



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

Αλγόριθμοι Απόφασης Μεταφόρτωσης Εργασιών σε
Πραγματικό Χρόνο σε Αρχιτεκτονική Υπολογισμού
Κινητού Νέφους Δύο Επιπέδων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΗΤΡΙΟΣ ΣΤΑΥΡΑΚΑΚΗΣ

Επιβλέπων : Παπαβασιλείου Συμεών
Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2018



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και
Υπολογιστών

Αλγόριθμοι Απόφασης Μεταφόρτωσης Εργασιών σε Πραγματικό Χρόνο σε Αρχιτεκτονική Υπολογισμού Κινητού Νέφους Δύο Επιπέδων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΗΤΡΙΟΣ ΣΤΑΥΡΑΚΑΚΗΣ

Επιβλέπων : Παπαβασιλείου Συμεών
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 5η Μαρτίου 2018.

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

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

.....
Ρουσσάκη Ιωάννα
Επίκουρη Καθηγήτρια Ε.Μ.Π.

Αθήνα, Μάρτιος 2018

.....
Δημήτριος Σταυρακάκης

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

Copyright © Δημήτριος Σταυρακάκης, 2018.
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

Στη σημερινή εποχή η χρήση "έξυπνων" κινητών συσκευών αποτελεί αναπόσπαστο κομμάτι της ζωής των ανθρώπων. Οι μεγάλες δυνατότητες τους αξιοποιούνται σε πολλούς τομείς και βρίσκουν διάφορες εφαρμογές όπως σε κοινωνικά δίκτυα (Facebook, Twitter), παιχνίδια, εφαρμογές υγείας κ.ά. Πολλές φορές καλούνται να εκτελέσουν αρκετά απαιτητικές εργασίες, όμως η επεξεργαστική τους δύναμη είναι περιορισμένη. Για το λόγο αυτό το μοντέλο του Υπολογισμού Κινητού Νέφους (Mobile Cloud Computing) εξελίσσεται και χρησιμοποιείται όλο και περισσότερο. Αποτελεί ένα υπολογιστικό μοντέλο με το οποίο απαιτητικές εργασίες μεταφορτώνονται για να εκτελεστούν σε εξωτερικές υπολογιστικές υποδομές (απομακρυσμένο υπολογιστικό νέφος, τοπική μονάδα υπολογιστικού νέφους) που έχουν περισσότερους πόρους από την ίδια την κινητή συσκευή. Όμως, στις κινητές συσκευές, πέρα από την επεξεργαστική ισχύ, περιορισμένη είναι και η ισχύς της μπαταρίας, ιδίως όταν καλούνται να εκτελέσουν κάποια δύσκολη και ενεργοβόρα εργασία.

Στην παρούσα διπλωματική εργασία εξετάζουμε το πρόβλημα απόφασης μεταφόρτωσης των εργασιών "έξυπνων" κινητών συσκευών σε πραγματικό χρόνο σε μια αρχιτεκτονική υπολογισμού νέφους δύο επιπέδων που αποτελείται από την τοπική μονάδα υπολογιστικού νέφους και το απομακρυσμένο νέφος. Σκοπός μας είναι ο διαμοιρασμός των εργασιών με τέτοιο τρόπο ώστε να αξιοποιούνται κατάλληλα οι διαθέσιμοι υπολογιστικοί πόροι και οι συσκευές να καταναλώνουν το ίδιο ποσοστό της υπολειπόμενης ενέργειας της μπαταρίας τους, ενώ πληρούνται οι απαιτήσεις του χρήστη σε επίπεδο υπηρεσιών. Για την επίτευξη αυτού παρουσιάζεται ένας αποδοτικός αλγόριθμος που λαμβάνει τις αποφάσεις για μεταφόρτωση ή μη των εργασιών σε πραγματικό χρόνο. Η αποδοτικότητα αυτού εξετάζεται μέσα από μία σειρά προσομοιώσεων. Τα αποτελέσματα που λαμβάνουμε δείχνουν ότι ο προτεινόμενος αλγόριθμος είναι αρκετά αποτελεσματικός σε πολλές περιπτώσεις, ξεπερνώντας αντίστοιχους που χρησιμοποιούν διαφορετικές ευριστικές μεθόδους.

Λέξεις κλειδιά

Υπολογισμός Κινητού Νέφους, Υπολογισμός Νέφους, Μεταφόρτωση Διεργασιών, Τοπική Μονάδα Υπολογιστικού Νέφους, Λήψη Απόφασης Μεταφόρτωσης Εργασίας σε Πραγματικό Χρόνο, Επίγνωση Ενεργειακού Πλαισίου, Μεγιστοποίηση Ποσοστού Υπολοιπόμενης Ενέργειας, Βελτιστοποίηση Αποφάσεων για Αποταμίευση Ενέργειας

Abstract

Nowadays wireless portable smart mobile devices play a major role in people's everyday lives. More and more people depend heavily on mobile devices such as smartphones and tablets. They can be used for a large variety of applications, i.e. social media applications, online mobile games etc. These applications usually demand heavy computation. However, the majority of the mobile devices is poor in resources. Mobile Cloud Computing (MCC) platforms aim to overcome resource limitations of mobile devices and wireless networks by offloading computation and data intensive tasks from resource-poor mobile devices to resource-rich clouds. Intensive tasks also consume high amount of mobile devices' energy which is another issue that we should take into consideration and deal with.

In this diploma thesis, we consider an online location-aware offloading problem in a two-tiered mobile cloud computing architecture consisting of a local cloudlet and remote cloud, with an objective to fair share the use of the cloudlet by consuming the same proportional of their mobile device energy, while keeping their individual SLA. In order to achieve this purpose we propose an efficient online algorithm for task offloading decision making. We also conduct experiments by simulations to evaluate the performance of the proposed algorithm. Experimental results demonstrate that the proposed algorithm is really efficient and outperforms other heuristics.

Key words

Mobile Cloud Computing, Cloud Computing, Task Offloading, Cloudlet, Real time decision making for task offloading, Energy efficiency, Maximizing minimum residual energy ratio

Ευχαριστίες

Θα ήθελα ιδιαίτερος να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Παπαβασιλείου Συμεών για την καθοδήγησή του και για την ευκαιρία που μου έδωσε να ασχοληθώ με ένα τόσο ενδιαφέρον αντικείμενο. Ακόμη, θα ήθελα να ευχαριστήσω τους Δεχουνιώτη Δημήτρη και Αυγέρη Μάριο για τις σημαντικές συμβουλές και το χρόνο που αφιέρωσαν βοηθώντας με καθ' όλη τη διάρκεια εκπόνησης της παρούσας εργασίας. Τέλος, θα ήθελα να πω ένα μεγάλο ευχαριστώ στην οικογένεια μου που με στήριξε σε κάθε τομέα σε όλα τα έτη των σπουδών μου αλλά και τους φίλους μου που έδειξαν αμέριστη συμπαράσταση και κατανόηση και επαίξαν καθοριστικό ρόλο στην πορεία μου όλα αυτά τα χρόνια.

Δημήτριος Σταυρακάκης,

Αθήνα, 5η Μαρτίου 2018

Περιεχόμενα

Περίληψη	5
Abstract	7
Ευχαριστίες	9
Περιεχόμενα	11
Κατάλογος πινάκων	13
Κατάλογος σχημάτων	15
1. Εισαγωγή	17
1.1 Θεμελιώδεις ορισμοί	17
1.2 Χαρακτηριστικά και ενδεικτικά πλεονεκτήματα του ΥΚΝ	18
1.2.1 Δυνατότητα για μείωση στο χρόνο εκτέλεσης – πιο εύκολη πλήρωση των απαιτήσεων των υπηρεσιών (SLAs - Service Level Agreements)	18
1.2.2 Δυνατότητα για μείωση στην κατανάλωση ενέργειας	19
1.2.3 Μείωση του κόστους από πλευράς καταναλωτή	19
1.3 Εφαρμογές ανάδειξης χρήσης του ΥΚΝ	19
1.3.1 Εκπαίδευση με χρήση κινητών συσκευών	19
1.3.2 Εφαρμογές Υγείας και Ευζωίας	19
1.3.3 Επεξεργασία Εικόνας και Φυσικής Γλώσσας	20
1.3.4 Αναζήτηση και Αναπαραγωγή Πολυμέσων	20
1.3.5 Εφαρμογές Επεξεργασίας Δεδομένων που λαμβάνονται από Αισθητήρες	20
1.3.6 Μέσα κοινωνικής δικτύωσης σε κινητές συσκευές	20
1.4 Αρχιτεκτονικές για τον ΥΚΝ	20
1.4.1 Υπολογισμός στα άκρα του δικτύου (ΥΑΔ – Mobile Edge Computing)	21
1.4.2 Υπολογισμός Κινητού Νέφους (ΥΚΝ – Mobile Cloud Computing)	21
1.4.3 Υπολογισμός Ομίχλης (ΥΟ-Fog Computing)	21
1.5 Ερευνητικά Θέματα	22
1.5.1 Λειτουργικό κόστος	22
1.5.2 Σύγκριση μεταξύ μεθοδολογιών μεταφόρτωσης σε διαφορετικές εφαρμογές	23
1.5.3 Επίδραση της κινητικότητας του χρήστη	23
1.5.4 Προγραμματισμός-Ρυθμίσεις σε επίπεδο δικτύου	23
1.5.5 Προγραμματισμός Εφαρμογών	23
1.5.6 Ρυθμίσεις και πολιτικές για την ασφάλεια και την εξασφάλιση της ατομικότητας	23
1.5.7 Διαχείριση μεγάλου όγκου Δεδομένων	23
1.5.8 Μεταφόρτωση Εφαρμογών με στόχο την ελαχιστοποίηση της καταναλισκόμενης ενέργειας και/ή του χρόνου εκτέλεσης	24

2. Η έννοια της Μεταφόρτωσης Υπολογισμών στον ΥΚΝ	25
2.1 Μεταφόρτωση Υπολογισμών	25
2.2 Τεχνικές Μεταφόρτωσης	25
2.3 Κριτήρια και τρόποι λήψης Απόφασης για Μεταφόρτωση	26
2.3.1 Τρόποι λήψης απόφασης	26
2.3.2 Τεχνική Διαμοιρασμού Κώδικα	27
2.3.3 Βασικά Κριτήρια Λήψης Απόφασης	28
3. Θεωρητικό Υπόβαθρο	31
3.1 Γενική Περιγραφή του Προβλήματος	31
3.2 Μοντέλο Συστήματος	31
3.3 Ανάλυση Μοντέλου Κατανάλωσης Ενέργειας μιας Μεταφόρτωσης	32
3.4 Αναλυτική Παρουσίαση του Προβλήματος προς Επίλυση	34
3.5 Αλγόριθμος Επίλυσης του Προβλήματος	34
3.5.1 Ανάλυση Μοντέλου Κοστολόγησης μιας Μεταφόρτωσης	35
3.5.2 Δημιουργία Γράφου	36
3.5.3 Εφαρμογή του Αλγορίθμου	38
4. Υλοποίηση και Αξιολόγηση Αποτελεσμάτων	41
4.1 Χρησιμοποιούμενο Λογισμικό	41
4.2 Περιβάλλον Προσομοιώσεων	41
4.3 Αξιολόγηση Αποτελεσμάτων	42
4.3.1 Εξέταση Επιρροής της Παραμέτρου α της Συνάρτησης Κόστους	42
4.3.2 Σύγκριση Απόδοσης τριών Αλγορίθμων (<i>MBM, EW, Random</i>)	47
4.3.3 Εξέταση Απόδοσης του Αλγορίθμου <i>MBM</i> για διαφορετικό πλήθος Χρηστών στον υπό παρατήρηση χώρο	50
4.3.4 Εξέταση Απόδοσης του Αλγορίθμου <i>MBM</i> για διαφορετικό πλήθος Σημείων Πρόσβασης στον υπό παρατήρηση χώρο	54
4.3.5 Εξέταση Μέσου Χρόνου Απόκρισης για την τοπική ΜΥΝ με χρήση του Αλγορίθμου <i>MBM</i> για διαφορετικό πλήθος Πυρήνων επεξεργαστή της Τοπικής ΜΥΝ	56
5. Συμπεράσματα και Προτάσεις για Μελλοντική Έρευνα	59
5.1 Σύνοψη και Συμπεράσματα	59
5.2 Ιδέες για μελλοντική έρευνα	60
Βιβλιογραφία	61
A. Ούγγρικός Αλγόριθμος (Hungarian algorithm)	65
A.1 Βασικές Έννοιες	65
A.1.1 Ετικέτες (Labels)	65
A.1.2 Γράφος Ισότητας (Equality Graph)	65
A.1.3 Ταιριασμένος και Ελεύθερος Κόμβος (Match and Free Vertex)	65
A.1.4 Εναλλάξ Μονπάτια (Alternating Paths)	65
A.1.5 Επαυξημένα Μονοπάτια (Augmenting Paths)	66
A.2 Εκτέλεση του Αλγορίθμου	66

Κατάλογος πινάκων

3.1	Συγκεντρωτικός Πίνακας Μεταβλητών	36
4.1	Ενεργειακά Κόστη ανά Δίκτυο	42
4.2	Κατανομή των εργασιών στις υπολογιστικές υποδομές για διαφορετικές τιμές του α	46
4.3	Κατανομή των εργασιών στις υπολογιστικές υποδομές για διαφορετικούς αλγορίθμους μεταφόρτωσης	50
4.4	Κατανομή των εργασιών στις υπολογιστικές υποδομές για διαφορετικές τιμές του α	56

Κατάλογος σχημάτων

1.1	Απλή γραφική αναπαράσταση του μοντέλου του YKN	18
1.2	Παραδείγματα Αρχιτεκτονικών YKN [24]	22
3.1	Παράδειγμα Αρχιτεκτονικής Πολυεπίπεδου Δικτύου [38]	32
3.2	Παράδειγμα Γράφου Δικτύου [38]	37
4.1	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α με 50 χρήστες και Poisson κατανομή άφιξης εργασιών	43
4.2	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α με 50 χρήστες και ομοιόμορφη κατανομή άφιξης εργασιών	43
4.3	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α με 150 χρήστες και Poisson κατανομή άφιξης εργασιών	44
4.4	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α με 150 χρήστες και ομοιόμορφη κατανομή άφιξης εργασιών	44
4.5	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α και για όλους τους συνδυασμούς των χρηστών (50,100,150,200) αθροιστικά με Poisson κατανομή άφιξης εργασιών	45
4.6	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α και για όλους τους συνδυασμούς των χρηστών (50,100,150,200) αθροιστικά με ομοιόμορφη κατανομή άφιξης εργασιών	45
4.7	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α , για όλους τους συνδυασμούς χρηστών (50,100,150,200) και για τις δύο διαφορετικές κατανομές άφιξης εργασιών (Poisson,ομοιόμορφη) αθροιστικά	46
4.8	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους με 50 χρήστες και Poisson κατανομή άφιξης εργασιών	47
4.9	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους με 50 χρήστες και ομοιόμορφη κατανομή άφιξης εργασιών	48
4.10	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους με 150 χρήστες και Poisson κατανομή άφιξης εργασιών	48
4.11	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους με 150 χρήστες και ομοιόμορφη κατανομή άφιξης εργασιών	49
4.12	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους για όλους τους συνδυασμούς χρηστών (50,100,150,200) και για τις δύο διαφορετικές κατανομές άφιξης εργασιών (Poisson,ομοιόμορφη) αθροιστικά	49
4.13	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για μεταβαλλόμενο Αριθμό Χρηστών με Poisson κατανομή άφιξης εργασιών	51
4.14	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για μεταβαλλόμενο Αριθμό Χρηστών με ομοιόμορφη κατανομή άφιξης εργασιών	51
4.15	Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για μεταβαλλόμενο Αριθμό Χρηστών	52
4.16	Μέσος Χρόνος Απόκρισης Τοπικής MYN και απομακρυσμένου YN για μεταβαλλόμενο Αριθμό Χρηστών	52

4.17 Μέσος Χρόνος Απόκρισης Τοπικής ΜΥΝ και απομακρυσμένου ΥΝ για μεταβαλλόμενο Αριθμό Χρηστών με Poisson (4.17a) και ομοιόμορφη (4.17b) κατανομή άφιξης εργασιών	53
4.18 Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας με 150 χρήστες και μεταβαλλόμενο αριθμό διαθέσιμων Σημείων Πρόσβασης	55
4.19 Μέσος Χρόνος Απόκρισης Τοπικής ΜΥΝ με 150 χρήστες και μεταβαλλόμενο αριθμό διαθέσιμων Σημείων Πρόσβασης	55
4.20 Μέσος Χρόνος Απόκρισης Τοπικής ΜΥΝ με 50 χρήστες και μεταβαλλόμενο Αριθμό Πυρήνων Επεξεργαστή της Τοπικής ΜΥΝ	57
4.21 Μέσος Χρόνος Απόκρισης Τοπικής ΜΥΝ με 200 χρήστες και μεταβαλλόμενο Αριθμό Πυρήνων Επεξεργαστή της Τοπικής ΜΥΝ	57
A.1 Βασικές Έννοιες	66

Κεφάλαιο 1

Εισαγωγή

1.1 Θεμελιώδεις ορισμοί

Ο όρος Υπολογισμός Κινητού-Νέφους (YKN – Mobile Cloud Computing) ακούγεται ολοένα και περισσότερο στις μέρες μας καθώς η τεχνολογία εξελίσσεται. Υπάρχουν πολλές θεωρήσεις για το τι είναι ο YKN. Θα δούμε πρώτα τι είναι γενικότερα ο Υπολογισμός Νέφους (YN – Cloud Computing).

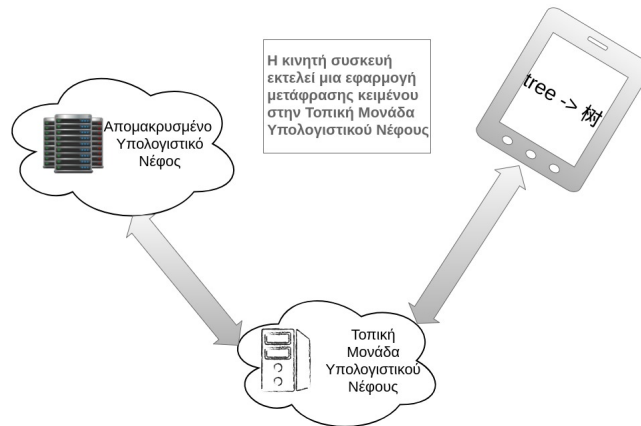
Ο YN είναι ένα μοντέλο το οποίο δίνει εύκολη κατ' απαίτηση πρόσβαση μέσω δικτύου στο χρήστη σε μια διαμοιραζόμενη υποδομή που παρέχει ρυθμιζόμενους υπολογιστικούς πόρους και υπηρεσίες, πχ. δίκτυα, εξυπηρετητές, χώρους αποθήκευσης, εφαρμογές, οι οποίοι μπορούν γρήγορα να του παρασχεθούν αλλά και να αποδεσμευθούν εύκολα με ελάχιστη προσπάθεια ή αλληλεπίδραση με τον πάροχο. Όμως ο όρος YKN εμπεριέχει και την έννοια του Υπολογισμού Κινητού (YK - Mobile Computing). Ο YK είναι ένα σύνολο τεχνολογιών, προϊόντων, υπηρεσιών και διαδικασιών πληροφορικής που επιτρέπουν στους χρήστες να έχουν πρόσβαση σε υπολογιστές, πχ κάποια φορητή συσκευή με υπολογιστική ισχύ, πληροφορίες και υπολογιστικούς πόρους και δυνατότητες ενώ είναι εν κινήσει.

Ο YKN έχει τις ρίζες του στη διεπιστημονική έρευνα στους κλάδους του YK και του YN. Η ήδη υπάρχουσα έρευνα επιδιώκει να σπάσει τα «φράγματα» μεταξύ αυτών των επιστημονικών τομέων εφαρμόζοντας λύσεις και τεχνικές του YN πάνω σε εφαρμογές του YK. Η ένωση, λοιπόν, των δύο αυτών κλάδων «γέννησε» αυτό που λέμε YKN [28].

1. Συνήθως, ο όρος YKN χρησιμοποιείται για ένα σύστημα που δίνει τη δυνατότητα εκτέλεσης μιας εφαρμογής για κινητή συσκευή, όπως το Gmail της Google, σε ένα απομακρυσμένο εξυπηρετητή με πολλούς υπολογιστικούς πόρους, πχ. Google servers, ενώ την ίδια στιγμή η κινητή συσκευή λειτουργεί σαν απλός πελάτης που συνδέεται μέσω τεχνολογίας 3G στον απομακρυσμένο εξυπηρετητή υπηρεσιών. Μερικά άλλα παραδείγματα του ίδιου τύπου είναι οι υπηρεσίες αναγνώρισης τοποθεσίας του Facebook¹, το Twitter² για κινητές συσκευές, οι εφαρμογές για τα προγνωστικά του καιρού κτλ.
2. Μια άλλη προσέγγιση είναι να θεωρήσουμε τις ίδιες τις κινητές συσκευές σαν παρόχους πόρων στο Νέφος δημιουργώντας έτσι ένα διομότιμο δίκτυο(peer2peer network). Έτσι, οι συσσωρευμένοι πόροι των διάφορων κινητών συσκευών στην τοπική περιοχή και άλλες στατικές συσκευές, εάν είναι διαθέσιμες, μπορούν να χρησιμοποιηθούν συλλογικά. Αυτή η προσέγγιση υποστηρίζει την κινητικότητα του χρήστη και αναγνωρίζει τη δυνατότητα στα κινητά νέφη (mobile clouds) να λαμβάνουν ερεθίσματα, πχ. από αισθητήρες, και να αλληλεπιδρούν με το περιβάλλον τους σαν σύνολο.
3. Η έννοια της Μονάδας Υπολογιστικού Νέφους (MYN-cloudlet) [Satyanarayanan] [33], είναι μια εκδοχή του YKN. Στην προσέγγιση αυτή, η κινητή συσκευή μεταφέρει το φόρτο εργασίας της στην τοπική MYN, η οποία απαρτίζεται από μερικούς εξυπηρετητές με επιπλέον σύνδεση YN. Αυτό γίνεται πιο ξεκάθαρα στον παρακάτω εικόνα [11]:

¹ Facebook: www.facebook.com

² Twitter: www.twitter.com



Σχήμα 1.1: Απλή γραφική αναπαράσταση του μοντέλου του YKN

Η τελευταία τεχνική είναι αρκετά διαδεδομένη και χρησιμοποιείται πολλές φορές για τον πειραματισμό και την επίλυση των προβλημάτων που έχουν να κάνουν με τους μηχανισμούς λειτουργίας αλλά και την αποδοτικότητα του YKN. Αυτή την τεχνική θα χρησιμοποιήσουμε κι εμείς στους πειραματισμούς μας στη συνέχεια της παρούσας διπλωματικής εργασίας.

1.2 Χαρακτηριστικά και ενδεικτικά πλεονεκτήματα του YKN

Ο YKN επιχειρεί να παρέχει στο χρήστη μια καλή εμπειρία χρήσης ανεξάρτητα της θέσης και των συνθηκών στις οποίες βρίσκεται αντιμετωπίζοντας πολλούς περιορισμούς, τόσο των συσκευών όσο και των δικτύων, που την δυσχεραίνουν. Πιο συγκεκριμένα, με τη χρήση του μοντέλου του YKN οι διεργασίες μιας κινητής συσκευής μπορούν είτε να εκτελεστούν τοπικά με τους δικούς της υπολογιστικούς πόρους είτε να ανατεθούν σε κάποιο πλούσιο σε πόρους μακρινό νέφος είτε να ανατεθούν σε κάποια, ισχυρότερη από την ίδια σε πόρους, υποδομή, πχ άλλες κινητές συσκευές, τοπικές μονάδες υπολογιστικού νέφους, που βρίσκεται πιο κοντά της. Με κατάλληλο αλγόριθμο αποφασίζεται σε κάθε σύστημα YKN, ποια από τις τρεις τεχνικές θα εφαρμοστεί για κάθε εργασία σε κάθε περίπτωση ανάλογα με το σκοπό που θέλει να επιτύχει. Έχοντας όλες αυτές τις επιλογές μπορούμε να επιτύχουμε τα ακόλουθα:

1.2.1 Δυνατότητα για μείωση στο χρόνο εκτέλεσης – πιο εύκολη πλήρωση των απαιτήσεων των υπηρεσιών (SLAs - Service Level Agreements)

Πολλά από τα κομμάτια των εφαρμογών μιας κινητής συσκευής απαιτούν αρκετούς υπολογισμούς αλλά ταυτόχρονα τα αποτελέσματα αυτών πρέπει να είναι διαθέσιμα σε συγκεκριμένο χρόνο, συνήθως χαμηλό, ώστε να επιτυγχάνεται σωστή αλληλεπίδραση με το χρήστη μέσω της εφαρμογής. Αυτοί οι υπολογισμοί ανατίθενται συνήθως στα μακρινά νέφη που έχουν μεγάλη επεξεργαστική ισχύ και δίνουν γρήγορα αποτέλεσμα, καθώς η συσκευή μόνη της δε μπορεί να δώσει αποτέλεσμα στον αναμενόμενο χρόνο. Ωστόσο, στον YKN με τις τοπικές υποδομές, στις οποίες μπορεί να ανατεθεί το εν λόγω κομμάτι της εφαρμογής, μπορούμε να έχουμε ακόμα πιο γρήγορα αποτέλεσμα. Αν και η υπολογιστική ισχύ των τοπικών υποδομών στατιστικά είναι χαμηλότερη από αυτή των απομακρυσμένων νεφών, κερδίζουμε χρόνο μέσω της ταχύτητας μετάδοσης των δεδομένων. Έχουμε με άλλα λόγια χαμηλότερη καθυστέρηση στην αποστολή και λήψη δεδομένων. Αν έχουμε δηλαδή μεγάλο όγκο δεδομένων που πρέπει να αποσταλούν σίγουρα η χρήση των τοπικών υποδομών θα μας μειώσει το χρόνο εκτέλεσης που βλέπει ο χρήστης και έτσι οι απαιτήσεις κάθε εργασίας-υπηρεσίας θα πληρούνται ευκολότερα.

1.2.2 Δυνατότητα για μείωση στην κατανάλωση ενέργειας

Με χρήση του μοντέλου YKN επιτυγχάνεται σημαντική μείωση στην κατανάλωση ενέργειας. Οι κινητές συσκευές συχνά έχουν να εκτελέσουν αρκετές ενεργοβόρες εργασίες. Σε ένα περιβάλλον YKN, αυτές οι διεργασίες μπορούν να μεταφορτωθούν και να εκτελεστούν σε κάποια απομακρυσμένη υπολογιστική υποδομή. Με αυτόν τον τρόπο, η συσκευή απλά παραμένει αδρανής και αναμένει να λάβει τα αποτελέσματα αυτών των ενεργοβόρων διεργασιών από τις εξωτερικές υπολογιστικές υποδομές, από όπου μεταφορτώθηκαν. Έτσι, η μπαταρία επιβαρύνεται μόνο με το ενεργειακό κόστος για τη σύνδεση μέσω δικτύου με τις υποδομές και την αποστολή των δεδομένων σε αυτές, καταναλώνοντας πολύ λιγότερη ενέργεια.

Ειδικότερα η μετάδοση των δεδομένων στο μακρινό νέφος γίνεται μέσω 3G/4G ενώ στις τοπικές MYN μέσω WiFi. Η κατανάλωση ενέργειας κατά τη διάρκεια της χρήσης του δικτύου 3G/4G είναι αρκετά μεγαλύτερη από αυτή του WiFi. Επομένως, όταν αναθέτουμε μια εργασία στις τοπικές MYN, αν και αυξάνεται ο χρόνος που η συσκευή παραμένει αδρανής, έχουμε πολλές φορές ακόμη μεγαλύτερη εξοικονόμηση ενέργειας από τη μετάδοση των δεδομένων.

1.2.3 Μείωση του κόστους από πλευράς καταναλωτή

Ένα τελευταίο κομμάτι, που όμως πρέπει να ληφθεί υπόψιν, είναι το χρηματικό κόστος. Όταν επιλέγεται η χρήση των τοπικών MYN έναντι του απομακρυσμένου νέφους, οπότε και γίνεται κυρίως χρήση του δικτύου WiFi, ο χρήστης δέχεται χαμηλότερες χρεώσεις (σε γενική περίπτωση) από τον πάροχο του δικτύου σε σχέση με την περίπτωση του δικτύου 3G/4G, όπου η χρέωση για τη μετάδοση δεδομένων είναι αρκετά μεγαλύτερη. Γενικότερα, με τον YKN μπορούμε να κάνουμε εκτιμήσεις για το ενεργειακό και το χρηματικό κόστος και την αποδοτικότητα κάθε εργασίας και να την αναθέσουμε στην κατάλληλη υποδομή ανάλογα με το τι έχει θέσει ως προτεραιότητα ο χρήστης κάθε φορά όπως για παράδειγμα απόδοση, εξοικονόμηση ενέργειας, χρηματικό κόστος.

1.3 Εφαρμογές ανάδειξης χρήσης του YKN

Ο YKN έχει πληθώρα εφαρμογών στις οποίες μπορεί να φανεί ιδιαίτερα χρήσιμος στην καθημερινότητα. Στο κομμάτι αυτό θα παρατεθούν ενδεικτικά κάποιες από αυτές [11, 28].

1.3.1 Εκπαίδευση με χρήση κινητών συσκευών

Η εκπαίδευση με χρήση κινητών συσκευών είναι πρακτικά η διαδικασία κατά την οποία γίνεται χρήση της κινητής τεχνολογίας (mobile technology) ώστε να βελτιώσει ή να ενισχύσει τις γνώσεις, τις ικανότητες και τη συμπεριφορά του ατόμου μέσω κατάρτισης και εκπαίδευσης οποτεδήποτε και οπουδήποτε. Ο τρόπος εκπαίδευσης αυτός, εισάγει ένα νέο επίπεδο φορητότητας στη διαδικασία της μάθησης μέσω της χρήσης των tablets και των «έξυπνων» συσκευών. Έτσι επιτυγχάνεται περισσότερη διαδραστικότητα μεταξύ εκπαιδευτή-εκπαιδευόμενου αλλά και μεταξύ των ίδιων των εκπαιδευόμενων, πράγμα που βελτιώνει την αποδοτικότητα και την ευκολία στη διαδικασία της μάθησης.

