



Πανεπιστήμιο Πελοποννήσου
Σχολή Οικονομίας και Τεχνολογίας
Τμήμα Πληροφορικής και Τηλεπικοινωνιών

Μοντέλα Απωλειών/Ουρών Κλήσεων
Μονοδιάστατης Τηλεπικοινωνιακής Κίνησης σε
Mobile Hotspots

Διπλωματική Εργασία

Συγγραφέας: Γεώργιος Μανέλης

Επιβλέπων: Ιωάννης Μοσχολιός
Αναπληρωτής Καθηγητής

Μάρτιος 2023

Μοντέλα Απωλειών/Ουρών Κλήσεων Μονοδιάστατης Τηλεπικοινωνιακής Κίνησης σε Mobile Hotspots

Περίληψη

Ένα ζήτημα που προκύπτει στα mobile hotspots οχημάτων είναι η διενέργεια μεταπομπής κλήσεων από έναν εξωτερικό σταθμό βάσης σε ένα σημείο πρόσβασης του hotspot του οχήματος. Η διαδικασία αυτή είναι αναγκαία καθώς μέσω αυτής, ο χρήστης/επιβάτης του οχήματος έχει τη δυνατότητα να συνδεθεί στο WLAN σημείο πρόσβασης, το οποίο του παρέχει καλύτερη ποιότητα υπηρεσιών χαμηλού ή ακόμα και μηδενικού κόστους.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη και η σύγκριση τεσσάρων αναλυτικών μοντέλων, που αφορούν τον έλεγχο της αποδοχής των κλήσεων μεταπομπής στο σημείο πρόσβασης του mobile hotspot. Τα μοντέλα αυτά βασίζονται σε Μαρκοβιανές αλυσίδες συνεχούς χρόνου και η μελέτη τους αφορά δύο φάσεις του οχήματος: την φάση κατά την οποία το όχημα βρίσκεται σε κίνηση και εκείνη κατά την οποία βρίσκεται σε στάση.

Πιο συγκεκριμένα, το πρώτο μοντέλο αφορά έναν αλγόριθμο ελέγχου της αποδοχής κλήσεων (Mobility-Aware Call Admission Control(MA-CAC)) όπου ένας ορισμένος αριθμός καναλιών, κατά τη διάρκεια της φάσης στην οποία το όχημα είναι σταματημένο, χρησιμοποιείται αποκλειστικά για την εξυπηρέτηση κλήσεων μεταπομπής. Το μοντέλο αυτό είναι εφοδιασμένο με μια ουρά αναμονής για τις κλήσεις μεταπομπής που δεν μπορούν να εξυπηρετηθούν από το σύστημα, λόγω κάλυψης της χωρητικότητάς του από άλλες κλήσεις. Το δεύτερο μοντέλο αφορά τον προαναφερθέντα αλγόριθμο, χωρίς την ουρά αναμονής, ενώ το τρίτο μοντέλο είναι ένα σύστημα στο οποίο δεν δίνεται κανενός είδους προτεραιότητα στις κλήσεις μεταπομπής, σε σχέση με τις υπόλοιπες κλήσεις του συστήματος. Στο τέταρτο και τελευταίο μοντέλο διατηρείται ένας αριθμός καναλιών υπέρ των κλήσεων μεταπομπής, τόσο στη διάρκεια κίνησης όσο και στη διάρκεια στάσης του οχήματος.

Σε καθένα από τα μοντέλα αυτά υπολογίζονται οι πιθανότητες απόρριψης των κλήσεων μεταπομπής και απώλειας των νέων κλήσεων που πραγματοποιούνται, όπως επίσης και το ποσοστό χρήσης του κάθε καναλιού. Επιπροσθέτως, γίνεται μια εκτίμηση του χρόνου αναμονής των κλήσεων στην ουρά μεταπομπής του μοντέλου MA-CAC με ουρά μεταπομπής. Οι παράμετροι εισόδου, που είναι κοινές και στα τέσσερα μοντέλα, είναι η χωρητικότητα του δικτύου WLAN και η μέση διάρκεια, ο αριθμός άφιξης και ο ρυθμός του χρόνου παραμονής στο σύστημα τόσο των κλήσεων μεταπομπής όσο και των νέων κλήσεων. Επιπλέον χρησιμοποιείται ο αριθμός των καναλιών που προορίζονται αποκλειστικά για κλήσεις μεταπομπής, με εξαίρεση το τρίτο μοντέλο, ενώ στο μοντέλο με την ουρά συμπεριλαμβάνεται στις παραμέτρους εισόδου και το μέγεθος της ουράς του αλγορίθμου MA-CAC. Τέλος, ερευνώνται οι επιδράσεις μερικών από τις παραμέτρους εισόδου, σε καθένα από τα τέσσερα μοντέλα απωλειών κλήσεων, αλλά και οι επιδράσεις

του μεγέθους της ουράς μεταπομπής στον μέσο χρόνο αναμονής των κλήσεων στην ουρά.

Call Loss/Queueing Models of Single Rate Telecommunication Traffic in Mobile Hotspots

Abstract

An issue that arises in vehicular mobile hotspots is handover from an external base station to an access point of the vehicle's hotspot. This procedure is necessary because through it, the passenger/user of the vehicle is allowed to be connected to the WLAN access point, which provides him a better quality of service at low or even zero cost.

The purpose of this thesis is the study and comparison of four analytical models concerning the control of the acceptance of handover calls at the access point of the mobile hotspot. These models are based on continuous time Markov chains and their study concerns two phases of the vehicle: the phase in which the vehicle is in motion and the one in which it is stationary.

Specifically, the first model is about a call admission control algorithm (Mobility-Aware Call Admission Control(MA-CAC)) where a certain number of channels, during the phase in which the vehicle is stopped, is used exclusively to service handover calls. This model is equipped with a waiting queue for the transfer calls that cannot be served by the system, due to its capacity being covered by other calls. The second model involves the aforementioned algorithm, without the waiting queue, while the third model is a system in which handover calls are not given any kind of priority over other calls in the system. The fourth and final model maintains a number of channels in favor of handover calls, both while the vehicle is in motion and while the vehicle is stationary.

In each of these models, the probabilities of new call blocking and handover call dropping that are made are being calculated, as well as the channel utilization. In addition, an estimation of the waiting time of calls in the handover queue of the MA-CAC model with handover queue is made. The input parameters, common to all four models, are the capacity of the WLAN network and the average duration, the number of arrivals, and the rate of dwell time in the system of both handover calls and new calls. Moreover, the number of channels which are reserved exclusively for handover calls is used, with the exception of the third model, while in the queueing model, the MA-CAC algorithm's queue size is included among the input parameters. Finally, the effects of some of the input parameters in each of the four call loss models are investigated, as well as the effects of the handover queue size, on the average call waiting time in the queue.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω όλους τους καθηγητές του Προγράμματος Μεταπτυχιακών Σπουδών "Επιστήμη Υπολογιστών" του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Πανεπιστημίου Πελοποννήσου, για τις γνώσεις και τη βοήθεια που μου προσέφεραν καθ'όλη τη διάρκεια μεταπτυχιακών μου σπουδών. Ιδιαίτερα, θα ήθελα να ευχαριστήσω θερμά τον Αναπληρωτή Καθηγητή κ. Ιωάννη Μοσχολιό για την ανάληψη του θέματος και την διαρκή και υποδειγματική καθοδήγησή του κατά τη διάρκεια της εκπόνησης της διπλωματικής εργασίας μου.

