



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών  
Υπολογιστών

**Μελέτη και αξιολόγηση αλγορίθμων δρομολόγησης  
υπολογιστικών διεργασιών από UE σε MEC servers.**

Διπλωματική Εργασία  
Κουμουτζέλης Στυλιανός

Επιβλέπων: Παναγιώτης Κωπτής  
Καθηγητής Ε.Μ.Π.

Αθήνα, Σεπτέμβριος 2021





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών  
Υπολογιστών

**Μελέτη και αξιολόγηση βιβλιογραφίας που αφορά  
αλγορίθμους δρομολόγησης υπολογιστικών διεργασιών  
από UE σε MEC servers.**

Διπλωματική Εργασία  
Κουμουτζέλης Στυλιανός

Επιβλέπων: Παναγιώτης Κωπτής  
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 04/10/2021

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Αθήνα, Σεπτέμβριος 2021



---

Κουμουτζέλης Στυλιανός

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

Copyright © Κουμουτζέλης Στυλιανός 2021

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση σε αρχείο πληροφοριών, διανομή, αναπαραγωγή, μετάφραση ή μετάδοση της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό, υπό οποιαδήποτε μορφή και με οποιοδήποτε μέσο επικοινωνίας, ηλεκτρονικό ή μηχανικό, χωρίς την προηγούμενη έγγραφη άδεια του συγγραφέως. Επιτρέπεται η αναπαραγωγή, 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 στη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Η έγκριση της διπλωματικής εργασίας από τη Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσοβίου Πολυτεχνείου δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέως (Ν. 5343/1932, Άρθρο 202).

## ***Ευχαριστίες***

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής μου εργασίας κ. Παναγιώτη Κωπτή για την καθοδήγηση, τις συμβουλές και το ενδιαφέρον που μου προσέφερε καθ' όλη τη διάρκεια της εκπόνησής της. Επιπλέον, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στην οικογένειά μου, και σε όσα μου προσέφερε σε όλα τα χρόνια φοίτησης μου στο Πολυτεχνείο. Τέλος, ευχαριστώ τα κοντινά μου πρόσωπα για τις όμορφες στιγμές που περάσαμε μαζί ως φοιτητές.



*Στους γονείς μου Λευτέρη και Ελένη,  
και στον αδερφό μου Απόστολο*







ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Μελέτη και αξιολόγηση αλγορίθμων δρομολόγησης υπολογιστικών διεργασιών από UE σε MEC servers.**

Κουμουτζέλης Στυλιανός  
Επιβλέπων: Καθηγητής Παναγιώτης Κωπτής  
Σεπτέμβριος 2021

**ΠΕΡΙΛΗΨΗ**

Η παρούσα διπλωματική εργασία αφορά τη μελέτη της επικοινωνίας μεταξύ χρηστών κινητής τηλεφωνίας και MEC servers. Στο πρώτο κεφάλαιο, γίνεται μια εισαγωγή στην τεχνολογία του Mobile Edge Computing. Μέσω της μελέτης βιβλιογραφίας σχετικής με το αντικείμενο, παρουσιάζονται αρχικά οι λόγοι της μετάβασης από το centralized μοντέλο του Cloud Computing σε αυτό των Edge Networks. Στη συνέχεια, περιγράφεται η αρχιτεκτονική του framework πάνω στο οποίο βασίζεται το MEC, καθώς και τα πλεονεκτήματα που προσφέρει στον χρήστη η υπηρεσία αυτή. Τέλος, αναφέρονται εφαρμογές και τεχνολογίες που επωφελούνται από τα χαρακτηριστικά του MEC.

Το δεύτερο κεφάλαιο ασχολείται με την εφαρμογή της τεχνολογίας MEC αρχικά σε 4G δίκτυα. Στη συνέχεια αφού γίνεται μια σύντομη ανασκόπηση της αρχιτεκτονικής του 5G, περιγράφεται η ενσωμάτωση του MEC στο framework του 5G, καθώς και τα πλεονεκτήματα μιας τέτοιας διαδικασίας. Στο τρίτο κεφάλαιο, αναφέρονται αρχικά μαθηματικά μοντέλα που χρησιμοποιούνται ως εργαλεία προσομοίωσης αφίξεων. Στη συνέχεια, γίνεται μια πρώτη αναφορά και εισαγωγή στα 2 papers των οποίων τις προσεγγίσεις θα αξιολογήσουμε (Paper 1: "Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices", Paper 2: "Comparison of scheduling algorithms for multiple mobile computing edge clouds"). Στην πρώτη περίπτωση, θα υλοποιηθεί σε MATLAB μια εκδοχή του αλγόριθμου που προτείνεται, ενώ στη δεύτερη τα πειραματικά δεδομένα θα ληφθούν αυτούσια από τη βιβλιογραφία.

Στο καταληκτικό κεφάλαιο της εργασίας, παρουσιάζονται εκτενώς οι προσεγγίσεις των δύο papers, και αξιολογούνται τα πειραματικά τους αποτελέσματα. Τέλος, επιχειρείται μια σύγκριση των δύο, καθώς και δικές μας προτάσεις σχετικά με τη χρήση της κάθε προσέγγισης σε πρακτικές εφαρμογές του MEC (λ.χ. Augmented Reality, Mobile Gaming, autonomous driving).

**Λέξεις κλειδιά:** Mobile Edge Computing, 5G, LTE, αλγόριθμος δρομολόγησης, Augmented Reality





NATIONAL TECHNICAL UNIVERSITY OF ATHENS  
SCHOOL OF ELECTRICAL AND COMPUTING ENGINEERING

DIPLOMA THESIS

**Study and evaluation of scheduling algorithms for routing computational processes from EU to MEC servers.**

Koumoutzelis Stylianos

Supervisor: Professor Panagiotis Kottis

September 2021

**ABSTRACT**

This thesis concerns the study of communication between mobile telephony users and MEC servers. In the first chapter, an introduction is made to Mobile Edge Computing technology. Through the study of bibliography tackling this particular subject, the reasons for the transition from the centralized model of Cloud Computing to that of Edge Networks are first presented. Next, the framework architecture on which MEC is based is described, as well as the advantages that this service offers to users. Finally, applications and technologies that benefit from the features of MEC are mentioned.

The second chapter deals with the application of MEC technology, initially in 4G networks. Then, after a brief review of the 5G architecture, the integration of MEC into the 5G framework is described, as well as the advantages of such a process. In the third chapter, mathematical models used as arrivals simulation tools are reported. Then, a first reference and introduction are made to the two papers whose approaches we will evaluate (Paper 1: "Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices", Paper 2: "Comparison of scheduling algorithms for multiple mobile computing edge clouds"). In the first case, a version of the proposed algorithm will be implemented in MATLAB, while in the second case the experimental data will be taken as is from bibliography.

In the final chapter of the paper, the approaches of the 2 papers are presented in detail, and their experimental results are evaluated. Finally, a comparison of the two is attempted, as well as our own proposals regarding the use of each approach in practical applications of MEC (e.g. Augmented Reality, Mobile Gaming, autonomous driving)

**Key words:** Mobile Edge Computing, 5G, LTE, scheduling algorithm, Augmented Reality



# ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	10
ABSTRACT.....	12
Κατάλογος Εικόνων.....	17
Κατάλογος Διαγραμμάτων.....	18
Κατάλογος Πινάκων.....	19
ΚΕΦΑΛΑΙΟ 1 <sup>ο</sup> .....	20
<b>MOBILE EDGE COMPUTING</b> .....	20
<b>1.1 Εισαγωγή στο MEC</b> .....	20
<b>1.2 Αρχιτεκτονική MEC</b> .....	20
<b>1.3 Πλεονεκτήματα MEC</b> .....	25
<b>1.3.1 Χαμηλή Καθυστέρηση</b> .....	25
<b>1.3.2 Εξοικονόμηση Ενέργειας</b> .....	25
<b>1.3.3 Επίγνωση Γενικού Πλαισίου</b> .....	26
<b>1.3.4 Ιδιωτικότητα/Ασφάλεια</b> .....	26
<b>1.4 Εφαρμογές MEC</b> .....	26
<b>1.4.1 Επαυξημένη πραγματικότητα</b> .....	26
<b>1.4.2 Έξυπνη επιτάχυνση παροχής βίντεο</b> .....	27
<b>1.4.3 Διασυνδεδεμένα Έξυπνα Αυτοκίνητα</b> .....	28
<b>1.4.4 Internet of Things</b> .....	28
ΚΕΦΑΛΑΙΟ 2 <sup>ο</sup> .....	29
<b>ΕΦΑΡΜΟΓΗ ΤΟΥ MEC ΣΕ 4G ΚΑΙ 5G ΔΙΚΤΥΑ</b> .....	29
<b>2.1 Τεχνολογία LTE</b> .....	29
<b>2.1.1 E-UTRAN</b> .....	30
<b>2.1.2 Evolved Packet Core</b> .....	31
<b>2.2 Mobile Edge Computing σε LTE δίκτυα</b> .....	32
<b>2.2.1. Bump in the wire</b> .....	32
<b>2.2.2. Κατανεμημένος EPC</b> .....	33
<b>2.2.3. Κατανεμημένα S-GW/P-GW</b> .....	34
<b>2.2.4. Κατανεμημένη S-GW με τοπικό breakout</b> .....	34
<b>2.3 Εφαρμογή MEC σε 5G δίκτυα</b> .....	35
<b>2.3.1 Εισαγωγή στα δίκτυα 5<sup>ης</sup> Γενιάς</b> .....	35
<b>2.3.2 Αρχιτεκτονική δικτύων 5<sup>ης</sup> Γενιάς</b> .....	37
<b>2.3.3. Αρχιτεκτονική MEC στα δίκτυα 5<sup>ης</sup> Γενιάς</b> .....	41
ΚΕΦΑΛΑΙΟ 3 <sup>ο</sup> .....	43

<b>ΔΡΟΜΟΛΟΓΗΣΗ MOBILE TASKS ΣΕ MEC ΕΞΥΠΗΡΕΤΗΤΕΣ</b> .....	43
<b>3.1 Εισαγωγή</b> .....	43
<b>3.2 Μοντέλα προσομοίωσης αφίξεων</b> .....	43
<b>3.2.1 Ουρά <math>M/M/1</math></b> .....	43
<b>3.2.2 Διαδικασία Bernoulli</b> .....	45
<b>3.3 Εισαγωγή στην πρώτη προσέγγιση: "Comparison of scheduling algorithms for multiple mobile computing edge clouds"</b> .....	46
<b>3.4 Εισαγωγή στην δεύτερη προσέγγιση: "Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices"</b> .....	46
<b>ΚΕΦΑΛΑΙΟ 4<sup>ο</sup></b> .....	49
<b>ΑΛΓΟΡΙΘΜΟΙ ΔΡΟΜΟΛΟΓΗΣΗΣ MOBILE TASKS ΓΙΑ ΠΟΛΛΑΠΛΑ MEC CLOUDS</b> .....	49
<b>4.1 Εισαγωγή</b> .....	49
<b>4.2 Παράμετροι του προβλήματος</b> .....	49
<b>4.2.1 Τοπολογικές συνθήκες</b> .....	49
<b>4.2.2 Κίνηση χρηστών</b> .....	52
<b>4.2.3 Συμβολισμοί μεγεθών</b> .....	52
<b>4.3 Αλγόριθμοι δρομολόγησης</b> .....	53
<b>4.3.1 Αλγόριθμος 1: Random Rule</b> .....	54
<b>4.3.2 Αλγόριθμος 2: Index Rule</b> .....	54
<b>4.3.3 Αλγόριθμος 3: SOL Rule</b> .....	55
<b>4.3.4 Αλγόριθμος 4: LFS Rule</b> .....	55
<b>4.4 Εφαρμογή Αλγορίθμων και παρουσίαση αποτελεσμάτων</b> .....	56
<b>4.4.1 Σενάριο 1</b> .....	56
<b>4.4.2 Σενάριο 2</b> .....	59
<b>4.5 Σχολιασμός Αποτελεσμάτων</b> .....	61
<b>4.5.1 Σενάριο 1</b> .....	61
<b>4.5.2 Σενάριο 2</b> .....	62
<b>ΚΕΦΑΛΑΙΟ 5<sup>ο</sup></b> .....	63
<b>ΑΛΓΟΡΙΘΜΟΣ LODCO</b> .....	63
<b>5.1 Εισαγωγή</b> .....	63
<b>5.2 Μοντέλα Συστήματος</b> .....	63
<b>5.2.1 Computational Model</b> .....	63
<b>5.2.2 Energy Harvesting Model</b> .....	65
<b>5.3 Διατύπωση Προβλήματος</b> .....	66
<b>5.3.1 Ελαχιστοποίηση κόστους εκτέλεσης</b> .....	66
<b>5.3.2 Προβλήματα αποφάσεων Markov</b> .....	67
<b>5.3.3 Χαρακτηριστικά αλγορίθμου</b> .....	67

5.3.4 Βήματα αλγορίθμου.....	68
5.4 Υλοποίηση αλγορίθμου στο MATLAB .....	68
5.4.1 Optimal Energy Harvesting .....	69
5.4.2 Optimal Computation Offloading .....	69
5.4.3 Κώδικας MATLAB.....	71
5.5 Αποτελέσματα προσομοίωσης .....	76
5.5.1 Εξέλιξη επιπέδου μπαταρίας .....	76
5.5.2 Μέσο συνολικό κόστος εκτέλεσης.....	77
5.5.3 Μέση επιλογή μεθόδου εκτέλεσης .....	78
ΚΕΦΑΛΑΙΟ 6° .....	81
ΣΥΜΠΕΡΑΣΜΑΤΑ .....	81
6.1 Γενικό σχόλιο .....	81
6.2 Σύγκριση των δύο προσεγγίσεων .....	81
6.2.1 Μοντελοποίηση UEs και πλατφόρμας MEC .....	82
6.2.2 Energy Harvesting και προσαρμοστικότητα μοντέλων .....	82
6.2.3 Μέτρα αξιολόγησης .....	82
6.3 Τελικά συμπεράσματα και προτάσεις.....	83
Βιβλιογραφία .....	84



# Κατάλογος Εικόνων

Εικόνα 1.1: MEC System .....	21
Εικόνα 1.2: MEC Framework [2] .....	22
Εικόνα 1.3: MEC Architecture Reference Model [2] .....	23
Εικόνα 1.4: MEC in Augmented Reality Scenario [4] .....	27
Εικόνα 1.5: MEC in Interconnected Cars Scenario [4] .....	28
Εικόνα 1.6: MEC in Internet of Things Scenario [6] .....	29
Εικόνα 2.1: E-UTRAN Scematic [9] .....	32
Εικόνα 2.2: RAN connected to EPC [10] .....	32
Εικόνα 2.3: "Bump in the wire" Scenario [11] .....	34
Εικόνα 2.3: Κατανεμημένος EPC .....	35
Εικόνα 2.4: Κατανεμημένα S/P-GW .....	35
Εικόνα 2.5: Αρχιτεκτονική Κατανεμημένου S-GW με τοπικό breakout [11] .....	36
Εικόνα 2.6: Σενάρια χρήσης 5G [12] .....	37
Εικόνα 2.7: NG-RAN [15] .....	38
Εικόνα 2.8: Serviced-Based Architecture [1] .....	39
Εικόνα 2.9: Reference-Point Architecture [1] .....	41
Εικόνα 2.10: MEC platform in SBA Architecture Scenario [18] .....	42
Εικόνα 2.11: MEC in 5G possible architectures [2] .....	43
Εικόνα 3.1: Μοντέλο ουράς M/M/1 1 [3] .....	45
Εικόνα 3.2: Energy-Harvesting Devices in MEC [26] .....	48
Εικόνα 4.1: Χάρτης Βουδαπέστης [4] .....	51
Εικόνα 4.2: Σενάριο 1 [4] .....	51
Εικόνα 4.3: Σενάριο 2 [4] .....	52
Εικόνα 4.4: Random rule algorithm [4] .....	55
Εικόνα 4.5: Index rule algorithm [4] .....	55
Εικόνα 4.6: SOL rule algorithm [4] .....	56
Εικόνα 4.5: Index rule algorithm [4] .....	57

## Κατάλογος Διαγραμμάτων

<b>Διάγραμμα 4.1: Blocking Probability for new sessions [4] (Σενάριο 1)</b> .....	57
<b>Διάγραμμα 4.2: Dropping Probability for new sessions [4] (Σενάριο 1)</b> .....	57
<b>Διάγραμμα 4.3: Blocking Probability for new sessions [4] (Σενάριο 1)</b> .....	59
<b>Διάγραμμα 4.4: Dropping Probability for new sessions [4] (Σενάριο 1)</b> .....	59
<b>Διάγραμμα 4.5: Blocking Probability for new sessions [4] (Σενάριο 2)</b> .....	60
<b>Διάγραμμα 4.6: Dropping Probability for new sessions [4] (Σενάριο 2)</b> .....	60
<b>Διάγραμμα 4.7: Blocking Probability for new sessions [4] (Σενάριο 2)</b> .....	61
<b>Διάγραμμα 4.8: Dropping Probability for new sessions [4] (Σενάριο 2)</b> .....	61
<b>Διάγραμμα 5.1: Battery energy level</b> .....	78
<b>Διάγραμμα 5.2: Μέσο κόστος εκτέλεσης</b> .....	79
<b>Διάγραμμα 5.3: Average ratio of chosen modes, d=50m</b> .....	80
<b>Διάγραμμα 5.4: Average ratio of chosen modes, d=40m</b> .....	81
<b>Διάγραμμα 5.5: Average ratio of chosen modes, d=40m</b> .....	81

# Κατάλογος Πινάκων

Πίνακας 2.1: Απαιτήσεις δικτύων 5G.....	39
Πίνακας 3.1: Μέση Πυκνότητα Ενέργειας σε τεχνολογίες συλλογής ενέργειας [5].....	49
Πίνακας 4.1: Εξυπηρέτηση UEs σε περιοχές συναρτήσεως της εμβέλειας των MEC [4].....	53
Πίνακας 4.2: Παρουσίαση παραμέτρων προβλήματος [4].....	54
Πίνακας 4.3: Ορισμός κοινών χαρακτηριστικών αλγορίθμων.....	55
Πίνακας 5.1: Αρχικοποίηση παραμέτρων αλγορίθμου LODCO.....	77

# ΚΕΦΑΛΑΙΟ 1<sup>ο</sup>

## MOBILE EDGE COMPUTING

### 1.1 Εισαγωγή στο MEC

Η τελευταία δεκαετία αποτελεί περίοδο άνθησης και εξάπλωσης της τεχνολογίας του υπολογιστικού νέφους (cloud computing). Το modus operandi της συγκεκριμένης τεχνολογίας βασίζεται στη μέθοδο της κεντρικής διαχείρισης πόρων. Συνοπτικά, ο εκάστοτε χρήστης εκμεταλλεύεται εταιρικά λογισμικά, αποθηκευτικούς χώρους, βάσεις δεδομένων και λοιπούς υπολογιστικούς πόρους που βρίσκονται διαθέσιμοι στο διαδίκτυο, το οποίο αποτελεί το προαναφερθέν «νέφος». Η μέθοδος αυτή όμως αποκτά ένα εγγενές μειονέκτημα, θεωρώντας πως κάθε cloud server βρίσκεται κατά κανόνα απομακρυσμένος τόσο από το χρήστη, όσο και από το υπόλοιπο δίκτυο. Το μειονέκτημα αυτό είναι η εισαγωγή καθυστέρησης στην ανταλλαγή δεδομένων μεταξύ χρήστη και εξυπηρετητή. Θα δειχθεί παρακάτω πως το χαρακτηριστικό αυτό αποτελεί σοβαρό ζήτημα σε ένα μεγάλο σύνολο εφαρμογών και υπηρεσιών οι οποίες απαιτούν ελάχιστο latency και απόκριση σε πραγματικό χρόνο.

Η ανάγκη για όσο το δυνατόν περισσότερο άμεση και δυναμική ανταλλαγή δεδομένων μεταξύ χρήστη και εξυπηρετητή οδήγησε στην ιδέα της μεταφοράς του νέφους στις άκρες του δικτύου. Με τον τρόπο αυτό, η απόσταση του χρήστη από τις πηγές υπολογιστικής ισχύος και δεδομένων που εκμεταλλεύεται, μειώνεται δραστικά. Η μείωση αυτή συνεπάγεται έτσι το επιθυμητό χρονικό κέρδος στην (από άκρο σε άκρο) επικοινωνία συσκευής με εξυπηρετητή. Η τεχνική αυτή ονομάζεται Mobile Edge Computing και βασίζεται σε αυτή ακριβώς την αποκεντροποίηση των υπολογιστικών πόρων του νέφους. Οι βασικές της αρχές θα παρουσιαστούν στο παρόν κεφάλαιο.

### 1.2 Αρχιτεκτονική MEC

#### 1.2.1 Μοντέλα Επικοινωνίας

Οι δίαυλοι επικοινωνίας ανάμεσα στο χρήστη και τους cloud εξυπηρετητές του MCC (Mobile Cloud Computing), μοντελοποιούνται συνήθως ως bit pipes με ένα συγκεκριμένο ρυθμό μετάδοσης δεδομένων. Στα συστήματα MEC, μικρής κλίμακας τοπικοί εξυπηρετητές δίνουν τη δυνατότητα ταχείας ασύρματης μετάδοσης δεδομένων. Ο τρόπος αυτός, διαφέρει σημαντικά από τις αντίστοιχες ενσύρματες διεπαφές κατά τα κάτωθι χαρακτηριστικά του [6].

I) Χρησιμοποιώντας την ατμόσφαιρα ως αγωγό το ασύρματο σήμα υπόκειται σε φαινόμενα όπως η διάθλαση και η περίθλαση που οδηγούν σε προβλήματα όπως η διασυμβολική παρεμβολή (inter-symbol inference, *ISI*). Παρουσιάζεται λοιπόν η ανάγκη για την αντιμετώπιση αυτών με σκοπό την αξιοπιστία των επικοινωνιών.

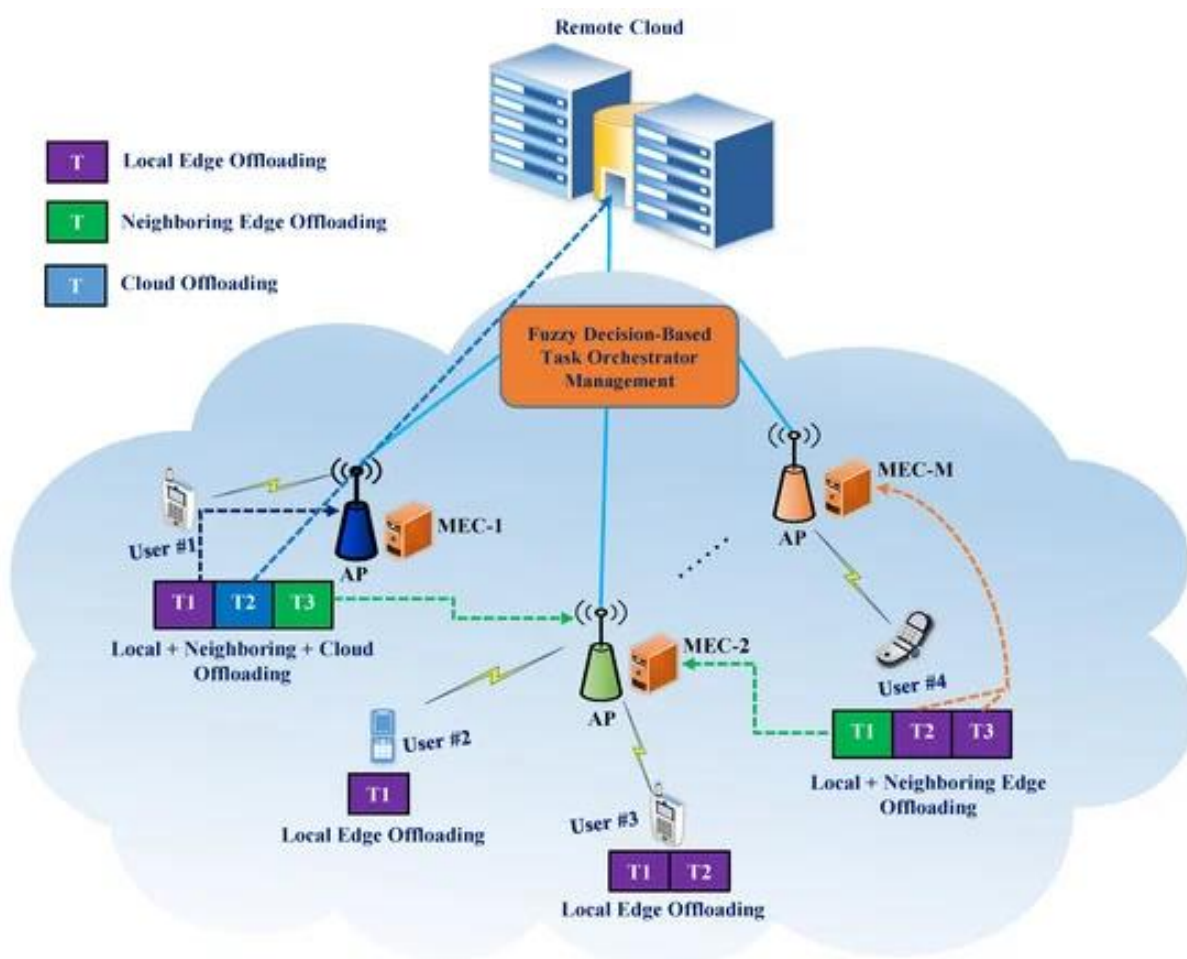
II) Η ασύρματη μετάδοση ενός σήματος μπορεί να οδηγήσει στην παρεμβολή αυτού με σήματα στο ίδιο φάσμα. Το γεγονός αυτό έχει ως αποτέλεσμα την πιθανότητα λαθών στην αποκωδικοποίηση από μεριάς δέκτη.

III) Η έλλειψη διαθέσιμου φάσματος (spectrum shortage) στις wireless μεταδόσεις αποτελεί ένα από τα σημαντικότερα σύγχρονα ζητήματα της επιστήμης των τηλεπικοινωνιών. Ως εκ τούτου, παρουσιάζεται η

ανάγκη ανάπτυξης μεθόδων που θα επιτρέψει την αξιόπιστη ασύρματη μετάδοση πολλαπλών σημάτων σε κοινό εύρος ζώνης.

Στα συστήματα MEC η επικοινωνία πραγματοποιείται μεταξύ κινητών συσκευών και σημείων πρόσβασης (access points, ή APs) με τη δυνατότητα άμεσης επικοινωνίας μεταξύ συσκευών (device-to-device, ή D2D).

Οι εξυπηρετητές MEC αποτελούν μικρής κλίμακας κέντρα δεδομένων που χειρίζεται ο εκάστοτε πάροχος υπηρεσιών τηλεπικοινωνιών/Cloud Computing. Αυτά συσχετίζονται με ασύρματα APs, όπως για παράδειγμα Wi-Fi routers και Base Stations. Όπως φαίνεται και στο σχήμα 1.1 τα ασύρματα αυτά APs όχι μόνο προσφέρουν στους εξυπηρετητές MEC το interface για την ασύρματη μετάδοση δεδομένων, αλλά δίνουν επίσης τη δυνατότητα σύνδεσης με άλλους MEC εξυπηρετητές ή με μεγάλα Cloud data centers. Με τον τρόπο αυτό, ο εξυπηρετητής μπορεί να αποθέσει την εκτέλεση μέρους των υπολογιστικών του διεργασιών σε άλλο φορέα με σκοπό τη γρηγορότερη ολοκλήρωσή τους. Ακόμη και σε συνθήκες που η επικοινωνία του χρήστη με κάποιον εξυπηρετητή δεν καθίσταται δυνατή, η τεχνολογία MEC προσφέρει τη δυνατότητα D2D συνδέσεων με γειτονικές συσκευές. Μια τέτοια περίπτωση προσφέρει στο χρήστη δύο λύσεις. Πρώτον, την έμμεση σύνδεσή του με MEC εξυπηρετητή χρησιμοποιώντας ως ενδιάμεσο κόμβο έναν γειτονικό χρήστη. Δεύτερον, οι D2D συνδέσεις πολλαπλών συσκευών που μοιράζονται τους υπολογιστικούς τους πόρους μπορούν να δημιουργήσουν τοπικά υπολογιστικά «νέφη» με σημαντική ισχύ επεξεργασίας δεδομένων.



**Εικόνα 1.1: MEC System**

### **1.2.2 Το πλαίσιο MEC**

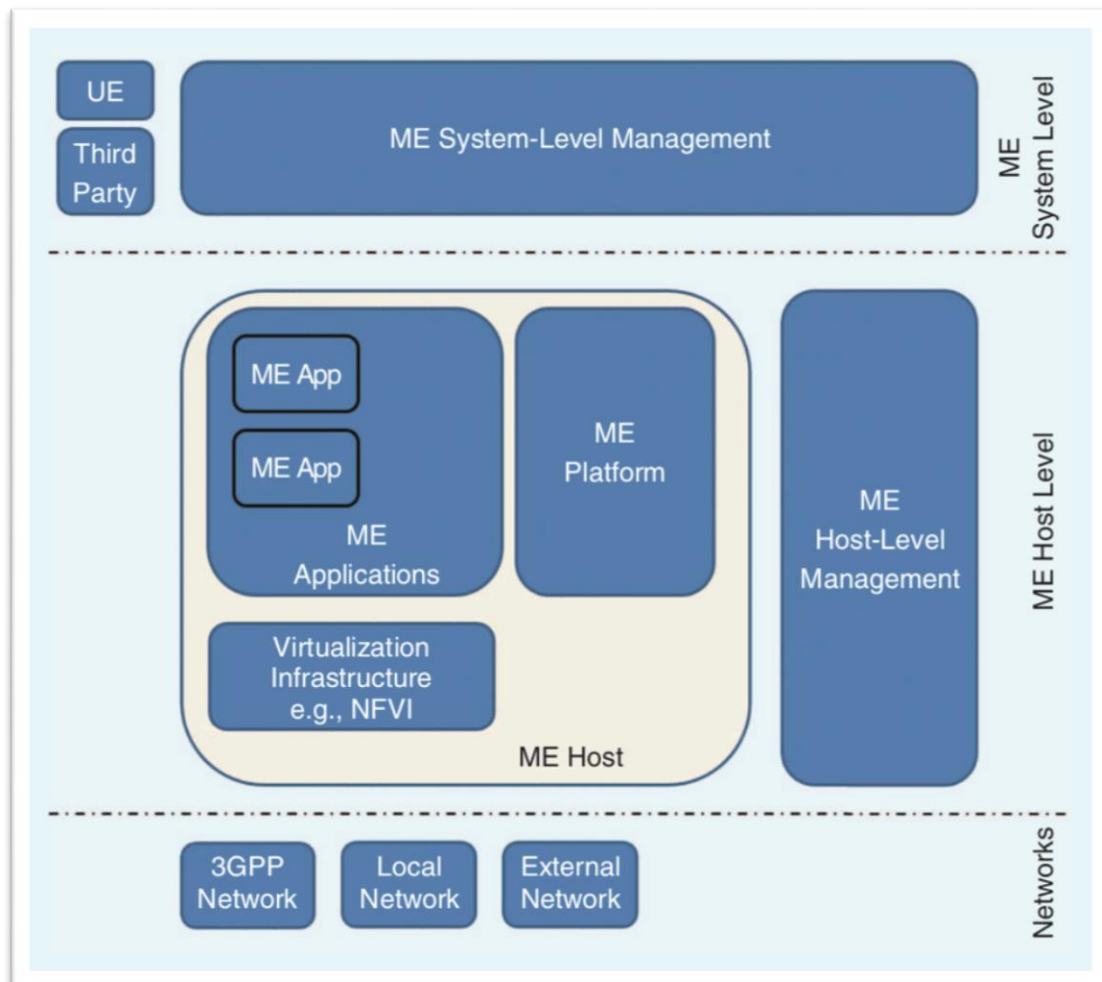
Στο σχήμα 1.2 παρουσιάζεται ο σκελετός ενός πλαισίου MEC. Περιλαμβάνει τις λειτουργικές μονάδες του συστήματος οι οποίες περιγράφονται σε υψηλό επίπεδο. Οι μονάδες αυτές χωρίζονται σε τρία επίπεδα [7]:

- 1) Το επίπεδο συστήματος (system level)
- 2) Το επίπεδο διακομιστή (host level)
- 3) Το επίπεδο δικτύου (network level)

Στη βάση του πλαισίου βρίσκεται το **επίπεδο δικτύου**. Αποτελείται από εξωτερικές οντότητες, όπως το δίκτυο του 3GPP (3<sup>rd</sup> Generation Partnership Project), καθώς και τα εκάστοτε τοπικά ή εξωτερικά δίκτυα. Το επίπεδο αυτό αναπαριστά τη σύνδεση του χρήστη με τοπικά δίκτυα, δίκτυα κινητής τηλεφωνίας, καθώς και εξωτερικά δίκτυα όπως το Διαδίκτυο.

Στο αμέσως επόμενο στρώμα, έχουμε το **επίπεδο του host**, που αποτελείται από δυο μεγάλες υποομάδες: α) Το ME host και β) το τμήμα διαχείρισης του host (Host-level management). Ο ME host χωρίζεται περαιτέρω για να περιλαμβάνει την πλατφόρμα MEC (MEC Platform), τις εφαρμογές MEC (MEC Applications) και τέλος τη δομή visualization.

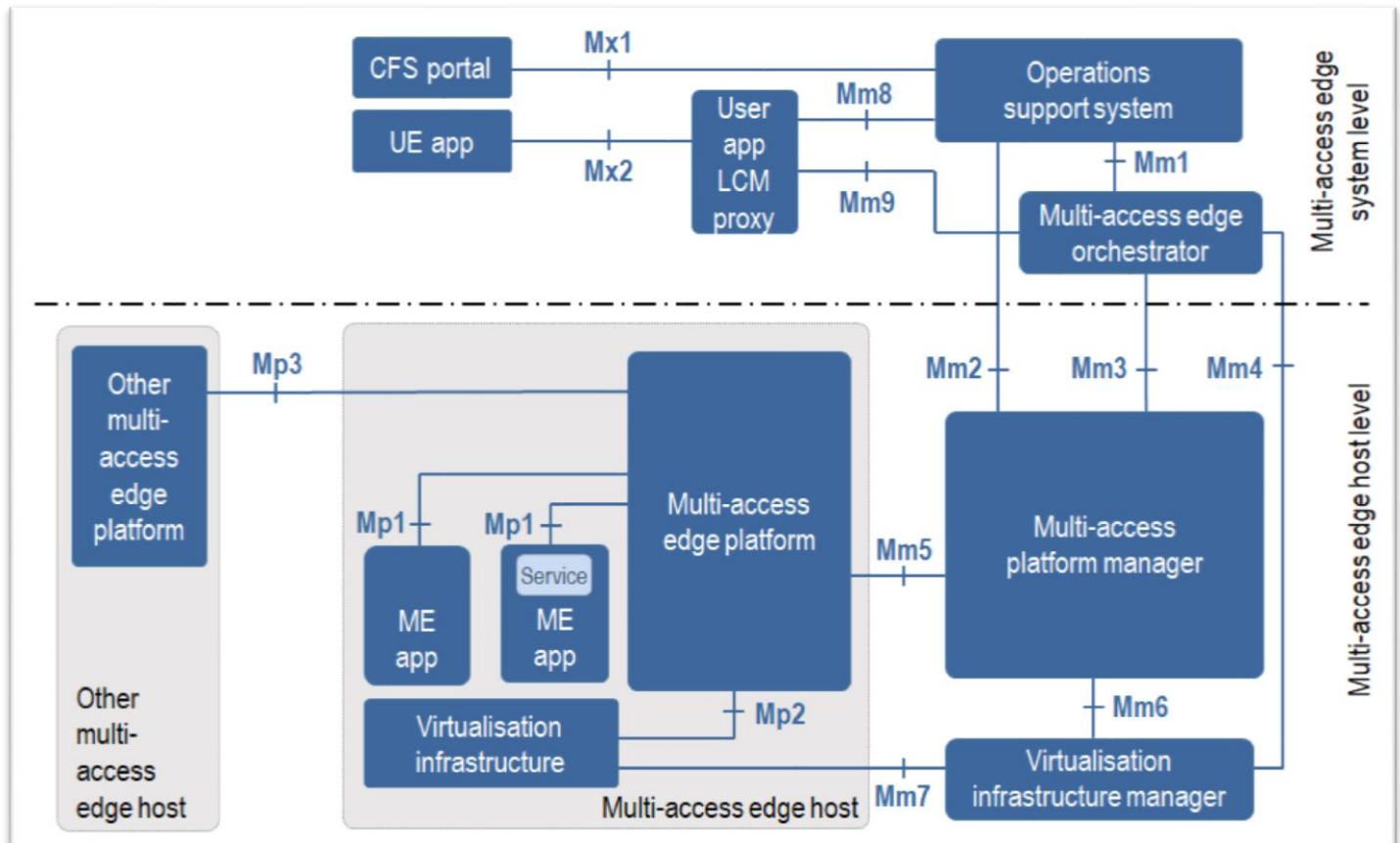
Στην κορυφή του πλαισίου έχουμε το **επίπεδο διαχείρισης**, το οποίο καθ' ορισμό κατέχει τη γενική επίβλεψη όλου του συστήματος MEC. Περιλαμβάνει το σύνολο του εξοπλισμού των χρηστών (*User Equipment, UE*) και τους φορείς που είναι υπεύθυνοι για την εκτέλεση μιας MEC εφαρμογής από το χρήστη, μέσω του δικτύου που προσφέρει ένας εξωτερικός πάροχος.



**Εικόνα 1.2: MEC Framework [2]**

### 1.2.3 Η Αρχιτεκτονική Αναφοράς MEC

Μια περισσότερο εις βάθος κατανόηση των συστημάτων Mobile Edge Computing μπορεί να αποκτηθεί από τη μελέτη της γενικευμένης αρχιτεκτονικής αναφοράς MEC. Η γενική δομή της ακολουθεί το σχήμα του πλαισίου MEC με ειδοποιό διαφορά την έλλειψη του στρώματος δικτύου (network level). Τα επί μέρους στοιχεία που θα αναλυθούν παρακάτω απεικονίζονται στο σχήμα 1.3. [7]



Εικόνα 1.3: MEC Architecture Reference Model [2]

Το επίπεδο του MEC host αποτελείται από την **πλατφόρμα ME** (*Multi-access Edge platform, MEP*), τη **μονάδα διαχείρισης της ME πλατφόρμας** (*Mobile Edge Platform Manager, MEPM*) και τη **δομή εικονικοποίησης** (*virtualization infrastructure*).