1.3.2 Εφαρμογές Υγείας και Ευζωίας

Μέσω εφαρμογών που θα χρησιμοποιούν την τεχνολογία του YKN γίνεται ευκολότερη η πρόληψη και αντιμετώπιση πολλών ασθενειών. Χρήστες, όπου και αν βρίσκονται και οποιαδήποτε στιγμή, μπορούν να στέλνουν δεδομένα για συμπτώματα, παρενέργειες και άλλα στοιχεία σχετικά με κάποια πάθηση ή κάποια φαρμακευτική αγωγή, τα οποία με κατάλληλη επεξεργασία και παρακολούθηση από επιστήμονες υγείας μπορεί – αν όχι σίγουρα – να φανούν ιδιαίτερα χρήσιμα σε μελέτες που θα οδηγήσουν στην εύρεση τρόπου αντιμετώπισης ή βελτίωσης των ασθενειών. Παράδειγμα τέτοιου συστήματος αποτελεί το (MoCAsH[16]).

1.3.3 Επεξεργασία Εικόνας και Φυσικής Γλώσσας

Πολλές εφαρμογές που κάνουν επεξεργασία εικόνας και ήχου με χρήση αναγνώρισης προτύπων μπορούν επίσης να επωφεληθούν από τη χρήση του ΥΚΝ. Γνωρίζουμε ότι πολλές φορές τέτοιες διαδικασίες απαιτούν πολλούς υπολογισμούς, οι οποίοι ενδέχεται να μη μπορούν να εκτελεστούν στον επιθυμητό χρόνο από την κινητή συσκευή του χρήστη. Χαρακτηριστικό παράδειγμα είναι η επιθυμία κάποιου ταξιδιώτη σε ξένη χώρα, όπου η τοπική γλώσσα δεν του είναι οικεία, να έχει τη μετάφραση όσων διαβάζει ή ακούει, για παράδειγμα σε μία ξενάγηση σε ένα μουσείο. Αυτό για να το επιτύχει με χρήση των 3G/4G δικτύων θα του κόστιζε αρκετά. Ωστόσο, αν έχει τους μηχανισμούς διαθέσιμους, μπορεί να αιτηθεί πόρους από συνταξιδιώτες του που έχουν την ίδια επιθυμία με αυτόν, και όλοι μαζί συνεργατικά να εκτελέσουν με επιτυχία αυτούς τους υπολογισμούς, δημιουργώντας πρακτικά ένα αυτοοργανούμενο (ad-hoc) δίκτυο που θα επιτυγχάνει την κάλυψη των αναγκών του.

1.3.4 Αναζήτηση και Αναπαραγωγή Πολυμέσων

Οι κινητές συσκευές πέραν από τη δυνατότητα της επικοινωνίας, πλέον μπορούν και περιέχουν πολλά δεδομένα πολυμέσων, όπως τραγούδια, φωτογραφίες, βίντεο κτλ. Με την τεχνολογία του ΥΚΝ κάποιος χρήστης που αναζητά δεδομένα, όπως πχ. ένα τραγούδι μέσω κάποια εφαρμογής αναγνώρισης, όπως το Shazam³, θα μπορεί αντί να ψάχνει αυτά τα δεδομένα σε κάποιο απομακρυσμένο κέντρο δεδομένων, να τα ζητά από τις κοντινές του συσκευές και αυτές να του τα παρέχουν ευκολότερα και χωρίς μεγάλη καθυστέρηση. Προφανώς εάν αυτά δεν υπάρχουν η αναζήτηση θα μεταβαίνει αναγκαστικά στο απομακρυσμένο κέντρο δεδομένων όπου θα βρίσκεται σίγουρα αυτό που αναζητά.

1.3.5 Εφαρμογές Επεξεργασίας Δεδομένων που λαμβάνονται από Αισθητήρες

Στη σημερινή εποχή όλες σχεδόν οι κινητές συσκευές είναι εξοπλισμένες με πληθώρα αισθητήρων όπως GPS, επιταχυνσιόμετρο, αισθητήρα φωτός, μικρόφωνο, θερμομέτρο, ρολόι και πυξίδα. Με την τεχνολογία του ΥΚΝ οι μετρήσεις που λαμβάνονται από τις συσκευές έχουν τη δυνατότητα να μεταφέρονται και να είναι διαθέσιμες σε γειτονικές κινητές συσκευές με αποτέλεσμα οι χρήστες να μπορούν να έχουν χρήσιμες πληροφορίες πχ σχετικά με την κίνηση στους δρόμους, εύρεση της ταχύτερης διαδρομής προς έναν προορισμό, την κατάσταση που βρίσκεται μια γειτονική τους περιοχή όσον αφορά τη θερμοκρασία κ.α.

1.3.6 Μέσα κοινωνικής δικτύωσης σε κινητές συσκευές

Πλέον τα μέσα κοινωνικής δικτύωσης έχουν κατακλύσει την καθημερινότητα του ανθρώπου. Η επικοινωνία και γενικότερα η αλληλεπίδραση με τον υπόλοιπο κόσμο γίνεται σε μεγάλο βαθμό μέσω διαδικτυακών πλατφορμών. Κατά τη διαδικασία αυτής της αλληλεπίδρασης ανταλλάσσονται πολλά μηνύματα με οπτικοακουστικό περιεχόμενο. Η εισαγωγή της τεχνολογίας του ΥΚΝ στον τομέα αυτόν θα επέφερε τη δημιουργία περισσότερων μικρών διομήτιμων δικτύων (peer2peer - p2p) όπου η ανταλλαγή δεδομένων, είτε είναι απλά μηνύματα κειμένου είτε είναι πολυμέσα, θα είναι αμεσότερη. Αυτό θα επιφέρει και μια ελάφρυνση στον τεράστιο όγκο δεδομένων που μέχρι τώρα αποθηκεύονται στο απομακρυσμένο κέντρο δεδομένων, καθώς μέρος αυτών θα μπορεί να βρίσκεται τοπικά.

1.4 Αρχιτεκτονικές για τον ΥΚΝ

Στο κομμάτι αυτό θα εξετάσουμε τις διαφορετικές αρχιτεκτονικές που έχουν, όπως φαίνονται και στο σχήμα 1.2, παρουσιασθεί και αφορούν το κομμάτι του ΥΚΝ [24].

³ Shazam: www.shazam.com

1.4.1 Υπολογισμός στα άκρα του δικτύου (ΥΑΔ – Mobile Edge Computing)

Σαν ένα παράδειγμα αρχιτεκτονικής του υπολογισμού στα άκρα (edge computing), ο ΥΑΔ [1] έχει προσελκύσει αρκετό ερευνητικό ενδιαφέρον. Θεωρείται ένας κύριος παράγοντας για την σύγχρονη εξέλιξη των κυψελωτών σταθμών βάσης. Αξιοποιεί τους εξυπηρετητές στην άκρη του δικτύου (edge servers) και τους κυψελωτούς σταθμούς βάσης ώστε να δουλεύουν συνεργατικά. Στο παράδειγμα της αρχιτεκτονικής αυτής μπορεί να υπάρχει σύνδεση με κάποιο απομακρυσμένο κέντρο δεδομένων ή/και κάποιο νέφος με υπολογιστικές υποδομές. Έτσι, μαζί με τις κινητές συσκευές των χρηστών που βρίσκονται στην άκρη του δικτύου, ο ΥΑΔ επιτρέπει την ανάπτυξη εφαρμογών δύο ή τριών ιεραρχικών επιπέδων στο δίκτυο. Ακόμα, ο ΥΑΔ στοχεύει στην προσαρμοστική και ταχύτερη εκκίνηση των κυψελωτών υπηρεσιών για τους πελάτες και ενισχύει την αποδοτικότητα του δικτύου. Στη σημερινή εποχή έχουν σημειωθεί σημαντικές βελτιώσεις στον ΥΑΔ ώστε να υποστηρίζει τη χρήση δικτύου επικοινωνιών 5G. Τέλος, στοχεύει στην πιο ευέλικτη πρόσβαση στις πληροφορίες του ραδιοφωνικού δικτύου (radio network) με σκοπό το διαμοιρασμό πληροφοριών και την ανάπτυξη εφαρμογών.

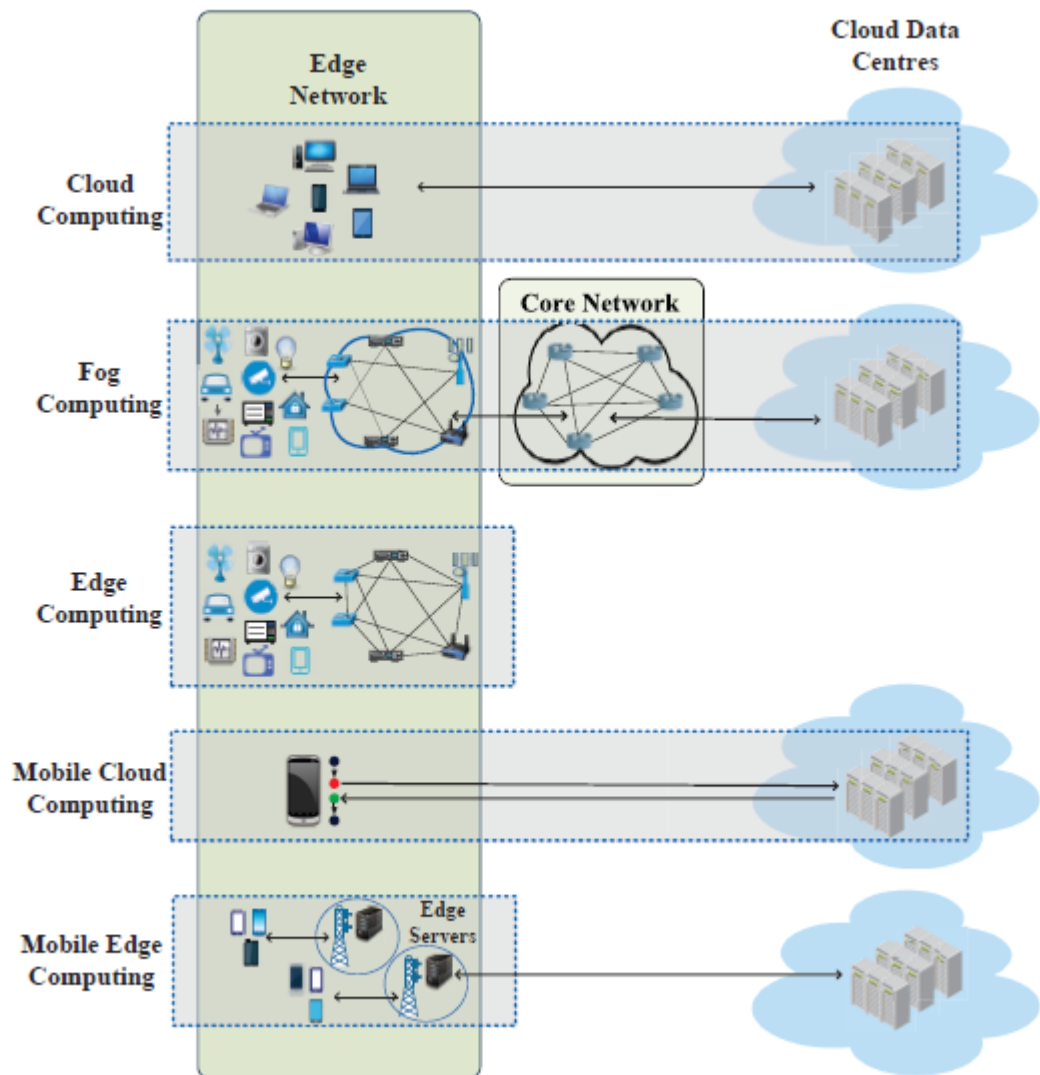
1.4.2 Υπολογισμός Κινητού Νέφους (ΥΚΝ – Mobile Cloud Computing)

Ο ΥΚΝ [10] είναι άλλη μία σύγχρονη τάση στην παροχή ηλεκτρονικών υπηρεσιών. Εξαιτίας του πολλαπλασιασμού των «έξυπνων» κινητών συσκευών, στη σημερινή εποχή οι χρήστες προτιμούν να εκτελούν τις απαραίτητες εφαρμογές σε αυτές τις φορητές συσκευές αντί στους υπολογιστές, όπως παραδοσιακά γινόταν ως τώρα. Ωστόσο, οι περισσότερες από αυτές τις συσκευές υπόκεινται σε περιορισμούς όσον αφορά την ενέργεια, το χώρο αποθήκευσης δεδομένων και τους υπολογιστικούς πόρους. Για αυτό, σε κρίσιμα σενάρια, είναι πιο πρακτικό να εκτελούνται οι εφαρμογές που εμπεριέχουν εντόνους και απαιτητικούς υπολογισμούς εκτός των φορητών συσκευών, λόγω των περιορισμών που έχουν αυτές. Σε αυτήν την περίπτωση, ο ΥΚΝ παρέχει τους απαραίτητους υπολογιστικούς πόρους για την υποστήριξη μιας τέτοιας απομακρυσμένης εκτέλεσης της μεταφορτωμένης εφαρμογής της κινητής συσκευής σε «κοντινή» απόσταση από τον τελικό χρήστη και όχι απαραίτητα σε κάποιο απομακρυσμένο κέντρο δεδομένων. Στο μοντέλο του ΥΚΝ, συχνά μικρής επεξεργαστικής ισχύος εξυπηρετητές, που αποκαλούνται ΜΥΝ, εγκαθίστανται στην άκρη του δικτύου. Αυτοί, μαζί με τις φορητές συσκευές των τελικών χρηστών και τα κέντρα δεδομένων του νέφους συνιστούν μια πλατφόρμα ανάπτυξης εφαρμογών τριών ιεραρχικών επιπέδων. Εν συντομία, το μοντέλο του ΥΚΝ συνδυάζει τον ΥΝ, τον ΥΚ και τις ασύρματες επικοινωνίες για να ενισχύσει την εμπειρία χρήσης των χρηστών των φορητών συσκευών και δημιουργεί νέες επιχειρηματικές ευκαιρίες τόσο για τους παρόχους των δικτύων όσο και για τους παρόχους υπηρεσιών μέσω του νέφους.

1.4.3 Υπολογισμός Ομίχλης (ΥΟ-Fog Computing)

Όπως ο ΥΑΔ και ο ΥΚΝ, ο ΥΟ [3] αποτελεί άλλη μία αρχιτεκτονική που εκμεταλλεύεται τους υπολογιστικούς πόρους των υποδομών στα άκρα του δικτύου. Όμως, πέρα από το δίκτυο μεταξύ των συσκευών που βρίσκονται στα άκρα, το μοντέλο του ΥΟ μπορεί να επεκταθεί και στο κεντρικό δίκτυο επίσης. Πιο συγκεκριμένα, τόσο οι υποδομές στα άκρα του δικτύου όσο και οι υποδομές του κεντρικού δικτύου, πχ. δρομολογητές, τοπικοί εξυπηρετητές κ.α., μπορούν να χρησιμοποιηθούν σαν υπολογιστοί πόροι. Κατά συνέπεια, είναι εφικτή η ανάπτυξη εφαρμογών πολλών επιπέδων και η μείωση της αίτησης υπηρεσιών από τεράστιο αριθμό συσκευών που ανήκουν στο Διαδίκτυο Αντικειμένων (ΔΑ - Internet of Things - IoT). Επίσης, στο μοντέλο αυτό η υποδομή στα άκρα του δικτύου μπορεί να τοποθετηθεί εγγύτερα στις συσκευές ΔΑ και στους αισθητήρες σε σχέση με τις ΜΥΝ ή τους κυψελωτούς εξυπηρετητές στην άκρη του δικτύου (cellular edge servers). Αφού, λοιπόν, οι συσκευές ΔΑ και οι αισθητήρες είναι πυκνά κατανομημένες και απαιτούν απαντήσεις σε πραγματικό χρόνο όταν κάνουν κάποια αίτηση υπηρεσίας, η προσέγγιση αυτή καθιστά δυνατή την επεξεργασία και αποθήκευση των δεδομένων κοντά στη συσκευή ΔΑ ή τον αισθητήρα. Ως αποτέλεσμα, θα μειωθεί σημαντικά η καθυστέρηση της παράδοσης των αποτελεσμάτων της εκτέλεσης μιας

εφαρμογής που απαιτεί αποτελέσματα σε πραγματικό χρόνο. Σε αντίθεση με τον ΥΑΔ, το μοντέλο του ΥΟ μπορεί να επεκτείνει υπηρεσίες που βασίζονται στο νέφος, όπως IaaS (Infrastructure as a Service), PaaS (Platform as a Service), SaaS (Software as a Service)⁴ κτλ, στα άκρα του δικτύου.



Σχήμα 1.2: Παραδείγματα Αρχιτεκτονικών ΥΚΝ [24]

1.5 Ερευνητικά Θέματα

Ο ΥΚΝ αποτελεί μια αρκετά σύγχρονη τεχνική στην πληροφορική. Στην ενότητα αυτή αναφέρονται ενδεικτικά κάποιες σημαντικές περιοχές που έχουν μεγάλο ερευνητικό ενδιαφέρον [11, 28].

1.5.1 Λειτουργικό κόστος

Αν και ο ΥΚΝ μπορεί να είναι αρκετά αποδοτικός, δεν παρέχεται δωρεάν στους χρήστες. Υπάρχει κόστος τόσο από τη χρήση δικτύου όσο και από την «ενοικίαση» υπολογιστικών πόρων εκτός της κινητής συσκευής. Πρέπει να γίνουν, λοιπόν, προσεγγίσεις ώστε να διερευνηθεί η σχέση κό-

⁴ IaaS,PaaS,SaaS: <https://apprenda.com/library/paas/iaas-paas-saas-explained-compared/>

στους – απόδοσης όταν λαμβάνεται μια απόφαση για τη μεταφόρτωση μιας εφαρμογής από την κινητή συσκευή σε κοντινή ή μακρινή υπολογιστική υποδομή.

1.5.2 Σύγκριση μεταξύ μεθοδολογιών μεταφόρτωσης σε διαφορετικές εφαρμογές

Ένα ακόμα θέμα που πρέπει να μελετηθεί είναι ποια τεχνική μεταφόρτωσης, βασισμένη στην επικοινωνία πελάτη-εξυπηρετητή(client-server) ή στην εικονικοποίηση(virtualization), είναι πιο αποδοτική σε διαφορετικές καταστάσεις ή τύπους εφαρμογών.

1.5.3 Επίδραση της κινητικότητας του χρήστη

Περαιτέρω έρευνα πρέπει να γίνει στο πώς επηρεάζει τα συστήματα αυτά η κινητικότητα του χρήστη, έτσι ώστε να εξασφαλίζεται η σταθερότητα αλλά και η σωστή λειτουργία των μηχανισμών με σκοπό την ικανοποίηση των απαιτήσεων του χρήστη με τον καλύτερο δυνατό τρόπο. Πρέπει να είμαστε σίγουροι για τη σωστή εκτέλεση των υπολογισμών και τη γρήγορη αναπροσαρμογή του συστήματος, όταν ένας χρήστης εισέρχεται ή εξέρχεται από την εμβέλεια ενός δικτύου.

1.5.4 Προγραμματισμός-Ρυθμίσεις σε επίπεδο δικτύου

Ο προγραμματισμός σε επίπεδο δικτύου επίσης πρέπει να διερευνηθεί από την άποψη του ποιου ήδη υπάρχοντες αλγόριθμοι δρομολόγησης και σύνταξης πακέτων μπορούν να χρησιμοποιηθούν σε δίκτυα που υποστηρίζουν τον ΥΚΝ. Το πώς αυτοί οι αλγόριθμοι επηρεάζουν την ποιότητα υπηρεσιών, την καθυστέρηση, την αξιοπιστία, το χρηματικό κόστος και την κατανάλωση ενέργειας σε μια κινητή συσκευή χρήζει περαιτέρω έρευνας.

1.5.5 Προγραμματισμός Εφαρμογών

Σε προγραμματιστικό επίπεδο πρέπει να διερευνηθεί πως και πόσο πρέπει να αλλάξει ο κώδικας των εφαρμογών ώστε να μπορούν να εκμεταλλευτούν τα προτερήματα που προσφέρει ο ΥΚΝ. Δεν έχει νόημα η εκ νέου συγγραφή μιας εφαρμογής. Αυτό δε θα συνέφερε από άποψη χρόνου και κόστους. Επομένως πρέπει να γίνουν τα κατάλληλα έργα ώστε κάθε εφαρμογή να μπορεί να αξιοποιήσει στο έπακρο τις δυνατότητες που δίνει ο ΥΚΝ ώστε να εξυπηρετούνται οι απαιτήσεις της. Θέλουμε επομένως να υπάρχει ένας προσαρμοστικός τρόπος προγραμματισμού που να είναι μάλιστα σχετικά εύκολο να εφαρμοστεί στις ήδη υπάρχουσες εφαρμογές.

1.5.6 Ρυθμίσεις και πολιτικές για την ασφάλεια και την εξασφάλιση της ατομικότητας

Αρκετές εφαρμογές χρησιμοποιούν προσωπικά στοιχεία του χρήστη τους, ενίοτε σημαντικά ή ευαίσθητα. Επομένως, για τη διασφάλιση αυτών πρέπει να μελετηθεί ο τομέας της ασφάλειας όταν τέτοιες εφαρμογές εκτελούνται στο μοντέλο του ΥΚΝ. Μιας και αυτά τα δεδομένα ταξιδεύουν μέσα στα δίκτυα, σίγουρα θα αποτελέσουν πόλο έλξης για διαδικτυακές επιθέσεις (hacking). Τέτοια φαινόμενα θα πρέπει να προβλέπονται και να υπάρχουν μηχανισμοί αντιμετώπισής τους.

1.5.7 Διαχείριση μεγάλου όγκου Δεδομένων

Οι κινητές συσκευές στη σημερινή εποχή έχουν αρκετούς αισθητήρες, παράγουν και αξιοποιούν μεγάλο όγκο δεδομένων. Πολλές εφαρμογές επίσης χρειάζονται αρκετό όγκο δεδομένων ώστε να δουλεύουν σωστά. Επομένως, στον ΥΚΝ είναι αρκετά σημαντικό να βρεθούν τρόποι για τη γρήγορη και αποδοτική διαχείριση όταν πρόκειται για μεγάλο όγκο δεδομένων, καθώς και αρκετός χώρος για την αποθήκευση αυτών όποτε χρειάζεται. Ακόμα, πρέπει να βρεθούν κατάλληλοι τρόποι δέσμευσης των υπολογιστικών πόρων όταν έχουμε τέτοιες απαιτητικές εφαρμογές με διαχείριση μεγάλου όγκου δεδομένων.

1.5.8 Μεταφόρτωση Εφαρμογών με στόχο την ελαχιστοποίηση της καταναλισκόμενης ενέργειας και/ή του χρόνου εκτέλεσης

Σημαντική προσοχή θα πρέπει να δοθεί στην ανάπτυξη αλγορίθμων για τη λήψη αποφάσεων στα συστήματα ΥΚΝ. Το πότε μια εφαρμογή ή ένα τμήμα αυτής θα μπορεί, θα πρέπει ή συμφέρει το χρήστη να δοθεί σε μια κοντινή (ΜΥΝ) ή μακρινή (υπολογιστικό νέφος) υπολογιστική υποδομή προς εκτέλεση, το ποιο θα είναι το βασικό κριτήριο, πχ. η κατανάλωση ενέργειας, ο χρόνος εκτέλεσης ή το χρηματικό κόστος, για αυτή την απόφαση και εάν είναι εφικτή η μεταφόρτωση που ζητήθηκε είναι ερωτήματα στα οποία απαντήσεις πρέπει να δοθούν μέσω περαιτέρω έρευνας στους αλγορίθμους λήψης αποφάσεων σε υπολογιστικό περιβάλλον που εφαρμόζει το μοντέλο του ΥΚΝ. Έναν τέτοιο αλγόριθμο εξετάζουμε και στη συνέχεια της παρούσας εργασίας.

Κεφάλαιο 2

Η έννοια της Μεταφόρτωσης Υπολογισμών στον ΥΚΝ

2.1 Μεταφόρτωση Υπολογισμών

Η Μεταφόρτωση Υπολογισμών ΜΥ (Computation Offloading) είναι μια μέθοδος με την οποία οι διεργασίες (tasks) των κινητών συσκευών, οι οποίες απαιτούν πολλούς υπολογιστικούς πόρους, μεταφέρονται για εκτέλεση σε κάποιον υπολογιστή, πχ. εξυπηρετητή, με μεγαλύτερη επεξεργαστική ισχύ (CPU, μνήμη, αποθηκευτικό χώρο). Η μεταφόρτωση στα συστήματα ΥΚΝ είναι διαφορετική από τον μηχανισμό του μοντέλου της μετανάστευσης (migration) που χρησιμοποιείται στα πολυεπεξεργαστικά συστήματα και σε συστήματα υπολογισμού πλέγματος (grid computing) και υπολογισμού νέφους (cloud computing), στα οποία μια διαδικασία μπορεί να μεταφορτωθεί αλλού για εξισορρόπηση φόρτου εργασίας. Η βασική διαφορά με την εν λόγω τεχνική είναι ότι η μεταφόρτωση σε συστήματα ΥΚΝ μπορεί να γίνει και σε υπολογιστικές υποδομές που βρίσκονται εκτός του άμεσου-κοντινού περιβάλλοντος της ίδιας της κινητής συσκευής. Ακόμη, στο μοντέλο του υπολογισμού πλέγματος εκτελούνται κατά βάση βαριές εφαρμογές κεντροποιημένα με παράλληλη επεξεργασία ενώ το μοντέλο του ΥΚΝ αναφέρεται σε πιο απλές εφαρμογές κινητών συσκευών. Επομένως, μιλάμε για ένα διαφορετικό μοντέλο από τα ήδη υπάρχοντα. Τα αρχικά ερευνητικά προβλήματα της ΜΥ ήταν η εύρεση αλγορίθμων που να την κάνουν εφικτή και συμφέρουσα. Μετέπειτα, αφού υπήρξαν σημαντικές εξελίξεις στον τομέα της τεχνολογίας γενικότερα, κύρια σημεία έρευνας έγιναν το πότε και αν συμφέρει ένα χρήστη σε ένα σύστημα κινητών συσκευών να μεταφορτώνει μια διαδικασία, το που θα τη μεταφορτώνει και στην ανάπτυξη κατάλληλων υποδομών που να υλοποιούν τα παραπάνω.

2.2 Τεχνικές Μεταφόρτωσης

Η μεταφόρτωση απαιτεί να υπάρχει άμεση πρόσβαση σε υποδομές με αρκετούς διαθέσιμους υπολογιστικούς πόρους. Συνήθως, χρησιμοποιείται εικονικοποίηση (virtualization) για την εξασφάλιση της απομόνωσης και της ασφάλειας. Έχουν ήδη αναπτυχθεί διάφορες τεχνικές με τις οποίες επιτυγχάνεται η μεταφόρτωση σε τέτοιες υπολογιστικές υποδομές. Πιο συγκεκριμένα η ταξινόμηση των συστημάτων ΜΥ γίνεται σε επίπεδο μεθόδου (method) [30], εργασίας (task) [41], εφαρμογής (application) [39] ή εικονικής μηχανής (virtual machine) [6].

Στις δύο πρώτες τεχνικές, η κινητή συσκευή αποφασίζει να αποστέλλει ένα μέρος του εκτελέσιμου κώδικα στην εξωτερική υπολογιστική υποδομή μαζί με τις κατάλληλες παραμέτρους-δεδομένα που απαιτούνται. Στην περίπτωση της μεταφόρτωσης σε επίπεδο μεθόδου, πρακτικά κλωνοποιείται το αντικείμενο στην πλευρά του εξυπηρετητή και εκτελεί τις μεθόδους του εκεί και απλά ενημερώνει την κινητή συσκευή για τη νέα κατάσταση του αντικειμένου ή των αντικειμένων γενικότερα μετά την εκτέλεση των κατάλληλων μεθόδων κι έτσι η κινητή συσκευή επιστρέφει στο χρήστη το τελικό αποτέλεσμα των υπολογισμών που απαιτήθηκαν. Όταν έχουμε μεταφόρτωση σε επίπεδο εργασίας, ο κώδικας μια εφαρμογής διασπάται σε κομμάτια, εκ των οποίων κάποια θα μεταφορτωθούν και κάποια όχι. Κυρίως, έχουμε διάσπαση σε «εργασίες» οι οποίες με κατάλληλους μηχανισμούς αποστέλλονται στην εξωτερική υπολογιστική υποδομή, εκτελούνται, διατάσσονται με τη σωστή σειρά (serialization) και επιστρέφονται τα αποτελέσματα στην κινητή συσκευή όπου συν-

δυάζονται με όσες εργασίες έγιναν τοπικά και επιστρέφουν τα αποτελέσματα στο χρήστη. Είναι προφανές πως στα παραπάνω εμπλέκονται αρκετοί μηχανισμοί για το διαχωρισμό του προγράμματος, τη λήψη απόφασης για τη μεταφόρτωση και την αποστολή και λήψη των κατάλληλων δεδομένων – αποτελεσμάτων.

Στην περίπτωση μεταφόρτωσης σε επίπεδο εφαρμογής είναι πολύ σημαντικό να λαμβάνονται σωστές αποφάσεις για τη μεταφόρτωση ή μη, καθώς εδώ πρόκειται για ολόκληρη εφαρμογή και όχι για απλά μια μέθοδο ή μια εργασία αυτής. Μία μέθοδος που χρησιμοποιείται έχει ένα χρονομετρητή (timer), ο οποίος πρακτικά αποφασίζει. Η εφαρμογή εκκινείται στην κινητή συσκευή. Αν αυτή ολοκληρωθεί σε ένα προκαθορισμένο χρόνο απόκρισης τότε δεν απαιτείται μεταφόρτωση. Αν αυτό δε συμβεί, τότε στέλνεται η εφαρμογή μαζί με τα δεδομένα εισόδου στον απομακρυσμένο εξυπηρετητή, όπου και αυτή εκτελείται και τα αποτελέσματα της επιστρέφονται στο χρήστη. Προφανώς, η εμπλοκή αυτή του χρονομετρητή επιβαρύνει το σύστημα. Γι' αυτό θα πρέπει με κατάλληλα εργαλεία ή χειροκίνητα οι προγραμματιστές να λαμβάνουν τις παραμέτρους για τη μεταφόρτωση ή μη (πχ. μέγεθος δεδομένων, πλήθος εντολών εφαρμογής κ.α.). Έτσι, θα μπορούν να λαμβάνουν καλύτερες αποφάσεις και χωρίς μεγάλη επιβάρυνση στο σύστημα. Για παράδειγμα, αν ο αναμενόμενος χρόνος τερματισμού είναι γνωστό εκ των προτέρων ότι είναι μεγαλύτερος από τον απαιτούμενο χρόνο απόκρισης, η εφαρμογή πάντα θα μεταφορτώνεται.