Επιπλέον, θα ήθελα να ευχαριστήσω τους πολύ καλούς φίλους μου Ειρήνη Κεραμίδη και Δημήτριο Ουζουνίδη, για την συμπαράστασή τους και για την έμπρακτη στήριξη που μου παρείχαν στην προσπάθειά μου να ανταπεξέλθω στις απαιτήσεις του μεταπτυχιακού προγράμματος.

Γεώργιος Μανέλης

Μοντέλα Απωλειών/Ουρών Κλήσεων
Μονοδιάστατης Τηλεπικοινωνιακής Κίνησης σε
Mobile Hotspots

Γεώργιος Μανέλης
dit2110cst@go.uop.gr

Μάρτιος 2023

Περιεχόμενα

Εισαγωγή	1
1 Μοντέλο Απωλειών MA-CAC με Ουρά Μεταπομπής	5
1.1 Εισαγωγή	5
1.2 Ο Αλγόριθμος MA-CAC με Ουρά Μεταπομπής	5
1.3 Ανάλυση του μοντέλου	7
1.4 Παράδειγμα	13
2 Μοντέλο Απωλειών MA-CAC χωρίς Ουρά Μεταπομπής	31
2.1 Εισαγωγή	31
2.2 Ο Αλγόριθμος MA-CAC χωρίς Ουρά Μεταπομπής	31
2.3 Ανάλυση του μοντέλου	32
2.4 Παράδειγμα	37
3 Μοντέλο Απωλειών Non Priority	51
3.1 Εισαγωγή	51
3.2 Ο Αλγόριθμος NP	51
3.3 Ανάλυση του μοντέλου	52
3.4 Παράδειγμα	57
4 Μοντέλο Απωλειών Cutoff Priority	69
4.1 Εισαγωγή	69
4.2 Ο Αλγόριθμος CP	69
4.3 Ανάλυση του μοντέλου	70
4.4 Παράδειγμα	75
5 Επιδράσεις των Παραμέτρων στα Μοντέλα Απωλειών/Ουρών Κλήσεων	87
5.1 Εισαγωγή	87
5.2 Επιδράσεις του β_H	88
5.3 Επιδράσεις του β_N	91
5.4 Επιδράσεις του t_S	92
5.5 Επιδράσεις του K	94
5.6 Επιδράσεις του $1/\theta_S$	96
5.7 Επιδράσεις του $1/\theta_M$	98
Ανακεφαλαίωση - Συμπεράσματα	101
Α' Ακρωνύμια και συντομογραφίες	103

Β΄ Προγράμματα σε Γλώσσα Python των Μοντέλων Απωλειών για τα Παραδείγματα των Κεφαλαίων 1, 2, 3 και 4	105
Β.1 Πρόγραμμα MA-CAC με HQ του Παραδείγματος 1.4	105
Β.2 Πρόγραμμα MA-CAC χωρίς HQ του Παραδείγματος 2.4	120
Β.3 Πρόγραμμα Non Priority του Παραδείγματος 3.4	134
Β.4 Πρόγραμμα Cutoff Priority του Παραδείγματος 4.4	141
Γ΄ Προγράμματα σε Γλώσσα Python των Επιδράσεων των Παραμέτρων στα Μοντέλα Απωλειών	153
Γ.1 Πρόγραμμα των Επιδράσεων του λ_H	153
Γ.2 Πρόγραμμα των Επιδράσεων του λ_N	181
Γ.3 Πρόγραμμα των Επιδράσεων του t_S	209
Γ.4 Πρόγραμμα των Επιδράσεων του K	236
Γ.5 Πρόγραμμα των Επιδράσεων του $1/\vartheta_S$	263
Γ.6 Πρόγραμμα των Επιδράσεων του $1/\vartheta_M$	290
Βιβλιογραφία	320

Κατάλογος Πινάκων

1.1	Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 1.4	25
1.2	Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 1.4 (Πρόγραμμα σε Python)	25
1.3	Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 1.4	27
1.4	Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 1.4 (Πρόγραμμα σε Python)	27
1.5	P_{NCBP} , P_{HCDP} , Ποσοστό Χρήσης Καναλιών και Χρόνος Αναμονής στην Ουρά Μεταπομπής του Παραδείγματος 1.4	29
1.6	P_{NCBP} , P_{HCDP} , Ποσοστό Χρήσης Καναλιών και Χρόνος Αναμονής στην Ουρά Μεταπομπής του Παραδείγματος 1.4 (Πρόγραμμα σε Python)	29
2.1	Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 2.4	46
2.2	Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 2.4 (Πρόγραμμα σε Python)	46
2.3	Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 2.4	48
2.4	Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 2.4 (Πρόγραμμα σε Python)	48
2.5	P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 2.4	49
2.6	P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 2.4 (Πρόγραμμα σε Python)	49
3.1	Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 3.4	65
3.2	Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 3.4 (Πρόγραμμα σε Python)	66
3.3	Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 3.4	67
3.4	Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 3.4 (Πρόγραμμα σε Python)	67
3.5	P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 3.4	68
3.6	P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 3.4 (Πρόγραμμα σε Python)	68
4.1	Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 4.4	83
4.2	Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 4.4 (Πρόγραμμα σε Python)	83
4.3	Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 4.4	84
4.4	Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 4.4 (Πρόγραμμα σε Python)	85
4.5	P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 4.4	85

4.6 P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 4.4 (Πρόγραμμα σε Python)	85
---	----

Κατάλογος Σχημάτων

1.1 Διάγραμμα Ροής του Αλγορίθμου MA-CAC με HQ.	6
1.2 Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για τον αλγόριθμο MA-CAC με HQ.	7
1.3 Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για τον αλγόριθμο MA-CAC με HQ του Παραδείγματος 1.4.	14
2.1 Διάγραμμα Ροής του Αλγορίθμου MA-CAC χωρίς HQ.	32
2.2 Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για τον αλγόριθμο MA-CAC χωρίς HQ.	33
2.3 Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για τον αλγόριθμο MA-CAC χωρίς HQ του Παραδείγματος 2.4.	37
3.1 Διάγραμμα Ροής του Αλγορίθμου NP.	52
3.2 Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου του Μοντέλου Απωλειών NP.	52
3.3 Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για το Μοντέλο Απωλειών NP του Παραδείγματος 3.4.	57
4.1 Διάγραμμα Ροής του Αλγορίθμου CP.	70
4.2 Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου του Μοντέλου Απωλειών CP.	70
4.3 Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για το Μοντέλο Απωλειών CP του Παραδείγματος 4.4.	76
5.1 Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσσει του β_H	88
5.2 Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσσει του β_H	89
5.3 Ποσοστό Χρήσης των Καναλιών συναρτήσσει του β_H	90
5.4 Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσσει του β_N	91
5.5 Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσσει του β_N	92
5.6 Ποσοστό Χρήσης των Καναλιών συναρτήσσει του β_N	92
5.7 Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσσει του t_S	93
5.8 Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσσει του t_S	93
5.9 Ποσοστό Χρήσης των Καναλιών συναρτήσσει του t_S	94
5.10 Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσσει του K	95
5.11 Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσσει του K	95
5.12 Ποσοστό Χρήσης των Καναλιών συναρτήσσει του K	96
5.13 Χρόνος Αναμονής των Κλήσεων στην Ουρά συναρτήσσει του K	96
5.14 Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσσει του θ_S	97
5.15 Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσσει του θ_S	98
5.16 Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσσει του θ_M	98
5.17 Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσσει του θ_M	99