Η **δομή εικονικοποίησης** (*virtualization infrastructure*) είναι υπεύθυνη για την παροχή εικονικών υπολογιστικών πόρων, αποθηκευτικού χώρου και πρόσβασης στο δίκτυο στις εφαρμογές που τρέχουν μέσω ενός MEC server. Επιπλέον, το virtualization infrastructure τρέχοντας το εικονικό λογισμικό και λαμβάνοντας το ρόλο του χειριστή του, είναι υπεύθυνο για τη συλλογή πληροφοριών σχετικά με την επίδοση των εικονικών πόρων και την εμφάνιση τυχόν σφαλμάτων. Η ανατροφοδότηση των δεδομένων που συλλέγονται προς τη διαχείριση υποδομών εικονικοποίησης (*Virtualization Infrastructure Manager, VIM*) γίνεται μέσω του σημείου αναφοράς Mm7, ώστε ο VIM να ενημερώνει το MEPM μέσω του σημείου αναφοράς Mm6 για την κατάσταση των πόρων του συστήματος.

Η **πλατφόρμα ME** (*Multi-access Edge platform, MEP*) περιέχει στο εσωτερικό της μια συλλογή βασικών λειτουργιών, οι οποίες είναι αναγκαίες για την εκτέλεση μιας MEC εφαρμογής σε κάποιο MEC host. Η πλατφόρμα λειτουργεί ουσιαστικά ως διαμεσολαβητής μεταξύ των εφαρμογών και δομής εικονικοποίησης. Οι λειτουργίες που περιέχει επιτρέπουν τη σωστή δρομολόγηση δεδομένων ανάμεσα σε εφαρμογές, υπηρεσίες δικτύου και εικονικών πόρων κάθε είδους. Η δρομολόγηση αυτή υπαγορεύεται από τη MEPM, βασίζεται σε πολιτικές και πρωτόκολλα που η ίδια θέτει, και ενημερώνεται στη MEP μέσω του σημείου αναφοράς Mm5. Τέλος, η MEP μπορεί να επικοινωνήσει άμεσα με άλλες πλατφόρμες ME, δημιουργώντας ένα δίκτυο επικοινωνίας με σκοπό τη βέλτιστη λειτουργία του συστήματος.

Η **μονάδα διαχείρισης της ME πλατφόρμας** (*Mobile Edge Platform Manager, MEPM*) με τη σειρά της χωρίζεται σε τρία διακριτά μέρη:

- 1) Μονάδα διαχείρισης κύκλου ζωής των εφαρμογών
- 2) Δομή διαχείρισης ME πλατφόρμας
- 3) Λειτουργίες διαχείρισης κανόνων, απαιτήσεων και πλαισίων ME εφαρμογών

Η μονάδα διαχείρισης κύκλου ζωής εφαρμογών είναι υπεύθυνη για τις διαδικασίες που αποφασίζουν κάθε στιγμή αν μια εφαρμογή θα εκτελεστεί ή θα τερματίσει, καθώς και για την ενημέρωση του Mobile Edge Operator (MEO) σχετικά με γεγονότα που αφορούν τις εφαρμογές. Οι λειτουργίες κανόνων και απαιτήσεων εμπεριέχουν πληροφορίες σχετικά με τους κανόνες, τις εξουσιοδοτήσεις και τις DNS παραμέτρους, οι οποίες χρησιμοποιούνται κάθε φορά που προκύπτει ζήτημα κανόνων σε σύγκρουση. Το σημείο αναφοράς Mm2 χρησιμοποιείται από τη MEPM για να αλληλεπιδράσει με το επίπεδο συστήματος και συγκεκριμένα με το σύστημα υποστήριξης λειτουργιών (*Operations Support System, OSS*), ώστε να πραγματοποιείται έλεγχος απόδοσης όλης της ME πλατφόρμας. Τέλος, το σημείο αναφοράς Mm3, συνδέει τη MEPM με τη μονάδα ενορχήστρωσης ME του συστήματος επιπέδου (*Mobile Edge Orchestrator, MEO*) παρέχοντας της υποστήριξη στη διαχείριση του κύκλου ζωής των εφαρμογών, καθώς και ανταλλαγή πληροφορίας σχετικά με τη διαθεσιμότητα των ME υπηρεσιών στο σύστημα.

Το **επίπεδο συστήματος** διαχωρίζεται και αυτό σε δυο βασικά επιμέρους στοιχεία, τη **μονάδα ενορχήστρωσης ME του συστήματος επιπέδου** (*Mobile Edge Orchestrator, MEO*) και το **σύστημα υποστήριξης λειτουργιών** (*Operations Support System, OSS*).

Η **μονάδα ενορχήστρωσης ME του συστήματος επιπέδου (MEO)** αποτελεί την κεντρική μονάδα ενός MEC συστήματος, καθώς επιβλέπει τη διαχείριση των πόρων και των δυνατοτήτων όλου του ME δικτύου. Στο εσωτερικό της περιέχεται το σύνολο των πληροφοριών που αφορούν το σύστημα, όπως τους ενεργούς ME hosts, τις υπηρεσίες και τους πόρους που βρίσκονται διαθέσιμοι σε αυτούς, τις εφαρμογές που τρέχουν σε κάθε στιγμή και την τοπολογία του δικτύου. Επιπλέον, ο Orchestrator είναι υπεύθυνος για την επίβλεψη των ME εφαρμογών ελέγχοντας την αξιοπιστία και τις πολιτικές που ακολουθούν κατά την εκτέλεσή τους. Ειδικότερα, αν μια εφαρμογή επιθυμεί να εκτελεστεί αναφέρει στον ενορχηστρωτή τις προδιαγραφές που χρειάζεται για να τρέξει (λ.χ. ελάχιστη καθυστέρηση, ελάχιστοι υπολογιστικοί πόροι συστήματος και δικτύου), ώστε αυτός να επιλέξει τον κατάλληλο ME host για την εκτέλεσή της. Το σημείο αναφοράς Mm4 χρησιμοποιείται από τη MEO, ώστε να λαμβάνει δεδομένα από την μονάδα διαχείρισης εικονικοποίησης (VIM) σχετικά με τους διαθέσιμους πόρους του εκάστοτε host.

Το **σύστημα υποστήριξης λειτουργιών (OSS)** είναι ιεραρχικά το ανώτερο επίπεδο διαχείρισης ενός MEC συστήματος. Λαμβάνει αιτήσεις εκτέλεσης και τερματισμού εφαρμογών, τόσο από τους χρήστες μέσω των συσκευών τους (*user equipment, UE*), όσο και από την υπηρεσία εξυπηρέτησης πελατών (*client-facing service, CFS*). Δεδομένου ότι το OSS βρίσκεται ακριβώς στο σύνορο μεταξύ του δικτύου και του εξωτερικού κόσμου, επωμίζεται την ευθύνη του ελέγχου της αξιοπιστίας κάθε εξωτερικού request. Αν κρίνει πώς ένα request είναι ασφαλές να εκτελεστεί στο σύστημα το προωθεί στη MEO για περαιτέρω επεξεργασία μέσω του σημείου αναφοράς Mm1. Τέλος, το σύστημα αυτό έχει τη δυνατότητα να μεταφέρει εφαρμογές μεταξύ διαφορετικών cloud systems.



Κλείνοντας την ανάλυση του επιπέδου συστήματος συναντάται η **υπομονάδα διαχείρισης κύκλου ζωής** (*Life Cycle Management, LCM*). Το στοιχείο αυτό χρησιμοποιείται ως δίαυλος από τους χρήστες του συστήματος MEC με σκοπό την επικοινωνία αιτημάτων τους σχετικά με την εκτέλεση ή τον τερματισμό εφαρμογών προς το OSS ή τη MEO (σημεία αναφοράς Mm8, Mm9). Οι εξωτερικές οντότητες (UEs, system clients) χρησιμοποιούν τα σημεία αναφοράς Mx1, Mx2 για να αποκτήσουν πρόσβαση σε όλο το σύστημα MEC.

## 1.3 Πλεονεκτήματα MEC

Το MEC υπερτερεί σημαντικά από τα MCC συστήματα σε συγκεκριμένους τομείς εφαρμογών. Παρακάτω θα αναφερθούν κάποια από τα βασικά πλεονεκτήματα που προσφέρει η τεχνολογία αυτή συγκριτικά με το Cloud Computing [6].

### 1.3.1 Χαμηλή Καθυστέρηση

Η καθυστέρηση (*latency*) σε μια υπηρεσία κινητής τηλεφωνίας είναι συνάρτηση τριών συνιστωσών:

- 1) Καθυστέρηση μετάδοσης (propagation latency)
- 2) Καθυστέρηση επικοινωνίας (communication latency)
- 3) Καθυστέρηση υπολογισμών (computation latency)

Η μετάδοση δεδομένων σε ένα MEC δίκτυο αφορά κατά κανόνα χαμηλές αποστάσεις της τάξεως μερικών εκατοντάδων μέτρων, εάν πρόκειται για μεταδόσεις D2D σε ένα small-cell δίκτυο και τυπικά όχι μεγαλύτερες του 1km σε γενικές περιπτώσεις. Το γεγονός αυτό έρχεται σε αντίθεση με τις μεταδόσεις δεδομένων σε απομακρυσμένους κόμβους του δικτύου και σε αποστάσεις με εύρος από δεκάδες χιλιόμετρα μέχρι διαηπειρικές διαδρομές που συναντάμε στο Cloud Computing. Σαν αποτέλεσμα, τα συστήματα MEC προσφέρουν σημαντικά λιγότερη καθυστέρηση στη μετάδοση δεδομένων από και προς το χρήστη.

Στα MCC συστήματα η καθυστέρηση επικοινωνίας εισάγεται διότι τα δεδομένα πρέπει να περάσουν από μια σειρά κόμβων. Κάποιοι από αυτούς όπως το δίκτυο ραδιοπρόσβασης (*Radio-Access Network, RAN*), το backhaul network και το Διαδίκτυο, όπου διαδικασίες όπως ο έλεγχος διακίνησης (traffic control, routing) εισάγουν σημαντική καθυστέρηση. Αντιθέτως, στην περίπτωση του MEC η επικοινωνία παραμένει και περιορίζεται στα σύνορα (edges) του δικτύου αποφεύγοντας έτσι τον παραπάνω κίνδυνο.

Τέλος, ένα MCC σύστημα διαθέτει υπολογιστική ισχύ που υπερτερεί σε σύγκριση με ένα αντίστοιχο MEC κατά πολλές τάξεις μεγέθους. Όμως την ίδια στιγμή το σύστημα αυτό χρησιμοποιείται από έναν αριθμό ενεργών χρηστών επίσης σημαντικά μεγαλύτερο. Σαν αποτέλεσμα, ένας Cloud Server μπορεί να εισάγει καθυστέρηση στην εκτέλεση μιας εφαρμογής καθώς αναζητεί ελεύθερους πόρους για να την εκτελέσει. Αντιθέτως, ένας Base Station (BS) που μπορεί να χρησιμοποιεί ένα MEC δίκτυο διαχειρίζεται τις λιγότερες αιτήσεις για εκτέλεση εφαρμογών που δέχεται σε μικρότερο χρόνο. Αξίζει να σημειωθεί σε αυτό το σημείο, πως οι σύγχρονοι BSs που χρησιμοποιούνται στην τεχνολογία του 5G και του MEC μπορεί να μην προσφέρουν τους πολυάριθμους πόρους ενός Cloud συστήματος, αλλά διαθέτουν υπολογιστική ισχύ ικανή να εκτελέσει επαρκώς απαιτητικές εφαρμογές και διεργασίες.

### 1.3.2 Εξοικονόμηση Ενέργειας

Ένα διαχρονικό πρόβλημα των φορητών συσκευών τεχνολογίας αποτελεί η συνεχώς αυξανόμενη ανάγκη για εξοικονόμηση ενέργειας, ώστε να διατηρούν υψηλό επίπεδο αυτονομίας. Επιπλέον, οι σύγχρονες συσκευές προσφέρουν ασύρματη συνδεσιμότητα χρησιμοποιώντας πληθώρα ανάλογων τεχνολογιών (λ.χ. Bluetooth, NFC, Wi-Fi) η χρήση των οποίων απαιτεί υψηλή κατανάλωση ενέργειας. Ένα από τα κυριότερα χαρακτηριστικά του MEC, όπως αναφέρθηκε, αποτελεί η δυνατότητα μεταφοράς της εκτέλεσης μιας διεργασίας από τη συσκευή ενός χρήστη σε έναν MEC server. Με αυτό τον τρόπο, υπολογιστικός φόρτος που υπό άλλες συνθήκες θα βάραινε τη συσκευή του χρήστη αυξάνοντας έτσι την κατανάλωση της αποθηκευμένης της ενέργειας μεταφέρεται στην άκρη του δικτύου, ώστε να εκτελεστεί εκεί. Ενδεικτικά, πειράματα έχουν δείξει πως η μετάθεση διεργασιών σε MEC servers από συσκευές που εκτελούν Augmented Reality (AR) εφαρμογές μπορεί να αυξήσει την αυτονομία της μπαταρίας τους από 30 έως 50 τοις εκατό.

### **1.3.3 Επίγνωση Γενικού Πλαισίου**

Με τον όρο επίγνωση γενικού πλαισίου (Context-Awareness) περιγράφεται η δυνατότητα ενός MEC server να χρησιμοποιεί την πρόσβασή του σε πολλαπλές συσκευές συνδεδεμένες σε αυτόν, ώστε να αντλήσει δεδομένα σχετικά με πληροφορίες, όπως η τοποθεσία και οι συνθήκες περιβάλλοντος του χρήστη. Η δυνατότητα αυτή επιτρέπει στο σύστημα να προσαρμόσει τις υπηρεσίες του στις ανάγκες του εκάστοτε χρήστη. Εφαρμογές που βασίζονται σε τέτοιου είδους δεδομένα μπορούν έτσι να εκτελεστούν αποδοτικά βελτιώνοντας την εμπειρία του χρήστη και την ποιότητα των υπηρεσιών.

### **1.3.4 Ιδιωτικότητα/Ασφάλεια**

Το μοντέλο του MCC υπαγορεύει τη συγκέντρωση μεγάλου αριθμού πληροφοριών χρηστών σε μια κεντρική τοποθεσία, το Cloud. Πλατφόρμες όπως το Microsoft Azure ή το Amazon Web Services αποτελούν κόμβους συγκέντρωσης δεδομένων σε απομονωμένα data centers καθιστώντας τα έτσι ευάλωτα σε κυβερνοεπιθέσεις ή απώλειες λόγω δυσλειτουργίας του λογισμικού. Επιπλέον, οι χρήστες καλούνται να παραχωρήσουν το δικαίωμα διαχείρισης των προσωπικών τους δεδομένων σε εξωτερικούς φορείς. Αντιθέτως, η κατανομημένη φύση των MEC servers καθώς και η σαφώς χαμηλότερη συγκέντρωση δεδομένων που προσφέρουν τους καθιστούν λιγότερο πιθανούς στόχους μιας κακόβουλης επίθεσης μέσω του δικτύου. Τέλος, δίνεται η δυνατότητα στους χρήστες να διαθέτουν ιδιωτικά υπολογιστικά νέφη μικρής κλίμακας (small-scale Cloudlets) μέσω των MEC servers. Η διαχείριση της προσβασιμότητας σε αυτά, θα υπόκειται στους ίδιους και όχι σε εξωτερικούς φορείς.

## **1.4 Εφαρμογές MEC**

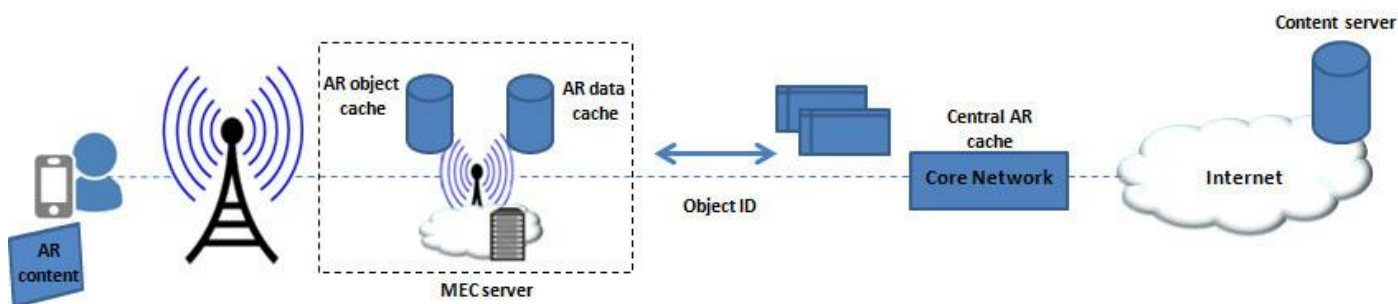
Στο παρόν κεφάλαιο θα αναλυθούν παραδείγματα χρήσης της τεχνολογίας MEC από ένα σύνολο εφαρμογών. Οι εφαρμογές αυτές χρησιμοποιώντας την πλατφόρμα του MEC προσφέρουν υψηλό επίπεδο QoS, καθώς εκμεταλλεύονται βέλτιστα τα μοναδικά πλεονεκτήματα που αυτή προσφέρει. Συγκεκριμένα, θα αναφερθούν οι περιπτώσεις της **επαυξημένης πραγματικότητας** (*Augmented Reality, AR*), της **έξυπνης επιτάχυνσης παροχής βίντεο** (*Intelligent Video Streaming Acceleration*), των **διασυνδεδεμένων έξυπνων αυτοκινήτων** (*Connected Smart Cars*) και τέλος του **Internet of Things** (*IoT*) [8].

### **1.4.1 Επαυξημένη πραγματικότητα**

Η τεχνολογία της επαυξημένης πραγματικότητας αναλύει ένα πραγματικό περιβάλλον και το εμπλουτίζει με ψηφιακά προβαλλόμενες πληροφορίες όπως ήχος, βίντεο, εικονικά γραφικά και δεδομένα GPS. Για

παράδειγμα, ένας επισκέπτης μουσείου, γκαλερί, αρχιτεκτονικού μνημείου ή αθλητικού γεγονότος μπορεί να επωφεληθεί άμεσα από αυτή την τεχνολογία. Ο χρήστης μπορεί στρέφοντας τη συσκευή του προς ένα σημείο ενδιαφέροντος, να λάβει στην οθόνη του δεδομένα σχετικά με την εικόνα που απεικονίζει η κάμερά του. Στην περίπτωση του εκθέματος ενός μουσείου ή μιας γκαλερί, ο χρήστης μπορεί να πληροφορηθεί σχετικά με το ιστορικό του υπόβαθρο ή τον καλλιτέχνη που το δημιούργησε. Η εφαρμογή μπορεί να αναγνωρίσει στην περίπτωση ενός αγώνα μπάσκετ έναν παίκτη και να πληροφορήσει με γραφικό το χρήστη για τα στατιστικά του στον εν λόγω αγώνα.

Τέτοιου είδους υπηρεσίες απαιτούν τη χρήση μιας εφαρμογής, η οποία θα λαμβάνει τα δεδομένα που καταγράφει η κάμερα του χρήστη και μέσω αυτών εξάγει συμπεράσματα σχετικά με τη θέση και τον προσανατολισμό του. Η εφαρμογή οφείλει να προσαρμόσει το output της καταλλήλως και σε πραγματικό χρόνο αν τα δεδομένα αυτά αλλάξουν στην περίπτωση που ο χρήστης μετακινηθεί. Η εκτέλεση μιας τέτοιας υπηρεσίας σε MEC πλατφόρμα αντί του cloud αποδεικνύεται άκρως αποδοτικότερη, καθώς το χαμηλό latency στη μετάδοση της πληροφορίας και η δυνατότητα ταχείας επεξεργασίας δεδομένων σε μικρό χρόνο βοηθούν στην άμεση ενημέρωση του interface σε πραγματικό χρόνο ανεξάρτητα των διαρκών αλλαγών στη θέση και τον προσανατολισμό του χρήστη. Η διαδικασία που μόλις περιεγράφηκε φαίνεται στην παρακάτω εικόνα σχηματικά:



*Εικόνα 1.4: MEC in Augmented Reality Scenario [4]*

### **1.4.2 Έξυπνη επιτάχυνση παροχής βίντεο**

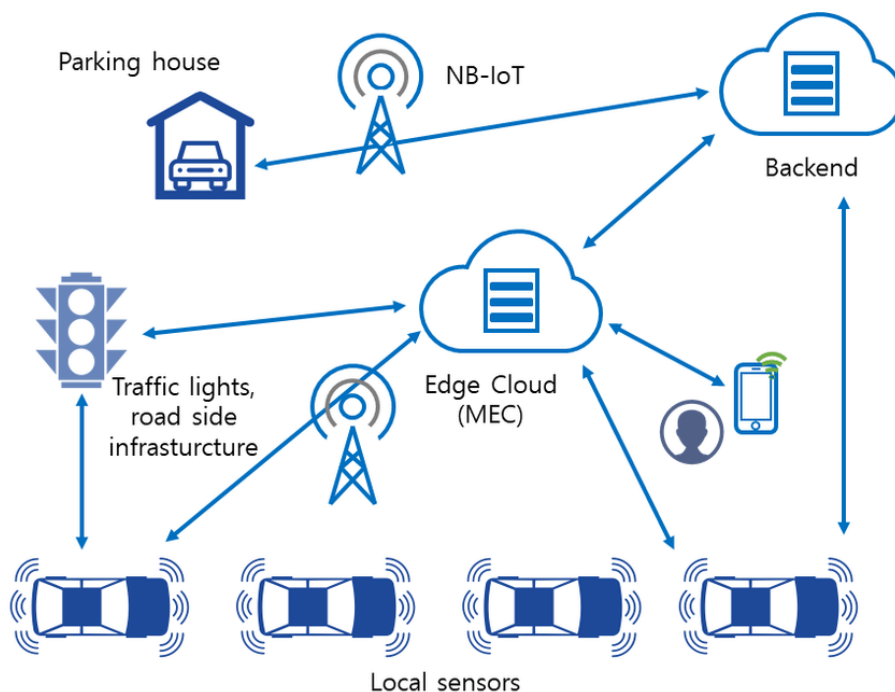
Τα διαδικτυακά μέσα όπως οι υπηρεσίες παροχής βίντεο χρησιμοποιούν για την ανταλλαγή δεδομένων το **Πρωτόκολλο Μεταφοράς Υπερκειμένου** (*HyperText Transfer Protocol, HTTP*) πάνω από το γνωστό **Πρωτόκολλο Ελέγχου Μεταφοράς** (*Transmission Control Protocol, TCP*). Η χωρητικότητα του ραδιοδικτύου μπορεί να περάσει από αλλαγές της τάξεως πολλών μονάδων μέτρησης μέσα σε πολύ μικρό χρονικό διάστημα, εξαιτίας της διαρκούς σύνδεσης και αποσύνδεσης χρηστών σε αυτό. Το πρωτόκολλο TCP είναι πιθανό να μην μπορεί έτσι να προσαρμοστεί στις συνεχώς εναλλασσόμενες αυτές συνθήκες, οδηγώντας σε αναποτελεσματική αξιοποίηση των πόρων του δικτύου. Το πρόβλημα αυτό μπορεί να επιλυθεί με τη χρήση μιας εφαρμογής επιταχυντή που θα τρέχει σε έναν MEC server.

Ο επιταχυντής αυτός παρέχει στον εξυπηρετητή βίντεο την πληροφορία της διαθεσιμότητας σχετικά με τη διεκπεραιωτική ικανότητα της downlink διεπαφής. Ο εξυπηρετητής προσαρμόζει κατάλληλα το ρυθμό μετάδοσης προς κάθε χρήστη, με τρόπο ώστε η διαθεσιμότητα του δικτύου να χρησιμοποιείται βέλτιστα, ενώ ταυτόχρονα βοηθάει το πρωτόκολλο TCP να πραγματοποιεί σωστό έλεγχο συμφόρησης.

### **1.4.3 Διασυνδεδεμένα Έξυπνα Αυτοκίνητα**

Η τεχνολογία των έξυπνων αυτοκινήτων γνωρίζει τα τελευταία χρόνια σημαντική άνθηση και ολοένα αυξανόμενο ερευνητικό ενδιαφέρον. Η επικοινωνία μεταξύ οχήματος και αισθητήρων παροχής πληροφοριών σχετικά με τη γεωμετρία του οδικού δικτύου, των καιρικών συνθηκών και της κίνησης γειτονικών οχημάτων, είναι κομβικής σημασίας για την ασφάλεια του οδηγού. Ένα έξυπνο όχημα μπορεί να προσφέρει στον οδηγό πληθώρα άλλων χαρακτηριστικών εκτός του προφανούς παραδείγματος της αυτόνομης οδήγησης, όπως εύρεση σημείου στάθμευσης και συστήματα ψυχαγωγίας. Καθώς η δημοφιλία των μέσων αυτών αυξάνεται ο αριθμός τους θα πληθαίνει, όπως και ο όγκος των εισερχόμενων εξ αυτών δεδομένων.

Γίνεται αντιληπτό πως η διασύνδεση των έξυπνων οχημάτων μεταξύ τους αποτελεί σημαντικό παράγοντα για την πλήρη εκμετάλλευση της εν λόγω τεχνολογίας. Κάθε μονάδα ξεχωριστά λειτουργεί σαν κόμβος συλλογής πληροφοριών, δημιουργώντας έτσι ένα δίκτυο διασυνδεδεμένων οχημάτων, τα οποία επικοινωνούν μεταξύ τους χρησιμοποιώντας το δίκτυο LTE (Long-Term Evolution). Στους σταθμούς βάσης του LTE φιλοξενούνται MEC servers οι οποίοι δεχόμενοι όλα αυτά τα δεδομένα, συμβάλλουν στην επεξεργασία και την άμεση αποστολή τους στους χρήστες. Έτσι, ένα έξυπνο όχημα μπορεί να βελτιστοποιήσει την ασφάλεια και την αξιοπιστία των συστημάτων του χρησιμοποιώντας τη δυνατότητα του MEC να προσφέρει πληροφορία σε πραγματικό χρόνο. Στο παρακάτω σχήμα φαίνεται συνοπτικά η διαδικασία που μόλις περιεγράφηκε:

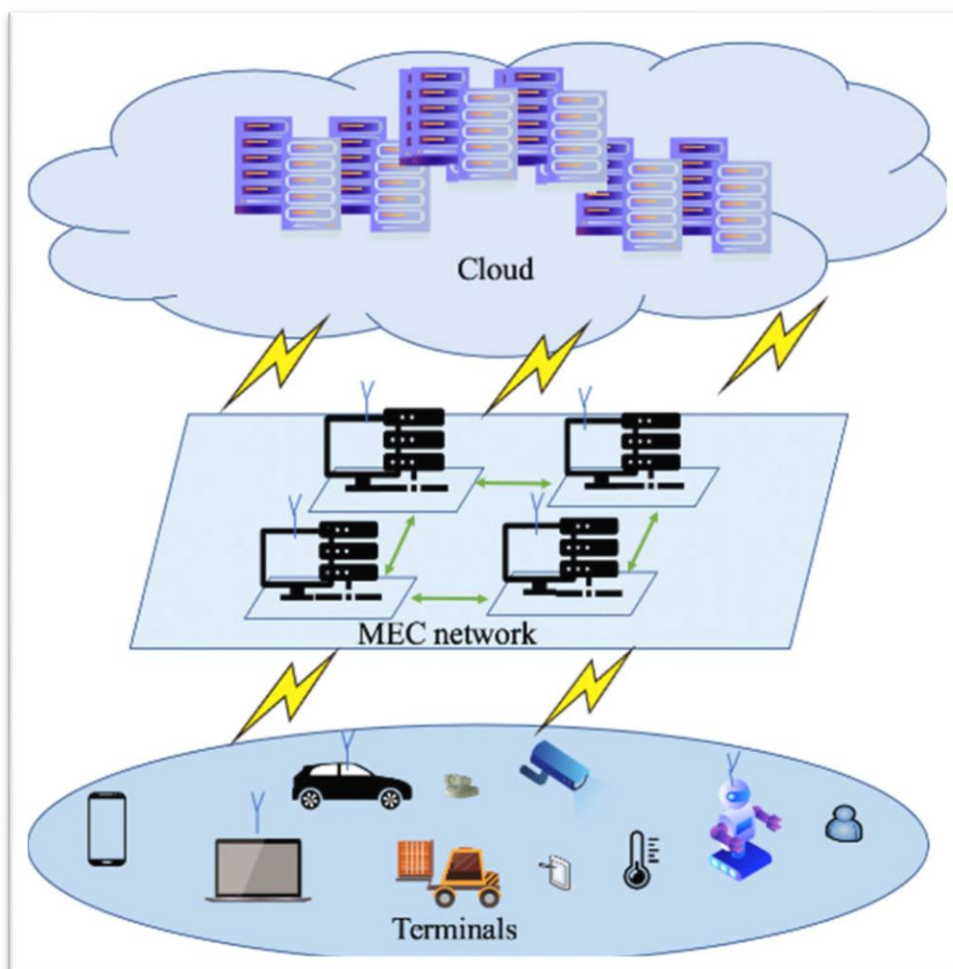


***Εικόνα 1.5: MEC in Interconnected Cars Scenario [4]***

### **1.4.4 Internet of Things**

Ο όρος Internet of Things (*IoT*) αναφέρεται στη διασύνδεση (μέσω του διαδικτύου) πληθώρας συσκευών (λ.χ. οικιακές συσκευές, οχήματα, ηλεκτρονικοί υπολογιστές, κινητά τηλέφωνα). Οι συσκευές αυτές φέρουν κάποιο τύπο λογισμικού ή αισθητήρων, με σκοπό την ανταλλαγή δεδομένων μεταξύ τους. Οι συσκευές αυτές διαθέτουν σε αρκετές περιπτώσεις υπολογιστικούς πόρους περιορισμένων δυνατοτήτων και αποθηκευτικής ικανότητας. Ως συνέπεια η σύνδεσή τους με κάποιον εξωτερικό εξυπηρετητή με τον οποίο θα συνδέονται

μέσω ενός δικτύου ασύρματης μετάδοσης, κρίνεται αναγκαία. Η τεχνολογία MEC προσφέρει αυτούς ακριβώς τους εξυπηρετητές, οι οποίοι βρίσκονται εγκατεστημένοι στους σταθμούς βάσης του LTE δικτύου. Με τον τρόπο αυτό, οι εφαρμογές IoT μπορούν να εκτελούνται άμεσα στους MEC servers αναλύοντας τα δεδομένα που δέχονται από τις διάφορες συσκευές, εκτελώντας τη μεταξύ τους επικοινωνία σε πραγματικό χρόνο, εκμεταλλευόμενες το χαμηλό latency που προσφέρουν οι MEC πλατφόρμες. Η δυναμική αυτή διαδικασία φαίνεται συνοπτικά στο παρακάτω σχεδιάγραμμα:



*Εικόνα 1. : MEC in Internet of Things Scenario [6]*

## ΚΕΦΑΛΑΙΟ 2<sup>ο</sup>

### ΕΦΑΡΜΟΓΗ ΤΟΥ MEC ΣΕ 4G ΚΑΙ 5G ΔΙΚΤΥΑ

#### 2.1 Τεχνολογία LTE

Η τεχνολογία LTE (*Long-Term Evolution*) εκφράζει ένα σύγχρονο πρότυπο ασύρματων συστημάτων επικοινωνίας για κινητές συσκευές και τερματικούς σταθμούς δεδομένων. Αποτελεί ουσιαστικά τον εξελικτικό ακόλουθο των συστημάτων τρίτης γενιάς (3G) και αναπτύχθηκε από τη 3<sup>rd</sup> Generation Project Partnership (3GPP). Η έκδοση 8 της 3GPP που εισήγαγε τις πρώτες προδιαγραφές για το LTE αναγνωρίζεται ως η τεχνολογία 4<sup>ης</sup> Γενιάς (4G), και βασίστηκε στην ολοένα αυξανόμενη ανάγκη των χρηστών για υψηλούς ρυθμούς μετάδοσης δεδομένων σε συνδυασμό με την ελάχιστη δυνατή καθυστέρηση. Επιπλέον, η αρχιτεκτονική του LTE στοχεύει κυρίως στη μεταγωγή πακέτου μέσω δικτύων IP. Οι εκδόσεις 9-14 της 3GPP διεύρυναν ακόμη περισσότερο τις δυνατότητες του δικτύου, ενώ οι εκδόσεις 10-12 ονομάζονται LTE-Advanced. Ακολούθησαν οι εκδόσεις 13 και 14 με το όνομα LTE-Advanced Pro και το 2018 η έκδοση 15 εισάγει την εποχή των νέων δικτύων 5<sup>ης</sup> Γενιάς (5G).

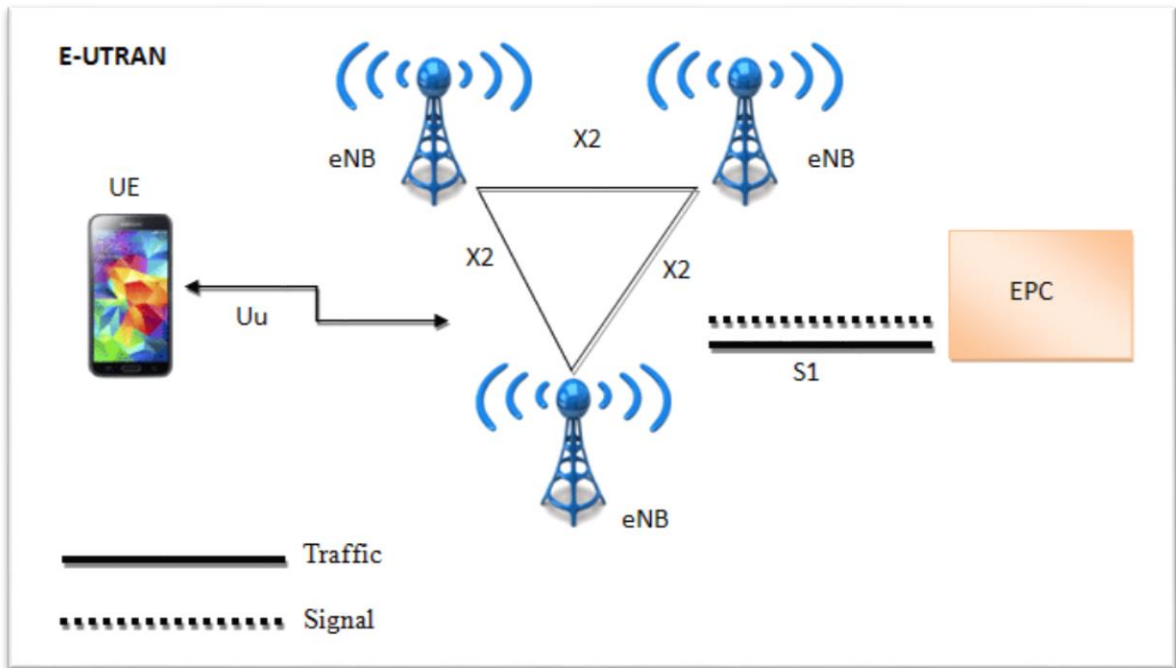
Η αρχιτεκτονική του δικτύου LTE χωρίζεται σε δύο κύρια μέρη, το **δίκτυο ραδιοπρόσβασης** (Radio Access Network, RAN) και το **δίκτυο κορμού** (*Core Network, CN*). Το πρώτο χρησιμοποιείται για τη σύνδεση των τερματικών του δικτύου με τους σταθμούς βάσης, ενώ το δεύτερο για την επικοινωνία των σταθμών αυτών με τα εξωτερικά δίκτυα. Οι σταθμοί βάσης του LTE δεν χρειάζονται κάποιου είδους κεντρική διαχείριση, σε αντίθεση με τα δίκτυα τρίτης γενιάς [9].

### **2.1.1 E-UTRAN**

Στο Σχήμα 2.1 παρουσιάζεται το σχεδιάγραμμα της αρχιτεκτονικής RAN για την τεχνολογία LTE που ονομάζεται **E-UTRAN** (*Evolved UTRAN*) [10]. Αποτελείται από τους σταθμούς βάσης Evolved Node B (*eNB*) και τα αντίστοιχα τερματικά, τα οποία αποτελούν οι χρήστες του δικτύου και ονομάζονται User Equipment (*UE*). Οι χρήστες συνδέονται με κάποιον σταθμό βάσης μέσω της διεπαφής Uu, ενώ οι eNBs επικοινωνούν μεταξύ τους μέσω των διεπαφών X2. Η ανταλλαγή δεδομένων των UEs από και προς το E-UTRAN πραγματοποιείται μέσω αυτών των σημείων σύνδεσης (κατευθύνσεις Uplink, Downlink), καθώς επίσης και η προώθηση πακέτων μεταξύ γειτονικών σταθμών βάσης στις περιπτώσεις μεταπομπής. Η τελική σύνδεση του δικτύου με τον κεντρικό κόμβο του CN γίνεται μέσω της διεπαφής S1.

Δεδομένου ότι οι eNBs του E-UTRAN δεν χρειάζονται κεντρική διαχείριση, γίνεται κατανοητό ότι οφείλουν να εκτελούν ορισμένες διαδικασίες εσωτερικά τους. Οι βασικότερες από αυτές καταγράφονται παρακάτω:

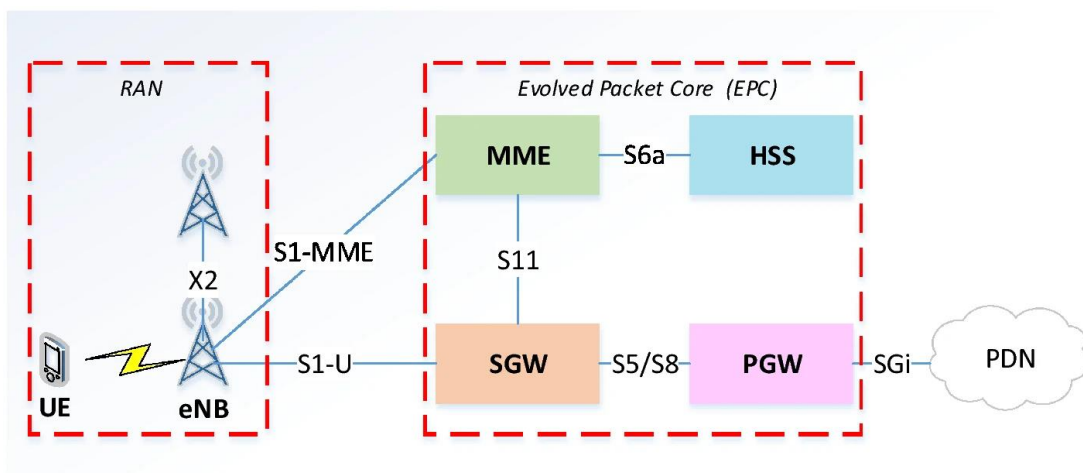
1. Διαχείριση ραδιοπόρων (*Radio Resource Management, RRM*). Η λειτουργία αυτή εμπίπτει σε όλες τις διαδικασίες που πραγματοποιούνται κατά τη σύνδεση μεταξύ δύο σημείων του δικτύου και τη ροή πακέτων στο εσωτερικό της. Οι διαδικασίες αυτές αφορούν τον έλεγχο της κινητικότητας, τον έλεγχο ραδιοπρόσβασης και τη δυναμική κατανομή πόρων από το δίκτυο προς τα UEs.
2. Έλεγχος ασφαλείας, όπως για παράδειγμα η κρυπτογράφηση δεδομένων στις διεπαφές και η συμπίεση και κρυπτογράφηση IP επικεφαλίδων.
3. Επικοινωνία με τον κεντρικό κόμβο του δικτύου. Η λειτουργία αυτή αφορά τις ροές δεδομένων προς τη μονάδα ελέγχου κινητικότητας, καθώς και την τελική σύνδεση μεταξύ των χρηστών και του CN.