Τέλος, η μεταφόρτωση με χρήση εικονικών μηχανών είναι αρκετά διαδεδομένη και έχει γίνει αρκετή έρευνα πάνω σε αυτή. Σε τέτοιες περιπτώσεις η κινητή συσκευή αρχικά δημιουργεί έναν κλώνο της στον απομακρυσμένο εξυπηρετητή με τον οποίο συγχρονίζεται περιοδικά ή κατ' απαίτηση. Η συσκευή μπαίνει σε μια διαδικασία αναμονής (sleeping process). Οι εφαρμογές-εργασίες εκτελούνται στον κλώνο (εικονική μηχανή) αυτόματα ή μετά από αίτηση της κινητής συσκευής. Αυτό γίνεται καθώς η εικονική μηχανή δεσμεύεται στον εξυπηρετητή και μεταφράζει ό,τι έλαβε από την κινητή συσκευή σε γλώσσα υλικού (hardware) που θα εκτελεστεί στην απομακρυσμένη υπολογιστική υποδομή. Μετά από την εκτέλεση, ο κλώνος επιστρέφει την κατάσταση του στην κινητή συσκευή, την επανεντάσσει στο σύστημα της δίνοντας έτσι τα αποτελέσματά των υπολογισμών και την «ξυπνάει» ώστε να συνεχίσει την εκτέλεση των διεργασιών της.

2.3 Κριτήρια και τρόποι λήψης Απόφασης για Μεταφόρτωση

Μεγάλο κομμάτι της έρευνας επικεντρώνεται στην εύρεση μεθοδολογιών απόφασης για το πότε και αν είναι συμφέρον να μεταφορτώνεται μια εφαρμογή ή μια εργασία σε ένα πλουσιότερο υπολογιστικό σύστημα αλλά και ποιο υπολογιστικό σύστημα θα είναι αυτό. Για τη λήψη των αποφάσεων αυτών υπάρχουν πολλές παράμετροι που οφείλουν να ελεγχθούν. Επίσης, ανάλογα με την περίπτωση δίνεται μεγαλύτερη βαρύτητα σε κάποιες εξ' αυτών [21].

2.3.1 Τρόποι λήψης απόφασης

Στατική Λήψη Απόφασης: Όταν η απόφαση λαμβάνεται στατικά, το πρόγραμμα διαχωρίζεται σε τμήματα κατάλληλα για το σκοπό αυτό κατά την συγγραφή του. Αυτή η τεχνική έχει το πλεονέκτημα να μην δεσμεύει επιπλέον υπολογιστικούς πόρους τοπικά κατά την εκτέλεση του προγράμματος. Ωστόσο, η προσέγγιση αυτή είναι έγκυρη όταν οι παράμετροι που επηρεάζουν το σύστημα είναι γνωστοί εκ των προτέρων. Τέτοιες παράμετροι είναι για παράδειγμα η ταχύτητα του επεξεργαστή της κινητής συσκευής, η ενέργεια που απαιτείται ώστε να στείλει και να λάβει δεδομένα η κινητή συσκευή μέσω δικτύου, η κατανάλωση ενέργειας σε κατάσταση αναμονής και η διαθέσιμη ενέργεια της κινητής συσκευής. Άλλες παράμετροι όμως, εξίσου σημαντικές, δε μπορούν να είναι γνωστές και διαθέσιμες κατά τη διάρκεια της ανάπτυξης του λογισμικού μιας εφαρμογής. Για παράδειγμα, η ταχύτητα των επεξεργασιών του εξυπηρετητή αλλάζει από πάροχο σε πάροχο, αν και η πλειοψηφία αυτών εγγυώνται στον πελάτη μια ελάχιστη υπολογιστική ισχύ, το εύρος ζώνης του δικτύου μέσω του οποίου θα μεταφερθούν τα απαραίτητα δεδομένα για τη μεταφόρτωση, ο όγκος των δεδομένων αυτών καθώς και ο όγκος των υπολογισμών που απαιτεί η εφαρμογή. Αυτές οι παράμετροι αλλάζουν ανάλογα με τις συνθήκες στις οποίες εκτελείται μια εφαρμογή. Ένα

στατικό μοντέλο, λοιπόν, θα πρέπει να χρησιμοποιεί κατάλληλους αλγορίθμους πρόβλεψης των παραμέτρων αυτών με χρήση πιθανοτήτων ώστε οι αποφάσεις που θα λαμβάνονται να ευνοούν την εκτέλεση της εφαρμογής. Τέτοιοι αλγόριθμοι περιλαμβάνουν πιθανοτικές προβλέψεις [31], προβλέψεις με βάση το ιστορικό εκτέλεσης [14, 19] αλλά και ασαφή έλεγχο (Fuzzy control) [13].

Δυναμική Λήψη Απόφασης: Σε αντίθεση με τα παραπάνω, οι δυναμικές αποφάσεις μπορούν να προσαρμοστούν σε διαφορετικές συνθήκες εκτέλεσης ενός προγράμματος, όπως για παράδειγμα σε περιπτώσεις κυμαινόμενου εύρους ζώνης δικτύου που χρησιμοποιεί η κινητή συσκευή. Οι δυναμικές προσεγγίσεις μπορούν επίσης να χρησιμοποιούν αλγορίθμους πρόβλεψης για τη λήψη των αποφάσεων. Το εύρος ζώνης του δικτύου για τη μεταφόρτωση, λ.χ., μπορεί να μοντελοποιηθεί και να προβλεφθεί με τη χρήση εκτιμητή Bayes [37]. Ωστόσο, οι περισσότερες δυναμικές αποφάσεις επιβαρύνουν με περισσότερους υπολογισμούς το σύστημα επειδή το πρόγραμμα πρέπει διαρκώς να παρακολουθεί και να είναι ενήμερο για τις συνθήκες που επικρατούν στο περιβάλλον εκτέλεσής του. Ακόμη και σε προγράμματα που χρησιμοποιούν την τεχνική των δυναμικών αποφάσεων είναι πιθανό τα κομμάτια του προγράμματος που πιθανώς να μεταφορτωθούν σε υπολογιστικές υποδομές εκτός της συσκευής να αναγνωρίζονται κατά τη διάρκεια της ανάπτυξης του λογισμικού της εφαρμογής.

2.3.2 Τεχνική Διαμοιρασμού Κώδικα

Όπως είναι ξεκάθαρο, οι παραπάνω μηχανισμοί δε ρυθμίζονται ιδιαίτερα εύκολα σε προγραμματιστικό επίπεδο. Πρέπει να γίνει κατάλληλος διαμοιρασμός του κώδικα του προγράμματος (program partitioning) όπου και θα καθορίζονται ποια κομμάτια της κάθε εφαρμογής θα είναι διαθέσιμα για μεταφόρτωση και ποια οφείλουν να εκτελούνται στην τοπική συσκευή. Όπως είναι λογικό, δεν είναι εφικτό να γίνεται δυναμική ανάλυση του προγράμματος κατά την εκτέλεση τους καθώς αυτό θα επιβάρυνε αισθητά το σύστημα και θα επηρέαζε αρνητικά την απόδοση αλλά και το ενεργειακό κόστος.

Διάφοροι αλγόριθμοι [5, 18, 17, 25, 27, 35, 36, 40] έχουν προταθεί για το διαχωρισμό των υπολογισμών του προγράμματος και το διαμοιρασμό αυτών μεταξύ της κινητής συσκευής και του εξυπηρετητή. Μια τυπική προσέγγιση είναι η αναπαράσταση του προγράμματος σαν ένα γράφο, όπου οι κορυφές αποτελούν τις συνιστώσες του προγράμματος που εκτελούν υπολογισμούς, πχ. όπως μια συνάρτηση, και οι ακμές αναδεικνύουν την επικοινωνία που απαιτείται μεταξύ αυτών των συνιστωσών για την σωστή εκτέλεση του κώδικα [26]. Οι ακμές δείχνουν τον όγκο δεδομένων που απαιτούνται σαν είσοδοι και έξοδοι των συνιστωσών αλλά και την αλληλουχία μεταξύ αυτών των προγραμματιστικών συνιστωσών, πχ. δείχνουν τη σειρά εκτέλεσης που πρέπει να τηρείται για να μην υπάρχουν προβλήματα κατά την εκτέλεση. Με αυτόν τον τρόπο, η επίλυση του προβλήματος ανάγεται στο να βρεθούν ποιες από αυτές τις συνιστώσες πρέπει να μεταφορτωθούν σε εξωτερική υπολογιστική υποδομή και ποιες να εκτελεστούν τοπικά στην κινητή συσκευή. Πολλοί παράγοντες παίζουν ρόλο σε αυτή την απόφαση. Δε μπορούμε μεμονωμένα να αποφασίσουμε για κάθε συνιστώσα αν θα μεταφορτωθεί ή όχι καθώς υπάρχει διαδραστικότητα μεταξύ αυτών. Για παράδειγμα, αν μεταξύ δύο υπολογιστικών συνιστωσών ανταλλάσσεται μεγάλος όγκος δεδομένων, καλό θα ήταν να εκτελεστούν και οι δύο στο ίδιο σύστημα (είτε τοπικά είτε σε εξωτερική υπολογιστική υποδομή) ώστε να αποφύγουμε την ανάγκη για μεταφορά μεγάλου όγκου δεδομένων μέσω δικτύου πράγμα που θα εισάγει καθυστέρηση αλλά και κατανάλωση ενέργειας. Ακόμα, κάποια συνιστώσα ενδέχεται να απαιτεί αρκετούς υπολογισμούς και θα πρέπει να μεταφορτωθεί αναγκαστικά σε εξωτερική υπολογιστική υποδομή τόσο για λόγους απόδοσης όσο και για μείωση της κατανάλωσης ενέργειας στην κινητή συσκευή. Γενικά, η βέλτιστη απόφαση και ο καλύτερος διαχωρισμός ενός προγράμματος εξαρτάται από τα πλεονεκτήματα και μειονεκτήματα σε επίπεδο υπολογισμών και επικοινωνίας. Αυτή η απόφαση γίνεται λαμβάνοντας υπόψη όλο το γράφο που προαναφέραμε σαν σύνολο, λύνοντας πρακτικά ένα πρόβλημα διαχωρισμού γράφου (graph partitioning problem) που είναι γνωστό ότι ανήκει στην κατηγορία των NP-complete προβλημάτων, ακόμα και αν όλες οι παράμετροι είναι γνωστοί εκ των προτέρων [15]. Σε άλλες περιπτώσεις, όπου οι πληροφορίες ενός προγράμματος είναι πλήρως άγνωστες [40], η εφαρμογή μπορεί να μη διαχωριστεί σε τέτοια μέρη

και είτε θα μεταφορτωθεί ολόκληρη είτε θα εκτελεστεί στην τοπική κινητή συσκευή.

2.3.3 Βασικά Κριτήρια Λήψης Απόφασης

Πέρα από τα παραπάνω, και όπως αναφέρθηκε και στην αρχή της παρούσας ενότητας, κάθε εφαρμογή, κάθε περίπτωση, κάθε χρήστης έχουν διαφορετικές απαιτήσεις από το σύστημα τους. Έτσι, διαφορετικά είναι και τα κριτήρια με τα οποία θα γίνεται σε κάθε τέτοια διαφορετική περίπτωση η μεταφόρτωση. Οι κύριοι στόχοι της μεταφόρτωσης είναι η μείωση στην κατανάλωση ενέργειας και η ικανοποίηση των μετρικών απόδοσης όπως ορίζονται από τους χρήστες [21].

Αναλυτικότερα, οι πόροι στις κινητές συσκευές δεν επαρκούν για τους υπολογισμούς που καλούνται αυτές να κάνουν. Αυτό έχει ως αποτέλεσμα είτε να απαιτείται πλήρης χρήση αυτών που οδηγεί σε υψηλή κατανάλωση ενέργειας είτε να μην ανταποκρίνονται «γρήγορα» και να παρατηρούνται έτσι χαμηλές επιδόσεις από το σύστημα. Στα παραπάνω, λύση δίνει η τεχνική της μεταφόρτωσης.

Βελτίωση Απόδοσης

Σε πολλές περιπτώσεις η χρόνος απόκρισης των εφαρμογών των κινητών συσκευών μπορεί να μειωθεί με την τεχνική της μεταφόρτωσης. Χωρίς βλάβη της γενικότητας, ένα πρόγραμμα μπορεί να χωριστεί σε δύο τμήματα. Το ένα τμήμα θα πρέπει υποχρεωτικά να εκτελείται τοπικά στην κινητή συσκευή ενώ το άλλο τμήμα έχει τη δυνατότητα να μεταφορτωθεί και να εκτελεστεί σε κάποια εξωτερική υπολογιστική υποδομή. Το πρώτο τμήμα συνήθως περιλαμβάνει τη διεπαφή (interface) που βλέπει ο χρήστης καθώς και κομμάτια κώδικα τα οποία διαχειρίζονται τα περιφερειακά της κινητής συσκευής, όπως για παράδειγμα την κάμερά της. Με απλά μαθηματικά, λοιπόν, αν θεωρήσουμε ως s_m την ταχύτητα εκτέλεσης υπολογισμών της κινητής συσκευής και ως w το μέγεθος των υπολογισμών που εμπεριέχει το δεύτερο μέρος του προγράμματος, τότε ο χρόνος για την εκτέλεση αυτού του μέρους από την ίδια την κινητή συσκευή θα είναι $\frac{w}{s_m}$. Αν τώρα το δεύτερο μέρος μεταφορτωθεί σε κάποιον εξυπηρετητή για να το εκτελέσει, τότε θα χρειαστεί χρόνος $\frac{d_i}{B}$ για την ανταλλαγή δεδομένων, όπου d_i το μέγεθος των δεδομένων που πρέπει να σταλούν από την εφαρμογή στον εξυπηρετητή και B το εύρος ζώνης του δικτύου που θα αναλάβει τη μεταφορά των δεδομένων. Υπάρχει ακόμη το ενδεχόμενο να χρειαστεί να σταλεί το πρόγραμμα το ίδιο στον εξυπηρετητή, ωστόσο δεν το λαμβάνουμε υπ' όψιν καθώς το μέγεθος του συγκριτικά με τα δεδομένα είναι αρκετά μικρό και μπορεί να αμεληθεί ή μπορεί να το βρει ο εξυπηρετητής σε κάποιο άλλο ιστότοπο και να το αποκτήσει με χρήση δικτύου υψηλής ταχύτητας [8]. Άρα, αν μεταφορτώσουμε το πρόγραμμα, ο χρόνος εκτέλεσης αυτού θα είναι $\frac{d_i}{B} + \frac{w}{s_s}$, όπου s_s η ταχύτητα εκτέλεσης υπολογισμών της εξωτερικής υπολογιστικής υποδομής όπου μεταφορτώνεται το πρόγραμμα προς εκτέλεση. Η μεταφόρτωση, λοιπόν, ενός προγράμματος – ή τμήματος αυτού είναι ευεργετική όταν η εκτέλεση του, στην οποία συμπεριλαμβάνεται τόσο ο χρόνος που δαπανάται για υπολογισμούς όσο και ο χρόνος για τη μεταφορά των δεδομένων, είναι ταχύτερη στον εξυπηρετητή απ' ότι στην κινητή συσκευή. Αυτό κυρίως συμβαίνει όταν υπάρχουν αρκετά απαιτητικοί υπολογισμοί στο πρόγραμμα, ο εξυπηρετητής έχει μεγάλη επεξεργαστική ισχύ, δεν απαιτείται μεταφορά μεγάλου πλήθους δεδομένων ή/και το εύρος ζώνης των συνδέσεων είναι μεγάλο.

Μείωση στην Κατανάλωση Ενέργειας

Η περιορισμένη μπαταρία είναι ένα μεγάλο πρόβλημα για τους χρήστες των κινητών συσκευών και ένας από τους πιο σοβαρούς περιορισμούς του ΥΚΝ. Οι «έξυπνες» κινητές συσκευές σήμερα έχουν πολλαπλές χρήσεις. Χρησιμοποιούνται για επικοινωνία, για παρακολούθηση πολυμέσων (βίντεο, ήχου κτλ.), για ψυχαγωγία μέσω παιχνιδιών και πολλά ακόμα. Σαν αποτέλεσμα, αυτά τα συστήματα τείνουν να απαιτούν ολοένα και περισσότερη ενέργεια ώστε να εκτελέσουν όλα τα παραπάνω με αποτέλεσμα αυτό να μειώνει τη διάρκεια ζωής της μπαταρίας τους. Αν και η τεχνολογία που σχετίζεται με τις μπαταρίες διαρκώς εξελίσσεται και βελτιώνεται, αδυνατεί να συμβαδί-

σει με την ταχύτατη αύξηση των ενεργειακών απαιτήσεων αυτών των συστημάτων. Η τεχνική της μεταφόρτωσης δύναται να «επεκτείνει» τη διάρκεια ζωής της μπαταρίας αν οι υπολογισμοί που απαιτούνται για την εκτέλεση μια εφαρμογής ή διεργασίας μεταφορτωθούν για να εκτελεστούν σε άλλη υπολογιστική υποδομή [22]. Όμως αυτό δε συμβαίνει πάντα. Υπάρχουν κριτήρια με τα οποία μπορούμε να κρίνουμε πότε μια μεταφόρτωση θα οδηγήσει σε μείωση κατανάλωσης ενέργειας.

Ας θεωρήσουμε ως p_m την ενέργεια του κινητού μας συστήματος (πχ. ενός smartphone). Με βάση και την παραπάνω ανάλυση που κάναμε, η ενέργεια που απαιτείται ώστε να εκτελεστεί μια εργασία στην ίδια τη συσκευή θα είναι

$$E_{device} = \frac{p_m \cdot w}{s_m} \quad (2.1)$$

Ας ονομάσουμε τώρα p_c την κατανάλωση ενέργειας που απαιτείται για την αποστολή δεδομένων μέσω του δικτύου. Μετά την αποστολή των δεδομένων, το σύστημα πρέπει να παρακολουθεί τη διεπαφή του με το δίκτυο για όσο αναμένει το αποτέλεσμα των μεταφορτωμένων υπολογισμών. Σε αυτό το χρονικό διάστημα ας πούμε ότι η κατανάλωση ενέργειας είναι p_i . Αν συνδυάσουμε όλα αυτά με όσα αναφέραμε παραπάνω, η κατανάλωση ενέργειας όταν μια εφαρμογή μεταφορτώνεται είναι

$$E_{offloaded} = \frac{p_c \cdot d_i}{B} + \frac{p_i \cdot w}{s_s} \quad (2.2)$$

Άρα, συγκρίνοντας αυτές τις δύο σχέσεις, έχουμε εξοικονόμηση ενέργειας όταν $E_{device} > E_{offloaded}$ όπως αυτές προκύπτουν από τις (2.1),(2.2). Με πιο απλά λόγια, αν δούμε τα παραπάνω, σημαντική μείωση στην κατανάλωση ενέργειας κυρίως έχουμε όταν η εφαρμογή μας απαιτεί αρκετούς απαιτητικούς υπολογισμούς ή το ενεργειακό κόστος της επικοινωνίας μέσω του δικτύου είναι χαμηλό. Επομένως, βλέπουμε πως γνωρίζοντας κάποιες παραμέτρους του συστήματός μας και του περιβάλλοντός του, μπορούμε να λαμβάνουμε κατάλληλες αποφάσεις για τη μεταφόρτωση ή μη μιας εφαρμογής σε κάποια εξωτερική υπολογιστική υποδομή προς εκτέλεση ανάλογα με το έχουμε ως προτεραιότητα σε κάθε περίπτωση.

Κεφάλαιο 3

Θεωρητικό Υπόβαθρο

3.1 Γενική Περιγραφή του Προβλήματος

Στην παρούσα εργασία θα μελετήσουμε το πρόβλημα λήψης απόφασης για μεταφόρτωση γνωρίζοντας και λαμβάνοντας υπ' όψη τη θέση των κινητών συσκευών των χρηστών (location aware task offloading problem) [38]. Το πρόβλημα αυτό θα επιλυθεί θεωρώντας ένα περιβάλλον ΥΚΝ με δύο επίπεδα υπολογιστικών υποδομών – το μακρινό υπολογιστικό νέφος και τις τοπικές ΜΥΝ. Το μακρινό νέφος είναι προσβάσιμο μέσω δικτύων 3G/4G από τους παρόχους των κυψελωτων δικτύων επικοινωνιών ενώ οι τοπικές ΜΥΝ μέσω των τοπικών δικτύων WiFi. Επίσης, αξίζει να σημειωθεί ότι θα θεωρήσουμε πως κάθε σημείο πρόσβασης σε τοπικό δίκτυο έχει περιορισμένο αριθμό χρηστών που μπορούν να είναι παράλληλα συνδεδεμένοι σε αυτό, πράγμα ιδιαίτερα σημαντικό που πολλές φορές έχει αμεληθεί σε άλλες ερευνητικές εργασίες [29].

Με την προαναφερθείσα αρχιτεκτονική, θα επικεντρωθούμε στο να διατυπώσουμε έναν αποδοτικό δυναμικό αλγόριθμο για τη λήψη απόφασης μεταφόρτωσης των διεργασιών των χρηστών των κινητών συσκευών, των οποίων η θέση θα είναι γνωστή, στο μακρινό υπολογιστικό νέφος ή στις τοπικές ΜΥΝ έτσι ώστε η κατανάλωση ενέργειας των συσκευών να είναι ισομοιρασμένη. Έτσι, θα επιμηκυνθεί η διάρκεια ζωής των κινητών συσκευών. Σκοπός μας είναι να εξασφαλίσουμε ότι κάθε κινητή συσκευή καταναλώνει το ίδιο, ή περίπου το ίδιο, ποσοστό ενέργειας σε σχέση με την ενεργειακή χωρητικότητά της, ενώ βέβαια πληρούνται οι απαιτήσεις του χρήστη σε επίπεδο υπηρεσιών.

3.2 Μοντέλο Συστήματος

Το μοντέλο συστήματος που θα μελετηθεί είναι μια αρχιτεκτονική ΥΚΝ δύο επιπέδων που αποτελείται από τοπικές ΜΥΝ με περιορισμένους υπολογιστικούς πόρους και μακρινά, πλούσια σε υπολογιστικούς πόρους υπολογιστικά νέφη (clouds). Στο σύστημα υπάρχουν χρήστες που έχουν πρόσβαση και στα δύο επίπεδα αυτής της αρχιτεκτονικής. Οι τοπικές ΜΥΝ είναι προσβάσιμες μέσω σημείου πρόσβασης δικτύου WiFi ενώ οι υπολογιστικές υποδομές νέφους που βρίσκονται μακριά μέσω των δικτύων 3G/4G των παρόχων κυψελωτών δικτύων επικοινωνιών. Επομένως, αυτοί οι χρήστες έχουν τη δυνατότητα να επιλέξουν για μια διεργασία τους αν αυτή θα εκτελεστεί στην συσκευή τους, σε κάποια τοπική ΜΥΝ ή στο μακρινό κέντρο δεδομένων.

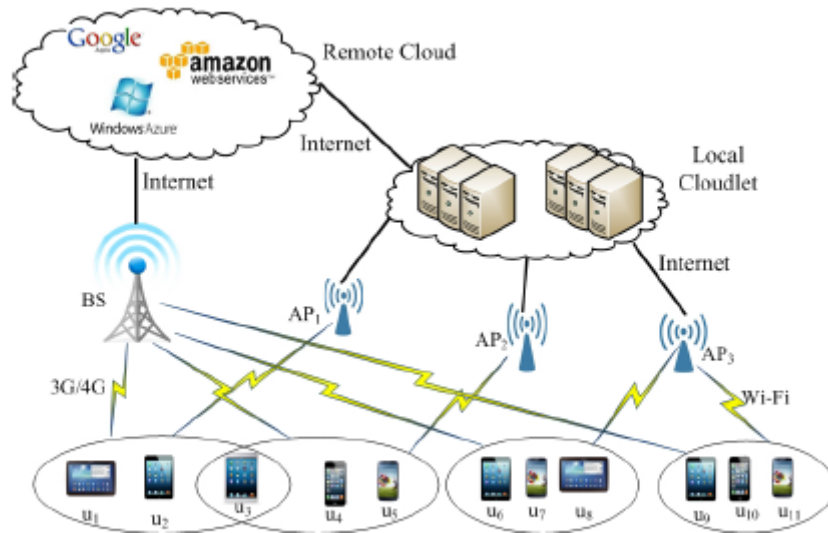
Συνήθως, η επικοινωνία μέσω δικτύου WiFi καταναλώνει λιγότερη ενέργεια απ' ότι μέσω δικτύου 3G/4G. Όμως, αν υπάρχει διαθέσιμο μικρό εύρος ζώνης σε ένα σημείο πρόσβασης WiFi δικτύου, τότε μπορεί να συμβαίνει και το αντίθετο. Πιο συγκεκριμένα, ας θεωρήσουμε ως C_r και C_l το μακρινό υπολογιστικό νέφος και τις ΜΥΝ με ταχύτητες εξυπηρέτησης (service rates) μ_r και μ_l αντίστοιχα. Είναι ξεκάθαρο ότι $\mu_r \geq \mu_l$. Η διαχείριση των πόρων στο υπολογιστικό νέφος γίνεται μέσω πολύπλοκων και δυναμικών συστημάτων [9, 23], αλλά στην περίπτωση μας θα θεωρούμε ότι η ταχύτητα εξυπηρέτησης στο υπολογιστικό νέφος δίνεται από το στατικό μέγεθος μ_r . Το απομακρυσμένο υπολογιστικό νέφος C_r , που είναι ιδιοκτησία των παρόχων υπηρεσιών νέφους, έχει πολλούς υπολογιστικούς πόρους για τους χρήστες κινητών συσκευών, για τους οποίους όμως εκείνοι είναι υποχρεωμένοι να καταβάλλουν χρηματικό αντίτιμο. Εν αντιθέσει, οι τοπικές ΜΥΝ

C_i , που διαχειρίζονται συνήθως από κάποιο τοπικό οργανισμό, έχουν περιορισμένους πόρους και είναι δωρεάν ή με μικρό χρηματικό κόστος.

Θεωρούμε το σύνολο των H Σημείων Πρόσβασης (ΣΠ-Access Points-AP) στην τοπική ΜΥΝ $AP = \{AP_1, AP_2, \dots, AP_H\}$. Ένας χρήστης έχει πρόσβαση σε ένα εξ αυτών των ΣΠ μόνο αν βρίσκεται εντός του χωρικού εύρους μετάδοσης αυτού του ΣΠ. Ονομάζουμε B_h τη χωρητικότητα σε εύρος ζώνης καθενός από τα H ΣΠ, με $1 \leq h \leq H$. Αφού, λοιπόν, το εύρος ζώνης ενός ΣΠ έχει μια δεδομένη τιμή, έτσι και ο αριθμός των χρηστών που μπορούν να συνδεθούν και να επικοινωνήσουν μέσω αυτού με την τοπική ΜΥΝ παράλληλα έχει κάποιο άνω φράγμα. Όταν ο αριθμός των χρηστών που χρησιμοποιούν ένα ΣΠ ισούται με τη χωρητικότητά του, τότε το ΣΠ δε θα αποδέχεται περαιτέρω αιτήσεις από άλλους χρήστες έως ότου κάποιος χρήστης από τους ήδη συνδεδεμένους, για οποιοδήποτε λόγο, αποσυνδεθεί. Θεωρούμε, ακόμα, ότι το εύρος ζώνης που δεσμεύεται από κάθε χρήστη σε ένα AP_h είναι ίσο με $\frac{AP_h}{n_a}$, όπου n_a είναι ο μέγιστος αριθμός χρηστών που μπορεί να δεχτεί παράλληλα το AP_h . Μοιράζεται το εύρος ζώνης ισόποσα σε όλους τους χρήστες. Σε περίπτωση που ένας χρήστης βρίσκεται σε ένα σημείο όπου έχουν εμβέλεια παραπάνω από ένα ΣΠ, τότε το σύστημα θα αποφασίσει σε ποιο ΣΠ ο χρήστης αυτός θα συνδεθεί, κυρίως με βάση την ισχύ σήματος που έχει κάθε ΣΠ στο συγκεκριμένο σημείο.

Όπως προαναφέρθηκε, κάθε χρήστης έχει και τη δυνατότητα πρόσβασης και στο απομακρυσμένο υπολογιστικό νέφος μέσω σύνδεσης 3G/4G δικτύου από κάποιο σταθμό βάσης. Σε αυτήν την περίπτωση, θεωρούμε ότι οι σταθμοί βάσης παρέχουν αρκετά μεγάλο εύρος ζώνης στους χρήστες, ικανό να καλύψει τις απαιτήσεις πρόσβασης των χρηστών. Ας θεωρήσουμε τώρα ως R_r και R_l την ταχύτητα μετάδοσης δεδομένων μέσω 3G/4G δικτύου και WiFi δικτύου, αντίστοιχα. Για λόγους απλότητας, θα θεωρήσουμε ότι για όλες τις συσκευές, οι ρυθμοί αποστολής και παραλαβής δεδομένων είναι ίσοι. Χωρίς αυτή τη θεώρηση, ο παρακάτω αλγόριθμος μπορεί εύκολα να τροποποιηθεί και να καλύψει και αυτήν την περίπτωση.

Όλα τα παραπάνω συνοψίζονται στο ακόλουθο σχήμα:



Σχήμα 3.1: Παράδειγμα Αρχιτεκτονικής Πολυεπίπεδου Δικτύου [38]

3.3 Ανάλυση Μοντέλου Κατανάλωσης Ενέργειας μιας Μεταφόρτωσης

Θεωρούμε $U = \{u_i | 1 \leq i \leq n\}$ το σύνολο των χρηστών κινητών συσκευών. Ένας χρήστης έχει πρόσβαση στις τοπικές ΜΥΝ μέσω διαφορετικών ΣΠ και μπορεί να έχει διαφορετικές εργασίες

προς μεταφόρτωση σε κάθε τοποθεσία που βρίσκεται. Κάθε φορά που ο χρήστης έχει μια διεργασία που πρέπει να εκτελεστεί οι εκδοχές είναι τρεις. Είτε αυτή θα εκτελεστεί τοπικά στη συσκευή του, είτε θα μεταφορτωθεί σε κάποια τοπική ΜΥΝ, είτε θα μεταφορτωθεί στο απομακρυσμένο Υπολογιστικό Νεφος- ΥΝ (Cloud). Η απόφαση αυτή λαμβάνεται από το σύστημα. Για παράδειγμα, αν μια εργασία έχει αυστηρές απαιτήσεις, ως προς την ποιότητα της υπηρεσίας, τότε η μεταφόρτωση αυτής της εργασίας σε τοπική ΜΥΝ ή σε απομακρυσμένο ΥΝ μπορεί να παραβιάζει τους περιορισμούς που θέτει η εφαρμογή, πχ. ως προς το χρόνο απόκρισης, εξαιτίας πιθανού μεγάλου φόρτου που ενδεχομένως θα έχει το ΣΠ που θα επιλεγεί εκείνη τη στιγμή. Αντίθετα, ο χρήστης μπορεί να εκτελέσει την εν λόγω διεργασία στην προσωπική του κινητή συσκευή. Διαφορετικά αν δεν έχουμε τόσο αυστηρούς περιορισμούς, ο χρήστης μπορεί να μεταφορτώσει τη συγκεκριμένη εργασία σε κάποια υπολογιστική υποδομή έτσι ώστε να εξοικονομήσει ενέργεια για την προσωπική κινητή συσκευή του πληρώνοντας ταυτόχρονα τις απαιτήσεις της εφαρμογής.