Εισαγωγή

Στη σύγχρονη εποχή, η χρήση κινητών/φορητών συσκευών επικοινωνίας αποτελεί αναπόσπαστο κομμάτι της καθημερινότητας των ανθρώπων. Ειδικότερα, τα τελευταία χρόνια η εξέλιξη των ασύρματων δικτύων σε συνδυασμό με τη ραγδαία ανάπτυξη των “έξυπνων” συσκευών, έχουν μετατρέψει συσκευές όπως τα παραδοσιακά κινητά τηλέφωνα σε μικρούς υπολογιστές, που έχουν τη δυνατότητα να συνδεθούν στο Διαδίκτυο. Έτσι οι χρήστες μπορούν να χρησιμοποιήσουν τις συσκευές επικοινωνίας τους τόσο για επαγγελματικούς όσο και για προσωπικούς/ψυχαγωγικούς σκοπούς, οπουδήποτε κι αν βρίσκονται. Όμως για να μπορέσουν οι “έξυπνες” συσκευές, όπως τα smartphones, τα tablets και τα netbooks να αξιοποιήσουν τις δυνατότητες που παρέχουν στους χρήστες τους στο μέγιστο δυνατό βαθμό, είναι απαραίτητη η απρόσκοπτη και όσο το δυνατόν πιο σταθερή σύνδεσή τους με το Διαδίκτυο.

Από τη στιγμή που ολοένα και περισσότεροι κάτοικοι των αστικών κέντρων περνούν όλο και περισσότερο χρόνο στα αυτοκίνητά τους, στα ταξί αλλά και στα μέσα μαζικής μεταφοράς όπως το μετρό, τα αστικά λεωφορεία και τα τρένα, η σύνδεση των “έξυπνων” συσκευών στο Διαδίκτυο γίνεται δυσκολότερη. Για να αντιμετωπιστεί αυτό το πρόβλημα, τα σιδηροδρομικά συστήματα υψηλών ταχυτήτων της Νοτίου Κορέας (*Korean Train eXpress* - KTX) αλλά και τα ταξί στην πρωτεύουσα Σεούλ [1], παρέχουν δωρεάν διαδικτυακές υπηρεσίες. Στο ίδιο μήκος κύματος σκοπεύει να κινηθεί και το δημόσιο βιομηχανικό και εμπορικό ίδρυμα *Société du Grand Paris* (SGP), που είναι υπεύθυνο για τον σχεδιασμό και την κατασκευή των νέων γραμμών του μετρό, περιφερειακά της πόλης του Παρισιού [2].

Για να επιτευχθεί η παροχή υπηρεσιών Διαδικτύου στις περιπτώσεις που αναφέρθηκαν στην προηγούμενη παράγραφο, εγκαθίσταται εντός των οχημάτων ένα ασύρματο τοπικό δίκτυο (Wireless local Area Network) στο οποίο οι επιβάτες/χρήστες μπορούν να συνδέσουν τις συσκευές τους μέσω ενός σημείου πρόσβασης (Access Point) . Έτσι δημιουργείται ένα mobile hotspot μέσω του οποίου οι χρήστες μπορούν να έχουν μια πιο σταθερή και γρήγορη σύνδεση της συσκευής τους στο Διαδίκτυο, απ’ ότι θα είχαν αν χρησιμοποιούσαν το WiFi της συσκευής τους, για να συνδεθούν ενδεχομένως σε ένα δημόσιο δίκτυο.

Σε ότι αφορά τα mobile hotspots, έχουν διεξαχθεί πολλές έρευνες σχετικά με την ποιότητα των υπηρεσιών (quality of service - QoS) που παρέχουν, την αποδοτικότητά τους και τη διαχείριση της κινητικότητας.

Στη βιβλιογραφία συναντάμε ένα πλήθος εργασιών και προτεινόμενων συστημάτων που αποσκοπούν στη βελτίωση της παροχής υπηρεσιών στα mobile hotspots. Οι Nayato και Hossain [3] πρότειναν συστήματα στα οποία γίνεται κατανομή του εύρους ζώνης σε ετερογενή multihop δίκτυα, ενώ οι Lee [4] κλπ ανέπτυξαν ένα σύστημα μεταπομπής

στο οποίο ο κινητός δρομολογητής (Mobile Router) διαπραγματεύεται τις συνολικές απαιτήσεις της ποιότητας υπηρεσιών των συνημμένων σε αυτόν κινητών κόμβων του. Οι Kirsal, Mapp και Sardis προτείνουν στην εργασία τους [5] ένα πλαίσιο παροχής υπηρεσιών, βασισμένο στην τεχνολογία υπολογιστικού νέφους, για την διαχείριση των πόρων του δικτύου, παρέχοντας συγχρόνως υψηλή ποιότητα υπηρεσιών στους χρήστες κινητών συσκευών. Σε αυτό το πλαίσιο, οι χρήστες καλούνται να επιλέξουν ανάμεσα στο αν θα παραμείνουν συνδεδεμένοι στο τωρινό δίκτυο ή αν θα εκτελεστεί κάθετη μεταπομπή προς ένα γειτονικό δίκτυο ή αν θα αιτηθούν να μετακινηθεί η υπηρεσία στο μέρος του χρήστη, χρησιμοποιώντας την τεχνική του εντοπισμού του διακομιστή (server localization). Στην εργασία [6] οι Wang κλπ εισήγαγαν ένα σύστημα διαμοιρασμού του εύρους ζώνης για τη δέσμευση των πόρων μεταξύ ενός κινητού δρομολογητή και του Home Agent του δρομολογητή, ώστε να ξεπεραστεί το πρόβλημα των πρωτοκόλλων δέσμευσης των πόρων στα mobile hotspots. Τέλος, οι Li κλπ [7] ανέπτυξαν ένα σύστημα μεταπομπής σε κυψελοειδή δίκτυα με femtocells, προκειμένου να επεκτείνουν την κάλυψη του δικτύου και να βελτιώσουν την παροχή της ποιότητας υπηρεσιών προς τους χρήστες.

