπιδοφόρα, αλλά βασίζεται σε πολλαπλούς κόμβους ομίχλης για να συνεργαστούν.

3. **Χωρητικότητα:**

Η χωρητικότητα έχει δύο πτυχές, το εύρος ζώνης δικτύου και τη χωρητικότητα αποθήκευσης. Προκειμένου να επιτευχθεί υψηλό εύρος ζώνης και αποδοτική χρήση της αποθήκευσης, είναι σημαντικό να διερευνηθεί ο τρόπος με τον οποίο τα δεδομένα τοποθετούνται στο δίκτυο ομίχλης δεδομένου ότι η τοποθεσία του χρήστη για τον υπολογισμό είναι πολύ σημαντική. Υπάρχουν παρόμοια έργα στο πλαίσιο του cloud [1] και του δικτύου αισθητήρων [31]. Ωστόσο, το πρόβλημα αυτό αντιμετωπίζει νέες προκλήσεις στον τομέα των υπολογιστών ομίχλης. Για παράδειγμα, ένας κόμβος ομίχλης μπορεί να χρειαστεί να υπολογίσει δεδομένα που διανέμονται σε διάφορους κοντινούς κόμβους. Ο υπολογισμός δεν μπορεί να ξεκινήσει πριν από τη λήξη της συνάθροισης δεδομένων, η οποία καθυστερεί τις προσφερόμενες υπηρεσίες. Για να επιλύσουμε αυτό το πρόβλημα, μπορούμε να εκμεταλλευτούμε το μοτίβο κινητικότητας των χρηστών και το μοτίβο των αιτημάτων εξυπηρέτησης για την τοποθέτηση δεδομένων σε κατάλληλους κόμβους ομίχλης για να ελαχιστοποιήσουμε το κόστος λειτουργίας, την καθυστέρηση ή να μεγιστοποιήσουμε τη διακίνηση. Η τοποθέτηση δεδομένων στην ομοσπονδία της ομίχλης και του νέφους χρειάζεται επίσης κριτική σκέψη. Οι προκλήσεις προέρχονται από το πώς να σχεδιάσουμε την αλληλεπίδραση μεταξύ και των clouds και cloudlets της ομίχλης της για να ικανοποιήσουμε τη διαφορετική κατανομή του συνολικού φόρτου εργασίας. Λόγω της δυναμικής τοποθέτησης δεδομένων και του μεγάλου συνολικού όγκου χωρητικότητας στον υπολογισμό ομίχλης, ίσως χρειαστεί επίσης να επανασχεδιάσουμε τη μηχανή αναζήτησης, η οποία μπορεί να επεξεργαστεί το ιστορικό δεδομένων στους κόμβους ομίχλης [35, 34]. Είναι επίσης πολύ ενδιαφέρον να επανασχεδιαστεί η κρυφή μνήμη στον κόμβο ομίχλης για να εκμεταλλευτεί την προσωρινή τοποθεσία και την ευρύτερη κάλυψη για να αποθηκεύσει το εύρος ζώνης δικτύου και να μειώσει την καθυστέρηση, ενώ υπάρχει η υπάρχουσα εργασία της κρυφής μνήμης στην τελική συσκευή [44] και στη κρυφή μνήμη των κόμβων της ομίχλης [36].

4. **Καθυστέρηση:**

Οι εφαρμογές που είναι ευαίσθητες στην καθυστέρηση, όπως η εξόρυξη δεδομένων ή η σύνθετη επεξεργασία συμβάντων, είναι τυπικές εφαρμογές που χρειάζονται υπολογιστές ομίχλης για την παροχή ροής σε πραγματικό χρόνο επεξεργασίας. Το άρθρο [13] προτείνει ένα σύστημα επεξεργασίας ευκαιριακού χωρικού και χρονικού γεγονότος της ομίχλης για να ικανοποιήσει την απαίτηση λανθάνουσας κατάστασης. Το σύστημά τους προβλέπει τη μελλοντική περιοχή εγκατάστασης μέσω ερωτήσεων για τη μετακίνηση των καταναλωτών και αρχίζει την επεξεργασία συμβάντων νωρίς για να παρέχει έγκαιρες πληροφορίες όταν οι καταναλωτές φθάνουν στις μελλοντικές τοποθεσίες. Οι εργασίες [25]

προτείνουν το RECEP, το οποίο εκμεταλλεύεται τα κοινά συμφέροντα στα δεδομένα και τα αποδεκτά ανακριβή αποτελέσματα για να ξαναχρησιμοποιηθούν στους υπολογισμούς και να μειώσουν τις ανάγκες στους πόρους. Το RECEP αυξάνει την επεκτασιμότητα και αποσβένει την καθυστέρηση των κινητών συστημάτων.

2.2 Το Πρόβλημα Της Απομακρυσμένης Εκτέλεσης Κώδικα Εφαρμογής

Υπάρχει ένας μεγάλος αριθμός μελετών σχετικά με τη μέθοδο εκφόρτωσης του κώδικα εφαρμογής από μια φτωχή σε πόρους κινητή συσκευή σε έναν πλούσιο σε πόρους υπολογιστικό κέντρο. Ωστόσο, η εκφόρτωση κώδικα σε απομακρυσμένο κέντρο εισάγει προβλήματα ασφάλειας και αξιοπιστίας [30]. Από την άλλη πλευρά, ένα κατανεμημένο μοντέλο υπολογιστών γνωστό ως cloud computing[5] μπορεί να επιφέρει και πολλά οφέλη όπως χαμηλή ενεργειακή κατανάλωση του χρήστη και τη μείωση του χρόνου απόκρισης των εφαρμογών προς εκτέλεση. Η ανάγκη ύπαρξης αλγορίθμου δρομολόγησης για ελαχιστοποίηση αυτών των παραμέτρων και μεγιστοποίηση της ποιότητα υπηρεσίας του χρήστη αποτελεί βασικό στόχο μελέτης πάνω στο κομμάτι του cloud computing.

Οι αρχιτεκτονικές κατατάσσονται σε δύο μεγάλες κατηγορίες σύμφωνα με το άρθρο [9] σε αυτές του ενός επιπέδου (single-tier architecture) όπου περιλαμβάνονται οι κινητές συσκευές (mobile devices) και τα απομακρυσμένα υπολογιστικά κέντρα (master clouds), και σε αυτές των δύο ή τριών επιπέδων (two-tier ή three-tier architecture) που περιλαμβάνει το απομακρυσμένο cloud, τα cloudlets και τις κινητές συσκευές. Η απόφαση επιλογής του κατάλληλου εξυπηρετητή για απομακρυσμένη εκτέλεση θα ληφθεί από την πλευρά του Master Cloud για να μειωθεί η επιβάρυνση των κινητών συσκευών.

2.2.1 Απομακρυσμένη Εκτέλεση Κώδικα Εφαρμογής στον Master Cloud

Στην ενότητα αυτήν περιγράφονται οι αρχιτεκτονικές όπου ο κώδικας εφαρμογής της κινητής συσκευής εκφορτώνεται σε απομακρυσμένο Master Cloud και τα σημαντικά έργα αυτής της κατηγορίας είναι τα MAUI[8], ThinkAir [18], CloneCloud [7], Cuckoo [16], CADA [21], μCloud [24].

2.2.1.1 Αρχιτεκτονική MAUI

Ο συγγραφέας του MAUI[8] εισήγαγε για πρώτη φορά μια παγκόσμια τεχνική βελτιστοποίησης στο κομμάτι του προβλήματος της εκφόρτωσης εργασίας στον Master Cloud για την ελαχιστοποίηση της ενεργειακής κατανάλωσης μιας κινητής συσκευής.

Ισχυρίστηκε ότι εξετάζοντας κάθε μία από τις παραμέτρους που επηρεάζουν το πρόβλημα δεν μπορεί ξεχωριστά να λύσει βέλτιστα το πρόβλημα της εκφόρτωσης, εφόσον είναι αδύνατον να καταγραφεί ολόκληρη η εικόνα της απομακρυσμένης εκτέλεσης μιας εφαρμογής. Έδειξαν ότι η εκφόρτωση κάποιας εργασίας μπορεί να οδηγήσει στην εκτέλεση περισσότερων εργασιών, εάν η εργασία που εκφορτώνεται στο κύριο σύννεφο επικαλείται άλλες εργασίες. Το MAUI αποφασίζει την εκφόρτωση εργασίας αναλύοντας το ενεργειακό κόστος ανάμεσα στην απομακρυσμένη και τοπική εκτέλεση. Αν και σε πολλές περιπτώσεις η εκφόρτωση εργασίας οδηγεί σε βελτίωση της απόδοσης, πολλές φορές αγνοεί την υπολογιστική πλευρά της εργασίας κάτι το οποίο θα μπορούσε να οδηγήσει σε καταστάσεις στις οποίες ενδέχεται να υποβαθμιστεί η απόδοση της εφαρμογής.

2.2.1.2 Αρχιτεκτονική CloneCloud

Η ανάλυση και ο διαχωρισμός της εφαρμογής εξετάζονται σε αυτό το άρθρο [7]. Οι συγγραφείς του CloneCloud εισήγαγαν μια τεχνική όπου η εφαρμογή διαχωρίζεται σε διάφορα τμήματα του κώδικα που θα μπορούσαν να εκφορτωθούν στο Master Cloud.

Στη συνέχεια δημιουργείται τυχαία σειρά εισόδων πριν από την εκτέλεση και με βάση κάθε τυχαία παραγόμενη είσοδο υλοποιείται γραμμική λύση βελτιστοποίησης προκειμένου να χωρίσουμε την εφαρμογή σε δύο ομάδες. Ένα από τα σύνολα θα εκτελεστεί στον Master Cloud και το άλλο πρέπει να εκτελεστεί στην τοπική συσκευή. Οι πληροφορίες κατάτμησης για αυτές τυχαίες σειρές εισόδων που δημιουργούνται από την εφαρμογή, αποθηκεύονται σε μια βάση δεδομένων και η εφαρμογή καθώς εκτελείται από τη κινητή συσκευή αντιστοιχίζεται με αυτή τη βάση δεδομένων και πληροφορίες διαχωρισμού από τη βάση δεδομένων που έχουν συλλεχθεί χρησιμοποιούνται για την εκφόρτωση κώδικα στο κύριο σύννεφο. Αυτή η τεχνική εξαρτάται σε μεγάλο βαθμό από τη τυχαιότητα με την οποία δημιουργούνται τα σύνολα εισόδου για την λήψη της απόφασης εκφόρτωσης. Ωστόσο ένα κύριο πρόβλημα που δημιουργείται είναι ότι μία τυχαία παραγόμενη είσοδος θα μπορούσε να μην έχει του κατάλληλους διαθέσιμους πόρους για να εκτελεστεί κατά την πορεία εκτέλεσης της εφαρμογής. Γενικά, ο χρήστης της εφαρμογής για κινητές συσκευές συμπεριφέρεται δυναμικά και αν ο χρήστης δίνει δεδομένα στην εφαρμογή που δεν μπορεί να ταιριάζει με κάποιες από τις συνθήκες που αποθηκεύονται στη βάση δεδομένων αυτής της μεθόδου τότε η εκφόρτωση θα αποτύχει δραστικά.

2.2.1.3 Αρχιτεκτονική ThinkAir

Το πρόβλημα των προηγούμενων έργων στα MAUI και CloneCloud αμφισβητήθηκε από τους συγγραφείς του άρθρου [18]. Για να βελτιώσουν την τεχνική της εκφόρτωσης κώδικα υιοθέτησαν μία online μέθοδο εκφόρτωσης εργασίας. Η αρχιτεκτονική κώδικα του ThinkAir απαιτεί από τον προγραμματιστή της εφαρμογής να σχολιάσει τις μεθόδους που θα μπορούσαν να εκφορτωθούν στο cloud. Αυτοί οι μέθοδοι θα πρέπει να ικανοποιούν ορισμένους περιορισμούς σε υπολογιστικούς/αποθηκευτικούς

πόρους. Σχεδίασαν επίσης έναν προσαρμοσμένο μεταγλωττιστή για τη μετατροπή ενός κώδικα εφαρμογής κατάλληλα σε κώδικα για απομακρυσμένη εκτέλεση. Αυτό το άρθρο λαμβάνει την απόφαση εκφόρτωσης με βάση ένα σύνολο προηγούμενων ιστορικών εκτέλεσης. Οι προηγούμενες τοπικές και μη τοπικές πληροφορίες εκτέλεσης αποθηκεύονται στη βάση δεδομένων. Κατά την εκφόρτωση η μέση τιμή του προηγούμενου τοπικού και απομακρυσμένου χρόνου εκτέλεσης χρησιμοποιούνται για την εκτίμηση του κόστους της απομακρυσμένης και τοπικής εκτέλεσης αντίστοιχα. Στη σύγκριση αυτών των δύο δαπανών του κόστους εκτέλεσης βασίζεται το ThinkAir προκειμένου να προχωρήσει στη λήψη της απόφασης για την εκφόρτωση του κώδικα ή όχι. Το πρόβλημα με αυτή την προσέγγιση είναι ότι με τον μέσο όρο του προηγούμενου ιστορικού εκτέλεσης δεν μπορεί να γίνει ακριβής πρόβλεψη στην εκτέλεση της τρέχουσας εργασίας εάν η συμπεριφορά της εργασίας δεν είναι συνεπής. Με άλλα λόγια ανακριβής εκτίμηση του κόστους εκτέλεσης μπορεί να οδηγήσει σε φτωχές επιλογές.

2.2.1.4 Αρχιτεκτονική CADA

Η προσέγγιση που λαμβάνει υπόψη το περιβάλλον για την εκφόρτωση εργασίας στον Master Cloud από την κινητή συσκευή εισήχθη στο άρθρο [21]. Ο συγγραφέας του CADA [21] ισχυρίστηκε ότι η κινητή συσκευή μετακινείται μέσα σε ένα συγκεκριμένο σύνολο θέσεων. Επιπλέον υπολόγισαν ότι η κατάσταση του δικτύου παραμένει η ίδια σε δεδομένη χρονική στιγμή και τοποθεσία. Έτσι οδηγήθηκαν στην αποθήκευση του ιστορικού εκτέλεσης εργασιών σε μια δεδομένη ώρα και τοποθεσία μέσα μια βάση δεδομένων. Κάθε φορά που μια εργασία εξετάζεται για την εκφόρτωση της στο κύριο σύννεφο, το κόστος της εκτιμάται με βάση το ιστορικό εκείνης της συγκεκριμένης χρονικής περιόδου και τοποθεσίας. Συγκρίνοντας αυτό το εκτιμώμενο κόστος, λαμβάνεται η απόφαση εκφόρτωσης. Καθώς ορισμένες από τις περιπτώσεις που η τεχνική αυτή μπορεί να εκτιμήσει αποτελεσματικά το κόστος εκτέλεσης μιας εργασίας, η εργασία αγνοεί τη δυναμική συμπεριφορά του χρήστη της εφαρμογής. Εάν ο χρήστης της εφαρμογής δεν συμπεριφέρονται με συνέπεια με τον χρόνο και τη θέση της κινητής συσκευής τότε η τεχνική αυτή δεν μπορεί να εκτιμήσει με ακρίβεια το κόστος εκτέλεσης εργασιών και αποτυγχάνει να λάβει απόφαση διόρθωσης σχετικά με την εκφόρτωση κώδικα.

2.2.1.5 Αρχιτεκτονική Cuckoo

Το Cuckoo [16] βασίζεται στην μερική εξάπλωση των εφαρμογών στο σύννεφο και έχει σχεδιαστεί με στόχο να καταστήσει εύκολο τον προγραμματισμό για τους προγραμματιστές, ενσωματώνοντας τα υπάρχοντα εργαλεία ανάπτυξης που είναι γνωστά στους προγραμματιστές. Επιπλέον, το Cuckoo έχει σχεδιαστεί για πλατφόρμα Android και υποστηρίζει τόσο τοπικές όσο και απομακρυσμένες υλοποιήσεις μεθόδων.

2.2.1.6 Αρχιτεκτονική μCloud

Το μοντέλο μCloud [24] επικεντρώνεται στη σύνθεση εφαρμογών για να υποστηρίξει την ευελιξία, τη δυνατότητα επαναχρησιμοποίησης και την ασφάλεια. Επομένως, για να επιτευχθεί η σύνθεση των εφαρμογών από ετερογενή συστατικά, τα συστατικά στοιχεία της εφαρμογής παρουσιάζονται με τη μορφή γραφήματος, όπου κάθε στοιχείο μπορεί να εκτελεστεί σε μία κινητή συσκευή, απομακρυσμένο cloud ή αμφότερα (που ονομάζονται υβριδικά συστατικά). Τα υβριδικά στοιχεία ενδέχεται να έχουν πολλαπλές υλοποιήσεις και απαιτούν την εκτέλεση ενός μεσαίου λογισμικού, όπως το WebOS. Επιπλέον, τα εξαρτήματα είναι εύκολα αναγνωρίσιμα και χαλαρά οριοθετημένα με παραμέτρους εισόδου και εξόδου, ιδιωτική μνήμη και πληροφορίες ρύθμισης. Στο μCloud, οι εφαρμογές παρουσιάζονται ως κατευθυνόμενα γραφήματα, όπου οι κόμβοι αντιπροσωπεύουν συστατικά και οι άκρες αντιπροσωπεύουν την ροή ελέγχου μεταξύ των συστατικών. Επομένως, όταν εκτελείται ένα γράφημα εφαρμογής, κάθε στοιχείο εισάγει την έξοδο του στις επόμενες συνιστώσες.

2.2.2 Απομακρυσμένη Εκτέλεση Κώδικα Εφαρμογής στα Cloudlets

Οι παραπάνω μέθοδοι περιλαμβάνουν μόνο τη κινητή συσκευή και τα απομακρυσμένα clouds, αλλά η εκφόρτωση κώδικα εφαρμογής σε απομακρυσμένο υπολογιστικό κέντρο συχνά οδηγεί σε υπερβολικά υψηλό κόστος, όπως υψηλή καθυστέρηση και χαμηλό εύρος ζώνης επειδή μπορεί να βρίσκεται πολύ μακριά από τη κινητή συσκευή. Έτσι, μερικά από τα άρθρα που παρουσιάζονται υποθέτουν ότι υπάρχει ένα ενδιάμεσο υπολογιστικό κέντρο που ονομάζεται cloudlet, στο οποίο κάθε κινητή συσκευή μπορεί να συνδεθεί μέσω ασύρματου δικτύου WiFi για να εκφορτώσει κώδικα εφαρμογής. Στη δεύτερη κατηγορία, μελετάται η εκφόρτωση εργασιών τόσο στα cloudlets όσο και στο απομακρυσμένο cloud με σημαντικά έργα σε αυτή την κατηγορία να είναι το ENDA[20], MuSIC[28], MAPCloud[27], Context Sensitive Offloading Scheme[45], Location-Aware Task Offloading[37], VM-based Cloudlet[29], Control-Theoretic [3, 2].

2.2.2.1 Αρχιτεκτονική MAPCloud

Το MAPCloud [27], θεωρείται μια αρχιτεκτονική δύο επιπέδων (two-tier architecture) που αποτελείται από τα cloudlets και το απομακρυσμένο cloud. Θεώρησαν την αίτηση εφαρμογής ως μία συνεχόμενη ροή εργασίας. Υποθέτοντας ότι η τοποθεσία ήταν γνωστή εκ των προτέρων, επέλεξαν τη σχεδόν βέλτιστη λύση για τη κατανομή των εφαρμογών των κινητών συσκευών στα cloudlets, που ικανοποιεί τη πολυδιάστατη ποιότητα υπηρεσίας της εφαρμογής ή τον περιορισμό της υψηλής ποιότητας υπηρεσίας του χρήστη, τη χαμηλή κατανάλωση ισχύος και τη μειωμένη καθυστέρηση. Ωστόσο δεν λαμβάνεται υπόψη η κινητικότητα των χρηστών στη προσπάθειά τους για την εύρεση της βέλτιστης κατανομής των πόρων των cloudlets. Ο χρήστης μπορεί να βρίσκεται πολύ μακριά από ένα cloudlet κατά το χρόνο επιστροφής του αποτελέσματος, εάν σε προηγούμενη χρονική στιγμή έχει επιλεγεί ως

η βέλτιστη λύση. Έτσι, η κινητικότητα αποτελεί σημαντικό παράγοντα στη περίπτωση της επιλογής των καταλληλότερων εξυπηρετητών του συστήματος.

2.2.2.2 Αρχιτεκτονική MUSIC

Για να αντιμετωπιστεί το θέμα της κινητικότητας, στο άρθρο του MuSIC [28] (three-tier architecture) επέκτειναν την εργασία του MAPCloud, όπου θεωρούσαν την αίτηση εφαρμογής ως ροή εργασίας με βάση το χρόνο παραμονής του χρήστη εντός του πεδίου εμβέλειας του cloudlet, το οποίο είναι γνωστό ως location-time workflow(LTW). Ο αλγόριθμος βελτιστοποίησης σχεδιάστηκε για να βρει τον κατάλληλο cloudlet, με βάση τον χρόνο σύνδεσης του κινητού και διακομιστή εξασφαλίζοντας τους περιορισμούς που έχουν να κάνουν με την υψηλή ποιότητα υπηρεσίας του χρήστη, την ισχύ και την καθυστέρηση. Αν και έλαβαν υπόψη την κινητικότητα των χρηστών σε αυτό το μοντέλο, δεν έλαβαν υπόψη την ταχύτητα και την αλλαγή κατεύθυνσης του χρήστη. Εκτός από την κινητικότητα, έχουν θεωρήσει ότι οι πόροι για όλα τα cloudlets είναι ομοιογενείς τόσο στο MAPCloud όσο και στο MUSIC κάτι το οποίο δεν αντιστοιχεί σε πραγματικό σενάριο.

2.2.2.3 Αρχιτεκτονική ENDA

Στη αρχιτεκτονική ENDA [20], επιλέγεται το βέλτιστο cloudlet βάσει της πρόβλεψης της κινητικότητας του χρήστη, του φόρτου εργασίας και της ποιότητας του δικτύου. Το μονοπάτι του χρήστη που δημιουργείται καθώς αυτός κινείται στο χώρο προβλέπεται από το ιστορικό κινητικότητας του χρήστη που είναι αποθηκευμένα στη βάση δεδομένων. Το cloudlet που μπορεί να εκπληρώσει τον επιθυμητό χρόνο απόκρισης με τη χαμηλότερη καθυστέρηση στην προβλεπόμενη διαδρομή επιλέγεται για να εκτελέσει το αίτημα της εφαρμογής για απομακρυσμένη εκτέλεση. Εάν δεν υπάρχουν διαθέσιμα cloudlets, η εφαρμογή εκφορτώνεται μέσω δικτύου 3G στον Master Cloud. Ακόμα, αν υπάρχουν αποτυχίες σύνδεσης κατά τη διάρκεια της εκφόρτωσης, η συσκευή περιμένει την επανασύνδεση. Όπως και το MuSIC, το έργο αυτό δεν έλαβε υπόψη ούτε την ταχύτητα και την αλλαγή κατεύθυνσης των χρηστών αλλά ούτε και την ετερογένεια των cloudlets.

2.2.2.4 Αρχιτεκτονική Context Sensitive Offloading Scheme

Στο άρθρο [45] εξέτασαν το περιεχόμενο των κινητών συσκευών όπως τη κατάσταση του δικτύου, τη τοποθεσία, το φόρτο εργασίας κ.λ.π. προκειμένου να γίνει η εκφόρτωση εργασίας σε ένα cloudlet, καθώς αυτοί οι παράμετροι αλλάζουν συνεχώς με την κίνηση των κινητών συσκευών καθ' όλη τη διάρκεια της ημέρας. Το cloudlet με το χαμηλότερο κόστος εκτέλεσης στο σύστημα επιλέγεται ως η θέση εκτέλεσης της εργασίας εξ' αποστάσεως. Αν και έλαβαν υπόψη την κατάσταση του δικτύου, τον φόρτο εργασίας, δεν θεωρούσαν την κινητικότητα των χρηστών και την ετερογένεια των cloudlets στην επιλογή του εξυπηρετητή.

2.2.2.5 Αρχιτεκτονική Location-Aware Task Offloading

Όλα τα έργα που αναφέρθηκαν νωρίτερα δεν θεωρούσαν τον περιορισμό της πρόσβασης του χρήστη μέσω της ασύρματης σύνδεσης που υλοποιείται. Στο άρθρο [37] θεωρούσαν αυτόν τον παράγοντα για την επιλογή του κατάλληλου εξυπηρετητή στο σύστημα. Πρώτα εντοπίζονται ποια σημεία πρόσβασης είναι ελεύθερα να συνδεθούν, και έπειτα επιλέγεται το cloudlet με το χαμηλότερο ενεργειακό κόστος ως η θέση για να εκφορτωθεί ο κώδικας της εφαρμογής.