Για την επίλυση του προβλήματος θα θεωρήσουμε ένα σχετικά μεγάλο χρονικό διάστημα παρατήρησης του συστήματος. Ο χρόνος διαχωρίζεται σε ίσες χρονοσχισμές (timeslots). Η δρομολόγηση για τις διεργασίες γίνεται στην εκκίνηση κάθε χρονοσχισμής. Ονομάζουμε $s_{i,t}$ μια εργασία του χρήστη u_i που δημιουργείται μεταξύ των χρονοσχισμών $t - 1$ και t με όγκο εργασίας i, t , όπου ως όγκο εργασίας μπορούμε να θεωρήσουμε τον αριθμό των εντολών της εργασίας αυτής. Κάθε εργασία $s_{i,t}$ μπορεί να αντιπροσωπευθεί από ένα σύνολο χαρακτηριστικών, το οποίο περιλαμβάνει τον όγκο εργασίας της $A_{i,t}$, τη χρονοσχισμή άφιξης t , τη θέση $l_{i,t}$ του χρήστη και τις απαιτήσεις σε επίπεδο υπηρεσίας της εφαρμογής. Μια εργασία παρουσιάζεται ως εξής: $s_{i,t} = \langle A_{i,t}, t, l_{i,t}, D(s_{i,t}) \rangle$. Ακόμα, υποθέτουμε ότι σε κάθε χρονοσχισμή, ένας χρήστης μπορεί να έχει το πολύ μια διεργασία προς εκτέλεση. Το σύστημα δεν έχει καμία επίγνωση για μελλοντικές αφίξεις εργασιών ούτε για το μελλοντικό ρυθμό αφίξεως αυτών. Οι απαιτήσεις σε επίπεδο υπηρεσίας μιας εργασίας $s_{i,t}$ πρακτικά αναφέρεται στην επιτρεπτή καθυστέρηση, δηλαδή στη διάρκεια της μεταφόρτωσης της εργασίας $s_{i,t}$ στην υπολογιστική υποδομή έως τη στιγμή που θα ληφθούν τα αποτελέσματα αυτής. Η επιτρεπτή καθυστέρηση δεν είναι τίποτα άλλο παρά ένα προκαθορισμένο άνω φράγμα $D(s_{i,t})$ στο χρόνο που θα λάβει ο χρήστης το αποτέλεσμα της εργασίας. Θα ονομάσουμε $D_r(s_{i,t})$ και $D_l(s_{i,t})$ τις μέσες καθυστερήσεις που υφίστανται όταν μεταφορτώνουμε την εργασία $s_{i,t}$ στο απομακρυσμένο ΥΝ και σε κάποια τοπική ΜΥΝ αντίστοιχα. Ακόμα υποθέτουμε ότι το μέγεθος δεδομένων που σχετίζεται με την εργασία $s_{i,t}$ αντιπροσωπεύεται από τη συνάρτηση $f(s_{i,t})$. Για απλότητα, υποθέτουμε ότι ο χρόνος που κάνουμε για να λάβουμε το αποτέλεσμα της $s_{i,t}$ ισούται με το χρόνο που απαιτείται για την αποστολή της $s_{i,t}$ στην εξωτερική υπολογιστική υποδομή. Έχοντας ως δεδομένα τον όγκο εργασίας $A_{i,t}$, το μέγεθος των δεδομένων $f(s_{i,t})$ της εργασίας, τους ρυθμούς μετάδοσης δεδομένων των δικτύων R_r και R_l καθώς και την ταχύτητα εξυπηρέτησης (service rate) μ_r και μ_l των υπολογιστικών υποδομών, για τη σύνδεση με το απομακρυσμένο ΥΝ και κάποια τοπική ΜΥΝ αντίστοιχα, μπορούμε να υπολογίσουμε τις δύο καθυστερήσεις όπως φαίνεται παρακάτω:

$$D_r(s_{i,t}) = 2 \cdot \frac{f(s_{i,t})}{R_r} + \frac{A(s_{i,t})}{\mu_r} \quad (3.1)$$

$$D_l(s_{i,t}) = 2 \cdot \frac{f(s_{i,t})}{R_l} + \frac{A(s_{i,t})}{\mu_l} \quad (3.2)$$

Επομένως αυτό που πρέπει να τηρείται ώστε να πληρούνται οι απαιτήσεις σε επίπεδο υπηρεσίας που έχει η αλληλεπίδραση εφαρμογής-χρήστη είναι:

- $D_r(s_{i,t}) \leq D(s_{i,t})$ αν η εργασία εκτελείται στο απομακρυσμένο ΥΝ
- $D_l(s_{i,t}) \leq D(s_{i,t})$ αν η εργασία εκτελείται σε κάποια τοπική ΜΥΝ

Με βάση τα παραπάνω μεγέθη μπορούμε να υπολογίσουμε την ενεργειακή κατανάλωση που θα προκαλέσει κάθε μεταφόρτωση μιας εργασίας. Ονομάζουμε P_{tr} και P_{idle} την κατανάλωση ενέργειας ανά μονάδα χρόνου μιας κινητής συσκευής όταν εκείνη είναι σε κατάσταση αποστολής-λήψης δεδομένων και αναμονής αντίστοιχα. Επίσης, θεωρούμε ότι τα P_d και μ_d αντιπροσωπεύουν

την κατανάλωση ενέργειας και την επεξεργαστική ταχύτητα (service rate) μιας κινητής συσκευής όταν αναλαμβάνει να εκτελέσει η ίδια την εργασία. Αν η εργασία $s_{i,t}$ εκτελεστεί στο απομακρυσμένο ΥΝ τότε η κατανάλωση ενέργεια $\Delta E_r(t)$ υπολογίζεται απο την εξίσωση:

$$\Delta E_r(t) = 2 \cdot \frac{f(s_{i,t})}{R_r} \cdot P_{tr} + \frac{A(s_{i,t})}{\mu_r} \cdot P_{idle} \quad (3.3)$$

όπου για απλότητα έχουμε υποθέσει ότι η ενέργεια που καταναλώνεται για την αποστολή των δεδομένων είναι ίση με την ενέργεια που απαιτείται για τη λήψη των αποτελεσμάτων μετά τους υπολογισμούς.

Αν η εργασία $s_{i,t}$ εκτελεστεί σε κάποια τοπική ΜΥΝ τότε η κατανάλωση ενέργειας $\Delta E_l(t)$ υπολογίζεται ως εξής:

$$\Delta E_l(t) = 2 \cdot \frac{f(s_{i,t})}{R_l} \cdot P_{tr} + \frac{A(s_{i,t})}{\mu_l} \cdot P_{idle} \quad (3.4)$$

Αν η εργασία αποφασιστεί να εκτελεστεί στην ίδια τη συσκευή, το ενεργειακό κόστος $\Delta E_d(t)$ δίνεται απο την ακόλουθη σχέση:

$$\Delta E_d(t) = \frac{A(s_{i,t})}{\mu_d} \cdot P_d \quad (3.5)$$

όπου μ_d είναι η επεξεργαστική ταχύτητα της κινητής συσκευής και P_d η κατανάλωση ενέργειας για τους υπολογισμούς στην ίδια την κινητή συσκευή.

3.4 Αναλυτική Παρουσίαση του Προβλήματος προς Επίλυση

Έχουμε την αρχιτεκτονική δύο επιπέδων που περιγράψαμε προηγουμένως και ένα σύνολο από χρήστες κινητών συσκευών $U = \{u_i | 1 \leq i \leq n\}$. Θεωρούμε ότι ο χρόνος παρατήρησης του συστήματός μας θα αποτελείται απο T χρονοσχιμές. Το πρόβλημα της δυναμικής, σε πραγματικό χρόνο (real time), μεταφόρτωσης μιας εφαρμογής λαμβάνοντας υπόψη τις θέσεις των χρηστών (location-aware mobile application offloading problem) ανάγεται στο να βρούμε μια δρομολόγηση των εργασιών των χρηστών σε κάθε χρονοσχιμή έτσι ώστε οι κινητές συσκευές, οι εργασίες των οποίων αποφασίζεται να μεταφορτωθούν, να μπορούν να μοιράζονται ισάξια τους πόρους των εξωτερικών υπολογιστικών υποδομών και να καταναλώνουν το ίδιο ποσοστό ενέργειας. Πιο συγκεκριμένα, στόχος είναι να μεγιστοποιηθεί το ελάχιστο ποσοστό υπολοιπόμενης ενέργειας σε σχέση με τη χωρητικότητα ενέργειας κάθε συσκευής με χρήση της μεταφόρτωσης, πάντα πληρώνοντας τις απαιτήσεις της εφαρμογής και του χρήστη, όπως για παράδειγμα κάποιο περιορισμό στο χρόνο απόκρισης.

Δεδομένης μιας χρονοσχιμής t εντός της περιόδου παρατήρησης T , ονομάζουμε (t) το σύνολο των εργασιών που φτάνουν μεταξύ των χρονοσχιμών $t - 1$ και t . Επίσης, υποθέτουμε ότι κάθε χρήστης έχει το πολύ μία εργασία προς εκτέλεση σε κάθε χρονοσχιμή. Ορίζουμε ως $r_i(t)$ το ποσοστό υπολοιπόμενης ενέργειας του χρήστη u_i στη χρονοσχιμή t . Άρα έχουμε

$$r_i(t) = \frac{E_{residual}(u_i, t)}{E(u_i)} \quad (3.6)$$

όπου $E_{residual}(u_i, t)$ είναι η υπολοιπόμενη ενέργεια της κινητής συσκευής του χρήστη u_i μετά τη χρονοσχιμή t και $E(u_i)$ είναι η χωρητικότητα της μπαταρίας της κινητής συσκευής του χρήστη u_i . Το πρόβλημα λοιπόν είναι το να μεγιστοποιήσουμε το ελάχιστο ποσοστό υπολοιπόμενης ενέργειας μεταξύ των χρηστών, του οποίου η μαθηματική μορφή είναι $r_{min} = \{r_i(t) | 1 \leq i \leq n\}$.

3.5 Αλγόριθμος Επίλυσης του Προβλήματος

Στο κομμάτι αυτό θα αναλύσουμε το μοντέλο κοστολόγησης της ενέργειας [38] για την επεξεργασία μια εργασίας στις διάφορες υπολογιστικές υποδομές και με βάση αυτό θα παρατεθεί ένας αλγόριθμος που επιλύει το πρόβλημα που αναφέρεται προηγουμένως.

3.5.1 Ανάλυση Μοντέλου Κοστολόγησης μιας Μεταφόρτωσης

Το ενεργειακό κόστος $\zeta_1(s_{i,t})$ της εκτέλεσης της εργασίας $s_{i,t}$ στην κινητή συσκευή u_i καθορίζεται όχι μόνο από τη χωρητικότητα της μπαταρίας αλλά και από την υπολοιπόμενη ενέργεια που διαθέτει και το ποσοστό υπολοιπόμενης ενέργειας $r_i(t)$ της. Αυτό το ενεργειακό κόστος συνήθως αυξάνεται σταδιακά καθώς μειώνεται το ποσοστό της υπολοιπόμενης ενέργειας της κινητής συσκευής. Η λογική πίσω από αυτό είναι ότι όσο λιγότερη υπολοιπόμενη ενέργεια έχει η συσκευή, τόσο πιθανότερο είναι η συσκευή να εξαντλήσει την ενέργεια που της απομένει αλλά και να παραβιάσει τις απαιτήσεις σε επίπεδο υπηρεσιών που υπάρχουν. Αυτό πρακτικά σημαίνει ότι όσο χαμηλότερο είναι το ποσοστό υπολοιπόμενης ενέργειας τόσο μεγαλύτερο το ενεργειακό κόστος της εργασίας $s_{i,t}$. Για αυτό μοντελοποιούμε το ενεργειακό κόστος για την διεκπεραίωση της εργασίας $s_{i,t}$ σε μια υπολογιστική υποδομή σαν μια εκθετική συνάρτηση του ποσοστού της υπολοιπόμενης ενέργειας όπως φαίνεται παρακάτω:

$$\zeta_1(s_{i,t}) = \begin{cases} \alpha^{1 - \frac{E_{residual}(u_i, t-1) - \Delta E_d(t)}{E(u_i)}} & , \text{if } (E_{residual}(u_i, t-1) - \Delta E_d(t)) \geq 0 \\ \infty & , \text{otherwise} \end{cases} \quad (3.7)$$

Παρόμοια, τα ενεργειακά κόστη $\zeta_2(s_{i,t})$ και $\zeta_3(s_{i,t})$ που αντιστοιχούν στην εκτέλεση της εργασίας $s_{i,t}$ σε κάποια τοπική ΜΥΝ και στο απομακρυσμένο ΥΝ ορίζονται ως εξής:

$$\zeta_2(s_{i,t}) = \begin{cases} \alpha^{1 - \frac{E_{residual}(u_i, t-1) - \Delta E_l(t)}{E(u_i)}} & , \text{if } (E_{residual}(u_i, t-1) - \Delta E_l(t)) \geq 0 \\ \infty & , \text{otherwise} \end{cases} \quad (3.8)$$

και

$$\zeta_3(s_{i,t}) = \begin{cases} \alpha^{1 - \frac{E_{residual}(u_i, t-1) - \Delta E_r(t)}{E(u_i)}} & , \text{if } (E_{residual}(u_i, t-1) - \Delta E_r(t)) \geq 0 \\ \infty & , \text{otherwise} \end{cases} \quad (3.9)$$

όπου $\alpha > 1$ είναι μια σταθερά η οποία αντανακλά το ρυθμό αύξησης της κατανάλωσης ενέργειας για τις εργασίες όσο μειώνεται η υπολοιπόμενη ενέργεια. Όσο μεγαλύτερη είναι η τιμή της σταθεράς α , τόσο μεγαλύτερο είναι και το κόστος εκτέλεσης μιας εργασίας προοδευτικά. Υπενθυμίζουμε ότι $E_{residual}(u_i, t-1)$ είναι η υπολοιπόμενη ενέργεια της κινητής συσκευής του χρήστη u_i μετά τη χρονοσχιστή $t-1$, $E(u_i)$ είναι η χωρητικότητα της μπαταρίας της κινητής συσκευής του χρήστη u_i , $\frac{E_{residual}(u_i, t-1)}{E(u_i)}$ είναι το ποσοστό υπολοιπόμενης ενέργειας του χρήστη u_i τη χρονοσχιστή $t-1$. Από τις εξισώσεις (3.7), (3.8) και (3.9) μπορεί να εξαχθεί ότι η τιμή του ενεργειακού κόστους $\zeta_\kappa(s_{i,t})$ λαμβάνει τιμές στο εύρος $[1, \alpha]$ με $1 \leq \kappa \leq 3$.

Μεταβλητή	Επεξήγηση
μ_r	Επεξεργαστική Ταχύτητα Απομακρυσμένου ΥΝ
μ_l	Επεξεργαστική Ταχύτητα Τοπικής ΜΥΝ
μ_d	Επεξεργαστική Ταχύτητα Συσκευής
AP_h	h-οστό Σημείο Πρόσβασης
B_h	Εύρος ζώνης h-οστού Σημείου Πρόσβασης
n_a	Αριθμός Καναλιών Σημείου Πρόσβασης
R_r	Ταχύτητα Μετάδοσης Δεδομένων μέσω 3G/4G Δικτύου
R_l	Ταχύτητα Μετάδοσης Δεδομένων μέσω Δικτύου WiFi
$s_{i,t}$	Εργασία του χρήστη u_i τη χρονοσχισμή t
$A_{i,t}$	Όγκος της εργασίας του χρήστη u_i τη χρονοσχισμή t
$l_{i,t}$	Θέση του χρήστη u_i τη χρονοσχισμή t
$f(s_{i,t})$	Όγκος Δεδομένων της Εργασίας $s_{i,t}$
$D(s_{i,t})$	Επιτρεπτή Καθυστέρηση της εργασίας $s_{i,t}$
$D_r(s_{i,t})$	Μέση Καθυστέρηση Μεταφόρτωσης στο απομακρυσμένο ΥΝ
$D_l(s_{i,t})$	Μέση Καθυστέρηση Μεταφόρτωσης στην τοπική ΜΥΝ
P_{tr}	Κατανάλωση Ενέργειας κατά την αποστολή και λήψη δεδομένων ανά μονάδα χρόνου
P_{idle}	Κατανάλωση Ενέργειας σε κατάσταση αναμονής ανά μονάδα χρόνου
P_d	Κατανάλωση Ενέργειας Συσκευής σε κατάσταση εκτέλεσης εργασίας ανά μονάδα χρόνου
$\Delta E_r(t)$	Κατανάλωση Ενέργειας για εκτέλεση εργασίας στο απομακρυσμένο ΥΝ
$\Delta E_l(t)$	Κατανάλωση Ενέργειας για εκτέλεση εργασίας στην τοπική ΜΥΝ
$\Delta E_d(t)$	Κατανάλωση Ενέργειας για εκτέλεση εργασίας στη συσκευή
$r_i(t)$	Ποσοστό Υπολειπόμενης Ενέργειας του χρήστη u_i τη χρονοσχισμή t
$E_{residual}(u_i, t)$	Υπολοιπόμενη Ενέργεια της κινητής συσκευής του χρήστη u_i μετά τη χρονοσχισμή t
$E(u_i)$	Χωρητικότητα της μπαταρίας της κινητής συσκευής του χρήστη u_i
$\zeta_1(s_{i,t})$	Κόστος εκτέλεσης της εργασίας $s_{i,t}$ στο απομακρυσμένο ΥΝ
$\zeta_2(s_{i,t})$	Κόστος εκτέλεσης της εργασίας $s_{i,t}$ στην τοπική ΜΥΝ
$\zeta_3(s_{i,t})$	Κόστος εκτέλεσης της εργασίας $s_{i,t}$ στη συσκευή

Πίνακας 3.1: Συγκεντρωτικός Πίνακας Μεταβλητών

3.5.2 Δημιουργία Γράφου

Θα θεωρήσουμε ότι η επεξεργασία μιας εργασίας ξεκινάει στην αρχή κάθε χρονοσχισμής. Για τη χρονοσχισμή t , ο αλγόριθμος αποφασίζει εάν οι εργασίες που δημιουργούνται στο σύστημά μας μεταξύ των χρονοσχισμών $t - 1$ και t θα πρέπει να εκτελεστούν στην ίδια την κινητή συσκευή, να μεταφορτωθούν σε κάποια τοπική ΜΥΝ ή στο απομακρυσμένο ΥΝ. Αυτή η απόφαση θα ληφθεί με στόχο την μεγιστοποίηση του ελάχιστου ποσοστού υπολοιπόμενης ενέργειας των συσκευών μετά από T χρονοσχισμές. Το πρόβλημα αυτό θα το προσεγγίσουμε σαν ένα πρόβλημα σταθμισμένου μέγιστου ταιριάσματος (weighted maximum matching) σε ένα διμερές γράφο $G(t) = (V_1(t), V_2(t), E(t))$.

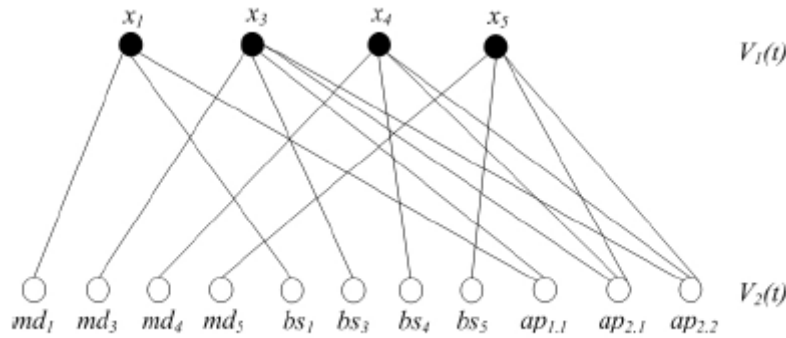
Υποθέτουμε ότι έχουμε n' εργασίες, δηλαδή ότι $T(t) = \{s_{i,t} | 1 \leq i \leq n'\}$. Επίσης, υποθέτουμε ότι $n' \leq n$. Κάθε εργασία του συνόλου $T(t)$ αντιστοιχεί σε έναν κόμβο του $V_1(t)$. Έχουμε δηλαδή $V_1(t) = \{x_i | s_{i,t} \in T(t)\}$ και $|V_1(t)| = |T(t)| = n'$. Κάθε κόμβος του συνόλου $V_2(t)$ αντιστοιχεί σε μια πιθανή υπολογιστική υποδομή που μπορεί να αναλάβει να εκτελέσει μια εργασία του συνόλου $T(t)$. Αυτές οι πιθανές υπολογιστικές υποδομές είναι κάποια τοπική ΜΥΝ (C_l), το απομακρυσμένο ΥΝ (C_r) ή η ίδια η κινητή συσκευή για κάθε εργασία του συνόλου $T(t)$.

Πιο αναλυτικά, το απομακρυσμένο ΥΝ είναι προσβάσιμο από όλες τις κινητές συσκευές μέσω κυψελωτού δικτύου με χρήση ενός σταθμού βάσης (ΣΒ). Έτσι, το σύνολο κόμβων $V_2(t)$ περιλαμβάνει n' κόμβους που αντιστοιχούν σε σύνδεση με το σταθμό βάσης για κάθε εργασία του συνόλου

$T(t)$ κάθε κινητής συσκευής. Άρα έχουμε $VR(t) = \{bs_i | s_{i,t} \in T(t)\}$. Οι τοπικές ΜΥΝ είναι προσβάσιμες από τις συσκευές μέσω των τοπικών σημείων πρόσβασης $AP_h \in AP$ για όλα τα h , όπου $1 \leq h \leq H$. Ένα σημείο πρόσβασης AP_h διαθέτει $rb(h,t)$ κανάλια τα οποία είναι κατειλημμένα τη χρονοσχισμή t . Τότε, το συγκεκριμένο σημείο πρόσβασης έχει $n_a - rb(h,t)$ κανάλια διαθέσιμα για τους χρήστες, πράγμα που θα πει πως το πολύ $n_a - rb(h,t)$ χρήστες μπορούν να συνδεθούν σε κάποια τοπική ΜΥΝ μέσω αυτού του σημείου πρόσβασης εκείνη τη χρονοσχισμή. Η τιμή n_a αντιπροσωπεύει το μέγιστο αριθμό χρηστών που μπορούν να είναι συνδεδεμένοι σε ένα σημείο πρόσβασης ταυτόχρονα. Άρα έχουμε $VL(t) = \{ap_{h,j} | s_{i,t} \in T(t), AP_h \in AP\}$ τα εικονικά σημεία πρόσβασης WiFi. Αν το σημείο πρόσβασης AP_h είναι εντός του εύρους εκπομπής του χρήστη u_i τότε κάθε κόμβος $ap_{h,j} \in VL(t)$, που ανήκει πρακτικά στο AP_h , μπορεί να είναι σημείο πρόσβασης για το χρήστη u_i για όλα τα $1 \leq i \leq n$ και $1 \leq j \leq n_a - rb(h,t)$. Ακόμη, υπάρχουν n' κόμβοι στο σύνολο $V_2(t)$ που αντιστοιχούν ο καθένας σε μια συσκευή ενός χρήστη που έχει μια εργασία προς εκτέλεση στο σύνολο $T(t)$. Επομένως θα έχουμε το σύνολο $VM(t) = \{md_i | s_{i,t} \in T(t)\}$. Με όλα τα παραπάνω, το σύνολο $V_2(t)$ πρακτικά είναι η ένωση των συνόλων $VR(t)$, $VL(t)$ και $VM(t)$. Είναι προφανές ότι $|V_2(t)| = |VR(t)| + |VL(t)| + |VM(t)| \leq n' + n_a \cdot H + n' \leq 2 \cdot n + n_a \cdot H$.

Το σύνολο των ακμών στο γράφο $E(t)$ αποτελείται από ακμές μεταξύ των κόμβων του συνόλου $V_1(t)$, που κάθε ένας αντιστοιχεί σε μία εργασία $s_{i,t}$, με κάποιον ή κάποιους από τους κόμβους του συνόλου $V_2(t)$, που αντιστοιχούν σε κάποια υπολογιστική μονάδα. Η κάθε τέτοια ακμή υπάρχει μόνο εάν η εργασία $s_{i,t}$, που υποδεικνύει ο κόμβος του $V_1(t)$, μπορεί να εκτελεστεί στην υπολογιστική μονάδα που αντιστοιχεί στον κόμβο του συνόλου $V_2(t)$ χωρίς να παραβιάζει τις απαιτήσεις σε επίπεδο υπηρεσίας(SLA). Με άλλα λόγια, ένας κόμβος του συνόλου $V_1(t)$ ενώνεται με έναν κόμβο του συνόλου $V_2(t)$ μόνο εάν η υπολογιζόμενη από τις εξισώσεις (3.1) και (3.2) καθυστέρηση είναι μικρότερη ή ίση από το προκαθορισμένο όριο $D(s_i, t)$. Ως βάρος κάθε ακμής θεωρούμε το ενεργειακό κόστος που προκύπτει για κάθε αντιστοίχιση της εργασίας σε μια υπολογιστική μονάδα όπως αυτό υπολογίζεται από την αντίστοιχη εξίσωση (3.7),(3.8) ή (3.9).

Ένα παράδειγμα γραφήματος σαν αυτό που περιγράψαμε φαίνεται στο σχήμα 3.2 που παρατίθεται στη συνέχεια.



Σχήμα 3.2: Παράδειγμα Γράφου Δικτύου [38]

Στο συγκεκριμένο παράδειγμα βλέπουμε ότι έχουμε δύο σημεία πρόσβασης. Το AP_1 έχει ένα κανάλι διαθέσιμο ενώ το AP_2 δύο. Υπάρχουν πέντε χρήστες στο σύστημα. Εργασίες προς εκτέλεση σε αυτή τη χρονοσχισμή φτάνουν μόνο από τους χρήστες 1, 3, 4 και 5. Θεωρούμε ότι το AP_1 και το AP_2 είναι προσβάσιμα από τους χρήστες 3 και 5. Ο χρήστης 1 έχει πρόσβαση μόνο στο AP_1 και ο χρήστης 4 μόνο στο AP_2 . Επίσης, θεωρούμε ότι ο χρήστης 5 δε μπορεί να μεταφορτώσει την εργασία του μέσω του AP_1 γιατί παραβιάζει το κατώφλι που ορίζεται για την εργασία.

3.5.3 Εφαρμογή του Αλγορίθμου

Έχοντας δημιουργήσει το γράφο $G(t)$, θα επεξηγηθεί με περισσότερες λεπτομέρειες ο αλγόριθμος που επιλύει το πρόβλημα. Αυτό που πρέπει να βρεθεί στο γράφο $G(t)$ είναι το μέγιστο ταίριασμα (maximum matching) με σκοπό την ελαχιστοποίηση του σταθμισμένου αθροίσματος των επιλεγμένων ακμών. Προφανώς, για κάθε κόμβο του συνόλου $V_1(t)$ θα επιλεγεί μία ακμή. Για κάθε «ταιριασμένη» (matched) ακμή (x_i, y_j) με $x_i \in V_1(t)$ και $y_j \in V_2(t)$

- αν $y_j \in VR(t)$ τότε η συγκεκριμένη εργασία $s_{i,t}$ επιλέχθηκε να μεταφορτωθεί στο απομακρυσμένο YN
- αν $y_j \in VL(t)$ τότε η συγκεκριμένη εργασία $s_{i,t}$ επιλέχθηκε να μεταφορτωθεί σε κάποια τοπική MYN μέσω του σημείου πρόσβασης που αντιπροσωπεύεται από το y_j
- αν $y_j \in VM(t)$ τότε η συγκεκριμένη εργασία $s_{i,t}$ επιλέχθηκε να εκτελεστεί στην ίδια τη συσκευή του χρήστη

Η λεπτομερότερη περιγραφή του αλγορίθμου αναφέρεται σαν Αλγόριθμος 1, και για συντομία τον ονομάζουμε *Alg – MBM* [38].

Αλγόριθμος 1 Αλγόριθμος για το πρόβλημα δυναμικής μεταφόρτωσης εφαρμογών κινητών συσκευών λαμβάνοντας υπ' όψη την τοποθεσία τους

Είσοδος: Ένα σύνολο χρηστών κινητών συσκευών $\{u_i | 1 \leq i \leq n\}$, ένας σταθμός βάσης BS , ένα σύνολο σημείων πρόσβασης στο δίκτυο WiFi AP , ένα Απομακρυσμένο YN, τοπικά MYN

Έξοδος: Μεγιστοποίηση του ελάχιστου ποσοστού υπολειπόμενης ενέργειας μεταξύ των κινητών συσκευών σε κάθε χρονοσχισημή t

- 1: Για κάθε χρονοσχισημή t επανάλαβε
 - 2: Θεωρούμε ότι $T(t) = \{s_{i,t} | 1 \leq i \leq n'\}$ είναι το σύνολο των εργασιών που καταφτάνουν για να εκτελεστούν μεταξύ των χρονοσχισημών $t - 1$ και t και ότι κάθε μία από αυτές τις εργασίες περιγράφεται από τον όγκο εργασίας, τη θέση από την οποία αποστέλεται και το χρόνο που φτάνει στο σύστημά.
 - 3: Δημιουργούμε το σταθμισμένο διμερές γράφημα $G(t) = (V_1(t), V_2(t), E(t))$ για τις εργασίες του συνόλου $T(t)$
 - 4: Βρίσκουμε μια σταθμισμένη μέγιστη αντιστοίχιση στο γράφο G έτσι ώστε το σταθμισμένο άθροισμα όλων των επιλεγόμενων προς αντιστοίχιση ακμών να είναι ελάχιστο
 - 5: Μεταφορτώνουμε τις εργασίες στις διαφορετικές υπολογιστικές υποδομές: τοπική MYN, μακρινό YN ή την εκτελούμε στην ίδια τη συσκευή αποφασίζοντας με βάση τις ακμές που επιλέχθηκαν στη σταθμισμένη μέγιστη αντιστοίχιση που έγινε στο γράφο G
 - 6: **Επιστρέφει** το ελάχιστο ποσοστό υπολοιπόμενης ενέργειας ανάμεσα στις κινητές συσκευές
 - 7: **Τέλος Επανάληψης**
-

Θεώρημα 1. Δεδομένων μιας αρχιτεκτονικής YKN δύο επιπέδων, μιας χρονικής περιόδου και ενός συνόλου χρηστών που έχουν εργασίες προς εκτέλεση που μπορούν να μεταφορτωθούν, οι οποίοι υποθέτουμε ότι δεν έχουν εκ των προτέρων γνώση για το πότε και που θα εκτελεστούν οι εργασίες τους, υπάρχει αποδοτικός αλγόριθμος για το πρόβλημα της δυναμικής μεταφόρτωσης εργασιών, ο οποίος χρειάζεται $O((n + n_a \cdot H)^3) = O((n + n_a)^3)$ χρόνο για να αποφανθεί εάν $H \ll n$, όπου n είναι ο μέγιστος αριθμός χρηστών σε κάθε χρονοσχισημή, n_a ο μέγιστος αριθμός καναλιών ενός σημείου πρόσβασης ενώ H είναι το πλήθος των σημείων πρόσβασης μέσω των οποίων οι χρήστες μπορούν να έχουν πρόσβαση σε κάποια τοπική MYN.