Η αποδοτικότητα στα mobile hotspots έχει μελετηθεί και αξιολογηθεί σε επίπεδο ανάλυσης, προσομοιώσεων και πειραμάτων. Οι Pack κλπ [8] ανέλυσαν την αποδοτικότητα του πρωτοκόλλου TCP-Friendly Rate (TFRC) που υποστηρίζει υπηρεσίες πολυμέσων σε mobile hotspots. Στην εργασία τους ανέπτυξαν μοντέλα ουρών αναμονής διακριτού χρόνου, για συνδέσεις τόσο με ασύρματα τοπικά δίκτυα (WLAN) όσο και με ασύρματα δίκτυα ευρείας περιοχής (WWAN) και το συμπέρασμα που προέκυψε είναι ότι η κατάλληλη κατανομή των πόρων του δικτύου καθώς και ο έλεγχος αποδοχής, είναι απαραίτητα στοιχεία για την ποιότητα και την αποτελεσματικότητα των υπηρεσιών πολυμέσων στα mobile hotspots. Οι Song κλπ [9] ανέλυσαν την αποδοτικότητα της καθυστέρησης στα mobile hotspots οχημάτων, όπως λεωφορεία, τρένα και αεροπλάνα, λαμβάνοντας υπόψη μόνο τη φάση κατά την οποία το όχημα βρίσκεται σε κίνηση. Επομένως, οι καταστάσεις επιβίβασης/αποβίβασης, που αποτελούν σημαντικό παράγοντα στα mobile hotspots των οχημάτων που συμμετέχουν στα μέσα μαζικής μεταφοράς, δεν μελετήθηκαν. Οι Choi κλπ [10] προτείνουν ένα δίκτυο μικροκυματικής ζώνης (millimeter wave) που μπορεί να εφαρμοστεί τόσο στα τρένα υψηλών ταχυτήτων όσο και στο μετρό. Το σύστημα αυτό μπορεί να ξεπεράσει τις εγγενείς υψηλές απώλειες διαδρομής, μέσω κεραίων υψηλής απολαβής. Τέλος, οι Lee, Flinn και Tonshal [11] εστιάζουν σε οχήματα που προσφέρουν συνδεσιμότητα αρχιτεκτονικής οχήματος-προς-υποδομή (vehicle-to-infrastructure - V2I) και συμπεριφέρονται ως mobile hotspots. Οι μελέτες τους έδειξαν πως ο υψηλός χρόνος απόκρισης του δικτύου (latency) ποικίλλει με απρόβλεπτο τρόπο κι έτσι ουσιαστικά υποβαθμίζει την εμπειρία του χρήστη. Έτσι ανέπτυξαν έναν χρονοδρομολογητή πρωτοκόλλου ελέγχου μεταφορών πολλαπλών διαδρομών (Multipath Transmission Protocol - MPTCP), που μετριάζει τον χρόνο απόκρισης του δικτύου, χρησιμοποιώντας πλεονάζουσα μετάδοση όταν η εμπιστοσύνη σχετικά με την πρόβλεψη του χρόνου απόκρισης του δικτύου είναι χαμηλή.

Για την διαχείριση της κινητικότητας στα mobile hotspots, οι Pack κλπ [12] μελέτησαν το ζήτημα της καθυστέρησης της μεταπομπής, συγκρίνοντας ένα πρωτόκολλο βασικής υποστήριξης της κινητικότητας του δικτύου (NEMO) με ένα πρωτόκολλο υποστήριξης NEMO, που βασίζεται σε έναρξη συνεδρίας (Session Initiation Protocol - SIP). Στο πλαίσιο των SIP - NEMO περιβαλλόντων, οι Chiang κλπ [13] πρότειναν έναν μηχανισμό μεταπομπής, υποβοηθούμενο από απόσταση, στον οποίο εισήχθη ένας διαχωριστής ε-

λέγχου συνεδρίας για ταυτόχρονη μεταπομπή σε mobile hotspots. Στην εργασία [14] οι Zhang κλπ εισήγαγαν ένα σύστημα επικοινωνίας εντός συστάδας, ώστε να εξασφαλίσουν την απρόσκοπτη μεταπομπή σε mobile hotspots ενώ οι Gilani κλπ [15] προτείνουν ένα σύστημα διαχείρισης της κινητικότητας σε WLAN, καθοριζόμενα από λογισμικό (Software Defined Network - SDN), δίκτυα. Το σύστημα αυτό βασίζεται σε λογικά σημεία πρόσβασης (Logical Access Points - LAP) και διατηρεί τη σύνδεση με το δίκτυο κατά τη διάρκεια της μεταπομπής, είτε η μεταπομπή ενεργοποιείται από τον χρήστη είτε από τον ελεγκτή SDN, για απρόσκοπτη κινητικότητα. Τέλος, οι Aljjeri και Boukerche [16] πρότειναν ένα σύστημα δύο επιπέδων, βασισμένο σε Μηχανική Μάθηση, για την διαχείριση της μεταπομπής στα ευφυή δίκτυα οχημάτων (Intelligent Vehicular Networks - IVN). Στο πρώτο επίπεδο, χρησιμοποίησαν ένα μοντέλο επαναλαμβανόμενου νευρωνικού δικτύου για την πρόβλεψη της ισχύος του σήματος λήψης των σημείων πρόσβασης του δικτύου ενώ στη δεύτερη φάση χρησιμοποίησαν ένα στοχαστικό μοντέλο Markov για την επιλογή του επόμενου σημείου πρόσβασης, λαμβάνοντας υπόψη τις προβλέψεις ροής του οχήματος.

Στα mobile hotspots των οχημάτων, η διενέργεια μεταπομπής μιας κλήσης από έναν εξωτερικό σταθμό βάσης (Base Station) σε ένα σημείο πρόσβασης του δικτύου WLAN του οχήματος είναι απαραίτητη καθώς με αυτόν τον τρόπο, ο χρήστης μπορεί να έχει μια σύνδεση με καλύτερες συνθήκες καναλιού και με χαμηλό, έως και μηδενικό τέλος σύνδεσης. Όμως στην περίπτωση που το όχημα ανήκει στα μέσα μαζικής μεταφοράς, όπως είναι τα αστικά λεωφορεία, τα βαγόνια του μετρό και άλλα, τα διαθέσιμα κανάλια του WLAN δικτύου του χρησιμοποιούνται από έναν μεγάλο αριθμό χρηστών και έτσι η πιθανότητα να απορριφθούν οι κλήσεις που προέρχονται από μεταπομπή είναι μεγάλη. Έτσι είναι αναγκαίο να είναι εγκατεστημένος, στο σημείο πρόσβασης του δικτύου, ένας μηχανισμός αποδοχής των κλήσεων ο οποίος θα είναι σε θέση να παίρνει αποφάσεις για το ποιες κλήσεις θα εξυπηρετηθούν από το δίκτυο και ποιες όχι, προκειμένου να υπάρχει μια ισορροπία ανάμεσα στις κλήσεις μεταπομπής και στις νέες κλήσεις και να μειωθεί η πιθανότητα απόρριψής τους.

Στην παρούσα εργασία παρουσιάζονται τέσσερα μοντέλα απωλειών κλήσεων μονοδιάστατης τηλεπικοινωνιακής κίνησης, σε mobile hotspots οχημάτων. Τα μοντέλα αυτά βασίζονται σε Μαρκοβιανές αλυσίδες συνεχούς χρόνου και λαμβάνουν υπόψη τους δύο φάσεις του οχήματος: τη φάση κατά την οποία το όχημα είναι στατικό (φάση στάσης) και τη φάση όπου το όχημα βρίσκεται σε κίνηση (φάση κίνησης).