2.2.2.6 Αρχιτεκτονική Vm-based Cloudlet

Στο άρθρο [29] προτείνεται ένα μοντέλο που βασίζεται στην τεχνική επαυξημένης εκτέλεσης. Το μοντέλο χρησιμοποιεί μια έννοια της εικονικής μηχανής που τρέχει σε έναν αξιόπιστο και πλούσιο σε πόρους υπολογιστή (Master Cloud) ή ένα σύμπλεγμα υπολογιστών (cloudlet). Επιπλέον, οι κινητές συσκευές λειτουργούν σαν ένα λεπτό στρώμα πελάτη και μεταφορτώνουν τις εργασίες που απαιτούν εντατικές πηγές πόρων στο cloudlet. Το άρθρο παρουσιάζει δύο προσεγγίσεις για τον υπολογισμό της ροής, της μετανάστευσης VM και της σύνθεσης VM. Η προσέγγιση μετανάστευσης VM αναστέλλει την εκτέλεση του VM και αποθηκεύει τις καταστάσεις του επεξεργαστή, του δίσκου και της μνήμης. Στη συνέχεια, το VM μεταφέρεται στο cloudlet και ξαναρχίζει την εκτέλεση από το αποθηκευμένο σημείο. Εναλλακτικά, η σύνθεση VM παράγει μια μικρή επικάλυψη VM από την κινητή συσκευή και μετακινεί την επικάλυψη στο cloudlet. Η επικάλυψη VM εφαρμόζεται στη βάση VM του cloudlet και η εκτέλεση συνεχίζεται από το αποθηκευμένο σημείο.

2.2.2.7 Αρχιτεκτονικές Control-Theoretic

Τα άρθρα [3] και [2] ασχολούνται με την διαχείριση των πόρων των cloudlet. Πιο συγκεκριμένα η [3] εστιάζει στη χρήση μιας Edge Computing υποδομής για την έγκαιρη ανίχνευση πυρκαγιών. Μη επανδρωμένα οχήματα επιτηρούν μεγάλες δασικές εκτάσεις και μεταφορτώνουν τις φωτογραφίες που συλλέγουν σε κοντινά cloudlet για περαιτέρω επεξεργασία. Σε περίπτωση πυρκαγιάς, ο αριθμός των αιτήσεων αυξάνεται σημαντικά με αποτέλεσμα να είναι αναγκαία η δυναμική διαχείρισή πόρων. Για τη μοντελοποίηση της εφαρμογής για την επεξεργασία εικόνας έγινε χρήση γραμμικών καταστατικών συστημάτων και σχεδιάστηκαν ελεγκτές ανάδρασης για τη διαχείριση των πόρων των cloudlets. Στην εργασία [2], σχεδιάστηκε μια γενική μεθοδολογία μοντελοποίησης και ελέγχου IoT (Internet Of Things) εφαρμογών που στηρίζονται σε Edge Computing υποδομές. Για τη μοντελοποίηση χρησιμοποιήθηκαν γραμμικά διακοπτικά συστήματα με περιορισμούς, ενώ για τη διαχείριση των πόρων σχεδιάστηκαν ελεγκτές ανάδρασης στηριζόμενοι στη σύγχρονη θεωρία ελέγχου. Επίσης επιλύθηκε ένα πρόβλημα μικτού ακεραίου προγραμματισμού για την τοποθέτηση και αυξομείωση των ενεργών VM ανάλογα με το μεταβαλλόμενο φορτίο των εφαρμογών.

Κεφάλαιο 3

Σχεδιασμός Συστήματος Και Ενεργειακά Μοντέλα

Στην ενότητα αυτή γίνεται η εισαγωγή στις κύριες πτυχές της απομακρυσμένης εκτέλεσης που πρέπει να ληφθούν υπόψη στη διαδικασία σχεδίασης, περιγράφονται οι βασικές έννοιες Υπολογιστικού Κινητού Νέφους, ο τρόπος λειτουργίας της προτεινόμενης αρχιτεκτονικής του Mobi-Het καθώς περιγράφεται και ένα σύνολο ενεργειακών μοντέλων που χρησιμοποιήθηκαν για τα πειραματικά αποτελέσματα αυτής της διπλωματικής εργασίας.

3.1 Αρχιτεκτονική Δομή Υπολογιστικού Κινητού Νέφους

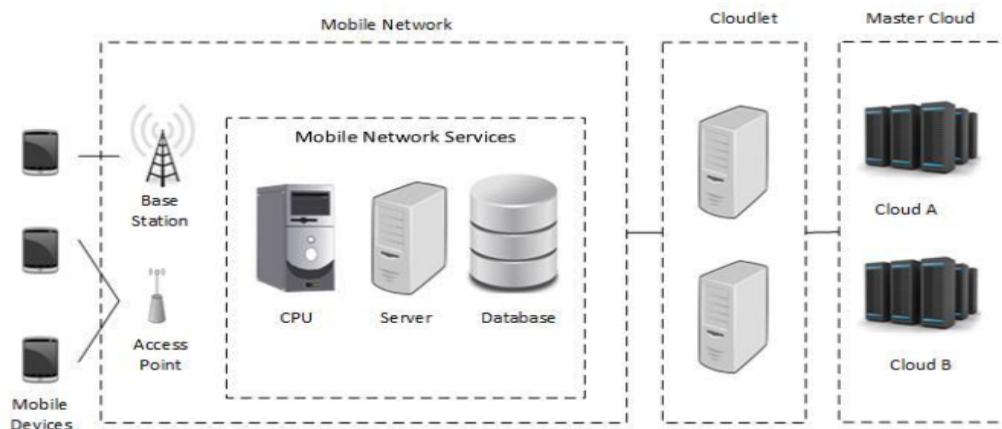
Οι κινητές συσκευές καθίστανται σταδιακά ένα αναπόσπαστο κομμάτι της ανθρώπινης ζωής με μεγάλη φορητότητα και ευελιξία. Μια βασική ανεπάρκεια της κινητής συσκευής είναι η έλλειψη πόρων. Το Mobile Cloud Computing (MCC) [11] έχει ξεπεράσει αυτή την ανεπάρκεια. Σε αυτή τη παράγραφο παρέχεται μια σύντομη περιγραφή του MCC, συμπεριλαμβανομένου του ορισμού, των εφαρμογών και των πλεονεκτημάτων του για μεγαλύτερη κατανόηση.

3.1.1 Ορισμός Υπολογιστικού Κινητού Νέφους

Το Mobile Cloud Computing (MCC) [11] είναι μια τεχνολογία όπου οι κινητές εφαρμογές αποκοτούν την απαιτούμενη μεγάλη πηγή πόρων από μεγαλύτερες υποδομές υπολογιστικών και αποθηκευτικών πόρων που λέγονται clouds. Οι εφαρμογές για κινητές συσκευές μετακινούν την υπολογιστική ισχύ και την αποθήκευση δεδομένων μακριά από τα κινητά τηλέφωνα αφήνοντας έτσι ελεύθερους τους διαθέσιμους πόρους του με αποτέλεσμα την εξοικονόμηση ενέργειας του χρήστη.

3.1.2 Οργάνωση Δομής Υπολογιστικού Κινητού Νέφους

Το Σχήμα 3.1 δείχνει τη βασική αρχιτεκτονική του Mobile Cloud Computing (MCC) [11]. Οι σταθμοί βάσης και τα σημεία ασύρματης πρόσβασης συνδέουν τις κινητές συσκευές στο δίκτυο κινητής τηλεφωνίας. Οι διασυνδέσεις του δικτύου μεταξύ των κινητών δικτύων και των συσκευών δημιουργούνται και ελέγχονται μέσω των σταθμών βάσης (Base Station) και των σημείων πρόσβασης (Access Points). Η CPU που είναι συνδεδεμένη στο δίκτυο κινητής τηλεφωνίας λαμβάνει το αίτημα από τις κινητές συσκευές. Υπηρεσίες όπως η πιστοποίηση ταυτότητας, η εξουσιοδότηση κ.λ.π. παρέχονται από το δίκτυο κινητής τηλεφωνίας βάσει των στοιχείων του τοπικού πράκτορα και των συνδρομητών από τη βάση δεδομένων. Στη συνέχεια, το αίτημα του συνδρομητή προωθείται στο σύννεφο μέσω του Διαδικτύου. Οι ελεγκτές της υποδομής επεξεργάζονται την αίτηση από τους συνδρομητές και παρέχουν τις ζητούμενες υπηρεσίες cloud στις κινητές συσκευές. Τα κοντινά cloudlets χρησιμοποιούνται για να αυξήσουν την απόδοση του συστήματος, καθώς μπορεί να απέχουν σε μεγάλο βαθμό οι χρήστες από τα clouds. Τα cloudlets χαρακτηρίζονται ως "ένας αξιόπιστος πλούσιος σε πόρους υπολογιστής ή ένα σύμπλεγμα υπολογιστών που είναι καλά συνδεδεμένο με το Διαδίκτυο και είναι διαθέσιμα για λήψη κώδικα εφαρμογών από κοντινές κινητές συσκευές" [29]. Τα cloudlets λειτουργούν ως μία μεσαία βαθμίδα μεταξύ των κινητών συσκευών και του απομακρυσμένου cloud. Όλες οι εργασίες που απαιτούν υψηλή υπολογιστική ισχύ και πολλούς αποθηκευτικούς πόρους για την εκτέλεση τους, αφαιρούνται από τα cloudlets για να επιτευχθεί ένας χαμηλός χρόνος απόκρισης.



Σχήμα 3.1: Αρχιτεκτονική Δομή Υπολογιστικού Κινητού Νέφους

3.1.3 Εφαρμογές Υπολογιστικού Κινητού Νέφους

Με τη χρήση των κινητών συσκευών, ο αριθμός των εφαρμογών κινητής τηλεφωνίας έχει αυξηθεί πάρα πολύ. Οι κινητές εφαρμογές που χρησιμοποιούν τα πλεονεκτήματα του Mobile Cloud Computing [11] αναφέρονται σύντομα παρακάτω.

- **m-learning:** Το m-learning μαθαίνει μέσω του τρόπου λειτουργίας της κινητής συσκευής. Είναι μια συγχώνευση της ηλεκτρονικής μάθησης και της κινητικότητας του χρήστη. Αλλά υποφέρει από περιορισμούς όπως το υψηλό κόστος των κινητών συσκευών, περιορισμούς στους εκπαιδευτικούς πόρους, το χαμηλό ρυθμό μετάδοσης κ.λ.π. Αυτοί οι περιορισμοί αντιμετωπίζονται με τη βοήθεια του cloud computing.
- **m-commerce:** Το m-commerce βοηθά στην παροχή ηλεκτρονικού εμπορίου με τη βοήθεια κινητών συσκευών. Στην πραγματικότητα αποτελείται από τις εφαρμογές που απαιτούν κινητικότητα όπως η μεταφορά χρημάτων μέσω κινητού τηλεφώνου, η έκδοση κινητών εισιτηρίων, τα κινητά κουπόνια, τα κουπόνια και η αφοσίωση κ.λ.π. Δεδομένου ότι το m-commerce περιλαμβάνει κινητικότητα, αντιμετωπίζει ορισμένα προβλήματα όπως η ασφάλεια και οι πολύπλοκες διαμορφώσεις κινητών συσκευών. Η ενσωμάτωση του m-commerce με το cloud computing μπορεί να βοηθήσει στη μείωση αυτών των προβλημάτων.
- **m-gaming:** Το εύρος των παιχνιδιών για κινητά είναι γενικά μικρό λόγω της περιορισμένης επεξεργασίας και της ισχύος των μπαταριών των συσκευών. Αλλά αυτό μπορεί να αντιμετωπιστεί με τη βοήθεια του Mobile Cloud Computing (MCC) [11]. Τα κινητά παιχνίδια μπορούν να εκφορτωθούν εντελώς στο σύννεφο. Αυτό μειώνει την εργασία υπολογισμού στις κινητές συσκευές και εξοικονομεί ενέργεια.

3.1.4 Πλεονεκτήματα Και Οφέλη Υπολογιστικού Κινητού Νέφους

Το Mobile Cloud Computing (MCC) [11] έχει πολλά πλεονεκτήματα για τους τελικούς χρήστες μερικά από τα οποία αναφέρονται παρακάτω:

- **Επεξεργαστική Ισχύς Κινητής Συσκευής:** Οι κινητές συσκευές έχουν περιορισμένους πόρους. Φυσικά, οι υπολογισμοί εφαρμογών που απαιτούν υψηλούς υπολογιστικούς/αποθηκευτικούς πόρους για να εκτελεστούν δεν είναι εφικτοί σε αυτές τις συσκευές. Το Mobile Cloud Computing (MCC) [11] παρέχει στις συσκευές μια τεράστια πηγή πόρων. Οι υπολογισμοί αυτού του είδους πραγματοποιούνται στην πλευρά των clouds. Έτσι, η περιορισμένη ισχύς επεξεργασίας των συσκευών δεν παρεμποδίζει την εφαρμογή υψηλής έντασης υπολογισμών να εκτελεστεί απομακρυσμένα.
- **Μπαταρία Κινητής Συσκευής:** Ένα άλλο σημαντικό μειονέκτημα των κινητών συσκευών είναι η περιορισμένη ισχύς της μπαταρίας. Η κατανάλωση ενέργειας μπορεί να μειωθεί με την αύξηση της απόδοσης της CPU, τη λειτουργία δίσκου και οθόνης. Η εκφόρτωση της εργασίας στο νέφος του κινητού βοηθά στην αντιμετώπιση αυτού του ζητήματος. Καθώς οι εργασίες έντασης υπολογισμών εκφορτώνονται στο σύννεφο, αποφεύγεται ο μεγάλος χρόνος εκτέλεσης των κινητών συσκευών, ο οποίος με τη σειρά του αυξάνει τη διάρκεια ζωής των συσκευών.

- **Ικανότητα αποθήκευσης δεδομένων:** Η χωρητικότητα αποθήκευσης είναι μια άλλη συνιστώσα των κινητών συσκευών. Οι συσκευές έχουν φυσικά μικρή χωρητικότητα αποθήκευσης. Το Mobile Cloud Computing (MCC) [11] μπορεί να χρησιμοποιηθεί για την κλιμάκωση της χωρητικότητας αποθήκευσης των συσκευών χρησιμοποιώντας την τεράστια χωρητικότητα αποθήκευσης του cloud μέσω του ασύρματου δικτύου.
- **Αξιοπιστία Απομακρυσμένης Εκτέλεσης:** Το Mobile Cloud Computing (MCC) [11] εντείνει την αξιοπιστία των κινητών συσκευών καθώς τα δεδομένα αποθηκεύονται σε αρκετούς υπολογιστές στο νέφος. Τελικά η πιθανότητα απώλειας των δεδομένων μειώνεται. Επιπλέον, ο σχεδιασμός του MCC μπορεί να γίνει με τρόπους ώστε να μετατραπεί σε ένα μοντέλο ασφάλειας δεδομένων για τους χρήστες και τους παρόχους των υπηρεσιών.

3.2 Σχεδίαση Αρχιτεκτονικής Συστήματος(Mobi-Het)

Το πρόβλημα της εφαρμογής μεγάλων δεδομένων που εκφορτώνεται από μια κινητή συσκευή σε ένα απομακρυσμένο cloud έχει μελετηθεί καλά στα άρθρα [8, 17, 19, 28, 20, 18, 43]. Ωστόσο, αυτά τα υπάρχοντα έργα περιορίζονται με τον ένα ή τον άλλο τρόπο, και μόνο μερικά από αυτά έχουν θεωρήσει τα cloudlets ως υπολογιστικά κέντρα. Για παράδειγμα, στο MuSIC[28], η δυναμική χαρτογράφηση των εφαρμογών σε πόρους σε μια πολυεπίπεδη αρχιτεκτονική διατυπώθηκε ως πρόβλημα ροής εργασίας (LTW) για τον εντοπισμό μοτίβων κινητικότητας των χρηστών. Το ENDA[20] προσπάθησε να προσδιορίσει το πρόβλημα της λήψης της πιο αποτελεσματικής απόφασης εκφόρτωσης εργασίας σε cloudlet, προβλέποντας ένα αρχείο ιστορικού χρήστη. Ωστόσο, και στα δύο έργα, δεν ελήφθησαν υπόψη ούτε τις απαιτήσεις ποιότητας υπηρεσίας του χρήστη ούτε την ετερογένεια στους πόρους των cloudlets. Σε ένα πραγματικό σενάριο, τα cloudlets έχουν ετερογενή υπολογιστικά, αποθηκευτικά και δικτυακά μέσα.

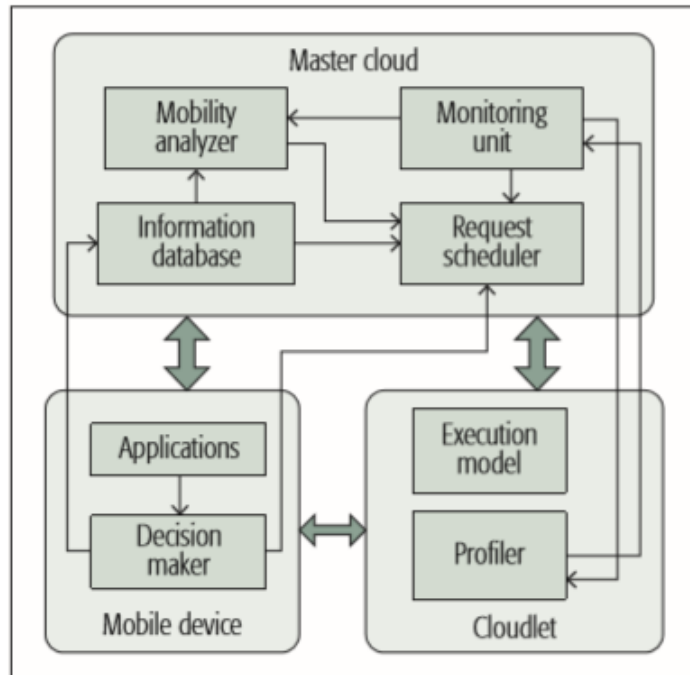
Το σχήμα 3.2 απεικονίζει τα τρία βασικά αρχιτεκτονικά στοιχεία τα οποία και αναλύονται στη συνέχεια, πάνω στα οποία βασίζεται ο προτεινόμενος αλγόριθμος δυναμικής δρομολόγησης εργασιών Mobi-Het:

1. **Κινητές Συσκευές (Mobile Devices):** Υποθέτουμε ότι κάθε κινητή συσκευή στην αρχιτεκτονική του Mobile Cloud Computing μπορεί να εκτελεί διάφορες εφαρμογές μεγάλου όγκου δεδομένων (εφαρμογές επεξεργασίας εικόνας κ.λ.π.) έχοντας διαφορετικές προτεραιότητες, επιδόσεις και απαιτήσεις πόρων. Συχνά είμαστε αναγκασμένοι να εκτελέσουμε μέρος του κώδικα εφαρμογής σε ένα απομακρυσμένο υπολογιστικό κέντρο. Η μονάδα λήψης αποφάσεων στην κινητή συσκευή λαμβάνει την απόφαση αν ένα αίτημα εργασίας πρέπει να εκτελείται εξ αποστάσεως ή τοπικά και η απόφαση αυτή εξαρτάται από πολλούς παράγοντες, συμπεριλαμβανομένου του χρόνου εκτέλεσης, τη συνδεσιμότητά του δικτύου και την τρέχουσα αποθηκευμένη ενέργεια στις μπαταρίες των κινητών συσκευών.

2. **Τοπικό Υπολογιστικό Νέφος (Cloudlet):** Στο περιβάλλον του Mobile Cloud Computing (MCC) όπου έχουμε πολλαπλούς κλιμακωτούς υπολογιστές, οι υπολογιστικοί πόροι που απαρτίζουν τα cloudlets, υποτίθεται ότι είναι διαθέσιμοι πίσω από κάθε σημείο ασύρματης πρόσβασης δικτύου WiFi. Κάθε cloudlet είναι ένας μικρός εξυπηρετητής (mini-cloud) που διαθέτει περιορισμένους πόρους υποδομής και λιγότερη υπολογιστική ισχύ από απομακρυσμένο cloud, με ετερογενή υπολογιστικά, επικοινωνιακά και αποθηκευτικά χαρακτηριστικά. Ωστόσο, βρίσκονται κοντά στις κινητές συσκευές και κάθε cloudlet είναι ικανό να εκτελεί πολλαπλές εικονικές μηχανές (VMs), επιτρέποντάς τους να εκτελούν αιτήσεις που υποβάλλονται από εφαρμογές κινητών συσκευών. Κάθε cloudlet $c \in C$ μπορεί να έχει διάφορους τύπους VMs και ζ_i αντιπροσωπεύει το μέγεθος ενός VM τύπου i [6]. Η υπολογιστική ισχύς ενός VM υπολογίζεται σε MIPS και μπορεί να διαφέρει σε κάθε VM του cloudlet. Στα πλαίσια της εργασίας αυτής υποθέτουμε ότι όλα τα VMs ενός cloudlet είναι του ίδιου τύπου με τα ίδια χαρακτηριστικά.
3. **Απομακρυσμένο Υπολογιστικό Νέφος (Master Cloud):** Υποθέτουμε ότι ένα κύριο ή αλλιώς απομακρυσμένο cloud (Master Cloud) έχει μεγάλους πόρους υποδομής και υψηλή υπολογιστική ισχύ. Ο αλγόριθμος κατανομής πόρων εκτελείται στο απομακρυσμένο cloud σε τακτικά διαστήματα προγραμματισμού. Το Master Cloud έχει τέσσερις κύριες μονάδες τα οποία και αναλύονται:
- (i) *Τη βάση δεδομένων πληροφοριών:* Η βάση δεδομένων περιέχει ιστορικές πληροφορίες κινητικότητας των ενεργών κινητών συσκευών για κάθε ένα από τα συνδεδεμένα σε αυτά cloudlets.
 - (ii) *Τον αναλυτή κινητικότητας:* Ο αναλυτής κινητικότητας αναλύει τα μοτίβα κινητικότητας μιας κινητής συσκευής για να εκτιμήσει την ταχύτητα και την κατεύθυνση της σε μελλοντικά βήματα χρόνου, με βάση τις ιστορικές πληροφορίες που αποθηκεύονται στη βάση δεδομένων πληροφοριών.
 - (iii) *Τη μονάδα παρακολούθησης:* Η μονάδα παρακολούθησης παρακολουθεί το συνολικό φορτίο όλου συστήματος και τη χρήση υπολογιστικών πόρων κάθε cloudlet, έτσι ώστε οι διαθέσιμοι ελεύθεροι πόροι να μπορούν να προγραμματιστούν με βάση τα αιτήματα εφαρμογής με αποτελεσματικό τρόπο. Η συντριπτική υπερφόρτωση ενός ενιαίου cloudlet με υπερβολικό φόρτο εργασίας μπορεί να υποβαθμίσει τη συνολική απόδοση των εκτελέσεων απομακρυσμένου κώδικα κάτι το οποίο φαίνεται και από τα πειραματικά αποτελέσματα. Έτσι, το προτεινόμενο σύστημα Mobi-Het στοχεύει στην επίτευξη ισορροπημένης κατανομής του φόρτου εργασίας μεταξύ όλων των διαθέσιμων cloudlets.
 - (iv) *Τον προγραμματισμό αιτημάτων:* Η μονάδα χρονοδιαγράμματος αιτήματος λαμβάνει όλες τις αιτήσεις εκτέλεσης κώδικα απομακρυσμένης εφαρμογής από τις κινητές συσκευές και τις προγραμματίζει μετά από ένα ορισμένο χρονικό διάστημα δ . Η ετερογένεια των διαθέσιμων υπολογιστικών πόρων σε κάθε cloudlet, καθώς και η μεταβλητή προθεσμία καθυστέρησης

Τα της αίτησης που ζητά μία εφαρμογή $a \in A$ ενός κινητού για απομακρυσμένη εκτέλεση μπορεί να θέσει επιπρόσθετες προκλήσεις στο πρόβλημα της κατανομής των πόρων.

Επιπρόσθετα γνωρίζει όλες τις πληροφορίες των σταθερών κέντρων εκπομπής σήματος WiFi για κάθε cloudlet (x_c, y_c) και το εύρος μετάδοσης (ακτίνα R_c) του αντίστοιχου σημείου πρόσβασης.



Σχήμα 3.2: Αρχιτεκτονική σχεδίασης Mobi-Het.

3.3 Βασικές Έννοιες Για Την Αναπαράσταση Δομικών Στοιχείων Της Αρχιτεκτονικής Mobi-Het

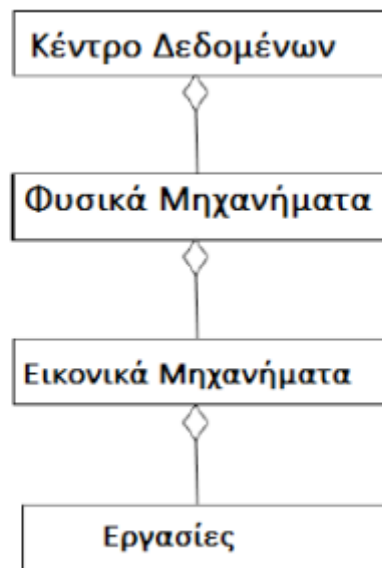
Στα πλαίσια αυτής της διπλωματικής εργασίας θα προσπαθήσουμε να αναπαραστήσουμε τα βασικά αρχιτεκτονικά στοιχεία του Mobi-Het σε οντότητες ενός προσομοιωτή στον οποίο θα σχεδιάσουμε όλο το σύστημα που προαναφέρθηκε για να εκτελέσουμε τα πειράματα που θα ακολουθήσουν. Ο προσομοιωτής της επιλογής μας ήταν το πακέτο εργαλείων του CloudSim Plus [32]. Το CloudSim Plus είναι ένα λειτουργικό, ιδιαίτερα εκτεταμένο πλαίσιο προσομοίωσης που επιτρέπει τη μοντελοποίηση, την προσομοίωση και τον πειραματισμό των αναδυόμενων υποδομών υπολογιστών υπολογιστικού νέφους, και των υπηρεσιών εφαρμογής. Επιτρέπει στους χρήστες να εστιάζουν σε συγκεκριμένα ζητήματα σχεδίασης του συστήματος που πρέπει να διερευνηθούν, χωρίς να ανησυχούν για τις λεπτομέρειες χαμηλού επιπέδου που σχετίζονται με τις υποδομές και τις υπηρεσίες που βασίζονται στο υπολογιστικό νέφος.