Εικόνα 2.1: E-UTRAN Scematic [9]

### 2.1.2 Evolved Packet Core

Στο σχήμα 2.2 απεικονίζεται η αρχιτεκτονική του Core Network για το LTE, το οποίο ονομάζεται **Evolved Packet Core (EPC)**. Αποτελείται από τη **μονάδα διαχείρισης κινητικότητας (Mobility Manager Entity, MME)**, την **πύλη δικτύου πακέτων (Packet Data Network Gateway, P-GW)**, την **πύλη εξυπηρέτησης (Serving Gateway, S-GW)** και τον **οικείο εξυπηρετητή συνδρομητών (Home Subscriber Server, HSS)** [9].



Εικόνα 2.2: RAN connected to EPC [10]

Η **MME** επικοινωνεί με το E-UTRAN μέσω της διεπαφής S1-MME και αποτελεί τη μονάδα ελέγχου της συνδεσιμότητας μεταξύ των UEs και του EPC. Ο κόμβος αυτός διαχειρίζεται τη λειτουργία των κομιστών (bearers), δηλαδή της σύνδεσης μεταξύ των χρηστών και του δικτύου δεδομένων. Επιβλέπει την

εγκατάσταση, τη συντήρηση και την ασφάλειά τους. Επιπλέον, η MME είναι υπεύθυνη για την επιλογή της κατάλληλης S-GW που θα ανατεθεί στην προσπάθεια κάποιου χρήστη είτε να συνδεθεί στο δίκτυο, είτε στην περίπτωση μεταπομπής του. Για τη λειτουργία αυτή η MME χρησιμοποιεί τη διεπαφή S11. Τα πρωτόκολλα που χρησιμοποιούνται για τη σύνδεση των UEs με το EPC ονομάζονται πρωτόκολλα στρώματος μη-πρόσβασης (*Non-Access Stratum, NAS*). Η MME ευθύνεται για την ασφαλή πρόσβαση σε αυτό το στρώμα.

Η **P-GW** χρησιμοποιείται από τον EPC για την επικοινωνία με εξωτερικά δίκτυα IP μέσω της διεπαφής SGi. Βασική της ευθύνη είναι η απόδοση διευθύνσεων και προθεμάτων IP στους χρήστες, επιτρέποντάς τους έτσι την επικοινωνία με το δίκτυο, την εφαρμογή υπηρεσιών QoS και την επιλογή της κατάλληλης πολιτικής χρεώσεων. Επιπλέον, ελέγχοντας την κατεύθυνση DL στους κομιστές, επιλέγει για τον καθένα την κίνηση των IP πακέτων που πραγματοποιείται ανάλογα με το QoS του καθενός. Τέλος, η P-GW αποτελεί το σημείο διασύνδεσης μεταξύ 3GPP και μη-3GPP δικτύων. Η επικοινωνία μεταξύ P-GW/S-GW λαμβάνει χώρα μέσω της διεπαφής S5, εάν οι συσκευές βρίσκονται στο ίδιο δίκτυο ή μέσω τις S8 σε διαφορετική περίπτωση.

Η **S-GW** είναι ο κύριος κόμβος μέσω του οποίου μεταφέρονται όλα τα πακέτα IP που έχουν σαν προορισμό ή σαν προέλευση τους χρήστες. Λειτουργεί σα συνδετικός κρίκος μεταξύ E-UTRAN και EPC και διαχειρίζεται τις περιπτώσεις μεταπομπής μεταξύ δυο γειτονικών eNBs ή εναλλαγής από ένα LTE δίκτυο σε άλλο πρότυπο 3GPP. Επιπλέον, η S-GW αποθηκεύει τα δεδομένα που μεταδίδει ένας UE στην DL διαδρομή όσο η MME οργανώνει την εγκατάσταση των κομιστών εντός του EPC. Η S-GW βοηθά τους παρόχους υπηρεσιών δικτύου στην κατάλληλη χρέωση των χρηστών μετρώντας τον όγκο των δεδομένων στις DL, UL κατευθύνσεις. Συνδέεται με το E-UTRAN χρησιμοποιώντας τη διεπαφή S1-U και με τη MME μέσω της διεπαφής S11.

Ο **HSS** αποτελεί μονάδα αποθήκευσης δεδομένων των χρηστών που εξυπηρετεί ανά πάσα στιγμή το δίκτυο. Οι πληροφορίες που εμπεριέχει μπορεί να είναι το γενικό QoS του χρήστη, καθώς και οι μονάδες MME/P-GW με τις οποίες μπορεί να επικοινωνήσει. Στον HSS βρίσκεται επίσης το κέντρο επαλήθευσης δεδομένων (*Authentication Center, AUC*), που περιλαμβάνει στοιχεία, όπως φορείς επαλήθευσης και κλειδιά ασφαλείας. Συνδέεται και ανταλλάσσει δεδομένα με τη MME χρησιμοποιώντας τη διεπαφή S6a. Σημειώνεται στο σημείο αυτό, πως το EPC περιλαμβάνει και άλλη μια μονάδα αποθήκευσης δεδομένων, την **Policy Control and Charging Rules Function (PCRF)**, στην οποία καταγράφονται τα στοιχεία των UEs που αφορούν τις πολιτικές χρεώσεων.

## 2.2 Mobile Edge Computing σε LTE δίκτυα

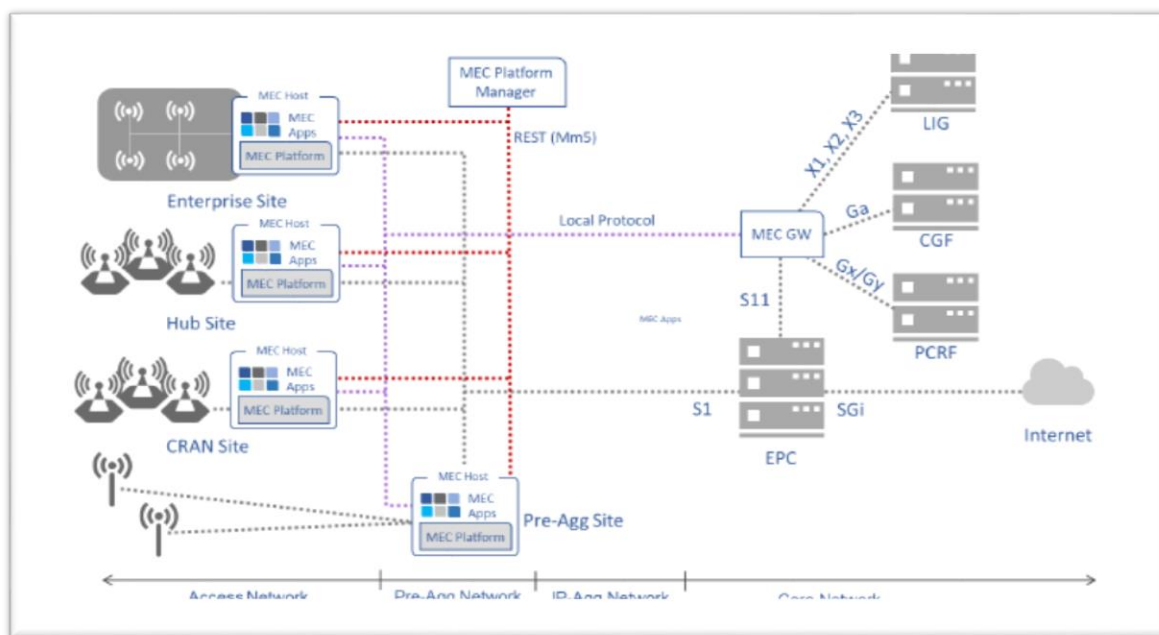
Το μοντέλο της 4<sup>ης</sup> Γενιάς αποδεικνύεται εξαιρετικά επιτυχημένο και αναμένεται να εφαρμόζεται από τους παρόχους υπηρεσιών κινητής τηλεφωνίας ακόμη και στο προσεχές μέλλον, ανεξαρτήτως της ραγδαίας ανάπτυξης των δικτύων 5<sup>ης</sup> Γενιάς. Για το λόγο αυτό, η εφαρμογή της τεχνολογίας του MEC στο ήδη διαδεδομένο LTE πρότυπο που εφαρμόζει το 4G εξετάζεται από τις εταιρείες τηλεπικοινωνιών. Είναι σημαντικό λοιπόν να αναλυθούν μερικές προσεγγίσεις που εξετάζονται για το εν λόγω ζήτημα [11].

### 2.2.1. Bump in the wire

Η έκφραση «Bump in the wire» περικλείει όλα τα σενάρια σύμφωνα με τα οποία η εγκατάσταση της MEC πλατφόρμας βρίσκεται σε τοποθεσίες μεταξύ των σταθμών βάσης του δικτύου και του Core Network. Στο



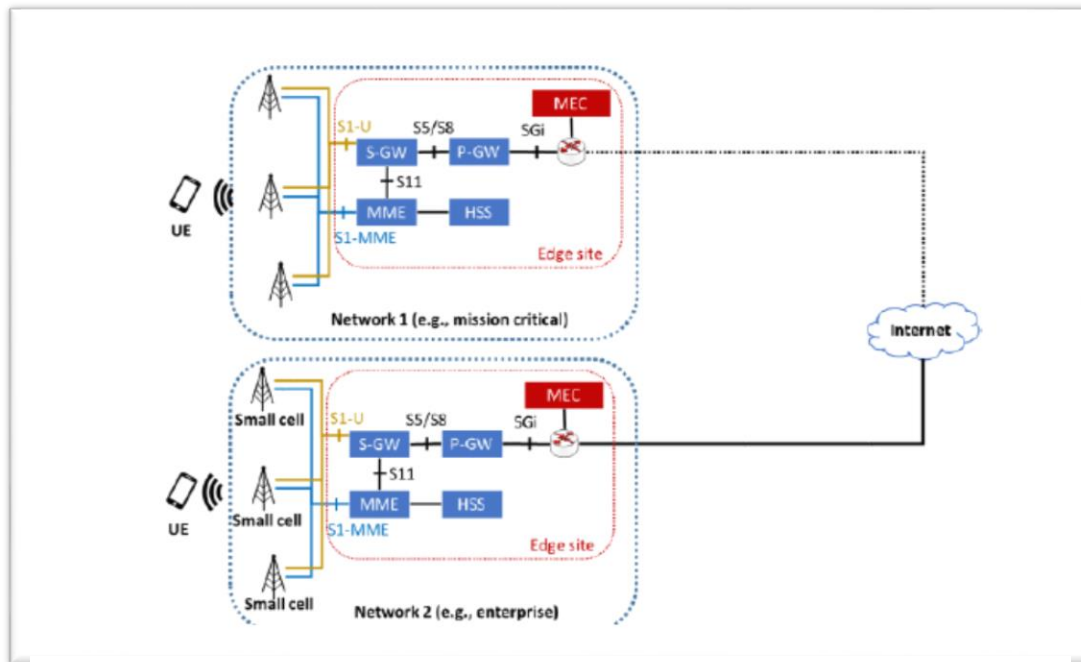
σενάριο κατά το οποίο η πλατφόρμα περιλαμβάνεται εντός των eNBs, δίνεται η δυνατότητα άμεσης διαχείρισης IP πακέτων από και προς τις MEC εφαρμογές μέσω της S-GW. Η πλατφόρμα εγκαθίσταται σε σημείο κοντά στο σταθμό βάσης και παρεμβάλλεται της διεπαφής S1 της LTE αρχιτεκτονικής σε όλες τις υπόλοιπες προσεγγίσεις. Συγκεκριμένα, όταν ο MEC host βρίσκεται στη διαδρομή της διεπαφής S1-U τα πακέτα που διέρχονται μέσω αυτής γίνονται αντιληπτά από το στρώμα δεδομένων του MEC host και μπορούν να γίνουν αντικείμενο επεξεργασίας από τη MEP και τις υπόλοιπες μονάδες της πλατφόρμας, όπως περιεγράφηκαν στο κεφάλαιο 1.1.2. Κατά την προσέγγιση αυτή είναι πιθανό μέρος των δεδομένων να φτάσουν στον EPC ήδη επεξεργασμένα από το MEC host. Επομένως, εγείρεται η ανάγκη τοποθέτησης μιας MEC Gateway (MEC-GW) μεταξύ των δύο όπως φαίνεται στο Σχήμα 2.2. Ο κόμβος αυτός αναλαμβάνει την αποτελεσματική διεκπεραίωση λειτουργιών όπως χρεώσεις και διασφάλιση νόμιμης πρόσβασης. Τέλος, πρέπει να τονιστεί η σημασία της τοπολογίας για την τοποθέτηση του host, καθώς για να διασφαλιστεί η προδιαγραφή του χαμηλού latency ο host πρέπει να τοποθετηθεί όσο το δυνατόν εγγύτερα των eNBs.



**Εικόνα 2.3: "Bump in the wire" Scenario [11]**

### **2.2.2. Κατανεμημένος EPC**

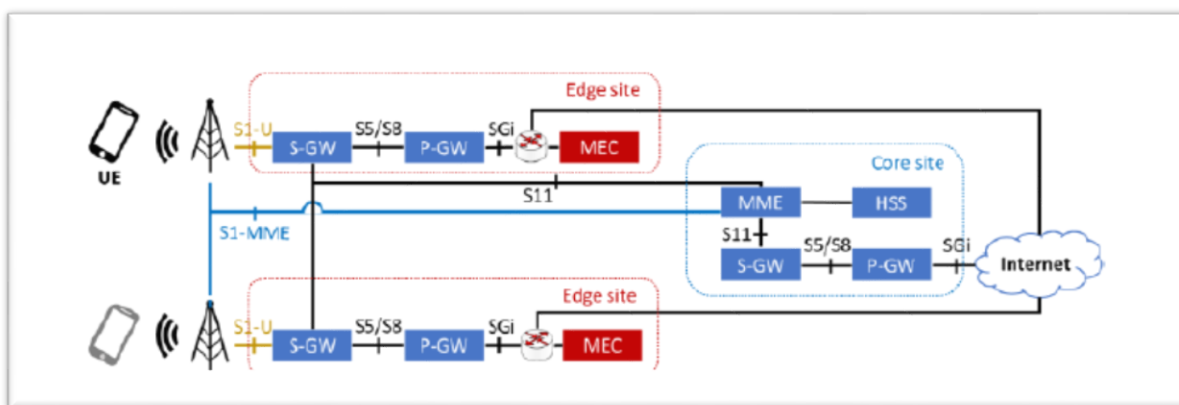
Σε αντίθεση με τη μέθοδο του «bump in the wire», η προσέγγιση του κατανεμημένου EPC βρίσκει το MEC host να περιλαμβάνει όλα ή κάποια από τα επιμέρους συστατικά του EPC, όπως περιεγράφηκε για την αρχιτεκτονική του 4G. Το στρώμα δεδομένων του MEC τοποθετείται στη διαδρομή της διεπαφής SGi, ώστε να διαχειρίζεται την κίνηση που δέχεται η P-GW. Η τοποθέτηση του HSS μπορεί να πραγματοποιηθεί με δύο τρόπους, ο καθένας εκ των οποίων έχει τα δικά του πλεονεκτήματα. Η πρώτη μέθοδος βρίσκει τον HSS να εμπεριέχεται στο MEC host, ώστε να μην απαιτείται σύνδεση με το CN για τη μεταφορά δεδομένων χρηστών, παρέχοντας έτσι ευνοϊκές συνθήκες για εφαρμογές που απαιτούν πολύ χαμηλές καθυστερήσεις. Η δεύτερη αφορά την κλασική αρχιτεκτονική, όπου ο HSS βρίσκεται εντός του CN. Η περίπτωση αυτή μπορεί να υστερεί σε ταχύτητα, υπερσχύει όμως σε θέματα ασφαλείας δεδομένων. Εν γένει, οι κατανεμημένοι EPCs απαιτούν λιγότερες τροποποιήσεις της αρχιτεκτονικής του δικτύου αναλογικά με το bump in the wire, ενώ ταυτόχρονα διευκολύνει τη ροή πακέτων μέσα στο δίκτυο, αφού μέρος του EPC ενθυλακώνεται εντός της MEC πλατφόρμας.



**Εικόνα 2.3: Κατανεμημένος EPC**

### **2.2.3. Κατανεμημένα S-GW/P-GW**

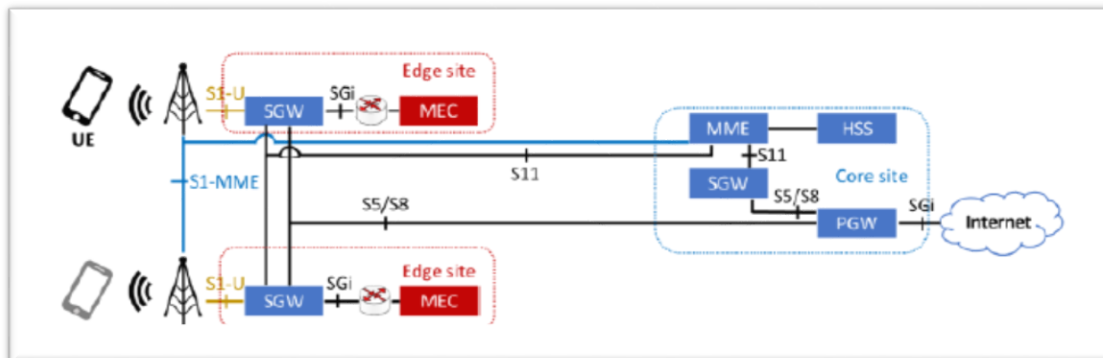
Η προσέγγιση αυτή θεωρείται αρκετά παρόμοια με αυτή που περιγράφηκε στην προηγούμενη ενότητα, με τη διαφορά ότι στην περίπτωση αυτή μόνο τα στοιχεία P-GW/S-GW ενθυλακώνονται στο MEC host. Οι υπόλοιπες μονάδες του LTE δικτύου όπως η MME και το HSS βρίσκονται στο CN, ενώ παρομοίως με πριν, όπως φαίνεται και στο Σχήμα 2.4, η MEC πλατφόρμα επαφίεται στη διεπαφή S-Gi. Ο σκοπός της διατήρησης της MME στο CN είναι η αποδοτικότερη διαχείριση καταστάσεων, όπως για παράδειγμα η μεταπομπή ενός UE.



**Εικόνα 2.4: Κατανεμημένα S/P-GW**

### **2.2.4. Κατανεμημένη S-GW με τοπικό breakout**

Ακόμη μια παρόμοια προσέγγιση, με τη διαφορά ότι στην περίπτωση αυτή μόνο η S-GW επαφίεται στο MEC host, ενώ η P-GW βρίσκεται εντός του CN. Η αρχιτεκτονική αυτή προφέρει τη δυνατότητα στο σύστημα να επιλέξει κατά πόσο η κίνηση δεδομένων από τους χρήστες θα κατευθυνθεί προς τη MEC πλατφόρμα, μέσω του τοπικού συστήματος φυγής (local breakout), ή θα περάσει άμεσα προς την P-GW και ύστερα στο Διαδίκτυο. Η διαδικασία αυτής της επιλογής γίνεται δυναμικά και πάντα με γνώμονα το QoS. Η αρχιτεκτονική της προσέγγισης αυτής φαίνεται στο Σχήμα 2.5.



**Εικόνα 2.5: Αρχιτεκτονική Κατανεμημένου S-GW με τοπικό breakout [11]**

## 2.3 Εφαρμογή MEC σε 5G δίκτυα

Στην προηγούμενη ενότητα περιεγράφηκαν σενάρια κατά τα οποία η τεχνολογία του Mobile Edge Computing εφαρμόζεται ήδη στα LTE δίκτυα 4<sup>ης</sup> Γενιάς. Στις προσεγγίσεις αυτές, η πλατφόρμα του MEC συνδέεται στην αρχιτεκτονική του δικτύου ως μια ξεχωριστή οντότητα, διαθέτοντας όλες τις μονάδες που την καθιστούν αυτοδιαχειριζόμενη στο εσωτερικό της. Στα 5G δίκτυα αντιθέτως, η φιλοσοφία και οι τεχνικές του MEC αποτελούν ένα εγγενές χαρακτηριστικό. Η αρχιτεκτονική του σχεδιάστηκε εξ αρχής με τη λογική του Edge Computing στον πυρήνα του συστήματος, καθώς αναπτύχθηκε για να προσφέρει υπηρεσίες που υπερέχουν σε ταχύτητα, απόδοση και QoS στους χρήστες.

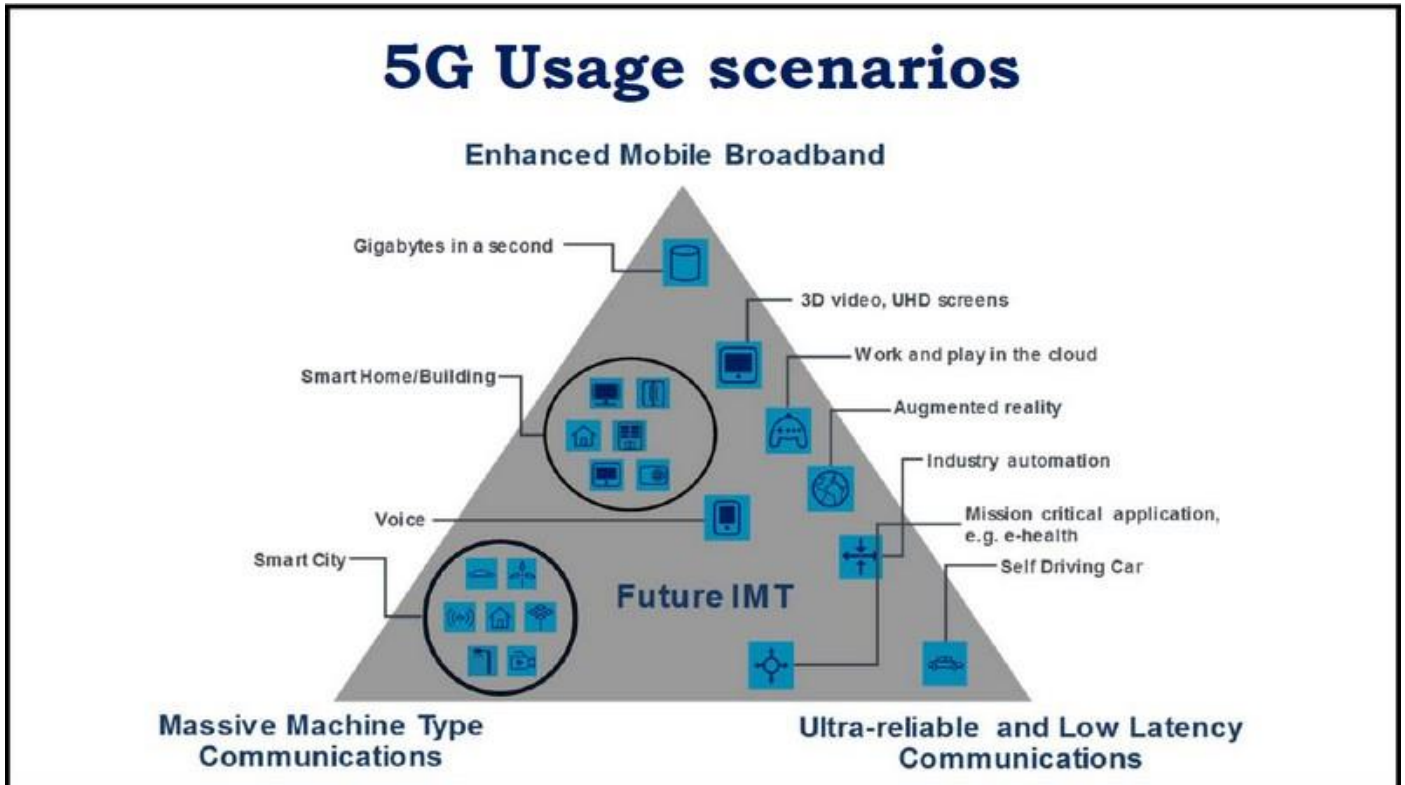
### 2.3.1 Εισαγωγή στα δίκτυα 5<sup>ης</sup> Γενιάς

Τα δίκτυα 5<sup>ης</sup> Γενιάς αποτελούν τη νεότερη έκδοση της 3GPP τεχνολογίας, που διαδέχεται σταδιακά τα 4G LTE δίκτυα. Η εισαγωγή τους έγινε από τη 15<sup>η</sup> έκδοση της 3GPP, με κύριο σκοπό την αναβάθμιση του δικτύου σε συγκεκριμένα χαρακτηριστικά του, όπως ο ρυθμός μετάδοσης δεδομένων, η καθυστέρηση, η πυκνότητα συσκευών, η υψηλή διαθεσιμότητα και η εξοικονόμηση ενέργειας. Σε αντίθεση με το πρότυπο του LTE-Advanced που αποσκοπεί κυρίως στην ευρυζωνική κάλυψη των κινητών συσκευών (mobile broadband), το 5G προσαρμόζεται στις ολοένα αυξανόμενες απαιτήσεις των χρηστών, καθώς προσφέρει ένα περιβάλλον στο οποίο μπορούν να αναπτυχθούν εφαρμογές υψηλών απαιτήσεων. Οι εφαρμογές αυτές μπορούν να κατηγοριοποιηθούν εν γένει στις ακόλουθες τρεις κατηγορίες [12]:

1. **Enhanced Mobile Broadband (eMBB)**. Οι χρήστες απολαμβάνουν αποδοτικότερη πρόσβαση σε δεδομένα και πολυμέσα μέσω της βελτιστοποίησης των ευρυζωνικών εφαρμογών του δικτύου
2. **Ultra-Reliable Low Latency Communications (uRLLC)**. Η επικοινωνία μεταξύ συσκευών στο 5G γίνεται με τη χαμηλότερη δυνατή καθυστέρηση και την υψηλότερη δυνατή αξιοπιστία. Η κατηγορία

αυτή ευνοεί εφαρμογές όπως τα self-driving οχήματα, το video streaming και η επαυξημένη πραγματικότητα.

3. **Massive Machine-Type Communications (mMTC).** Η κατηγορία αυτή αφορά τη δυνατότητα των δικτύων 5<sup>ης</sup> Γενιάς να υποστηρίζουν τη διασύνδεση πολύ μεγάλου αριθμού συσκευών, χωρίς μεγάλες επιπτώσεις στην κατανάλωση ενέργειας και το χρόνο μετάδοσης δεδομένων.



*Εικόνα 2.6: Σενάρια χρήσης 5G [12]*

Η Διεθνής Ένωση Τηλεπικοινωνιών (International Telecommunications Union, ITU) έχει ορίσει συγκεκριμένες προδιαγραφές-στόχους οι οποίοι που παρουσιάζονται στον Πίνακα 1:

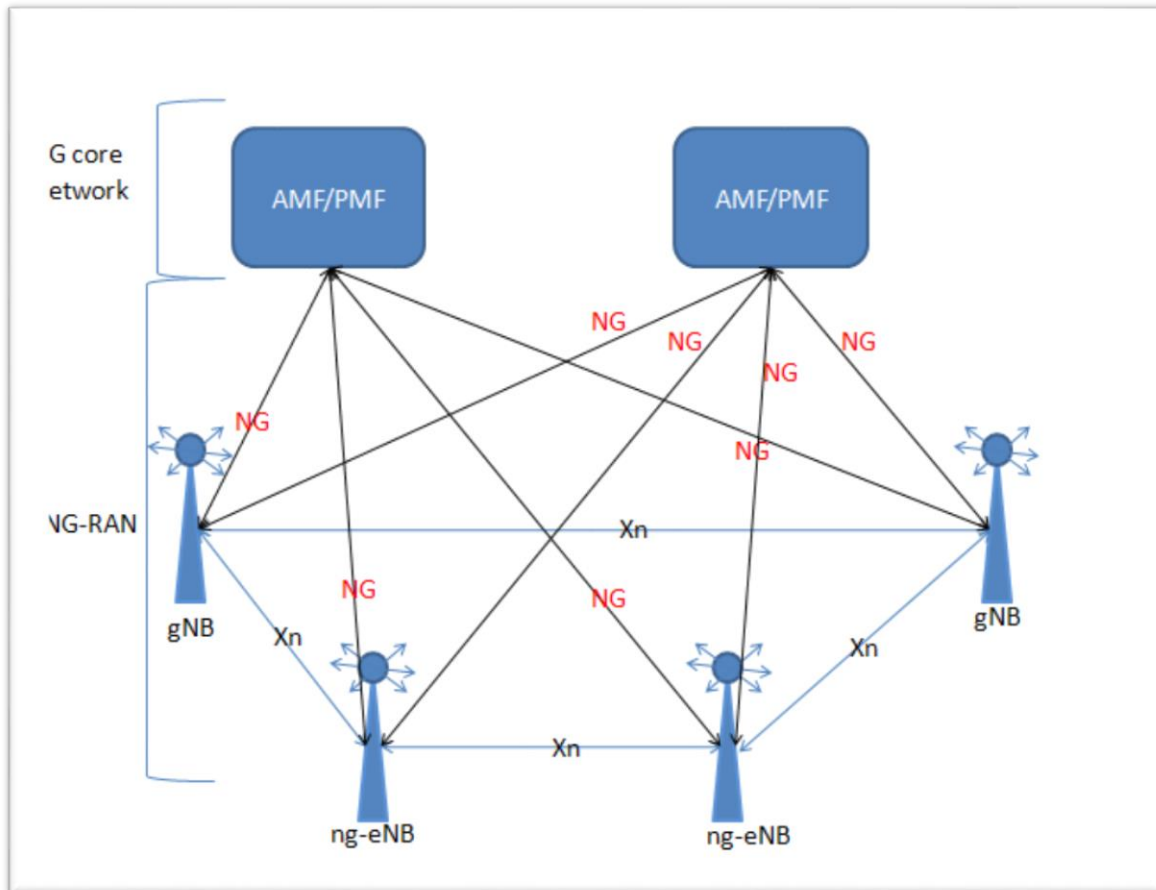
Πίνακας 2.1: Απαιτήσεις δικτύων 5G [13]		
Προδιαγραφή	Τιμή	Μονάδα μέτρησης
Μέγιστος εφικτός ρυθμός μετάδοσης στη DL κατεύθυνση	20	Gbps
Μέσος ρυθμός μετάδοσης που αντλαμβάνονται οι χρήστες	100	Mbps
Φασματική απόδοση	4 φορές αυτή του 4G	Bps/Hz
Καθυστέρηση δικτύου	1	ms
Πυκνότητα συσκευών	$10^6$	Devices/km <sup>2</sup>
Ενεργειακή απόδοση	100 φορές αυτή του αυτή του 4G	Bit/Joule
Πυκνότητα κίνησης	10	Mbps/m <sup>2</sup>
Ταχύτητα διατήρησης επικοινωνίας σε κινούμενα UEs	Έως και 500	km/h

### **2.3.2 Αρχιτεκτονική δικτύων 5<sup>ης</sup> Γενιάς**

Τα δίκτυα 5G ακολουθούν σαν γενικό πλαίσιο τη λογική των προκατόχων τους σχετικά με τα κύρια δομικά τους στοιχεία, καθώς η αρχιτεκτονική τους χωρίζεται σε δύο βασικές μονάδες: το δίκτυο πρόσβασης, που για την περίπτωση του 5G ονομάζεται **Next Generation Radio Access Network (NG-RAN)** και τον πυρήνα του δικτύου ή **5G Core (5GC)**.

#### **2.3.2.(α) Next Generation Radio Access Network**

Το **Next Generation Radio Access Network** αποτελείται από δύο είδη κόμβων, τους **next Generation NodeB (gNB)** και τους **next Generation Evolved NodeB (ng-eNB)** [1]. Οι πρώτοι παρέχουν τερματισμό των πρωτοκόλλων UP (User Plane, στρώμα δεδομένων χρήστη) και CP (Control Plane, στρώμα ελέγχου δικτύου) νέας γενιάς προς τους χρήστες, ενώ οι δεύτεροι είναι υπεύθυνοι για την επικοινωνία με τα πρωτόκολλα του E-UTRAN που αναλύθηκε στα δίκτυα LTE. Το ρόλο των διεπαφών X2 που συναντήθηκε στο LTE αναλαμβάνουν οι διεπαφές Xn, ενώ όλα τα στοιχεία του NG-RAN επικοινωνούν με το 5GC μέσω των διεπαφών NG.



**Εικόνα 2.7: NG-RAN [15]**

Η ενσωμάτωση του NG-RAN στο ήδη υπάρχον δίκτυο αναμένεται να ακολουθήσει δύο στάδια. Αρχικά, οι κόμβοι του θα επικοινωνούν με τον EPC του LTE μέσω των ήδη εγκατεστημένων eNBs, δημιουργώντας μια μη-αυτόνομη έκδοση του 5G. Τελικά, με την εισαγωγή της αυτόνομης έκδοσης οι gNB και ng-eNB θα επικοινωνούν απευθείας με το δίκτυο κορμού του 5G.

### **2.3.2.(β) 5G Core Network**

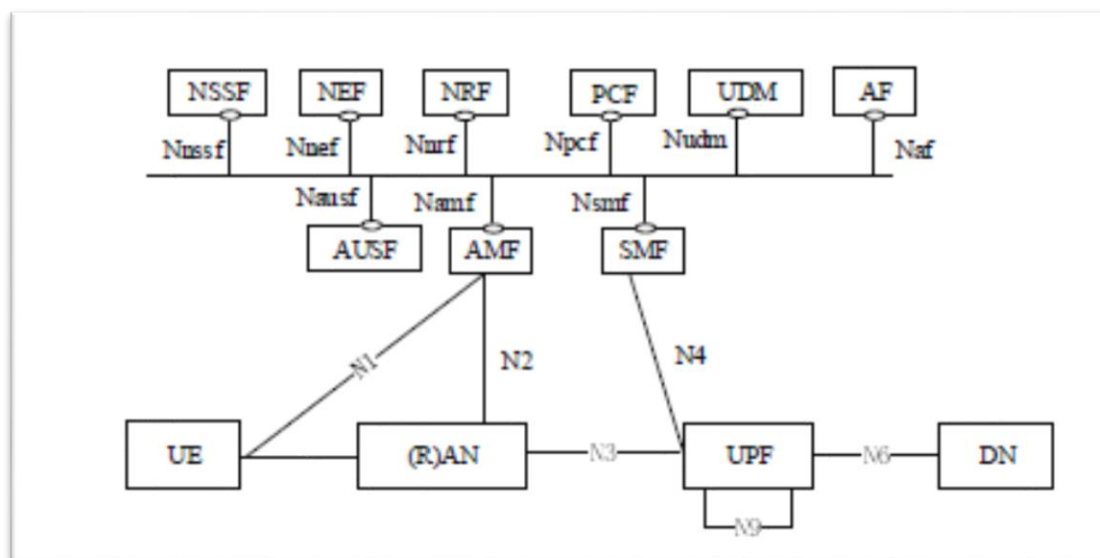
Το δίκτυο κορμού του 5G αξιοποιεί σε μεγάλο βαθμό τις τεχνολογίες **Network Function Virtualization (NFV)** και **Software-Defined Networking (SDN)** [14].

1. Η NFV αναφέρεται στη δυνατότητα δημιουργίας ενός δικτύου, στο οποίο οι λειτουργίες που προσφέρει θα είναι διαχωρισμένες από το φυσικό υλικό (hardware) στο οποίο εκτελούνται. Σε αντίθεση με τα δίκτυα που αποτελούνται από φυσικές οντότητες όπως servers, routers και μονάδες επεξεργασίας, τα δίκτυα βασισμένα σε NFV τρέχουν σε οντότητες λογισμικού που ονομάζονται Virtual Network Functions (VNF). Οι μονάδες αυτές αποτελούνται από Virtual Machines και η συμπεριφορά τους γίνεται αντιληπτή από μια εξωτερική οντότητα που θα συνδεθεί στο δίκτυο ακριβώς με τον ίδιο τρόπο με τις φυσικές οντότητες που αναφέρθηκαν. Τα δίκτυα 5G στοχεύουν στην κλιμάκωσή τους χωρίς την ανάγκη εγκατάστασης φυσικού υλικού χρησιμοποιώντας αυτή την τεχνική, υλοποιώντας έτσι μια οικονομικότερη προσέγγιση τόσο από άποψη χώρου όσο και κόστους.
2. Το SDN [15] αποτελεί μια νέα προσέγγιση στη διαχείριση δικτύων, καθώς χρησιμοποιεί χειριστές βασισμένους σε λογισμικό (software-based controllers) και application programming interfaces (APIs), ώστε να επικοινωνεί με τις φυσικές δομές ενός δικτύου και να αναλαμβάνει τη διαχείριση κίνησης δεδομένων εντός αυτού. Σε αντίθεση με τα παραδοσιακά δίκτυα που χρησιμοποιούν συγκεκριμένες

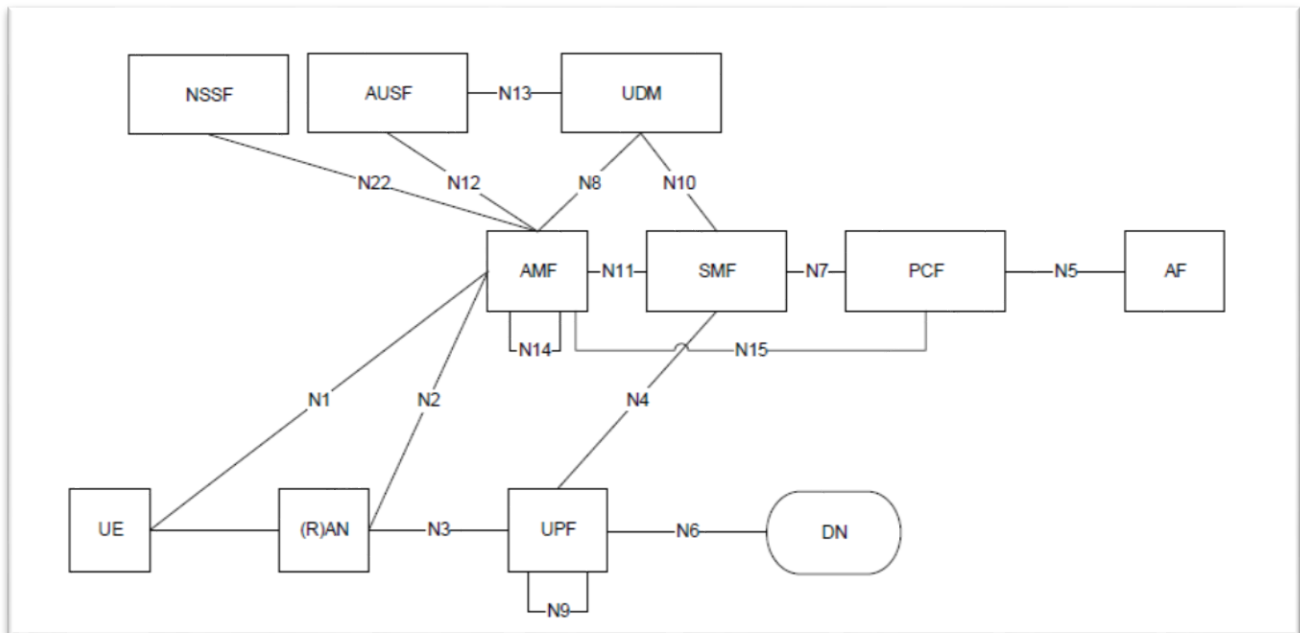
φυσικές δομές, όπως routers και switches για τον έλεγχο διακίνησης δεδομένων, το μοντέλο αυτό δημιουργεί εικονικές δομές μέσω λογισμικού. Με αυτό τον τρόπο η κλιμάκωση και ο έλεγχος του δικτύου απλουστεύονται σημαντικά και προσφέρουν ένα περισσότερο δυναμικό και ευκόλως διαχειρίσιμο περιβάλλον.