Πιο συγκεκριμένα, τα δύο πρώτα Κεφάλαια της εργασίας αφορούν το μοντέλο Mobility-Aware Call Admission Control (MA-CAC). Το σύστημα αυτό, κατά τη διάρκεια τη φάσης στάσης του οχήματος, διατηρεί έναν σταθερό αριθμό καναλιών του δικτύου WLAN του οχήματος προκειμένου να χρησιμοποιηθούν αποκλειστικά για την εξυπηρέτηση των κλήσεων που προέρχονται από μεταπομπή ενώ απελευθερώνει τα κανάλια αυτά στη φάση που το όχημα κινείται. Ειδικότερα στο Κεφάλαιο 1 αναπτύσσεται το μοντέλο MA-CAC, εφοδιασμένο με μία ουρά αναμονής στην οποία εισέρχονται οι κλήσεις μεταπομπής (ουρά μεταπομπής) όταν όλα τα κανάλια του δικτύου είναι απασχολημένα ενώ στο Κεφάλαιο 2, αναλύεται το μοντέλο MA-CAC χωρίς την ουρά μεταπομπής. Στο Κεφάλαιο 3 παρουσιάζεται το μοντέλο απωλειών Non Priority (NP), το οποίο δεν διατηρεί δεσμευμένα κανάλια για την αποκλειστική χρήση τους από κλήσεις μεταπομπής, σε καμία από τις δύο φάσεις κίνησης του οχήματος. Στο Κεφάλαιο 4 ερευνάται το σύστημα απωλειών Cutoff Priority (CP) όπου διατηρείται δεσμευμένος ένας σταθερός αριθμός καναλιών υπέρ των κλήσεων μεταπομπής, τόσο στη φάση στάσης όσο και στη

φάση κατά την οποία το όχημα κινείται. Σε καθένα από τα προαναφερθέντα Κεφάλαια γίνεται η αναλυτική περιγραφή του αντίστοιχου μοντέλου απωλειών κλήσεων, παρουσιάζονται τα διαγράμματα ροής των αλγορίθμων τους και περιγράφεται μια μέθοδος υπολογισμού των πιθανοτήτων απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής αντίστοιχα, καθώς και του ποσοστού χρήσης των καναλιών του καθενός συστήματος. Για το μοντέλο MA-CAC με ουρά μεταπομπής υπολογίζεται επιπλέον και ο μέσος χρόνος αναμονής των κλήσεων στην ουρά μεταπομπής. Στην τελευταία Ενότητα του κάθε Κεφαλαίου παρατίθεται ένα αριθμητικό παράδειγμα για το καθένα από τα μοντέλα απωλειών κλήσεων. Στο τελευταίο Κεφάλαιο της εργασίας (Κεφάλαιο 5) παρουσιάζονται οι επιδράσεις κάποιων εκ των παραμέτρων εισόδου των συστημάτων απωλειών κλήσεων. Μέσω γραφικών παραστάσεων μελετάται η επιρροή των παραμέτρων αυτών στην πιθανότητα απώλειας των νέων κλήσεων, στην πιθανότητα απόρριψης των κλήσεων μεταπομπής, στο ποσοστό χρήσης των καναλιών και στον χρόνο αναμονής στην ουρά μεταπομπής.

Μετά το πέρας του 5ου Κεφαλαίου γίνεται μία ανασκόπηση των όσων παρουσιάστηκαν στο κυρίως σώμα της εργασίας και παρατίθενται τα συμπεράσματα που προκύπτουν από την ανάλυση των τεσσάρων μοντέλων απωλειών κλήσεων. Τέλος, σε ότι αφορά τα παραρτήματα, το Παράρτημα Α' περιέχει τα ακρωνύμια και τις συντομογραφίες που συναντώνται στην εργασία, στο Παράρτημα Β' παρατίθενται τα προγράμματα για τα μοντέλα απωλειών κλήσεων των παραδειγμάτων 1.4, 2.4, 3.4 και 4.4, που υλοποιήθηκαν σε γλώσσα Python και στο Παράρτημα Γ' παρουσιάζονται τα προγράμματα, των οποίων η υλοποίηση έγινε επίσης σε γλώσσα Python, των επιδράσεων των παραμέτρων στα μοντέλα απωλειών κλήσεων.

Κεφάλαιο 1

Μοντέλο Απωλειών MA-CAC με Ουρά Μεταπομπής

1.1 Εισαγωγή

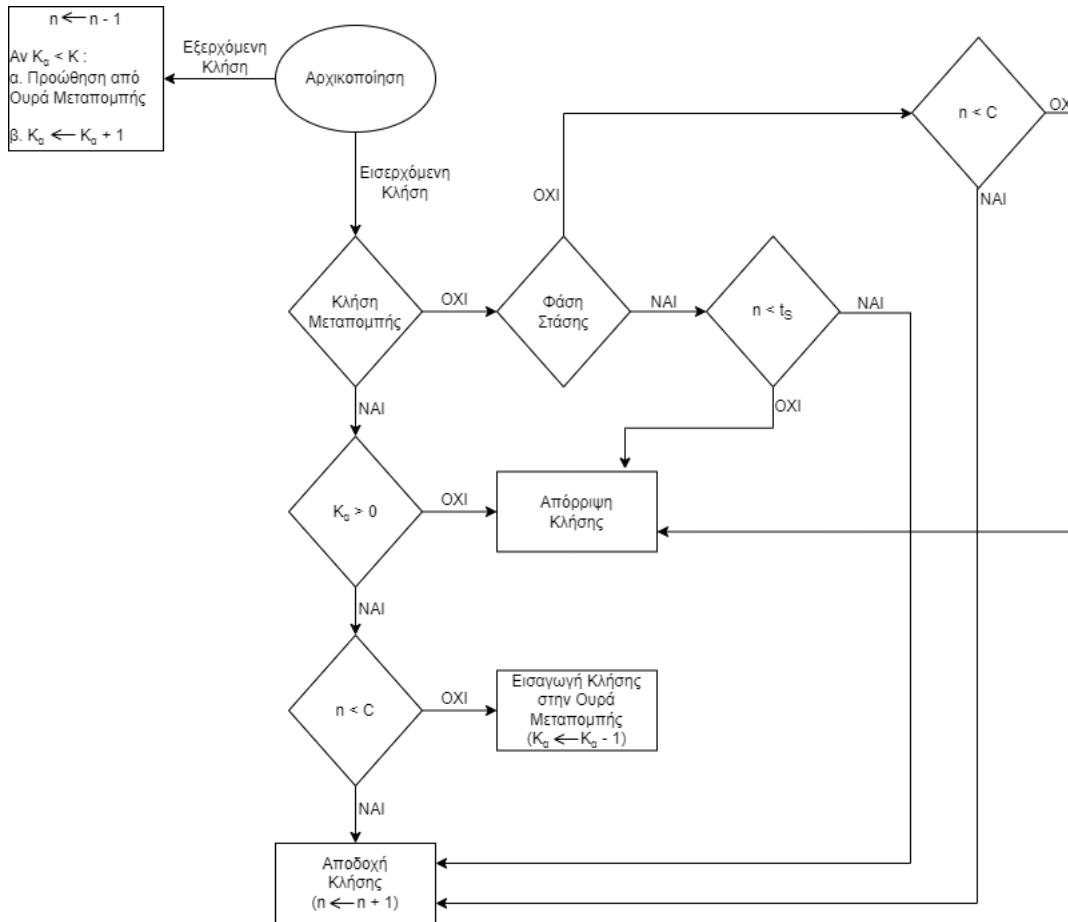
Σε αυτό το Κεφάλαιο παρουσιάζεται το σύστημα MA-CAC με ουρά μεταπομπής (Hand-over Queue - HQ). Πρόκειται για ένα μοντέλο στο οποίο ένας αλγόριθμος [1], που εφαρμόζεται στο σημείο πρόσβασης του δικτύου WLAN του οχήματος, είναι υπεύθυνος για την αποδοχή ή την απόρριψη τόσο των κλήσεων μεταπομπής όσο και των νέων κλήσεων που εισέρχονται στο σύστημα, έχοντας συγχρόνως επίγνωση της κινητικότητας του οχήματος.