Μια περιγραφή των κύριων κλάσεων που χρησιμοποιούνται για τη δημιουργία

ενός τοπικού υπολογιστικού νέφους (π.χ cloudlet) και κατ' επέκταση ενός απομακρυσμένου διακομιστή (π.χ Master Cloud) παρουσιάζεται παρακάτω:

- (i) *Datacenter και VmAllocationPolicy*: Ένα Datacenter περιέχει ένα σύνολο φυσικών μηχανών (PMs) κεντρικών υπολογιστών που παρέχει τη βασική υποδομή cloud. Για κάθε δημιουργημένο Datacenter, πρέπει να οριστεί μια παράμετρος VmAllocationPolicy. Αυτό το αντικείμενο αποφασίζει ποιο PM θα φιλοξενήσει κάθε εικονικό μηχανήμα (VM).
- (ii) *Host, Pe και VmScheduler*: Ένας Host αντιπροσωπεύει στην ουσία ένα φυσικό μηχανήμα (PM) και για κάθε Host πρέπει να οριστεί ένας κατάλογος των στοιχείων επεξεργασίας δηλαδή ένας κατάλογος από τους πυρήνες CPU του μηχανήματος (Pes). Καθώς ο Host είναι ικανός να φιλοξενήσει VMs, είναι απαραίτητο η ύπαρξη ενός αλγορίθμου (VmScheduler) που θα χρησιμοποιηθεί για τη διαχείριση ταυτόχρονης εκτέλεσης πολλαπλών VMs. Υπάρχουν διάφοροι VmSchedulers, όπως το κοινόχρηστο στο χρόνο και στο διάστημα, που μπορούν να χρησιμοποιηθούν.
- (iii) *DatacenterBroker*: Αντιπροσωπεύει ένα λογισμικό που ενεργεί για λογαριασμό ενός πελάτη στο cloud, λαμβάνει αιτήματα και εκτελεί απαιτούμενες ενέργειες για να τους παρακολουθήσει. Αυτές οι ενέργειες περιλαμβάνουν την υποβολή των αιτημάτων εφαρμογών για απομακρυσμένη εκτέλεση (Requests) που θα εκτελεστούν μέσα σε κάποια από τα δημιουργημένα VMs των cloudlets.
- (iv) *Vm και CloudletScheduler*: Ένα αντικείμενο VM αντιπροσωπεύει μια εικονική μηχανή που τρέχει μέσα σε ένα Host και θα εκτελέσει τα υποβληθέντα Requests. Ένας CloudletScheduler προγραμματίζει πως η παράλληλη εκτέλεση πολλαπλών Request θα γίνει μέσα σε ένα VM. Ακολουθεί το ίδιο σκεπτικό του VmScheduler, όπως περιγράφηκε προηγουμένως.
- (v) *Requests*: Ένα Request αντιπροσωπεύει ένα πρόγραμμα που θα τρέχει μέσα σε ένα Vm, προσδιορισμένο με βάση τα χαρακτηριστικά της εφαρμογής της κινητής συσκευής, όπως τον αριθμό των εκατομμυρίων εκτέλεσης εντολών (MIPS), τον αριθμό των στοιχείων επεξεργασίας (Pes) και τα μοντέλα αξιοποίησης για CPU, RAM και bandwidth. Κάθε αντικείμενο UtilizationModel προσδιορίζει τον τρόπο με τον οποίο ένας δεδομένος πόρος θα χρησιμοποιηθεί από το Cloudlet κατά τη διάρκεια του χρόνου. Γνωστό τέτοιο αντικείμενο είναι το UtilizationModelFull, το οποίο και χρησιμοποιήθηκε και υποδηλώνει ότι ένας διαθέσιμος πόρος θα χρησιμοποιηθεί στο 100% καθ' όλη τη διάρκεια της προσομοίωσης.

Στο Σχήμα 3.3 παρουσιάζεται συνοπτικά την αλληλουχία των κλάσεων που μόλις περιγράφηκαν για τη λειτουργία ενός τοπικού υπολογιστικού νέφους.



Σχήμα 3.3: Διάγραμμα Κλάσεων Του Υπολογιστικού Νέφους.

3.4 Ενεργειακά Μοντέλα Κινητής Συσκευής

Σημαντικό ρόλο παίζει και η ενεργειακή κατανάλωση της κινητής συσκευής καθώς για να είναι εφικτή η εκτέλεση εφαρμογών τοπικά και μη θα πρέπει το κινητό να έχει επαρκή μπαταρία. Μία από τις κύριες συνιστώσες της απομακρυσμένης εκτέλεσης εργασιών είναι εξάλλου και η εξοικονόμηση ενέργειας του χρήστη προκειμένου η κινητή συσκευή να παραμείνει όσο γίνεται πιο ενεργή για να μπορεί να εκτελεί τοπικά ή μη εφαρμογές στο λιγότερο δυνατό χρονικό διάστημα απόκρισης. Στη συνέχεια εξετάζονται τα τρία κύρια ενεργειακής κατανάλωσης στοιχεία της κινητής συσκευής τα οποία και αναλύονται:

1. *Ενεργειακή Κατανάλωση Υπολογιστικής Επεξεργασίας(CPU),*
2. *Ενεργειακή Κατανάλωση Ασύρματου Δικτύου WiFi,*
3. *Ενεργειακή Κατανάλωση Δικτύου 3G.*

3.4.1 Ενεργειακή Κατανάλωση Υπολογιστικής Μονάδας Επεξεργασίας(CPU)

Σύμφωνα με το άρθρο [42] οι επεξεργαστές αποτελούν τη κύρια πηγή κατανάλωσης ενέργειας περισσότερο από κάθε παθητικό στοιχείο της κινητής συσκευής. Κάθε υπολογιστική μονάδα έχει μια συχνότητα λειτουργίας (πόσες εντολές μπορεί να εκτελέσει

στη μονάδα του χρόνου) που είναι ανάλογη της ενεργειακής κατανάλωσης για ένα πρότυπο επεξεργαστή. Η CPU δεν μπορεί να απενεργοποιηθεί, επομένως στη συνολική κατανάλωση ισχύος της CPU πρέπει να προστεθεί και η ενεργειακή κατανάλωση σε κατάσταση αδράνειας (κατάσταση μη χρησιμοποίησης της επεξεργαστικής μονάδας). Για αυτό, με βάση τον Πίνακα 3.1 δημιουργούμε το διακριτό γραμμικό μοντέλο ισχύος. Το μοντέλο ισχύος CPU έχει ως εξής:

$$P_{CPU} = \beta_i^{freq} \times u + \beta_i^{idle} \quad 0 \leq u \leq 100\%, \quad i = 0, 1, 2, \dots, n \quad (3.1)$$

, όπου i είναι ο δείκτης του Πίνακα Κατανομής Τιμών 3.1, β_i^{freq} είναι η κατανάλωση ισχύος της CPU στην i_{η} συχνότητα της CPU, u η ποσοστιαία χρήση της CPU και β_i^{idle} η αδρανής κατανάλωση ισχύος της i_{η} συχνότητας. Οι παράμετροι β_i^{freq} , β_i^{idle} καθορίζονται δυναμικά από τη προσομοίωση του πειράματος για κάθε επεξεργαστική μονάδα.

$freq(Mhz)$	β_i^{freq}	β_i^{idle}
245.0	201.0	35.1
384.0	257.2	39.5
460.8	286.0	35.2
499.2	303.7	36.5
576.0	332.7	39.5
614.4	356.3	38.5
652.8	378.4	36.7
691.2	400.3	39.6
768.0	443.4	40.2
806.4	470.7	38.4
844.8	493.1	43.5
998.4	559.5	45.6

Πίνακας 3.1: Πίνακας Κατανομής Τιμών β_i^{freq} , β_i^{idle} με βάση τη Συχνότητα (freq) λειτουργίας του επεξεργαστή.

3.4.2 Ενεργειακή Κατανάλωση Ασύρματου Δικτύου WiFi

Σύμφωνα με το άρθρο [33] η απομακρυσμένη μεταφορά μιας εφαρμογής μέσω ασύρματης σύνδεσης WiFi μιας συσκευής με ένα δεδομένο cloudlet μπορεί να ποικίλει στην ενεργειακή κατανάλωση της μπαταρίας του χρήστη εφόσον αυτή μπορεί να γίνει σε διαφορετικές ζώνες εμβέλειας (μονάδα μέτρησης RSSI) του cloudlet. Στόχος μας είναι η μοντελοποίηση της ενεργειακής κατανάλωσης μέσω WiFi του χρήστη που θα βασίζεται εξ' ολοκλήρου στο σήμα εμβέλειας μεταξύ του σημείου πρόσβασης (AP) του cloudlet και της κινητής συσκευής που πρόκειται να στείλει ένα αίτημα εφαρμογής ή να πάρει πίσω κάποιο εκτελεσμένο αρχείο. Στο Σχήμα 3.5 παρουσιάζεται μια προκαθορισμένη διαδικασία καμπύλης της κατανάλωσης ενέργειας με την πάροδο του χρόνου και η αντίστοιχη εξίσωση γραμμικής παλινδρόμησης έχει την ακόλουθη μορφή:

$$E(t) = \beta_1 t + \beta_0 \quad (3.2)$$

όπου $E(t)$ είναι η συνολική κατανάλωση ενέργειας του κινητού εντός της χρονικής περιόδου t και β_0, β_1 είναι δύο παράμετροι του ενεργειακού μοντέλου.

- (i) Προσδιορισμός της τιμής β_0 : Κάτω από το επίπεδο RSSI των -38 dBm, δημιουργήσαμε ένα μοντέλο γραμμικής παλινδρόμησης για την ενέργεια σε συνάρτηση του χρόνου, που παρουσιάζεται στη Στήλη 2, Σειρά 2 στον πίνακα 3.2, χρησιμοποιώντας τις στατιστικές τιμές των 60 μαύρων σημείων που απεικονίζονται στο Σχήμα 3.5. Όπως βρέθηκε από το μοντέλο, η παράμετρος β_0 είναι 1,76.
- (ii) Προσδιορισμός της τιμής β_1 : Για τα άλλα επίπεδα RSSI -40 έως -49 dBm, χρησιμοποιούμε τις ενεργειακές τιμές έναντι του χρόνου που παριστάνονται στο Σχήμα 3.5, μαζί με την τιμή β_0 που είναι 1,76, για να δημιουργήσουμε επτά μοντέλα γραμμικής παλινδρόμησης, όπως φαίνεται από στη Σειρά 3 έως Σειρά 9 της Στήλης 2 στον Πίνακα 3.2.
- (iii) Επικύρωση μοντέλων: Για να αξιολογήσουμε την προβλεπόμενη απόκλιση δεδομένων μοντέλου από τα πειραματικά δεδομένα, χρησιμοποιούμε τις εξής μαθηματικές εξισώσεις που φαίνονται στις Στήλη 3 έως Στήλη 5 του Πίνακα 3.2. Αυτές είναι:

$$* SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (\text{Sum of Squared Errors})$$

$$* Rsquare = 1 - \frac{SSE}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (\text{R-square value})$$

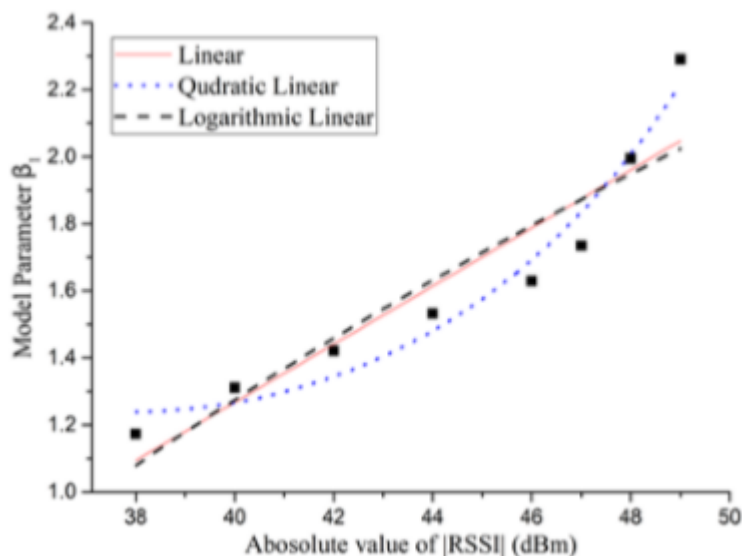
$$* RMSE = \sqrt{\frac{SSE}{v}} \quad (\text{Root Mean Squared Error})$$

Όπου τα y_i, \hat{y}_i και \bar{y} ($0 \leq i \leq 60$) αντιπροσωπεύουν τα πειραματικά δεδομένα, τη πρόβλεψη δεδομένων και τη μέση τιμή πειραματικών δεδομένων αντίστοιχα. Η τιμή v δηλώνει τη διαφορά μεταξύ του αριθμού των πειραματικών δεδομένων και του αριθμού των ρυθμιζόμενων παραμέτρων. Οι τιμές R-square είναι κοντά στη μονάδα και τα αποτελέσματα από τις τρεις συναρτήσεις είναι συνεκτικά μεταξύ τους. Παρατηρείται ότι τα μοντέλα μας είναι σε θέση να προβλέψουν πολύ καλά τα πειραματικά δεδομένα χρησιμοποιώντας τις μεθόδους SSE, R-square και RMSE. Επομένως, κάτω από ένα δεδομένο επίπεδο RSSI, το μοντέλο κατανάλωσης ενέργειας σε συνάρτηση του χρόνου είναι το ακόλουθο:

$$E(t) = \beta_1 t + 1.76 \quad (3.3)$$

Στην εξίσωση 3.3, η τιμή του β_1 εξαρτάται από την τιμή του επιπέδου RSSI, όπως φαίνεται στην Στήλη 1 και 2 στον Πίνακα 3.2.

- (iv) Μοντελοποίηση Ενέργειας: Για τον προσδιορισμό της τιμής της παραμέτρου β_1 χρησιμοποιήσαμε τα μοντέλα της γραμμικής παλινδρόμησης, της τετραγωνικής γραμμικής παλινδρόμησης και της λογαριθμικής γραμμικής παλινδρόμησης όπου οι οκτώ τιμές απόλυτης τιμής |RSSI|: 38, 40, 42, 44, 46, 47, 48 και 49 προέρχονται από τη Στήλη 1 του Πίνακα 3.2 και οι οκτώ αντίστοιχες τιμές β_1 (από 1.173 έως 2.291) προέρχονται από τη στήλη 2 του Πίνακα 3.2, όπως φαίνεται στο Σχήμα 3.4.



Σχήμα 3.4: Ενεργειακό μοντέλο σε συνάρτηση του $|RSSI|$ και της παραμέτρου β_1 .

Από τα τρία αυτά μοντέλα προσέγγισης παρατηρούμε ότι το τετραγωνικό μοντέλο γραμμικής παλινδρόμησης έχει τη μεγαλύτερη τιμή του R-τετραγώνου και οι τιμές SSE και RMSE είναι μικρότερες από τις άλλες δύο μεθόδους. Μπορούμε να πάρουμε την τετραγωνική σχέση ως εξής:

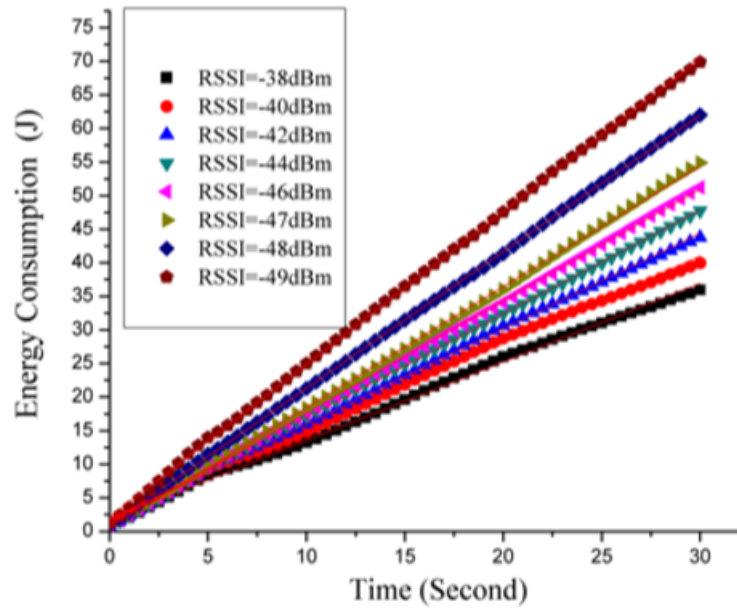
$$\beta_1 = 0.009|RSSI|^2 + 0.7|RSSI| + 14.87 \quad (3.4)$$

Με την αντικατάσταση του β_1 στη σχέση 3.3 με το δεξιό μέρος της σχέσης 3.4, μπορούμε να πάρουμε το **ενεργειακό μοντέλο σε συνάρτηση με το χρόνο και τα επίπεδα RSSI** ως εξής:

$$E(RSSI, t) = (0.009|RSSI|^2 + 0.7|RSSI| + 14.87)t + 1.76 \quad (3.5)$$

<i>RSSI</i>	<i>Energy Model</i>	<i>SSE</i>	<i>R-Square</i>	<i>RMSE</i>
-38 dBm	$E_1(t) = 1.173t + 1.76$	0.56	0.99	0.21
-40 dBm	$E_2(t) = 1.311t + 1.76$	8.78	0.98	0.56
-42 dBm	$E_3(t) = 1.421t + 1.76$	8.36	0.99	0.52
-44 dBm	$E_4(t) = 1.532t + 1.76$	5.54	0.99	0.30
-46 dBm	$E_5(t) = 1.629t + 1.76$	8.75	0.98	0.55
-47dBm	$E_6(t) = 1.735t + 1.76$	8.86	0.98	0.58
-48dBm	$E_7(t) = 1.995t + 1.76$	6.75	0.99	0.42
-49dBm	$E_8(t) = 2.291t + 1.76$	6.94	0.99	0.48

Πίνακας 3.2: RSSI,Ενεργειακό μοντέλο,SSE,R-Square,RMSE.



Σχήμα 3.5: Κατανάλωση ενέργειας σε συνάρτηση με το χρόνο.

3.4.3 Ενεργειακή Κατανάλωση Δικτύου 3G

Σύμφωνα με το άρθρο [14] πρόκειται για την ενέργεια που καταναλώνει μία κινητή συσκευή όταν αποφασίζει να στείλει κώδικα εφαρμογής για απομακρυσμένη εκτέλεση στο Master Cloud. Γνωρίζουμε ότι τέτοιας μορφής δίκτυα, ένα γραμμικό μοντέλο λειτουργεί καλά τόσο για το uplink όσο και για downlink. Αξίζει να σημειωθεί ότι τόσο το uplink όσο και το downlink στο Master Cloud είναι αισθητά μικρότερες ταχύτητες μετάδοσης δεδομένων από αυτές που επικρατούν σε ένα πεδίο ασύρματου δικτύου WiFi ενός cloudlet. Η αξία της απομακρυσμένης εκτέλεσης στο Master Cloud έγκειται στο γεγονός ότι υπερτερεί σε υπολογιστικούς και αποθηκευτικούς πόρους από κάθε κινητό και τοπικό υπολογιστικό νέφος. Ας υποθέσουμε ότι η ταχύτητα uplink είναι t_u (Mbps) και η ταχύτητα downlink είναι t_d (Mbps) τότε το επίπεδο ισχύος για το uplink είναι: $P_u = \alpha_u t_u + \beta$ (mW) και για το downlink είναι: $P_d = \alpha_d t_d + \beta$ (mW). Οι συχνότερες εμφανιζόμενες τιμές για τις παραπάνω παραμέτρους είναι: $\alpha_u = 868.98$ (mW/Mbps), $\alpha_d = 122.12$ (mW/Mbps) και $\beta = 817.88$ (mW). Να σημειωθεί ότι συγκρίνοντας την ενεργειακή κατανάλωση στο WiFi με αυτή στο 3G δίκτυο παρατηρούμε ότι το κινητό καταναλώνει πολύ περισσότερη ενέργεια χρησιμοποιώντας το 3G δίκτυο, κάτι το οποίο φαίνεται και στα πειραματικά αποτελέσματα της προσομοίωσης.

Κεφάλαιο 4

Περιγραφή Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het)

Στην ενότητα αυτή περιγράφεται ο αλγόριθμος δυναμικής δρομολόγησης (Mobi-Het) για τη κατανομή των αιτημάτων εφαρμογών στα cloudlets του συστήματος. Σκοπός του αλγορίθμου είναι να εξασφαλίσει μειωμένο χρόνο απόκρισης στις εφαρμογές, ομοιομορφία του φόρτου εργασίας στους συνολικούς πόρους των cloudlets ενώ ταυτόχρονα να μεγιστοποιήσει τη ποιότητα υπηρεσίας του χρήστη.

4.1 Παρουσίαση Προβλήματος Δυναμικής Δρομολόγησης

Η προτεινόμενη Mobi-Het είναι μια αρχιτεκτονική πολλαπλών επιπέδων που αποτελείται από κινητές συσκευές, τοπικά νέφη και το κύριο σύννεφο που αλληλεπιδρούν μεταξύ τους, όπως φαίνεται στο 3.2 του Κεφ.3. Η βασική ιδέα του Mobi-Het είναι να προγραμματίσει τις απομακρυσμένες αιτήσεις εκτέλεσης κώδικα για εφαρμογές μεγάλων δεδομένων στα διαθέσιμα VMs των cloudlets έτσι ώστε να ελαχιστοποιείται ο συνολικός χρόνος εκτέλεσης και να αυξάνεται η αξιοπιστία εκτέλεσης, ενώ παράλληλα να ελαχιστοποιείται η απόκλιση της κατανομής φορτίου μεταξύ των cloudlets. Το Master Cloud είναι υπεύθυνο για τον πιο πολύπλοκο υπολογισμό ο οποίος είναι η **εύρεση της βέλτιστης αντιστοιχίας αιτημάτων εκτέλεσης κώδικα εφαρμογών των κινητών συσκευών στα cloudlets**. Πρόκειται στην ουσία για ένα πρόβλημα ελαχιστοποίησης το οποίο διατυπώνεται μαθηματικά και επιλύεται. Επιπρόσθετα αναπτύσσουμε τους παράγοντες που επηρεάζουν το πρόβλημα ελαχιστοποίησης που αναφέρθηκε.

4.2 Επίλυση Προβλήματος επιλεκτικής κατανομής πόρων σε εφαρμογές

Το πρόβλημα της επιλογής της βέλτιστης κατανομής υπολογιστικών πόρων για μια ομάδα εφαρμογών των κινητών συσκευών που αιτούνται απομακρυσμένη εκτέλεση είναι δύσκολο λόγω της παρουσίας πολλαπλών παραμέτρων. Η απαίτηση μιας εφαρμογής για ολοκλήρωση εκτέλεσης εντός μεγίστου επιτρεπόμενου χρόνου T_a πρέπει επίσης να ληφθεί υπόψη στο πρόβλημα ελαχιστοποίησης. Επιπλέον, οι πόροι cloudlet είναι ετερογενείς και ένα σύννεφο μπορεί να πάρει διαφορετικά χρονικά διαστήματα κατά την εκτέλεση μιας εφαρμογής. Έστω $t_{c,a}$ υποδηλώνει τον χρόνο εκτέλεσης μιας εφαρμογής a εάν αυτή εκτελείται στο cloudlet $c \in C$. Για την ικανοποίηση της υψηλής ποιότητας υπηρεσίας του χρήστη, ο αλγόριθμος προγραμματισμού πρέπει να προσπαθήσει να κρατήσει το $T_{c,a}$ στο ελάχιστο. Λόγω της κινητικότητας ενός κινητού, μπορεί να υπάρχει ανεπαρκής ισχύς σήματος για να φτάσουμε στο υπολογιστικά προτιμώμενο σύννεφο. Επιπλέον, ο φόρτος εργασίας ενός cloudlet (που δηλώνεται από το W_c) έχει μεγάλη επίδραση στον χρόνο απόκρισης της εκτέλεσης της εφαρμογής, που μπορεί να ελαχιστοποιηθεί διατηρώντας την απόκλιση του φόρτου εργασίας του cloudlet όσο το δυνατόν χαμηλότερα. Ως εκ τούτου, πρέπει να κάνουμε μια ανταλλαγή μεταξύ των παραμέτρων που θα περιγραφούν αναλυτικότερα στη συνέχεια.