Απόδειξη. Αρχικά θα αποδείξουμε ότι ο προτεινόμενος αλγόριθμος μας δίνει μια εφικτή λύση ώστε όλοι οι χρήστες κινητών συσκευών να μοιράζονται δίκαια τους πόρους των υπολογιστικών υποδομών παρά την ανταγωνιστικότητα. Ας θεωρήσουμε δύο χρήστες, A και B, με ποσοστό υπολοιπόμενης ενέργειας $r_A(t-1)$ και $r_B(t-1)$ μετά τη χρονοσχισημή $t-1$ αντίστοιχα. Ακόμη, υποθέτουμε ότι

$r_A(t-1) < r_B(t-1)$ και ότι ο χρήστης A έχει το χαμηλότερο ποσοστό υπολειπόμενης ενέργειας από όλους του χρήστες μετά τη χρονοσχισμή $t-1$.

Έχουμε ότι $r_A(t) = \frac{E_{\text{residual}}(u_A, t-1) - \Delta E_x^A(t)}{E(u_A)} = r_A(t-1) - \frac{\Delta E_x^A(t)}{E(u_A)}$ όπου $x = "l"$ ή $x = "r"$ ή $x = "d"$ ανάλογα με το σε ποια υπολογιστική υποδομή θα εκτελεστεί η εργασία του χρήστη A και $\Delta E_x^A(t)$ η ενέργεια που θα καταναλωθεί στην κινητή συσκευή του χρήστη A κατά τη διάρκεια της χρονοσχισμής t , όταν, δηλαδή, θα εκτελεστεί η εργασία στην υπολογιστική υποδομή x . Η ποσότητα $r_B(t) = r_B(t-1) - \frac{\Delta E_x^B(t)}{E(u_B)}$ μπορεί να οριστεί αντίστοιχα. Υποθέτουμε ότι $r_A(t) \leq r_B(t)$ όταν $x = "l"$ και οι δύο χρήστες ανταγωνίζονται για τη χρήση της τοπικής ΜΥΝ ενώ αυτή μπορεί να εξυπηρετήσει μόνο έναν εκ των δύο εκείνη τη χρονοσχισμή. Εάν ο A επιτύχει, τότε θα καταναλώσει λιγότερη ενέργεια απ'ότι αν εκτελούσε την εργασία στη συσκευή του ή την έστειλε για εκτέλεση στο απομακρυσμένο ΥΝ και ταυτόχρονα ο χρήστης B πρέπει να μεταφορτώσει την εργασία του στο απομακρυσμένο ΥΝ ή να την εκτελέσει στην ίδια του τη συσκευή. Με άλλα λόγια, όσον αφορά τη λύση που θα επιστρέψει ο αλγόριθμος για το γράφημα εκείνη τη χρονοσχισμή, το βάρος της ακμής που θα επιλεγεί για το χρήστη A είναι χαμηλότερο από αυτή που θα επιλεγεί για το χρήστη B. Διαφορετικά, αν για παράδειγμα ο χρήστης B επιτύχει και στείλει την εργασία του για εκτέλεση στην τοπική ΜΥΝ, θα σημαίνει πως υπάρχει και ένα άλλο μέγιστο ταίριασμα (maximum matching) στο γράφο $G(t)$ που οδηγεί σε ένα μικρότερο άθροισμα βαρών των ακμών με την εναλλαγή των ήδη επιλεγμένων ζευγαριών για τους χρήστες A και B στο ήδη υπάρχον μέγιστο ταίριασμα που αναφέραμε πριν. Αυτό έρχεται σε αντίθεση με το ότι το τωρινό μέγιστο ταίριασμα είναι ένα μέγιστο ταίριασμα με ελάχιστο βάρος. Στη συνέχεια αναλύεται η χρονική πολυπλοκότητα του εν λόγω αλγορίθμου. Η κατασκευή του γράφου $G(t)$ απαιτεί $O(|VR(t)| + |VM(t)| + |V_1(t)| \cdot |VL(t)|) = O(n + n + n \cdot n_a \cdot H) = O(n \cdot n_a \cdot H)$ χρόνο, αφού ο αριθμός των ακμών στο γράφο G είναι άνω φραγμένος από την τιμή $n \cdot n_a \cdot H$ και ο αριθμός των κόμβων στο γράφο G είναι άνω φραγμένος από την τιμή $n + n_a \cdot H$, όπου n ο μέγιστος αριθμός χρηστών σε κάθε χρονοσχισμή t και n_a ο μέγιστος αριθμός καναλιών σε κάθε σημείο πρόσβασης ενώ H είναι ο αριθμός των σημείων πρόσβασης στο περιβάλλον αρχιτεκτονικής ΥΚΝ δύο επιπέδων. Το να βρεθεί το σταθμισμένο μέγιστο ταίριασμα στο γράφο G , λοιπόν, απαιτεί $O((3 \cdot n + n_a \cdot H)^3) = O((n + n_a)^3)$ χρόνο χρησιμοποιώντας κάποιον αποδοτικό αλγόριθμο εύρεσης σταθμισμένου ταιριάσματος ελαχίστου κόστους (minimum cost weighted matching) [7], πχ. Hungarian Algorithm [20] (Βλ. Παράρτημα A), καθώς υποθέτουμε ότι $H \ll n$. \square

Κεφάλαιο 4

Υλοποίηση και Αξιολόγηση Αποτελεσμάτων

4.1 Χρησιμοποιούμενο Λογισμικό

Για την αξιολόγηση της απόδοσης του αλγορίθμου που παρουσιάστηκε παραπάνω εκτελέστηκε μια σειρά προσομοιώσεων. Για τις προσομοιώσεις αυτές χρησιμοποιήθηκε το λογισμικό CloudSim Plus.

Το CloudSim Plus [12] αποτελεί ένα πλαίσιο (framework) προσομοιώσεων ανοιχτού κώδικα (open source) σε περιβάλλον Java 8 το οποίο επιτρέπει τη μοντελοποίηση και την προσομοίωση διαφορών υπηρεσιών και λειτουργιών περιβάλλοντος υπολογιστικού νέφους σε επίπεδο υλικού αλλά και λογισμικού. Είναι ένα λογισμικό-εργαλείο που χρησιμοποιείται για τον πειραματισμό και την αξιολόγηση διάφορων σεναρίων και αλγορίθμων σε περιβάλλον ΥΝ. Ο προγραμματιστής έχει τη δυνατότητα να υλοποιήσει οποιοδήποτε νέο χαρακτηριστικό επιθυμεί να ερευνήσει επεκτείνοντας τους βασικούς μηχανισμούς που παρέχει το CloudSim Plus. Δηλαδή μπορεί να εφαρμόσει αλγορίθμους που θα αποφασίζουν πού και πώς θα εκτελείται κάθε εργασία σε περιβάλλον ΥΝ ώστε να επιτυγχάνει κάθε φορά το συγκεκριμένο στόχο του εκάστοτε αλγορίθμου. Τέτοιοι στόχοι μπορούν να είναι για παράδειγμα εξισορρόπηση φόρτου εργασίας, εξοικονόμηση ενέργειας, παρακολούθηση και διαχείριση κίνησης δικτύου, ελαχιστοποίηση κόστους με χρήση ελαστικών διεργασιών κ.ά. Ακόμη, μπορεί να δει τη συμπεριφορά αυτών τόσο σε μικρή όσο και σε μεγάλη κλίμακα μέσω των προσομοιώσεων και να έχει έτσι σωστά αποτελέσματα στη μελέτη του.

Το CloudSim Plus ξεκίνησε ως μια αυτόνομη, ανεξάρτητη έκδοση λογισμικού βασισμένο στο εργαλείο CloudSim [4], που αποτελεί ένα από τα πιο ευρέως χρησιμοποιούμενα εργαλεία για προσομοιώσεις σε περιβάλλον ΥΝ. Εξελίχθηκε και πλέον παρέχει στο χρήστη καλύτερη ιεραρχία των κλάσεων και του κώδικα γενικότερα, πιο πλήρες και οργανωμένο εγχειρίδιο χρήσης, ανακατασκευασμένα τμήματα του κώδικα που διευκολύνουν το χρήστη να κατανοήσει τη λειτουργία του αλλά και αρκετά παραδείγματα με τα οποία εξετάζονται αρκετά σενάρια περιβάλλοντος ΥΝ.

Οι προσομοιώσεις εκτελέστηκαν σε φορητό υπολογιστή Acer Aspire 5755G με τετραπύρρηνο επεξεργαστή Intel Core i7-2630QM και 4GB μνήμης RAM χρησιμοποιώντας το λογισμικό Eclipse IDE με έκδοση Neon (4.6).

4.2 Περιβάλλον Προσομοιώσεων

Σε αυτό το τμήμα θα αξιολογήσουμε την αποδοτικότητα του αλγορίθμου που αναφέρθηκε παραπάνω (Αλγόριθμος 1) μέσω αρκετών προσομοιώσεων. Ακόμη, θα εξεταστεί και η επιρροή διάφορων παραμέτρων στην απόδοση του αλγορίθμου αυτού.

Για τα πειράματα, θεωρούμε μια αρχιτεκτονική YKN δύο βαθμίδων. Αποτελείται από το απομακρυσμένο ΥΝ, από μία τοπική MYN και από ένα σύνολο χρηστών όπως αυτό φαίνεται και στο σχήμα 3.1. Η τοπική MYN διαθέτει 32 πυρήνες επεξεργαστή με ταχύτητα 3GHz. Το απομακρυσμένο ΥΝ παρέχει στο χρήστη επεξεργαστική ταχύτητα 5.5GHz. Οι αντίστοιχες ταχύτητες των κινητών συσκευών κυμαίνονται από 0.8 έως 2.2GHz. Η χωρητικότητα των μπαταριών των κινητών συσκευών βρίσκεται στο εύρος των 500 έως 1000Joules. Θεωρούμε έναν χώρο παρακολούθησης διαστάσεων 1000m × 1000m. Ο χώρος αυτός καλύπτεται εξ' ολοκλήρου από ασύρματο

δίκτυο 4G/LTE και το εύρος ζώνης αυτού κυμαίνεται στα 60 έως 80Mbps ανάλογα με τη θέση και τη χρονική στιγμή. Μέσω αυτού του δικτύου οι χρήστες έχουν πρόσβαση στο απομακρυσμένο ΥΝ. Ακόμα, υπάρχουν 25 σημεία πρόσβασης δικτύου WiFi τυχαία κατανομημένα στο χώρο. Κάθε ένα από αυτά διαθέτει 1 ή 2 κανάλια (channels). Το εύρος ζώνης κάθε καναλιού είναι από 40 έως 66Mbps. Η ακτίνα διάδοσης αυτών των σημείων πρόσβασης είναι 250m και η υπάρχει γραμμική εξασθένηση του σήματος τους και μείωση του εύρους ζώνης που παρέχουν έτσι ώστε σε απόσταση 250m να παρέχουν το 75% του αρχικού τους εύρους ζώνης.

Στις προσομοιώσεις που εκτελέστηκαν θεωρήσαμε ότι κάθε χρονοσχισμή διαρκεί 5 δευτερόλεπτα και ότι η χρονική διάρκεια εποπτείας και μετρήσεων είναι 1000 χρονοσχισμές (5000 δευτερόλεπτα). Κάθε χρήστης σε αυτό το χρονικό διάστημα έχει 80 έως 120 εργασίες προς εκτέλεση. Σε κάθε χρονοσχισμή το πολύ 1 εργασία μπορεί να υπάρχει προς εκτέλεση ανά χρήστη. Το πλήθος δεδομένων που μπορεί να σχετίζεται με μια εργασία και θα πρέπει ενδεχομένως να μεταβιβαστεί μέσω του δικτύου παίρνει τιμές από 1 έως 5Mb. Το μέγεθος, όσον αφορά το πλήθος εντολών κάθε εργασίας, κυμαίνεται από 10000 έως 40000MIPS. Αν ένας χρήστης αποφασιστεί να μεταφορτώσει κάποια εργασία στην τοπική MYN ή στο μακρινό ΥΝ τότε το ενεργειακό κόστος σε κατάσταση αδράνειας και μεταβίβασης δεδομένων φαίνεται στον ακόλουθο πίνακα.

Κατάσταση	WiFi	4G/LTE
Μετάδοση Δεδομένων	[1, 1.8]	[1.5, 3]
Αδράνεια	[0.5, 1]	[1, 2]

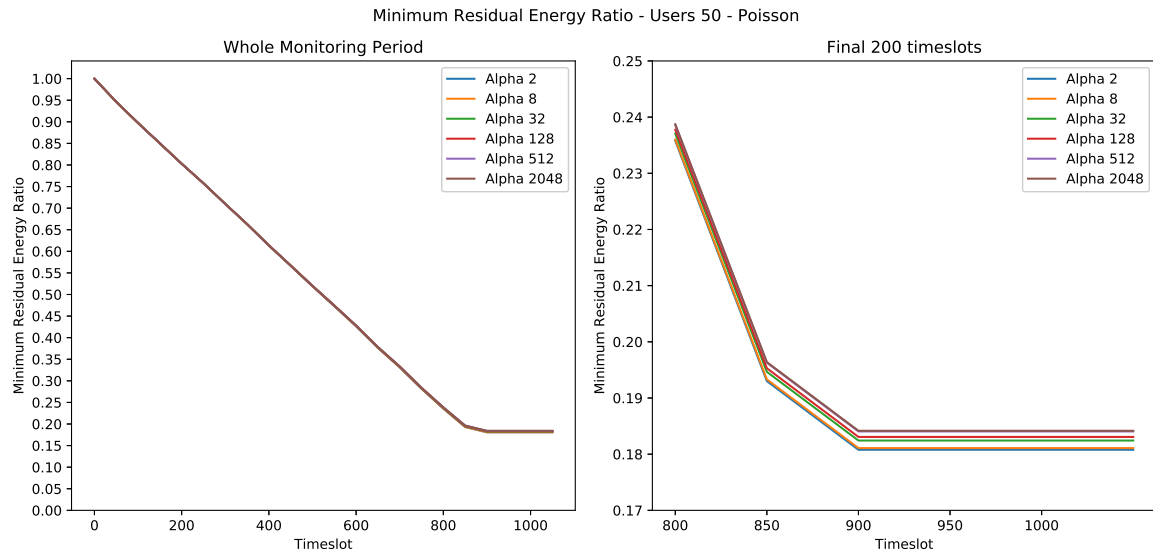
Πίνακας 4.1: Ενεργειακά Κόστη ανά Δίκτυο

Στα πειράματα που εκτελέστηκαν ο χρόνος αφίξεως των εργασιών των χρηστών ακολουθεί είτε Poisson είτε ομοιόμορφη κατανομή σε ολόκληρη την περίοδο παρατήρησης. Στα αποτελέσματα παρακάτω θα επισημαίνεται ο διαχωρισμός αυτών των δύο περιπτώσεων. Για κάθε πείραμα εκτελέστηκαν και λήφθηκαν μετρήσεις από 20 περιπτώσεις με την κάθε παραμετροποίηση, καθώς υπάρχει τυχαιότητα, και τα αποτελέσματα που παρατίθενται στη συνέχεια αποτελούν τις μέσες τιμές αυτών των 20 περιπτώσεων κάθε ξεχωριστού πειράματος.

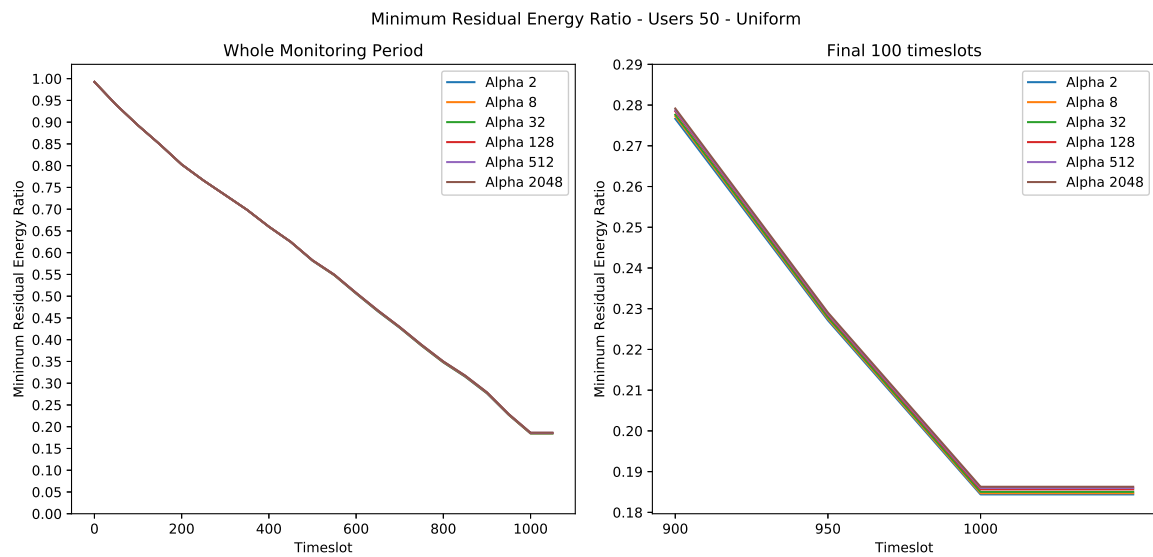
4.3 Αξιολόγηση Αποτελεσμάτων

4.3.1 Εξέταση Επιρροής της Παραμέτρου α της Συνάρτησης Κόστους

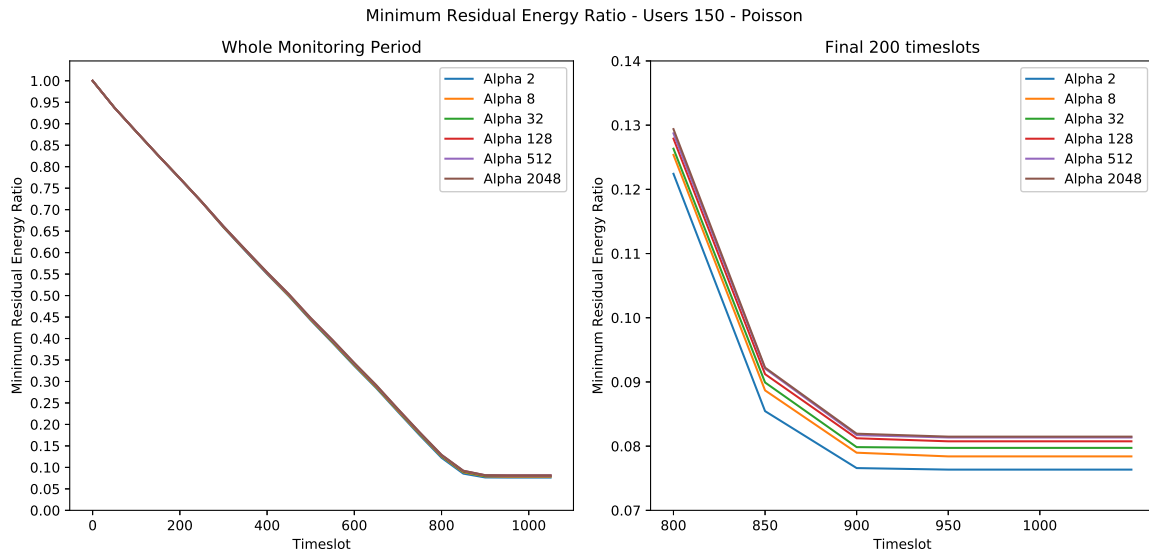
Αρχικά, εξετάστηκε η επιρροή της παραμέτρου α που υπάρχει στις συναρτήσεις κόστους όπως ορίστηκαν παραπάνω (3.7,3.8,3.9). Το περιβάλλον προσομοιώσεων είναι αυτό που περιγράφηκε προηγουμένως. Η μόνη τιμή που μεταβαλλόταν ήταν αυτή του α . Εξετάστηκε η εν λόγω παράμετρος για τις τιμές $2^1, 2^3, 2^5, 2^7, 2^9, 2^{11}$. Οι προσομοιώσεις έγιναν τόσο για Poisson όσο και για ομοιόμορφη κατανομή χρόνου αφίξεως των εργασιών των χρηστών. Ακόμη, εκτελέστηκαν για περιπτώσεις όπου οι χρήστες ήταν 50, 100, 150 και 200. Τα αποτελέσματα που λήφθηκαν παρουσιάζονται στη συνέχεια, στα σχήματα 4.1, 4.2, 4.3, 4.4, 4.5, 4.6 και 4.7. Κάθε τιμή που παρουσιάζεται στις ακόλουθες γραφικές παραστάσεις αποτελεί το μέσο όρο των πειραμάτων που εκτελέστηκαν για κάθε παραμετροποίηση.



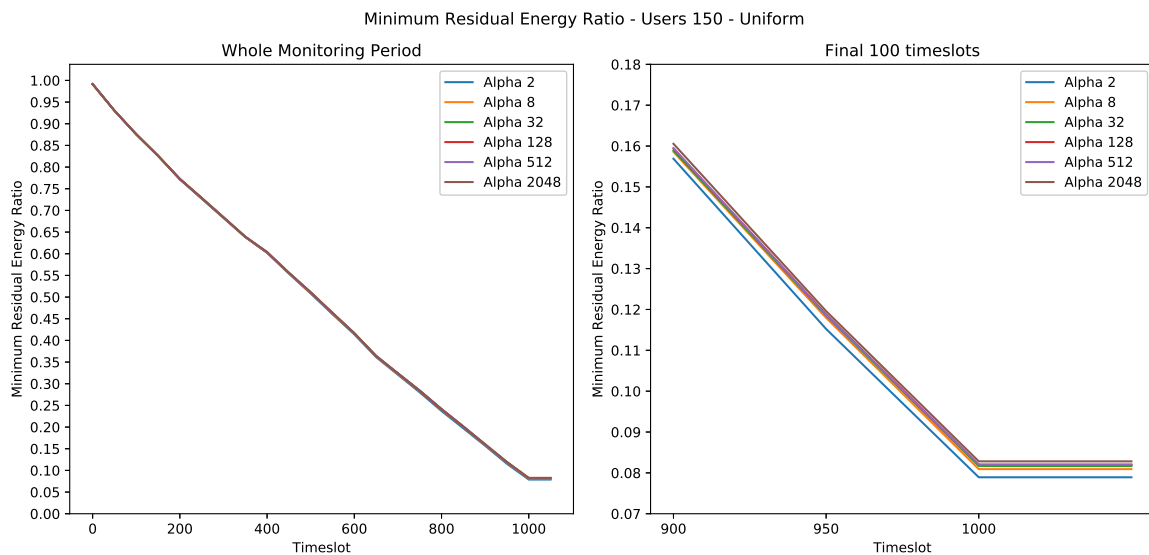
Σχήμα 4.1: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α με 50 χρήστες και Poisson κατανομή άφιξης εργασιών



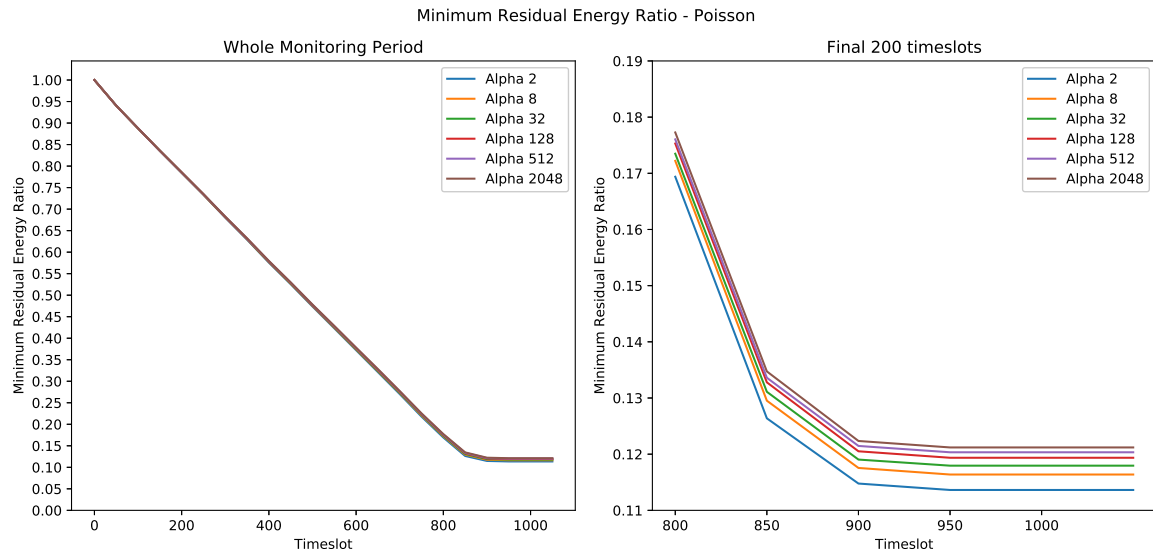
Σχήμα 4.2: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α με 50 χρήστες και ομοιόμορφη κατανομή άφιξης εργασιών



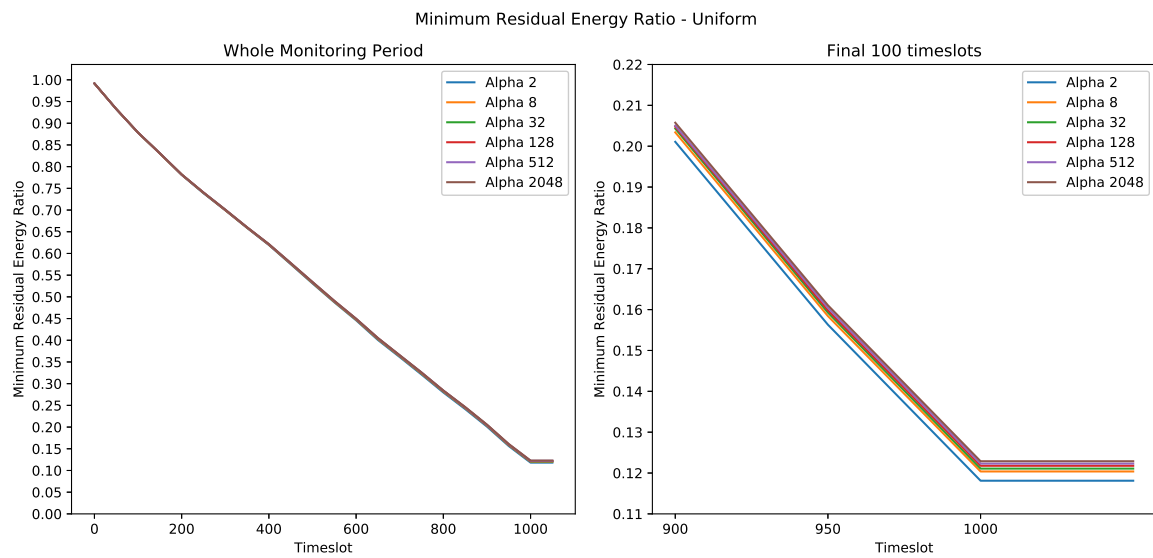
Σχήμα 4.3: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α με 150 χρήστες και Poisson κατανομή άφιξης εργασιών



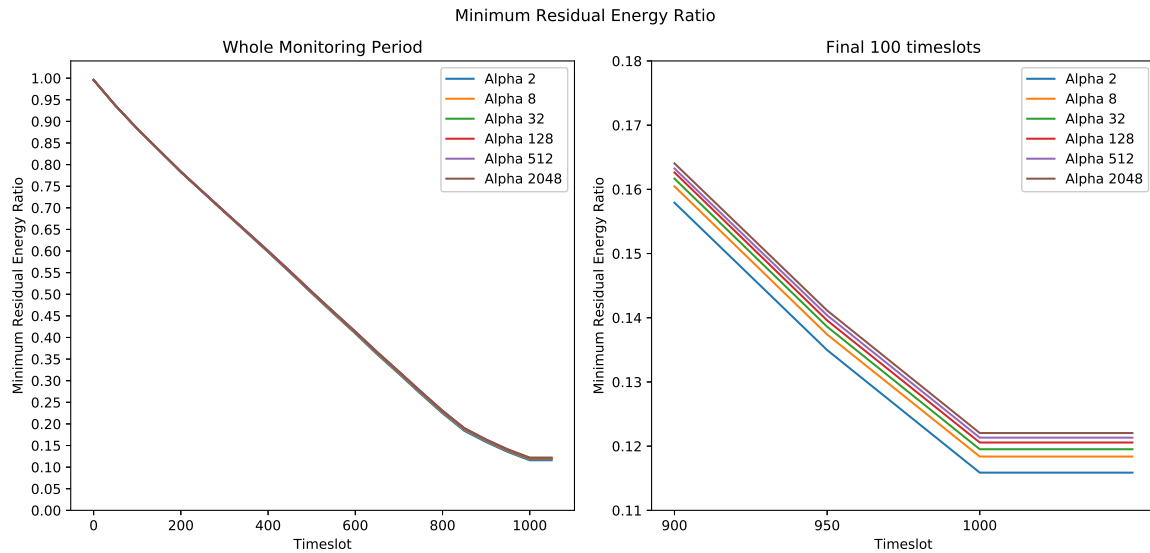
Σχήμα 4.4: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α με 150 χρήστες και ομοιόμορφη κατανομή άφιξης εργασιών



Σχήμα 4.5: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α και για όλους τους συνδυασμούς των χρηστών (50,100,150,200) αθροιστικά με Poisson κατανομή άφιξης εργασιών



Σχήμα 4.6: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α και για όλους τους συνδυασμούς των χρηστών (50,100,150,200) αθροιστικά με ομοιόμορφη κατανομή άφιξης εργασιών



Σχήμα 4.7: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για διάφορες τιμές του α , για όλους τους συνδυασμούς χρηστών (50,100,150,200) και για τις δύο διαφορετικές κατανομές άφιξης εργασιών (Poisson,ομοιόμορφη) αθροιστικά

Από τις παραπάνω γραφικές βλέπουμε ότι για όλες τις τιμές του α η πορεία της γραφικής παράστασης της ελάχιστης υπολειπόμενης ενέργειας είναι παρόμοια σε κάθε περίπτωση ξεχωριστά. Για το λόγο αυτό έχουμε μεγεθύνει το τέλος των γραφικών παραστάσεων για να είναι πιο φανερές οι τελικές διαφορές. Παρατηρούμε ότι οι διαφορές είναι εντονότερες όταν έχουμε να κάνουμε με πολλούς χρήστες απ' ότι με λίγους. Αυτό είναι λογικό καθώς έχοντας περισσότερους χρήστες στην προσομοίωση υπάρχουν και περισσότερες εργασίες και το πλήθος των εργασιών καθιστά τη λήψη διαφορετικής απόφασης ανάλογα με το α πιο πιθανή. Έτσι, οι διαφορές είναι πιο εμφανείς. Ακόμη, όταν ο χρόνος άφιξης των εργασιών ακολουθεί *Poisson* κατανομή τα αποτελέσματά μας είναι πιο ξεκάθαρα όπως φαίνεται παραπάνω. Αυτό οφείλεται κυρίως στη μαζική άφιξη των εργασιών στην κατανομή *Poisson* που έχει σαν αποτέλεσμα ο αλγόριθμος MBM, που εξετάζεται, να λαμβάνει διαφορετικές αποφάσεις ανάλογα με την τιμή του α σε τέτοιες περιπτώσεις. Παρατίθεται και η κατανομή των εργασιών στις υπολογιστικές υποδομές κατά μέσο όρο για τις τιμές της παραμέτρου α . Οι σχετικά μικρές διαφορές στην απόδοση επιβεβαιώνονται και από τις μικρές διαφορές στο διαμορισμό των εργασιών στο απομακρυσμένο ΥΝ, την τοπική ΜΥΝ και τις συσκευές (πίνακας 4.2).