Στην Ενότητα 1.2 περιγράφεται λεπτομερώς ο τρόπος λειτουργίας του αλγορίθμου MA-CAC με ουρά μεταπομπής και παρουσιάζεται το διάγραμμα ροής του. Στην Ενότητα 1.3 γίνεται η ανάλυση του μοντέλου, παρουσιάζεται η Μαρκοβιανή αλυσίδα συνεχούς χρόνου μέσω της οποίας γίνεται η αξιολόγηση του αλγορίθμου MA-CAC με ουρά μεταπομπής και παρατίθενται οι εξισώσεις ισορροπίας της αλυσίδας Markov. Στη συνέχεια, αναπτύσσεται αναλυτικά μία μέθοδος επίλυσης του συστήματος των εξισώσεων αυτών προκειμένου να καθοριστούν οι πιθανότητες μετάβασης από τη μία κατάσταση στην άλλη, μέσω των οποίων υπολογίζονται οι πιθανότητες απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής αντίστοιχα, το ποσοστό χρήσης των καναλιών του συστήματος, καθώς και ο μέσος χρόνος παραμονής των κλήσεων μεταπομπής στην ουρά. Τέλος, στην Ενότητα 1.4 δίνεται ένα αριθμητικό παράδειγμα του μοντέλου και τα αποτελέσματά του υπολογίζονται μέσω των εξισώσεων που παρουσιάστηκαν στην Ενότητα 1.3.

1.2 Ο Αλγόριθμος MA-CAC με Ουρά Μεταπομπής

Στο Σχήμα 1.1 παρουσιάζεται το διάγραμμα ροής του αλγορίθμου αποδοχής κλήσεων MA-CAC με ουρά μεταπομπής. Ο αριθμός των ενεργών κλήσεων, που εξυπηρετούνται από το σημείο πρόσβασης (Access Point - AP) του δικτύου WLAN, συμβολίζεται με n ενώ ο αριθμός των κενών θέσεων στην ουρά μεταπομπής συμβολίζεται με K_a .

Όταν μια κλήση εισέρχεται στο σύστημα, ελέγχεται αν είναι κλήση μεταπομπής ή νέα κλήση. Αν είναι κλήση μεταπομπής, γίνεται έλεγχος για την ύπαρξη διαθέσιμου



Σχήμα 1.1: Διάγραμμα Ροής του Αλγορίθμου MA-CAC με HQ.

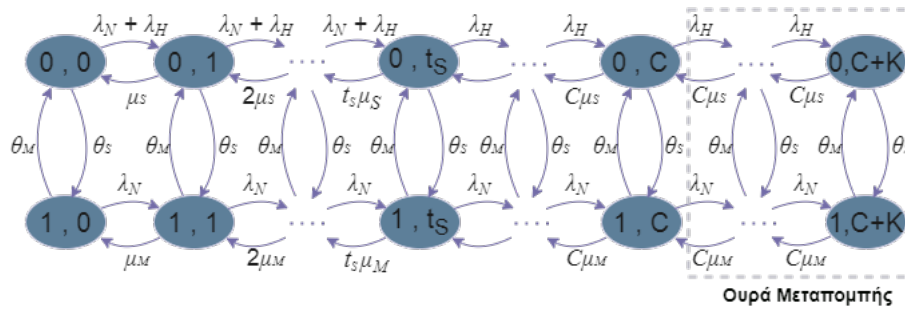
καναλιού προκειμένου να φιλοξενηθεί η κλήση και εξετάζεται αν υπάρχουν κενές θέσεις στην ουρά μεταπομπής. Στην περίπτωση που υπάρχει διαθέσιμο κανάλι η κλήση γίνεται αποδεκτή και εξυπηρετείται από το σύστημα, διαφορετικά η κλήση εισέρχεται στην ουρά μεταπομπής όπου παραμένει έως ότου απελευθερωθεί κάποιο από τα απασχολημένα κανάλια, προκειμένου να εξυπηρετηθεί. Αν δεν υπάρχουν είτε διαθέσιμα κανάλια είτε κενές θέσεις στην ουρά μεταπομπής, η κλήση απορρίπτεται.

Όταν η εισερχόμενη κλήση δεν είναι κλήση μεταπομπής, ο αλγόριθμος εξετάζει τη φάση στην οποία βρίσκεται το όχημα (φάση στάσης/φάση κίνησης). Στην περίπτωση που το όχημα βρίσκεται στη φάση στάσης, γίνεται έλεγχος της διαθεσιμότητας ελεύθερων καναλιών ανάμεσα στα κανάλια εκείνα που μπορούν να χρησιμοποιηθούν τόσο από κλήσεις μεταπομπής όσο και από νέες κλήσεις, προκειμένου η κλήση να γίνει αποδεκτή. Στην περίπτωση πάλι που το όχημα βρίσκεται σε κατάσταση κίνησης, ο έλεγχος διαθέσιμου καναλιού ώστε να γίνει αποδοχή της κλήσης, γίνεται ανάμεσα σε όλα τα κανάλια του συστήματος. Και στις δύο περιπτώσεις η κλήση απορρίπτεται αν όλα τα αντίστοιχα κανάλια είναι απασχολημένα.

Τέλος, όταν μια κλήση εξέρχεται του συστήματος, γίνεται απελευθέρωση του καναλιού που εκείνη χρησιμοποιούσε και αν υπάρχουν κλήσεις στην ουρά μεταπομπής, προωθείται στο διαθέσιμο πλέον κανάλι εκείνη η κλήση η οποία εισήλθε πρώτη στην ουρά (first in - first out queue).