Σε αυτή την εργασία μετράμε τον βαθμό της συσχέτισης ($\Psi_{c,a}$) μιας αιτούμενης εφαρμογής για απομακρυσμένη εκτέλεση $a \in A$ από μια συγκεκριμένη κινητή συσκευή σε cloudlet $c \in C$ με τρόπο που να υπολογίζουμε πόσο χρόνο θα παραμείνει συνδεδεμένο σε αυτό. Η αίτηση περιλαμβάνει το αναγνωριστικό της κινητής συσκευής από την οποία αποστέλλεται η αίτηση. Ο υπεύθυνος λήψης αποφάσεων δηλαδή ο Master Cloud προσπαθεί στη συνέχεια να μεγιστοποιήσει το $\Psi_{c,a}$ για την ανάθεση εκτέλεσης κώδικα σε ένα συγκεκριμένο cloudlet. Δεδομένου του συνόλου των αιτημάτων εφαρμογών που έφτασαν κατά το χρονικό διάστημα d από όλες τις ενεργές κινητές συσκευές, το κύριο σύννεφο (Master Cloud) επικαλείται την ακόλουθη αντικειμενική συνάρτηση για να καθορίσει τον βέλτιστο προγραμματισμό ανακατανομής εκτέλεσης κώδικα στα τοπικά υπολογιστικά νέφη (cloudlets).

$$Z = \sum_{c \in C} \sum_{a \in A} \left(\frac{t_{c,a}}{\Psi_{c,a}} + \sigma_c \right) \quad (4.1)$$

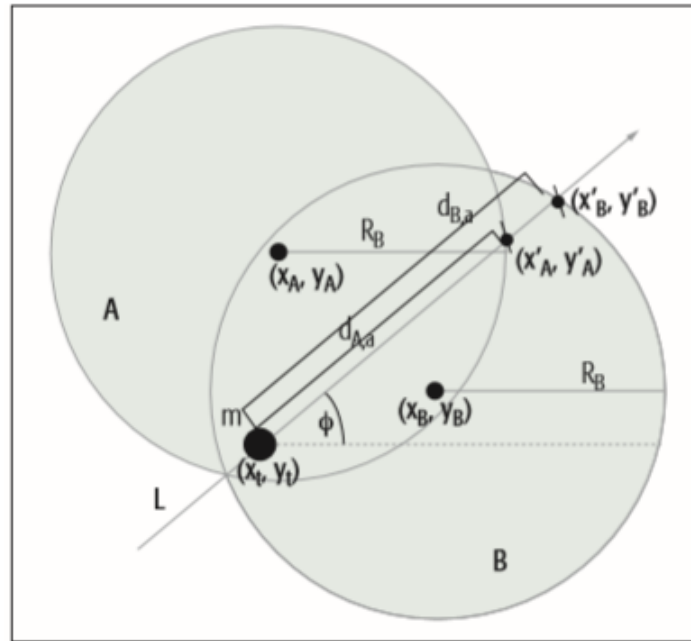
Εδώ, η αντικειμενική συνάρτηση Z διαμορφώνεται ως ένα πρόβλημα MILP που πρέπει να λυθεί από τον Master Cloud. Είναι αρκετά λογικό να υποθέσουμε ότι ο Master Cloud διαθέτει επαρκείς υπολογιστικούς πόρους για την επίλυση του προβλήματος MILP εντός της καθορισμένης προθεσμίας. Στην περίπτωση όπου, ο αριθμός των cloudlets και των αιτήσεων εφαρμογών δεν είναι επίσης πολύ μεγάλος, το πρόβλημα γίνεται λιγότερο περίπλοκο. Υπάρχουν ορισμένοι περιορισμοί της αντικειμενικής συνάρτησης Z που πρέπει να ικανοποιηθούν. Για παράδειγμα:

- Ο περιορισμός της γρήγορης εξυπηρέτησης του χρήστη (QoS) $t_{c,a} \leq T_a, c, \forall c \in C, \forall a \in A$ δηλώνει ότι ο χρόνος για την απομακρυσμένη εκτέλεση μιας εφαρμογής πρέπει να είναι μικρότερος από τον μέγιστο επιτρεπόμενο χρόνο.

- Ο περιορισμός κατανομής φόρτου εργασίας $\mu_c - \sigma_c \leq W_c \leq \mu_c + \sigma_c$ δηλώνει ότι μετά την αντιστοίχιση μιας εφαρμογής σε ένα cloudlet $c \in C$ για απομακρυσμένη εκτέλεση το φορτίο εργασίας W_c του cloudlet δεν πρέπει να αποκλίνει περισσότερο από το σ_c από το μέσο όρο εργασίας μ_c σε όλο το σύστημα, επιτυγχάνοντας έτσι την ισορροπημένη κατανομή φόρτου εργασίας σε ένα περιβάλλον υπολογιστικού νέφους υπολογιστών.
- Ο περιορισμός χωρητικότητας $W_c \leq W_c^{max}$, $\forall c \in C$ δηλώνει ότι ο συνολικός φόρτος εργασίας ενός cloudlet δεν πρέπει να υπερβαίνει τη μέγιστη υπολογιστική του ικανότητα W_c^{max} .
- Ο περιορισμός $RSSI_c \geq \Gamma_{RSSI}$, $\forall c \in C$ δηλώνει ότι η ένδειξη της ληφθείσας έντασης σήματος (RSSI) κάθε κινητής συσκευής που βρίσκεται εντός της περιοχής ενός cloudlet πρέπει να είναι μεγαλύτερη από ένα προκαθορισμένο όριο διαφορετικά, δεν θα ήταν δυνατό για την κινητή συσκευή να μεταφέρει και να λαμβάνει κώδικα και αποτελέσματα αντίστοιχα, εντός της καθορισμένης προθεσμίας.

4.2.1 Προσδιορισμός Βαθμού Συσχέτισης

Ο βαθμός συσχέτισης (Ψ_c) είναι η προβλεπόμενη χρονική διάρκεια κατά την οποία μια κινητή συσκευή που στέλνει μία εφαρμογή $a \in A$ θα παραμείνει κάτω από ένα cloudlet $c \in C$. Σκοπός μας είναι η μεγιστοποίηση αυτής της παραμέτρου το οποίο μας δίνει και υψηλότερη ποιότητα υπηρεσίας στους τελικούς χρήστες (QoS). Για τη μέτρηση του Ψ_c , πρέπει να εξετάσουμε τα μοτίβα κινητικότητας κάθε κινητής συσκευής. Η αρχιτεκτονική Mobi-Het επιτρέπει σε έναν χρήστη της κινητής συσκευής να κινείται με διαφορετικές ταχύτητες και σε τυχαίες κατευθύνσεις. Επομένως, η βασική πρόκληση του ακριβούς καθορισμού του βαθμού της συσχέτισης είναι στην ουσία ο υπολογισμός μιας ακριβούς θέσης τομής του κινητού με τη μέγιστη ζώνη εμβέλειας σήματος του AP του cloudlet. Στο Σχήμα 4.1 απεικονίζεται ένα παράδειγμα προσδιορισμού του Ψ_c . Εδώ, τα A και B είναι δύο συν-τοποθετημένα cloudlets με ακτίνα (περιοχή μετάδοσης) $R_A = R_B$ και γεωγραφική θέση (x_A, y_A) και (x_B, y_B) , αντίστοιχα. Μια κινητή συσκευή m με θέση (x_t, y_t) βρίσκεται εντός της περιοχής και των δύο cloudlets και η αναμενόμενη κατεύθυνση της είναι ϕ . Τώρα, χρησιμοποιώντας το μοντέλο δυναμικής τυχαίας κινητικότητας που αναφέρθηκε και τις τιμές των (x_A, y_A) , R_A , (x_B, y_B) και ϕ , μπορούμε να υπολογίσουμε το σημείο τομής του cloudlet A με τη κινητή συσκευή (x'_A, y'_A) . Με τον ίδιο τρόπο, μπορούμε να υπολογίσουμε και το σημείο τομής του cloudlet B με τη κινητή συσκευή (x'_B, y'_B) . Τώρα, αν η απόσταση μεταξύ (x_t, y_t) και (x'_A, y'_A) είναι d_A και η απόσταση μεταξύ (x_t, y_t) και (x'_B, y'_B) είναι d_B και η ταχύτητα του κινητού είναι v m/s, μπορούμε να υπολογίσουμε το $Y_A = d_A/v$ και το $Y_B = d_B/v$.



Σχήμα 4.1: Προσδιορισμός Του Βαθμού Συσχέτισης.

4.2.2 Υπολογισμός Τυπικής Απόκλισης Συστήματος

Όπως αναφέρθηκε προηγουμένως το Mobi-Het είναι μια ετερογενής αρχιτεκτονική που διαχειρίζεται αιτήματα εφαρμογών για απομακρυσμένη εκτέλεση κώδικα με ποικίλες απαιτήσεις μεγέθους και προθεσμίας. Οι υπολογιστικοί πόροι που διατίθενται στα τοπικά υπολογιστικά νέφη μπορεί επίσης να διαφέρουν σημαντικά μεταξύ τους και ως εκ τούτου μπορεί να θεωρηθεί ότι έχουν ετερογενείς συνολικές υπολογιστικές ικανότητες. Κάθε cloudlet $c \in C$ υποστηρίζει ένα σύνολο από VMs , που δηλώνεται ως V_c , και κάθε τύπος VM διαθέτει διαφορετικό ποσοστό χρησιμοποίησης. Στο πείραμα που ακολουθεί ορίζουμε τη μέγιστη υπολογιστική ισχύ ζ_i για κάθε VM τύπου $i \in V_c$ και ανά πάσα στιγμή μπορούμε να γνωρίζουμε το ποσοστό χρησιμοποίησης (utilization Υ_{vm}^i) σε οποιαδήποτε χρονική στιγμή της προσομοίωσης. Η συνολική μέγιστη χωρητικότητα υπολογισμού (W_c^{max}) ενός cloudlet $c \in C$ θα είναι $\Sigma(\zeta_i \times 100\%)$ για όλα τα $i \in V_c$. Ωστόσο σε ένα συγκεκριμένο χρονικό διάστημα προγραμματισμού, μόνο ένα μέρος των πηγών cloudlet θα είναι απασχολημένο και αυτό καθορίζει το τρέχων φορτίο εργασίας. Επομένως για τον τρέχοντα αριθμό VMs που χρησιμοποιούνται για ένα σύννεφο $c \in C$, η ποσότητα των πόρων που καταναλώνονται επί του παρόντος, W'_c , θα είναι $\Sigma(\zeta_i \times \Upsilon_{vm}^i)$ για όλα για όλα $i \in V_c$. Τώρα, ο τρέχων φόρτος εργασίας, W_c ενός cloudlet $c \in C$ θα είναι ο λόγος W'_c με τη συνολική μέγιστη χωρητικότητα υπολογισμού W_c^{max} . Επομένως, η τυπική απόκλιση (σ) για το φόρτο εργασίας όλων των cloudlets υπολογίζεται όπως στην Εξίσωση 4.2 και ποσοτικοποιεί την ποσότητα της μεταβολής στις μέσες τιμές του φόρτου εργασίας του συστήματος.

$$\sigma = \sqrt{\frac{1}{|c|} \sum_{c=1}^{|c|} \left(\frac{W'_c}{W_c^{max}} - \mu \right)^2} \quad (4.2)$$

,όπου μ είναι ο μέσος φόρτος εργασίας όλων των τοπικών υπολογιστικών νεφών. Η χαμηλή τιμή σ υποδεικνύει ότι η κατανομή του φόρτου εργασίας τείνει να είναι πολύ κοντά στον μέσο όρο, ενώ μια υψηλή τιμή σ υποδηλώνει ότι η κατανομή του φόρτου εργασίας κατανέμεται σε ένα ευρύτερο εύρος τιμών. Καθώς τα ετερογενή cloudlets έχουν διαφορετικούς υπολογιστικούς πόρους, διαφορετικά cloudlets επιφέρουν διαφορετικές μεταβολές στη τιμή της εξίσωσης 4.2 όταν τα αιτήματα εφαρμογών κατανέμονται σε αυτά. Μια προσέγγιση φόρτου εργασίας χωρίς προγραμματισμό μπορεί να αυξήσει σημαντικά την τυπική απόκλιση το οποίο θα φανεί και από τα πειραματικά αποτελέσματα που θα ακολουθήσουν, με αποτέλεσμα πολλές αιτήσεις εφαρμογών να απορρίπτονται. Στόχος μας είναι να ελαχιστοποιήσουμε όσο το δυνατόν περισσότερο την τυπική απόκλιση των φόρτου εργασίας του cloudlet σε σχέση με το μέσο όρο φόρτου εργασίας του συστήματος, όπως αναφέρεται στην αντικειμενική συνάρτηση 4.1, αλλά η λειτουργία βελτιστοποίησης δεσμεύεται αναγκαστικά από ένα συμβιβασμό μεταξύ του χρόνου εκτέλεσης εργασίας, της συσχέτισης και της κατανομής του φορτίου.

4.2.3 Υπολογισμός Χρόνου Εκτέλεσης Εφαρμογής

Ο χρόνος εκτέλεσης μιας εφαρμογής σε ένα τοπικό υπολογιστικό νέφος περιλαμβάνει τόσο τους χρόνους μετάδοσης και λήψης του κώδικα μαζί με τον πραγματικό χρόνο υπολογισμού. Πιο συγκεκριμένα, ορισμένες εφαρμογές επιρρεπείς σε μεγάλο όγκο δεδομένων ενδέχεται να χρειαστούν αρκετό χρονικό διάστημα για να μεταφερθούν στο cloudlet. Ίδιο φαινόμενο παρατηρείται ανάλογα με το μέγεθος των αποτελεσμάτων και στη διαδικασία λήψης της κινητής συσκευής. Ωστόσο υπάρχουν και εφαρμογές που μπορεί να χρειαστούν να μεταφέρουν μερικές μόνο δεκάδες bytes κώδικα στο cloudlet και να απαιτούν μεγάλης έντασης εργασίας. Ως αποτέλεσμα, ο συνολικός χρόνος απόκρισης ενδέχεται να ποικίλει σημαντικά ανάλογα με το μέγεθος των δεδομένων εφαρμογής καθώς και την ποιότητα της σύνδεσης μεταξύ της κινητής συσκευής και του AP του cloudlet. Ο συνολικός χρόνος για την απομακρυσμένη εκτέλεση ($t_{c,a}$) περιλαμβάνει το χρονικό διάστημα αποστολής των αρχικών δεδομένων και του εκτελέσιμου κώδικα ($t_{c,a}^s$), του χρόνου εκτέλεσης ($t_{c,a}^e$) του κώδικα και του χρόνου λήψης του αρχείου αποτελέσματος ($t_{c,a}^r$).

Οι τιμές των $t_{c,a}^s$ και $t_{c,a}^r$ πέρα από το μέγεθος της αποστολής και της λήψης κώδικα όπως εξηγήθηκε εξαρτώνται σε μεγάλο βαθμό και από την τιμή RSSI που λαμβάνεται από τη κινητή συσκευή και από ένα AP ενός cloudlet $c \in C$. Όταν η τιμή RSSI είναι χαμηλή, ο ρυθμός μετάδοσης δεδομένων είναι μικρός, δηλαδή η σύνδεση είναι κακή και επομένως ο χρόνος για την αποστολή των αρχικών δεδομένων της εφαρμογής είναι υψηλός και αντίστροφα. Το ίδιο ισχύει και στη διαδικασία λήψης. Ένας πίνακας όπως ο Πίνακας 4.1 μπορεί να χρησιμοποιηθεί για τη χαρτογράφηση των τιμών RSSI στις αντίστοιχες ταχύτητες δεδομένων και μπορεί να χρησιμοποιηθεί ως αναζήτηση για την εκτίμηση του χρόνου μεταφοράς δεδομένων.

Ορίζουμε το S_a και το S_a' ως την ποσότητα των *bits* δεδομένων που πρέπει να αποστέλλονται και να λαμβάνονται από μια κινητή συσκευή αντίστοιχα για την εκτέλεση μιας εφαρμογής μεγάλων δεδομένων $a \in A$. Επίσης ορίζουμε το j_a ως τον αριθμό

των εντολών του κώδικα της εφαρμογής προς εκτέλεση $a \in A$, και $D_{c,a}$ ορίζουμε τον τρέχων ρυθμό μετάδοσης δεδομένων που λαμβάνει η κινητή συσκευή από το AP που είναι προσαρτημένο στο cloudlet $c \in C$. Έτσι μπορούμε να υπολογίσουμε τον συνολικό χρόνο απομακρυσμένης εκτέλεσης ($t_{c,a}$) ως εξής:

$$t_{c,a} = \frac{s_a}{D_{c,a}} + \frac{j_a}{MIPS_i} + \frac{s'_a}{D_{c,a}} \quad \forall c \in C, \forall a \in A, \forall i \in V_c \quad (4.3)$$

Καθώς τα cloudlets κοντά στην περιοχή του κινητού είναι ετερογενή στις δυνάμεις των σημάτων τους, ο χρόνος εκτέλεσης μιας δεδομένης εφαρμογής μπορεί να διαφέρει πολύ όταν εκτελείται σε διαφορετικά cloudlets. Στόχος μας είναι να ελαχιστοποιήσουμε όσο το δυνατόν περισσότερο τον χρόνο εκτέλεσης κάθε εξ αποστάσεως εφαρμογής, όπως αναφέρεται στην αντικειμενική συνάρτηση (Εξίσωση 4.1). Όπως αναφέρθηκε, η λειτουργία βελτιστοποίησης παρουσιάζει αντισταθμίσεις μεταξύ του χρόνου εκτέλεσης εργασίας, της συσχέτισης και της κατανομής του φορτίου.

<i>RSSI(dBm)</i>	<i>Data rate(Mb/s)</i>
-81	1
-79	2
-77	6
-75	11
-73	24
-69	36
-65	48
-64	54

Πίνακας 4.1: Αντίστοιχη RSSI με τον ρυθμό μετάδοσης.

4.3 Γενική Σύνοψη Αλγορίθμου

Η βασική ιδέα του Mobi-Het είναι να προγραμματίζει τα αιτήματα εφαρμογών για απομακρυσμένη εκτέλεση κώδικα στα διαθέσιμα ετερογενή τοπικά υπολογιστικά νέφη (cloudlets) και να επιτρέπει στις κινητές συσκευές να εκτελούν εφαρμογές μεγάλων δεδομένων ενώ απολαμβάνουν την κινητικότητα τους στα πλαίσια της "έξυπνης" πόλης. Μια προσέγγιση προγραμματισμού φόρτου εργασίας χωρίς επίγνωση όλων των απαραίτητων παραμέτρων που προαναφέρθηκαν μπορεί να αυξήσει σημαντικά την τυπική απόκλιση, με αποτέλεσμα τη συνολική αύξηση του χρόνου αναμονής του χρήστη. Στόχος μας είναι να ελαχιστοποιήσουμε όσο το δυνατόν περισσότερο την τυπική απόκλιση του φόρτου εργασίας και του χρόνου απόκρισης των εργασιών στα cloudlets ενώ παράλληλα να μεγιστοποιήσουμε τη ποιότητα υπηρεσίας του χρήστη (Εξίσωση 4.1). Ωστόσο η λειτουργία βελτιστοποίησης δεσμεύεται αναγκαστικά από μία ισορροπία μεταξύ του χρόνου εκτέλεσης εργασίας, της συσχέτισης και της κατανομής του φορτίου.

Κεφάλαιο 5

Πειραματική Αξιολόγηση Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het)

Στο κεφάλαιο αυτό θα παρουσιαστούν οι τεχνικές λεπτομέρειες και η πειραματική αξιολόγηση του αλγορίθμου που περιγράφηκε.

5.1 Λεπτομέρειες Υλοποίησης

Όπως αναλύθηκε και στο Κεφ. 3 τα τρία βασικά αρχιτεκτονικά στοιχεία από τα οποία θα ξεκινήσουμε την προσομοίωση είναι τα εξής: *Κινητές Συσκευές (Mobile Devices)*, *Τοπικό Υπολογιστικό Νέφος (Cloudlets)* και *Απομακρυσμένο Υπολογιστικό Νέφος (Master Cloud)*. Η υλοποίηση της κάθε μία κλάσης ξεχωριστά στο εργαλείο προσομοίωσης του CloudSim Plus παρουσιάζεται παρακάτω.

1. **Κινητές Συσκευές (Mobile Devices):** Κάθε κινητό ακολουθεί ένα τυχαίο αλγόριθμο μονοπατιού (τυχαία αλλαγή μέτρου και κατεύθυνσης ταχύτητας ανά 2 sec) το οποίο εκφράζεται με μία μέση τιμή που λέγεται **Κινητικότητα (Mobility)** μέσα στα πλαίσια ενός πεδίου προσομοίωσης (grid). Οι υπολογιστικοί και αποθηκευτικοί πόροι είναι μικροί σε σχέση με τις απαιτήσεις των εφαρμογών (tasks), οι οποίες δημιουργούνται ανά τακτικά και ίσα χρονικά διαστήματα. Για κάθε κινητή συσκευή κατασκευάσαμε τα ενεργειακά μοντέλα της CPU, WiFi και 3G που περιγράφηκαν στο Κεφ.3 και μετράμε τη συνολική δαπάνη ενέργειας (μέθοδος δειγματοληψίας σε παράθυρο χρόνου 1sec, 2sec, 2sec αντίστοιχα). Επίσης έχουμε δημιουργήσει γραφικές παραστάσεις που προσομοιάζουν τη μπαταρία και το μέσο ρυθμό εκτέλεσης της κινητής συσκευής (μέθοδος δειγματοληψίας σε παράθυρο χρόνου 10 sec). Γνωρίζουμε από το Κεφ.3 ότι η ενέργεια της κινητής συσκευής καταναλίσκεται κατά φθίνουσα σειρά στα στοιχεία CPU, 3G, WiFi και επομένως για να προσομοιώσουμε τη μπαταρία του χρήστη κάνουμε τις εξής δικές μας παραδοχές :

i Εκτέλεση Εφαρμογής Στη Κινητή Συσκευή Με Κόστος 14mAh.

- ii Αποστολή/Λήψη Μέσω Δικτύου WiFi Με Κόστος 1mAh.
- iii Αποστολή/Λήψη Μέσω Δικτύου 3G Με Κόστος 4mAh.
- iv Περίοδος Άεργης Κατάστασης (idle time) Κινητής Συσκευής Με Κόστος 1mAh Ανά Χρονικό Διάστημα 2sec (Interval Time=2sec)

Τα κόστη αυτά στη μπαταρία του χρήστη παρατηρούμε ότι είναι ανάλογα της ενεργειακής κατανάλωσης στα τρία παθητικά στοιχεία CPU, 3G, WiFi σύμφωνα με τα ενεργειακά μοντέλα που περιγράψαμε στη θεωρία στο Κεφάλαιο 3. Η προσέγγιση αυτή αποτελεί δική μας παραδοχή και προσομοιώνει στην πραγματικότητα τον ρυθμό μείωσης της συνολικής διαθέσιμης ενέργειας της μπαταρίας του χρήστη καθ' όλη τη διάρκεια του πειράματός μας.

Χαρακτηριστικά Κινητής Συσκευής: Αριθμός Φυσικών Μηχανημάτων (Number of Hosts): 1, Αριθμός Στοιχείων Επεξεργασίας Του Κάθε Φυσικού Μηχανήματος (Number of Host Pes): 2, Αριθμός Εκτέλεσης Εντολών Ανά Δευτερόλεπτο Του Κάθε Στοιχείου Επεξεργασίας Του Κάθε Φυσικού Μηχανήματος (Mips of Host Pe): 800MIPS, Μνήμη Του Φυσικού Μηχανήματος (Ram of Host): 1024MB, Αριθμός Εικονικών Μηχανημάτων (Number of Vms): 2, Αριθμός Στοιχείων Επεξεργασίας Του Κάθε Εικονικού Μηχανήματος (Number of Vm Pes): 2, Αριθμός Εκτέλεσης Εντολών Ανά Δευτερόλεπτο Του Κάθε Στοιχείου Επεξεργασίας Του Κάθε Εικονικού Μηχανήματος (Mips of Vm Pe): 400MIPS, Μέγεθος Εικονικού Μηχανήματος (Vm Size): 10000MB, Μνήμη Του Εικονικού Μηχανήματος (Ram of Vm): 256MB, Προγραμματιστής Εκτέλεσης Εικονικών Μηχανημάτων (VmScheduler): VmAllocationPolicySimple, Προγραμματιστής Εκτέλεσης Εργασιών (CloudletScheduler): CloudletSchedulerSpaceShared, Χωρητικότητα Σκληρού Δίσκου (storage): 500000MB, Αρχική Μπαταρία (Start Battery): 3000mAh.

Χαρακτηριστικά Εφαρμογής (task): Απαιτούμενος Αριθμός Φυσικών Μηχανημάτων Για Εκτέλεση (Number of Pes): 2, Μέγεθος Εφαρμογής (length): (3000, 5000)MB, Μέγεθος Αρχείου Εξόδου (fileSize outputSize): (300, 400)MB, Μέθοδος Χρησιμοποίησης Πόρων Μηχανήματος (utilizationModel): UtilizationModelFull,

Δημιουργία Εφαρμογής Για Κάθε Κινητή Συσκευή Ανά 5sec (Creation interval for each mobile): 5 sec.