Το 5GC αποτελείται από ένα σύνολο NFs που υλοποιούνται ως VNFs και το καθένα αναλαμβάνει συγκεκριμένες λειτουργίες του δικτύου [1]. Το 5GC καταφέρνει με τον τρόπο αυτό να διαχωρίσει το User Plane με το Control Plane, ώστε κάθε ένα να αναπτύσσεται ανεξαρτήτως του άλλου. Επιπλέον, οι λειτουργίες του UP μπορούν να αναπτυχθούν κοντά στο NG-RAN μειώνοντας έτσι το propagation latency και κατά επέκταση την καθυστέρηση του δικτύου.

Η σύνδεση του δικτύου κορμού με το δίκτυο πρόσβασης στο 5G πραγματοποιείται με δύο προσεγγίσεις και αναπτύσσεται με δύο διαφορετικές αρχιτεκτονικές. Στη πρώτη περίπτωση συναντάται η αρχιτεκτονική βασισμένη στις υπηρεσίες (Serviced-Based Architecture, SBA) στην οποία κάθε NF του 5GC συνδέεται με τις υπόλοιπες μέσω μιας και μοναδικής διεπαφής. Αντίθετα, στην περίπτωση της αρχιτεκτονικής βασισμένης στα σημεία αναφοράς (Reference-Point Architecture, PRA) ορίζονται από το δίκτυο κατάλληλες διεπαφές σημείου προς σημείο μεταξύ δύο NFs που επικοινωνούν εντός του. Τα σχεδιαγράμματα των δύο αρχιτεκτονικών παρουσιάζονται παρακάτω στα σχήματα 2.8, 2.9:



**Εικόνα 2.8: Serviced-Based Architecture [1]**



**Εικόνα 2.9: Reference-Point Architecture [1]**

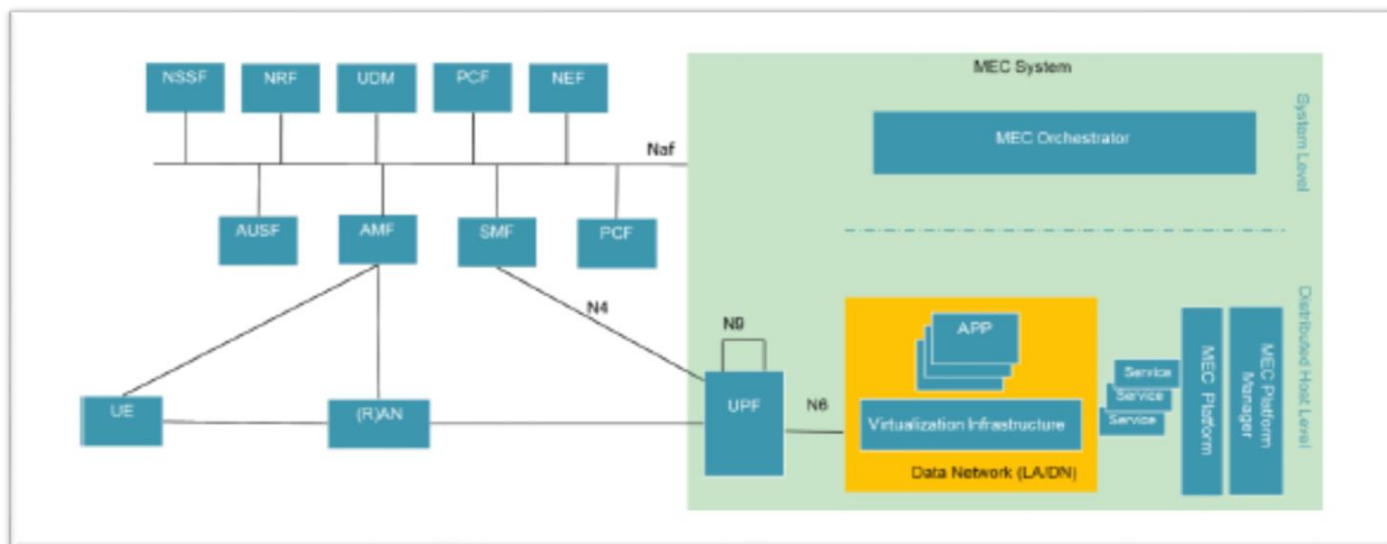
Όπως αναφέρθηκε, το 5GC αποτελείται από επιμέρους NFs [14] [2]. Παρακάτω αναλύονται συντόμως οι σημαντικότερες εξ αυτών:

- **Access and Mobility Management Function (AMF):** Το τερματικό άκρο σύνδεσης του NG-RAN με το 5GC που είναι υπεύθυνο για την εγγραφή χρηστών, διαχείριση κινητικότητας και ασφάλεια πρόσβασης στο δίκτυο.
- **Application Function (AF):** Επικοινωνεί με τις υπόλοιπες NFs με σκοπό την παροχή υπηρεσιών όπως δρομολόγηση και έλεγχος πολιτικών χρέωσης.
- **Authentication Server Function (AUSF):** Αποθηκεύει στοιχεία των χρηστών χρήσιμα σε άλλες μονάδες, αντίστοιχα με τον HSS στα LTE δίκτυα.
- **Location Management Function (LMF):** Μονάδα που χρησιμοποιείται για τον καθορισμό της θέσης ενός UE που συνδέεται στο δίκτυο.
- **Network Exposure Function (NEF):** Παρουσιάζει τις διαθέσιμες λειτουργίες του δικτύου σε άλλες NFs και παρέχει ασφαλή πρόσβαση σε αυτές από μονάδες εκτός του δικτύου 3GPP.
- **NF Repository Function (NRF):** Λειτουργεί ως κόμβος στον οποίο αποστέλλονται αιτήματα ανακάλυψης υπηρεσιών από άλλες NFs, παρέχοντας σε αυτές τις πληροφορίες που αναζητούν.
- **Network Slice Selection Function (NSSF):** Καθορίζει την AMF και τα τμήματα του δικτύου (network slices) που εξυπηρετούν τους χρήστες. Χωρίζει ουσιαστικά το δίκτυο σε επιμέρους εικονικά υποδίκτυα, τα οποία έχουν τις προδιαγραφές να εξυπηρετήσουν μια ή περισσότερες εφαρμογές.
- **Policy Control Function (PCF):** Παρέχει ένα σύνολο κανόνων που ορίζουν τη συμπεριφορά του δικτύου. Για το σκοπό αυτό ενημερώνει τις λειτουργίες του CP ώστε να συμμορφώνονται με τους εν λόγω κανόνες και πολιτικές.
- **Unified Data Management (UDM):** Διαχειρίζεται πληροφορίες σχετικές με τους συνδρομητές. Είναι υπεύθυνη για τη διαχείριση των εγγραφών των UEs σε NFs που τα εξυπηρετούν, την εξουσιοδότηση πρόσβασης σε υπηρεσίες και την αναγνώριση των χρηστών μέσω πιστοποιητικών επιβεβαίωσης.
- **Unified Data Repository (UDR):** Αποτελεί βάση δεδομένων για πληροφορίες που παρέχονται από άλλες NFs.
- **Unstructured Data Storage Function (UDSF):** Βάση αποθήκευσης αδόμητων δεδομένων.



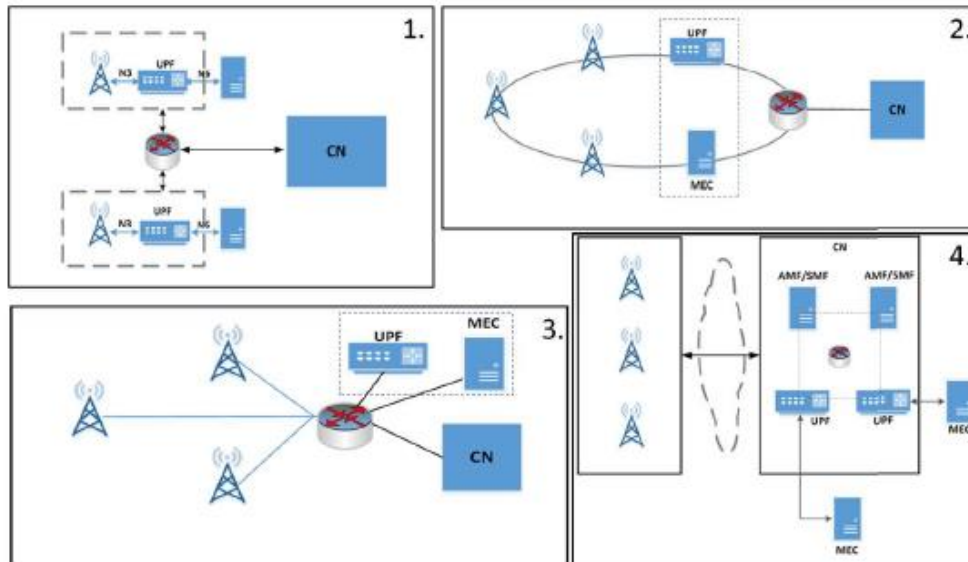
### **2.3.3. Αρχιτεκτονική MEC στα δίκτυα 5<sup>ης</sup> Γενιάς**

Όπως αναφέρθηκε σε προηγούμενη ενότητα, επικρατούν δύο βασικές προσεγγίσεις στην αρχιτεκτονική των συστημάτων MEC [2]. Στο υπόλοιπο της ενότητας 2.3.3. θα αναλυθεί η προσέγγιση της προσαρμογής της MEC πλατφόρμας στη SBA έκδοση της αρχιτεκτονικής των 5G δικτύων. Η σχηματική αναπαράσταση του εν λόγω σεναρίου φαίνεται στο Σχήμα 2.10 που ακολουθεί:



***Εικόνα 2.10: MEC platform in SBA Architecture Scenario [18]***

Στο σύστημα MEC που παρουσιάζεται ανωτέρω, ο Orchestrator αποτελεί τη μονάδα του συστήματος που λειτουργώντας ως AF επικοινωνεί με το NEF ανταλλάσσοντας δεδομένα, ώστε να πραγματοποιείται άμεσα και εύκολα η πρόσβαση σε πληροφορίες ραδιοπρόσβασης από εφαρμογές MEC. Η MEC Platform που συναντάται στο επίπεδο του host αλληλοεπιδρά απευθείας με τις εκάστοτε NFs του δικτύου 5G, ώστε να φέρει εις πέρας κάποια λειτουργία. Όλες οι μονάδες του host level υλοποιούνται στην αρχιτεκτονική του δικτύου ως δίκτυα δεδομένων (data networks). Γενικά, όλες οι NFs μπορούν είτε να παραμείνουν στο δίκτυο κορμού του 5G είτε να υλοποιηθούν εντός των MEC hosts με τη μορφή VNFs. Σχετικά με το ζήτημα της φυσικής τοποθέτησης των MEC hosts στο δίκτυο, οι επιλογές διαφέρουν και καθορίζονται από τις εκάστοτε προδιαγραφές σχετικά με ζητήματα όπως η ασφάλεια και το QoS. Στο Σχήμα 2.11 που ακολουθεί παρουσιάζονται μερικές από τις προσεγγίσεις που εφαρμόζονται:



**Εικόνα 2.11: MEC in 5G possible architectures [2]**

Οι αρχιτεκτονικές που απεικονίζονται είναι οι ακόλουθες:

1. Τοποθέτηση MEC στο σταθμό βάσης
2. Τοποθέτηση MEC στη γραμμή μετάδοσης
3. Τοποθέτηση MEC σε κόμβο συγκέντρωσης του δικτύου
4. Τοποθέτηση MEC εσωτερικά του Core Network

Το φάσμα των επιλογών που προκύπτει, προέρχεται από τη δυνατότητα που προσφέρει η τεχνολογία MEC να εγκαθίσταται σε οποιοδήποτε σημείο μεταξύ σταθμών βάσης και κεντρικού δικτύου. Προφανώς, κάθε αρχιτεκτονική ευνοεί συγκεκριμένες εφαρμογές και η επιλογή της ανατίθεται στις ανάγκες του εκάστοτε παρόχου υπηρεσιών δικτύου.

# ΚΕΦΑΛΑΙΟ 3<sup>ο</sup>

## ΔΡΟΜΟΛΟΓΗΣΗ MOBILE TASKS ΣΕ MEC ΕΞΥΠΗΡΕΤΗΤΕΣ

### 3.1 Εισαγωγή

Όπως αναφέρθηκε σε προηγούμενη ενότητα, παρατηρείται μια ραγδαία εξέλιξη στην ανάπτυξη εφαρμογών του διαδικτύου, στις οποίες απαιτείται η άμεση επεξεργασία δεδομένων και μεταφορά αυτών με τη χαμηλότερη δυνατή καθυστέρηση. Εξαιτίας των φυσικών περιορισμών στην υπολογιστική ισχύ και τον αποθηκευτικό χώρο που χαρακτηρίζουν τις κινητές συσκευές, συγκριτικά πάντα με αντίστοιχες μη φορητές υπολογιστικές μονάδες, η εισαγωγή της τεχνολογίας MEC ως λύση σε αυτό το πρόβλημα υιοθετείται από τέτοιου είδους εφαρμογές.

Η μέθοδος ανάθεσης της εκτέλεσης μιας εφαρμογής σε έναν εξωτερικό φορέα αποτελεί ένα από τα μεγαλύτερα οφέλη της τεχνολογίας MEC και στο πιο γενικό της πλαίσιο συναντάται σε δυο διαφορετικές προσεγγίσεις [16]. Η πρώτη αφορά την εκτέλεση κάθε υπολογιστικής διαδικασίας της εφαρμογής σε κάποιο mobile edge server χρησιμοποιώντας VMs (**application-level offloading**). Σε μια MEC πλατφόρμα, ενώ η τεχνική αυτή χαρακτηρίζεται από απλότητα στον σχεδιασμό και χαμηλή προγραμματιστική πολυπλοκότητα, παράλληλα εισάγει μεγάλο φόρτο εργασίας στους κόμβους της άκρης του δικτύου. Η δεύτερη προσέγγιση (**task-level offloading**) υλοποιείται με τον διαμοιρασμό ξεχωριστών υπολογιστικών διαδικασιών (**computational tasks**) σε διαφορετικούς edge servers, ώστε ο φόρτος εργασίας να διανέμεται με αποδοτικό τρόπο, διευκολύνοντας έτσι τόσο το χρήστη όσο και το δίκτυο. Όπως είναι λογικό, η μέθοδος αυτή βελτιώνει την απόδοση του δικτύου, αλλά εισάγει πολυπλοκότητα στην υλοποίηση και διαχείρισής της.

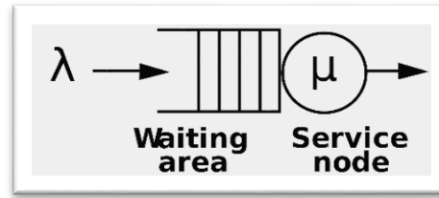
Γίνεται κατανοητό πως η διαχείριση των αιτημάτων εξυπηρέτησης από τα UEs στους εξυπηρετητές MEC παρουσιάζει ιδιαίτερο ερευνητικό ενδιαφέρον. Σε περιπτώσεις που μεγάλος αριθμός από αιτήματα ανάθεσης καταφτάσει προς το σύστημα MEC αυτό οφείλει να προσαρμοστεί, ώστε να ελαχιστοποιήσει το κόστος δρομολόγησης (**scheduling cost**), χωρίς να ελαττώσει σε μη ικανοποιητικά επίπεδα την απόδοση του συστήματος. Μεταξύ των δύο αυτών χαρακτηριστικών (scheduling cost/απόδοση) παρατηρείται το φαινόμενο της αντιστάθμισης (**tradeoff**), καθώς πάντα το ένα βελτιστοποιείται εις βάρος του άλλου. [17]

### 3.2 Μοντέλα προσομοίωσης αφίξεων

Το πρόβλημα δρομολόγησης αιτημάτων σε πολλαπλά διασυνδεδεμένα συστήματα MEC αποτελεί ιδιαίτερη πρόκληση για τον τομέα των τηλεπικοινωνιών. Αρχικά, τα τερματικά των χρηστών κινούνται στην πλειοψηφία τους διαρκώς, μεταβαίνοντας πιθανώς κατά τη διάρκεια της εξυπηρέτησής τους σε διαφορετικά μέσα εξυπηρέτησης. Επιπροσθέτως, τόσο η διαρκής αύξηση των τερματικών που εισέρχονται στο σύστημα όσο και η κλιμάκωση των δικτύων για να τα υποδεχτούν οδηγούν αναπόφευκτα στην αύξηση της πολυπλοκότητας του προβλήματος. Η μοντελοποίηση της διαδικασίας κατά την οποία τα συστήματα MEC δέχονται αιτήματα εξυπηρέτησης και εν συνεχεία τα εκτελούν, μπορεί να πραγματοποιηθεί με αρκετές τεχνικές. Παρακάτω θα παρουσιαστούν συντόμως δυο εξ αυτών, όπως συναντώνται στη σχετική βιβλιογραφία.

#### 3.2.1 Ουρά M/M/1

Οι εξυπηρετητές MEC περιγράφονται με τη μορφή ουράς αναμονής M/M/1 σε αρκετές περιπτώσεις βιβλιογραφίας [18] που αφορά τον τρόπο εξυπηρέτησης των UEs από την MEC πλατφόρμα. Στο συμβολισμό M/M/1, το πρώτο M υποδεικνύει ότι η διαδικασία αφίξεων είναι χωρίς μνήμη (memoryless) και το δεύτερο M ότι οι χρόνοι εξυπηρέτησης είναι επίσης χωρίς μνήμη. Συμβαίνει η μόνη κατανομή χωρίς μνήμη να είναι η εκθετική. Το (1) στο M/M/1 υποδεικνύει ότι η ουρά αναμονής έχει ένα μόνο εξυπηρετητή.



**Εικόνα 3.1: Μοντέλο ουράς M/M/1 2 [3]**

Θεωρείται [19] πως αιτήσεις εξυπηρέτησης από τα UEs καταφθάνουν στον server τις χρονικές στιγμές  $0 < T_1 < T_2 < T_3 < \dots < T_n$ . Τα χρονικά διαστήματα μεταξύ των αφίξεων ( $T_1, T_2 - T_1, T_3 - T_2, \dots$ ) αποτελούν ανεξάρτητες τυχαίες μεταβλητές που ακολουθούν εκθετική κατανομή με παράμετρο  $\lambda$ , δηλαδή:

$$P\{T_{n+1} - T_n \geq t\} = e^{-\lambda t}, \quad t > 0, \quad \lambda > 0 \quad (3.1)$$

Αντίστοιχα, θεωρώντας τις χρονικές στιγμές κατά τις οποίες ο εξυπηρετητής ολοκληρώνει την εκτέλεση του αιτήματος ως  $S_1, S_2, S_3, \dots, S_n$  διαπιστώνεται ότι οι χρόνοι αυτοί ακολουθούν επίσης εκθετική κατανομή, με διαφορετική παράμετρο  $\mu$ , δηλαδή:

$$P\{S_n \geq t\} = e^{-\mu t}, \quad t > 0, \quad \mu > 0 \quad (3.2)$$

Επιπλέον, το σύστημα ακολουθεί την πολιτική First Come - First Served (**FCFS**) που σημαίνει πως ο πρώτος χρονικά χρήστης που θα συνδεθεί με τον εξυπηρετητή, θα δει το αίτημα του να εκτελείται.

Η ποσότητα  $\rho = \frac{\lambda}{\mu}$  ονομάζεται **βαθμός χρησιμοποίησης** και ουσιαστικά εκφράζει την πιθανότητα ο εξυπηρετητής να βρεθεί κατειλημμένος από κάποιο χρήστη που επιθυμεί να συνδεθεί μαζί του. Προφανώς η συνθήκη αυτή είναι αληθής αν και μόνο αν  $\lambda < \mu$ , καθώς στην αντίθετη περίπτωση ο ρυθμός αφίξεων ξεπερνά τον ρυθμό εξυπηρέτησης. Συνεπώς, ο εξυπηρετητής να είναι σε κάθε χρονική στιγμή κατειλημμένος. Σε ένα τέτοιο ενδεχόμενο, η ουρά αναμονής χρηστών προς εξυπηρέτηση μεγαλώνει με το χρόνο χωρίς όριο.

Εάν το σύστημα βρίσκεται σε στατιστική ισορροπία ( $\lambda < \mu$ ), τότε η πιθανότητα να υπάρχει  $n$  αριθμός χρηστών εντός του εξυπηρετητή σε μια στιγμή, δεδομένων των  $\lambda$  και  $\mu$  υπολογίζεται από τον τύπο:

$$P\{n \text{ users in MEC server}\} = (1 - \rho)\rho^n, \quad n \in \mathbb{N} \quad (3.3)$$

Ο **μέσος αριθμός χρηστών** εντός του εξυπηρετητή υπολογίζεται μέσω του τύπου:

$$E[n] = \sum_{n=0}^{\infty} nP\{n \text{ users in MEC server}\} \Rightarrow E[n] = \frac{\rho}{1-\rho} \quad (3.4)$$

Ο **μέσος χρόνος ολοκλήρωσης** ενός αιτήματος στο μοντέλο αυτό υπολογίζεται ως  $\mu^{-1}$ , ενώ ο **μέσος χρόνος αναμονής του χρήστη** για να εξυπηρετηθεί από το σύστημα είναι:

$$T = \frac{E[n]}{\lambda} \Rightarrow T = \frac{1}{\mu(1-\rho)} \quad (3.5)$$

Δεδομένου πως ο συνολικός χρόνος παραμονής αποτελείται από το άθροισμα του μέσου χρόνου αναμονής και του μέσου χρόνου εξυπηρέτησης, τότε ως **μέσο χρόνο αναμονής** εκφράζεται η ποσότητα:

$$W = T - \frac{1}{\mu} \Rightarrow W = \frac{\rho}{\mu(1-\rho)} \quad (3.6)$$

Σύμφωνα με τον νόμο του Little [20], ο μέσος αριθμός χρηστών του συστήματος που αναμένει για να συνδεθεί στον εξυπηρετητή θα είναι:

$$N_W = \lambda W = \frac{\rho^2}{1-\rho} \quad (3.7)$$

Το μοντέλο αυτό είναι προφανές πως περιγράφει μια απλουστευμένη μορφή του προβλήματος της δρομολόγησης mobile tasks σε MEC εξυπηρετητές, καθώς δεν περιέχει μεταβλητές που συναντώνται στις πρακτικές εφαρμογές. Για παράδειγμα, όπως αναφέρθηκε σε προηγούμενη ενότητα, η πλατφόρμα του MEC εμπεριέχει μονάδες όπως ο MEO, οι οποίες επιτρέπουν στους εξυπηρετητές του συστήματος να επικοινωνούν μεταξύ τους και δίνει οδηγίες, ώστε η κατανομή των αιτημάτων να γίνεται όσο πιο αποδοτικά γίνεται. Αν θεωρήσουμε μια μονάδα εξυπηρέτησης ως ουρά M/M/1, τότε σύμφωνα με την σχέση (3.7) όσο ο βαθμός χρησιμοποίησης της πλησιάζει στο 100% ο μέσος αριθμός χρηστών που βρίσκονται σε αναμονή πλησιάζει το άπειρο, αφού  $\lim_{\rho \rightarrow 1} \frac{\rho^2}{1-\rho} = \infty$ . Ένα τέτοιο ενδεχόμενο θα απέδιδε μη ρεαλιστικά δεδομένα. Δεδομένων των μειονεκτημάτων αυτών, εκδόσεις ή παραλλαγές της ουράς M/M/1 συναντώνται ως εργαλεία σε περιπτώσεις όπως αυτή του [18] ώστε να αναπτυχθούν αλγόριθμοι δρομολόγησης με χρήσιμα ερευνητικά αποτελέσματα.

### **3.2.2 Διαδικασία Bernoulli**

Η διαδικασία Bernoulli [21] βασίζεται στο απλούστερο δυνατό πείραμα τύχης που συναντάται στην επιστήμη των πιθανοτήτων, τη **δοκιμή Bernoulli**. Το πείραμα αυτό καταλήγει σε δυο αμοιβαίως αποκλειόμενα δυνατά αποτελέσματα, τα οποία συνήθως μοντελοποιούνται με δυαδικό σύστημα επιτυχίας/αποτυχίας (0 ή 1). Η πιθανότητα επιτυχίας συμβολίζεται με  $p$ , επομένως η πιθανότητα αποτυχίας θα είναι  $1-p$ , με  $0 < p < 1$ . Γενικά, η διαδικασία Bernoulli περιλαμβάνει μια σειρά επαναλαμβανόμενων δοκιμών Bernoulli με πιθανότητα επιτυχίας  $p$  και αποτυχίας  $1-p$ , ανεξαρτήτως του τι συμβαίνει σε προηγούμενες επαναλήψεις. Αν  $X$  η τυχαία μεταβλητή λοιπόν που συμβολίζει τον αριθμό των επιτυχιών σε μια από τις δοκιμές, ισχύει:

$$P[X = x] = p^x(1-p)^{1-x}, \quad x \in \{0,1\} \quad (3.8)$$

Πέρα από το κλασσικό παράδειγμα της ρίψης νομίσματος ως τυχαίο πείραμα Bernoulli, η διαδικασία αυτή χρησιμοποιείται συχνά για τη μοντελοποίηση συστημάτων για αφίξεις σε ένα σύστημα εξυπηρέτησης. Συγκεκριμένα, στην περίπτωση του MEC, το μοντέλο εφαρμόζεται σε περιπτώσεις της βιβλιογραφίας [22], [23] για να περιγράψει τα αιτήματα των UEs προς εξυπηρέτηση στους αντίστοιχους MEC servers. Διαχωρίζοντας το χρόνο σε διακριτές περιόδους, μια άφιξη αιτήματος στην  $k$ -στη επανάληψη είναι συνώνυμη με τη σύνδεση ενός χρήστη στο σύστημα MEC κατά τη διάρκεια της χρονικής περιόδου  $k$ .

Άμεση σύνδεση με τη διαδικασία Bernoulli συναντάται στις τυχαίες μεταβλητές που ακολουθούν τη **δωνυμική κατανομή**. Η κατανομή αυτή με παραμέτρους  $n, p$  υπολογίζει τον αριθμό  $S$  των επιτυχιών σε  $n$  ανεξάρτητες εκτελέσεις δοκιμών και περιγράφεται από τη συνάρτηση μάζας πιθανότητας:

$$p_S(k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad k = 0, 1, 2, \dots, n \quad (3.9)$$

με μέση τιμή  $E[S] = np$  και διασπορά  $Var(S) = np(1-p)$ .

Στην πλειοψηφία των περιπτώσεων η διωνυμική κατανομή χρησιμοποιείται από αλγόριθμους [22], [23] ώστε με δεδομένη την παράμετρο  $p$  να παράγει τυχαίες τιμές από την κατανομή σε συγκεκριμένα χρονικά διαστήματα, προσομοιώνοντας έτσι το trafficking στην είσοδο μιας MEC πλατφόρμας. Το μοντέλο αυτό είναι ιδιαίτερος χρήσιμο σε τέτοιου είδους ερευνητικές προσεγγίσεις, αλλά η πολυπλοκότητά του υστερεί σημαντικά στο να συμπεριλάβει τη δυναμικότητα της διαδικασίας αυτής σε πραγματικές εφαρμογές.

### 3.3 Εισαγωγή στην πρώτη προσέγγιση: “Comparison of scheduling algorithms for multiple mobile computing edge clouds”.

Η πρώτη προσέγγιση που θα αναλυθεί στην παρούσα εργασία [4] βασίζεται σε μια έννοια που έχει ήδη εισαχθεί σε προηγούμενη ενότητα, αυτή του network slicing. Η πλατφόρμα του MEC αναπτύχθηκε για να υποστηρίξει ένα μεγάλο φάσμα εφαρμογών οι οποίες ενώ μπορεί να διαφέρουν ως προς τη φύση των υπηρεσιών που προσφέρουν, χαρακτηρίζονται από παρόμοιες απαιτήσεις σχετικά με τον τρόπο εκτέλεσής τους. Για παράδειγμα, τα διασυνδεδεμένα οχήματα (connected vehicles) και η ζωντανή αναμετάδοση βίντεο (live video streaming services) προσφέρουν δυο κατά βάσει διαφορετικές υπηρεσίες στους εκάστοτε χρήστες τους. Όμως στον πυρήνα της λειτουργίας τους οι δύο αυτές εφαρμογές χαρακτηρίζονται από την ανάγκη για την ελάχιστη δυνατή χρονική απόκριση του συστήματός τους στις αλλαγές των συνθηκών στις οποίες εκτελούνται.

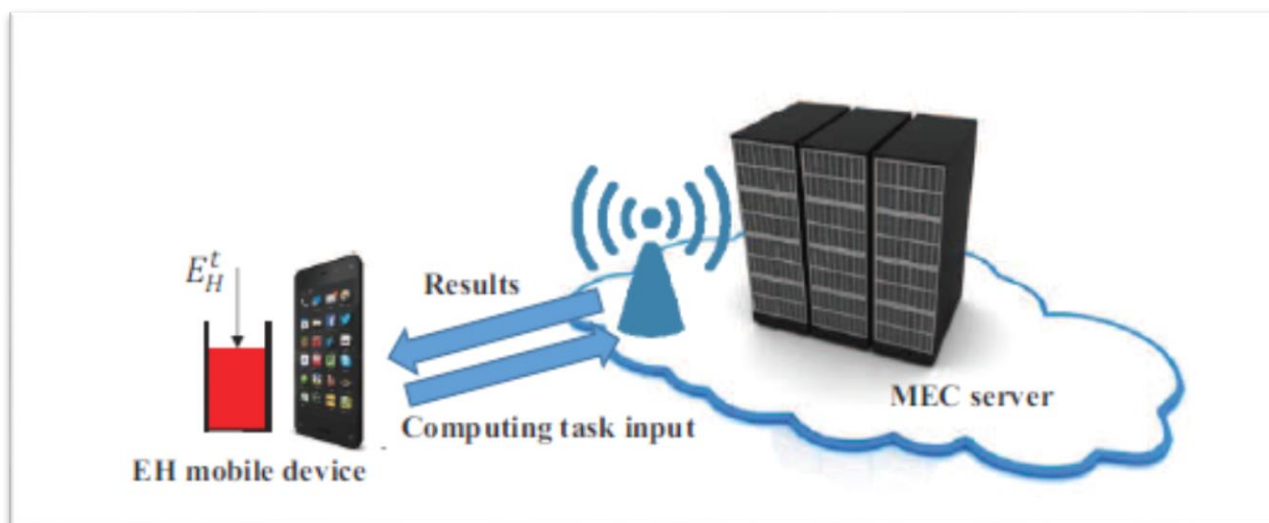
Σε τέτοιες περιπτώσεις η πλατφόρμα του MEC προσαρμόζεται στις ανάγκες των χρηστών χαρτογραφώντας τους κοινούς πόρους που αποτελούν το σύστημα και εν συνεχεία διαχωρίζοντάς τους σε επιμέρους «κομμάτια» (slices). Κάθε ένα από αυτά περιλαμβάνει φυσικούς και εικονικούς πόρους που θα εξυπηρετήσουν την εκτέλεση της εφαρμογής που θα αναλάβουν. Μέσω αυτής της διαδικασίας, οι εφαρμογές που συνδέονται στο δίκτυο μαζί με το αντίστοιχο «κομμάτι» του δικτύου που χρησιμοποιούν, σχηματίζουν ένα σύστημα πολλαπλών MEC clouds τα οποία λειτουργούν βάσει των προδιαγραφών QoS της εκάστοτε εφαρμογής. Ο orchestrator του συστήματος (MEO), όπως περιεγράφηκε στην ενότητα 1.2.3, είναι υπεύθυνος για τη διαχείριση της εκτέλεσης κάθε MEA.

Κατά τη διάρκεια της εκτέλεσης κάθε εφαρμογής παρουσιάζεται το ενδεχόμενο της ανάγκης για μεταπήδηση σε διαφορετικό MEC cloud λόγω μεταβολής των συνθηκών του χρήστη (αλλαγές σε θέση, επιτάχυνση, υψόμετρο κ.λπ.). Στην περίπτωση αυτή, ένας αλγόριθμος δρομολόγησης εκτελείται για να βρεθεί ο κατάλληλος εξυπηρετητής προορισμού. Στην παρούσα εργασία θα σχολιαστούν οι 4 αλγόριθμοι δρομολόγησης που προτείνονται στο [4]: (a) **Random**, (b) **Index**, (c) **Smallest-Occupancy-Level**, (d) **Largest-Free-Slots**.

### 3.4 Εισαγωγή στην δεύτερη προσέγγιση: “Dynamic Computation Offloading for Mobile-Edge Computing with Energy Harvesting Devices”.

Στην περίπτωση του [22] το μοντέλο θα αποτελείται από ένα χρήστη μαζί με τη συσκευή του και ένα MEC server, σε αντίθεση με την πρώτη προσέγγιση. Ο εξυπηρετητής αυτός θεωρείται πως αποτελείται από ένα μικρό κέντρο δεδομένων (data center), το οποίο παρέχεται από κάποιον εξυπηρετητή κινητής τηλεφωνίας. Η συσκευή του χρήστη βρίσκεται σε μια ορισμένη απόσταση από τον server και επικοινωνεί με αυτόν μέσω ενός ασύρματου καναλιού.

Η συσκευή έχει τη δυνατότητα, κατά τη διάρκεια της εκτέλεσης μιας εφαρμογής, να εναποθέσει την εκτέλεση εξ ολοκλήρου στον εξυπηρετητή ο οποίος θα την εκτελέσει μέσω μιας εικονικής μηχανής (VM).



**Εικόνα 3.2: Energy-Harvesting Devices in MEC [26]**

Το μοντέλο στην συγκεκριμένη εργασία λαμβάνει υπόψιν του μια παράμετρο που δεν έχει αναφερθεί εκτενώς στις προηγούμενες ενότητες, αυτή των συσκευών με δυνατότητα συλλογής ενέργειας (**Energy Harvesting Devices, EHD**). Καθώς η υπολογιστική ισχύς που απαιτείται για να υποστηρίξουν οι μοντέρνες κινητές συσκευές τις απαιτήσεις των εφαρμογών τους αυξάνεται διαρκώς, η τεχνολογία των μπαταριών λιθίου ως μέσο αποθήκευσης ενέργειας τείνει να μη συμβαδίζει ανάλογα. οι ανάγκες του μέσου χρήστη καλύπτονται για σχετικά μικρά χρονικά διαστήματα, παρόλο που οι high-end συσκευές της αγοράς πλέον διαθέτουν μπαταρίες χωρητικότητας μεγαλύτερης των 2000 mAh [24].

Η βιομηχανία των κινητών συσκευών στρέφεται όλο και περισσότερο στη λύση των EHD ώστε να υπερκεραστεί το εν λόγω ζήτημα. Η τεχνολογία αυτή, μέσω φορητών συσκευών φόρτισης ή προσθήκη εξοπλισμού στις συσκευές αυτές καθαυτές, εκμεταλλεύεται τις κατάλληλες μεθόδους συλλογής ενέργειας. Στον πίνακα που ακολουθεί δίνονται παραδείγματα αυτών, καθώς και δεδομένα σχετικά με τον τρόπο εκμετάλλευσης και την πυκνότητα ενέργειας που προσφέρουν κατά μέσο όρο:

Πίνακας 3.1: Μέση Πυκνότητα Ενέργειας σε τεχνολογίες συλλογής ενέργειας [5]		
Harvesting technology	Usage Information	Power Density

<i>Photovoltaics</i>	<i>indoors</i>	<b>20 <math>\mu\text{W}/\text{cm}^2</math></b>
<i>Photovoltaics</i>	<i>outdoors at noon</i>	<b>15 <math>\text{mW}/\text{cm}^2</math></b>
<i>Piezoelectric</i>	<i>inserted in shoes</i>	<b>330 <math>\mu\text{W}/\text{cm}^3</math></b>
<i>Thermoelectric</i>	<i>10 °C gradient</i>	<b>40 <math>\mu\text{W}/\text{cm}^3</math></b>
<i>Acoustic noise</i>	<i>100 dB</i>	<b>960 <math>\text{nW}/\text{cm}^3</math></b>

Η συλλογή ενέργειας (**energy harvesting, EH**) στην περίπτωση που εξετάζεται στο [22] μοντελοποιείται ως διαδοχικά πακέτα ενέργειας που καταφθάνουν στη συσκευή και αποθηκεύονται στην μπαταρία.

Ο αλγόριθμος που προτείνεται στη συνέχεια διαθέτει ως βασικές παραμέτρους την άφιξη ενός αιτήματος εκτέλεσης διεργασίας (task request), την ενέργεια που συλλέγεται, τα επίπεδα της μπαταρίας και την υπολογιστική ισχύς που απαιτείται. Στη συνέχεια, λαμβάνοντας υπόψιν έναν αριθμό υπολοίπων παραμέτρων, υπολογίζει το κόστος εκτέλεσης τόσο τοπικά στη συσκευή όσο και στο MEC server. Σε επόμενη ενότητα θα αναλυθεί λεπτομερώς η λειτουργία του αλγορίθμου και τα πειραματικά αποτελέσματα που προκύπτουν.



# ΚΕΦΑΛΑΙΟ 4<sup>ο</sup>

## ΑΛΓΟΡΙΘΜΟΙ ΔΡΟΜΟΛΟΓΗΣΗΣ MOBILE TASKS ΓΙΑ ΠΟΛΛΑΠΛΑ MEC CLOUDS

### 4.1 Εισαγωγή

Όπως αναφέρθηκε περιληπτικά στην ενότητα 3.3, η βασική ιδέα πίσω από την προσέγγιση του [4] είναι ο διαχωρισμός μιας ευρύτερης MEC πλατφόρμας σε τμήματα δικτύου που εξυπηρετούνται από MEC clouds, ενθυλακωμένους σε eNBs ενός 5G δικτύου. Κάθε τέτοιο «νέφος» εμπεριέχει έναν αριθμό από εξυπηρετητές οι οποίοι εκτελούν τα αιτήματα των εκάστοτε χρηστών. Η σύνδεση μεταξύ UEs/servers προσομοιώνεται θεωρώντας πως οι χρήστες είναι ομοιόμορφα κατανεμημένοι στο χώρο που μελετάται παράγοντας αιτήματα σύμφωνα με την κατανομή Poisson ρυθμού  $\lambda$ , ενώ οι servers εξυπηρετούν τα αιτήματα ακολουθώντας εκθετική κατανομή ρυθμού  $\mu$ . Η διαδικασία αυτή αποτελεί ουσιαστικά μια ουρά M/M/n.