1.3 Ανάλυση του μοντέλου

Το μοντέλο αποδοχής κλήσεων με επίγνωση της κινητικότητας του οχήματος, συμπεριλαμβανομένης της ουράς μεταπομπής (MA-CAC with HQ) περιέχει τις δύο φάσεις/στάδια που αναφέρθηκαν στην Ενότητα 1.2, οι οποίες αφορούν στην κινητικότητα του οχήματος: τη φάση κατά την οποία το όχημα είναι σταματημένο (φάση στάσης - stop phase) και τη φάση κατά την οποία το όχημα κινείται (φάση κίνησης - moving phase). Κατά τη διάρκεια της φάσης στάσης του οχήματος, δύνανται να εισέλθουν στο σύστημα τόσο κλήσεις μεταπομπής (handover calls) όσο και νέες κλήσεις (new calls) ενώ στη φάση κίνησης μπορούν να πραγματοποιηθούν μόνο νέες κλήσεις, καθώς δεν υπάρχει η δυνατότητα επιβίβασης ατόμων στο όχημα.



Σχήμα 1.2: Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για τον αλγόριθμο MA-CAC με HQ.

Θεωρούμε ότι οι διαδικασίες άφιξης, τόσο των κλήσεων μεταπομπής όσο και των νέων κλήσεων, ακολουθούν κατανομές Poisson με ρυθμούς λ_H και λ_N αντίστοιχα. Επιπλέον η μέση διάρκεια των κλήσεων ακολουθεί και στις δύο φάσεις του οχήματος εκθετική κατανομή, με τιμές $1/\mu_S$ για τη φάση στάσης και $1/\mu_M$ για τη φάση κίνησης. Επομένως ο ρυθμός άφιξης των κλήσεων, μέχρι την κατάσταση όπου δεν υπάρχουν άλλα κοινά κανάλια διαθέσιμα, είναι $\lambda_H + \lambda_N$. Επίσης ο χρόνος παραμονής, τόσο στη φάση στάσης όσο και στη φάση κίνησης, ακολουθεί εκθετική κατανομή με τιμές $1/\theta_S$ και $1/\theta_M$ αντίστοιχα. Τέλος, η χωρητικότητα του συστήματος είναι C μονάδες εύρους ζώνης (bandwidth units - b.u.), το μέγεθος της ουράς μεταπομπής είναι K ενώ ο αριθμός των καναλιών που διατίθενται από τον αλγόριθμο, τόσο για νέες κλήσεις όσο και για κλήσεις μεταπομπής, συμβολίζεται με t_S .

Το διάγραμμα μετάβασης από τη μία κατάσταση στην άλλη του MA-CAC με ουρά μεταπομπής, παρουσιάζεται στο Σχήμα 1.2. Οι καταστάσεις του συστήματος συμβολίζονται με (i, n) , όπου i είναι η φάση του οχήματος και n είναι ο αριθμός των κλήσεων του συστήματος. Το i δύναται να πάρει δύο διακριτές τιμές 0 και 1. Η τιμή $i = 0$, αντιπροσωπεύει την φάση στάσης, ενώ η τιμή $i = 1$ την φάση κίνησης του του οχήματος. Οι τιμές του n κυμαίνονται μεταξύ του 0 και του $C + K$ ($0 \leq n \leq C + K$), δηλαδή έως ότου καλυφθεί το άθροισμα της χωρητικότητας C του σημείου πρόσβασης του δικτύου WLAN με τη χωρητικότητα K της ουράς μεταπομπής.

Στο Σχήμα 1.2, παρατηρείται επίσης ότι ο ρυθμός μετάβασης από τη φάση στάσης στη φάση κίνησης είναι θ_S και αντίστροφα, από τη φάση στάσης στη φάση κίνησης είναι θ_M . Στη φάση στάσης, ο αλγόριθμος MA-CAC με ουρά μεταπομπής διατηρεί έναν σταθερό αριθμό $C - t_S$ καναλιών, προκειμένου να εξυπηρετήσει αποκλειστικά κλήσεις μεταπομπής. Αυτό έχει ως αποτέλεσμα όταν $i = 0$ και $0 \leq n < t_S$ να γίνονται αποδεκτές όλες οι κλήσεις (μεταπομπής και νέες) ενώ όταν $i = 0$ και $n \geq t_S$ να γίνονται αποδεκτές

μόνο οι κλήσεις μεταπομπής. Από την άλλη μεριά, στη φάση κίνησης του οχήματος δεν πραγματοποιούνται κλήσεις μεταπομπής. Γι' αυτό το λόγο, ο αλγόριθμος MA-CAC με ουρά μεταπομπής διαθέτει όλα τα μη απασχολημένα κανάλια του συστήματος για την πραγματοποίηση νέων κλήσεων.

Σε κάθε κατάσταση (i, n) , το ποσοστό αναχώρησης της κλήσης από εκείνη είναι $n \cdot \mu_S$ για τη φάση στάσης και $n \cdot \mu_M$ για τη φάση κίνησης αντίστοιχα, υπό την προϋπόθεση ο αριθμός των κλήσεων του συστήματος να είναι μικρότερος ή ίσος της χωρητικότητας του συστήματος ($n \leq C$). Στην περίπτωση που $C < n \leq C + K$, το ποσοστό αναχώρησης της κλήσης υπολογίζεται ως $C \cdot \mu_S$ και $C \cdot \mu_M$ για τις φάσεις στάσης και κίνησης αντίστοιχα.

Συνοπτικά, αν θεωρήσουμε ότι ο ρυθμός μετάβασης από μία κατάσταση (i, n) σε μια άλλη κατάσταση (i', n') είναι $r(i, n; i', n')$, έχουμε [1]:

$$\begin{aligned}
r(0, n; 1, n) &= \vartheta_S, & \text{για } 0 \leq n \leq C + K \\
r(1, n; 0, n) &= \vartheta_M, & \text{για } 0 \leq n \leq C + K \\
r(0, n; 0, n - 1) &= n \cdot \mu_S, & \text{για } 0 < n \leq C \\
r(0, n; 0, n - 1) &= C \cdot \mu_S, & \text{για } C < n \leq C + K \\
r(1, n; 1, n - 1) &= n \cdot \mu_M, & \text{για } 0 < n \leq C \\
r(1, n; 1, n - 1) &= C \cdot \mu_M, & \text{για } C < n \leq C + K \\
r(0, n; 0, n + 1) &= \hat{\lambda}_N + \hat{\lambda}_H, & \text{για } 0 \leq n < t_S \\
r(0, n; 0, n + 1) &= \hat{\lambda}_H, & \text{για } t_S \leq n < C + K \\
r(1, n; 1, n + 1) &= \hat{\lambda}_N, & \text{για } 0 \leq n < C
\end{aligned} \tag{1.1}$$

Θεωρώντας ότι η πιθανότητα να βρισκόμαστε στην κατάσταση (i, n) είναι $P(i, n)$, οι εξισώσεις ισορροπίας για τη Μαρκοβιανή αλυσίδα συνεχούς χρόνου του Σχήματος 1.2 είναι [1]:

1) Για $i = 0$ και $n = 0$ έχουμε:

$$((\hat{\lambda}_N + \hat{\lambda}_H) + \vartheta_S) P(i, n) = (n + 1) \mu_S P(i, n + 1) + \vartheta_M P(1, n)$$