2. **Τοπικό Υπολογιστικό Νέφος (Cloudlets):** Κάθε Τοπικό υπολογιστικό νέφος προσομοιώνεται με ένα datacenter με κύκλο ο οποίος έχει σταθερό και αμετάβλητο στο χρόνο κέντρο το οποίο καθορίζεται στην αρχή του πειράματος. Όπως κάθε cloudlet διαφέρει σε υπολογιστικούς/αποθηκευτικούς πόρους το ένα με το άλλο έτσι μπορεί να διαφέρει και στην εμβέλεια σήματος WiFi για κάθε προσαρτημένο σε αυτό AP. Για να το δείξουμε αυτό στη προσομοίωση χρησιμοποιούμε έναν παράγοντα *radius scale* το οποίο προσαρμόζει ξεχωριστά την ακτίνα εμβέλειας του πεδίου WiFi του κάθε cloudlet. Όσο μεγαλύτερος είναι αυτός ο παράγοντας τόσο μεγαλύτερη θα είναι και η απόκλιση μεταξύ των πεδίων που εκπέμπουν τα AP των cloudlets. Στο κέντρο του κύκλου βρίσκεται το AP με το οποίο συνδέονται οι χρήστες και η ακτίνα του

κύκλου υποδιαιρείται τόσες φορές ώστε σε κάθε μία ζώνη που δημιουργείται να αντιστοιχούν οι τιμές Data Rate($Mb \setminus s$) και RSSI(dBm) σύμφωνα με τον Πίνακα 4.1 του Κεφ.4. Η κύρια εργασία του cloudlet είναι να λαμβάνει κώδικες εφαρμογών κινητών συσκευών, εφόσον έχει διαθέσιμους πόρους (ειδάλλως απορρίπτονται) μέσω σύνδεσης ασύρματου δικτύου WiFi, να τους εκτελεί και να στέλνει πίσω στους χρήστες τα τελικά αποτελέσματα πάλι μέσω δικτύου WiFi. Η επιλογή του βέλτιστου cloudlet σε περίπτωση αλληλοκάλυψης των πεδίων είναι δουλειά του Master Cloud οποίος και είναι υπεύθυνος για τη συνολική ελαχιστοποίηση της αναμονής των χρηστών του συστήματος σύμφωνα με τον αλγόριθμο δυναμικής δρομολόγησης(Mobi-Het) του Κεφ.4. Όπως και για τις κινητές συσκευές έτσι και εδώ έχουμε δημιουργήσει γραφικές παραστάσεις που προσομοιάζουν το μέσο ρυθμό εκτέλεσης του cloudlet (μέθοδος δειγματοληψίας σε παράθυρο χρόνου 10 sec). Η ενεργειακή κατανάλωση σε τοπικά και απομακρυσμένα υπολογιστικά νέφη δε προσομοιάζεται στα πλαίσια αυτής της διπλωματικής εργασίας.

Χαρακτηριστικά Cloudlet: Αριθμός Φυσικών Μηχανημάτων (Number of Hosts): 4, Αριθμός Στοιχείων Επεξεργασίας Του Κάθε Φυσικού Μηχανήματος (Number of Host Pes): 2, Αριθμός Εκτέλεσης Εντολών Ανά Δευτερόλεπτο Του Κάθε Στοιχείου Επεξεργασίας Του Κάθε Φυσικού Μηχανήματος (Mips of Host Pe): $(800 + \text{Cloudlet_id} \times 400)$ MIPS, Μνήμη Του Φυσικού Μηχανήματος (Ram of Host): 2048MB, Αριθμός Εικονικών Μηχανημάτων (Number of Vms): 4, Αριθμός Στοιχείων Επεξεργασίας Του Κάθε Εικονικού Μηχανήματος (Number of Vm Pes): 2, Αριθμός Εκτέλεσης Εντολών Ανά Δευτερόλεπτο Του Κάθε Στοιχείου Επεξεργασίας Του Κάθε Εικονικού Μηχανήματος (Mips of Vm): $(800 + \text{Cloudlet_id} \times 400)$ MIPS, Μέγεθος Εικονικού Μηχανήματος (Vm Size): 10000MB, Μνήμη Του Εικονικού Μηχανήματος (Ram of Vm): 1024MB, Προγραμματιστής Εκτέλεσης Εικονικών Μηχανημάτων (VmScheduler): VmAllocationPolicySimple, Προγραμματιστής Εκτέλεσης Εργασιών (CloudletScheduler): CloudletSchedulerSpaceShared, (Ιδιότητα Κλιμάκωσης Υπολογιστικών Πόρων Εικονικών Μηχανημάτων Σε Συνθήκες Υπερφόρτωσης) Vertical VM Scaling, Χωρητικότητα Σκληρού Δίσκου(storage): 1000000MB.

3. **Απομακρυσμένο Υπολογιστικό Νέφος (Master Cloud):** Στα πλαίσια αυτής της διπλωματικής εργασίας χρησιμοποιούμε ένα Master Cloud το οποίο η θέση του είναι εκτός grid πεδίου προσομοίωσης. Το Master Cloud προσομοιώνεται ως ένα datacenter που έχει πολύ υψηλούς πόρους επεξεργασίας και αποθήκευσης και είναι υπεύθυνο για την εκτέλεση του αλγορίθμου δυναμικής δρομολόγησης των εργασιών **όλων των κινητών συσκευών του συστήματος** στα κατάλληλα cloudlets με στόχο τη συνολική ελαχιστοποίηση της απόκρισης. Επιπρόσθετα μπορεί να δέχεται αιτήματα εφαρμογών μεγάλων δεδομένων μέσω σύνδεσης δικτύου 3G, τα οποία είτε δε μπορεί να εκτελέσει το cloudlet (κορεσμός πόρων του cloudlet) είτε είναι πολύ χρονοβόρο για το ίδιο το κινητό να τα εκτελέσει. Για την αποδοτικότερη χρησιμοποίηση όλων των πόρων θεωρήσαμε ένα κατώφλι στη μπαταρία του χρήστη, κάτω από το οποίο όταν το κινητό έχει εφαρμογή προς εκτέλεση και δεν βρίσκεται στην εμβέλεια

κάποιου cloudlet, τότε το Master Cloud αναλαμβάνει την επεξεργασία της εφαρμογής ανεξαρτήτως μεγέθους. Η αποστολή και η λήψη γίνονται σύμφωνα με τις ταχύτητες uplink και downlink του μοντέλου της Ενεργειακής Κατανάλωσης 3G που περιγράφηκε στο Κεφ.3. Επίσης έχουμε και εδώ γραφικές παραστάσεις όπως και στις προηγούμενες περιπτώσεις για το μέσο ρυθμό εκτέλεσης.

Χαρακτηριστικά Master Cloud: Αριθμός Φυσικών Μηχανημάτων (Number of Hosts): 4, Αριθμός Στοιχείων Επεξεργασίας Του Κάθε Φυσικού Μηχανήματος (Number of Host Pes): 2, Αριθμός Εκτέλεσης Εντολών Ανά Δευτερόλεπτο Του Κάθε Στοιχείου Επεξεργασίας Του Κάθε Φυσικού Μηχανήματος (Mips of Host Pe): 2000MIPS, Μνήμη Του Φυσικού Μηχανήματος (Ram of Host): 4096MB, Αριθμός Εικονικών Μηχανημάτων (Number of Vms): 4, Αριθμός Στοιχείων Επεξεργασίας Του Κάθε Εικονικού Μηχανήματος (Number of Vm Pes): 2, Αριθμός Εκτέλεσης Εντολών Ανά Δευτερόλεπτο Του Κάθε Στοιχείου Επεξεργασίας Του Κάθε Εικονικού Μηχανήματος (Mips of Vm): 2000MIPS, Μέγεθος Εικονικού Μηχανήματος (Vm Size): 10000MB, Μνήμη Του Εικονικού Μηχανήματος (Ram of Vm): 4096MB, Προγραμματιστής Εκτέλεσης Εικονικών Μηχανημάτων (VmScheduler): VmAllocationPolicySimple, Προγραμματιστής Εκτέλεσης Εργασιών (CloudletScheduler): CloudletSchedulerSpaceShared, (Ιδιότητα Κλιμακωτής Υπολογιστικών Πόρων Εικονικών Μηχανημάτων Σε Συνθήκες Υπερφόρτωσης) Vertical VM Scaling, Χωρητικότητα Σκληρού Δίσκου (storage): 2000000MB.

Ο χρόνος προσομοίωσης παραμένει σταθερός στα 30 λεπτά για τα Πειράματα Β, Γ, Δ και 60 λεπτά για το Πείραμα Α. Οι αρχικές θέσεις των κινητών καθορίζονται τυχαία στην αρχή του πειράματος. Όλα τα παρακάτω πειράματα διεξάχθηκαν σε προσωπικό υπολογιστή με επεξεργαστή Intel (R) Core (TM) i5-6600K CPU @ 3.50GHz με μνήμη 8.00 GB και x64-based processor.

5.2 Πειραματική Αξιολόγηση

Οι επόμενες υποενότητες είναι χωρισμένες ως εξής:

Στην υποενότητα 5.2.1 κάνουμε μία εισαγωγή στο τρόπο εκτέλεσης του πειράματος επικεντρώνοντας τη προσοχή μας στην ενεργειακή κατανάλωση του χρήστη, στο μέσο χρόνο εκτέλεσης εφαρμογών του καθώς και στη διαδρομή που ακολούθησε μέσα στο πεδίο της προσομοίωσης των cloudlets, χρησιμοποιώντας τη μέθοδο της Δυναμικής Δρομολόγησης (Mobi-Het).

Στην υποενότητα 5.2.2 συγκρίνουμε τα πειραματικά αποτελέσματα του αλγορίθμου δυναμικής δρομολόγησης (Mobi-Het) με αυτά ενός τυχαίου μοντέλου δυναμικής δρομολόγησης μέσω της κλιμακωτής αλλαγής της ακτίνας κάθε cloudlet (μεταβλητός παράγοντας: Radius Scale Cloudlets).

Στην υποενότητα 5.2.3 συγκρίνουμε τα πειραματικά αποτελέσματα του αλγορίθμου δυναμικής δρομολόγησης (Mobi-Het) με αυτά ενός τυχαίου μοντέλου δυναμικής δρομολόγησης μέσω της κλιμακωτής αλλαγής του αριθμού των κινητών συσκευών που

υπάρχουν στο σύστημα (μεταβλητός παράγοντας: Number of Devices).

Στην υποενότητα 5.2.4 συγκρίνουμε τα πειραματικά αποτελέσματα του αλγορίθμου δυναμικής δρομολόγησης (Mobi-Het) με αυτά ενός τυχαίου μοντέλου δυναμικής δρομολόγησης μέσω της κλιμακωτής αλλαγής του βαθμού της μέσης κινητικότητας των κινητών συσκευών που υπάρχουν στο σύστημα (μεταβλητός παράγοντας: Average Speed).

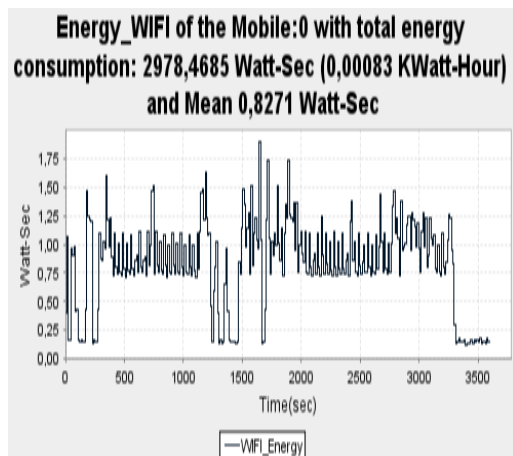
5.2.1 Πείραμα Α:Επισκόπηση Αποτελεσμάτων Κινητής Συσκευής και Λειτουργία Μοντέλου Δυναμικής Δρομολόγησης (Mobi-Het)

Στο πείραμα αυτό θα εξετάσουμε τη κίνηση του χρήστη μέσα στο πεδίο grid των cloudlets, θα μελετήσουμε την ενεργειακή του κατανάλωση (Κόστος χρησιμοποίησης CPU, WiFi, 3G, Γράφημα μπαταρίας), το μέσο χρόνο εκτέλεσης εργασιών καθώς και τα οφέλη της Δυναμικής Δρομολόγησης (Mobi-Het).

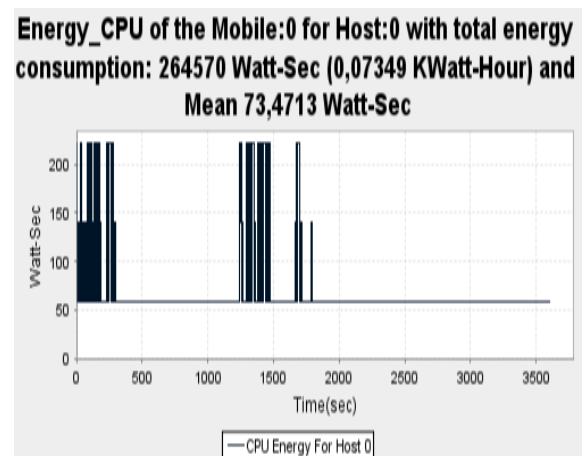
Για τη καλύτερη κατανόηση της προσομοίωσης θα χρησιμοποιήσουμε μία κινητή συσκευή στο σύστημα εφόσον ανάλογα πειραματικά αποτελέσματα ισχύουν και για περισσότερες συσκευές. Επομένως για αρχή του πειράματος θα ακολουθήσουμε για τις σταθερές παραμέτρους προσομοίωσης:

- Αριθμός Κινητών Συσκευών (num of mobiles) = 1
- Κινητικότητα Κινητής Συσκευής (rangeMin, rangeMax)=(4.0, 6.0) $m \setminus s$
- Radius Scale = 100

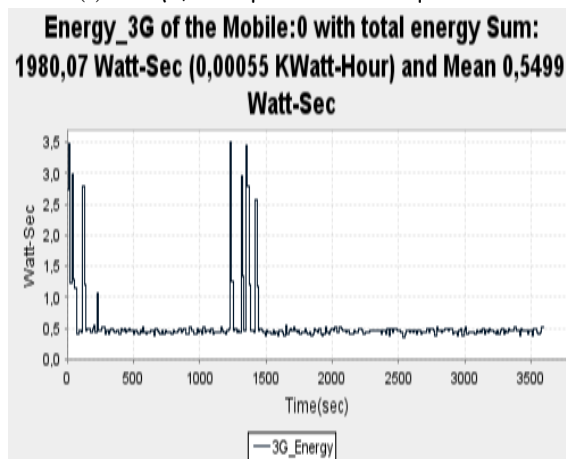
Όπως φαίνεται και από το Σχήμα 5.2 παρατηρούμε ότι υπάρχει αλληλοκάλυψη μεταξύ των cloudlet οπότε και υφίσταται η σημασία της δυναμικής δρομολόγησης εφαρμογής κινητής συσκευής σε cloudlet. Εφαρμόζοντας τα ενεργειακά μοντέλα για CPU, WiFi και 3G στην αρχιτεκτονική δυναμικής δρομολόγησης Mobi-Het που δημιουργήσαμε παίρνουμε τις αντίστοιχες γραφικές παραστάσεις κατανάλωσης ενέργειας όπως φαίνεται στα Σχήματα 5.1. Η συνολική κατανάλωση στη μπαταρία του χρήστη σύμφωνα με το πρότυπο που χρησιμοποιήσαμε παρατίθεται στο Σχήμα 5.1iv. Όσον αναφορά το μέσο χρόνο απόκρισης εργασιών του χρήστη και όλων των υποβληθέντων εργασιών στα cloudlets του συστήματος μέσω του αλγορίθμου Mobi-Het έχουμε τα Σχήματα: 5.3i, 5.3ii, 5.3iii, 5.3iv, 5.3v, 5.3vi. Από τη μελέτη των σχημάτων παρατηρούμε σαν επιπρόσθετη πληροφορία την ομοιομορφία που επικρατεί για τις γραφικές παραστάσεις ενεργειακής κατανάλωσης: της CPU (του Σχήματος 5.1ii) με το Σχήμα 5.3i, του WiFi (του Σχήματος 5.1i) με τα Σχήματα 5.3ii, 5.3iii, 5.3iv, 5.3v και του 3G (του Σχήματος 5.1iii) με το Σχήμα 5.3vi.



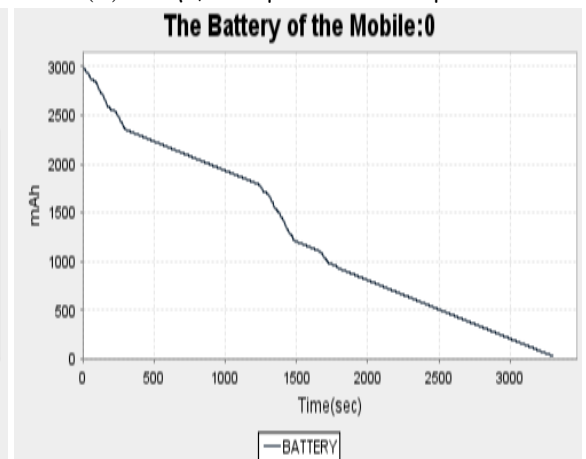
(i) Ενεργειακή Κατανάλωση Wi-Fi



(ii) Ενεργειακή Κατανάλωση CPU



(iii) Ενεργειακή Κατανάλωση 3G

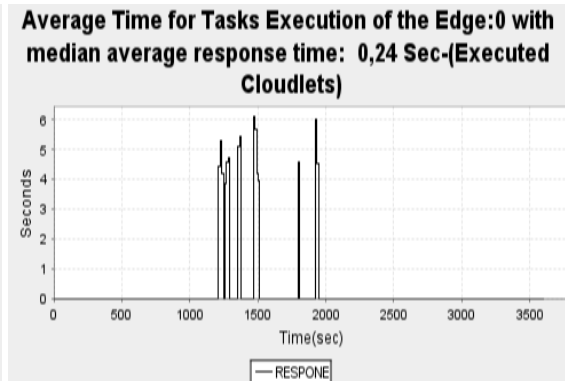
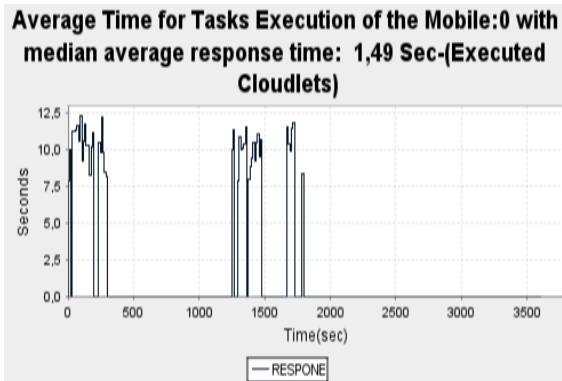


(iv) Μπαταρία Κινητής Συσκευής

Σχήμα 5.1: Ενεργειακές Καταναλώσεις Κινητής Συσκευής Με Αλγόριθμο Mobi-Het

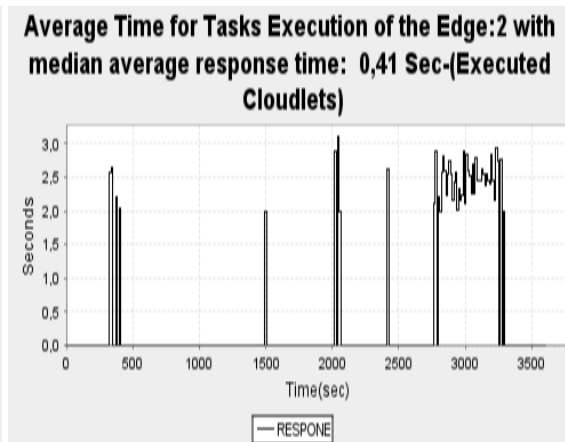
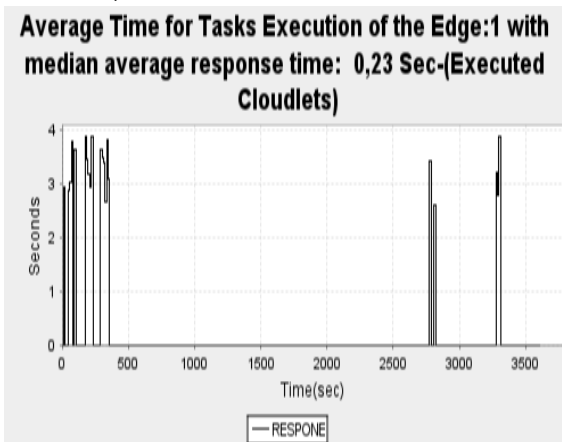


Σχήμα 5.2: Κάτοψη Κίνησης Κινητής Συσκευής Εντός Πεδίου (grid).



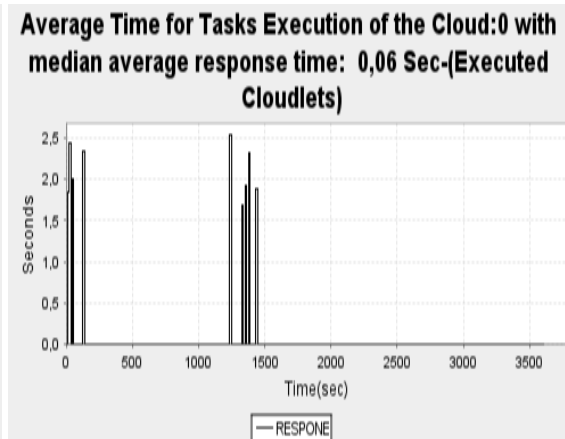
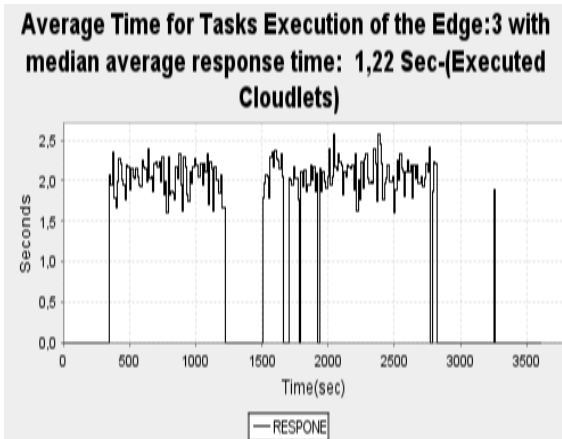
(i) Μέση Απόκριση Εργασιών Κινητής Συσκευής

(ii) Μέση Απόκριση Εργασιών Cloudlet 0



(iii) Μέση Απόκριση Εργασιών Cloudlet 1

(iv) Μέση Απόκριση Εργασιών Cloudlet 2

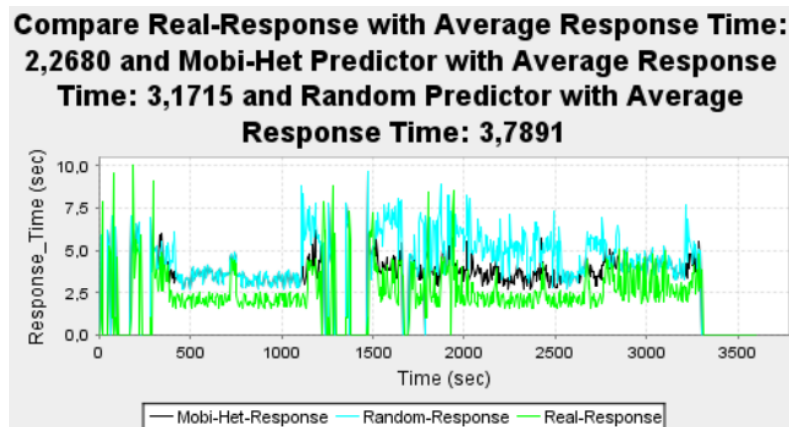


(v) Μέση Απόκριση Εργασιών Cloudlet 3

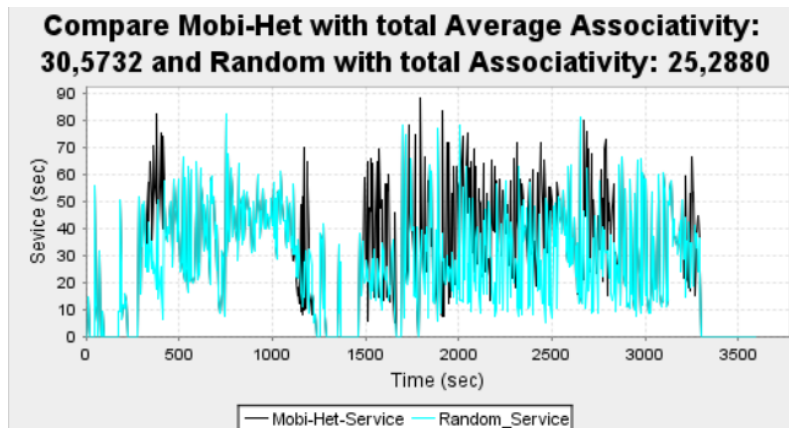
(vi) Μέση Απόκριση Εργασιών Master Cloud

Σχήμα 5.3: Συνολική Μέση Απόκριση Εργασιών Με Αλγόριθμο Δυναμικής Δρομολόγησης Mobi-Het

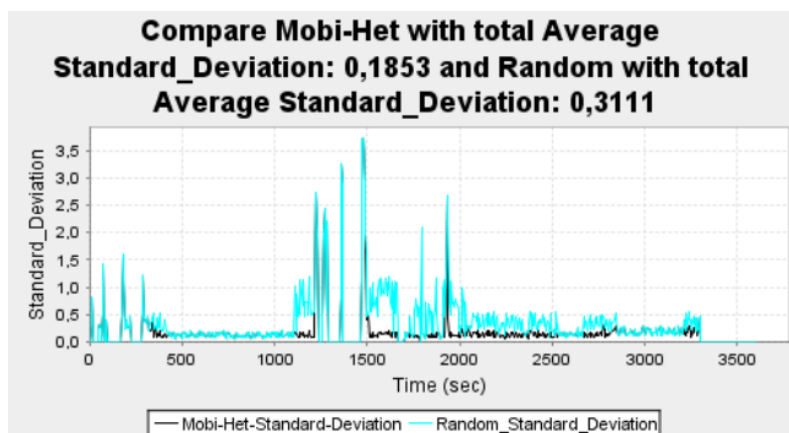
Στη συνέχεια θα εξετάσουμε την αξία της Δυναμικής Δρομολόγησης Mobi-Het και την ανάγκη ύπαρξης βέλτιστης κατανομής των πόρων του συστήματος στα αιτήματα εφαρμογών. Επομένως για το ίδιο πείραμα έχουμε τα Σχήματα 5.4, 5.5, 5.6 τα οποία και παρατίθενται παρακάτω.