Στο σημείο αυτό σημειώνεται πως στο 5<sup>ο</sup> κεφάλαιο του [4], η αριθμητική ανάλυση των αλγορίθμων που προτείνονται πραγματοποιείται με τρεις διαφορετικούς τρόπους σε τρεις διαφορετικές υποενότητες. Οι δυο τελευταίες που αφορούν την επιρροή των χρηστών που κινούνται με όχημα εντός της πόλης (υποενότητα 5.2) και την εφαρμογή της μεθόδου Fractional Guard Channel (FGC) [25] δεν θα αναλυθούν εκτενώς στα πλαίσια αυτής της εργασίας. Αντιθέτως, η προσέγγιση της υποενότητας 5.1 θα παρουσιαστεί εκτενώς στις επόμενες ενότητες.

### 4.2 Παράμετροι του προβλήματος

Οι βασικές μεταβλητές που καθορίζουν τη λειτουργία των τεσσάρων αλγορίθμων που εισάγονται κατηγοριοποιούνται σε 2 μέρη. Το πρώτο αφορά κάθε τοπολογικό χαρακτηριστικό του περιβάλλοντος που επηρεάζει σε μεγάλο βαθμό την επικοινωνία του χρήστη με την πλατφόρμα του MEC, ενώ το δεύτερο περιλαμβάνει τις αριθμητικές παραμέτρους που αφορούν τα χαρακτηριστικά και τα επιμέρους στοιχεία του συστήματος MEC.

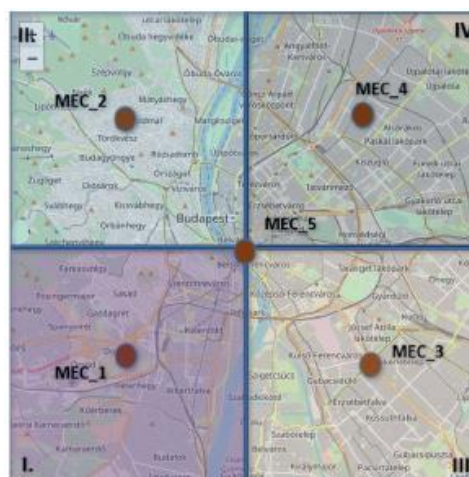
#### 4.2.1 Τοπολογικές συνθήκες

Στις υποθετικές συνθήκες που θέτει το μοντέλο της μελέτης εισάγεται η κάλυψη 5G δικτύου σε ένα συγκεκριμένο τμήμα αστικού ιστού που βρίσκεται στην πόλη της Βουδαπέστης. Η περιοχή κάλυψης έχει εμβαδό 9km x 9km, και παρουσιάζεται στο Σχήμα 21. Θεωρείται επίσης, πως κάθε eNB καλύπτει μια περιοχή εμβαδού 150m x 150m. Επομένως, για πλήρη κάλυψη της εν προκειμένω περιοχής χρειάζονται 3600 eNBs. Σε κάθε ένα από τα σενάρια που θα αναλυθούν το δίκτυο απαρτίζεται από 5 σταθμούς MEC, οι οποίοι τοποθετούνται σε διαφορετικά σημεία, όπως φαίνεται στα Σχήματα 22,23.

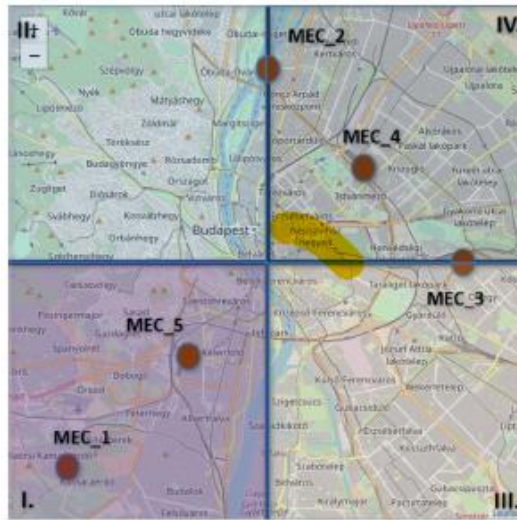


**Εικόνα 4.1: Χάρτης Βουδαπέστης [4]**

Η περιοχή μελέτης χωρίζεται σε 4 ίσα τεταρτημόρια το καθένα από τα οποία καθορίζει μια ξεχωριστή περιοχή. Ξεκινώντας από το νοτιοδυτικό τμήμα και αριθμώντας το ως Περιοχή I, ορίζουμε τα υπολοιπα αντίστοιχα ως Περιοχές II, III και IV ακολουθώντας την ορολογιακή φορά. Οι επιλογή του συγκεκριμένου διαχωρισμού πραγματοποιήθηκε βασισμένη σε στοιχεία όπως η γεωγραφία του εδάφους και η κατανομή του αστικού πληθυσμού.



**Εικόνα 4.2: Σενάριο 1 [4]**



**Εικόνα 4.3: Σενάριο 2 [4]**

Στην πρώτη περίπτωση οι σταθμοί MEC 1-5 φαίνονται ομοιόμορφα κατανομημένοι στο χώρο, καθώς ένας βρίσκεται σημείο τομής των κάθετων αξόνων που ορίζουν τα τεταρτημόρια και οι υπόλοιποι 4 διαμοιράζονται στις 4 περιοχές. Αντίθετα, στο δεύτερο σενάριο οι MEC τοποθετούνται με μεγαλύτερο βαθμό τυχαιότητας, καθώς οι Περιοχές II, III διαθέτουν MEC μόνο στο σύνορό τους.

Η εμβέλεια του κάθε MEC ορίζεται από τις προδιαγραφές ελάχιστης καθυστέρησης που διαθέτει κάθε εφαρμογή, τις συνθήκες κίνησης των χρηστών και την γενική αρχιτεκτονική του δικτύου. Με βάση τα παραπάνω, στον Πίνακα 3 που ακολουθεί περιγράφεται ακριβώς ποιες MEC έχουν τη δυνατότητα να εξυπηρετήσουν τους χρήστες που βρίσκονται σε κάθε περιοχή:

Πίνακας 4.1: Εξυπηρέτηση UEs σε περιοχές συναρτήσει της εμβέλειας των MEC [4]								
Περίπτωση:	Σενάριο 1				Σενάριο 2			
Περιοχές:	I.	II.	III.	IV.	I.	II.	III.	IV.
MEC <sub>1</sub>	X	X	X		X			
MEC <sub>2</sub>	X	X		X		X		X
MEC <sub>3</sub>	X		X	X			X	X
MEC <sub>4</sub>		X	X	X		X	X	X
MEC <sub>5</sub>	X	X	X	X	X	X	X	

Για τις περιπτώσεις που μελετιούνται, κάθε MEC<sub>i</sub> έχει χωρητικότητα 5424 χρηστών, επομένως συνολικά το σύστημα διαθέτει χωρητικότητα 21696 θέσεων. Από τον Πίνακα 3 γίνεται εμφανές πως ενώ εν γένει αρκετοί παράγοντες επηρεάζουν την εμβέλεια κάθε σταθμού, η τοποθέτησή τους με ομοιόμορφο τρόπο σε μια περιοχή κάλυψης όπως π.χ. στο πρώτο σενάριο εξασφαλίζει σε ένα σημείο καλύτερες συνθήκες συνδεσιμότητας για τους χρήστες.

### 4.2.2 Κίνηση χρηστών

Μια βασική παράμετρος του μοντέλου υπό μελέτη αφορά τον τρόπο κίνησης των χρηστών εντός του δικτύου. Αρχικά, θεωρείται πως οι χρήστες βρίσκονται ομοιόμορφα κατανεμημένοι στο χώρο. Στη βιβλιογραφία συναντώνται αρκετά μαθηματικά μοντέλα που χρησιμοποιούνται για την προσομοίωση τυχαίας κίνησης στο χώρο. Από απλές προσεγγίσεις όπως το **Random Walk Mobility Model**, που βασίζεται στην επιλογή τυχαίων ταχυτήτων και αλλαγών κατεύθυνσης, μέχρι πιο περίπλοκα, όπως το **Gauss-Markov Mobility Model (GMM)** [26]. Το GMM εφαρμόζεται από το μοντέλο που μελετάται σε αυτή την ενότητα.

Στο GMM αρχικά ανατίθεται σε κάθε χρήστη μια αρχική ταχύτητα και κατεύθυνση. Στη συνέχεια, με τη βοήθεια της παραμέτρου ελέγχου  $a$  ( $0 \leq a \leq 1$ ), η ταχύτητα και η κατεύθυνση ανανεώνονται σε χρονική στιγμή  $n$  σύμφωνα με τις ακόλουθες εξισώσεις:

$$s_n = a s_{n-1} + (1 - a) \bar{s} + \sqrt{(1 - a^2) s_{x_{n-1}}} \quad (4.1)$$

$$d_n = a d_{n-1} + (1 - a) \bar{d} + \sqrt{(1 - a^2) d_{x_{n-1}}} \quad (4.2)$$

όπου  $\bar{s}$  η μέση ταχύτητα κίνησης, που στην περίπτωση του [4] θα είναι 1.5, 5 και 10 m/s. Τέλος, επιλέγεται η τιμή 0.7 για την ρυθμιστική παράμετρο  $a$ .

### 4.2.3 Συμβολισμοί μεγεθών

Στην υποενότητα αυτή θα εισαχθούν οι συμβολισμοί και οι κωδικοποιήσεις που θα χρησιμοποιηθούν τόσο στην παρουσίαση των αλγορίθμων σε ψευδοκώδικα όσο και στην ανάλυση των αποτελεσμάτων. Στον Πίνακα 4 που ακολουθεί σημειώνονται οι βασικές μεταβλητές του προβλήματος μαζί με την αντίστοιχη κωδικοποίησή τους:

Πίνακας 4.2: Παρουσίαση παραμέτρων προβλήματος [4]		
Μέγεθος	Συμβολισμός	Σχόλιο
Σύνολο MEC clouds που εξυπηρετούν $N$ τύπους MEA <sup>1</sup>	$\Gamma$	Κάθε υποσύνολο MEC που συμμετέχει στο cloud συμβολίζεται ως $\gamma \subseteq \Gamma$
Σύνολο evolved NodeBs (eNodeBs) που συνδέουν τους χρήστες με το 5G δίκτυο	$E$	-
Μέγιστος αριθμός χρηστών που μπορούν να εκτελέσουν την εφαρμογή $n$ , στο MEC $\gamma$	$C_{\gamma,n}$	$n = 1, 2, \dots, N$ και $\gamma \subseteq \Gamma$
eNodeB που προσφέρει συνδεσιμότητα στο UE $e$ τη χρονική στιγμή $t$	$\varepsilon_e(t)$	$\varepsilon_e(t) \in E$
Σύνολο MEC clouds για εφαρμογές τύπου $n$ που εξυπηρετούν UEs συνδεδεμένα στο eNB $\varepsilon_e(t)$	$\Gamma_{n,\varepsilon_e(t)}$	$\Gamma_{n,\varepsilon_e(t)} \subseteq \Gamma$

<sup>1</sup> Σημειώνεται πως  $\sum_{\gamma \in \Gamma} C_{n,\gamma} = 27120$

Χωρητικότητα MEC γ τη χρονική στιγμή $t$	$W_\gamma(t)$	$0 \leq W_\gamma(t) \leq C_\gamma$
Ρυθμιστική παράμετρος GMM	$\alpha$	$0 \leq \alpha \leq 1$
Πιθανότητα απόρριψης task από το σύστημα κατά την εκτέλεσή του	$P_H$	$0 \leq P_H \leq 1$
Πιθανότητα απόρριψης νέας συνεδρίας εκτέλεσης (blocking probability of new sessions)	$P_F$	$0 \leq P_F \leq 1$
Μέση ταχύτητα χρήστη	$\bar{s}$	$\bar{s} \in \{1.5m/s, 5m/s, 10m/s\}$
Ρυθμός Poisson για τις αφίξεις αιτημάτων στο σύστημα	$\lambda$	$\lambda \in [44.85, 45.45]$
Εκθετική παράμετρος ρυθμού εξυπηρέτησης συστήματος	$\mu$	$\frac{1}{\mu} \rightarrow$ μέσος χρόνος εξυπηρέτησης

Με βάση τις παραπάνω παραμέτρους θα εκτελεστούν οι αλγόριθμοι δρομολόγησης αλλάζοντας κάθε φορά τις τιμές των παρακάτω μεταβλητών:

1. Η **απόδοση** (performance)<sup>2</sup> των αλγορίθμων δρομολόγησης.
2. Η **μέση διάρκεια** ενός user session.
3. Η **μέση ταχύτητα χρηστών** συνδεδεμένων στο δίκτυο.

Τέλος, οι συγγραφείς του [4] καθορίζουν στην υποενότητα 5.1 με την οποία ασχολείται το παρόν κεφάλαιο το βαθμό εμπιστοσύνης (confidence level) των αποτελεσμάτων τους στο 99%.

### 4.3 Αλγόριθμοι δρομολόγησης

Οι τέσσερις αλγόριθμοι-κανόνες που προτείνονται στο [4] ονομάζονται **Random**, **Index**, **Smallest-Occupancy-Level (SOL)** και **Largest-Free-Slots (LFS)**. Η λογική στον πυρήνα των τεσσάρων αυτών προσεγγίσεων είναι κοινή και αφορά την επιλογή ενός ελεύθερου MEC ( $\gamma$ , με  $\gamma \subseteq \Gamma$ ) για την αρχικοποίηση ενός νέου user session. Όπως σε κάθε αλγοριθμική διαδικασία, έτσι και εδώ οφείλεται να οριστεί αυστηρά η είσοδος (**Input**), η έξοδος (**Output**) και η αρχικοποίηση μεγεθών (**Initialization**). Στον Πίνακα 5 που ακολουθεί καθορίζονται τα προαναφερθέντα χαρακτηριστικά και για τους 4 αλγορίθμους:

Πίνακας 4.3: Ορισμός κοινών χαρακτηριστικών αλγορίθμων				
Αλγόριθμος:	Random	Index	SOL	LFS
Input	$\Gamma_{to}$	$\Gamma_{to}$	$\Gamma_{to}$	$\Gamma_{to}$
Output	A random $\gamma$	$\gamma$	$\gamma_{to}, bestR$	$\gamma_{to}, bestR$
Initialization 1	$\Gamma_{ret} \leftarrow \emptyset$	$\Gamma_{ret} \leftarrow \emptyset$	$\Gamma_{ret} \leftarrow \emptyset$	$\Gamma_{ret} \leftarrow \emptyset$
Initialization 2	$\gamma_{to} \leftarrow Nil$	$\gamma_{to} \leftarrow Nil$	$\gamma_{to} \leftarrow Nil$	$\gamma_{to} \leftarrow Nil$
Initialization 3	bestR $\leftarrow -1$	bestR $\leftarrow -1$	bestR $\leftarrow -1$	bestR $\leftarrow -1$

<sup>2</sup> Βασισμένη στα  $P_H, P_F$

### 4.3.1 Αλγόριθμος 1: Random Rule

Ο πρώτος αλγόριθμος αποτελεί μια απλοϊκή προσέγγιση του ζητήματος, ίσως την απλούστερη δυνατή που συναντάται. Βασίζεται στη λογική της τυχαίας επιλογής ενός MEC από ένα υποσύνολο αυτών που θα βρεθεί, με την προϋπόθεση η στιγμιαία χωρητικότητα της επιλογής αυτής να μην ισούται ή υπερβαίνει τη μέγιστη δυνατή. Η συνθήκη αυτή εκφράζεται μέσω του if statement **if** ( $W_\gamma(t) < C_\gamma$ ) **then**. Εάν βρεθεί ένα τέτοιο υποσύνολο τότε ο αλγόριθμος επιστρέφει μια τυχαία επιλογή  $\gamma$ . Σε διαφορετική περίπτωση επιστρέφει τον μηδενικό δείκτη Nil εκφράζοντας την απόρριψη του task από το σύστημα. Στο σχήμα που ακολουθεί παρουσιάζεται ο αλγόριθμος σε ψευδοκώδικα:

**Random rule (RND): Choose a random MEC with free slots**

```
1: procedure FIND_SUITABLE_MEC( $\Gamma_{to}$ )
2:   for ( $\gamma \in \Gamma_{to}$ ) do
3:     if ( $W_\gamma(t) < C_\gamma$ ) then                                     ▷ Found a MEC with free slots
4:        $\Gamma_{ret} \leftarrow \Gamma_{ret} \cup \gamma$ 
5:     end if
6:   end for
7:   if ( $\Gamma_{ret} \neq \emptyset$ ) then
8:     return a random MEC from  $\Gamma_{ret}$ 
9:   else
10:    return Nil
11:  end if
12: end procedure
```

*Εικόνα 4.4: Random rule algorithm [4]*

### 4.3.2 Αλγόριθμος 2: Index Rule

Ο δεύτερος αλγόριθμος ακολουθεί αρχικά την λογική υλοποίησης του Random Rule. Στο υποσύνολο MEC που πραγματοποιείται η αναζήτηση, αρκεί να βρεθεί το πρώτο  $\gamma$  που θα διαθέτει ελεύθερες θέσεις. Μόλις λοιπόν ικανοποιηθεί για πρώτη φορά η συνθήκη **if** ( $W_\gamma(t) < C_\gamma$ ) **then**, ο αλγόριθμος τερματίζει επιστρέφοντας το MEC που την ικανοποίησε. Αν όχι, επιστρέφει τον κενό δείκτη. Στο σχήμα που ακολουθεί παρουσιάζεται ο αλγόριθμος σε ψευδοκώδικα:

**Index rule (IND): Choose a first MEC with free slots**

```
13: procedure FIND_SUITABLE_MEC( $\Gamma_{to}$ )
14:   for ( $\gamma \in \Gamma_{to}$ ) do
15:     if ( $W_\gamma(t) < C_\gamma$ ) then                                     ▷ Found a MEC with free slots
16:       return  $\gamma$                                                ▷ Return a first MEC with a free slot
17:     end if
18:   end for
19:   return Nil
20: end procedure
```

*Εικόνα 4.5: Index rule algorithm [4]*

### 4.3.3 Αλγόριθμος 3: SOL Rule

Ο τρίτος αλγόριθμος εισάγει τη μεταβλητή rank σαν μέθοδο ταξινόμησης των λύσεων που αναζητά. Κάθε φορά που ο αλγόριθμος βρίσκει ένα MEC με κενές θέσεις του αναθέτει έναν βαθμό (rank), ο οποίος ποσοτικοποιεί τη διαθεσιμότητά του βάσει των κενών θέσεων που διαθέτει. Το rank κάθε MEC υπολογίζεται από τη σχέση  $rank \leftarrow 1 - W_\gamma(t)/C_\gamma$  η οποία δίνει το ποσοστό στο οποίο ο MEC είναι κατειλημμένος, δεδομένου ότι  $C_\gamma$  είναι η μέγιστη χωρητικότητα του. Για παράδειγμα, αν ένας MEC έχει χωρητικότητα 1000, και 500 από αυτές τις θέσεις έχουν ήδη παραχωρηθεί σε χρήστες, τότε ο τύπος υπολογίζει  $rank = 0.5$  ή 50%. Κάθε φορά που ο αλγόριθμος βρίσκει MEC με καλύτερο rank, ανανεώνει τα στοιχεία που επιστρέφει μέσω των εντολών  $\gamma_{to} \leftarrow \gamma$ ,  $bestR \leftarrow rank$ . Έτσι το τελικό output θα είναι ο MEC  $\gamma_{to}$ , δηλαδή αυτός με το μικρότερο Occurancy Level. Στο σχήμα που ακολουθεί παρουσιάζεται ο αλγόριθμος σε ψευδοκώδικα:

**Smallest-Occupancy-Level rule (SOL): Find a MEC with the smallest session occupancy ratio**

```
21: procedure FIND_SUITABLE_MEC( $\Gamma_{to}$ )
22:   for ( $\gamma \in \Gamma_{to}$ ) do
23:     if ( $W_\gamma(t) < C_\gamma$ ) then                                     ▷ Found a MEC with free slots
24:        $rank \leftarrow 1 - W_\gamma(t)/C_\gamma$                              ▷ Rank based on the occupancy ratio
25:       if ( $rank > bestR$ ) then
26:          $\gamma_{to} \leftarrow \gamma$ ,  $bestR \leftarrow rank$ 
27:       end if
28:     end if
29:   end for
30:   return  $\gamma_{to}$ 
31: end procedure
```

Εικόνα 4.6: SOL rule algorithm [4]

### 4.3.4 Αλγόριθμος 4: LFS Rule

Ο τέταρτος και τελευταίος αλγόριθμος χρησιμοποιεί επίσης τη μεταβλητή rank για να κατηγοριοποιήσει MEC με ελεύθερες θέσεις. Η μέθοδος ταξινόμησης που εισάγεται σε αυτή την περίπτωση αφορά τη στιγμιαία χωρητικότητα κάθε MEC που εξετάζεται. Το rank υπολογίζεται ως η διαφορά της μέγιστης χωρητικότητας MEC με τη στιγμιαία ( $rank \leftarrow C_\gamma - W_\gamma(t)$ ). Κάθε φορά που ο αλγόριθμος βρίσκει ένα  $\gamma$  με μεγαλύτερο rank από το αμέσως προηγούμενο που υπολογίστηκε ( $rank > bestR$ ) αντικαθιστά το τελικό output με το νέο, καταλληλότερο  $\gamma_{to}$  ( $\gamma_{to} \leftarrow \gamma$ ,  $bestR \leftarrow rank$ ). Έτσι η τελική έξοδος του αλγορίθμου θα είναι το  $\gamma$  με τον μεγαλύτερο αριθμό κενών θέσεων εξυπηρέτησης. Στο σχήμα που ακολουθεί παρουσιάζεται ο αλγόριθμος σε ψευδοκώδικα:

**Largest-Free-Slots rule (LFS): Find a MEC with the largest number of free slots**

```
32: procedure FIND_SUITABLE_MEC( $\Gamma_{to}$ )
33:   for ( $\gamma \in \Gamma_{to}$ ) do
34:     if ( $W_\gamma(t) < C_\gamma$ ) then
35:        $rank \leftarrow C_\gamma - W_\gamma(t)$ 
36:       if ( $rank > bestR$ ) then
37:          $\gamma_{to} \leftarrow \gamma, bestR \leftarrow rank$ 
38:       end if
39:     end if
40:   end for
41:   return  $\gamma_{to}$ 
42: end procedure
```

▷ Found a MEC with free slots  
▷ Rank by number of free slots

*Εικόνα 4.5: Index rule algorithm [4]*

## 4.4 Εφαρμογή Αλγορίθμων και παρουσίαση αποτελεσμάτων

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα της εκτέλεσης των τεσσάρων αλγορίθμων που μόλις παρουσιάστηκαν. Τα μεγέθη που αποτελούν το μέτρο αξιολόγησης της μελέτης είναι δύο μέτρα πιθανότητας. Το πρώτο αφορά την πιθανότητα η σύνδεση του χρήστη για την εκτέλεση μιας εφαρμογής να απορριφθεί από το σύστημα και ονομάζεται **blocking probability** ( $P_F$ ). Το δεύτερο μέγεθος περιγράφει την πιθανότητα μια σύνδεση που εκτελείται σε κάποιο MEC server να διακοπεί από το σύστημα πριν την ολοκλήρωσή της και ονομάζεται **dropping probability** ( $P_H$ ). Οι δύο αυτές πιθανότητες υπολογίζονται για κάθε αλγόριθμο συναρτήσει:

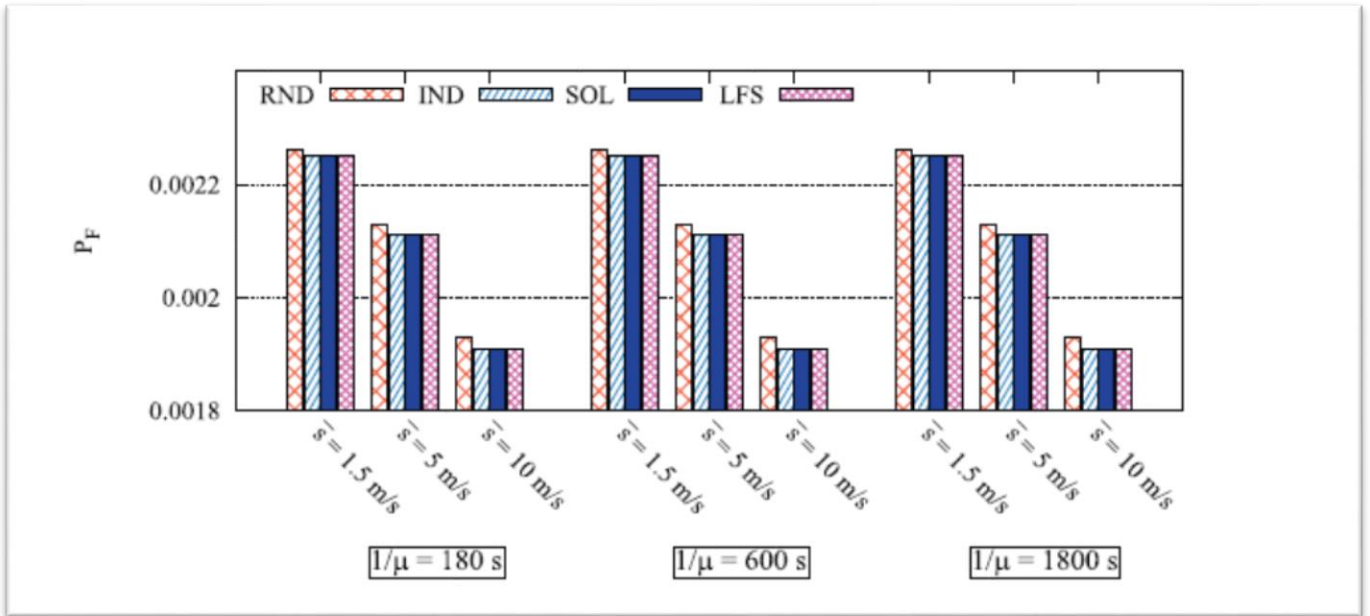
1. Του μέσου χρόνου εξυπηρέτησης  $\frac{1}{\mu}$
2. Του ρυθμού άφιξης αιτημάτων  $\lambda$
3. Των τριών μέσων ταχυτήτων χρήστη  $\bar{s} \in \{1.5m/s, 5m/s, 10m/s\}$

Οι υπολογισμοί γίνονται ξεχωριστά για κάθε ένα από τα δύο σενάρια. Σημειώνεται στο σημείο αυτό πως στα πλαίσια της παρούσας διπλωματικής εργασίας τα αποτελέσματα θα αναπαραχθούν όπως παρουσιάζονται στο [4], και θα σχολιαστούν σε επόμενη ενότητα.

### 4.4.1 Σενάριο 1

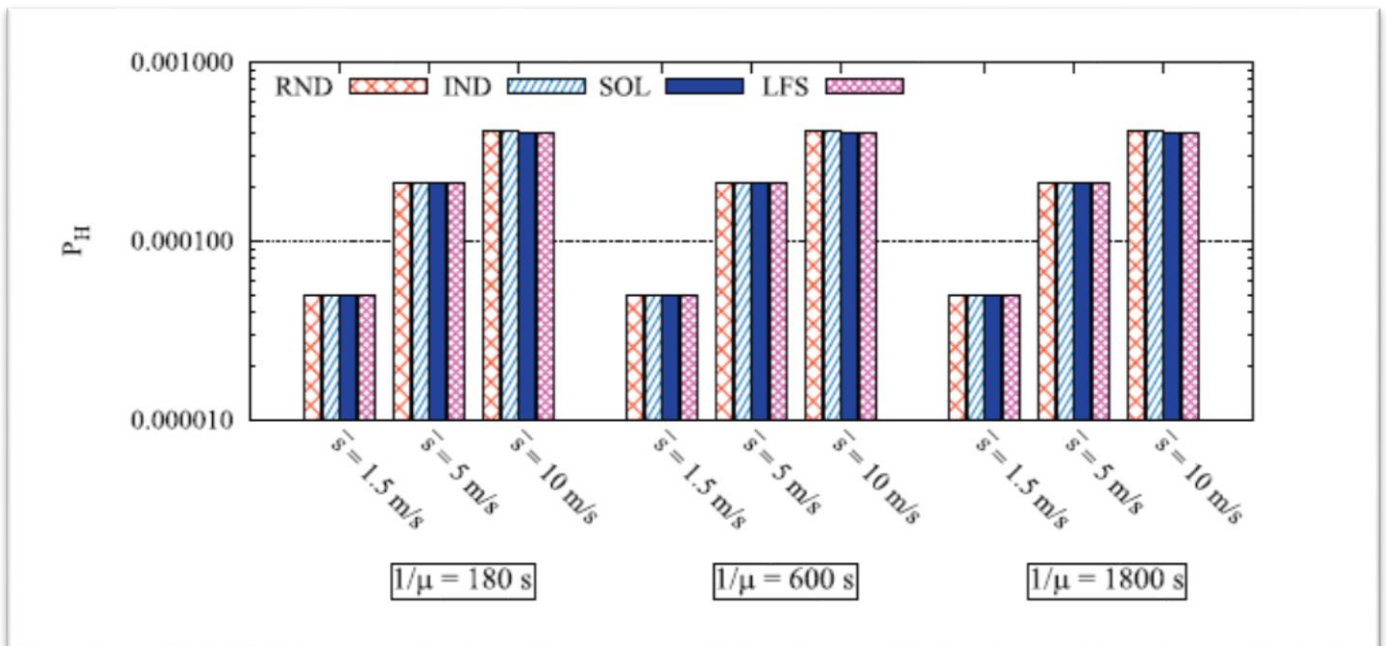
Το πρώτο ραβδόγραμμα που παρατίθεται αφορά τον υπολογισμό της **blocking probability**  $P_F$  σε κάθε αλγόριθμο για τρεις διαφορετικές τιμές του  $\frac{1}{\mu}$  οι οποίες θέτονται στις τιμές **180s**, **600s**, και **1800s**. Για κάθε μια εξ αυτών το  $P_F$  υπολογίζεται για 3 διαφορετικές τιμές του  $\bar{s}$  οι οποίες είναι **1.5m/s**, **5m/s** και **10m/s** αντίστοιχα:





**Διάγραμμα 4.1: Blocking Probability for new sessions [4] (Σενάριο 1)**

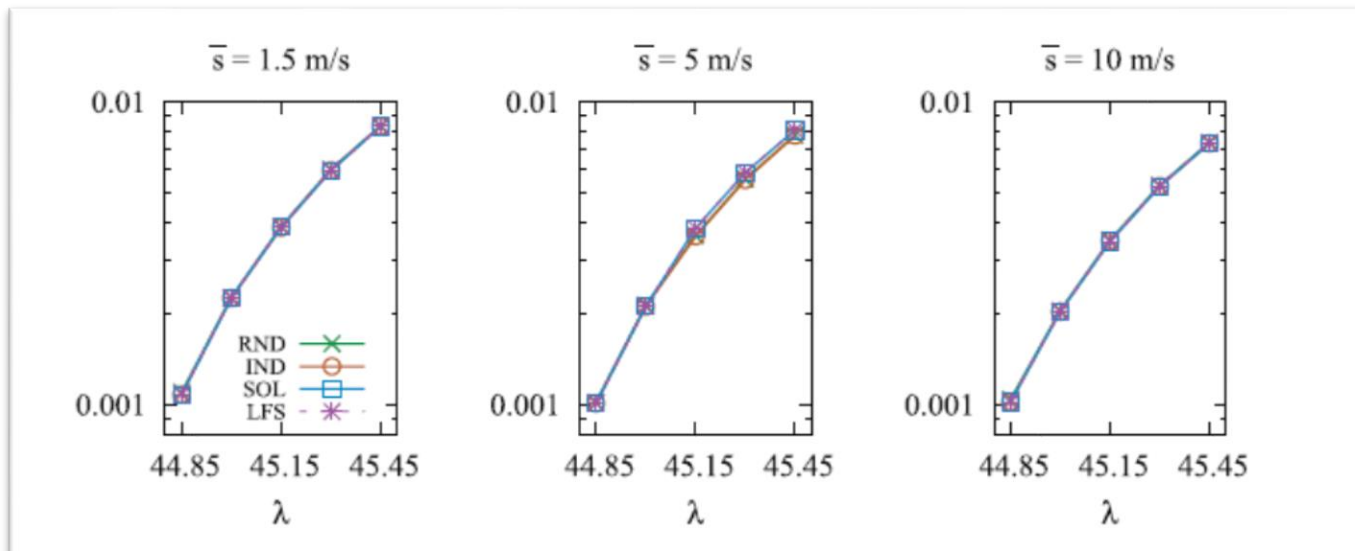
Στο επόμενο ραβδόγραμμα παρουσιάζεται με ακριβώς την ίδια λογική ο υπολογισμός της **dropping probability  $P_H$** :



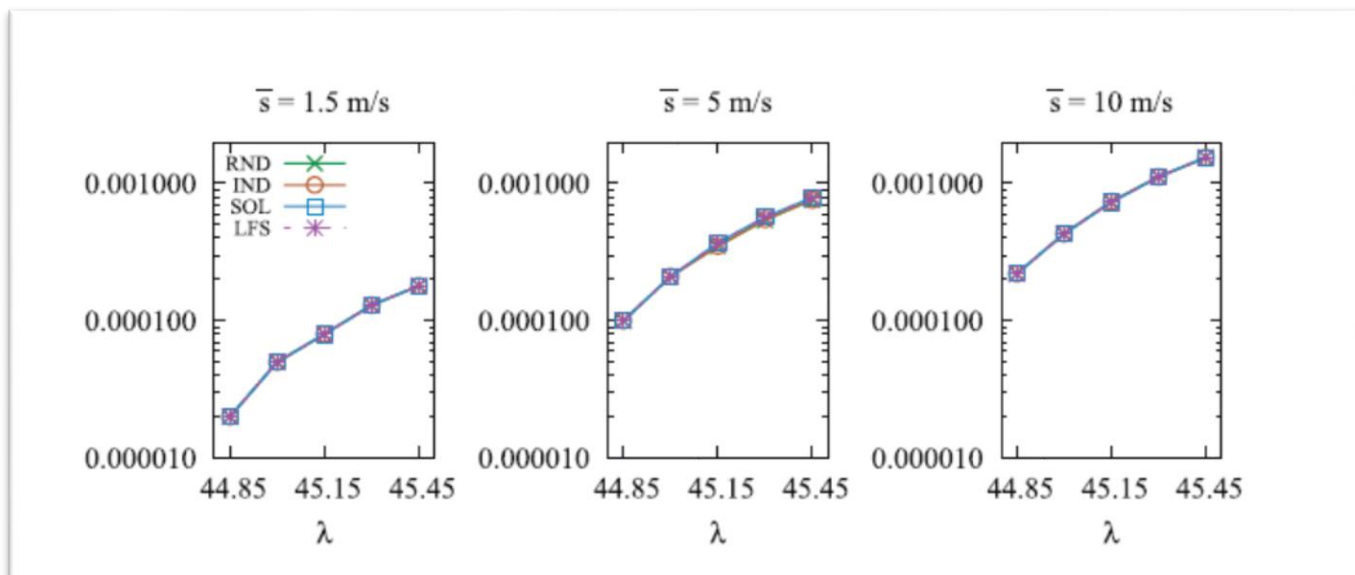
**Διάγραμμα 4.2: Dropping Probability for new sessions [4] (Σενάριο 1)**

Σημειώνεται πως και για τα δύο ανωτέρω Διαγράμματα ισχύει η αναλογία  $\frac{\lambda}{\mu} = 27000$ . Επομένως για κάθε μέσο χρόνο εξυπηρέτησης  $\frac{1}{\mu}$  ο ρυθμός άφιξης αιτημάτων προς τους εξυπηρετητές **παραμένει σταθερός**.