	Απομακρυσμένο ΥΝ	Τοπική ΜΥΝ	Κινητές Συσκευές
$\alpha=2^1$	27.40%	56.98%	15.62%
$\alpha=2^3$	27.37%	57.00%	15.63%
$\alpha=2^5$	27.33%	57.03%	15.64%
$\alpha=2^7$	27.30%	57.05%	15.65%
$\alpha=2^9$	27.27%	57.08%	15.65%
$\alpha=2^{11}$	27.27%	57.10%	15.63%

Πίνακας 4.2: Κατανομή των εργασιών στις υπολογιστικές υποδομές για διαφορετικές τιμές του α

Αυτό που εξάγεται σαν συμπέρασμα από τις παραπάνω γραφικές παραστάσεις είναι ότι η τιμή του $\alpha = 2^{11}$ είναι η καλύτερη από όσες δοκιμάστηκαν. Σε όλες τις γραφικές παραστάσεις, είτε έχουμε πολλούς είτε λίγους χρήστες, είτε *Poisson* είτε ομοιόμορφη κατανομή χρόνου άφιξης των εργασιών, όταν χρησιμοποιήθηκε η τιμή 2^{11} για την παράμετρο α το τελικό ελάχιστο ποσοστό υπολειπόμενης ενέργειας μεταξύ των χρηστών καταλήγει να είναι το μεγαλύτερο. Αν και οι δια-

φορές είναι μικρές, για τα επόμενα πειράματα πρέπει να σημειωθεί ότι ως τιμή του α λήφθηκε η βέλτιστη που βρέθηκε, δηλαδή θέσαμε $\alpha = 2^{11}$.

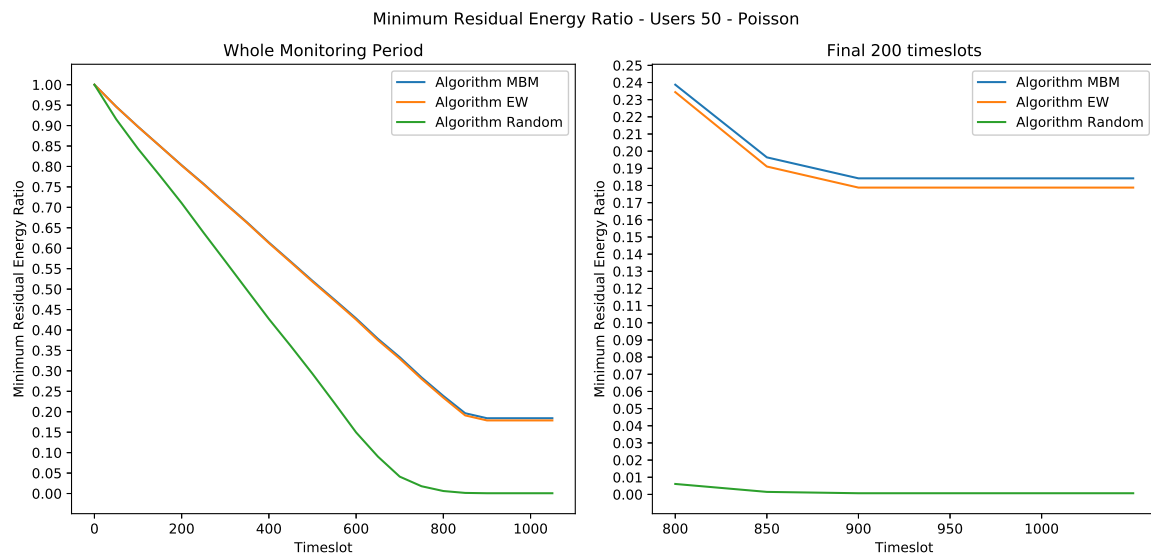
4.3.2 Σύγκριση Απόδοσης τριών Αλγορίθμων (MBM, EW, Random)

Στο κομμάτι αυτό πραγματοποιείται μια σύγκριση του κύριου αλγορίθμου MBM με δύο άλλους που χρησιμοποιούν διαφορετικές ευριστικές για τη λήψη της απόφασης δρομολόγησης κάθε εργασίας.

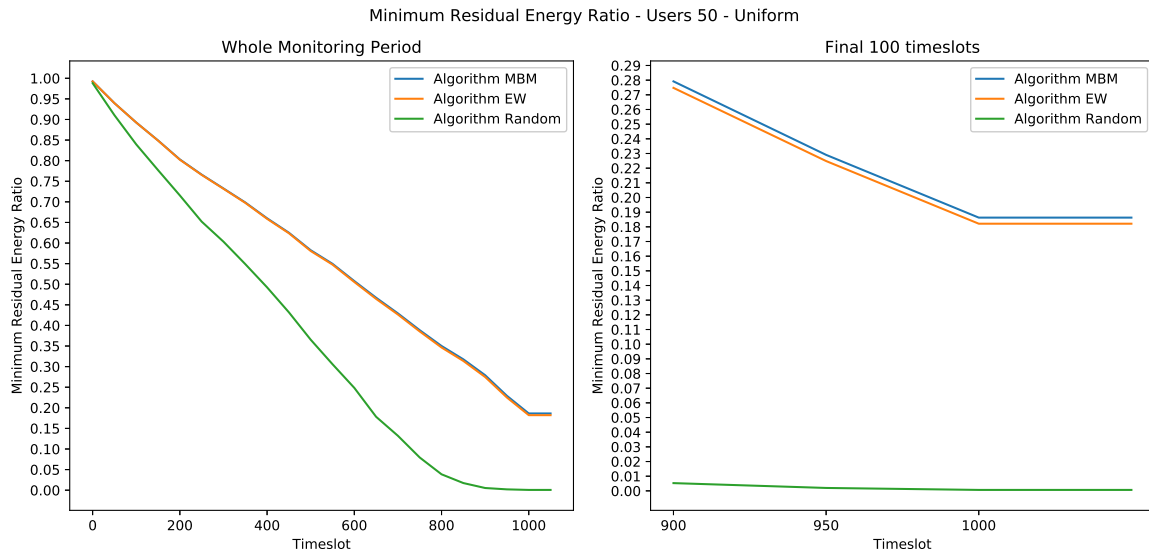
Ο πρώτος αλγόριθμος είναι παρόμοιος με τον MBM. Η διαφορά τους έγκειται στο ότι τα βάρη στο διμερές γράφο που κατασκευάζεται σε κάθε χρονισχισμή δεν προκύπτουν από κάποια συνάρτηση κόστους αλλά είναι το πραγματικό ενεργειακό κόστος για κάθε εργασία όπως προκύπτει από τις εξισώσεις (3.3), (3.4), (3.5). Με βάση αυτό το γράφο και εφαρμόζοντας τον ίδιο αλγόριθμο βρίσκεται σε κάθε χρονοσχιμή ένα σταθμισμένο μέγιστο ταίριασμα και από αυτό προκύπτουν οι αποφάσεις για τη μεταφόρτωση ή μη κάθε εργασίας. Αυτόν τον αλγόριθμο τον ονομάζουμε EW.

Ο δεύτερος αλγόριθμος είναι απλούστερος. Συγκεντρώνει σε κάθε χρονοσχιμή τις εργασίες, ελέγχει σε ποιες υπολογιστικές υποδομές μπορεί να εκτελεστεί η κάθε μία και αποφασίζει τυχαία αν μια εργασία θα μεταφορτωθεί στο μακρινό YN, στην τοπική MYN ή θα εκτελεστεί από την ίδια τη συσκευή χωρίς να λαμβάνει υπόψη της το ενεργειακό κόστος. Αυτόν τον αλγόριθμο τον ονομάζουμε Τυχαίο (Random).

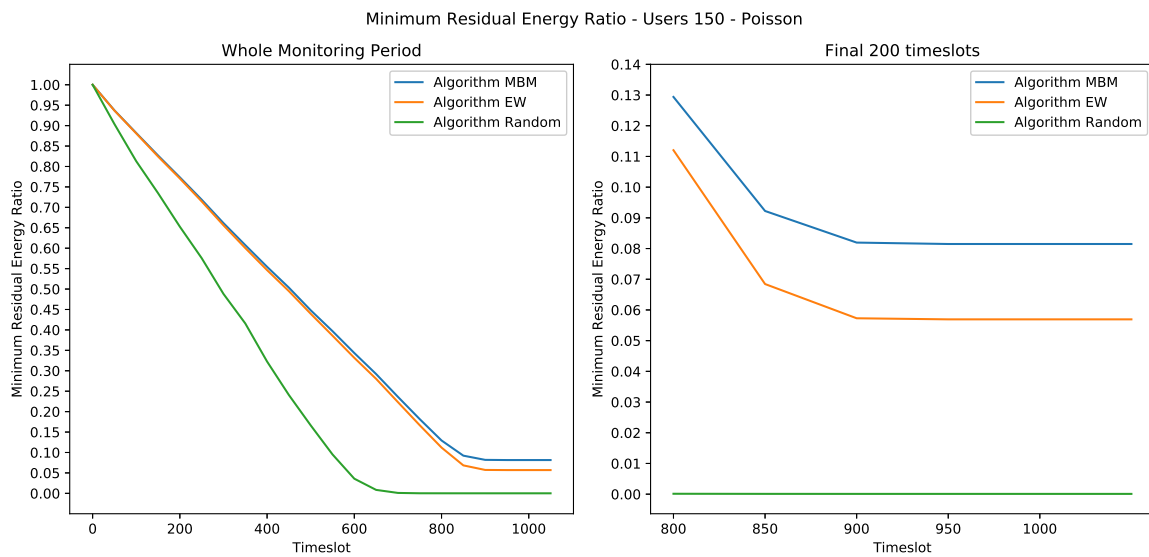
Για τη σύγκριση της απόδοσης αυτών των τριών αλγορίθμων εκτελέστηκαν προσομοιώσεις τόσο για Poisson όσο και για ομοιόμορφη κατανομή χρόνου αφίξεως των εργασιών των χρηστών. Ακόμη, εκτελέστηκαν για περιπτώσεις όπου οι χρήστες ήταν 50, 100, 150 και 200. Τα αποτελέσματα που λήφθηκαν παρουσιάζονται στη συνέχεια, στα σχήματα 4.8, 4.9, 4.10, 4.11, 4.12. Κάθε τιμή που παρουσιάζεται στις ακόλουθες γραφικές παραστάσεις αποτελεί το μέσο όρο των πειραμάτων που εκτελέστηκαν για κάθε παραμετροποίηση.



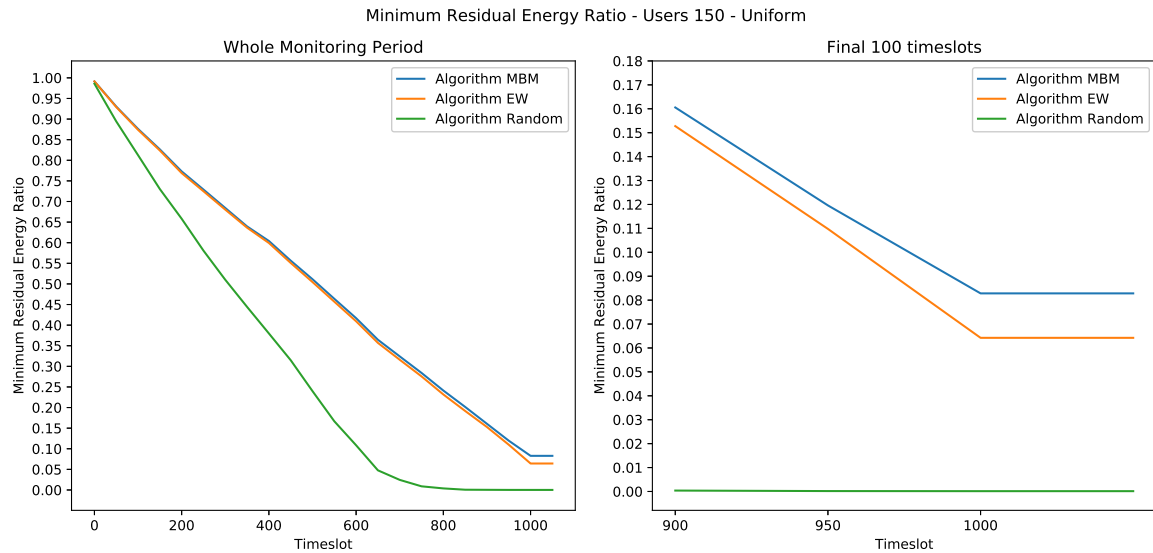
Σχήμα 4.8: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους με 50 χρήστες και Poisson κατανομή άφιξης εργασιών



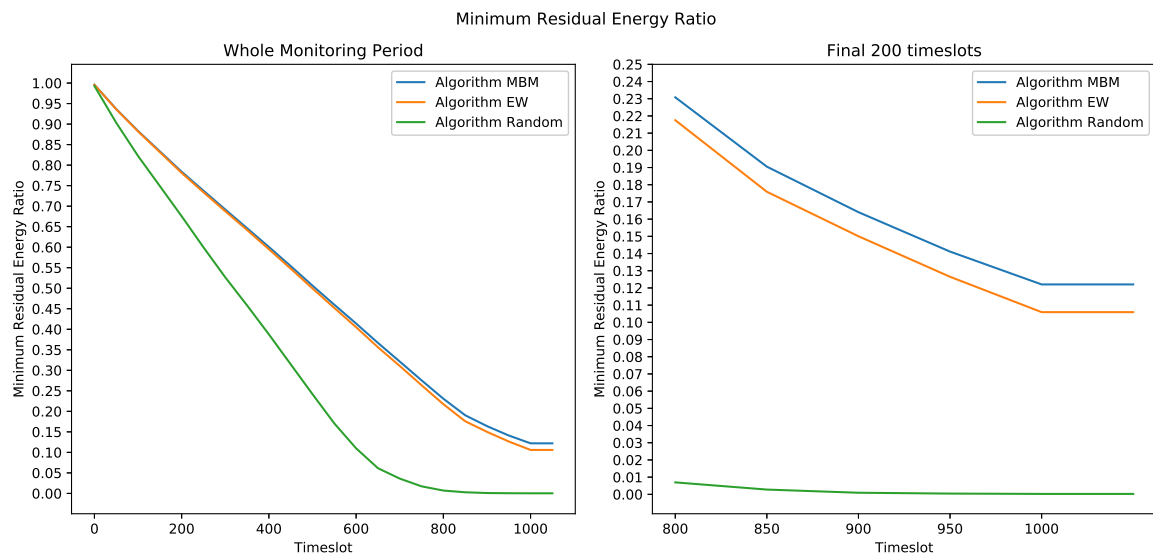
Σχήμα 4.9: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους με 50 χρήστες και ομοιόμορφη κατανομή άφιξης εργασιών



Σχήμα 4.10: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους με 150 χρήστες και Poisson κατανομή άφιξης εργασιών



Σχήμα 4.11: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους με 150 χρήστες και ομοιόμορφη κατανομή άφιξης εργασιών



Σχήμα 4.12: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για τους 3 Αλγορίθμους για όλους τους συνδυασμούς χρηστών (50,100,150,200) και για τις δύο διαφορετικές κατανομές άφιξης εργασιών (Poisson,ομοιόμορφη) αθροιστικά

Από τις παραπάνω γραφικές παραστάσεις γίνεται φανερή η αποδοτικότητα του αλγορίθμου MBM σε σχέση με τους άλλους δύο. Αρχικά, παρατηρούμε ότι οι αλγόριθμοι MBM και EW έχουν αρκετά μεγάλη διαφορά από τον Random ήδη από τις πρώτες 400 χρονοσχισμές. Ο αλγόριθμος της τυχαίας επιλογής είναι αρκετά κακός καθώς σχεδόν σε κάθε προσομοίωση καταλήγει να υπάρχει χρήστης χωρίς υπολειπόμενη ενέργεια. Αντίθετα, με άλλες επιλογές, όπως αυτές που κάνουν οι αλγόριθμοι MBM και EW βλέπουμε ότι το ελάχιστο ποσοστό υπολειπόμενης ενέργειας μπορεί να κρατηθεί σε καλά επίπεδα σχετικά με το πλήθος των εργασιών που έχει προς εκτέλεση ο κάθε χρήστης. Και οι δύο αλγόριθμοι παρουσιάζουν παρόμοια συμπεριφορά. Ωστόσο, μπορεί να παρατηρηθεί ότι σε όλες τις γραφικές παραστάσεις η καμπύλη του MBM είναι συνέχεια πάνω από την αντίστοιχη του EW και τελικά καταλήγει σε υψηλότερο ποσοστό υπολειπόμενης ενέργειας όπως

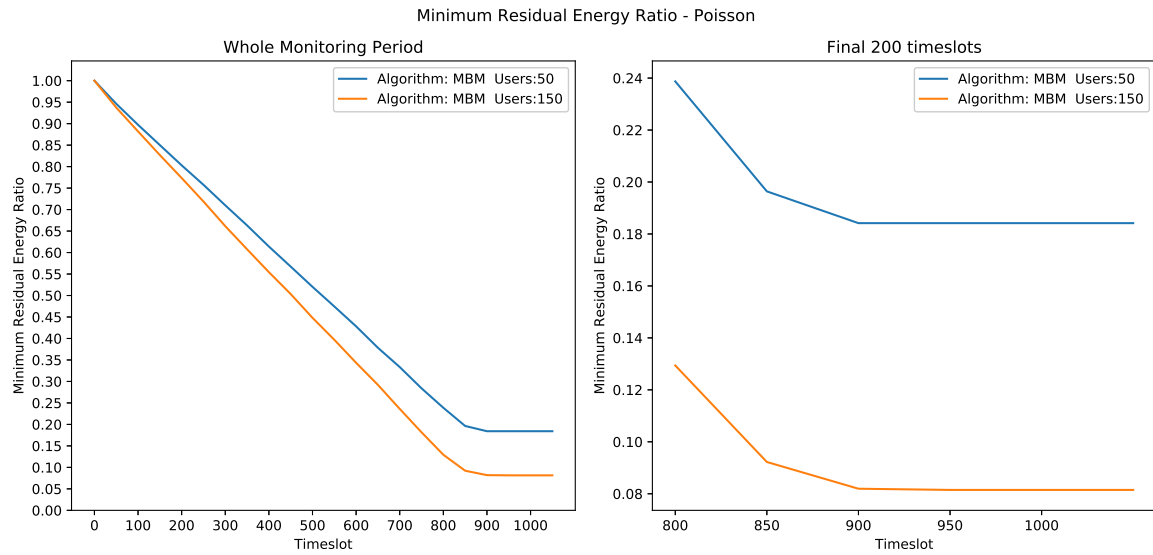
φαίνεται στα μεγενθυμένα σχήματα της δεξιάς στήλης. Η διαφορά τους κυμαίνεται από 2% έως 6% σε όλες τις περιπτώσεις. Μπορούμε να παρατηρήσουμε ότι οι διαφορές μεταξύ των δύο αυτών αλγορίθμων (MBM,EW) είναι πιο εμφανείς όταν έχουμε περισσότερους χρήστες (4.10, 4.11) καθώς τότε λαμβάνονται περισσότερες διαφορετικές αποφάσεις στις δύο ευριστικές για τη μεταφόρτωση λόγω των περισσότερων εργασιών. Ακόμη, οι δύο αλγόριθμοι έχουν αρκετά παρόμοια συμπεριφορά στις περιπτώσεις που επιλέγεται ομοιόμορφη κατανομή για το χρόνο άφιξης των εργασιών και αυτό είναι λογικό καθώς διαμοιράζονται οι εργασίες στο χρόνο ώστε πολλές από τις επιλογές μεταφόρτωσης είναι ίδιες ανεξάρτητα από το αν τα βάρη στο γράφο θα είναι τα πραγματικά ενεργειακά κόστη (EW) ή το αποτέλεσμα που δίνει η συνάρτηση κόστους (MBM). Στο τελευταίο σχήμα (4.12) φαίνονται συγκεντρωτικά τα αποτελέσματα των προσομοιώσεων τόσο για ομοιόμορφη όσο και για Poisson κατανομή άφιξης εργασιών, για 50, 100, 150, 200 χρήστες. Βλέπουμε ότι και από το μέσο όρο αυτών, δηλαδή σε γενικότερη περίπτωση, ο αλγόριθμος MBM δίνει καλύτερα αποτελέσματα από τους άλλους δύο (EW, Random). Αυτό οφείλεται στο γεγονός ότι λαμβάνει υπόψη κατά την κοστολόγηση μιας μεταφόρτωσης το ποσοστό υπολειπόμενης ενέργειας της κάθε συσκευής σε αντίθεση με τον αλγόριθμο EW που εξετάζει μόνο την τιμή της πραγματικής κατανάλωσης ενέργειας κάθε εργασίας. Αυτό επιβεβαιώνεται και από την κατανομή των εργασιών στις υπολογιστικές υποδομές που φαίνεται στον πίνακα 4.3, καθώς ο αλγόριθμος MBM επιλέγει και μεταφορτώνει περισσότερες εργασίες στην τοπική MYN που είναι γενικά και η πιο συμφέρουσα "ενεργειακά" επιλογή. Αν και η διαφορά ποσοστιαία είναι μικρή παίζει σημαντικό ρόλο στην τελική στάθμη του ελαχίστου ποσοστού υπολειπόμενης ενέργειας.

	Απομακρυσμένο YN	Τοπική MYN	Κινητές Συσκευές
MBM	27.33%	57.04%	15.63%
EW	27.44%	56.90%	15.66%
Random	31.59%	36.78%	31.63%

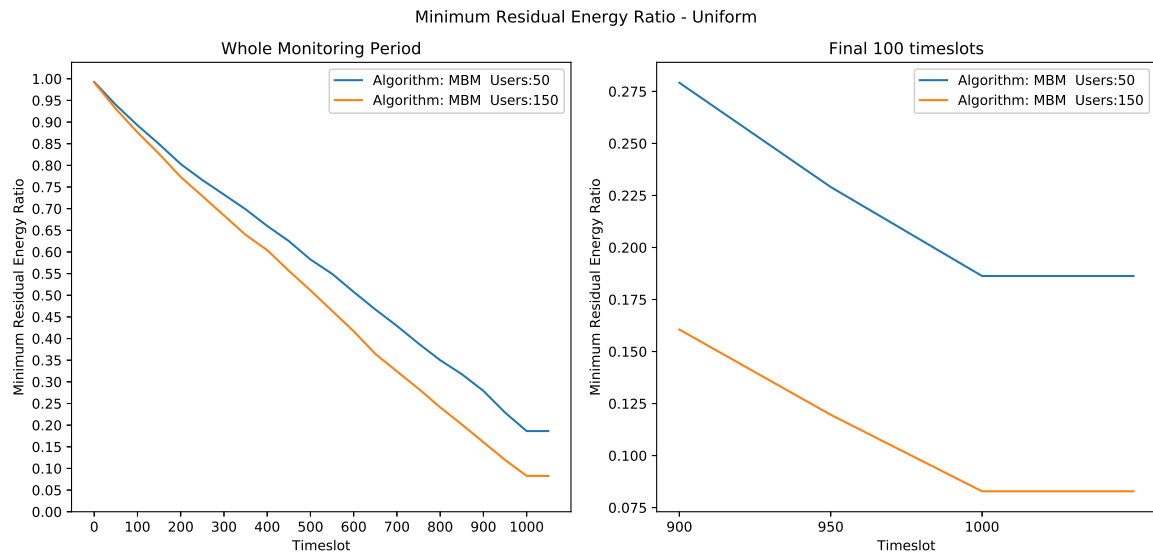
Πίνακας 4.3: Κατανομή των εργασιών στις υπολογιστικές υποδομές για διαφορετικούς αλγορίθμους μεταφόρτωσης

4.3.3 Εξέταση Απόδοσης του Αλγορίθμου MBM για διαφορετικό πλήθος Χρηστών στον υπό παρατήρηση χώρο

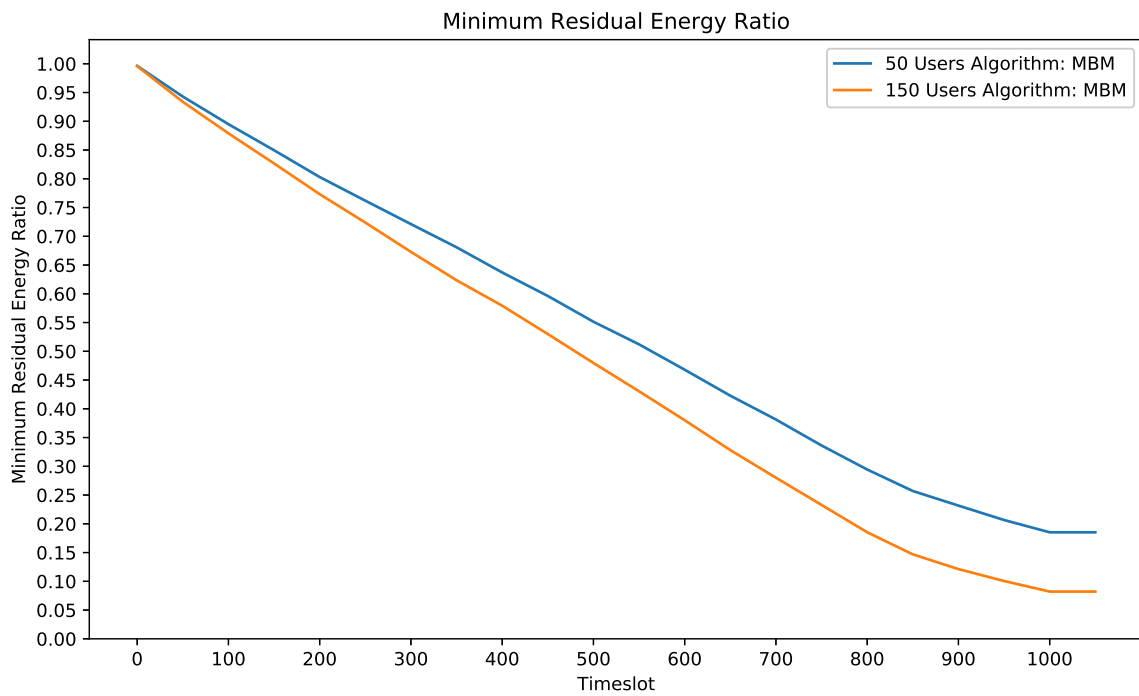
Στην ενότητα αυτή θα εξεταστεί η απόδοση του αλγορίθμου MBM όταν έχουμε διαφορετικό πλήθος χρηστών στον υπό παρατήρηση χώρο. Για το σκοπό αυτό εκτελέστηκαν προσομοιώσεις με 50 και 150 χρήστες ώστε να δούμε πως η διαφορά στον αριθμό τους επηρεάζει τόσο το ελάχιστο ποσοστό υπολειπόμενης ενέργειας όσο και το μέσο χρόνο απόκρισης των εργασιών που μεταφορτώνονται στο μακρινό YN και στην τοπική MYN. Υπενθυμίζεται ότι στο χώρο θεωρούμε ότι υπάρχουν 25 σημεία πρόσβασης δικτύου WiFi μέσω των οποίων γίνεται η μεταφόρτωση στην τοπική MYN. Τα αποτελέσματα που λήφθηκαν παρουσιάζονται στη συνέχεια, στα σχήματα 4.13, 4.14, 4.15, 4.16, 4.17a και 4.17b.