Σχήμα 5.4: Μέσος Χρόνος Απόκρισης Εφαρμογής, Αλγόριθμου Mobi-Het, Μοντέλου Πρόβλεψης Mobi-Het Και Μοντέλου Πρόβλεψης Τυχαίας Κατανομής Εργασιών.



Σχήμα 5.5: Μέση Τιμή Associativity, Μοντέλου Mobi-Het Και Τυχαίας Κατανομής Εργασιών.



Σχήμα 5.6: Μέση Τυπική Απόκλιση Ανά Χρονικό Διάστημα Εκφόρτωσης Εφαρμογής Σε Cloudlet, Μοντέλου Mobi-Het Και Τυχαίας Κατανομής Εργασιών.

Από το Σχήμα 5.4 παρατηρούμε ότι ο αλγόριθμος δυναμικής δρομολόγησης Mobi-Het προβλέπει πολύ λιγότερο συνολικό χρόνο απόκρισης σε σχέση με τη τυχαία δρομολόγηση. Σε κάθε απόφαση δρομολόγησης η κατανομή των εργασιών είναι τέτοια ώστε να εξασφαλίζεται η ελαχιστοποίηση του συνολικού χρόνου απόκρισης όπως ακριβώς αναλύθηκε και στο Κεφ.4. Επιπρόσθετα ο αλγόριθμος "τείνει" να εξυπηρετήσει τις κινητές συσκευές με τον υψηλότερο προβλεπόμενο χρόνο παραμονής (υψηλό associativity, Παρ.4.2.1) και αυτό φαίνεται και από το Σχήμα 5.5. Τέλος ο αλγόριθμος στοχεύει στην μειωμένη τυπική απόκλιση του συστήματος και άρα καλύτερη κατανομή των πόρων κάτι το οποίο αποδεικνύεται και από το Σχήμα 5.6.

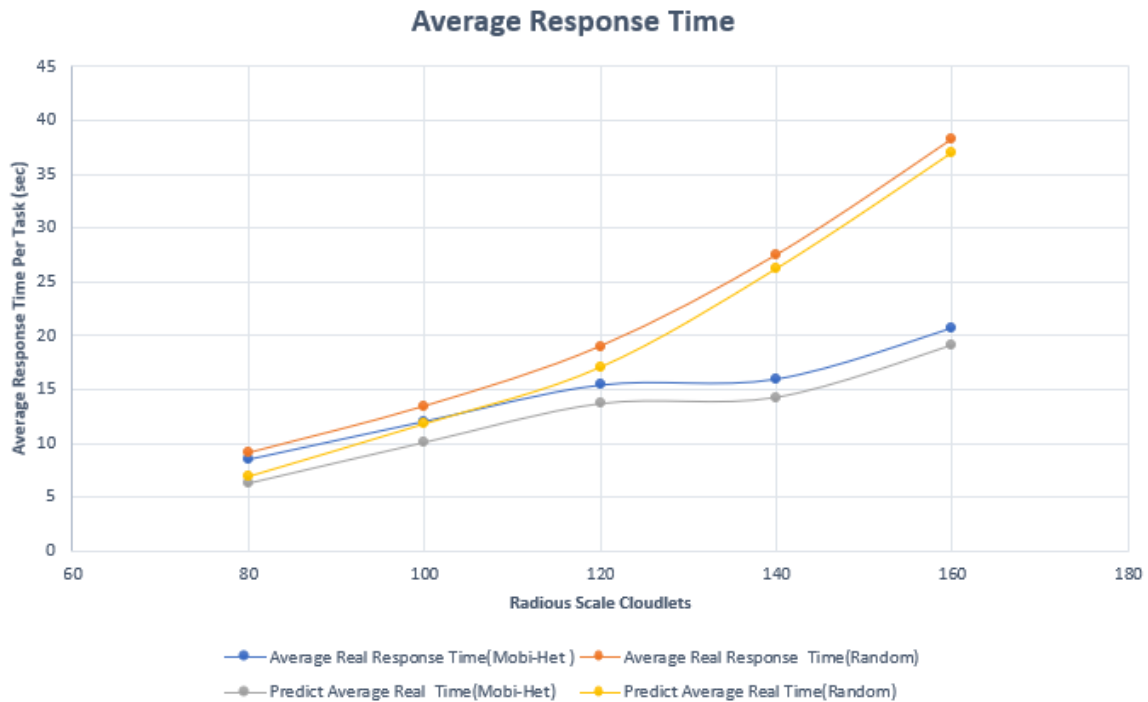
Αυτό που παρατηρείται είναι η προσπάθεια του Αλγορίθμου Mobi-Het για βελτιστοποίηση της αντικειμενικής συνάρτησης 4.1 του Κεφ.4 και σε σύγκριση με το Τυχαίο Μοντέλο Δρομολόγησης οδηγούμαστε στο γεγονός ότι η ανάγκη ύπαρξης δυναμικής δρομολόγησης στο σύστημα είναι ζωτικής σημασίας για την μεγιστοποίηση της ποιότητας εξυπηρέτησης του χρήστη (QoS).

5.2.2 Πείραμα Β: Σύγκριση Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het) με Μοντέλο Τυχαίας Δρομολόγησης Με Μεταβλητή Παράμετρο Την Ακτίνα Τοπικού Υπολογιστικού Νέφους

Στο πείραμα αυτό τρέξαμε για τις ακόλουθες σταθερές συνθήκες προσομοίωσης:

- Αριθμός Κινητών Συσκευών (num of mobiles) = 30
- Κινητικότητα Κινητών Συσκευών (rangeMin, rangeMax)=(4.0, 6.0)m \ s

Θα εξετάσουμε τα πειραματικά αποτελέσματα για κάθε τιμή του παράγοντα radius scale για το μοντέλο Δυναμικής Δρομολόγησης (Mobi-Het) σε σύγκριση με ένα απλό Μοντέλο Τυχαίας Δρομολόγησης. Στο Σχήμα 5.7 έχουμε το μέσο χρόνο απόκρισης του συστήματος ανά αίτηση εφαρμογής εκ των οποίων οι γραφικές: Average Real Response Time (Mobi-Het) και Average Real Response Time (Random) αναφέρονται στο μέσο χρόνο απόκρισης μιας εφαρμογής πάνω στα cloudlets για Δυναμική Δρομολόγηση (Mobi-Het) και για το Μοντέλο Τυχαίας Δρομολόγησης αντίστοιχα. Επιπρόσθετα οι γραφικές: Predict Average Real Time (Mobi-Het) και Predict Average Real Time (Random) αναφέρονται στον **προβλεπόμενο μέσο χρόνο απόκρισης μίας εφαρμογής**(Παρ.4.2.3, Υπολογισμός Προβλεπόμενου Χρόνου Απόκρισης Εφαρμογής πάνω στα cloudlets), για Δυναμική Δρομολόγηση (Mobi-Het) και για το Μοντέλο Τυχαίας Δρομολόγησης αντίστοιχα.

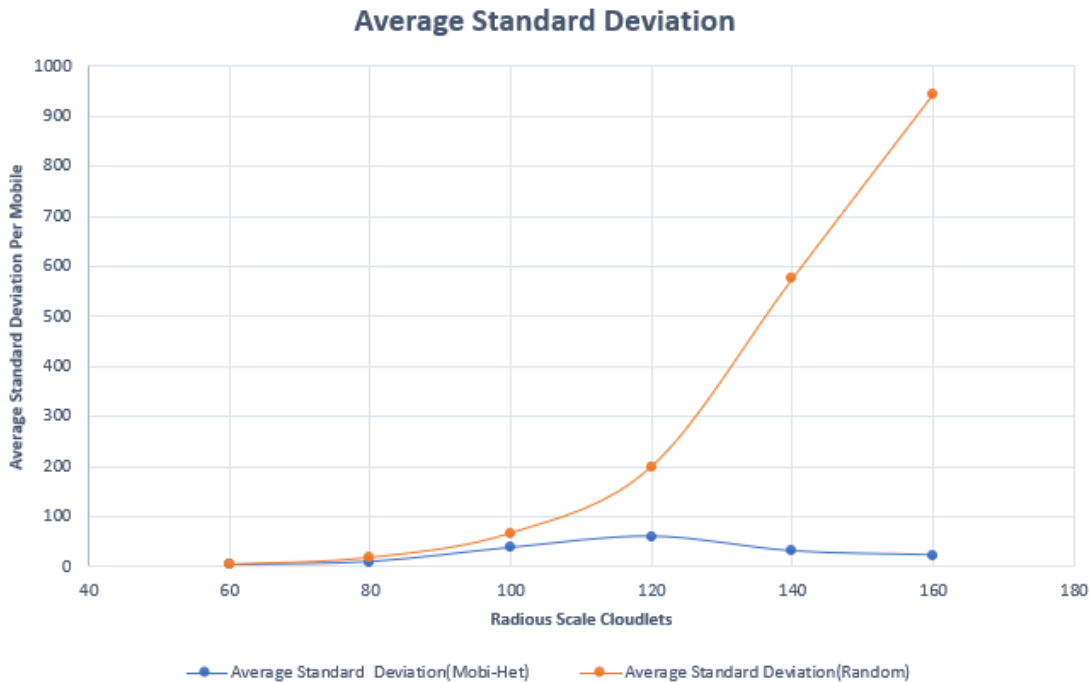


Σχήμα 5.7: Συνολικός Μέσος Χρόνος Απόκρισης Ανά Εφαρμογή Σε Συνάρτηση Του Radius Scale.

Παρατηρούμε ότι τα πειραματικά αποτελέσματα της πρόβλεψης του μέσου χρόνου απομακρυσμένης εκτέλεσης μιας εφαρμογής πλησιάζουν τα πραγματικά παρά τη τυχαιότητα που επικρατεί στη κίνηση των χρηστών. Το μοντέλο της πρόβλεψης του χρόνου εργασίας χρησιμοποιείται στην Απόφαση Δυναμικής Δρομολόγησης (Mobi-Het) όπως περιγράφηκε στο Κεφ. 4 σε αντίθεση με το Μοντέλο της Τυχαίας Επιλογής που βασίζεται στη τυχαία επιλογή. Παρόλο αυτά αξίζει να σημειωθεί ότι **και τα δύο μοντέλα δεν δρομολογούν εργασία σε cloudlet σε περίπτωση που η πρόβλεψη παραμονής της κινητής συσκευής εντός του πεδίου είναι μικρότερη από τον προβλεπόμενο χρόνο εξυπηρέτησης**. Αυτό που παρατηρείται είναι η κλιμακωτή αύξηση της διαφοράς μεταξύ των γραφικών Real Response Mobi-Het Time και Real Response Random Time καθώς αυξάνεται ο παράγοντας radius scale. Αυτό είναι λογικό από την άποψη ότι αυξάνεται η αλληλοκάλυψη των cloudlets που σημαίνει ότι ο αλγόριθμος (Mobi-Het) έχει καλύτερη εποπτεία του συστήματος εφόσον αυξάνεται η πιθανότητα ένας χρήστης να βρίσκεται εντός δύο ή και παραπάνω πεδίων και άρα αυξάνονται οι επιλογές σωστής δρομολόγησης. Ωστόσο και οι δύο γραφικές είναι γνησίως αύξουσες εφόσον με την αύξηση της ακτίνας εμβέλειας των cloudlets αυξάνεται και το φόρτο εργασίας τους (ο χρήστης έχει μεγαλύτερη πιθανότητα να βρίσκεται εντός ενός πεδίου), με άμεση συνέπεια την αύξηση της συνολικής μέσης απόκρισης μιας εφαρμογής. Παρατηρούμε επίσης για τη τιμή 80 του radius scale (στην οποία τιμή έχουμε απειροελάχιστη αλληλοκάλυψη των cloudlets), οι γραφικές είναι πολύ κοντά η μία από την άλλη και στην ουσία να μη φαίνεται η ουσιαστική διαφορά ανάμεσα στα δύο εξεταζόμενα μοντέλα.

Τα πειραματικά αυτά αποτελέσματα μπορούμε να τα συμπεράνουμε και από την αλ-

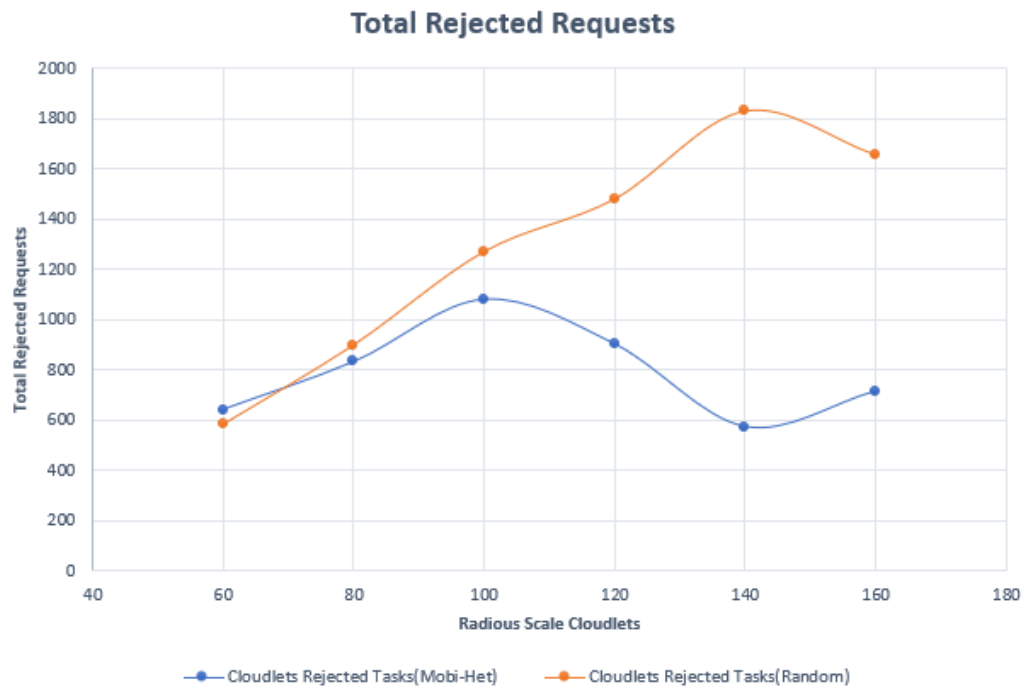
λαγή της μέσης τυπικής απόκλισης του συστήματος που βλέπει μια κινητή συσκευή σε κάθε πρόβλημα απόφασης δυναμικής δρομολόγησης εργασίας (σε κάθε χρονικό διάστημα δημιουργίας μιας εφαρμογής) από τον Master Cloud όπως φαίνεται στο Σχήμα 5.8.



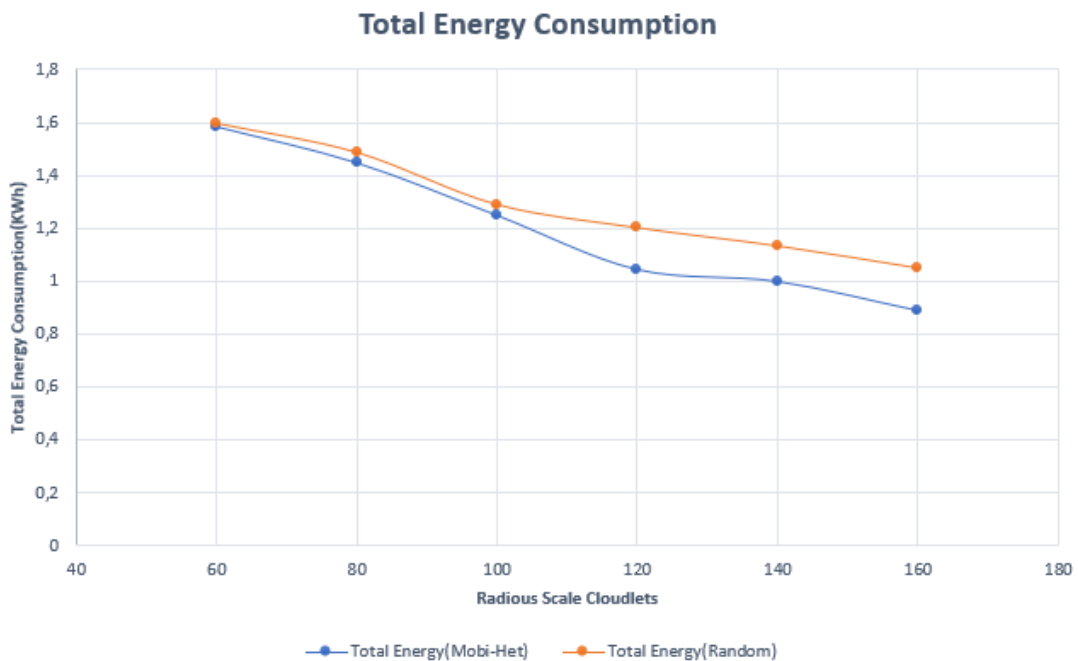
Σχήμα 5.8: Μέση Τυπική Απόκλιση Συνολικού Φορτίου Των Cloudlets Ανά Χρονικό Διάστημα Δημιουργίας Μιας Εφαρμογής Σε Συνάρτηση Του Radius Scale.

Παρατηρούμε ότι το μοντέλο Δυναμικής Δρομολόγησης (Mobi-Het) κάνει καλή κατανομή των πόρων όλων των cloudlets σε αντίθεση με το Μοντέλο Τυχαίας Δρομολόγησης που μπορεί να οδηγήσει στην υπερφόρτωση ενός cloudlet με άμεση συνέπεια την αύξηση της απόκρισης και της αναμονής του χρήστη για απομακρυσμένη εκτέλεση. Αξίζει να σημειωθεί και η μικρή τυπική απόκλιση συστήματος που επιτυγχάνεται για μεγάλη τιμή του radius scale και κατά συνέπεια υψηλό φόρτο εργασίας πάνω στα cloudlets στο μοντέλο Δυναμικής Δρομολόγησης (Mobi-Het), το οποίο επιδεικνύει τη σωστή κατανομή εργασιών που επιτυγχάνει ο αλγόριθμος για μεγάλη επιφάνεια αλληλοκάλυψης των cloudlets.

Ως άμεση συνέπεια της συχνής υπερφόρτωσης να έχουμε και αυξημένη τιμή απόρριψης αιτημάτων στο Μοντέλο Τυχαίας Δρομολόγησης όπως φαίνεται στο Σχήμα 5.9 το οποίο έχει ως προφανές αποτέλεσμα μεγαλύτερη ενεργειακή κατανάλωση (Σχήμα 5.10), εφόσον οι χρήστες είναι αναγκασμένοι ένα αίτημα είτε να το εκτελέσουν τοπικά είτε να το δρομολογήσουν στον Master Cloud (μέσω δικτύου 3G) το οποίο έχει μεγαλύτερο ενεργειακό κόστος.



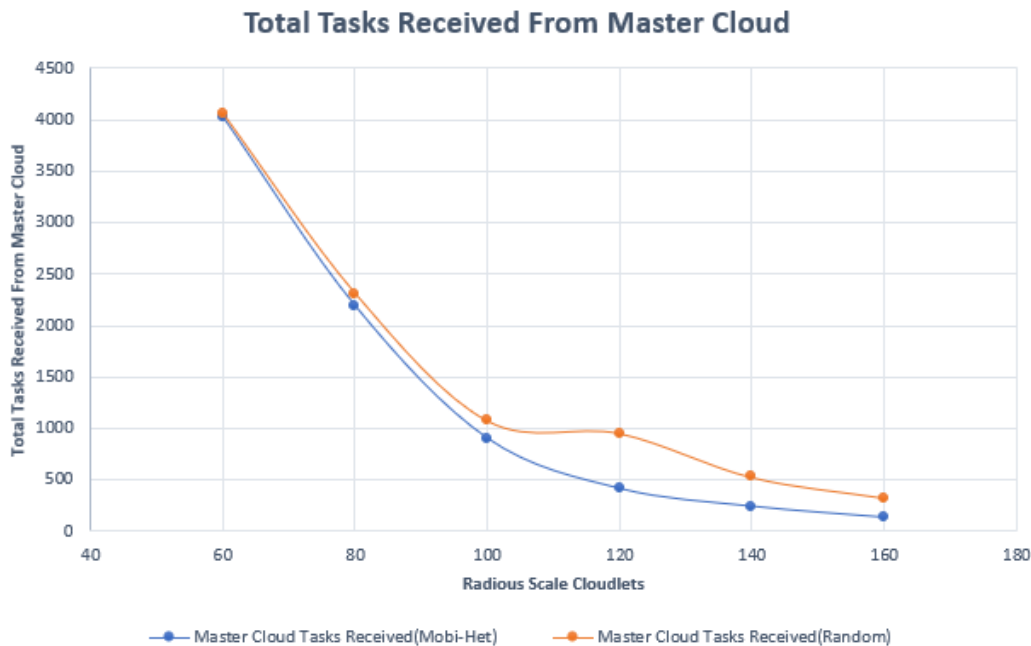
Σχήμα 5.9: Συνολικά Απορριφθέντα Αιτήματα Εφαρμογών Σε Συνάρτηση Του Radius Scale.



Σχήμα 5.10: Συνολική Κατανάλωση Ενέργειας Κινητών Συσκευών Σε Συνάρτηση Του Radius Scale.

Η μικρή διαφορά στο ενεργειακό κόστος των δύο μοντέλων οφείλεται στο γεγονός ότι ο Master Cloud τείνει να επιβαρυνθεί προκειμένου να ελαττώσει την ενεργειακή

δαπάνη των χρηστών όπως φαίνεται στο Σχήμα 5.11 όπου παρατηρούμε ότι στο Μοντέλο Τυχαίας Δρομολόγησης απορροφά συνολικά περισσότερα απορριφθέντα αιτήματα εφαρμογών σε σχέση με το Μοντέλο Δυναμικής Δρομολόγησης (Mobi-Het). Επιπρόσθετα παρατηρούμε μικρό αριθμό απόρριψης πακέτων στη Δυναμική Δρομολόγηση (Mobi-Het) για μεγάλη τιμή του radius scale ως απόρροια ότι ο χρήστης έχει μεγαλύτερη πιθανότητα να μείνει εντός της εμβέλειας ενός cloudlet για μεγάλο χρονικό διάστημα προκειμένου να εξυπηρετηθεί και να μην απορριφθεί.



Σχήμα 5.11: Συνολικά Ληφθέντα Αιτήματα Εφαρμογών Από Τον Master Cloud Σε Συνάρτηση Του Radius Scale.

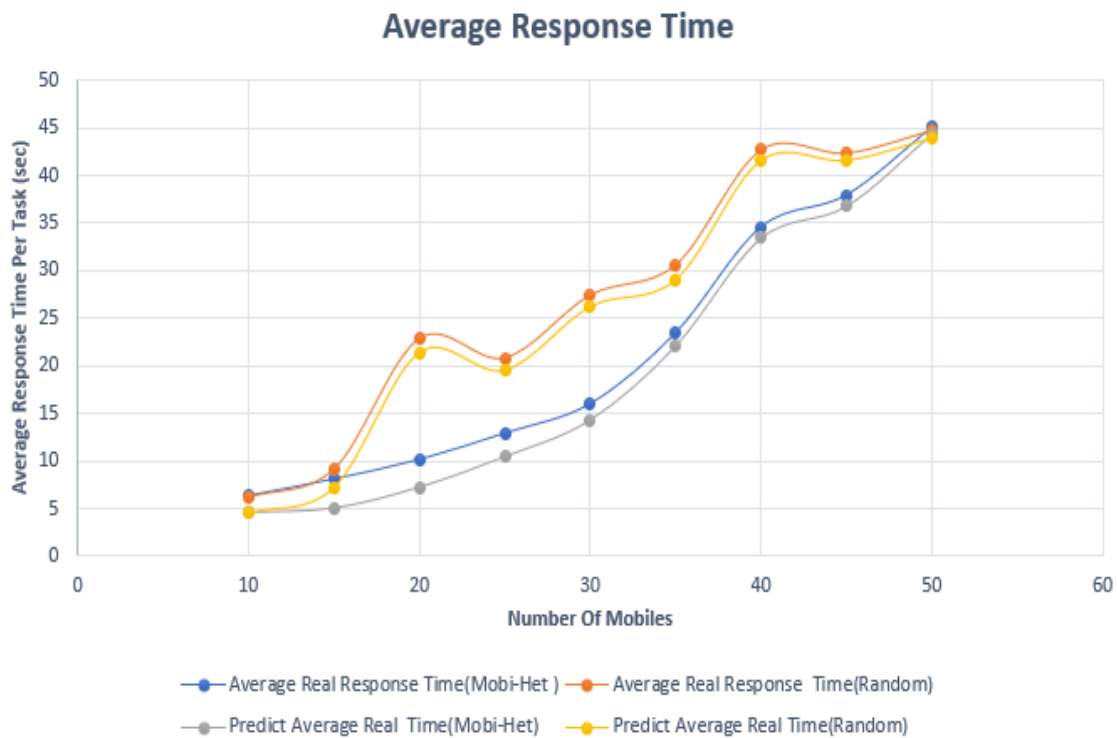
Σαν συμπέρασμα καταλήγουμε ότι για να λειτουργήσει αποδοτικά ο αλγόριθμος δυναμικής δρομολόγησης (Mobi-Het) θα πρέπει να έχουμε περιοχές αλληλοκάλυψης ανάμεσα στα πεδία WiFi των AP των cloudlets προκειμένου να έχουμε αρκετές επιλογές δρομολόγησης και άρα να υπάρχει η δυνατότητα της βέλτιστης επιλογής.