Στη συνέχεια, οι ίδιες ποσότητες  $P_H, P_F$  υπολογίζονται για κάθε αλγόριθμο συναρτήσει του ρυθμού άφιξης νέων αιτημάτων εξυπηρέτησης  $\lambda$ , ο οποίος παίρνει όλες τις τιμές του διαστήματος  $[44.85, 45.45]$ . Για αυτό το εύρος τιμών οι αλγόριθμοι εκτελούνται ξεχωριστά για κάθε μια από τις τιμές που λαμβάνει η μέση ταχύτητα χρήστη. Τα Διαγράμματα των  $P_H, P_F$  παρατίθενται παρακάτω:



**Διάγραμμα 4.3: Blocking Probability for new sessions [4] (Σενάριο 1)**

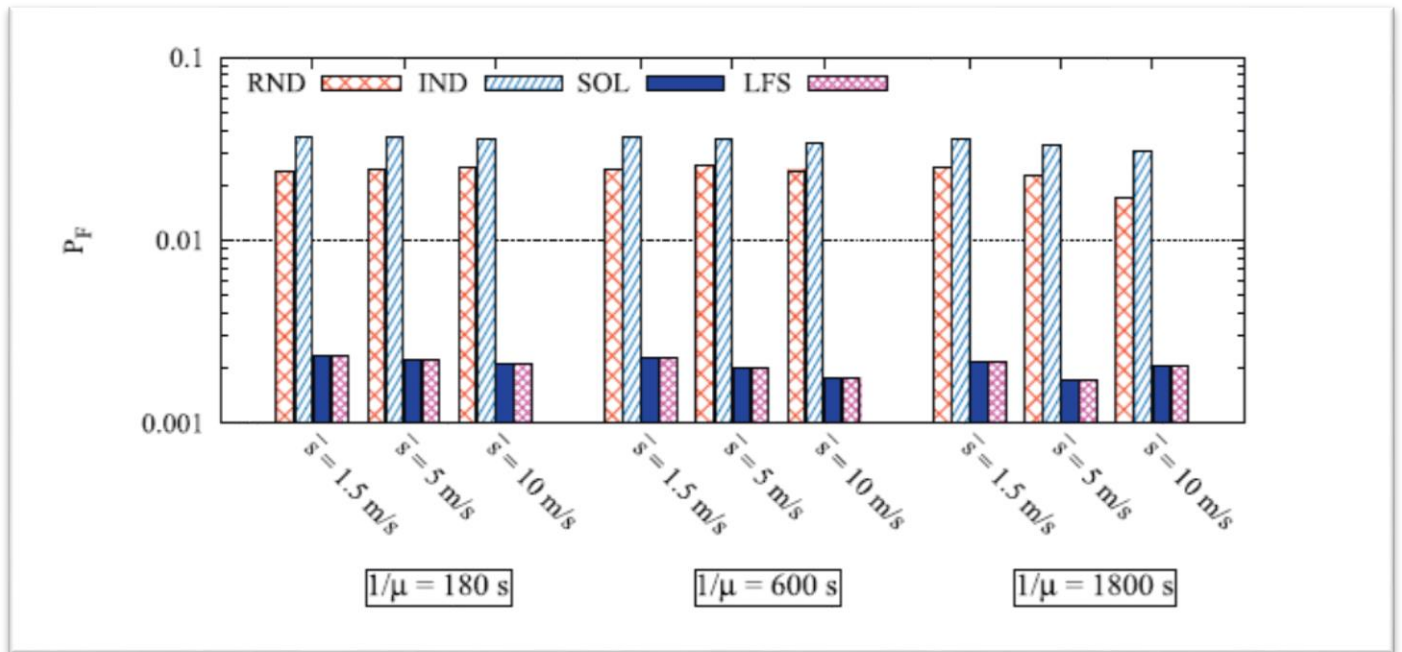


**Διάγραμμα 4.4: Dropping Probability for new sessions [4] (Σενάριο 1)**

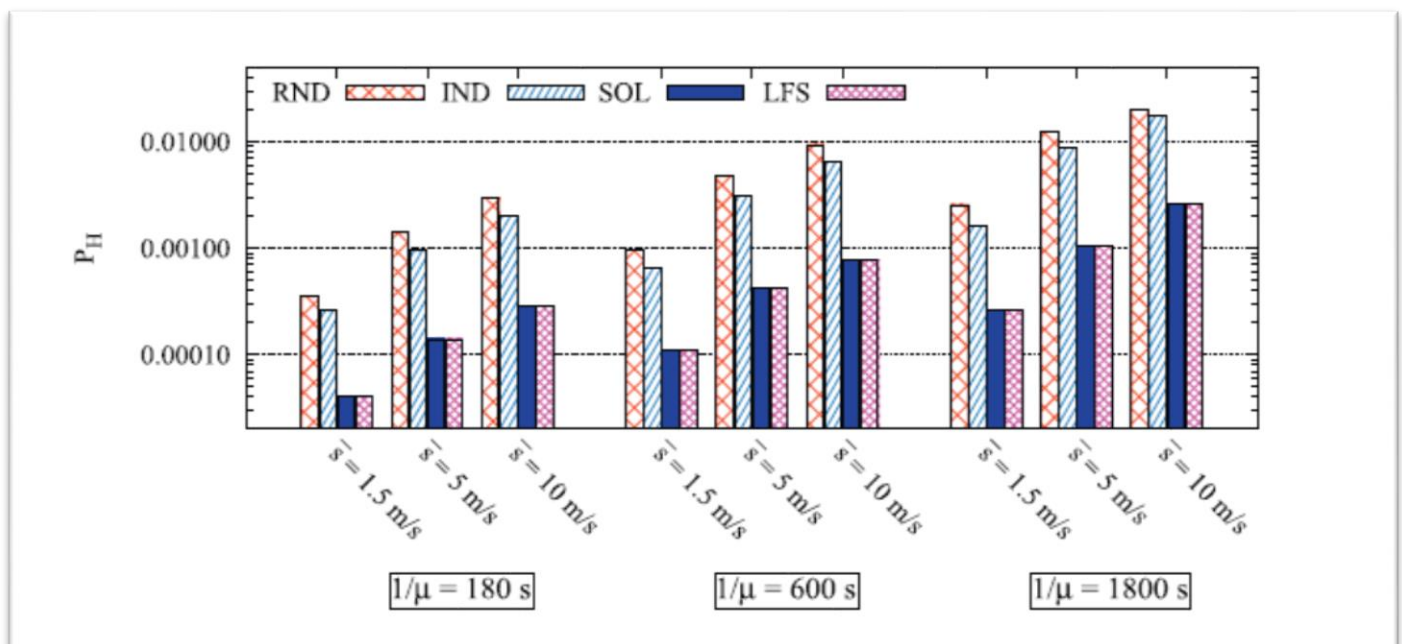
Στα Διαγράμματα 4.3, 4.4 ο μέσος χρόνος εξυπηρέτησης θεωρείται σταθερός και ίσος με 600s για κάθε εκτέλεση των αλγορίθμων.

#### 4.4.2 Σενάριο 2

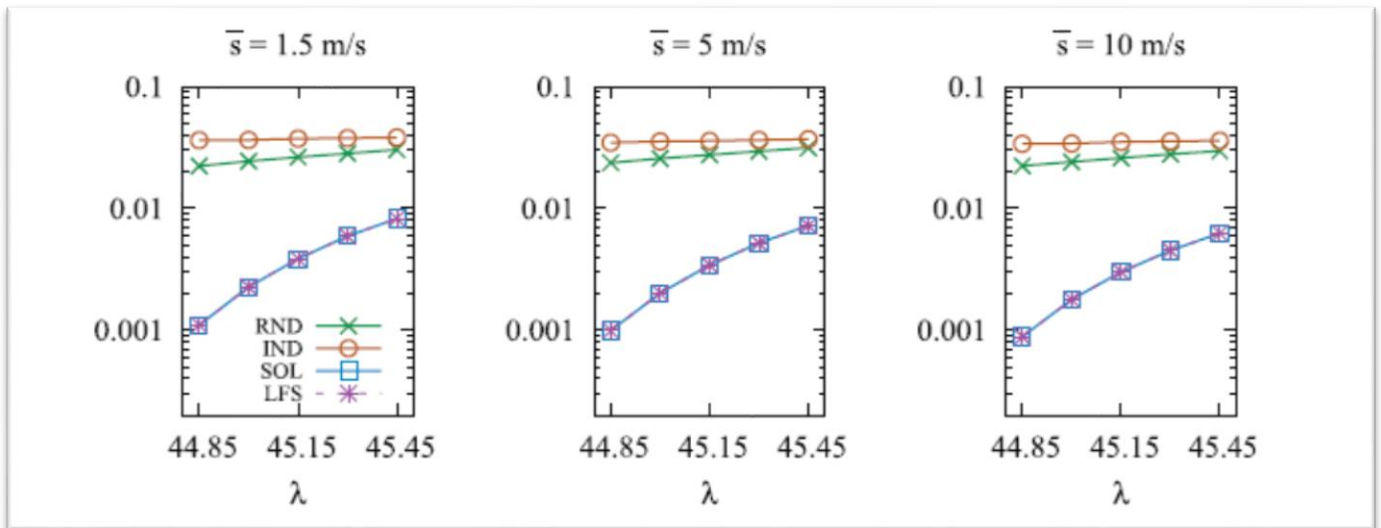
Στην περίπτωση του Σεναρίου 2 όσο αφορά τον τρόπο εκτέλεσης των αλγορίθμων και παραγωγής των αποτελεσμάτων οι συνθήκες και οι προϋποθέσεις είναι ακριβώς ίδιες με αυτές που περιεγράφηκαν την υποενότητα 4.4.1. Η μόνη ειδοποιός διαφορά έγκειται στην τοπολογία των MEC, η οποία παρουσιάστηκε αναλυτικά στην υποενότητα 4.2.1. Τα αποτελέσματα που παράχθηκαν με αυτή τη μεθοδολογία παρουσιάζονται παρακάτω στα Διαγράμματα 4.5, 4.6, 4.7 και 4.8:



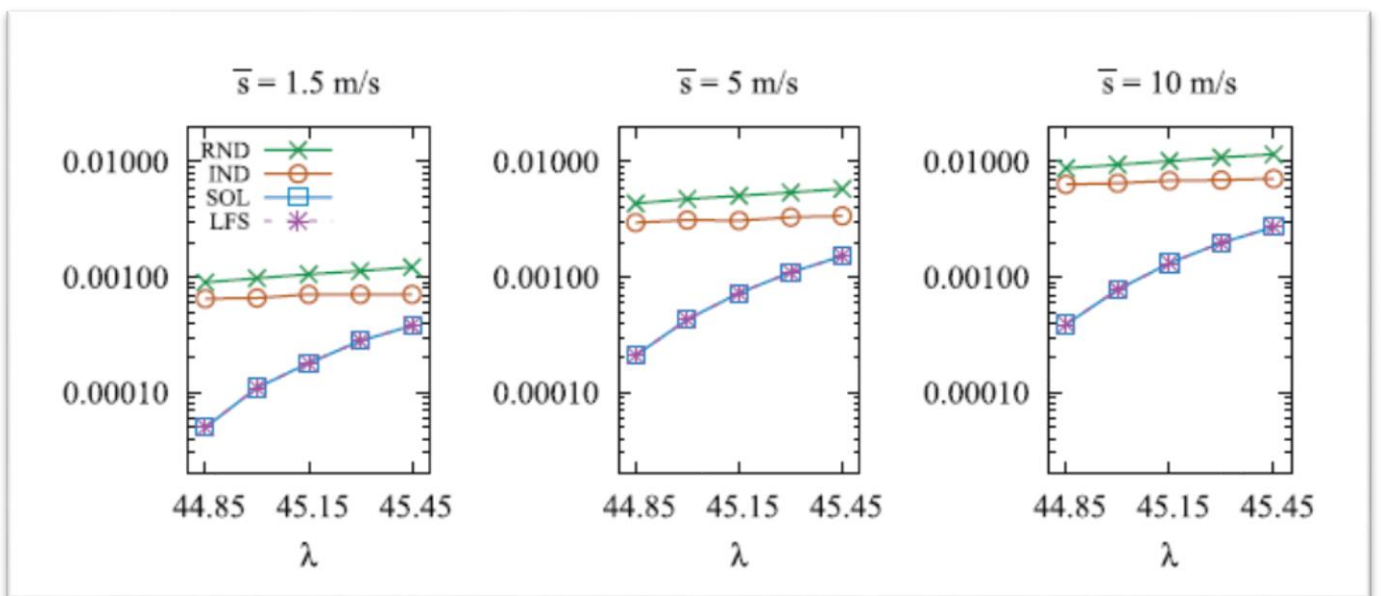
Διάγραμμα 4.5: Blocking Probability for new sessions [4] (Σενάριο 2)



Διάγραμμα 4.6: Dropping Probability for new sessions [4] (Σενάριο 2)



**Διάγραμμα 4.7: Blocking Probability for new sessions [4] (Σενάριο 2)**



**Διάγραμμα 4.8: Dropping Probability for new sessions [4] (Σενάριο 2)**

## 4.5 Σχολιασμός Αποτελεσμάτων

### 4.5.1 Σενάριο 1

Στο πρώτο Σενάριο τα αποτελέσματα και στις τέσσερις διαφορετικές εκδοχές εκτέλεσης των αλγορίθμων (Διαγράμματα 4.1, 4.2, 4.3 και 4.4) χαρακτηρίζονται από έναν κοινό παρονομαστή, τη σχεδόν πανομοιότυπη απόδοσή τους. Ιδιαίτερη αναφορά αξίζει στα διαγράμματα που αφορούν την απόδοση των αλγορίθμων συναρτήσει του μέσου χρόνου εξυπηρέτησης, καθώς φαίνεται πως οι τιμές των  $P_H, P_F$  είναι ανεξάρτητες του  $\frac{1}{\mu}$ . Το φαινόμενο αυτό έχει ως πιθανή εξήγηση την τοπολογία του συστήματος. Στα πλαίσια του πρώτου σεναρίου οι MEC τοποθετούνται σχεδόν ομοιόμορφα στο χάρτη του Σχήματος 21. Αν αυτή η παρατήρηση συνδυαστεί με το γεγονός ότι οι χρήστες βρίσκονται ομοιόμορφα κατανεμημένοι στο χώρο, είναι λογικό η σύνδεση με το σύστημα να συναντά ευνοϊκές συνθήκες ανεξαρτήτως μεθοδολογίας. Επιπλέον, όταν η σύνδεση καταστεί εφικτή, η dropping probability δείχνει να μην επηρεάζεται από τη μέθοδο σύνδεσης, συμπέρασμα που προκύπτει από τη σχετικά μεγάλη πυκνότητα MEC στο χώρο. Παρόλα αυτά, διαφορές στην απόδοση εντοπίζονται σχετικά με τη μέση ταχύτητα χρήστη. Θεωρώντας ξεχωριστά κάθε ένα από τα τέσσερα διαγράμματα του Σεναρίου 1, θα σχολιαστούν παρακάτω οι σημαντικότερες παρατηρήσεις που προέκυψαν από αυτά:

1. Στην περίπτωση του **Διαγράμματος 4.1**, φαίνεται πως ο Random Rule τείνει να είναι λιγότερο αποδοτικός, αλλά η διαφορά συγκριτικά με τους υπολοίπους τρεις είναι σχεδόν αμελητέα. Αντιθέτως, σημαντική πτώση στην blocking probability παρατηρείται καθώς η ταχύτητα του χρήστη αυξάνεται. Το φαινόμενο αυτό έχει ως πιθανή εξήγηση την πυκνότητα των MEC στο χάρτη, καθώς κινούμενος με μεγάλη ταχύτητα ο χρήστης μπορεί σε περίπτωση που η σύνδεσή του σε MEC απορριφθεί, να μεταβεί στο επόμενο σε μικρότερο χρονικό διάστημα.
2. Στο **Διάγραμμα 4.2**, ενώ ισχύει επίσης η οριακή επικράτηση των SOL, LFS έναντι των Round και Index, η εξάρτηση από την μέση ταχύτητα χρήστη είναι αντίστροφη. Όσο πιο γρήγορα κινείται ο UE τόσο μεγαλύτερη η πιθανότητα να απορριφθεί η εκτέλεση της εφαρμογής τρέχει κατά τη διάρκεια του user session. Η παρατήρηση αυτή ίσως προκύπτει από το γεγονός ότι οι χρήστες που κινούνται με μεγάλη σχετικά ταχύτητα είναι πιο πιθανό να βρεθούν στιγμιαία εκτός εμβέλειας MEC αυξάνοντας έτσι την πιθανότητα στιγμιαίας αποσύνδεσης από το σύστημα.
3. Το **Διάγραμμα 4.3** έχει ως μεταβλητή αναφοράς το ρυθμό άφιξης αιτημάτων  $\lambda$ . Παρατηρείται πως η απόδοση των αλγορίθμων είναι παρόμοια και σχεδόν ανεξάρτητη της μέσης ταχύτητας χρήστη. Η  $P_F$  αυξάνεται σχεδόν γραμμικά συναρτήσει του  $\lambda$ , ένα λογικό αποτέλεσμα καθώς όσο το σύστημα επιβαρύνεται με επιπλέον αιτήματα τόσο πιθανότερο για έναν χρήστη να μην καταφέρει να συνδεθεί.
4. Τέλος, στην περίπτωση του **Διαγράμματος 4.4**, αν και για ακόμη μια φορά δεν παρατηρείται διαφορά απόδοσης μεταξύ των αλγορίθμων, φαίνεται πως η αύξηση της μέσης ταχύτητας χρήστη οδηγεί σε αύξηση του μέσου  $P_H$ . Η εξήγηση του φαινομένου ίσως προκύπτει από το συλλογισμό για την παρόμοια περίπτωση του Διαγράμματος 4.2.

Σημειώνεται στο σημείο αυτό πως οι τιμές των  $P_F$  είναι εν γένει μεγαλύτερες των  $P_H$  για την ίδια τιμή των μεταβλητών αναφοράς καθώς όπως προαναφέρθηκε, το μοντέλο καθιστά ευκολότερη την διατήρηση μιας σύνδεσης παρά την εξ αρχής πραγματοποίηση της.

## 4.5.2 Σενάριο 2

Οι συνθήκες εκτέλεσης των αλγορίθμων στο δεύτερο Σενάριο δεν διαφέρουν από τις μεθόδους που περιεγράφηκαν στην προηγούμενη υποενότητα. Τα Διαγράμματα 4.5, 4.6, 4.7 και 4.8 προκύπτουν ακριβώς με τον ίδιο τρόπο όπως τα 4.1, 4.2, 4.3 και 4.4. Η ειδοποιός διαφορά μεταξύ των δύο Σεναρίων είναι η τοπολογία MEC στο χάρτη της μελέτης. Κατά την ανάλυση των αποτελεσμάτων που ακολουθεί, παρουσιάζονται οι διαφορές στα αποτελέσματα που προκύπτουν ακριβώς εξαιτίας της διαφοράς αυτής:

1. Στο **Διάγραμμα 4.5** φαίνεται καθαρά η διαφορά στην απόδοση των SOL και LFS συγκριτικά με αυτή των Random και Index. Η ανομοιόμορφη κατανομή των MEC στο χάρτη που χαρακτηρίζει το Σενάριο 2 δείχνει πως δίνει τη δυνατότητα στους αποδοτικότερους αλγορίθμους που δε βασίζονται τόσο στην τυχαιότητα να αναδείξουν τα πλεονεκτήματά τους. Ανάμεσα στους Index, Random ο δεύτερος φαίνεται πιο αξιόπιστος, βέβαια σε μικρό βαθμό. Όσο αφορά τους SOL, LFS η απόδοσή τους είναι σημαντικά καλύτερη των άλλων δύο, καθώς δίνουν τιμές της blocking probability αρκετά μικρότερες. Αν συγκριθούν αναμεταξύ τους, η απόδοσή τους δείχνει να είναι ακριβώς η ίδια. Τέλος, οι διαφορές στην απόδοση των αλγορίθμων σε σχέση με τις διαφορετικές τιμές των  $\frac{1}{\mu}, \bar{s}$  είναι αμελητέες και δεν επηρεάζουν σε σημαντικό βαθμό το τελικό αποτέλεσμα.
2. Τα αποτελέσματα του **Διαγράμματος 4.6** οδηγούν στο συμπέρασμα πως η υπεροχή των SOL και LFS εναντίων των Random και Index διατηρείται και στην περίπτωση του  $P_H$ . Η διαφορά στην απόδοση δεν είναι τόσο μεγάλη αλλά είναι ακόμα σημαντική. Επιπλέον, σε αντίθεση με πριν, οι αλγόριθμοι δείχνουν να επηρεάζονται τόσο από την αύξηση της μέσης ταχύτητας χρήστη όσο και από την αύξηση του μέσου χρόνου εξυπηρέτησης. Όσο αυτές οι δύο παράμετροι αυξάνονται τόσο πιθανότερο είναι ο χρήστης να απορριφθεί από το σύστημα κατά τη διάρκεια της εξυπηρέτησης του. Η παρατήρηση αυτή φαίνεται σαν λογικό επακόλουθο, καθώς όσο πιο γρήγορα απομακρύνεται από το σημείο σύνδεσης ο χρήστης και όσο περισσότερος χρόνος χρειάζεται για την ολοκλήρωση του αιτήματός του τόσο μεγαλύτερος ο κίνδυνος να μην περατωθεί το αίτημά του. Για ακόμη μια φορά, ο Random κατέχει ελαφρύ πλεονέκτημα εναντίων του Index, ενώ η απόδοση των SOL, LFS είναι σχεδόν ταυτοτική.
3. Περνώντας στον υπολογισμό του  $P_F$  συναρτήσει του  $\lambda$ , στο **Διάγραμμα 4.7** φαίνεται για ακόμη μια φορά η υπεροχή στην απόδοση των SOL, LFS εναντίων των Index, Random. Μάλιστα, παρατηρείται πως για χαμηλές τιμές του  $\lambda$ , η πιθανότητα απόρριψης νέου session για το χρήστη αυξάνεται σημαντικά ποσοστιαία για τους δύο τελευταίους. Επιπλέον, η αύξηση της εκθετικής παραμέτρου έστω και για μικρές τιμές μειώνει την αξιοπιστία των SOL και LFS. Αντίθετα, οι Index και Random δεν επηρεάζονται σχεδόν καθόλου στο εύρος του  $\lambda$  που έχει τεθεί. Κλείνοντας, αξίζει να αναφερθεί πως η αύξηση της ταχύτητας δεν φαίνεται να επηρεάζει την εκτέλεση των αλγορίθμων, ενώ ο Index έχει ελαφρύ πλεονέκτημα έναντι του Random.
4. Τα γενικά συμπεράσματα για το Σενάριο 2 δεν αλλάζουν ούτε στο **Διάγραμμα 4.8**, καθώς οι SOL και LFS, χωρίς να παρουσιάζουν διαφορά στην απόδοση μεταξύ τους, δείχνουν για μικρότερες τιμές του  $\lambda$  σαφή ανωτερότητα. Βέβαια, υπάρχουν ακόμη στοιχεία που αξίζει να αναφερθούν σαν μεταβολές σε σχέση με όσα αναφέρθηκαν στο (3). Ο Index δείχνει ελαφρώς μεγαλύτερο αριθμό αξιοπιστίας σε αντίθεση με το **Διάγραμμα 4.7**. Επιπροσθέτως, η αύξηση της μέσης ταχύτητας χρήστη μειώνει την αξιοπιστία του συστήματος σε μικρό αλλά παρατηρήσιμο βαθμό. Κλείνοντας, παρατηρείται πως για τη μέγιστη τιμή του  $\lambda$  (45.45) οι διαφορές στην απόδοση μεταξύ των τεσσάρων αλγορίθμων μειώνονται σημαντικά.

Το κυριότερο σημείο αναφοράς που προκύπτει από τη μελέτη του Σεναρίου 2 και των αποτελεσμάτων του είναι πως σε ένα περιβάλλον αυξημένης πολυπλοκότητας για τα UEs, οι αλγόριθμοι SOL και LFS βελτιώνουν τα  $P_F, P_H$  σχεδόν κατά μια ολόκληρη τάξη μεγέθους, συγκριτικά με τους πιο απλούς Random, Index.

# ΚΕΦΑΛΑΙΟ 5ο

## ΑΛΓΟΡΙΘΜΟΣ LODCO

### 5.1 Εισαγωγή

Στο παρόν κεφάλαιο θα εξερευνηθεί η διαδικασία του mobile task offloading, με βάση το μοντέλο που προτείνεται μέσω του αλγορίθμου που εισάγεται στο [22]. Η προσέγγιση αυτή διαφέρει σημαντικά ως προς την υλοποίηση και τη μεθοδολογία που ακολουθήθηκε στο προηγούμενο κεφάλαιο. Η πρώτη σημαντική αλλαγή έγκειται στο γεγονός ότι τα UEs δεν παρουσιάζονται στο σύστημα ως απλές αιτήσεις εξυπηρέτησης που προκύπτουν και κινούνται στο χώρο, αλλά σαν Energy Harvesting συσκευές. Αυτό σημαίνει πως τα επίπεδα αποθηκευμένης ενέργειας που διαθέτει κάθε χρήστης και το πως αυτά ανανεώνονται αποτελούν σημαντική μεταβλητή του προβλήματος. Επιπλέον, στο μοντέλο ερευνάται η αλληλεπίδραση ενός χρήστη με έναν εξυπηρετητή σε δεδομένη και σταθερή απόσταση μεταξύ τους. Τέλος, ο αλγόριθμος που θα εισαχθεί προβλέπει περιπτώσεις στις οποίες η εκτέλεση ενός task πραγματοποιείται εν τέλει από τη συσκευή και όχι από κάποιον MEC server.

Το μοντέλο που θα μελετηθεί προσομοιώνει ένα σύστημα MEC, το οποίο αποτελείται από μια EH συσκευή και έναν εξυπηρετητή. Ο αλγόριθμος LODCO προβλέπει τρία διαφορετικά σενάρια για την εκτέλεση ενός task:

1. Εκτέλεση τοπικά στη συσκευή (**Mobile Execution**)
2. Εκτέλεση στον εξυπηρετητή (**Server Execution**)
3. Αδυναμία εκτέλεσης task και επακόλουθη απόρριψη (**Drop Task**)

Ο αλγόριθμος αρχικά ελέγχει τη δυνατότητα πραγματοποίησης του task και στη συνέχεια επιλύοντας ένα πρόβλημα απόφασης Markov (**Markov Decision Problem, MDP**) έχοντας σαν μεταβλητή αναφοράς το χρονικό κόστος της εκτέλεσης, επιλέγει την κατάλληλη μέθοδο.

Στις επόμενες ενότητες, θα παρουσιαστούν οι βασικές αρχές του μοντέλου που περιγράφεται στο [22], καθώς και η παραμετροποίηση του προβλήματος. Στη συνέχεια, θα υλοποιηθεί μια εκδοχή του αλγορίθμου σε MATLAB βασισμένη εν μέρει στο [27] και στην προσέγγιση του αλγορίθμου που εισάγεται στο [28]. Τέλος, τα αποτελέσματα της προσομοίωσης θα σχολιαστούν εκτιμώντας έτσι τη χρησιμότητα του αλγορίθμου σαν εργαλείο σε ένα σύστημα MEC.

### 5.2 Μοντέλα Συστήματος

#### 5.2.1 Computational Model

Το πρώτο βήμα προς την σύλληψη του αλγορίθμου είναι η κωδικοποίηση των μεγεθών που ορίζουν το πρόβλημα. Οι μεταβλητές αυτές θα χρησιμοποιηθούν στην πορεία για την εκτέλεση του αλγορίθμου και την παραγωγή πρακτικών αποτελεσμάτων. Αρχικά, θεωρείται πως το γενικό σύστημα, δηλαδή η EH συσκευή και ο MEC server επικοινωνούν μέσω ενός ασύρματου καναλιού, καθώς εντός του server το σύστημα κλωνοποιεί τη συσκευή μέσω ενός VM, ώστε να εκτελέσει μια υπολογιστική διαδικασία (**computational task**).

Ο χρόνος που απαιτείται για την εκτέλεση του αλγορίθμου θεωρείται χωρισμένος σε διακριτές χρονικές στιγμές μήκους  $\tau$  ενώ το σύνολο των διακριτών αυτών τμημάτων συμβολίζεται ως  $T \triangleq \{0,1,2 \dots\}$ . Το ασύρματο κανάλι είναι ανεξάρτητο και ομοιόμορφα κατανομημένο. Ενώ παραμένει στατικό για κάθε χρονική στιγμή ανεξάρτητα, διαφέρει για κάθε  $\tau$ . Θέτοντας  $h^t$  το κέρδος ισχύος του καναλιού, ισχύει ότι  $h^t \sim F_H(x)$ ,  $t \in T$  (όπου  $F_H(x)$  η συνάρτηση κατανομής πιθανότητας του  $h^t$ ).

Η αναπαράσταση ενός task γίνεται ως  $A(L, \tau_d)$  με  $L$  το μέγεθος του task σε bits και  $\tau_d$  το μέγιστο χρόνο αναμονής μέχρι την εκτέλεσή του. Αν ο χρόνος  $t$  υπερβεί αυτή την τιμή, τότε το task απορρίπτεται. Η παραγωγή αιτημάτων για εκτέλεση αποτελεί μια ανεξάρτητη διαδικασία Bernoulli, καθώς στις αρχές κάθε χρονικού διαστήματος  $\tau$  υπάρχει πιθανότητα  $\rho$  να εμφανιστεί αίτημα και  $1 - \rho$  σε αντίθετο ενδεχόμενο. Αν κατά τη χρονική στιγμή  $t$  υπάρξει αίτημα εκτέλεσης task, τότε το γεγονός αυτό συμβολίζεται ως  $\zeta^t = 1$ . Προφανώς η αντίθετη περίπτωση συμβολίζεται ως  $\zeta^t = 0$ . Επομένως ισχύει:

$$P\{\zeta^t = 1\} = 1 - P\{\zeta^t = 0\} = \rho, t \in T$$

Οι εφαρμογές που εντέλει θα εκτελούνται στο σύστημα θα πρέπει να ικανοποιούν τη συνθήκη  $\tau_d \leq \tau$ . Το σύστημα δε διαθέτει buffer, επομένως οι εφαρμογές που δεν πληρούν την παραπάνω συνθήκη θα απορρίπτονται. Η εκτέλεση ενός task μπορεί να πραγματοποιηθεί εντός της συσκευής (mobile execution) ή να ανατεθεί στον server (server execution). Σε περίπτωση που η ενέργεια της συσκευής δεν επαρκεί, τότε το task απορρίπτεται. Σε κάθε χρονική στιγμή ένα από τα τρία ενδεχόμενα και μόνο μπορεί να πραγματοποιηθεί. Για να συμβολιστεί ο τρόπος εκτέλεσης, χρησιμοποιείται η κωδικοποίηση  $I_j^t \in \{0,1\}$ ,  $j = \{m, s, d\}$ .

Προφανώς, ο συμβολισμός προκύπτει από τα αρχικά του τρόπου εκτέλεσης (**mobile/server/drop task**). Για παράδειγμα, αν την χρονική στιγμή  $t$  είναι εφικτή η εκτέλεση στη συσκευή τότε ισχύει  $I_m^t = 1$ . Ο περιορισμός που προκύπτει από το γεγονός ότι μόνο ένα από τα τρία ενδεχόμενα πραγματοποιείται για μια δεδομένη χρονική στιγμή εκφράζεται ως:

$$I_m^t + I_s^t + I_d^t = 1, t \in T \quad (5.1)$$

Τα δύο βασικά μεγέθη που πρέπει να υπολογίζονται σε κάθε εκδοχή εκτέλεσης μιας εφαρμογής από το χρήστη είναι η καθυστέρηση που εισάγει στο σύστημα και η ενέργεια που καταναλώνει η συσκευή. Αυτές οι δύο παράμετροι θα βοηθήσουν στην πορεία, ώστε να δημιουργήσουμε μια έκφραση για το συνολικό κόστος εκτέλεσης ενός task. Ο αλγόριθμος του μοντέλου θα μπορεί χρησιμοποιώντας αυτή την έκφραση να διενεργεί υπολογισμούς ώστε να παράγει τα αποτελέσματα που ζητούνται. Στις επόμενες παραγράφους, θα υπολογιστούν εκφράσεις για τα δύο αυτά μεγέθη.

Το **Μοντέλο Τοπικής Εκτέλεσης** (Local Execution Model) περιγράφει την εκτέλεση ενός task εντός της συσκευής. Η υπολογιστική ισχύς που καταβάλλεται για να επεξεργαστεί η συσκευή ένα task εκφράζεται από τους κύκλους ρολογιού της CPU της. Ο αριθμός κύκλων που απαιτούνται για την επεξεργασία ενός bit εισόδου (input bit) συμβολίζεται ως  $X$ . Επομένως, για να γίνει επεξεργασία ενός task  $A(L, \tau_d)$ , απαιτούνται  $W = LX$  CPU cycles. Οι συχνότητες ρολογιού που απαιτούνται για την εκτέλεση  $W$  CPU cycles κατά την χρονική στιγμή  $t$  συμβολίζονται ως  $f_w^t$ ,  $w = 1, 2, \dots, W$ . Η τελική έκφραση για την καθυστέρηση που προκύπτει από την εκτέλεση ενός task τοπικά την χρονική στιγμή  $t$  είναι:

$$D_{mobile}^t = \sum_{w=1}^W (f_w^t)^{-1} \quad (5.2)$$

Αντίστοιχα, η ενέργεια που καταναλώνεται εκφράζεται ως:

$$E_{mobile}^t = \kappa \sum_{w=1}^W (f_w^t)^2 \quad (5.3)$$



Με  $\kappa$  συμβολίζεται η effective switch capacitance, ένα μέγεθος που εξαρτάται από την αρχιτεκτονική του chip [29]. Οι επιμέρους συχνότητες κύκλου ρολογιού φράσσονται από μια μέγιστη τιμή, δηλαδή  $f_w^t \leq f_{CPU}^{max}, \forall w$ .

Το **Μοντέλο Εκτέλεσης MEC** (Mobile-Edge Executing Model) αναφέρεται στην μετάδοση ενός task  $A(L, \tau_d)$  στον MEC server για εκτέλεση εντός του συστήματός του. Ο server θεωρείται πως διαθέτει τους απαραίτητους υπολογιστικούς πόρους ώστε να αγνοείται η καθυστέρηση εκτέλεσης του task εντός του. Επιπλέον το output των υπολογισμών του έχει αρκετά μικρό μέγεθος, ώστε η καθυστέρηση αποστολής των αποτελεσμάτων στον χρήστη είναι αμελητέα. Επομένως, η καθυστέρηση που μένει να ληφθεί υπόψιν αφορά τη μετάδοση του task στον server από τη συσκευή. Η ισχύς μετάδοσης συμβολίζεται ως  $p^t$  και το άνω όριο της ως  $p_{tx}^{max}$ . Σύμφωνα με τον τύπο των Shannon-Hartley για τον ρυθμό μετάδοσης δεδομένων που μπορεί να επιτευχθεί κατά τη χρονική στιγμή  $t$ , συμβολίζοντας ως  $\omega$  το εύρος ζώνης του συστήματος και  $\sigma$  την ισχύς θορύβου του δέκτη, προκύπτει η ακόλουθη έκφραση για τον ρυθμό:

$$r(h^t, p^t) = \omega \log_2 \left( 1 + \frac{h^t p^t}{\sigma} \right) \quad (5.4)$$

Επομένως, η καθυστέρηση που εισάγεται στο σύστημα για την αποστολή των δεδομένων θα είναι:

$$D_{server}^t = \frac{L}{r(h^t, p^t)} \quad (5.5)$$

Αντίστοιχα η κατανάλωση ενέργειας στην συσκευή:

$$E_{server}^t = p^t \cdot D_{server}^t \quad (5.6)$$

### **5.2.2 Energy Harvesting Model**

Η ενέργεια που φτάνει στη συσκευή μοντελοποιείται ως κβαντισμένα πακέτα ενέργειας που καταφτάνουν στην συσκευή στην αρχή κάθε χρονικής στιγμής  $t$ , τα οποία συμβολίζονται ως  $E_H^t$  και έχουν ως άνω όριο την τιμή  $E_H^{max}$ . Τα πακέτα αυτά είναι ομοιόμορφα κατανομημένα και ανεξάρτητα αναμεταξύ τους. Μεταξύ δυο διαδοχικών χρονικών στιγμών, μέρος του πακέτου ενέργειας μπορεί να αποθηκευτεί στην μπαταρία της συσκευής και να είναι διαθέσιμο προς κατανάλωση στην αρχή της επόμενης χρονικής στιγμής. Τα μέρη αυτά συμβολίζονται ως  $e^t, 0 \leq e^t \leq E_H^t, t \in \mathbb{T}$ .

Ορίζεται επίσης το επίπεδο ενέργειας μπαταρίας στην αρχή της χρονικής στιγμής  $t$  ως  $B^t, B^t < +\infty, t \in \mathbb{T}$ . Χωρίς βλάβη της γενικότητας θεωρείται  $B^0 = 0$ . Σημειώνεται επίσης πως η κατανάλωση ενέργειας για σκοπούς πέρα της εκτέλεσης του task που εισήλθε στο σύστημα αγνοείται για λόγους απλότητας. Αν λοιπόν συμβολιστεί η συνολική ενέργεια που καταναλώνει η συσκευή την χρονική στιγμή  $t$  ως  $\xi(I^t, f^t, p^t)$ , συναρτήσει των συχνοτήτων κύκλου ρολογιού, της ισχύς μετάδοσης και της μεθόδου εκτέλεσης, ισχύουν οι παρακάτω σχέσεις:

$$\xi(I^t, f^t, p^t) = I_m^t E_{mobile}^t + I_s^t E_{server}^t \quad (5.7)$$

$$\xi(I^t, f^t, p^t) \leq B^t < +\infty, t \in \mathbb{T} \quad (5.8)$$

$$B^{t+1} = B^t - \xi(I^t, f^t, p^t) + e^t, \quad t \in \mathbb{T} \quad (5.9)$$

### 5.3 Διατύπωση Προβλήματος

Η καθυστέρηση κατά την εκτέλεση είναι η βασική παράμετρος που εκφράζει σε πολλές εφαρμογές το QoE του χρήστη. Το μέγεθος αυτό εκφράζεται στο μοντέλο που εισάγεται στο [22] ως κόστος εκτέλεσης (**execution cost**). Το μοντέλο προβλέπει μια ακόμα συνιστώσα εκτός της προφανούς που αφορά την εκτέλεση ενός task. Κάθε φορά που ένα task απορρίπτεται, το συνολικό κόστος αυξάνεται κατά μια μονάδα κόστους. Η συνιστώσα αυτή ελέγχεται από έναν παράγοντα βάρους. Ο τύπος για το κόστος εκτέλεσης με βάση όσα παρουσιάστηκαν θα είναι το άθροισμα με βάρη (**weighted sum**) των δύο συνιστωσών:

$$cost^t = D(I^t, f^t, p^t) + \varphi \cdot \mathbf{1}(\zeta^t = 1, I_a^t = 1) \quad (5.10)$$

Όπου  $\varphi$  (σε second) το βάρος του κόστους απόρριψης task,  $\mathbf{1}(\cdot)^3$  η συνάρτηση δείκτη, και  $D(I^t, f^t, p^t)$  δίνεται από τον τύπο:

$$D(I^t, f^t, p^t) = \mathbf{1}(\zeta^t = 1) \cdot (I_m^t D_{mobile}^t + I_s^t D_{server}^t) \quad (5.11)$$

Θεωρείται πως η επιτυχής ολοκλήρωση ενός task πάντα προτιμάται από την απόρριψή του ( $\tau_d \leq \varphi$ ). Επιπλέον, αν ένα task εγκριθεί για εκτέλεση τότε θα πρέπει να ολοκληρωθεί αυστηρά πριν το πέρας του ορίου  $\tau_d$ :

$$D(I^t, f^t, p^t) \leq \tau_d, t \in T \quad (5.12)$$

#### 5.3.1 Ελαχιστοποίηση κόστους εκτέλεσης

Μια πρώτη διατύπωση του προβλήματος με βάση όσα ορίστηκαν ανωτέρω αφορά την ελαχιστοποίηση κόστους εκτέλεσης (**Execution Cost Minimization Problem, ECM Problem**) και περιγράφεται ως εξής:

$$ECM: \min_{I^t, f^t, p^t, e^t} \lim_{T \rightarrow \infty} \frac{1}{T} \mathbb{E} \left[ \sum_{t=0}^{T-1} cost^t \right]$$

$$I_m^t + I_s^t \leq \zeta^t, t \in T \quad (5.13)$$

$$\varepsilon(I^t, f^t, p^t) \leq E_{max}, t \in T \quad (5.14)$$

$$0 \leq p^t \leq p_{tx}^{max} \cdot \mathbf{1}(I_s^t = 1), t \in T \quad (5.15)$$

$$0 \leq f_w^t \leq f_{CPU}^{max} \cdot \mathbf{1}(I_m^t = 1), t \in T, w = 1, 2, \dots, W \quad (5.16)$$

$$I_m^t, I_s^t, I_a^t \in \{0, 1\}, t \in T \quad (5.17)$$

Η συνθήκη (5.13) αναφέρει πως αν δεν έρθει αίτημα εξυπηρέτησης δεν υπάρχει η δυνατότητα εκτέλεσης τοπικά ή στον server. Η συνθήκη (5.14) εμποδίζει την μπαταρία της συσκευής να φορτιστεί άνω ενός συγκεκριμένου ορίου και τέλος οι συνθήκες (5.15), (5.16) ορίζουν τους περιορισμούς υπολογιστικών δυνατοτήτων για τα server και mobile execution αντίστοιχα. Προφανώς η συνθήκη (5.17) καθορίζει τη δυαδική φύση των τρόπων εκτέλεσης.