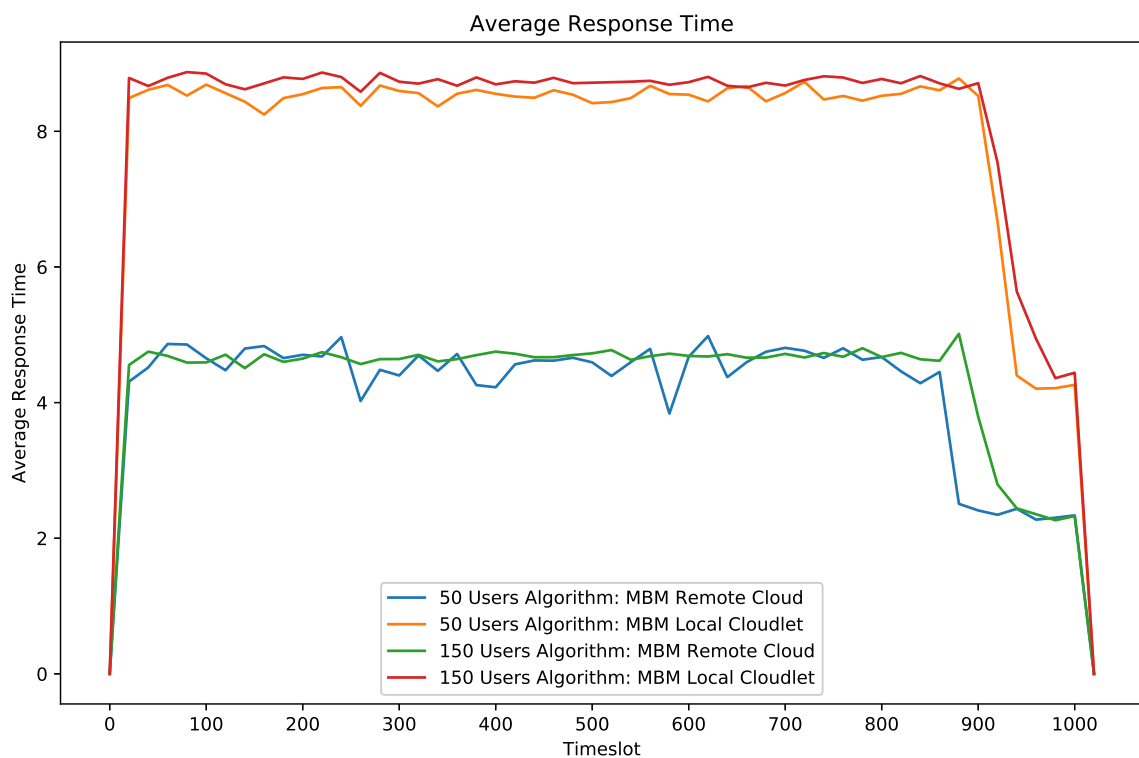
Σχήμα 4.13: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για μεταβαλλόμενο Αριθμό Χρηστών με Poisson κατανομή άφιξης εργασιών



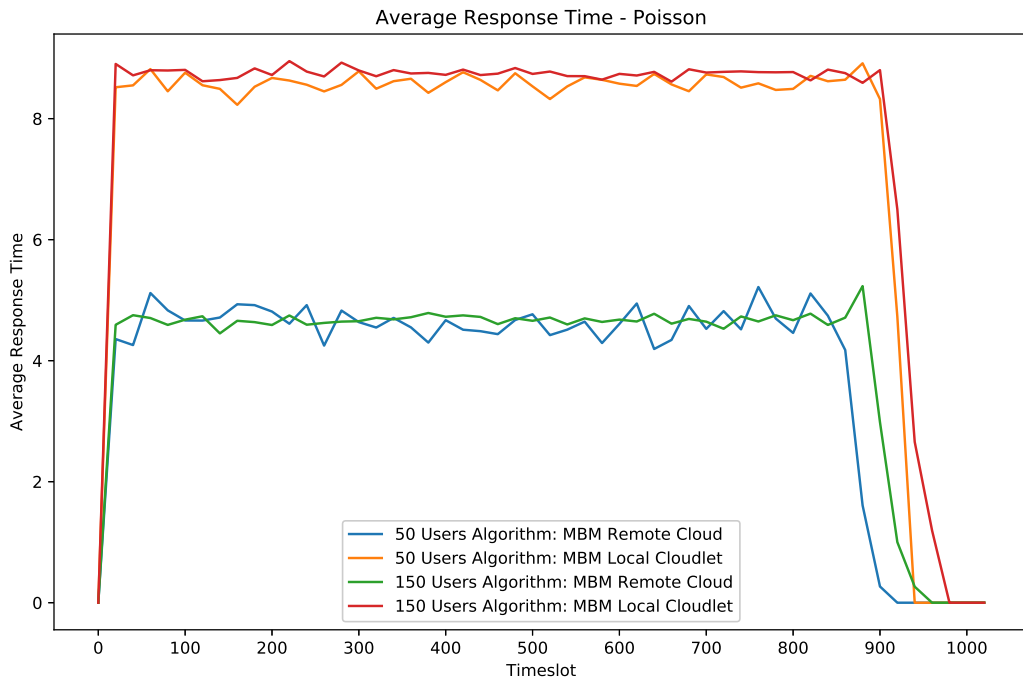
Σχήμα 4.14: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για μεταβαλλόμενο Αριθμό Χρηστών με ομοιόμορφη κατανομή άφιξης εργασιών



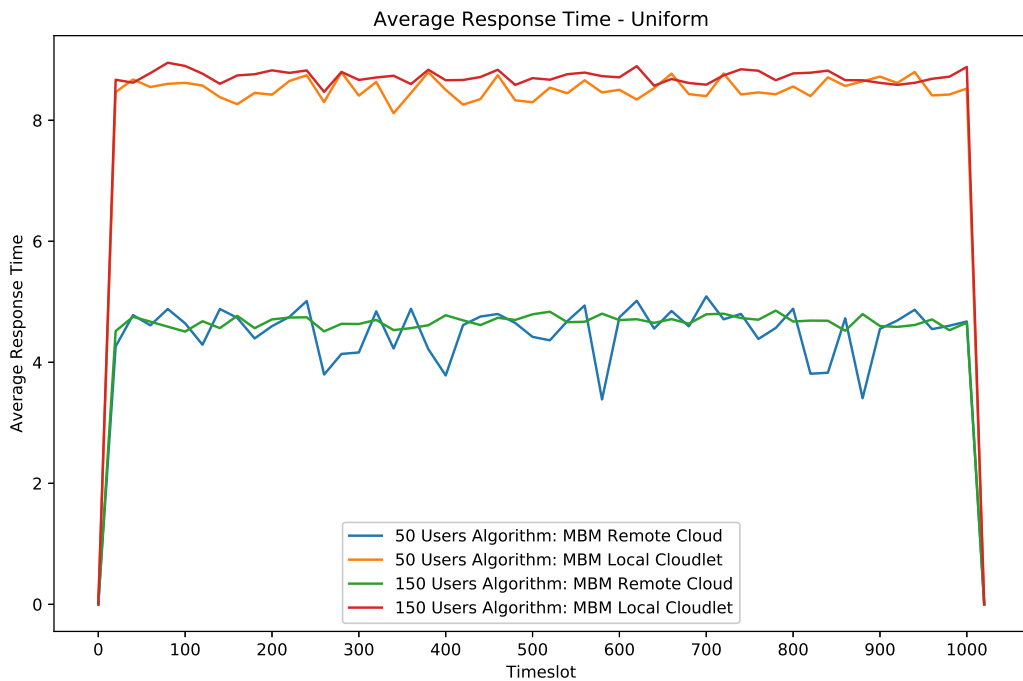
Σχήμα 4.15: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας για μεταβαλλόμενο Αριθμό Χρηστών



Σχήμα 4.16: Μέσος Χρόνος Απόκρισης Τοπικής ΜΥΝ και απομακρυσμένου ΥΝ για μεταβαλλόμενο Αριθμό Χρηστών



(a)



(b)

Σχήμα 4.17: Μέσος Χρόνος Απόκρισης Τοπικής ΜΥΝ και απομακρυσμένου ΥΝ για μεταβαλλόμενο Αριθμό Χρηστών με Poisson (4.17a) και ομοιόμορφη (4.17b) κατανομή άφιξης εργασιών

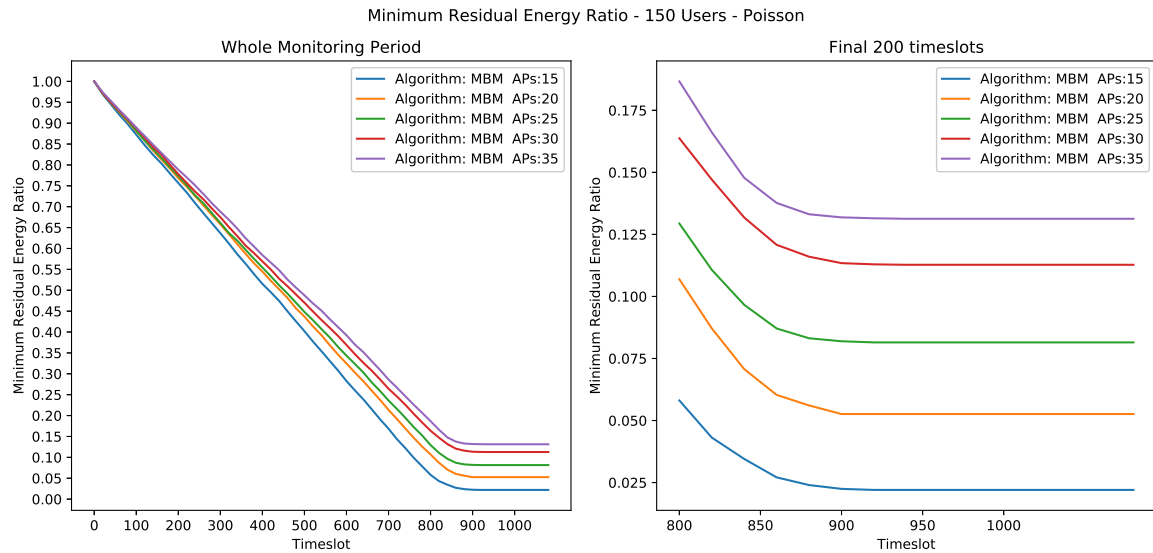
Από τις τρεις πρώτες γραφικές παραστάσεις (4.13, 4.14, 4.15) εξάγεται το συμπέρασμα ότι όσο περισσότεροι είναι οι χρήστες στον υπό παρατήρηση χώρο τόσο χαμηλότερο καταλήγει τελικά να είναι το ελάχιστο ποσοστό υπολειπόμενης ενέργειας μεταξύ αυτών. Αυτό εξηγείται καθώς όταν αυξάνονται οι χρήστες, με δεδομένο πλήθος σημείων πρόσβασης δικτύου WiFi, αυξάνεται και ο αριθμός των εργασιών που θα αναγκαστούν να εκτελεστούν στο απομακρυσμένο ΥΝ ή στις ίδιες τις συσκευές των χρηστών καθώς τα διαθέσιμα κανάλια για μεταφόρτωση στην τοπική ΜΥΝ,

που αποτελεί την πιο συμφέρουσα επιλογή από ενεργειακή άποψη, είναι περιορισμένα σε αριθμό. Έτσι, έχοντας περισσότερες εργασίες να εκτελούνται στις συσκευές και στο απομακρυσμένο YN υπάρχει και μεγαλύτερη κατανάλωση ενέργειας, πράγμα που είναι φανερό και στις καμπύλες που φαίνονται στα τρία σχήματα. Παρατηρούμε ότι αυτή η συμπεριφορά υπάρχει ανεξάρτητα από την κατανομή που ακολουθεί ο χρόνος άφιξης των εργασιών.

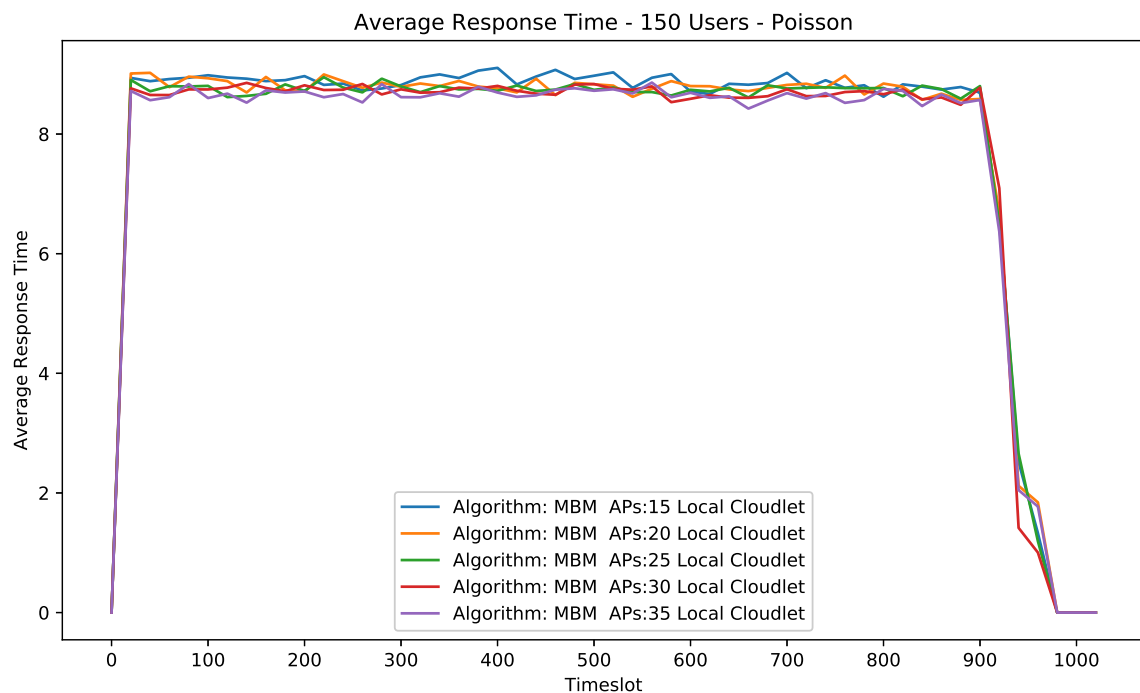
Στις επόμενες τρεις γραφικές παραστάσεις (4.16, 4.17a και 4.17b) φαίνονται οι μέσοι χρόνοι απόκρισης των εργασιών που εκτελούνται στο απομακρυσμένο YN και στην τοπική MYN. Είναι προφανές ότι οι μέσοι χρόνοι απόκρισης στο απομακρυσμένο YN είναι αρκετά χαμηλότεροι από αυτούς της τοπικής MYN και αυτό οφείλεται στην κατά πολύ μεγαλύτερη επεξεργαστική ισχύ που διαθέτει. Υπενθυμίζεται πως η επεξεργαστική ταχύτητα του απομακρυσμένου YN είναι $5GHz$ ενώ της τοπικής MYN $3GHz$. Ακόμη, παρατηρείται ότι με την αύξηση των χρηστών αυξάνεται ελάχιστα ο μέσος χρόνος απόκρισης τόσο στο απομακρυσμένο YN όσο και στην τοπική MYN παρά το γεγονός ότι μεταφορτώνονται σε αυτές τις υπολογιστικές υποδομές περισσότερες εργασίες. Αυτό σημαίνει ότι οι επεξεργαστικές δυνατότες αυτών είναι ικανές για την άμεση εκτέλεση των εργασιών χωρίς να αναγκαστούν εκείνες να μπουν σε κάποια ουρά αναμονής. Αν το πλήθος των χρηστών, και συνεπώς των εργασιών, αυξανόταν ακόμη περισσότερο θα ήταν λογικό και οι αντίστοιχες τιμές για το μέσο χρόνο απόκρισης να αυξάνονται στα χρονικά σημεία όπου πληθώρα εργασιών θα έπρεπε να εκτελεστεί ταυτόχρονα. Αυτό αφορά κυρίως την τοπική MYN καθώς διαθέτει περιορισμένους πόρους σε αντίθεση με το απομακρυσμένο YN που κατέχει ικανή επεξεργαστική δύναμη για μεγάλο φόρτο εργασιών. Παρατηρούμε ότι και εδώ παρόμοια συμπεριφορά υπάρχει ανεξάρτητα από την κατανομή που ακολουθεί ο χρόνος άφιξης των εργασιών.

4.3.4 Εξέταση Απόδοσης του Αλγορίθμου MBM για διαφορετικό πλήθος Σημείων Πρόσβασης στον υπό παρατήρηση χώρο

Στην ενότητα αυτή εξετάζεται πως μεταβάλλεται η απόδοση του αλγορίθμου MBM όταν έχουμε διαφορετικό αριθμό σημείων πρόσβασης δικτύου WiFi κατανεμημένα στον υπό παρατήρηση χώρο. Κάθε σημείο πρόσβασης έχει ένα ή δύο κανάλια και μέσω αυτών γίνεται η μεταφόρτωση των εργασιών στην τοπική MYN. Η εκτέλεση μιας εργασίας στην τοπική MYN είναι συνήθως ενεργειακά η πιο συμφέρουσα επιλογή με βάση τις συναρτήσεις κόστους. Όμως, αφού τα σημεία πρόσβασης είναι περιορισμένα δε μπορούν να έχουν όλοι πρόσβαση σε αυτά. Έτσι, οι χρήστες ανταγωνίζονται πολλές φορές για ένα κανάλι, δηλαδή μια μεταφόρτωση στην τοπική MYN. Εξετάσαμε την περίπτωση που υπάρχουν 150 χρήστες στο χώρο και ο χρόνος άφιξης των εργασιών αυτών ακολουθεί κατανομή Poisson. Ο λόγος που λήφθηκε υπόψη μόνο μεγάλος αριθμός χρηστών είναι διότι μόνο σε τέτοιες περιπτώσεις έχει νόημα η αυξομείωση των κινητών σημείων πρόσβασης καθώς αν είχαμε να κάνουμε με λίγους χρήστες ακόμη και μικρός αριθμός σημείων πρόσβασης θα κάλυπτε τις ανάγκες όλων. Για την αξιολόγηση αυτής της παραμέτρου λάβαμε μετρήσεις τόσο για το ελάχιστο ποσοστό υπολειπόμενης ενέργειας(4.18) όσο και για το μέσο χρόνο απόκρισης των εργασιών στην τοπική MYN(4.19).



Σχήμα 4.18: Ελάχιστο Ποσοστό Υπολειπόμενης Ενέργειας με 150 χρήστες και μεταβαλλόμενο αριθμό διαθέσιμων Σημείων Πρόσβασης



Σχήμα 4.19: Μέσος Χρόνος Απόκρισης Τοπικής ΜΥΝ με 150 χρήστες και μεταβαλλόμενο αριθμό διαθέσιμων Σημείων Πρόσβασης

Από το σχήμα 4.18 φαίνεται πως όσο περισσότερα σημεία πρόσβασης έχουμε τόσο καλύτερα αποτελέσματα δίνει ο αλγόριθμος MBM. Αυτό είναι λογικό καθώς με περισσότερα σημεία πρόσβασης μεγαλύτερο ποσοστό χρηστών μπορεί να μεταφορτώνει σε κάθε χρονοσχιμή τις εργασίες του στην τοπική ΜΥΝ καταναλώνοντας έτσι λιγότερη ενέργεια. Αντίθετα, λίγα σημεία πρόσβασης δικτύου WiFi αναγκάζουν μεγάλο μερίδιο των εργασιών να εκτελούνται στο απομακρυσμένο ΥΝ ή στην ίδια τη συσκευή καταλήγωντας έτσι το ελάχιστο ποσοστό υπολειπόμενης ενέργειας να είναι

χαμηλότερο. Βέβαια, αυτό είναι σχετικό καθώς αν είχαμε λίγους χρήστες (π.χ. 50) ακόμη και ένας μικρότερος αριθμός σημείων πρόσβασης (π.χ. 15) θα ήταν αρκετός και περαιτέρω αύξηση αυτών δε θα βελτίωνε τα αποτελέσματα. Αυτό φαίνεται και στον πίνακα 4.4, όπου παρουσιάζεται η κατανομή των εργασιών στις υπολογιστικές υποδομές ανάλογα με τον αριθμό των διαθέσιμων σημείων πρόσβασης που υπάρχουν στον υπό παρατήρηση χώρο. Για το λόγο αυτό και στις προσομοιώσεις επιλέχθηκε μεγάλος αριθμός χρηστών (150).

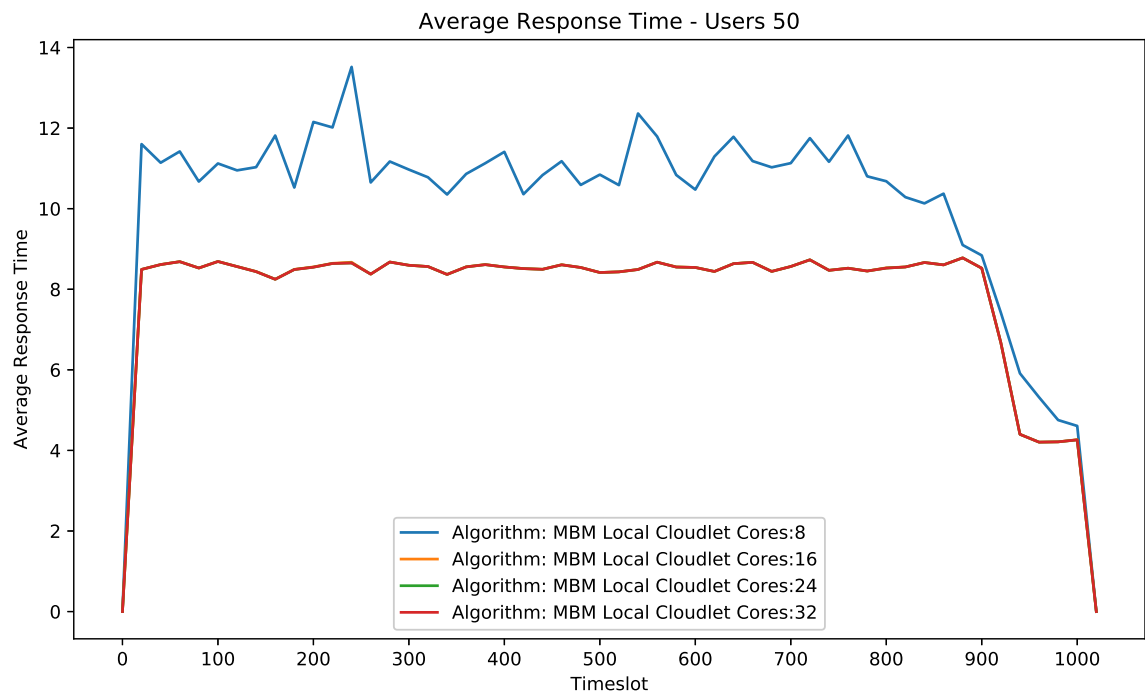
	Απομακρυσμένο ΥΝ	Τοπική ΜΥΝ	Κινητές Συσκευές
15 Σημεία Πρόσβασης	43.81%	34.17%	22.02%
20 Σημεία Πρόσβασης	36.01%	44.60%	19.39%
25 Σημεία Πρόσβασης	27.71%	56.08%	16.21%
30 Σημεία Πρόσβασης	22.83%	62.76%	14.41%
35 Σημεία Πρόσβασης	17.61%	70.26%	12.13%

Πίνακας 4.4: Κατανομή των εργασιών στις υπολογιστικές υποδομές για διαφορετικές τιμές του α

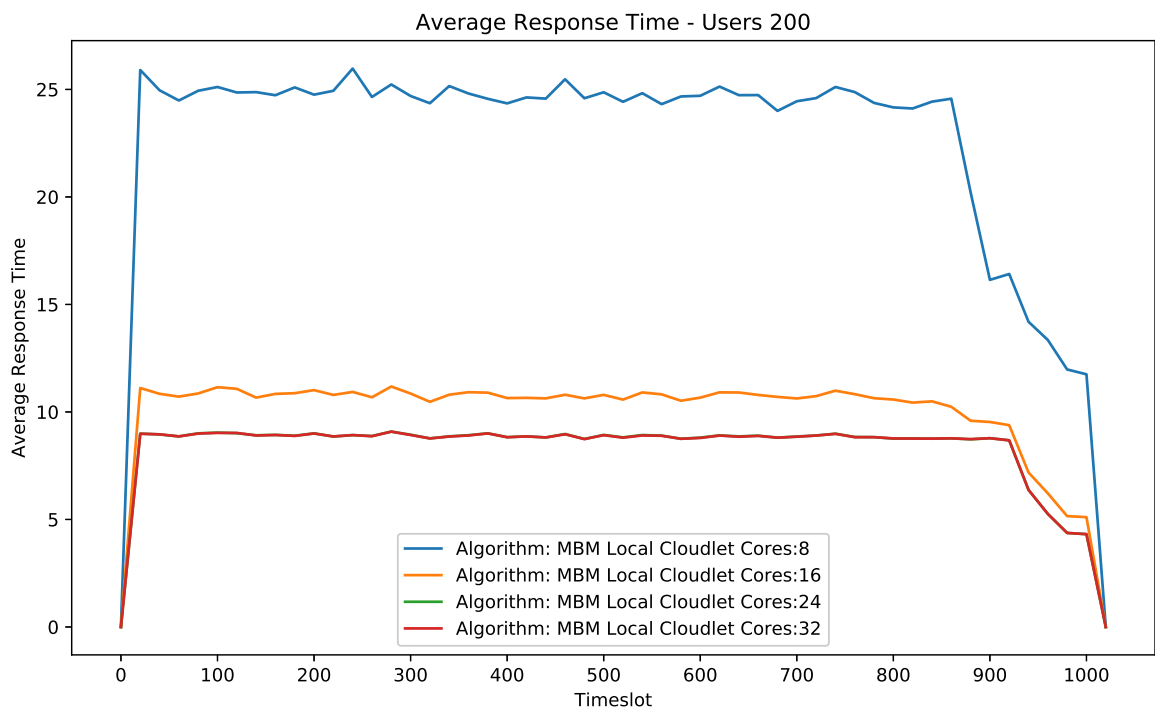
Όσον αφορά το μέσο χρόνο απόκρισης της τοπικής ΜΥΝ που φαίνεται στο σχήμα 4.19, βλέπουμε ότι σε όλες τις περιπτώσεις παραμένει περίπου σταθερός. Αυτό οφείλεται στο γεγονός ότι παρόλο που περισσότερες εργασίες μεταφορτώνονται στην τοπική ΜΥΝ, εκείνη διαθέτει τους απαραίτητους υπολογιστικούς πόρους (32 υπολογιστικούς πυρήνες σε ταχύτητα $3GHz$) ώστε να μπορεί να αντεπεξέλθει στις απαιτήσεις της κάθε μίας άμεσα. Αν είχαμε μεγαλύτερο αριθμό χρηστών ή λιγότερους υπολογιστικούς πόρους ο μέσος χρόνος απόκρισης θα αυξανόταν για πολλά σημεία πρόσβασης καθώς δε θα ήταν δυνατό η τοπική ΜΥΝ να τις εκτελέσει όλες άμεσα και θα παραβίαζε πιθανώς τις απαιτήσεις κάποιων από αυτές αναγκάζοντας το σύστημα να πάρει διαφορετική απόφαση. Αξίζει εδώ να σημειωθεί ότι το μέγεθος των εργασιών, όπως προαναφέρθηκε, κυμαίνεται στα 10000 έως 40000 *MIPS*, δηλαδή απαιτούν κατά μέσο όρο περίπου 8.5 δευτερόλεπτα να εκτελεστούν εάν δεν υποστούν καμία καθυστέρηση λόγω ουράς εργασιών προς εκτέλεση. Το νούμερο αυτό είναι αρκετά κοντά σε αυτό που φαίνεται στη γραφική παράσταση 4.19.

4.3.5 Εξέταση Μέσου Χρόνου Απόκρισης για την τοπική ΜΥΝ με χρήση του Αλγορίθμου MBM για διαφορετικό πλήθος Πυρήνων επεξεργαστή της Τοπικής ΜΥΝ

Σε αυτό το κομμάτι εξετάζεται το πως επηρεάζεται ο μέσος χρόνος απόκρισης της τοπικής ΜΥΝ όταν μεταβάλλεται το πλήθος των επεξεργαστικών πυρήνων που διαθέτει. Εκτελέστηκαν προσομοιώσεις με χρήση του αλγορίθμου MBM για 50 και 200 χρήστες με 25 σημεία πρόσβασης δικτύου WiFi στο χώρο παρατήρησης. Η μεταβλητή που άλλαξε ανά προσομοίωση είναι το πλήθος των επεξεργαστικών πυρήνων της τοπικής ΜΥΝ. Εξετάστηκαν οι περιπτώσεις όπου η τοπική ΜΥΝ διέθετε 8, 16, 24 και 32 πυρήνες. Ο μέσος χρόνος απόκρισης των εργασιών από τις μετρήσεις που λήφθηκαν παρουσιάζονται στα ακόλουθα σχήματα (4.20,4.21).



Σχήμα 4.20: Μέσος Χρόνος Απόκρισης Τοπικής MYN με 50 χρήστες και μεταβαλλόμενο Αριθμό Πυρήνων Επεξεργαστή της Τοπικής MYN



Σχήμα 4.21: Μέσος Χρόνος Απόκρισης Τοπικής MYN με 200 χρήστες και μεταβαλλόμενο Αριθμό Πυρήνων Επεξεργαστή της Τοπικής MYN

Από τις παραπάνω γραφικές παραστάσεις μπορούμε να συμπεράνουμε ότι όσο περισσότερους πυρήνες διαθέτει η τοπική MYN τόσο μικρότερο μέσο χρόνο απόκρισης έχει για τις εργασίες που αναθέτονται σε αυτή. Ωστόσο, στην περίπτωση που έχουμε λίγους χρήστες ακόμη και λίγοι πυρήνες είναι αρκετοί καθώς οι εργασίες που φτάνουν στην τοπική MYN σε κάθε χρονοσχιμή είναι λίγες αντίστοιχα. Γι' αυτό και στο σχήμα 4.20 οι καμπύλες για 16, 24, 32 πυρήνες βρίσκονται στα ίδια ακριβώς επίπεδα. Δεν υπάρχει περαιτέρω βελτίωση από 16 πυρήνες και πάνω καθώς και οι 16 είναι αρκετοί για να καλύψουν άμεσα τις υπολογιστικές ανάγκες των 50 χρηστών. Αντίθετα, όταν έχουμε πολλούς χρήστες βλέπουμε ότι παρατηρείται βελτίωση ακόμη και όταν η τοπική MYN διαθέτει πάνω από 16 υπολογιστικούς πυρήνες. Παρατηρούμε ότι και εδώ (σχήμα 4.21) δεν υπάρχει διαφορά όταν από 24 αυξήσουμε τους πυρήνες σε 32. Επομένως, συμπεραίνουμε ότι στην περίπτωση 200 χρηστών δεν παρατηρείται βελτίωση για περισσότερους από 24 πυρήνες. Η συμπεριφορά αυτή επίσης οφείλεται και στο γεγονός ότι ο αριθμός των σημείων πρόσβασης δικτύου WiFi, μέσω των οποίων μεταφορτώνονται οι εργασίες στην τοπική MYN, είναι περιορισμένος. Δηλαδή, σε κάθε χρονοσχιμή συγκεκριμένο πλήθος εργασιών δύναται να μεταφορτωθεί στην τοπική MYN. Επομένως, με δεδομένο αυτό, πάντα θα υπάρχει ένα όριο πυρήνων που θα είναι αρκετοί για την άμεση εκτέλεση των εργασιών στην τοπική MYN και περαιτέρω αύξηση αυτών δε θα βελτιώνει την απόδοση της τοπικής MYN.