5.2.3 Πείραμα Γ: Σύγκριση Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het) με Μοντέλο Τυχαίας Δρομολόγησης Με Μεταβλητή Παράμετρο Τον Αριθμό Των Κινητών Συσκευών

Στο πείραμα αυτό τρέξαμε για τις ακόλουθες σταθερές συνθήκες προσομοίωσης:

- Radius Scale = 140
- Κινητικότητα Κινητών Συσκευών (rangeMin, rangeMax) = $(4.0, 6.0)m \setminus s$

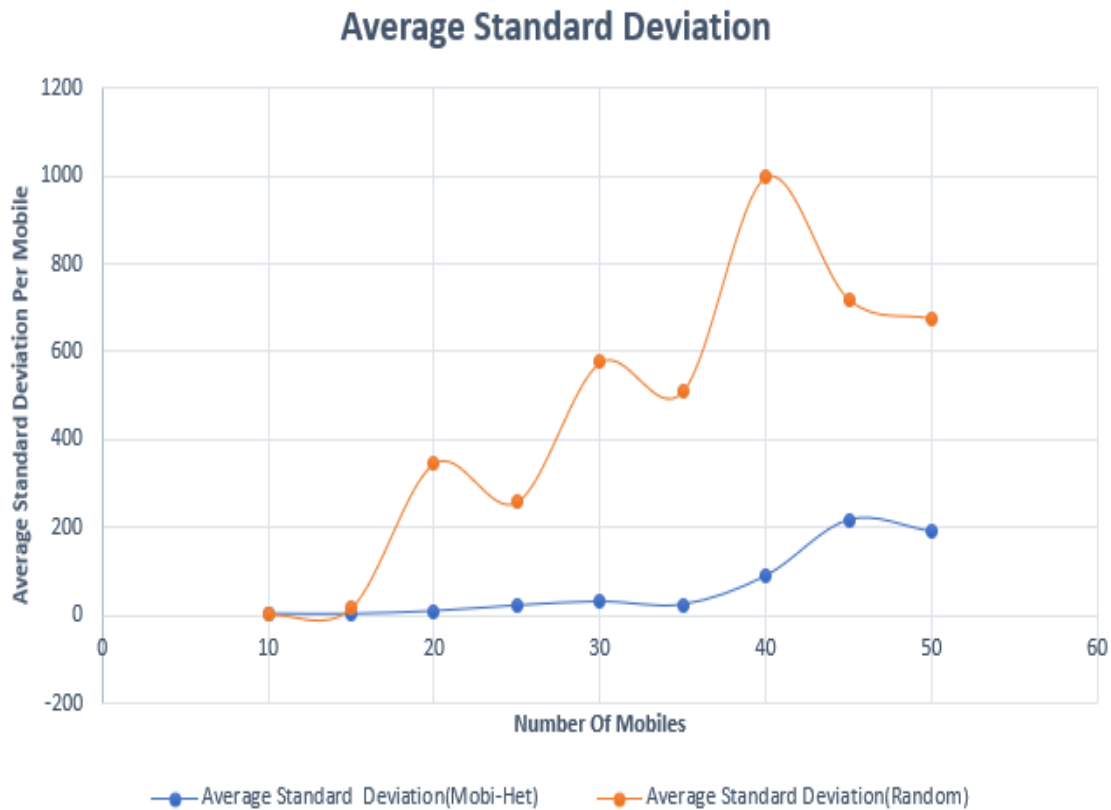
Θα εξετάσουμε τα πειραματικά αποτελέσματα για κάθε τιμή της μεταβλητής παραμέτρου (number of mobiles), τόσο για το μοντέλο Δυναμικής Δρομολόγησης (Mobi-Het) όσο και για το Μοντέλο Τυχαίας Δρομολόγησης.



Σχήμα 5.12: Συνολικός Μέσος Χρόνος Απόκρισης Ανά Εφαρμογή Σε Συνάρτηση Του Αριθμού των Κινητών Συσκευών.

Από το Σχήμα 5.12 παρατηρούμε ότι για μικρό αριθμό κινητών συσκευών τα δύο μοντέλα δεν απέχουν πολύ στον πραγματικό μέσο χρόνο απόκρισης εφαρμογής. Αυτό συμβαίνει γιατί οι διαθέσιμοι πόροι του συστήματος είναι υπεραρκετοί σε σχέση με το φόρτο εργασίας και το σύστημα δύσκολα θα φτάνει σε καταστάσεις κορεσμού καθ' όλη τη διάρκεια της προσομοίωσης. Όσο αυξάνεται ο αριθμός των κινητών συσκευών και κατά συνέπεια το συνολικό φόρτο εργασίας τόσο πιο πολύ νόημα αποκτά η έννοια της δυναμικής δρομολόγησης και αυτό φαίνεται στο Σχήμα 5.12, όπου για ένα εύρος τιμών των κινητών συσκευών το μοντέλο της Δυναμικής Δρομολόγησης (Mobi-Het) έχει μικρότερο συνολικά μέσο χρόνο απόκρισης. Ωστόσο αυτό που παρατηρείται είναι ότι τα δύο μοντέλα συγκλίνουν από μία τιμή της παραμέτρου και μετά. Αυτό συμβαίνει γιατί ο αλγόριθμος δυναμικής δρομολόγησης (Mobi-Het) δεν μπορεί να δρομολογήσει το υπερβολικά μεγάλο φόρτο εργασίας εφόσον η δυνατότητα της επιλεκτικής κατανομής διαθέσιμων πόρων σε εργασίες παύει να υφίσταται. Στην ουσία και τα δύο μοντέλα σε τέτοιες τιμές καταλήγουν στην εύρεση του πρώτου διαθέσιμου ελεύθερου πόρου όλων των cloudlets εντός εμβέλειας του χρήστη για άμεση δρομολόγηση εργασίας. Οι παρατηρήσεις αυτές όπως και στο Πείραμα Β, μπορούν να επεξηγηθούν και μέσα από τη πορεία της τυπικής απόκλισης που βλέπει μία κινητή συσκευή μέσα στο σύστημα των cloudlets όπως φαίνεται στο Σχήμα 5.13 και για τα

δύο μοντέλα.



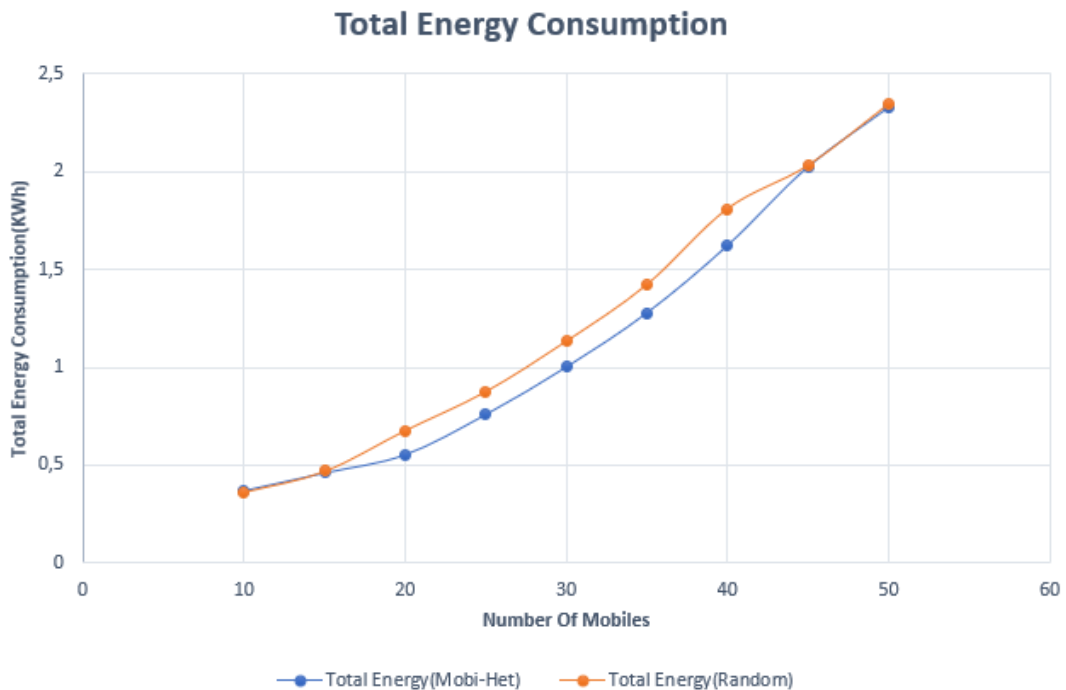
Σχήμα 5.13: Μέση Τυπική Απόκλιση Συνολικού Φορτίου Των Cloudlets Ανά Χρονικό Διάστημα Δημιουργίας Μιας Εφαρμογής Σε Συνάρτηση Του Αριθμού των Κινητών Συσκευών.

Παρατηρούμε ότι ακολουθούν την ίδια μορφή καμπύλης με τις καμπύλες μέσου χρόνου απόκρισης εφαρμογής που περιγράψαμε προηγουμένως, με το μοντέλο δυναμικής δρομολόγησης να έχει μικρότερη τυπική απόκλιση για ένα εύρος τιμών, το οποίο δικαιολογεί και το χαμηλό μέσο χρόνο απόκρισης εφαρμογής για αυτές τις τιμές. Για τις τιμές 10 και 20 της μεταβλητής παραμέτρου παρατηρούμε ότι έχουμε σχεδόν ίδιες τυπικές αποκλίσεις και αυτό δικαιολογείται από το χαμηλό φορτίο που επικρατεί στο σύστημα σε σχέση με τους συνολικούς διαθέσιμους με άμεση συνέπεια να έχουν και παραπλήσιους χρόνους απόκρισης όπως αποδείχθηκε. Για τις τιμές 45 και 50 παρατηρούμε την αυξημένη τυπική απόκλιση που βλέπει η κινητή συσκευή στο σύστημα στο μοντέλο Δυναμικής Δρομολόγησης (Mobi-Het) σε σχέση με τις υπόλοιπες τιμές της μεταβλητής παραμέτρου που δείχνει την αδυναμία του συστήματος να κρατήσει μια ομοιόμορφη κατανομή εργασίας στο σύστημα. Στην ουσία όλες οι ουρές αναμονής των VMs όλων των cloudlets είναι πάντα γεμάτες και δεν έχει σημασία η έννοια της επιλεκτικής κατανομής εργασίας σε τέτοιες τιμές. Επιπρόσθετα παρατηρούμε την ανάλογη μορφολογία καμπυλών και στο ρυθμό απόρριψης αιτημάτων όπως φαίνεται στο Σχήμα 5.14 με το Μοντέλο Τυχαίας Δρομολόγησης να έχει υψηλότερο ρυθμό απόρριψης το οποίο όπως και στο Πείραμα A οδηγεί σε μεγαλύτερη ενεργειακή κα-

τανάλωση για τις κινητές συσκευές το οποίο αποδεικνύει και από το Σχήμα 5.15.



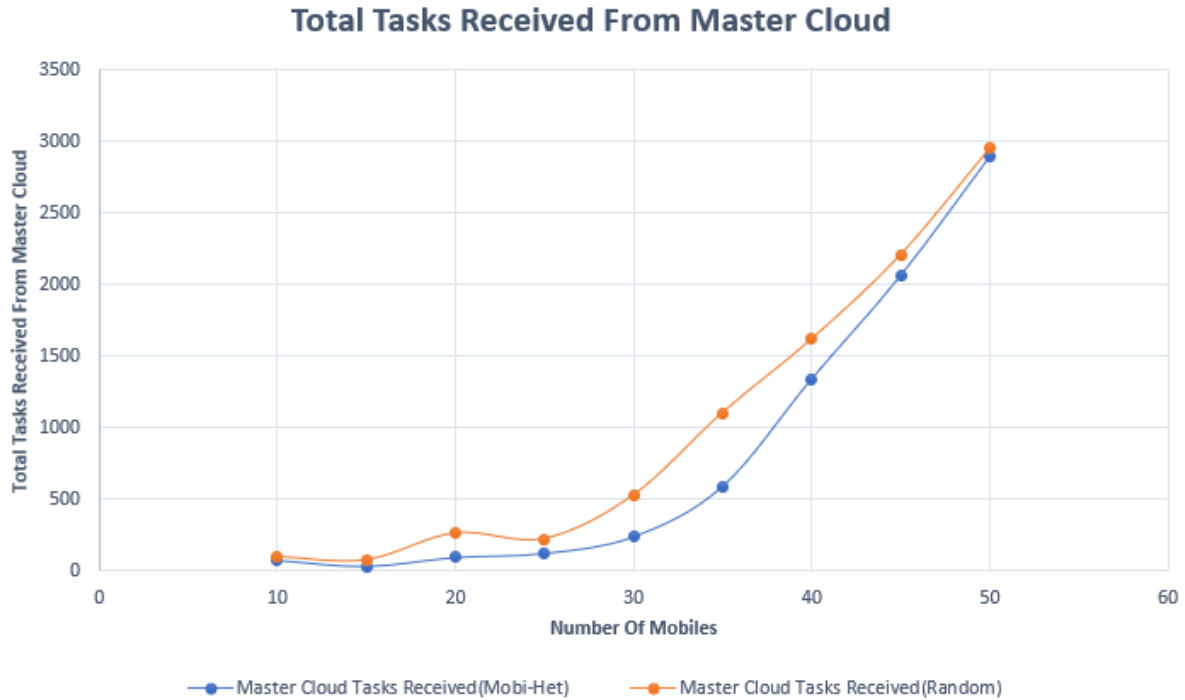
Σχήμα 5.14: Συνολικά Απορριφθέντα Αιτήματα Εφαρμογών Σε Συνάρτηση Του Αριθμού των Κινητών Συσκευών.



Σχήμα 5.15: Συνολική Κατανάλωση Ενέργειας Κινητών Συσκευών Σε Συνάρτηση Του Αριθμού των Κινητών Συσκευών.

Η μικρή ενεργειακή διαφορά έγκειται στην ιδιότητα του Master Cloud να δέχεται όσο

γίνεται περισσότερα απορριφθέντα πακέτα από τα cloudlets κάτι το οποίο φαίνεται και από το Σχήμα 5.16. Τέλος αξίζει να σημειωθεί η πολύ ψηλή τιμή απόρριψης αιτημάτων και για τα δύο μοντέλα (για *num of mobiles* = 50), πράγμα το οποίο επιβεβαιώνει και το φαινόμενο του κορεσμού του συστήματος.



Σχήμα 5.16: Συνολικά Ληφθέντα Αιτήματα Εφαρμογών Από Τον Master Cloud Σε Συνάρτηση Του Αριθμού των Κινητών Συσκευών.

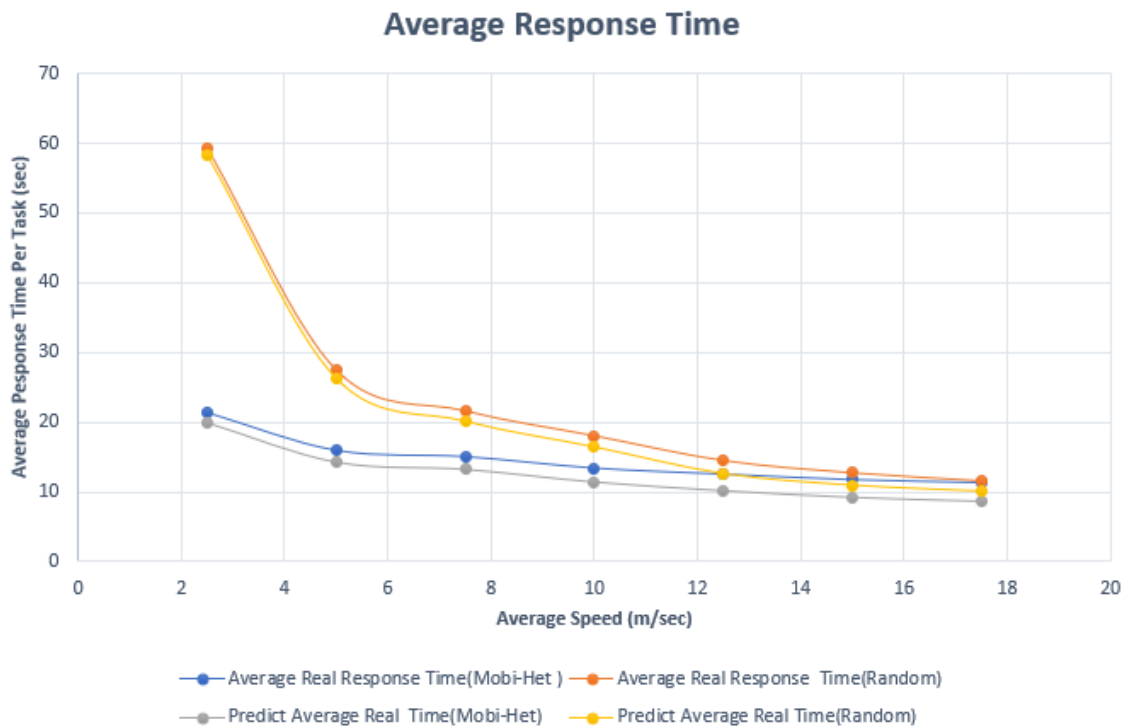
Σαν συμπέρασμα καταλήγουμε ότι για να λειτουργήσει αποδοτικά ο αλγόριθμος δυναμικής δρομολόγησης (Mobi-Het) θα πρέπει να έχουμε έναν ικανοποιητικό αριθμό κινητών συσκευών (ελάχιστο και μέγιστο) προκειμένου και να μπορούμε να εκμεταλλευτούμε όλους τους διαθέσιμους πόρους των cloudlets αποδοτικά αλλά και να μη φτάνουμε σε καταστάσεις που δε μπορεί να αντεπεξέλθει η σχεδιαζόμενη αρχιτεκτονική Mobi-Het (κατάσταση κορεσμού).

5.2.4 Πείραμα Δ: Σύγκριση Αλγορίθμου Δυναμικής Δρομολόγησης (Mobi-Het) με Μοντέλο Τυχαίας Δρομολόγησης Με Μεταβλητή Παράμετρο Τον Βαθμό Κινητικότητας Των Κινητών Συσκευών

Στο πείραμα αυτό τρέξαμε για τις ακόλουθες σταθερές συνθήκες προσομοίωσης:

- Radius Scale = 140
- Αριθμός Κινητών Συσκευών(number of mobiles) = 30

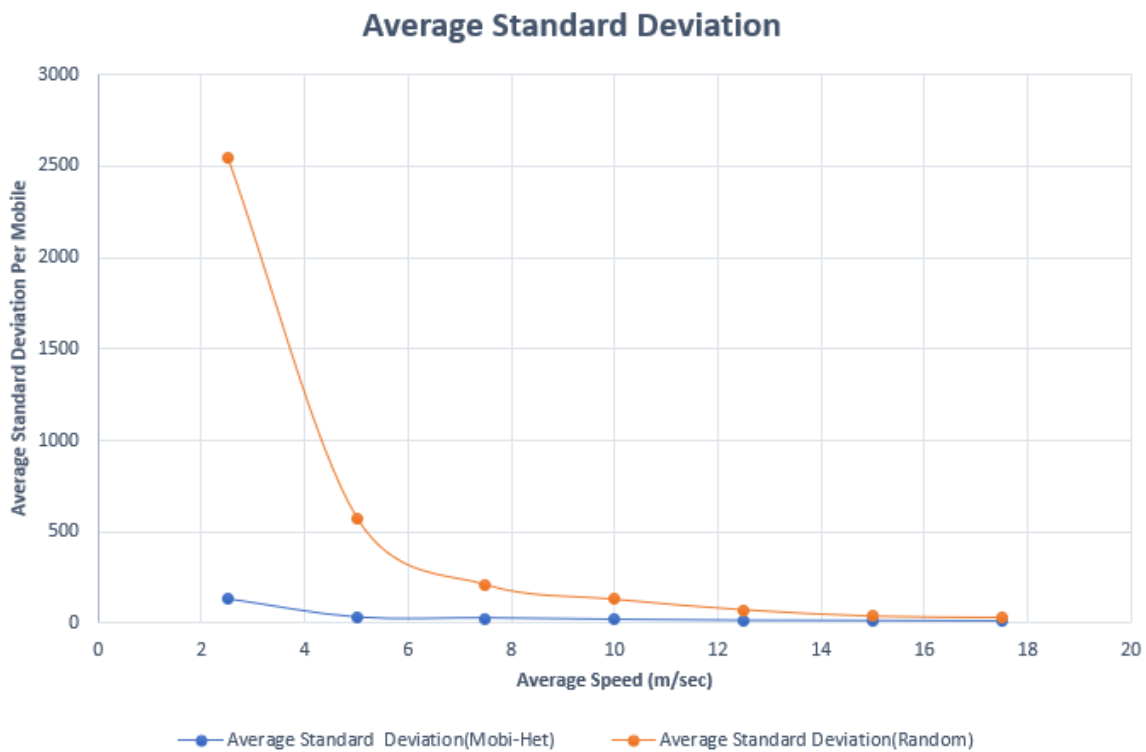
Θα εξετάσουμε τα πειραματικά αποτελέσματα για κάθε τιμή της μεταβλητής παραμέτρου της μέσης ταχύτητας κινητικότητας του χρήστη (Average Mobility Speed of Devices), τόσο για το μοντέλο Δυναμικής Δρομολόγησης (Mobi-Het) όσο και για το Μοντέλο Τυχαίας Δρομολόγησης.



Σχήμα 5.17: Συνολικός Μέσος Χρόνος Απόκρισης Ανά Εφαρμογή Σε Συνάρτηση Της Μέσης Ταχύτητας Κινητικότητας Των Κινητών Συσκευών.

Από το Σχήμα 5.17 παρατηρούμε ότι για μικρή τιμή της μεταβλητής παραμέτρου το μοντέλο της Δυναμικής Δρομολόγησης (Mobi-Het) ανταποκρίνεται πολύ πιο αποτελεσματικά από το μοντέλο της Τυχαίας Δρομολόγησης όσον αναφορά το μέσο χρόνο απόκρισης εργασίας. Αυτό οφείλεται στο γεγονός ότι για χαμηλή μέση ταχύτητα κινητικότητας το μοντέλο πρόβλεψης του χρόνου παραμονής της κινητής συσκευής εντός του πεδίου του cloudlet που χρησιμοποιεί ο αλγόριθμος δυναμικής δρομολόγησης (Mobi-Het) είναι πολύ πιο ακριβής (σωστός υπολογισμός του associativity του χρήστη στην εξίσωση 4.1 του Κεφ. 4). Ωστόσο με την αύξηση της τιμής της παραμέτρου τόσο τα δύο μοντέλα συγκλίνουν και αυτό συμβαίνει επειδή ο χρήστης έχει μεγαλύτερη πιθανότητα να φύγει εκτός πεδίου και να βρεθεί εντός άλλων πεδίων cloudlet ή και όχι. Το περιθώριο λάθους στη λήψη απόφασης δυναμικής δρομολόγησης στη τελευταία περίπτωση είναι πολύ υψηλό με αποτέλεσμα να έχουμε πολλές απορρίψεις (μη σωστή αξιοποίηση των πόρων που σημαίνει ανεπάρκεια ωφέλιμου φορτίου), εφόσον η τελική θέση του χρήστη είναι απροσδιόριστη προκειμένου να γίνει ο σωστός υπολογισμός του χρόνου παραμονής του χρήστη εντός του πεδίου και άρα να συγκλίνουμε προς το Τυχαίο Μοντέλο Δρομολόγησης. Με άλλα λόγια στην υψηλή μέση κινητικότητα και για τα δύο μοντέλα ο μέσος χρόνος απόκρισης εφαρμογής γίνεται αισθητά μικρότερος σε σχέση με αυτή της χαμηλής μέσης κινητικότητας κάτι

το οποίο αποδεικνύει και το χαμηλό συνολικό φόρτο εργασίας πάνω στα cloudlets. Όπως και στα προηγούμενα πειράματα έτσι και εδώ η μέση τυπική απόκλιση φορτίου που βλέπει η κινητή συσκευή πάνω στα cloudlets σε κάθε πρόβλημα απόφασης δυναμικής δρομολόγησης εργασίας, είναι αναμενόμενο να έχει την ανάλογη μορφή όπως φαίνεται στο Σχήμα 5.18.