<sup>3</sup> Συνάρτηση δείκτη (indicator function): Παίρνει την τιμή 1 αν ικανοποιείται η συνθήκη εντός του ορισματός της και την τιμή 0 σε οποιαδήποτε άλλη περίπτωση.

### 5.3.2 Προβλήματα αποφάσεων Markov

Ένα πρόβλημα απόφασης Markov (**Markov Decision Problem, MDP**) αποτελείται από έναν πεπερασμένο αριθμό καταστάσεων και αποφάσεων [30]. Κάθε χρονική στιγμή παρατηρείται μια κατάσταση (**state**) και επιλέγεται η πραγματοποίηση μιας πράξης-απόφασης (**action**). Η πράξη αυτή απαιτεί ένα κόστος. Το κόστος και η επόμενη κατάσταση εξαρτώνται αποκλειστικά από την κατάσταση στην οποία βρίσκεται το πρόβλημα τη συγκεκριμένη χρονική στιγμή και την απόφαση που θα παρθεί. Συχνά η απόφαση αυτή καθορίζεται πιθανοτικά.

Στο ECM παρατηρείται πως κάθε κατάσταση του συστήματος αποτελείται από ένα task request, την διαθέσιμη ενέργεια προς απορρόφηση, το επίπεδο της μπαταρίας και την κατάσταση του καναλιού επικοινωνίας. Επιπλέον, η αλλαγή σε επόμενη κατάσταση προκύπτει μέσω των πράξεων της απορρόφησης ενέργειας και τις επιλογής τρόπου εκτέλεσης. Οι πράξεις αυτές εξαρτώνται αποκλειστικά και μόνο από την παρούσα κατάσταση του συστήματος. Επομένως το ECM αποτελεί ένα πρόβλημα απόφασης Markov.

Το ECM μπορεί να επιλυθεί χρησιμοποιώντας αλγόριθμους επίλυσης MDP που συναντώνται στη βιβλιογραφία [31]. Στις προσεγγίσεις αυτές όμως ο αριθμός των μεταβλητών και των πιθανών πράξεων τείνει να παίρνει τιμές δύσκολα διαχειρίσιμες ακόμη και στην περίπτωση που ο συνολικός αριθμός καταστάσεων είναι σχετικά μικρός. Αντίθετα, η μέθοδος του [22] εισάγει έναν αλγόριθμο βασισμένο στη μέθοδο της βελτιστοποίησης Lyapunov, τον **Lyapunov optimization-based dynamic computation offloading algorithm (LODCO Algorithm)**. Τα σημαντικότερα πλεονεκτήματά του είναι τα ακόλουθα:

1. Δεν παρουσιάζεται η ανάγκη κβάντισης του συνόλου πιθανών καταστάσεων και του συνόλου δυνατών πράξεων. Επιπλέον, η απόφαση που λαμβάνει ο αλγόριθμος είναι χαμηλής πολυπλοκότητας και δεν απαιτείται σύστημα μνήμης ώστε να αποθηκευτούν οι βέλτιστες στρατηγικές.
2. Ο αλγόριθμος δεν απαιτεί πρόωμη γνώση των στατιστικών που αφορούν την κατάσταση του καναλιού, τα επίπεδα ενέργειας και την κατανομή των αιτημάτων που λαμβάνονται.
3. Η επίδοση του αλγόριθμου βασίζεται στον έλεγχο δύο παραμέτρων που αφορούν την βελτιστοποίηση Lyapunov, όπως θα δειχθεί σε επόμενη ενότητα.

### 5.3.3 Χαρακτηριστικά αλγορίθμου

Για να παρουσιαστεί αναλυτικά ο αλγόριθμος, πρέπει αρχικά να οριστούν τρία βασικά μεγέθη:  $\tilde{E}_{max}$

A) Η παράμετρος διατάραξης (**perturbation parameter**)  $\theta$ .

B) Η ουρά εικονικής ενέργειας (**virtual energy queue**)  $\tilde{B}^t$ .

Γ) Η παράμετρος ελέγχου (**control parameter**)  $V$ .

Οι ανωτέρω παράμετροι συνδέονται μεταξύ τους μέσω της σχέσης:

$$\theta \geq \tilde{E}_{max} + V\varphi E_{min}^{-1} \quad (5.19)$$

$$\tilde{E}_{max} = \min\{\max\{\kappa W (f_{CPU}^{max})^2, p_{tx}^{max} \tau\}, E_{max}\}$$

$$0 < V < +\infty \text{ (Joule}^2 \cdot \text{second}^{-1}\text{)}$$

Η virtual energy queue ορίζεται ως  $\tilde{B}^t = B^t - \theta$  και εκφράζει μια μετατοπισμένη εκδοχή του μεγέθους που ορίστηκε ως επίπεδο ενέργειας μπαταρίας. Η βασική λειτουργία του αλγορίθμου είναι η ελαχιστοποίηση του αθροίσματος μεταξύ κόστους εκτέλεσης και ενέργειας που απορροφάται. Οι παράμετροι  $\tilde{B}^t$ ,  $V$  χρησιμοποιούνται σαν βάρη για τις δύο αυτές ποσότητες.

Η χρησιμοποίηση αυτού του είδους ζυγισμένου αθροίσματος βασίζεται στην τεχνική της βελτιστοποίησης Lyapunov [32]. Η μέθοδος αυτή βασίζεται στην ελαχιστοποίηση αθροισμάτων της μορφής  $\Delta L(t) + Vp(t)$ , όπου  $\Delta L(t)$  η μετατόπιση ενός μεγέθους που εξελίσσεται σε διακριτές χρονικές στιγμές και  $Vp(t)$  το γινόμενο της παραμέτρου ελέγχου με μια συνάρτηση κόστους. Ακριβώς η ίδια μεθοδολογία εφαρμόζεται στη σχέση (19) και για αυτό στο όνομα του αλγορίθμου εμφανίζεται ο όρος "*Lyapunov optimization-based*".

### **5.3.4 Βήματα αλγορίθμου**

Παρακάτω παρουσιάζεται ο αλγόριθμος LODCO [22] σε μορφή ψευδοκώδικα:

<b>The LODCO Algorithm</b>
1: At the beginning of time slot $t$ , obtain the task request indicator $\zeta^t$ , virtual energy queue length $\tilde{B}^t$ , harvestable energy $E_H^t$ , and channel gain $h^t$ .
2: Decide $I^t, f^t, e^t$ and $p^t$ by solving the following deterministic problem: $\min_{I^t, f^t, e^t, p^t} \tilde{B}^t [e^t p^t - \varepsilon(I^t, f^t, p^t) + V[D(I^t, f^t, p^t) + \varphi \cdot \mathbf{1}(\zeta^t = 1, I_d^t = 1)]]$ $(1), (6), (12)-(18)$
3: Update the virtual energy queue according to (9)
4: Set $t = t + 1$ .

Στο βήμα 2 ζητούμενο από μια πρακτική εφαρμογή του αλγορίθμου είναι ο καθορισμός τριών τιμών (παραμέτροι  $f^t, e^t, p^t$ ) που περιγράφουν το βέλτιστο σενάριο για κάθε πιθανή περίπτωση. Οι τιμές αυτές θα χρησιμοποιηθούν για να καθοριστεί μια συνολική τιμή που συμβολίζει το γενικό κόστος επιλογής για κάθε πιθανό σενάριο (mobile execution, server execution). Η τελική επιλογή βασίζεται τελικά στην μικρότερη δυνατή τιμή κόστους που θα προκύψει. Σε περίπτωση που προκύψει  $I_d^t = 1$  η εκτέλεση του task προφανώς απορρίπτεται. Η υλοποίηση του αλγορίθμου αφορά τον καθορισμό των παραμέτρων που αναφέρθηκαν ανωτέρω μέσω της επίλυσης επιμέρους προβλημάτων βελτιστοποίησης για κάθε σενάριο εκ των  $I_i^t$  ξεχωριστά.

## **5.4 Υλοποίηση αλγορίθμου στο MATLAB**

Η πρακτική υλοποίηση του αλγορίθμου βασίζεται στην επίλυση επιμέρους προβλημάτων βελτιστοποίησης για κάθε πιθανό σενάριο. Η διαδικασία αυτή χωρίζεται στα πλαίσια της υλοποίησης στη γλώσσα MATLAB σε έξι διακριτά βήματα:

**Βήμα 1:** Υπολογισμός της βέλτιστης ενέργειας απορρόφησης. (ανεξάρτητο της ύπαρξης ή μη task προς εκτέλεση)

**Βήμα 2:** Υπολογισμός κόστους για κάθε πιθανό σενάριο (**mobile execution, server execution, task dropped**).

**Βήμα 3:** Επιλογή βέλτιστης μεθόδου εκτέλεσης.

**Βήμα 4:** Ανάλογα την επιλογή μεθόδου, υπολογισμός τελικής καθυστέρησης και κατανάλωσης ενέργειας.

**Βήμα 5:** Ενημέρωση επιπέδου μπαταρίας και μετάβαση στην επόμενη χρονική στιγμή.

**Βήμα 6:** Παρουσίαση αποτελεσμάτων σε γραφικές παραστάσεις.

Η υλοποίηση που παρουσιάζεται στην παρούσα εργασία και ακολουθεί τα ανωτέρω στάδια βασίζεται στον κώδικα που προτείνεται στο [27]. Οι μέθοδοι εύρεσης των βέλτιστων τιμών για τα  $f^t, e^t, p^t$  περιγράφονται αναλυτικά στο κεφάλαιο IV-B: *Optimal Computation Offloading in Each Time Slot του* [22]. Στην επόμενη υποενότητα θα παρατεθούν τα κύρια σημεία των μεθόδων αυτών, παραλείποντας τις αποδείξεις των θεωρημάτων στα οποία βασίζονται.

### **5.4.1 Optimal Energy Harvesting**

Πριν την επιλογή μεθόδου εκτέλεσης ο αλγόριθμος πρέπει να υπολογίσει την βέλτιστη τιμή της ενέργειας που απορροφά η συσκευή στην αρχή κάθε χρονικής στιγμής. Η τιμή αυτή είναι κρίσιμη για τη συνέχεια της διαδικασίας καθώς τα επίπεδα της μπαταρίας αποτελούν σημαντική προϋπόθεση για τον έλεγχο υλοποιησιμότητας (**feasibility**) κάθε μεθόδου εκτέλεσης και προκύπτει από την παρακάτω έκφραση:

$$\min_{0 \leq e^t \leq E_H^t} \tilde{B}^t e^t \quad (5.20)$$

Η βέλτιστη τιμή με βάση την (5.20) θα είναι:

$$e^{t*} = E_H^t \cdot \mathbf{1}\{\tilde{B}^t \leq 0\} \quad (5.21)$$

Οι (5.20), (5.21) περιγράφουν πως όταν το επίπεδο της εικονικής μπαταρίας φτάσει στο 0, τότε η συσκευή θα απορροφήσει ένα εκ των κβαντισμένων πακέτων ενέργειας που είναι διαθέσιμα.

### **5.4.2 Optimal Computation Offloading**

Έχοντας υπολογίσει το  $e^{t*}$  μένει να καθοριστεί η βέλτιστη στρατηγική εκτέλεσης. Στην περίπτωση που  $\zeta^t = 0$  δεν έχει νόημα η πραγματοποίηση των βημάτων 2 και 3, επομένως γίνεται απευθείας μετάβαση σε επόμενη χρονική στιγμή. Αντίθετα, αν το σύστημα αντιληφθεί αίτημα για εκτέλεση task τότε πρέπει να ελεγχθεί η υλοποιησιμότητα των δύο διαφορετικών σεναρίων και να αποφασιστεί η βέλτιστη στρατηγική. Δεδομένου του  $e^{t*}$ , το ECM μετατρέπεται ως εξής:

$$\min_{I^t, f^t, e^t, p^t} \{-\tilde{B}^t \cdot \varepsilon(I^t, f^t, p^t) + V[D(I^t, f^t, p^t) + \varphi \cdot \mathbf{1}(\zeta^t = 1, I_d^t = 1)]\} \quad (5.22)$$

Η έκφραση αυτή θα χρησιμοποιηθεί ως βάση για τον έλεγχο υλοποιησιμότητας των **mobile/server execution**.

#### **5.4.2 (a) Mobile Execution**

Στο σενάριο της εκτέλεσης του task εντός της συσκευής η ποσότητα που αναζητείται είναι η βέλτιστη συχνότητα ρολογιού της CPU  $f^{t*}$ . Για  $I_m^t = 1, I_s^t = I_d^t = 0, p_t = 0$  η σχέση (5.22) παίρνει την ακόλουθη μορφή:

$$\mathbf{ECM}_{me}: \min_{f^t} (-\tilde{B}^t \cdot \kappa W (f^t)^2 + V \cdot \frac{W}{f^t}) \quad (5.23)$$

$$0 \leq f_w^t \leq f_{CPU}^{max}, \frac{W}{f^t} \leq \tau_d, \kappa W (f^t)^2 \in [E_{min}, E_{max}].$$

Αποδεικνύεται [22] πως το  $ECM_{me}$  είναι υλοποιήσιμο αν και μόνο αν  $f_L \leq f_U$ , όπου  $f_L = \max\left\{\sqrt{\frac{E_{min}}{\kappa W}}, \frac{W}{\tau_d}\right\}$  και  $f_U = \min\left\{\sqrt{\frac{E_{max}}{\kappa W}}, f_{max}\right\}$ . Επιπλέον αν είναι υλοποιήσιμο, το  $f^{t*}$  προκύπτει ως εξής:

$$f^{t*} = \begin{cases} f_U, & \tilde{B}^t \geq 0 \text{ or } \tilde{B}^t < 0, f_0^t > f_U \\ f_0^t, & \tilde{B}^t < 0, f_L \leq f_0^t \leq f_U \\ f_L, & \tilde{B}^t < 0, f_0^t < f_L \end{cases} \quad (5.24)$$

με  $f_0^t = \sqrt[3]{\frac{V}{-2\tilde{B}^t\kappa}}$ . Μόλις υπολογιστεί το  $f^{t*}$  η τιμή του εισάγεται στη σχέση (5.23) και παράγει την τιμή που εκφράζει το βέλτιστο κόστος τοπικής εκτέλεσης, η οποία συμβολίζεται με  $J_m(f^{t*})$ . Το κόστος αυτό αποτελείται από το ζυγισμένο άθροισμα του χρόνου εκτέλεσης του task  $\left\{\frac{W}{f^t}\right\}$  και της κατανάλωσης ενέργειας  $\{\kappa W (f^t)^2\}$ .

#### **5.4.2 (β) Server Execution**

Η ποσότητα που αναζητείται στο σενάριο της εκτέλεσης του task εντός του MEC server είναι η βέλτιστη ισχύς μετάδοσης  $p^{t*}$ . Για  $I_s^t = 1, I_m^t = I_d^t = 0, f^t = 0$  η σχέση (5.22) παίρνει την ακόλουθη μορφή:

$$ECM_{se}: \min_{p^t} \left( -\tilde{B}^t \cdot \frac{p^t}{r(h^t, p^t)} + V \cdot \frac{L}{r(h^t, p^t)} \right) \quad (5.25)$$

$$\text{με } 0 \leq p^t \leq p_{tx}^{max}, \frac{L}{r(h^t, p^t)} \leq \tau_d, \frac{p^t L}{r(h^t, p^t)} \in [E_{min}, E_{max}].$$

Αποδεικνύεται πως το  $ECM_{se}$  είναι υλοποιήσιμο αν και μόνο αν  $p_L^t \leq p_U^t$ , όπου<sup>4</sup>:

$$p_L^t = \begin{cases} \frac{\frac{L}{(2^{\omega\tau_d}-1)\sigma}}{h^t}, & \frac{\sigma L \ln 2}{\omega h^t} \geq E_{min} \\ \max\left\{\frac{\frac{L}{(2^{\omega\tau_d}-1)\sigma}}{h^t}, p_{E_{min}}^t\right\}, & \frac{\sigma L \ln 2}{\omega h^t} < E_{min} \end{cases} \quad \text{και} \quad p_U^t = \begin{cases} \min\{p_{tx}^{max}, p_{E_{max}}^t\}, & \frac{\sigma L \ln 2}{\omega h^t} < E_{max} \\ 0, & \frac{\sigma L \ln 2}{\omega h^t} \geq E_{max} \end{cases}$$

Αν το πρόβλημα είναι υλοποιήσιμο τότε το  $p^{t*}$  προκύπτει ως εξής<sup>5</sup>:

$$p^{t*} = \begin{cases} p_U^t, & \tilde{B}^t \geq 0 \text{ or } \tilde{B}^t < 0, p_U^t < p_0^t \\ p_L^t, & \tilde{B}^t < 0, p_L^t < p_0^t \\ p_0^t, & \tilde{B}^t < 0, p_L^t \leq p_0^t < p_U^t \end{cases} \quad (5.26)$$

<sup>4</sup> Σημειώνεται πως με  $p_{E_{min}}^t$  συμβολίζεται η μοναδική λύση της εξίσωσης  $pL = r(h^t, p^t) \cdot E_{min}$  και με  $p_{E_{max}}^t$  αυτή της  $pL = r(h^t, p^t) \cdot E_{max}$

<sup>5</sup> Με  $p_0^t$  συμβολίζεται η μοναδική λύση της εξίσωσης  $-\tilde{B} \log_2\left(1 + \frac{hp}{\sigma}\right) - \frac{h}{(\sigma+hp)\ln 2}(V - \tilde{B}p) = 0$ .

Μόλις υπολογιστεί το  $p^{t*}$  εισάγεται στη σχέση (5.25) και παράγει την τιμή που εκφράζει το βέλτιστο κόστος εκτέλεσης στον MEC server, η οποία συμβολίζεται με  $J_s(p^{t*})$ . Το κόστος αυτό αποτελείται από το ζυγισμένο άθροισμα του χρόνου εκτέλεσης του task  $\frac{L}{r(h^t, p^t)}$  και της κατανάλωσης ενέργειας  $\frac{p^t L}{r(h^t, p^t)}$ .

### 5.4.3 Κώδικας MATLAB

Παρακάτω παρουσιάζεται ο κώδικας σε γλώσσα MATLAB [27] που υλοποιεί τον αλγόριθμο LODCO, Χρησιμοποιώντας τις μεθόδους και τα μοντέλα που περιεγράφηκαν στις προηγούμενες υποενότητες:

```

%% This script simulates the Lyapunov Optimization-based Dynamic Computation
Offloading (LODCO) algorithm.
% author: Hailiang Zhao
clc, clear
opt = optimset('Display', 'none');

%% basic parameter settings
k = 1e-28; % effective switched capacitance (a constant decided by the
chip architecture)
tau = 0.002; % the length of time slot (in second)
phi = 0.002; % the cost of task dropping (in second)
omega = 1e6; % the bandwidth of MEC server (in Hz)
sigma = 1e-13; % the noise power of the receiver (in W)
p_tx_max = 1; % the maximum transmit power of mobile device (in W)
f_max = 1.5e9; % the maximum CPU-cycle frequency of mobile device (in Hz)
E_max = 0.002; % the maximum amount of battery output energy (in J)
L = 1000; % the input size of the computation task (in bit)
X = 737.5; % the number of CPU cycles needed on processing one bit of
task
W = L * X; % the number of CPU cycles needed on processing one task
E_H_max = 48e-6; % the upper bound of the energy arrive at the mobile device
(in J)
p_H = E_H_max / (2*tau); % the average Energy Harvesting (EH) power (in W)
g0 = power(10, -4); % the path-loss constant

%% parameter control
T = 50000; % the number of time slot (a.k.a. the size of the time
horizon)
tau_d = 0.002; % execution deadline (in second)
d = 50; % the distance between the mobile device and the MEC server
(in meter)
E_min = 0.02e-3; % the minimum amount of battery output energy (in J)
V = 1e-5; % the weight of penalty (the control parameter introduced by
Lyapunov Optimization)
P_req_task = 0.6; % the probability that the computation task is
requested

% the lower bound of perturbation parameter
E_max_hat = min(max(k * W * (f_max)^2, p_tx_max * tau), E_max);
theta = E_max_hat + V * phi / E_min;

%% allocate storage for valuable results
B = zeros(T, 1); % the battery energy level (in J)
B_hat = zeros(T, 1); % the virtual battery energy level ($B_hat = B - theta$)
e = zeros(T, 1); % the amount of the harvested and stored energy (in J)
chosen_mode = zeros(T, 1); % {1: local, 2: remote, 3: drop, 4: no task request}
f = zeros(T, 1); % the CPU-cycle frequency of local execution (in Hz)
p = zeros(T, 1); % the transmit power of computation offloading (in W)
cost = zeros(T, 3); % execution delay for mobile execution, MEC server execution
and final choice, respectively (in second)

```

```

E = zeros(T, 3);           % energy consumption for mobile execution, MEC server
execution and final choice, respectively (in J)
%% simulation begin
t = 1;
while t <= T
    disp(['====> Time slot #', num2str(t), ' <===='])
    %% initialization
    zeta = binornd(1, P_req_task);
    B_hat(t) = B(t) - theta;
    %% step 1: get the optimal energy harvesting no matter whether task is requested
    E_H_t = unifrnd(0, E_H_max);
    if B_hat(t) <= 0
        e(t) = E_H_t;
    end
    %% step 2: get the optimal computation offloading strategy (I_m, I_s, I_d, f(t),
p(t))
    if zeta == 0
        % chosen mode has to be 4
        disp('no task request generated!')
        chosen_mode(t) = 4;
    else
        % chosen_mode is chosen from {1, 2, 3}
        disp('task request generated!')
        % task request exists, generate the channel power gain
        h = exprnd(g0 / power(d, 4));

        %% step 2.1: solve the optimization problem for mobile execution
        % calculate f_L and f_U
        f_L = max(sqrt(E_min / (k * W)), W / tau_d);
        f_U = min(sqrt(E_max / (k * W)), f_max);
        if f_L <= f_U
            disp('mobile execution ( $P_{ME}$ ) is feasible!')
            if B_hat(t) < 0
                f_0 = (V / (-2 * B_hat(t) * k))^(1/3);
            else
                f_0 = -(V / (2 * B_hat(t) * k))^(1/3);
            end

            if (f_0 > f_U && B_hat(t) < 0) || (B_hat(t) >= 0)
                f(t) = f_U;
            elseif f_0 >= f_L && f_0 <= f_U && B_hat(t) < 0
                f(t) = f_0;
            elseif f_0 < f_L && B_hat(t) < 0
                f(t) = f_L;
            end

            % calculate the delay of mobile execution
            cost(t, 1) = W / f(t);
            % calculate the energy consumption of mobile execution
            E(t, 1) = k * W * (f(t)^2);
            % calculate the value of optimization goal
            J_m = -B_hat(t) * k * W * (f(t))^2 + V * W / f(t);
        else
            % the sub-problem is not fasible because (i) the limited
            % computation capacity or (ii) time cosumed out of deadline or
            % (iii) the energy consumed out of battery energy level
            disp('mobile execution is not feasible!')
            f(t) = 0;
            cost(t, 1) = 0;
            E(t, 1) = 0;
            J_m = inf;
        end

        %% step 2.2: solve the optimization problem for server execution
        E_tmp = sigma * L * log(2) / (omega * h);
    end
end

```



```

p_L_taud = (power(2, L / (omega * tau_d)) - 1) * sigma / h;
% calculate p_L
if E_tmp >= E_min
    p_L = p_L_taud;
else
    % calculate p_E_min (use inline function and fsolve)
    y = @(x) x * L - omega * log2(1 + h*x/sigma) * E_min;
    % according to the function figure, p_L_taud is a positive
    % number around 0.2
    p_E_min = fsolve(y, 0.2, opt);
    p_L = max(p_L_taud, p_E_min);
end
% calculate p_U
if E_tmp >= E_max
    p_U = 0;
else
    % calculate p_E_max (use inline function and fsolve)
    y = @(x) x * L - omega * log2(1 + h*x/sigma) * E_max;
    % according to the function figure, p_E_max is a large positive
    % number around 20
    p_E_max = fsolve(y, 100, opt);
    p_U = min(p_tx_max, p_E_max);
end

if p_L <= p_U
    disp('MEC server execution is feasible!')
    virtual_battery = B_hat(t);
    y = @(x) virtual_battery * log2(1 + h*x/sigma) + ...
        h * (V - virtual_battery*x) / log(2) / (sigma + h*x);
    p_0 = fsolve(y, 0.5, opt);

    if (p_U < p_0 && B_hat(t) < 0) || B_hat(t) >= 0
        p(t) = p_U;
    elseif p_0 < p_L && B_hat(t) < 0
        p(t) = p_L;
    elseif p_0 >= p_L && p_0 <= p_U && B_hat(t) < 0
        p(t) = p_0;
    end

    % calculate the delay of MEC server execution
    cost(t, 2) = L / (omega * log2(1 + h*p(t)/sigma));
    % calculate the energy consumption of MEC server execution
    E(t, 2) = p(t) * cost(t, 2);
    % calculate the value of optimization goal
    J_s = (-B_hat(t) * p(t) + V) * cost(t, 2);
else
    % the sub-problem is not feasible because (i) the limited transmit
    % power or (ii) time cosumed out of deadline or (iii) the energy
    % consumed out of battery energy level
    disp('MEC server execution ( $P_{SE}$ ) is not feasible!')
    p(t) = 0;
    cost(t, 2) = 0;
    E(t, 2) = 0;
    J_s = inf;
end

%% step 3: choose the best execution mode
J_d = V * phi;
disp(['J_m:', num2str(J_m)])
disp(['J_s:', num2str(J_s)])
[~, mode] = min([J_m, J_s, J_d]);
chosen_mode(t) = mode;
end

```

```

    % step 4: according to the chosen execution mode, calculate the real delay and
energy consumption
    if chosen_mode(t) == 1
        % mobile execution is chosen
        cost(t, 3) = cost(t, 1);
        E(t, 3) = E(t, 1);
    elseif chosen_mode(t) == 2
        % MEC server execution is chosen
        cost(t, 3) = cost(t, 2);
        E(t, 3) = E(t, 2);
    elseif chosen_mode(t) == 3
        % task is dropped, the delay is the task dropping penalty and the
        % energy consumption is zero
        cost(t, 3) = phi;
        E(t, 3) = 0;
    else
        % no task is requested, the delay and the energy consumption are
        % both zero
        cost(t, 3) = 0;
        E(t, 3) = 0;
    end

    %% step 5: update the battery energy level and go to next time slot
    B(t + 1) = B(t) - E(t, 3) + e(t);
    t = t + 1;
end

%% step 6: evaluate the simulation results
% 1. the battery energy level vs. time slot
figure
plot(1:T, B(1:T));
hold on
plot(1:T, repmat(theta + E_H_max, [T, 1]), '-')
title('Envolution of battery energy level')
xlabel('time slot')
ylabel('battery energy level $B_t$', 'Interpreter','latex')

% 2. the average execution cost vs. time slot
accumulated = 0;
average_cost = zeros(T, 1);
request_num = 0;
for t = 1: T
    accumulated = accumulated + cost(t, 3);
    if cost(t, 3) ~= 0
        % there exists task request
        request_num = request_num + 1;
    end
    average_cost(t) = accumulated / request_num;
end
figure
plot(1:T, average_cost);
title('Envolution of average execution cost')
xlabel('time slot')
ylabel('average execution cost $\frac{1}{T} \sum_{t=0}^{T-1} cost^t$',
'Interpreter','latex')

% 3. the average ratio of each chosen mode vs. time slot
average_ratio = zeros(T, 3);
mobile_exe = 0; server_exe = 0; drop = 0;
request_num = 0;
for t = 1: T
    if cost(t, 3) == 0
        continue
    else
        request_num = request_num + 1;
    end
end

```

```

if chosen_mode(t) == 1
    mobile_exe = mobile_exe + 1;
elseif chosen_mode(t) == 2
    server_exe = server_exe + 1;
else
    drop = drop + 1;
end
end
average_ratio(t, :) = [mobile_exe, server_exe, drop] / request_num;
end
figure
plot(1:T, average_ratio(:, 1));
hold on
plot(1:T, average_ratio(:, 2));
hold on
plot(1:T, average_ratio(:, 3));
legend('mobile execution', 'MEC server execution', 'drop')
title('Envolution of average ratio of chosen modes')
xlabel('time slot')
ylabel('average ratio of chosen modes $\frac{1}{T} \sum_{t=0}^{T-1} \{I_m^t, I_s^t, I_d^t\}$', 'Interpreter','latex')

```

Τα γραφήματα που παράγει το πρόγραμμα αφορούν τρία βασικά χαρακτηριστικά του μοντέλου:

1. Το επίπεδο της μπαταρίας
2. Το μέσο κόστος εκτέλεσης task
3. Το μέσο ποσοστό επιλογής κάθε  $I_i^t$ ,  $i \in \{m, s, d\}$

Κάθε ένα από αυτά έχει ως μεταβλητή αναφοράς το χρόνο. Επιλέγεται ο αλγόριθμος να τρέξει για 50.000 χρονικές «περιόδους» (**time slots**). Η πιθανότητα άφιξης task σε οποιοδήποτε time slot τίθεται στο 60%. Οι τιμές που αποδίδονται στις υπόλοιπες παραμέτρους του προγράμματος προκύπτουν από τις αντίστοιχες τιμές που αναφέρονται στο κεφάλαιο VI. *Simulation Results* του [22] και παρουσιάζονται συνοπτικά στον Πίνακα 5:

Πίνακας 5.1: Αρχικοποίηση παραμέτρων αλγορίθμου LODCO		
Συμβολισμός	Μέγεθος	Τιμή
$\kappa$	Effective switch capacitance	$10^{-28}$
$\tau$	Μήκος χρονικού διαστήματος	0,002 sec
$\varphi$	Κόστος απόρριψης task	0,002 sec
$\omega$	Εύρος ζώνης MEC server	$10^6$ Hz
$\sigma$	Ισχύς θορύβου στον δέκτη	$10^{-13}$ Watt
$p_{tx}^{max}$	Μέγιστη ισχύς μετάδοσης συσκευής	1 Watt
$f_{max}$	Μέγιστη συχνότητα κύκλου ρολογιού συσκευής	$1,5 \cdot 10^9$ Hz
$E_{max}$	Μέγιστη τιμή μπαταρίας συσκευής	0,002 Joule
$L$	Μέγεθος εισερχόμενων tasks	1000 bits
$X$	Αριθμός κύκλων ρολογιού για την επεξεργασία ενός bit πληροφορίας	737,5
$E_H^{max}$	Άνω όριο εισερχόμενης ενέργειας	$48 \cdot 10^{-6}$ Joule
$g_0$	Σταθερά απώλειας διαδρομής	$10^{-4}$
$T$	Αριθμός χρονικών εποχών	50000
$\tau_d$	Διορία χρόνου εκτέλεσης	0,002 sec
$d$	Απόσταση μεταξύ συσκευής και MEC server	50 m
$V$	Παράμετρος ελέγχου βελτιστοποίησης Lyapunov	$\text{Joule}^2 \cdot \text{sec}^{-1}$

$E_{min}$	Ελάχιστη τιμή επιπέδου μπαταρίας	$0,02 \cdot 10^{-3}$
$P\{z^t = 1\}$	Πιθανότητα άφιξης task	0,6

Για την εύρεση των  $p_{E_{min}}^t, p_{E_{max}}^t, p_0^t$  απαιτείται η εύρεση μοναδικής λύσης σε ένα σύνολο μη γραμμικών εξισώσεων. Για το λόγο αυτό, το MATLAB διαθέτει την συνάρτηση `fsolve` [33], η οποία δέχεται σαν ορίσματα μια συνάρτηση, ένα σημείο ή σύνολο σημείων και μια μέθοδο επίλυσης.

## 5.5 Αποτελέσματα προσομοίωσης

Ο κώδικας της υποενότητας 5.4.3 παράγει τρεις γραφικές παραστάσεις που αφορούν την εξέλιξη τριών βασικών παραμέτρων του αλγορίθμου στο χρόνο:

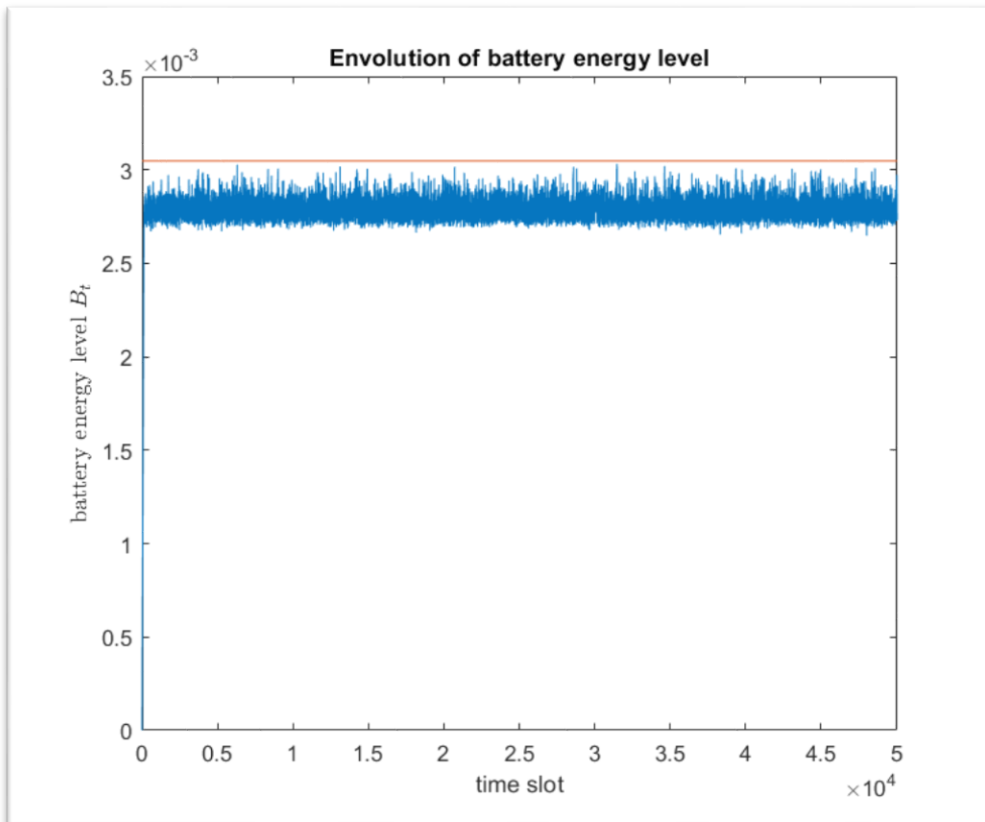
1. Το **επίπεδο της μπαταρίας**
2. Την **επιλογή μεθόδου εκτέλεσης task**
3. Το **μέσο συνολικό κόστος εκτέλεσης**

Η αξιολόγηση των μετρικών αυτών μπορεί να οδηγήσει σε κάποια πρώτα συμπεράσματα για την απόδοση του μοντέλου που προτείνεται από τον αλγόριθμο LODCO καθώς και να αναδείξει την πρακτική του χρησιμότητα.

### 5.5.1 Εξέλιξη επιπέδου μπαταρίας

Στο Διάγραμμα 5.1 παρουσιάζεται η εξέλιξη του επιπέδου μπαταρίας της κινητής συσκευής  $B_t$  στη διάρκεια των 50000 χρονικών slots. Παρατηρείται πως σε κάθε χρονική περίοδο εμφανίζονται αυξομειώσεις της μπαταρίας μεταξύ περίπου των τιμών 0.0027 και 0.003, ενώ το μέγιστο άνω όριο των 0.00348 J δεν παραβιάζεται. Η διακύμανση της μπαταρίας είναι αρκετά μικρή σε απόλυτο μέγεθος, καθώς το μοντέλο του Energy Harvesting φροντίζει, ώστε όταν η μπαταρία δείχνει να μειώνεται σε ένα κρίσιμο σημείο να απορροφάται κομμάτι της διαθέσιμης ενέργειας από το περιβάλλον. Φαίνεται λοιπόν ο πρώτος στόχος του μοντέλου, δηλαδή η επίλυση του προβλήματος της ταχείας μείωσης του επιπέδου της μπαταρίας λόγω εκτέλεσης διεργασιών από τη συσκευή, να επιτυγχάνεται.

Επιπλέον σημαντικό είναι το γεγονός ότι το επίπεδο της της διαθέσιμης ενέργειας δεν εμφανίζει ακραίες διακυμάνσεις. Αντίθετα, κατά μέσο όρο διατηρεί μια τιμή κοντά στο μέγιστο επίπεδο φόρτισης. Με αυτό τον τρόπο όχι μόνο δίνει τη δυνατότητα στον αλγόριθμο LODCO να δρομολογεί την εκτέλεση των tasks με βέλτιστο τρόπο αλλά επιπλέον διατηρεί τη γενική απόδοση της συσκευής σε υψηλά επίπεδα, βελτιώνοντας σημαντικά το QoE του χρήστη.



**Διάγραμμα 5.1: Battery energy level**

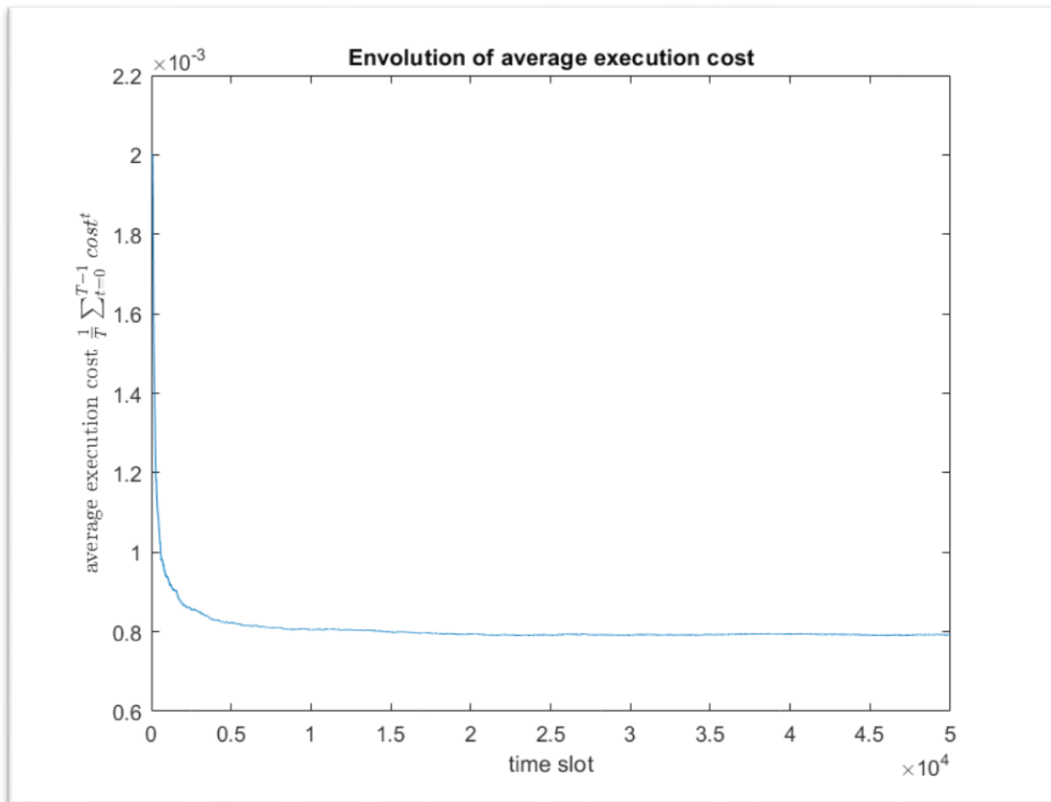
### **5.5.2 Μέσο συνολικό κόστος εκτέλεσης**

Το κόστος εκτέλεσης για κάθε δεδομένη χρονική στιγμή εξαρτάται άμεσα από την επιλογή σεναρίου εκτέλεσης task, καθώς και από την ίδια την ύπαρξη task προς εκτέλεση. Τα επιμέρους κόστη σε κάθε περίπτωση είναι τα εξής:

1. Κόστος **mobile execution**:  $\frac{W}{f^t}$
2. Κόστος **server execution**:  $\frac{L}{r(h^t, p^t)}$
3. Κόστος **task dropping**:  $\varphi$

Τα κόστη αυτά αθροίζονται και έπειτα διαιρούνται με τη χρονική περίοδο της εκτέλεσης του αλγορίθμου ώστε να προκύψει το μέσο συνολικό κόστος εκτέλεσης task. Στο σημείο αυτό αξίζει να σημειωθεί πως χρονικές στιγμές στις οποίες δεν παρατηρείται άφιξη task προς εκτέλεση δεν λαμβάνονται υπόψιν στον υπολογισμό, καθώς το κόστος στην περίπτωση αυτή είναι μηδενικό.