Κεφάλαιο 5

Συμπεράσματα και Προτάσεις για Μελλοντική Έρευνα

Στην ενότητα αυτή παρατίθεται μία σύνοψη των αποτελεσμάτων της παρούσας διπλωματικής εργασίας καθώς και των συμπερασμάτων που προέκυψαν από τη μελέτη και αξιολόγηση του αλγορίθμου λήψης απόφασης για μεταφόρτωση σε αρχιτεκτονική κινητού νέφους δύο επιπέδων (MBM) που παρουσιάστηκε παραπάνω. Στη συνέχεια παρουσιάζονται μερικές ιδέες για μελλοντική επέκταση και βελτίωση του προτεινόμενου αλγορίθμου.

5.1 Σύνοψη και Συμπεράσματα

Στην παρούσα εργασία μελετήθηκε το πρόβλημα λήψης απόφασης για τη μεταφόρτωση μιας εργασίας σε ένα περιβάλλον υπολογισμού κινητού νέφους με αρχιτεκτονική δύο επιπέδων γνωρίζοντας και λαμβάνοντας υπόψη τη θέση των κινητών συσκευών των χρηστών στο χώρο. Για την επίλυση αυτού χρησιμοποιήθηκε ένας αλγόριθμος δρομολόγησης ο οποίος ήταν υπεύθυνος για τη λήψη αυτών των αποφάσεων. Σκοπός του αλγορίθμου αυτού είναι να διαμοιράσει με κατάλληλο τρόπο τις εργασίες των χρηστών στις υπολογιστικές υποδομές (τοπική ΜΥΝ, απομακρυσμένο ΥΝ, συσκευές χρηστών) κατά την περίοδο παρατήρησης ώστε στο τέλος αυτής το ελάχιστο ποσοστό υπολειπόμενης ενέργειας μεταξύ των συσκευών των χρηστών να είναι το μέγιστο δυνατό. Ο αλγόριθμος που χρησιμοποιήθηκε λαμβάνει υπόψη στις αποφάσεις του το ποσοστό υπολειπόμενης ενέργειας κάθε χρήστη προτού πάρει την απόφαση για τη μεταφόρτωση ή μη κάθε εργασίας. Στόχευε ουσιαστικά στο να υπάρχει ισορροπημένη κατανάλωση ενέργειας ανάμεσα στους χρήστες. Αυτό το επιτυγχάνει δημιουργώντας ένα γράφο ανάλογα με τις εργασίες σε κάθε στιγμή δρομολόγησης στον οποίο βρίσκει ένα μέγιστο ταίριασμα (maximum matching) με σκοπό την ελαχιστοποίηση του σταθμισμένου αθροίσματος των επιλεγμένων ακμών στις οποίες έχει ανατεθεί βάρος με μια εκθετική συνάρτηση κόστους.

Αρχικά, για τη βέλτιστη λειτουργία του αλγορίθμου έγιναν πειράματα μεταβάλλοντας τη βασική παράμετρο της συνάρτησης κόστους που χρησιμοποιεί (4.3.1). Αφού βρέθηκε η βέλτιστη τιμή αυτής έγινε σύγκριση του αλγορίθμου MBM με δύο άλλους που χρησιμοποιούν διαφορετικές ευριστικές (4.3.2). Όπως εξηγήθηκε παραπάνω, η ευριστική που χρησιμοποιεί ο αλγόριθμος MBM δίνει καλύτερα αποτελέσματα σε σχέση με αυτούς σε όλες τις περιπτώσεις που εξετάστηκαν.

Ακόμη, εξετάστηκε το πως επηρεάζεται η συμπεριφορά του αλγορίθμου όταν αλλάζουν οι συνθήκες του περιβάλλοντος. Παρατηρήθηκε ότι όσο περισσότεροι χρήστες βρίσκονται στον υπό παρατήρηση χώρο τόσο χαμηλότερο τείνει να γίνεται το ελάχιστο ποσοστό υπολειπόμενης ενέργειας μεταξύ αυτών καθώς υπάρχει μεγαλύτερος ανταγωνισμός για τη χρήση της ενεργειακά "οικονομικότερης" τοπικής ΜΥΝ (4.3.3). Επίσης, είδαμε ότι όσο αυξάνεται το πλήθος των σημείων πρόσβασης στο χώρο τόσο καλύτερη είναι η απόδοση του αλγορίθμου MBM καθώς περισσότεροι χρήστες μπορούν να μεταφορτώσουν εργασίες στην τοπική ΜΥΝ επιτυγχάνοντας έτσι χαμηλότερη ενεργειακή κατανάλωση και τελικά υψηλότερο ελάχιστο ποσοστό υπολειπόμενης ενέργειας (4.3.4).

Τέλος, πέρα από το ελάχιστο ποσοστό υπολειπόμενης ενέργειας, εξετάστηκε και ο μέσος χρόνος απόκρισης των εργασιών που προκύπτει μετά τις μεταφορτώσεις που αποφασίζει ο αλγόριθμος MBM, καθώς είναι ένας παράγοντας που ενδιαφέρει πολύ το χρήστη. Παρατηρήθηκε ότι ακόμη και με πολλές εργασίες κατά την περίοδο παρατήρησης οι αποκρίσεις είναι οι ταχύτερες δυνατές με δεδομένη την υπολογιστική ισχύ της τοπικής ΜΥΝ και του απομακρυσμένου ΥΝ. Είδαμε ακόμα ότι

αυξάνοντας τους υπολογιστικούς πόρους της τοπικής MYN έως ένα σημείο μειώνεται η μέση απόκριση των εργασιών. Από εκείνο τη σημείο και έπειτα όμως δεν παρατηρείται περαιτέρω μείωση στο χρόνο απόκρισης των εργασιών της τοπικής MYN (4.3.5). Επομένως οι υπολογιστικοί πόροι της τοπικής MYN έχει νόημα να είναι ανάλογοι με το πλήθος των εργασιών που προβλέπεται να αιτηθούν εξυπηρέτηση από αυτή.

5.2 Ιδέες για μελλοντική έρευνα

Κάποιες περιπτώσεις που δεν ελέγχθησαν στην παρούσα διπλωματική εργασία όμως είναι αρκετά ενδιαφέρουσες και αξίζουν να διερευνηθούν περαιτέρω είναι οι ακόλουθες:

- Ιδιαίτερο ενδιαφέρον έχει η εξέταση του εν λόγω αλγορίθμου σε περιπτώσεις που στο χώρο παρατήρησης εισέρχονται και εξέρχονται χρήστες με την πάροδο του χρόνου. Αυτό θα άλλαζε αρκετά τα δεδομένα και θα επηρέαζε τα αποτελέσματα καθώς ο αλγόριθμος θα λαμβάνει υπόψη και το ποσοστό ενέργειας των νέων χρηστών αλλάζοντας πιθανόν τις αποφάσεις μεταφόρτωσης.
- Πέρα από το ενεργειακό κόστος θα μπορούσε να εξεταστεί και το χρηματικό κόστος χρήσης των δικτύων 3G/4G και WiFi όταν χρησιμοποιείται αυτός ο αλγόριθμος για να φανεί πως επηρεάζει το χρήστη και από αυτήν την άποψη.
- Ο αλγόριθμος αυτός θα μπορούσε να επεκταθεί και να αποφασίζει και διαχωρισμό των εφαρμογών σε τμήματα και εκτέλεσή τους σε διαφορετικές υπολογιστικές υποδομές ανάλογα με το ενεργειακό συμφέρον του κάθε χρήστη. Έτσι μπορεί να επιτύχει ακόμα καλύτερα αποτελέσματα.

Βιβλιογραφία

- [1] AI, Y., PENG, M., AND ZHANG, K. Edge cloud computing technologies for internet of things: A primer. *Digital Communications and Networks* (2017).
- [2] BECKER, M. J., AND VORONCOVS, A. Hungarian method. https://www-m9.ma.tum.de/graph-algorithms/matchings-hungarian-method/index_en.html, 2015.
- [3] BONOMI, F., MILITO, R., ZHU, J., AND ADDEPALLI, S. Fog computing and its role in the internet of things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing* (New York, NY, USA, 2012), MCC '12, ACM, pp. 13–16.
- [4] CALHEIROS, R. N., RANJAN, R., BELOGLAZOV, A., DE ROSE, C. A. F., AND BUYYA, R. Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exper.* 41, 1 (Jan. 2011), 23–50.
- [5] CHEN, G., KANG, B. T., KANDEMIR, M., VIJAYKRISHNAN, N., IRWIN, M. J., AND CHANDRAMOULI, R. Studying energy trade offs in offloading computation/compilation in java-enabled mobile devices. *IEEE Transactions on Parallel and Distributed Systems* 15, 9 (Sept 2004), 795–809.
- [6] CHUN, B.-G., AND MANIATIS, P. Augmented smartphone applications through clone cloud execution. In *Proceedings of the 12th Conference on Hot Topics in Operating Systems* (Berkeley, CA, USA, 2009), HotOS'09, USENIX Association, pp. 8–8.
- [7] CORMEN, T. H., STEIN, C., RIVEST, R. L., AND LEISERSON, C. E. *Introduction to Algorithms*, 2nd ed. McGraw-Hill Higher Education, 2001.
- [8] CUERVO, E., BALASUBRAMANIAN, A., CHO, D.-K., WOLMAN, A., SAROIU, S., CHANDRA, R., AND BAHL, P. Maui: Making smartphones last longer with code offload. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2010), MobiSys '10, ACM, pp. 49–62.
- [9] DECHOUNIOTIS, D., LEONTIOU, N., ATHANASOPOULOS, N., CHRISTAKIDIS, A., AND DENAZIS, S. A control-theoretic approach towards joint admission control and resource allocation of cloud computing services. *International Journal of Network Management* 25, 3 (3 2015), 159–180.
- [10] DINH, H. T., LEE, C., NIYATO, D., AND WANG, P. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing* 13, 18 (2013), 1587–1611.
- [11] FERNANDO, N., LOKE, S. W., AND RAHAYU, W. Mobile cloud computing. *Future Gener. Comput. Syst.* 29, 1 (Jan. 2013), 84–106.
- [12] FILHO, M. C. S., OLIVEIRA, R. L., MONTEIRO, C. C., INÁCIO, P. R. M., AND FREIRE, M. M. 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)* (May 2017), pp. 400–406.

- [13] GU, X., NAHRSTEDT, K., MESSER, A., GREENBERG, I., AND MILOJICIC, D. Adaptive offloading inference for delivering applications in pervasive computing environments. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003)*. (March 2003), pp. 107–114.
- [14] GURUN, S., KRINTZ, C., AND WOLSKI, R. Nwslite: A light-weight prediction utility for mobile devices. In *Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services* (New York, NY, USA, 2004), MobiSys '04, ACM, pp. 2–11.
- [15] HENDRICKSON, B., AND LELAND, R. A multilevel algorithm for partitioning graphs. In *Proceedings of the 1995 ACM/IEEE Conference on Supercomputing* (New York, NY, USA, 1995), Supercomputing '95, ACM.
- [16] HOANG, D. B., AND CHEN, L. Mobile cloud for assistive healthcare (mocash). In *2010 IEEE Asia-Pacific Services Computing Conference* (Dec 2010), pp. 325–332.
- [17] HONG, Y. J., KUMAR, K., AND LU, Y. H. Energy efficient content-based image retrieval for mobile systems. In *2009 IEEE International Symposium on Circuits and Systems* (May 2009), pp. 1673–1676.
- [18] HUA CHU, H., SONG, H., WONG, C., KURAKAKE, S., AND KATAGIRI, M. Roam, a seamless application framework. *Journal of Systems and Software* 69, 3 (2004), 209 – 226. Ubiquitous Computing.
- [19] HUERTA-CANEPA, G., AND LEE, D. An adaptable application offloading scheme based on application behavior. In *22nd International Conference on Advanced Information Networking and Applications - Workshops (aina workshops 2008)* (March 2008), pp. 387–392.
- [20] KUHN, H. W. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1-2 (1955), 83–97.
- [21] KUMAR, K., LIU, J., LU, Y.-H., AND BHARGAVA, B. A survey of computation offloading for mobile systems. *Mob. Netw. Appl.* 18, 1 (Feb. 2013), 129–140.
- [22] KUMAR, K., AND LU, Y. H. Cloud computing for mobile users: Can offloading computation save energy? *Computer* 43, 4 (April 2010), 51–56.
- [23] LEONTIOU, N., DECHOUNIOTIS, D., ATHANASOPOULOS, N., AND DENAZIS, S. On load balancing and resource allocation in cloud services. In *22nd Mediterranean Conference on Control and Automation* (June 2014), pp. 773–778.
- [24] MAHMUD, R., AND BUYYA, R. Fog computing: A taxonomy, survey and future directions. *CoRR abs/1611.05539* (2016).
- [25] NIMMAGADDA, Y., KUMAR, K., LU, Y. H., AND LEE, C. S. G. Real-time moving object recognition and tracking using computation offloading. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (Oct 2010), pp. 2449–2455.
- [26] OU, S., YANG, K., AND LIOTTA, A. An adaptive multi-constraint partitioning algorithm for offloading in pervasive systems. In *Fourth Annual IEEE International Conference on Pervasive Computing and Communications (PERCOM'06)* (March 2006), pp. 10 pp.–125.
- [27] OU, S., YANG, K., LIOTTA, A., AND HU, L. Performance analysis of offloading systems in mobile wireless environments. In *2007 IEEE International Conference on Communications* (June 2007), pp. 1821–1826.

- [28] RAHIMI, M. R., REN, J., LIU, C. H., VASILAKOS, A. V., AND VENKATASUBRAMANIAN, N. Mobile cloud computing: A survey, state of art and future directions. *Mobile Networks and Applications* 19, 2 (Apr 2014), 133–143.
- [29] RAHIMI, M. R., VENKATASUBRAMANIAN, N., AND VASILAKOS, A. V. Music: Mobility-aware optimal service allocation in mobile cloud computing. In *2013 IEEE Sixth International Conference on Cloud Computing* (June 2013), pp. 75–82.
- [30] RIM, H., KIM, S., KIM, Y., AND HAN, H. Transparent method offloading for slim execution. In *2006 1st International Symposium on Wireless Pervasive Computing* (Jan 2006), pp. 1–6.
- [31] RONG, P., AND PEDRAM, M. Extending the lifetime of a network of battery-powered mobile devices by remote processing: a markovian decision-based approach. In *Proceedings 2003. Design Automation Conference (IEEE Cat. No.03CH37451)* (June 2003), pp. 906–911.
- [32] ROUGHGARDEN, T. Hungarian algorithm. <http://theory.stanford.edu/~tim/w16/l/l5.pdf>, Jan. 2016.
- [33] SATYANARAYANAN, M., BAHL, P., CACERES, R., AND DAVIES, N. The case for vm-based cloudlets in mobile computing. *IEEE Pervasive Computing* 8, 4 (Oct. 2009), 14–23.
- [34] SATYANARAYANAN, M., SCHUSTER, R., EBLING, M., FETTWEIS, G., FLINCK, H., JOSHI, K., AND SABNANI, K. An open ecosystem for mobile-cloud convergence. *IEEE Communications Magazine* 53, 3 (March 2015), 63–70.
- [35] TILEVICH, E., AND SMARAGDAKIS, Y. *J-Orchestra: Automatic Java Application Partitioning*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 178–204.
- [36] WANG, C., AND LI, Z. Parametric analysis for adaptive computation offloading. *SIGPLAN Not.* 39, 6 (June 2004), 119–130.
- [37] WOLSKI, R., GURUN, S., KRINTZ, C., AND NURMI, D. Using bandwidth data to make computation offloading decisions. In *2008 IEEE International Symposium on Parallel and Distributed Processing* (April 2008), pp. 1–8.
- [38] XIA, Q., LIANG, W., XU, Z., AND ZHOU, B. Online algorithms for location-aware task offloading in two-tiered mobile cloud environments. In *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing* (Dec 2014), pp. 109–116.
- [39] XIAN, C., LU, Y.-H., AND LI, Z. Adaptive computation offloading for energy conservation on battery-powered systems. In *2007 International Conference on Parallel and Distributed Systems* (Dec 2007), vol. 2, pp. 1–8.
- [40] XIAN, C., LU, Y.-H., AND LI, Z. Adaptive computation offloading for energy conservation on battery-powered systems. In *Proceedings of the 13th International Conference on Parallel and Distributed Systems - Volume 01* (Washington, DC, USA, 2007), ICPADS '07, IEEE Computer Society, pp. 1–8.
- [41] YANG, K., OU, S., AND CHEN, H. H. On effective offloading services for resource-constrained mobile devices running heavier mobile internet applications. *IEEE Communications Magazine* 46, 1 (January 2008), 56–63.

Παράρτημα Α

Ούγγρικος Αλγόριθμος (Hungarian algorithm)

Ο ούγγρικος αλγόριθμος [2, 32] επιλύει το πρόβλημα εύρεσης σταθμισμένου ταιριάσματος ελαχίστου κόστους (minimum cost weighted matching) σε ένα διμερές γράφο (bipartite graph) $G = (V, E)$. Ο αλγόριθμος αυτό αναπτύχθηκε από τον Harold W. Kuhn ο οποίος βασίστηκε σε ιδέες δύο ούγγρων μαθηματικών (Dénes Kőnig και Jenő Egerváry). Αργότερα ο αλγόριθμος βελτιώθηκε από τον James Munkres.

A.1 Βασικές Έννοιες

Το σύνολο των κόμβων του διμερούς γραφήματος $G = (V, E)$ διαχωρίζεται σε δύο επί μέρους σύνολα, έστω X, Y (σχήμα A.1a). Ισχύει ότι $X \cup Y = V$ και $X \cap Y = \emptyset$.

A.1.1 Ετικέτες (Labels)

Χρησιμοποιείται μια βοηθητική συνάρτηση ή ετικέτα l , η οποία αντιστοιχεί έναν πραγματικό αριθμό σε κάθε κόμβο του γράφου G έτσι ώστε το άθροισμα των ετικετών δύο κόμβων να είναι μεγαλύτερο ή ίσο του βάρους της ακμής w που ενώνει αυτούς τους δύο κόμβους (σχήμα A.1b).

$$l(x) + l(y) \geq w(x, y) \quad (\text{A.1})$$

A.1.2 Γράφος Ισότητας (Equality Graph)

Χρησιμοποιώντας αυτές τις ετικέτες μπορούμε να κατασκευάσουμε έναν γράφο ισότητας $G_l = (V, E_l)$ ο οποίος περιέχει όλους τους κόμβους του αρχικού γράφου. Όμως, το σύνολο των ακμών του γράφου ισότητας E_l περιέχει μόνο εκείνες τις ακμές που έχουν ακριβώς το βάρος όσο το άθροισμα των ετικετών των κόμβων που συνενώνουν (σχήμα A.1c).

$$E_l = \{(x, y) \in E \mid l(x) + l(y) = w(x, y)\} \quad (\text{A.2})$$

A.1.3 Ταιριασμένος και Ελεύθερος Κόμβος (Match and Free Vertex)

Ένας κόμβος θεωρείται ως ήδη ταιριασμένος, αν και μόνο αν έχει μια ακμή η οποία συμπεριλαμβάνεται στο υπάρχον έως εκείνη τη στιγμή ταιρίασμα. Διαφορετικά αποκαλείται ελεύθερος (σχήμα A.1d).

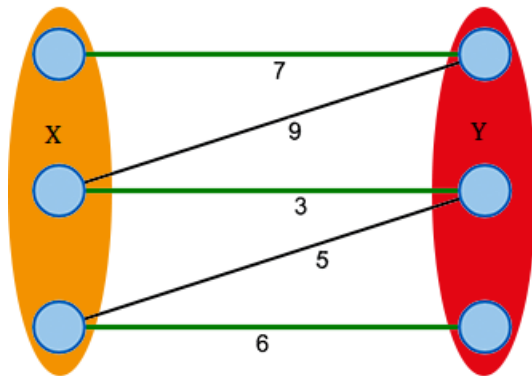
A.1.4 Εναλλάξ Μονπάτια (Alternating Paths)

Ένα μονοπάτι λαμβάνει τον χαρακτηρισμό ως εναλλάξ εάν αποτελείται από μία ακολουθία ακμών όπου ανά δύο συνεχόμενες η μία ανήκει στο ταιρίασμα M που έχει προκύψει ως εκείνη τη στιγμή ενώ η άλλη όχι (σχήμα A.1e).

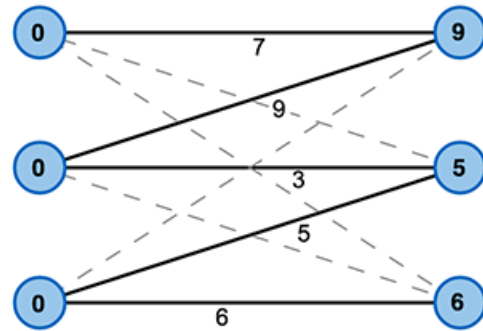
A.1.5 Επαυξημένα Μονοπάτια (Augmenting Paths)

Ένα εναλλάξ μονοπάτι θεωρείται επαυξημένο εάν τόσο ο αρχικός του κόμβος όσο και ο τελικός του είναι ελεύθεροι (σχήμα A.1f).

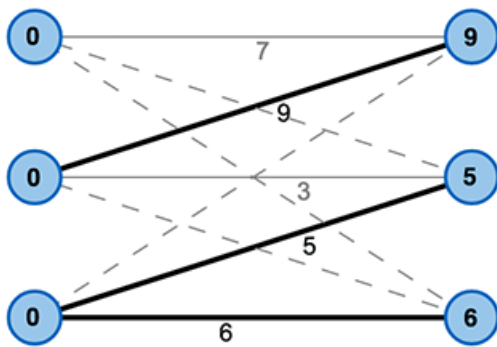
Σχήμα A.1: Βασικές Έννοιες



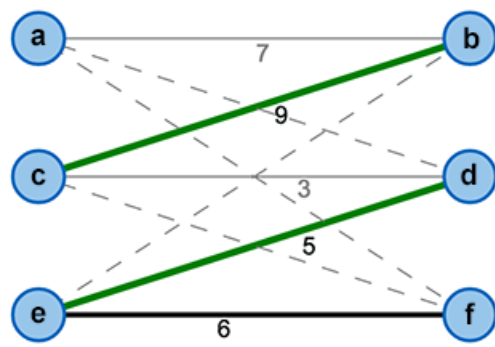
(a) Βέλτιστο Ταίριασμα



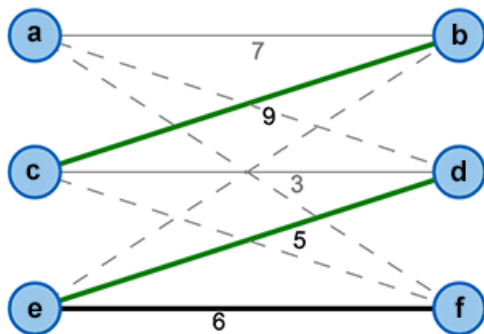
(b) Ετικέτες Κόμβων



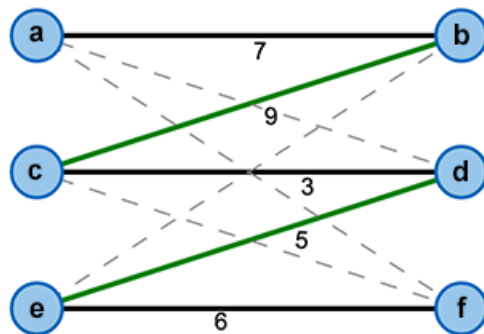
(c) Γράφος Ισότητας



(d) Ελεύθεροι (a,f) και Ταίριασμένοι (b,c,d,e) Κόμβοι



(e) Εναλλάξ Μονοπάτι (f,e,d)



(f) Επαυξημένο Μονοπάτι (f,e,d,c,b,a)

A.2 Εκτέλεση του Αλγορίθμου

Το θεώρημα Kuhn-Munkres αναφέρει ότι κάθε τέλει ταίριασμα στο γράφο ισότητας αποτελεί μέγιστο ταίριασμα στον αρχικό γράφο. Έτσι, το πρόβλημα εύρεσης ενός βέλτιστου ταίριασματος ανάγεται στην εύρεση ενός τέλει ταίριασματος στο γράφο ισότητας.

Ο αλγόριθμος χρησιμοποιεί μια επαναληπτική διαδικασία για την εύρεση τέλει ταίριασματος στον γράφο ισότητας. Εκκινεί από το κενό ταίριασμα (empty matching) και προσπαθεί σε κάθε

βήμα να το βελτιώνει έως ότου φτάσει στο τελικό αποτέλεσμα.

Αρχικά ο αλγόριθμος δίνει μία ετικέτα, έναν πραγματικό αριθμό δηλαδή, σε κάθε κόμβο. Σαν ετικέτα κάθε κόμβου του συνόλου X εκχωρείται το βάρος της βαρύτερης ακμής που προσπίπτει σε αυτόν. Οι κόμβοι του συνόλου Y λαμβάνουν σαν ετικέτα την τιμή 0. Χρησιμοποιώντας αυτές τις ετικέτες σαν εκκίνηση, ο αλγόριθμος κατασκευάζει το γράφο ισότητας.

Προκειμένου να βελτιωθεί το ήδη υπάρχον ταίριασμα (το αρχικό ταίριασμα είναι αρχικοποιημένο στο κενό σύνολο), ο αλγόριθμος επιλέγει έναν κόμβο του συνόλου X ο οποίος δεν έχει κανένα "ταίρι" (matching partner) ως τώρα. Έχοντας ως εκκίνηση εκείνον τον κόμβο, αναζητείται ένα επαυξημένο μονοπάτι. Για να βρεθεί αυτό κατασκευάζεται ένα εναλλάξ μονοπάτι βήμα βήμα έως ότου γίνει επαυξημένο ή εξαντλήσει όλες τις ακμές με τις οποίες εκείνο μπορεί να προχωρήσει. Στη συνέχεια οι κόμβοι που επισκέφθηκαν σε αυτή τη διαδικασία διαχωρίζονται σε δύο σύνολα:

- Το σύνολο S περιλαμβάνει τους κόμβους που επισκέφθηκαν και ανήκουν στο σύνολο X
- Το σύνολο T περιλαμβάνει τους κόμβους που επισκέφθηκαν και ανήκουν στο σύνολο Y

Όμως, δεν είναι πάντα δυνατό να βρεθεί επαυξημένο μονοπάτι. Σε μια τέτοια περίπτωση ο αλγόριθμος πρέπει να ενημερώσει τις ετικέτες που έχουν δοθεί στους κόμβους του γράφου. Η ενημέρωση αυτή θα πρέπει να γίνει με τέτοιο τρόπο ώστε το ταίριασμα που έχει βρεθεί μέχρι εκείνη τη στιγμή (current matching) να διατηρηθεί στο γράφο ισότητας. Ακόμη, στο γράφο ισότητας θα πρέπει να προστεθούν επιπλέον ακμές έτσι ώστε μετά τη διαδικασία αυτή να γίνει δυνατή η εύρεση επαυξημένου μονοπατιού. Αυτό μπορεί να επιτευχθεί υπολογίζοντας μια τιμή Δ . Η τιμή αυτή προκύπτει λαμβάνοντας υπόψη όλα τα ζευγάρια των κόμβων που ήδη έχουμε επισκεφθεί $s \in S$ και αυτών που δεν έχουμε επισκεφθεί $y \in Y \setminus T$ και υπολογίζοντας το ελάχιστο από τα αθροίσματα των ετικετών των κόμβων όπου ανά δύο ανήκουν σε διαφορετικό σύνολο μειωμένο κατά το βάρος της μεταξύ τους ακμής. Δηλαδή:

$$\Delta = \min_{s \in S \wedge y \in Y \setminus T} \{l(s) + l(y) - w(s, y)\} \quad (\text{A.3})$$

Οι ετικέτες όλων των κόμβων μπορούν τώρα να τροποποιηθούν χρησιμοποιώντας το Δ πριν δημιουργηθεί ο νέος γράφος ισότητας προκειμένου να γίνει νέα αναζήτηση για επαυξημένα μονοπάτια. Οι νέες ετικέτες ορίζονται ως εξής:

$$l'(u) = \begin{cases} l(u) - \Delta & , \text{αν } u \in S \\ l(u) + \Delta & , \text{αν } u \in T \\ l(u) & , \text{διαφορετικά} \end{cases} \quad (\text{A.4})$$

Εάν μετά από αυτή την ενημέρωση των ετικετών ο αλγόριθμος βρει επαυξημένο μονοπάτι ενημερώνει το ταίριασμα που έχει βρεθεί ως εκείνη τη στιγμή προσθέτοντας σε αυτό τις ακμές του μονοπατιού που δεν περιέχει και αφαιρώντας όσες ήταν ήδη μέσα στο ταίριασμα. Αυτό το βήμα επαναλαμβάνεται έως ότου κάθε κόμβος να αποκτήσει ένα "ταίρι". Έτσι, το ταίριασμα καταλήγει να είναι τέλειο.

Σημείωση: Η μέθοδος αυτή όπως περιγράφηκε βρίσκει μέγιστα ταιριάσματα, δηλαδή ταιριάσματα όπου το άθροισμα των βαρών των ακμών είναι όσο το δυνατό μεγαλύτερο. Για την εύρεση ελαχίστων ταιριασμάτων το μόνο που πρέπει να γίνει πριν την εκτέλεση του αλγορίθμου είναι να μεταβληθούν τα βάρη του γράφου αλλάζοντας τους πρόσημο, δηλαδή όπου έχουμε $w(x, y)$ να γίνει $-w(x, y)$.

Ακόμη πρέπει να αναφερθεί ότι ο διμερής γράφος θα πρέπει να είναι πλήρης, δηλαδή το πλήθος των κόμβων στο σύνολο X να είναι ίσο με το αντίστοιχο στο σύνολο Y και να υπάρχει τουλάχιστον μία ακμή μεταξύ δύο κόμβων που βρίσκονται σε διαφορετικά τμήματα του αρχικού συνόλου κόμβων V . Αν δεν καλύπτονται τα παραπάνω εισάγονται κόμβοι και ακμές στο γράφο με ετικέτες και βάρη αντίστοιχα:

- 0 εάν αναζητούμε μέγιστο ταίριασμα

- με αρκετά μεγάλη τιμή για το εκάστοτε πρόβλημα εάν αναζητούμε ελάχιστο ταίριασμα ώστε ο γράφος να γίνει πλήρης. Αν κάποιος κόμβος ή κάποια ακμή από αυτές που εισάγαμε επιλεγεί στο τελικό μας αποτέλεσμα τον/την αφαιρούμε και εκτελούμε εκ νέου τον αλγόριθμο.