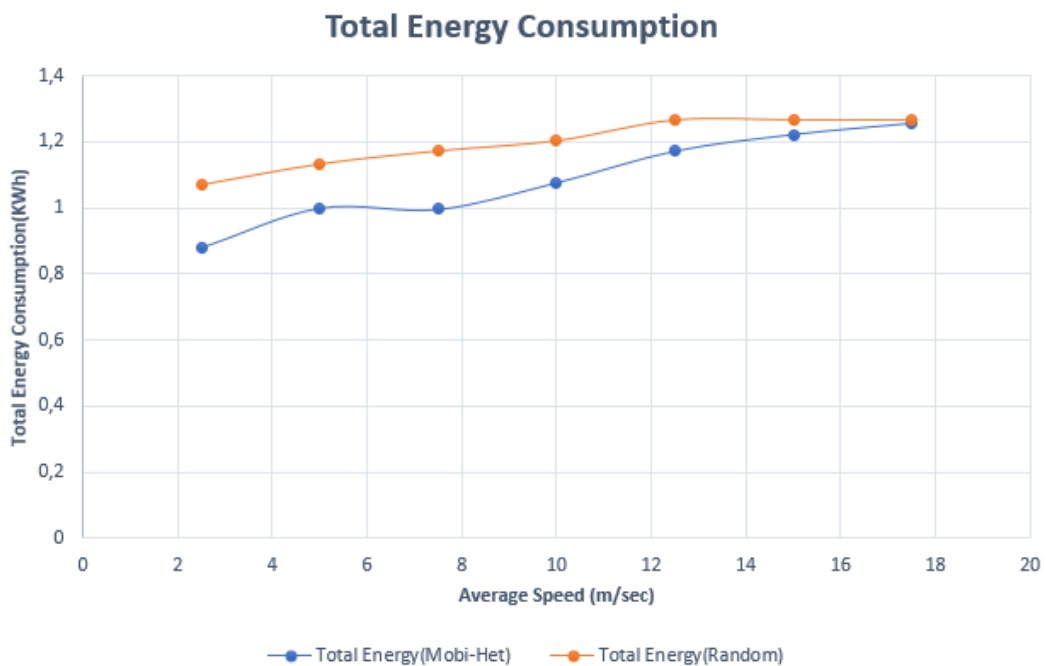


Σχήμα 5.18: Μέση Τυπική Απόκλιση Συνολικού Φορτίου Των Cloudlets Ανά Χρονικό Διάστημα Δημιουργίας Μιας Εφαρμογής Σε Συνάρτηση Της Μέσης Ταχύτητας Κινητικότητας Των Κινητών Συσκευών.

Παρατηρούμε ότι οι καμπύλες είναι γνησίως φθίνουσες εφόσον με την αύξηση της μέσης ταχύτητας κινητικότητας του χρήστη τόσο περισσότερα αιτήματα εφαρμογών για απομακρυσμένη εκτέλεση απορρίπτονται (που σημαίνει ότι πολλοί πόροι του συστήματος μένουν αναξιοποίητοι) και από τα δύο μοντέλα (υπενθυμίζουμε ότι και τα δύο μοντέλα δεν δρομολογούν εργασία σε cloudlet σε περίπτωση που η πρόβλεψη παραμονής της κινητής συσκευής εντός του πεδίου είναι μικρότερη από τον προβλεπόμενο χρόνο εξυπηρέτησης). Αυτό φαίνεται και από τον συνολικό αριθμό απόρριψης πακέτων που είναι αρκετά υψηλός και για τα δύο μοντέλα από μία τιμή της παραμέτρου και μετά όπως φαίνεται στο Σχήμα 5.19. Για μικρή κινητικότητα το μοντέλο δυναμικής δρομολόγησης (Mobi-Het) διαχειρίζεται ικανοποιητικά τις αιτήσεις με μικρό ρυθμό απόρριψης, σε αντίθεση με το μοντέλο της τυχαίας δρομολόγησης όπου φτάνουμε συχνά σε φαινόμενα υπερφόρτωσης ενός cloudlet (φαινόμενο κορεσμού) σε αρκετές φάσεις του πειράματος με αποτέλεσμα να έχουμε και πολύ μεγάλο ρυθμό απόρριψης αλλά και πολύ υψηλή μέση τυπική απόκλιση ανά κινητή συσκευή.



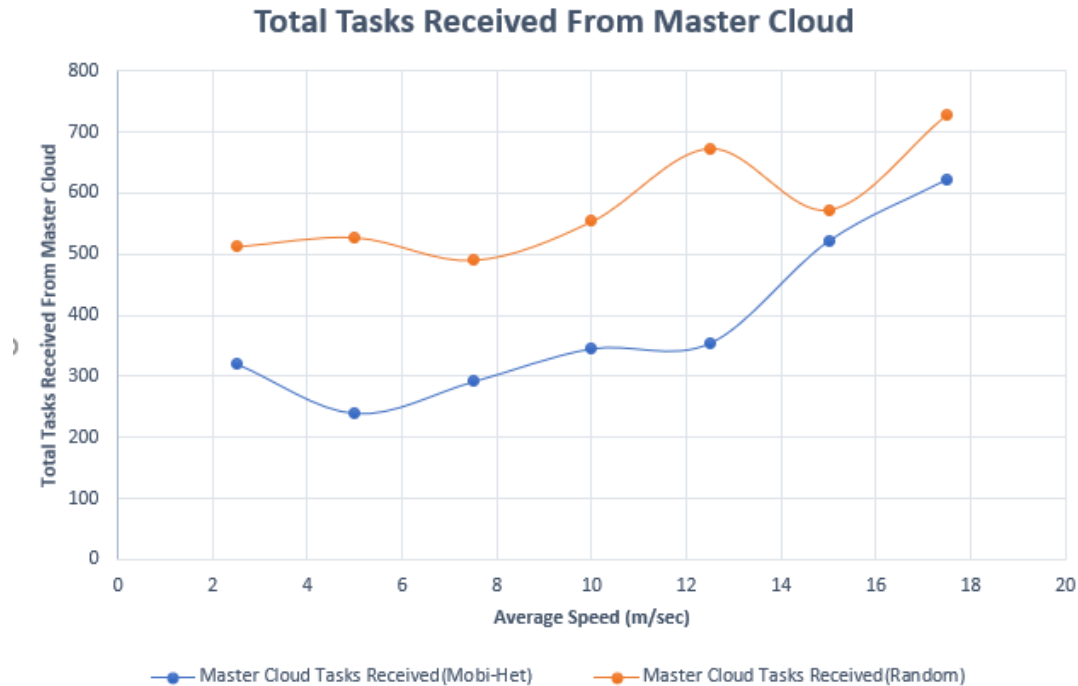
Σχήμα 5.19: Συνολικά Απορριφθέντα Αιτήματα Εφαρμογών Σε Συνάρτηση Της Μέσης Ταχύτητας Κινητικότητας Των Κινητών Συσκευών.



Σχήμα 5.20: Συνολική Κατανάλωση Ενέργειας Κινητών Συσκευών Σε Συνάρτηση Της Μέσης Ταχύτητας Κινητικότητας Των Κινητών Συσκευών.

Στο Σχήμα 5.20 παρατηρούμε ότι η ενεργειακή κατανάλωση των κινητών συσκευών αυξάνεται με την αύξηση της τιμής της παραμέτρου, το οποίο οφείλεται στην αύξηση του ρυθμού απόρριψης αιτημάτων και για τα δύο μοντέλα όπως περιγράψαμε προη-

γούμενος. Αξιοσημείωτο είναι το γεγονός ότι το μοντέλο Δυναμικής Δρομολόγησης έχει μικρότερο ενεργειακό κόστος για ένα εύρος της παραμέτρου και η μικρή διαφορά ενεργειακής κατανάλωσης για τα δύο μοντέλα όπως και στα προηγούμενα πειράματα οφείλεται στην ιδιομορφία του Master Cloud στη προσπάθεια του να απορροφήσει το απορριφθέν φορτίο (Σχήμα 5.21).



Σχήμα 5.21: Συνολικά Ληφθέντα Αιτήματα Εφαρμογών Από Τον Master Cloud Σε Συνάρτηση Της Μέσης Ταχύτητας Κινητικότητας Των Κινητών Συσκευών.

Σαν συμπέρασμα καταλήγουμε ότι για να λειτουργήσει αποδοτικά ο αλγόριθμος δυναμικής δρομολόγησης (Mobi-Het) θα πρέπει να έχουμε μία μικρή μέση κινητικότητα των χρηστών προκειμένου η πρόβλεψη μελλοντικής επόμενης θέσης του χρήστη να είναι όσο γίνεται πιο κοντά στη πραγματική και να είμαστε όσο γίνεται πιο σίγουροι για το associativity του χρήστη και άρα πιο σίγουροι για τη βέλτιστη κατανομή των αιτημάτων στα cloudlets.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντική Εξέλιξη Εργασίας

6.1 Σύνοψη συμπερασμάτων αξιολόγησης

Στα πλαίσια αυτής της διπλωματικής, υλοποιήσαμε και κάναμε πειραματική αξιολόγηση στη Δυναμική Δρομολόγηση Εφαρμογών Κινητών Συσκευών (Mobi-Het) σε τοπικά υπολογιστικά νέφη γνωστά και ως cloudlets με στόχο την ελαχιστοποίηση της συνολικής απόκρισης και του ενεργειακού κόστους του χρήστη. Ο αλγόριθμος ουσιαστικά ταξινομεί τα αιτήματα εφαρμογών στους διαθέσιμους πόρους των cloudlet του συστήματος χρησιμοποιώντας γραμμικό προγραμματισμό για την ελαχιστοποίηση της συνάρτησης 4.1 η οποία βασίζεται στους εξής τρεις παράγοντες:

- **Την Μεγιστοποίηση Του Βαθμού Συσχέτισης**, δηλαδή την εξυπηρέτηση των κινητών συσκευών με τη μεγαλύτερη πρόβλεψη παραμονής εντός του πεδίου ενός cloudlet, γνωστό και ως associativity (Παρ.4.2.1, Κεφ.4).
- **Την Ελαχιστοποίηση Της Τυπικής Απόκλισης Του Συνολικού Φορτίου**, που στοχεύει στην καλή κατανομή και αξιοποίηση όλων των πόρων του συστήματος για μην έχουμε φαινόμενα κορεσμού ή υπερφόρτωσης (Παρ.4.2.2, Κεφ.4).
- **Την Ελαχιστοποίηση Του Συνολικού Χρόνου Εκτέλεσης Της Εφαρμογής**, ο οποίος αποτελείται από τον χρόνο αποστολής του κώδικα εφαρμογής, το χρόνο της απομακρυσμένης εκτέλεσης και του χρόνου λήψης του αρχείου αποτελέσματος (Παρ.4.2.3, Κεφ.4).

Για σύγκριση του αλγορίθμου χρησιμοποιήθηκε το Μοντέλο Τυχαίας Επιλογής που στην ουσία δεν χρησιμοποιεί την εξίσωση 4.1 αλλά αντιστοιχεί τυχαία τα αιτήματα εφαρμογών με τα διαθέσιμα και εντός εμβέλειας cloudlets. Να σημειωθεί ότι και για τα δύο μοντέλα όταν ο Χρόνος Εκτέλεσης της Εφαρμογής είναι μεγαλύτερος από τη πρόβλεψη παραμονής του χρήστη εντός του πεδίου τότε η κινητή συσκευή δεν γίνεται

αποδεκτή για απομακρυσμένη εκτέλεση. Συνοψίζοντας τα πειραματικά αποτελέσματα, μπορούμε να πούμε ότι ο αλγόριθμος Δυναμικής Δρομολόγησης (Mobi-Het) που χρησιμοποιήθηκε στα πλαίσια αυτής της διπλωματικής είναι αρκετά αποδοτικός και καταφέρνει να ρίξει το μέσο χρόνο απόκρισης εφαρμογής, ενώ ταυτόχρονα μας δίνει μικρότερο ενεργειακό κόστος για τις κινητές συσκευές του συστήματος. Είναι εύκολα κλιμακώσιμος και χρονικά αποδοτικός εφόσον υπολογίζεται σε πολύ ισχυρούς εξυπηρετητές (Master Clouds), και μπορεί να εφαρμοστεί σε πραγματικές συνθήκες. Ωστόσο, για μεγάλο αριθμό κινητών συσκευών, υψηλή μέση κινητικότητα και μικρή επιφάνεια αλληλοκάλυψης των cloudlets ο αλγόριθμος φαίνεται να μην αποδίδει αποτελεσματικά, όπως φάνηκε από τα προηγούμενα πειραματικά αποτελέσματα. Παρ' όλα αυτά είναι μία καλή προσέγγιση για τις μέσες τιμές των παραμέτρων που αναλύθηκαν στο Κεφ.5, οι οποίες δεν απέχουν και πολύ από τις πραγματικές τιμές που επικρατούν στα πλαίσια μιας "έξυπνης" πόλης.

6.2 Μελλοντικές Επεκτάσεις

Το έργο της παρούσας διπλωματικής εργασίας μπορεί να επεκταθεί και στις εξής ακόλουθες ενδεικτικές, αλλά συγκεκριμένες, κατευθύνσεις:

- Την εφαρμογή ενός δυναμικού μοντέλου για τη πρόβλεψη της θέσης της κινητής συσκευής που θα βασίζεται στο ιστορικό κίνησης και κατεύθυνσης του χρήστη, το οποίο θα μας δώσει μεγαλύτερη ακρίβεια για το χρόνο παραμονής του κινητού και άρα πιο σωστές δρομολογήσεις στα cloudlets. Ενδεικτικά προτείνουμε τη χρήση του μοντέλου τυχαίας κινητικότητας SMOOTH [4] το οποίο βασίζεται σε πραγματικές δειγματοληψίες κίνησης κινητών συσκευών στα πλαίσια μιας "έξυπνης" πόλης.
- Χρήση μοντέλου πρόβλεψης πιθανής εισόδου της κινητής συσκευής σε άλλο cloudlet. Από το cloudlet αυτό θα μπορεί να λάβει εκτελέσιμα, τα οποία έλαβε από κάποια άλλα cloudlets στα οποία βρέθηκε πρωτύτερα η κινητή συσκευή και έστειλε κώδικες εφαρμογών για απομακρυσμένη εκτέλεση. Πρόκειται στην ουσία για ένα μοντέλο που θα βασίζεται σε ένα πρωτόκολλο επικοινωνίας ανάμεσα στα cloudlets με στόχο την ελαχιστοποίηση των συνολικών απορριφθέντων αιτημάτων των κινητών συσκευών του συστήματος. Ως αποτέλεσμα είναι ο χρήστης να απολαμβάνει μεγαλύτερη ποιότητα υπηρεσίας με μικρότερο ενεργειακό κόστος εφόσον αυξάνεται ο συνολικός αριθμός εφαρμογών που γίνονται αποδεκτά από τα cloudlets για απομακρυσμένη εκτέλεση. Το μοντέλο αυτό πρέπει να έχει χαμηλή σε κόστος υπολογιστική πολυπλοκότητα προκειμένου να μην έχουμε σημαντική αύξηση στο συνολικό χρόνο απόκρισης του συστήματος.
- Η παράλληλη εκτέλεση διαφορετικών τμημάτων μιας εφαρμογής μεγάλων δεδομένων σε πολλαπλά υπολογιστικά νέφη μπορεί να αποτελεί πιθανό μελλοντικό ερευνητικό πρόβλημα. Η ιεράρχηση εργασιών και ο έλεγχος εισαγωγής μεγάλων εργασιών δεδομένων μπορεί επίσης να βελτιώσει τις επιδόσεις των εφαρμογών σε αυτόν τον τομέα.

- Επέκταση της αρχιτεκτονικής του συστήματος και του αλγορίθμου Mobi-Het που περιγράφηκε στη θεωρία του Κεφ. 4 σε τρεις διαστάσεις με τη χρήση πολικών συντεταγμένων. Τα cloudlets μπορούν να προσομοιωθούν με τέλειες σφαίρες στο χώρο, ενώ η κίνηση των κινητών συσκευών μπορεί να υλοποιηθεί σε τρεις κάθετους άξονες αντί για δύο. Με αυτόν τον τρόπο μπορούμε να προσομοιώσουμε και να προσεγγίσουμε καλύτερα ένα πραγματικό σενάριο δυναμικής δρομολόγησης.
- Την πειραματική αξιολόγηση και μελέτη σε πραγματικό σύστημα για να φανούν οι πλήρεις δυνατότητες του αλγορίθμου δυναμικής δρομολόγησης (Mobi-Het).

Βιβλιογραφία

- [1] Sharad Agarwal, John Dunagan, Navendu Jain, Stefan Saroiu, Alec Wolman, and Habinder Bhogan. Volley: Automated data placement for geo-distributed cloud services. 2010.
- [2] Marios Avgeris, Dimitrios Dechouniotis, Nikolaos Athanasopoulos, and Symeon Papavassiliou. Adaptive resource allocation for computation offloading: A control-theoretic approach. *ACM Transactions on Internet Technology (TOIT)*, 19(2):23, 2019.
- [3] Marios Avgeris, Dimitrios Spatharakis, Dimitrios Dechouniotis, Nikos Kalatzis, Ioanna Roussaki, and Symeon Papavassiliou. Where there is fire there is smoke: a scalable edge computing framework for early fire detection. *Sensors*, 19(3):639, 2019.
- [4] Christian Bettstetter. Smooth is better than sharp: a random mobility model for simulation of wireless networks. In *Proceedings of the 4th ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 19–27. ACM, 2001.
- [5] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, 25(6):599–616, 2009.
- [6] Min Chen, Yixue Hao, Yong Li, Chin-Feng Lai, and Di Wu. On the computation offloading at ad hoc cloudlet: architecture and service modes. *IEEE Communications Magazine*, 53(6):18–24, 2015.
- [7] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. Clonecloud: elastic execution between mobile device and cloud. In *Proceedings of the sixth conference on Computer systems*, pages 301–314. ACM, 2011.
- [8] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. Maui: making smartphones last longer with code offload. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 49–62. ACM, 2010.

- [9] Asma Enayet. *Mobility Aware Optimal Resource Allocation in Heterogeneous Mobile Cloud Computing*. PhD thesis, MS thesis, Department of Computer Science and Engineering, University of Dhaka, 2015.
- [10] Asma Enayet, Md Abdur Razzaque, Mohammad Mehedi Hassan, Atif Alamri, and Giancarlo Fortino. A mobility-aware optimal resource allocation architecture for big data task execution on mobile cloud in smart cities. *IEEE Communications Magazine*, 56(2):110–117, 2018.
- [11] Niroshinie Fernando, Seng W Loke, and Wenny Rahayu. Mobile cloud computing: A survey. *Future generation computer systems*, 29(1):84–106, 2013.
- [12] Keke Gai, Meikang Qiu, Hui Zhao, Lixin Tao, and Ziliang Zong. Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. *Journal of Network and Computer Applications*, 59:46–54, 2016.
- [13] Kirak Hong, David Lillethun, Umakishore Ramachandran, Beate Ottenwalder, and Boris Koldehofe. Opportunistic spatio-temporal event processing for mobile situation awareness. In *Proceedings of the 7th ACM international conference on Distributed event-based systems*, pages 195–206. ACM, 2013.
- [14] Junxian Huang, Feng Qian, Alexandre Gerber, Z Morley Mao, Subhabrata Sen, and Oliver Spatscheck. A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 225–238. ACM, 2012.
- [15] Yaser Jararweh, Loai Tawalbeh, Fadi Ababneh, and Fahd Dosari. Resource efficient mobile computing using cloudlet infrastructure. In *2013 IEEE 9th International Conference on Mobile Ad-hoc and Sensor Networks*, pages 373–377. IEEE, 2013.
- [16] Roelof Kemp, Nicholas Palmer, Thilo Kielmann, and Henri Bal. Cuckoo: a computation offloading framework for smartphones. In *International Conference on Mobile Computing, Applications, and Services*, pages 59–79. Springer, 2010.
- [17] Mahbub E Khoda, Md Abdur Razzaque, Ahmad Almogren, Mohammad Mehedi Hassan, Atif Alamri, and Abdulhameed Alelaiwi. Efficient computation offloading decision in mobile cloud computing over 5g network. *Mobile Networks and Applications*, 21(5):777–792, 2016.
- [18] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *2012 Proceedings IEEE Infocom*, pages 945–953. IEEE, 2012.
- [19] Jianqiang Li, Luxiang Huang, Yaoming Zhou, Suiqiang He, and Zhong Ming. Computation partitioning for mobile cloud computing in a big data environment. *IEEE Transactions on Industrial Informatics*, 13(4):2009–2018, 2017.

- [20] Jiwei Li, Kai Bu, Xuan Liu, and Bin Xiao. Enda: Embracing network inconsistency for dynamic application offloading in mobile cloud computing. In *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*, pages 39–44. ACM, 2013.
- [21] Ting-Yi Lin, Ting-An Lin, Cheng-Hsin Hsu, and Chung-Ta King. Context-aware decision engine for mobile cloud offloading. In *2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 111–116. IEEE, 2013.
- [22] A Tawalbeh Lo'ai, Waseem Bakheder, and Houbing Song. A mobile cloud computing model using the cloudlet scheme for big data applications. In *2016 IEEE First International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE)*, pages 73–77. IEEE, 2016.
- [23] Henrik Madsen, Bernard Burtschy, G Albeanu, and FL Popentiu-Vladicescu. Reliability in the utility computing era: Towards reliable fog computing. In *2013 20th International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 43–46. IEEE, 2013.
- [24] Verdi March, Yan Gu, Erwin Leonardi, George Goh, Markus Kirchberg, and Bu Sung Lee. μ cloud: towards a new paradigm of rich mobile applications. *Procedia Computer Science*, 5:618–624, 2011.
- [25] Beate Ottenwalder, Boris Koldehofe, Kurt Rothermel, Kirak Hong, and Umakishore Ramachandran. Recep: Selection-based reuse for distributed complex event processing. In *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems*, pages 59–70. ACM, 2014.
- [26] Zhengyuan Pang, Lifeng Sun, Zhi Wang, Erfang Tian, and Shiqiang Yang. A survey of cloudlet based mobile computing. In *2015 International Conference on Cloud Computing and Big Data (CCBD)*, pages 268–275. IEEE, 2015.
- [27] M Reza Rahimi, Nalini Venkatasubramanian, Sharad Mehrotra, and Athanasios V Vasilakos. Mapcloud: Mobile applications on an elastic and scalable 2-tier cloud architecture. In *Proceedings of the 2012 IEEE/ACM fifth international conference on utility and cloud computing*, pages 83–90. IEEE Computer Society, 2012.
- [28] M Reza Rahimi, Nalini Venkatasubramanian, and Athanasios V Vasilakos. Music: Mobility-aware optimal service allocation in mobile cloud computing. In *2013 IEEE Sixth International Conference on Cloud Computing*, pages 75–82. IEEE, 2013.
- [29] Mahadev Satyanarayanan, Paramvir Bahl, Ramon Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, pages 14–23, 2009.

- [30] Mohsen Sharifi, Somayeh Kafaie, and Omid Kashefi. A survey and taxonomy of cyber foraging of mobile devices. *IEEE Communications Surveys & Tutorials*, 14(4):1232–1243, 2011.
- [31] Bo Sheng, Qun Li, and Weizhen Mao. Data storage placement in sensor networks. In *Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing*, pages 344–355. ACM, 2006.
- [32] Manoel C Silva Filho, Raysa L Oliveira, Claudio C Monteiro, Pedro RM Inácio, and Mário M Freire. Cloudsim plus: a cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 400–406. IEEE, 2017.
- [33] Yuxia Sun, Junxian Chen, Yong Tang, and Yanjia Chen. Energy modeling of iot mobile terminals on wifi environmental impacts. *Sensors*, 18(6):1728, 2018.
- [34] Haodong Wang, Chiu C Tan, and Qun Li. Snoogle: A search engine for pervasive environments. *IEEE Transactions on Parallel and Distributed Systems*, 21(8):1188–1202, 2009.
- [35] Haodong Wang, Chiu Chiang Tan, and Qun Li. Snoogle: A search engine for the physical world. In *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*, pages 1382–1390. IEEE, 2008.
- [36] Xiaofei Wang, Min Chen, Tarik Taleb, Adlen Ksentini, and Victor CM Leung. Cache in the air: Exploiting content caching and delivery techniques for 5g systems. *IEEE Communications Magazine*, 52(2):131–139, 2014.
- [37] Qiufen Xia, Weifa Liang, Zichuan Xu, and Bingbing Zhou. Online algorithms for location-aware task offloading in two-tiered mobile cloud environments. In *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, pages 109–116. IEEE Computer Society, 2014.
- [38] Fengyuan Xu, Chiu C Tan, Qun Li, Guanhua Yan, and Jie Wu. Designing a practical access point association protocol. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. IEEE, 2010.
- [39] Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Yasir Mehmood, Abdullah Gani, Salimah Mokhtar, and Sghaier Guizani. Enabling communication technologies for smart cities. *IEEE Communications Magazine*, 55(1):112–120, 2017.
- [40] Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data*, pages 37–42. ACM, 2015.

-
- [41] ChuanTao Yin, Zhang Xiong, Hui Chen, JingYuan Wang, Daven Cooper, and Bertrand David. A literature survey on smart cities. *Science China Information Sciences*, 58(10):1–18, 2015.
- [42] Chanmin Yoon, Dongwon Kim, Wonwoo Jung, Chulkoo Kang, and Hojung Cha. Appscope: Application energy metering framework for android smartphone using kernel activity monitoring. In *Presented as part of the 2012 {USENIX} Annual Technical Conference ({USENIX}{ATC} 12)*, pages 387–400, 2012.
- [43] Weiwen Zhang, Yonggang Wen, and Dapeng Oliver Wu. Collaborative task execution in mobile cloud computing under a stochastic wireless channel. *IEEE Transactions on Wireless Communications*, 14(1):81–93, 2014.
- [44] Yifan Zhang, Chiu Tan, and Li Qun. Cachekeeper: a system-wide web caching service for smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 265–274. ACM, 2013.
- [45] Bowen Zhou, Amir Vahid Dastjerdi, Rodrigo N Calheiros, Satish Narayana Srirama, and Rajkumar Buyya. A context sensitive offloading scheme for mobile cloud computing service. In *2015 IEEE 8th International Conference on Cloud Computing*, pages 869–876. IEEE, 2015.