Με βάση τα παραπάνω, το μέσο κόστος εκτέλεσης παράγεται από τον τύπο  $\frac{1}{T} \sum_{t=0}^{T-1} cost^t$  και η χρονική του εξέλιξη φαίνεται στο Διάγραμμα 5.2. Παρατηρείται πως η μείωσή του είναι εκθετική από τις πρώτες χρονικές στιγμές, καθώς από τη μέγιστη τιμή του (2ms) σταθεροποιείται στην ελάχιστη τιμή των περίπου 0.8ms μετά από μόλις 500 time slots. Επομένως, το συμπέρασμα που εκπονείται από αυτή την παρατήρηση είναι ότι ο στόχος του μοντέλου για ελαχιστοποίηση του χρόνου εκτέλεσης ενός task μέσω του δυναμικού τρόπου offloading που εισάγει ο αλγόριθμος LODCO περατώνεται με επιτυχία.



**Διάγραμμα 5.2: Μέσο κόστος εκτέλεσης**

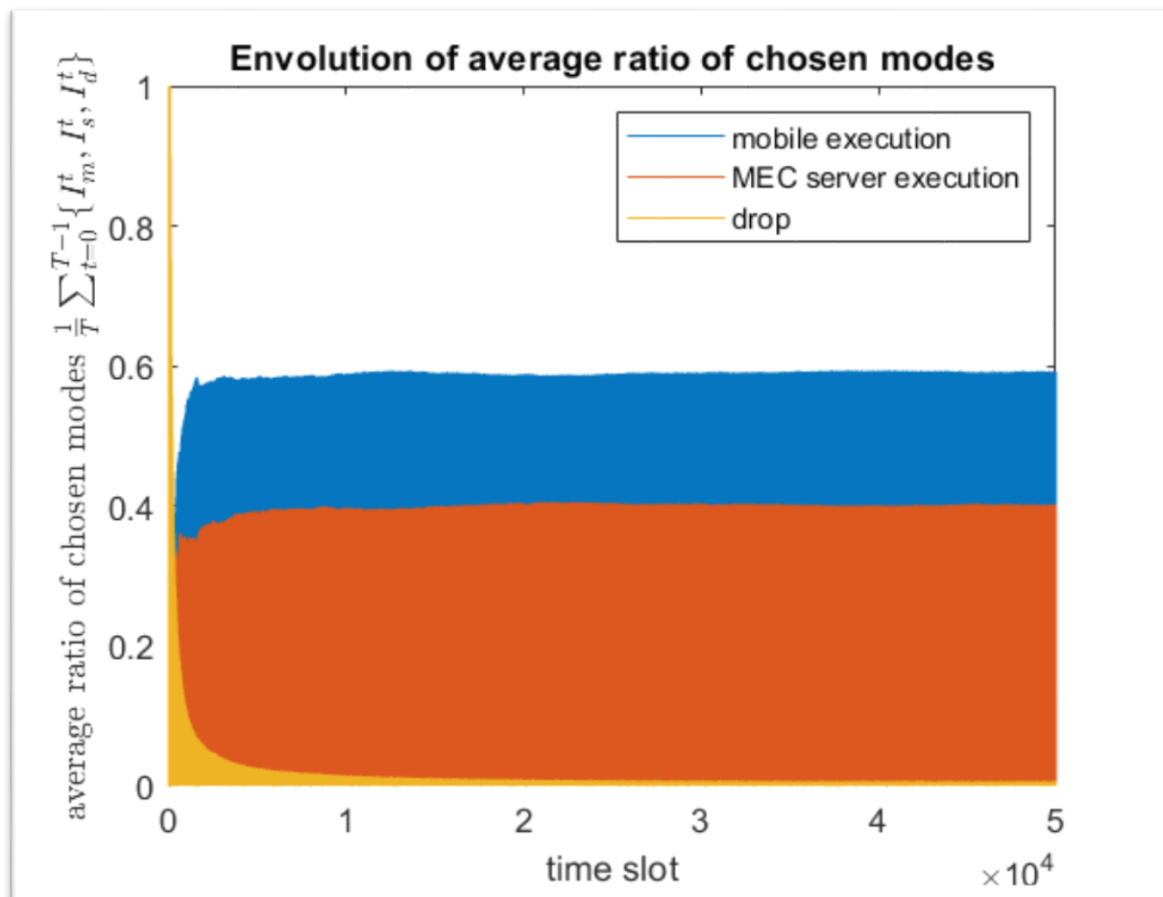
### **5.5.3 Μέση επιλογή μεθόδου εκτέλεσης**

Στο Διάγραμμα 5.3 παρουσιάζεται η κατανομή των tasks στα τρία πιθανά σενάρια που προβλέπει ο αλγόριθμος LODCO (**mobile execution, server execution, task dropped**). Η πρώτη σημαντική παρατήρηση αφορά το ποσοστό απόρριψης task από το μοντέλο. Φαίνεται ότι το μοντέλο επιτυγχάνει στο να περιορίσει σχεδόν σταθερά τις ανεπιτυχείς εκτελέσεις διεργασιών ακόμα και στο 0.08%. Το αποτέλεσμα αυτό είναι αναμενόμενο καθώς το μέσο κόστος εκτέλεσης είναι αρκετά χαμηλότερα από το άνω όριο  $\tau_a$ , ενώ τα επίπεδα της μπαταρίας βρίσκονται κατά μέσο όρο πάνω από την ελάχιστη απαιτούμενη τιμή για την εκτέλεση ενός task. Το συμπέρασμα που προκύπτει είναι πως η αξιοπιστία του μοντέλου σχετικά με την περάτωση μιας διεργασίας ανεξαρτήτως μεθόδου είναι αρκετά υψηλή και ως εκ τούτου βελτιώνει το QoE των χρηστών.

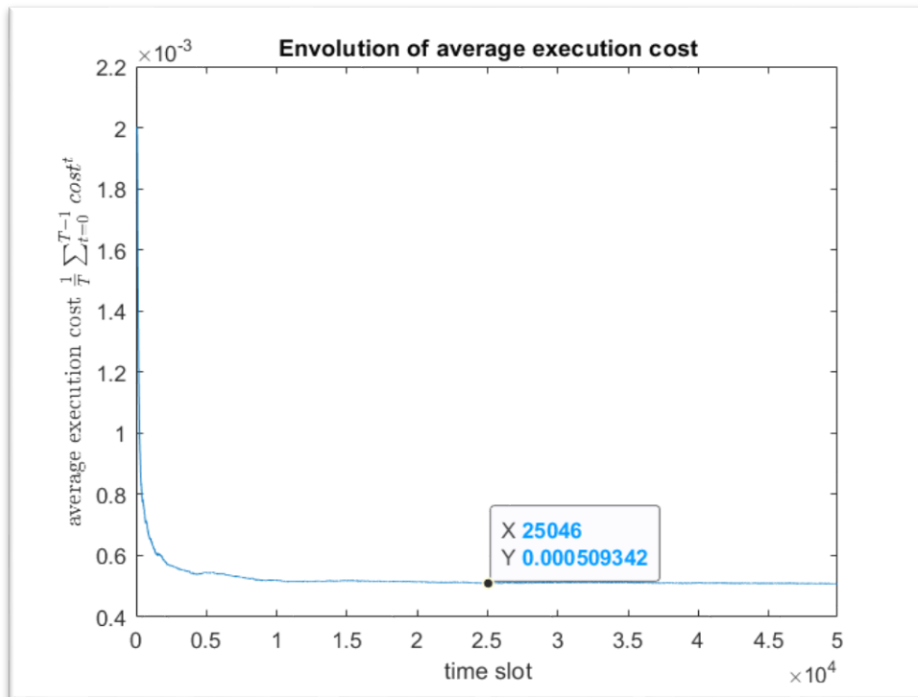
Η δεύτερη σημαντική παρατήρηση που πηγάζει από τη μελέτη του Διαγράμματος 5.3 είναι πως ο αλγόριθμος δείχνει να προτιμά στη μέση περίπτωση την εκτέλεση εντός της συσκευής, αντί για την ανάθεσή της στον MEC server. Η διαφορά αυτή βέβαια δεν είναι σημαντική, καθώς η επιλογή των δύο μεθόδων κυμαίνεται περίπου στο 60/40% υπέρ της τοπικής εκτέλεσης. Προφανώς η παράμετρος που παίζει το σημαντικότερο ρόλο στην επιλογή τρόπου εκτέλεσης είναι η απόσταση από τον MEC server. Για παράδειγμα, αν τεθεί  $d=40m$  και εκτελεστεί εκ νέου ο αλγόριθμος παρατηρείται πως η διαφορά των 10 μόλις μέτρων οδηγεί στην εύνοια του server execution, όπως φαίνεται και στο Διάγραμμα 5.5 (περίπου 66% προτίμηση).

Επιπροσθέτως, η εγγύτητα στον MEC server δείχνει να ευνοεί και το μέσο κόστος εκτέλεσης το οποίο στο Διάγραμμα 5.4 φαίνεται να σταθεροποιείται στην τιμή των 0.5ms, απόδοση καλύτερη από αυτή που υπολογίστηκε για  $d=50m$ . Η διαφορά αυτή είναι απολύτως αναμενόμενη, καθώς ένας MEC server διαθέτει εγγενώς ισχυρότερους υπολογιστικούς πόρους από μια μεμονωμένη κινητή συσκευή και ως εκ τούτου μπορεί να φέρει εις πέρας κατά μέσο όρο ένα task σε λιγότερο χρόνο.

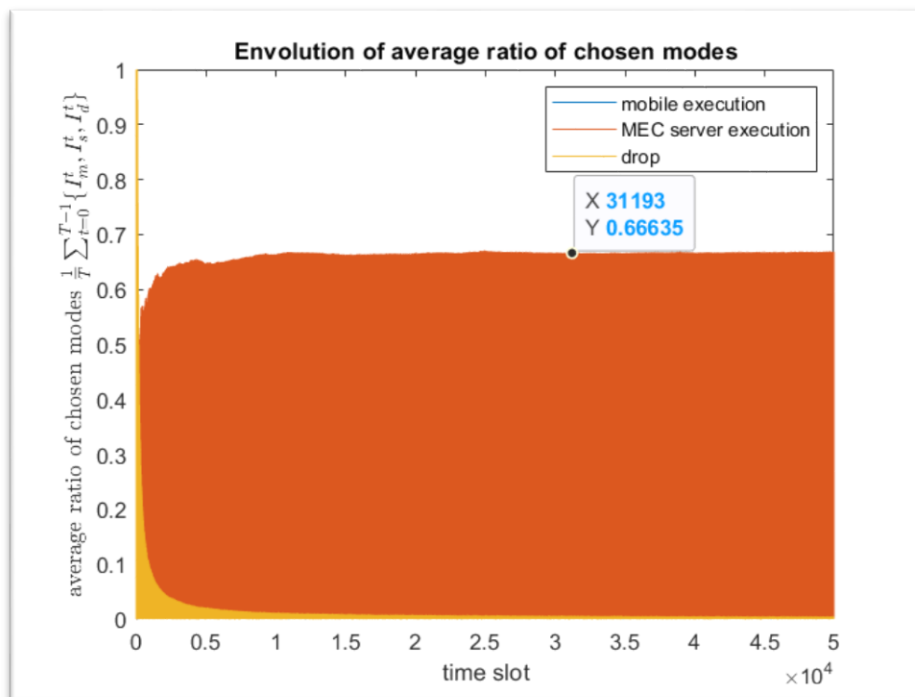
Το γεγονός πως όσο πλησιάζει ο χρήστης στον server η εκτέλεση σε αυτόν ευνοείται δείχνει πως το μοντέλο του αλγορίθμου προσαρμόζεται δυναμικά και με βέλτιστο τρόπο στις αλλαγές των εξωγενών παραγόντων όπως η θέση του χρήστη. Ο αλγόριθμος επιτυγχάνει στο να βρίσκει σε κάθε χρονική στιγμή την σωστή αναλογία mobile/server execution ώστε ο χρήστης να παρατηρεί την μικρότερη δυνατή καθυστέρηση στην εκτέλεση μιας εφαρμογής, βελτιώνοντας έτσι το QoE του.



**Διάγραμμα 5.3: Average ratio of chosen modes,  $d=50m$**



**Διάγραμμα 5.4: Average ratio of chosen modes,  $d=40m$**



**Διάγραμμα 5.5: Average ratio of chosen modes,  $d=40m$**



# ΚΕΦΑΛΑΙΟ 6°

## ΣΥΜΠΕΡΑΣΜΑΤΑ

### 6.1 Γενικό σχόλιο

Η τεχνολογία του Mobile Edge Computing φαίνεται πως αποτελεί μια προσέγγιση, που σε συνδυασμό με την σταδιακή εγκαθίδρυση του 5G σαν το κυρίαρχο δίκτυο κινητών υπηρεσιών θα φέρει μια νέα εποχή στις τηλεπικοινωνίες. Οι ολοένα και αυξανόμενες απαιτήσεις των εφαρμογών που εκτελούνται μέσω του Διαδικτύου καλούν για ανάπτυξη νέων μεθόδων, ώστε να επιτευχθεί όσο το δυνατόν καλύτερη εκμετάλλευση των πόρων που διαθέτει η αρχιτεκτονική των 5G δικτύων. Για το λόγο αυτό, ένα από τα σημαντικότερα πεδία έρευνας αποτελεί η διαδικασία δρομολόγησης mobile task εντός του συστήματος του Mobile Edge Computing.

Όπως περιεγράφηκε σε προηγούμενο κεφάλαιο, τα πλεονεκτήματα της χρήσης του ορίου των δικτύων κινητής τηλεφωνίας ως πλατφόρμα είναι πολυάριθμα και στοχεύουν στην βελτίωση του QoE των χρηστών. Είναι λογικό να παρίσταται η ανάγκη εφαρμογής μεθόδων που θα εκμεταλλεύονται τα πλεονεκτήματα αυτά με βέλτιστο τρόπο. Ακολουθώντας την κατεύθυνση αυτή, διαπιστώνεται πως η δρομολόγηση εφαρμογών που εκτελούνται εντός της πλατφόρμας του MEC αποτελεί ένα κρίσιμο πεδίο έρευνας. Η αναζήτηση δυναμικών τεχνικών που επιτυγχάνουν στο να βελτιστοποιούν τη χρήση του MEC μέσω της αποδοτικής δρομολόγησης διεργασιών εντός και εκτός του απασχολεί τα τελευταία χρόνια την επιστημονική βιβλιογραφία, η οποία παρουσιάζει ενδιαφέρουσες προτάσεις επί του ζητήματος.

Στην παρούσα εργασία παρουσιάστηκαν και αναλύθηκαν δύο εκ των προαναφερθέντων προτάσεων [4], [22]. Η πρώτη αφορούσε την αξιολόγηση τεσσάρων αλγορίθμων δρομολόγησης task εντός ενός συστήματος από MEC servers ενσωματωμένους εντός των eNBs ενός 5G δικτύου, ενώ η δεύτερη την εισαγωγή ενός δυναμικού αλγορίθμου που μεταξύ δύο σεναρίων εκτέλεσης επιλέγει ανά πάσα χρονική στιγμή τη βέλτιστη στρατηγική. Κάθε μια εκ των δύο μεθόδων παρουσιάζει τα δικά της πλεονεκτήματα, τα οποία μπορούν να αναδειχθούν, αν η εφαρμογή τους πραγματοποιηθεί στο κατάλληλο περιβάλλον.

### 6.2 Σκοπός της μελέτης

Βασικός σκοπός του συγγραφέως αποτέλεσε η παρουσίαση του περιβάλλοντος MEC στο οποίο ενθυλακώνονται οι αλγόριθμοι που μελετήθηκαν και η περιγραφή της λειτουργίας τους εντός αυτού. Η επιλογή των συγκεκριμένων μεθόδων έγινε μετά από εκτενή μελέτη της σχετικής βιβλιογραφίας, καθώς θεωρήθηκε πως προσφέρουν μια καινοτόμο προσέγγιση στο ζήτημα της δρομολόγησης διεργασιών σε MEC servers. Επιπλέον, σημαντικό κριτήριο για την επιλογή τους αποτέλεσε η προσαρμοστικότητά που επιδεικνύουν σε πιθανή εφαρμογή τους σε πρακτικές, real-life εφαρμογές όπως θα αναφερθεί παρακάτω. Τέλος, σημαντικό ρόλο στην συγγραφή της παρούσας εργασίας, διαδραμάτισε η ανάγκη για έμφαση στο γεγονός ότι η τεχνολογία του MEC σε συνδυασμό με την ραγδαία ανάπτυξη των 5G δικτύων, μπορεί να οδηγήσει στην ανάπτυξη τεχνολογιών που θα φέρουν σημαντικές εξελίξεις σε καιρία πεδία της επιστήμης των τηλεπικοινωνιών.

### 6.3 Σύγκριση των δύο προσεγγίσεων

Με μια πρώτη ανάγνωση, φαίνεται πως οι μέθοδοι των [4], [22] στοχεύουν αμφότεροι στη βέλτιστη εκμετάλλευση μιας πλατφόρμας MEC, με σκοπό την παροχή υπηρεσιών στους χρήστες που χαρακτηρίζονται από αξιοπιστία και προσαρμοστικότητα. Όπως αναφέρθηκε και στα επιμέρους κεφάλαια που παρουσίασαν τις δύο προσεγγίσεις, τελικό κριτήριο αποδοτικότητας υπήρξε το QoE του χρήστη. Το συμπέρασμα που προέκυψε είναι πως και στις δύο περιπτώσεις ο στόχος επετεύχθη, θέτοντας όμως στην κάθε μια διαφορετικά κριτήρια μέτρησης της αποτελεσματικότητάς τους.

### **6.3.1 Μοντελοποίηση UEs και πλατφόρμας MEC**

Αρχικά, η ειδοποιός διαφορά που συναντάται μεταξύ των δύο paper αφορά το μοντέλο των χρηστών. Στην περίπτωση του [22], οι χρήστες είναι πολυάριθμοι και κινούνται στον χώρο με ένα συγκεκριμένο μοτίβο. Αντίθετα, ο αλγόριθμος LODCO θεωρεί έναν μοναδικό χρήστη, ο οποίος βρίσκεται σε ορισμένη απόσταση από τον MEC server και η κίνησή του στο χώρο δε λαμβάνεται υπόψιν. Επιπλέον, ο αλγόριθμος LODCO θεωρεί σύστημα με έναν και μοναδικό διαθέσιμο MEC server, επομένως σε περίπτωση απόρριψης task από αυτόν δεν υπάρχει η δυνατότητα μετάβασης σε άλλο διαθέσιμο server. Αντίθετα, το μοντέλο του [22] όχι μόνο διαθέτει πολλαπλούς MEC server που δημιουργούν επιμέρους MEC clouds, αλλά επιπλέον λαμβάνει υπόψιν την τοπολογία που ακολουθούν. Τέλος, το μοντέλο του MEC server που περιγράφει ο αλγόριθμος LODCO αντιλαμβάνεται τα tasks που καταφθάνουν σε αυτόν ως bits δεδομένων και τα διαχειρίζεται αναλόγως, ενώ οι αλγόριθμοι του [22] χρησιμοποιούν ένα απλό μοντέλο αφίξεων και εξυπηρέτησης που χαρακτηρίζονται από συγκεκριμένες κατανομές και μοντελοποιούν τα tasks ως αιτήματα προς εξυπηρέτηση προσομοιάζοντας μια ουρά M/M/n.

### **6.3.2 Energy Harvesting και προσαρμοστικότητα μοντέλων**

Μια σημαντική παράμετρος που εισάγεται από τον αλγόριθμο LODCO είναι το επίπεδο διαθέσιμης ενέργειας στην μπαταρία της συσκευής του χρήστη. Η μεταβλητή αυτή συνδέεται άμεσα με τη δυνατότητα ενός οποιουδήποτε συστήματος να διαχειριστεί ένα αίτημα προς εξυπηρέτηση, ανεξαρτήτως μεθόδου διαχείρισης. Το μοντέλο του EH που περιγράφεται στην υποενότητα 5.4.1 διασφαλίζει στους χρήστες πως το πρόβλημα απόρριψης μιας διεργασίας λόγω έλλειψης διαθέσιμης ενέργειας αντιμετωπίζεται σε σημαντικό βαθμό. Αντιθέτως, κανένας από τους τέσσερις αλγορίθμους του [22] δεν επεξεργάζεται αυτή την παράμετρο.

Ακόμη μια διαφορά μεταξύ των δύο προσεγγίσεων εντοπίζεται στο μέσο επιλογής για την εκτέλεση ενός task. Οι αλγόριθμοι του [22] αναθέτουν τα αιτήματα εξυπηρέτησης εξολοκλήρου στους MEC servers. Ο αλγόριθμος LODCO επιτρέπει αντιθέτως την εκτέλεση εντός της συσκευής του χρήστη, αν αυτό κρίνεται αποδοτικότερο. Προφανώς, το δεύτερο μοντέλο δείχνει μεγαλύτερη προσαρμοστικότητα όσο αφορά τον τρόπο εκτέλεσης, καθώς σε κάθε περίπτωση αντιμετωπίζει το πρόβλημα της δρομολόγησης ενός task εξετάζοντας δύο διαφορετικές εναλλακτικές μεθόδους και επιλέγοντας κάθε φορά την καταλληλότερη.

Εξετάζοντας το ζήτημα της προσαρμοστικότητας των μοντέλων από τη σκοπιά της τοπολογίας του περιβάλλοντος, το συμπέρασμα που προκύπτει είναι πως ο αλγόριθμος LODCO υστερεί σε σχέση με το μοντέλο του [22]. Αυτό συμβαίνει διότι, όπως περιεγράφηκε στην υποενότητα 4.2.1, οι αλγόριθμοι SOL και LFS παρουσιάζουν αποτελέσματα που προσαρμόζονται καταλλήλως στην κατανομή των MEC servers στο περιβάλλον. Η ιδιότητα αυτή δεν συναντάται στις μεθόδους του [20], καθώς η πλατφόρμα MEC αποτελείται από έναν και μοναδικό server.

### **6.3.3 Μέτρα αξιολόγησης**

Κάθε προσέγγιση ορίζει μια ή περισσότερες παραμέτρους, βάσει των οποίων θα αξιολογηθούν τα πειραματικά τους αποτελέσματα. Οι προδιαγραφές αυτές, έχουν ως σκοπό να καθορίσουν το βαθμό στον οποίο το κάθε μοντέλο πετυχαίνει τους στόχους που έχουν τεθεί κατά την ανάπτυξή του. Προφανώς, κάθε τέτοιο μέτρο αξιολόγησης αποτελεί μια μετρική που καθορίζει το βαθμό βελτίωσης του QoE των χρηστών.

Στο μοντέλο του [4], οι δύο παράμετροι που χρησιμοποιούνται για να αξιολογηθεί η αξιοπιστία του συστήματος είναι οι  $P_F, P_H$ . Οι δύο αυτές πιθανότητες εκφράζουν τα ενδεχόμενα (α) να απορριφθεί εξ αρχής ένα αίτημα προς εξυπηρέτηση και (β) να απορριφθεί η εκτέλεση ενός task, ενώ ήδη γίνεται προσπάθεια περάτωσης της. Το βασικό μέτρο αξιολόγησης της αξιοπιστίας του μοντέλου αφορά την επιτυχή (ή μη) ολοκλήρωση των αιτημάτων προς εξυπηρέτηση.

Στο μοντέλο του αλγορίθμου LODCO, η αξιολόγηση των αποτελεσμάτων είναι πιο πολυεπίπεδη. Στόχος του αλγορίθμου είναι η ελαχιστοποίηση μιας έκφρασης που αφορά το συνολικό κόστος εκτέλεσης μιας εφαρμογής από το σύστημα. Όπως περιεγράφηκε στην υποενότητα 5.5.2, το κόστος αυτό εμπερικλείει τόσο το ενδεχόμενο απόρριψης ενός task από το σύστημα όσο και την δυναμική καθυστέρηση που εισάγεται από την εκάστοτε μέθοδο εκτέλεσης που επιλέγεται. Αντιθέτως, οι αλγόριθμοι δρομολόγησης του [4] θεωρούν πως η χρονική περάτωση ενός task είναι καθορισμένη και εξαρτάται από τον ρυθμό εξυπηρέτησης  $\mu$ , καθώς η εκτέλεση λαμβάνει χώρα πάντα εντός ενός MEC server.

## 6.4 Τελικά συμπεράσματα και προτάσεις

Οι δύο προσεγγίσεις που παρουσιάστηκαν στην παρούσα εργασία έδειξαν σημαντικά αποτελέσματα στο ζήτημα της δρομολόγησης tasks εντός ενός συστήματος MEC. Η κάθε μια χρησιμοποίησε διαφορετικές παραμέτρους μοντελοποίησης και αξιολόγησης των αποτελεσμάτων της, αλλά το κοινό τους στοιχείο δείχνει να είναι πως λειτουργούν αποτελεσματικά στις συνθήκες που θέτουν. Η πλατφόρμα του Mobile Edge Computing, όπως αναλύθηκε εκτενώς σε προηγούμενες ενότητες, προσφέρει ένα low-latency, context-aware περιβάλλον για την ανάπτυξη πλειάδας πρακτικών εφαρμογών (βλ. ενότητα 1.4). Τα μοντέλα των [22], [4] μπορούν, εκμεταλλευόμενα τα επιμέρους πλεονεκτήματα που προσφέρουν, να ενισχύσουν περαιτέρω την απόδοση τέτοιων εφαρμογών.

Ο αλγόριθμος LODCO προσφέρει για μια δεδομένη μη μεταβαλλόμενη απόσταση από έναν MEC server τη βέλτιστη αναλογία mobile/server execution. Σε εφαρμογές όπου ο χρήστης στέκεται για μεγάλη χρονική διάρκεια ακίνητος η ενσωμάτωσή του LODCO σε ένα σύστημα MEC δείχνει να είναι μια αποδοτική στρατηγική. Για παράδειγμα, έστω μια εφαρμογή AR η οποία προσφέρει τη δυνατότητα σε έναν θεατή ενός αθλητικού αγώνα να δέχεται στο κινητό του πληροφορίες που αφορούν τα στατιστικά ενός παίκτη του γηπέδου. Δεδομένου πως η θέση του χρήστη είναι δεδομένη για σχεδόν όλη τη διάρκεια του αγώνα, ένα MEC σύστημα που χρησιμοποιεί τον αλγόριθμο σαν μέθοδο δρομολόγησης θα εμφανίζει υψηλές επιδόσεις, βελτιώνοντας την εμπειρία του θεατή. Όπως αναφέρθηκε άλλωστε και στην υποενότητα 1.4.1, οι AR εφαρμογές ευνοούνται σημαντικά από low-latency συστήματα, επομένως η ελαχιστοποίηση της καθυστέρησης που προβλέπει το μοντέλο του [22] αποτελεί άλλο ένα επιχείρημα υπέρ της εφαρμογής του. Επιπροσθέτως, η ίδια παρατήρηση μπορεί να εφαρμοστεί στην περίπτωση διασύνδεσης οικιακών συσκευών μέσω του IoT. Οι συσκευές αυτές προφανώς διαθέτουν συγκεκριμένη θέση εντός μιας κατοικίας, επομένως ένα MEC σύστημα που εφαρμόζει τον αλγόριθμο LODCO μπορεί να τις ευνοήσει με τον ίδιο τρόπο που μόλις περιεγράφηκε.

Οι αλγόριθμοι δρομολόγησης<sup>6</sup> του [4] επιτυγχάνουν στη δημιουργία ενός περιβάλλοντος υψηλής αξιοπιστίας για χρήστες που βρίσκονται εν κινήσει εντός μιας περιοχής που καλύπτεται από ένα 5G δίκτυο. Επομένως, η εφαρμογή τους σε ένα σύστημα Interconnected Smart Cars (βλ. υποενότητα 1.4.3), όπου οι χρήστες βρίσκονται σε διαρκή κίνηση και έχουν την ανάγκη για αξιόπιστες υπηρεσίες εντός ενός διαρκώς μεταβαλλόμενου περιβάλλοντος, φαίνεται να αποτελεί μια αποδοτική στρατηγική. Τέλος, η αξιοπιστία που

<sup>6</sup> Κυρίως οι SOL, LFS.

προσφέρει το μοντέλο του [4] σε κινούμενες συσκευές μπορεί ενδεχομένως να ευνοήσει εφαρμογές έξυπνης επιτάχυνσης βίντεο, καθώς όλο και περισσότεροι χρήστες χρησιμοποιούν streaming services όπως το Netflix ή το Amazon Prime εντός κινούμενων μέσων.

Συμπερασματικά, φαίνεται πως τα δύο μοντέλα μπορούν να προσφέρουν σημαντικές υπηρεσίες πάνω στο αντικείμενο της δρομολόγησης διεργασιών εντός μιας MEC πλατφόρμας. Επομένως, η περαιτέρω έρευνα και ανάπτυξη πάνω στο θέμα της πρακτικής ενσωμάτωσής τους εντός της αρχιτεκτονικής του MEC κρίνεται σημαντική και πολλά υποσχόμενη για τον τομέα των τηλεπικοινωνιών.

## **Βιβλιογραφία**

- [1] ETSI, *5G, NR, Overall Description, Stage-2, 3GPP TS 38.300 version 15.8.0 Release 15*, ETSI, 2020.
- [2] S. Kekki, W. Featherstone, Y. Fang, K. Pekka, A. Li, F. Jiangping, D. Frydman, G. Verin, K.-W. Wen, K. Kim, R. Arora, A. Odgers, L. M. Contreras και S. Scarpina, «MEC in 5G networks,» *ETSI White Paper No. 28*, June 2018.
- [3] «Wikipedia,» 2021. [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/M/M/1\\_queue](https://en.wikipedia.org/wiki/M/M/1_queue).
- [4] T. Van Do, N. Do, H. Nguyen, C. Rotter, A. Hegyi και P. Hegyi, «Comparison of scheduling algorithms for multiple mobile computing edge clouds,» *Elsevier*, 17 October 2018.
- [5] C. Schuss και T. Rahkonen, «Solar Energy Harvesting Strategies for Portable Devices such as Mobile Phones,» *Proceeding of the 14th Conference of FRUCT Association*, 2012.
- [6] Y. Mao, C. You, Y. Zhang, K. Huang και K. B. Letaief, «A Survey on Mobile Edge Computing: The Communication Perspective,» 13 June 2017.
- [7] D. Sabella, A. Vaillant, P. Kuure, U. Rauschenbach και F. Giust, *Mobile-Edge Computing Architecture: The role of MEC in Internet of Things*, IEEE Consumer Electronics Magazine, 2016.
- [8] Y. Chao Hu, M. Patel, D. Sabella, N. Sprecher και V. Young, «Mobile Edge Computing, a key technology towards 5G,» *ETSI White Paper No. 11*, September 2015.
- [9] M. Hicham, N. Abghour και M. Ouzzif, «4G System: Network Architecture and Performance,» *IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC)*.
- [10] G. A. Abed, I. Mahamod και K. Jumari, «The Evolution to 4G Cellular Systems: Architecture,» *IRACST – International Journal of Computer Networks and Wireless Communications (IJCNWC)*, τόμ. Vol.2, αρ. Νο1, 2012.
- [11] F. Giust, G. Verin, K. Antevski, J. Chou, Y. Fang, W. Featherstone, F. Fontes, D. Frydman, A. Li, A. Manzalini, D. Purkayastha, D. Sabella, C. Wehner, K.-W. Wen και Z. Zhou, *MEC Deployments in 4G and Evolution Towards 5G*, ETSI, 2018.
- [12] S. F. III, «[www.zdnet.com](http://www.zdnet.com),» ZDNet, 27 April 2021. [Ηλεκτρονικό]. Available: <https://www.zdnet.com/article/what-is-5g-the-business-guide-to-next-generation-wireless-technology/>.

- [13] «IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond,» *Recommendation ITU-R M.2083-0*, September 2015.
- [14] ETSI, «5G: System Architecture for the 5G System (3GPP TS 38.300 version 15.8.0 Release 15)». Ευρεσιτεχνία ETSI TS 138 300 V15.8.0, January 2021.
- [15] «www.vmware.com,» VMware, Inc., 2021. [Ηλεκτρονικό]. Available: <https://www.vmware.com/topics/glossary/content/software-defined-networking>.
- [16] B. Liu, X. Xu, L. Qi, Q. Ni και W. Dou, «Task scheduling with precedence and placement constraints for resource,» *Journal of Systems Architecture*, 2020.
- [17] Y. Chen, Y. Zhang και X. Chen, *Dynamic Service Request Scheduling for Mobile Edge Computing Systems*, K. Wong, Επιμ., Beijing: BISTU, 2018.
- [18] Y. Nan, W. Li, W. Bao, F. C. Delicato, P. F. Pires, Y. Dou και A. Y. Zomaya, *Adaptive Energy-Aware Computation Offloading for Cloud of Things System*, IEEE Access, 2017.
- [19] Α.-Γ. Σταφυλοπάτης και Γ. Σιόλας, *Ανάλυση Επίδοσης Υπολογιστικών Συστημάτων*, Αθήνα: Ελληνικά Ακαδημαϊκά Ηλεκτρονικά Συγγράμματα και Βοηθήματα, 2015.
- [20] «Wikipedia,» 2 August 2021. [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Little%27s\\_law](https://en.wikipedia.org/wiki/Little%27s_law).
- [21] Δ. Π. Μπερτσέκας και Γ. Ν. Τσιτσικλής, *Εισαγωγή στις Πιθανότητες με Στοιχεία Στατιστικής*, Εκδόσεις Τζιόλα, 2016.
- [22] Y. Mao, J. Zhang και K. B. Letaief, «Dynamic Computation Offloading for Mobile-Edge Computing With Energy Harvesting Devices,» *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, τόμ. 34, αρ. 12, December 2016.
- [23] H. Badri, T. Bahreini, D. Grosu και K. Yang, «A Sample Average Approximation-Based Parallel Algorithm for Application Placement in Edge Computing Systems,» *IEEE International Conference on Cloud Engineering*, 2018.
- [24] C. Schuss και T. Rahkonen, «Photovoltaic (PV) Energy as Recharge Source for Portable Devices such as Mobile Phones,» *Proceeding of the 12th Conference of FRUCT Association*, 2012.
- [25] D. Tien, D. Nam και C. Ram, «A new queueing model for spectrum renting in mobile cellular networks,» *Computer Communications*, τόμ. 35, αρ. 10, pp. 1165-1171, June 2012.
- [26] T. Camp, J. Boleng και V. Davies, «A survey of mobility models for ad hoc network research,» *WIRELESS COMMUNICATIONS AND MOBILE COMPUTING*, αρ. 2, pp. 483-502, 2002.
- [27] H. Zhao, «github.com,» 2019. [Ηλεκτρονικό]. Available: <https://github.com/hliangzhao/Edge-Computing-Codes>. [Πρόσβαση September 2021].
- [28] H. Zhao, W. Du, W. Liu, T. Lei και Q. Lei, «QoE Aware and Cell Capacity Enhanced Computation Offloading for Multi-Server Mobile Edge Computing Systems with Energy Harvesting Devices.,» *Proceedings of the 15th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC'18)*, 2018.
- [29] T. D. Burd και R. W. Brodersen, «Processor design for portable systems,» *ournal of VLSI signal processing systems for signal, image and video technology volume*, αρ. 13, pp. 203-221, August 1996.
- [30] S. Edelkamp και S. Schroedl, *Heuristic Search, Theory and Applications*, Morgan Kaufmann, 2012.

- [31] D. Bertsekas, Dynamic Programming and Optimal Control, 3rd επιμ., Belmoth, MA: Athens Scientific, 2005.
- [32] «Wikipedia,» July 2021. [Ηλεκτρονικό]. Available: [https://en.wikipedia.org/wiki/Lyapunov\\_optimization](https://en.wikipedia.org/wiki/Lyapunov_optimization).
- [33] «MathWorks,» The MathWorks, [Ηλεκτρονικό]. Available: <https://www.mathworks.com/help/optim/ug/fsolve.html>.
- [34] «www.blacktechnoguys.com,» November 2018. [Ηλεκτρονικό]. Available: <https://www.blacktechnoguys.com/2018/11/5g-basic-part-i.html>.
- [35] W.-K. Chiang και M.-W. Wang, «Edge-Based Evolved Packet Core (EPC) Refactoring for High Speed Mobility,» 19 June 2021.
- [36] Y. Egi, «Statistical Comparison of Measurement Hardware Used in LTE Performance Assessment,» December 2014.
- [37] G. Chen, X. Wen, W. Kaiqiang και L. Xu, «Radio Frequency Fingerprint-Based Intelligent Mobile Edge Computing for Internet of Things Authentication,» August 2019.
- [38] [Ηλεκτρονικό]. Available: [https://www.researchgate.net/figure/Architecture-of-autonomous-vehicles-with-edge-computing\\_fig11\\_336338442/download](https://www.researchgate.net/figure/Architecture-of-autonomous-vehicles-with-edge-computing_fig11_336338442/download).
- [39] [Ηλεκτρονικό]. Available: [https://www.researchgate.net/figure/Architecture-of-autonomous-vehicles-with-edge-computing\\_fig11\\_336338442/download](https://www.researchgate.net/figure/Architecture-of-autonomous-vehicles-with-edge-computing_fig11_336338442/download).