2) Για $i = 0$ και $0 < n < t_S$ έχουμε:

$$((\hat{\lambda}_N + \hat{\lambda}_H) + n \mu_S + \vartheta_S) P(i, n) = (\hat{\lambda}_N + \hat{\lambda}_H) P(i, n - 1) + (n + 1) \mu_S P(i, n + 1) + \vartheta_M P(1, n)$$

3) Για $i = 0$ και $n = t_S$ έχουμε:

$$(\hat{\lambda}_H + n \mu_S + \vartheta_S) P(i, n) = (\hat{\lambda}_N + \hat{\lambda}_H) P(i, n - 1) + (n + 1) \mu_S P(i, n + 1) + \vartheta_M P(1, n)$$

4) Για $i = 0$ και $t_S < n < C$ έχουμε:

$$(\hat{\lambda}_H + n \mu_S + \vartheta_S) P(i, n) = \hat{\lambda}_H P(i, n - 1) + (n + 1) \mu_S P(i, n + 1) + \vartheta_M P(1, n)$$

5) Για $i = 0$ και $C \leq n < C + K$ έχουμε:

$$(\hat{\lambda}_H + C \mu_S + \vartheta_S) P(i, n) = \hat{\lambda}_H P(i, n - 1) + C \mu_S P(i, n + 1) + \vartheta_M P(1, n)$$

6) Για $i = 0$ και $n = C + K$ έχουμε:

$$(C\mu_S + \vartheta_S)P(i, n) = \hat{\rho}_H P(i, n - 1) + \vartheta_M P(1, n)$$

7) Για $i = 1$ και $n = 0$ έχουμε:

$$(\hat{\rho}_N + \vartheta_M)P(i, n) = (n + 1)\mu_M P(i, n + 1) + \vartheta_S P(0, n)$$

8) Για $i = 1$ και $0 < n < C$ έχουμε:

$$(\hat{\rho}_N + n\mu_M + \vartheta_M)P(i, n) = \hat{\rho}_N P(i, n - 1) + (n + 1)\mu_M P(i, n + 1) + \vartheta_S P(0, n)$$

9) Για $i = 1$ και $n = C$ έχουμε:

$$(n\mu_M + \vartheta_M)P(i, n) = \hat{\rho}_N P(i, n - 1) + C\mu_M P(i, n + 1) + \vartheta_S P(0, n)$$

10) Για $i = 1$ και $C < n < C + K$ έχουμε:

$$(C\mu_M + \vartheta_M)P(i, n) = C\mu_M P(i, n + 1) + \vartheta_S P(0, n)$$

11) Για $i = 1$ και $n = C + K$ έχουμε:

$$(C\mu_M + \vartheta_M)P(i, n) = \vartheta_S P(0, n) \tag{1.2}$$

Μια πιο γενική εξίσωση για να διατυπωθούν οι οριακές καταστάσεις $(0, 0)$ και $(1, 0)$ των εξισώσεων (1.2) είναι:

$$(\hat{\rho}_i(0) + \vartheta_i)P(i, 0) - \mu_i P(i, 1) - \vartheta_j P(j, 0) = 0 \tag{1.3}$$

Για τις ενδιάμεσες καταστάσεις $(1, 0), (2, 0), \dots, (C + K - 1, 0)$ και $(1, 1), (2, 1), \dots, (C + K - 1, 1)$, έχουμε:

• Για $0 < n < C$:

$$(\hat{\rho}_i(n) + n\mu_i + \vartheta_i)P(i, n) - \hat{\rho}_i(n - 1)P(i, n - 1) - \vartheta_j P(j, n) - (n + 1)\mu_i P(i, n + 1) = 0 \tag{1.4}$$

• Για $C \leq n < C + K$:

$$(\hat{\rho}_i(n) + C\mu_i + \vartheta_i)P(i, n) - \hat{\rho}_i(n - 1)P(i, n - 1) - \vartheta_j P(j, n) - C\mu_i P(i, n + 1) = 0 \tag{1.5}$$

Τέλος για τις οριακές καταστάσεις $(0, C + K)$ και $(1, C + K)$, έχουμε:

$$(C\mu_i + \vartheta_i)P(i, C + K) - \hat{\rho}_i(C + K - 1)P(i, C + K - 1) - \vartheta_j P(j, C + K) = 0 \tag{1.6}$$

όπου:

$$i = \begin{cases} 0, & \text{για τη φάση στάσης} \\ 1, & \text{για τη φάση κίνησης} \end{cases} \quad \mu_i = \begin{cases} \mu_S, & \text{όταν } i = 0 \\ \mu_M, & \text{όταν } i = 1 \end{cases}$$

$$\vartheta_i = \begin{cases} \vartheta_S, & \text{όταν } i = 0 \\ \vartheta_M, & \text{όταν } i = 1 \end{cases} \quad \hat{\lambda}_i(n) = \begin{cases} \hat{\lambda}_N + \hat{\lambda}_H, & \text{όταν } i = 0 \text{ και } 0 \leq n < t_S \\ \hat{\lambda}_H, & \text{όταν } i = 0 \text{ και } t_S \leq n < C + K \\ \hat{\lambda}_N, & \text{όταν } i = 1 \text{ και } 0 \leq n < C + K \\ 0, & \text{όταν } n < 0 \end{cases}$$

και $j = 1 - i$

Για να υπολογίσουμε πιθανότητες $P(i, n)$ των καταστάσεων του συστήματος, τις εκφράζουμε μέσω των πιθανοτήτων των οριακών καταστάσεων $P(0, 0)$ και $P(1, 0)$, ως εξής:

$$P(i, n) = S_{i,n}^0 \cdot P(0, 0) + S_{i,n}^1 \cdot P(1, 0) \quad (1.7)$$

Για τον προσδιορισμό των συντελεστών $S_{i,n}^0$ και $S_{i,n}^1$ της εξίσωσης (1.7), την αντικαθιστούμε στις σχέσεις (1.4) και (1.5) και προκύπτουν οι ακόλουθες σχέσεις:

- Για $0 < n < C$:

$$\begin{aligned} S_{i,n+1}^0 \cdot P(0, 0) + S_{i,n+1}^1 \cdot P(1, 0) &= \frac{\hat{\lambda}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot (S_{i,n}^0 \cdot P(0, 0) + S_{i,n}^1 \cdot P(1, 0)) \\ &\quad - \frac{\hat{\lambda}_i(n-1)}{(n+1)\mu_i} \cdot ((S_{i,n-1}^0 \cdot P(0, 0) + S_{i,n-1}^1 \cdot P(1, 0)) \\ &\quad - \frac{\vartheta_j}{(n+1)\mu_i} \cdot ((S_{j,n}^0 \cdot P(0, 0) + S_{j,n}^1 \cdot P(1, 0)) \end{aligned} \quad (1.8)$$

- Για $C \leq n < C + K$:

$$\begin{aligned} S_{i,n+1}^0 \cdot P(0, 0) + S_{i,n+1}^1 \cdot P(1, 0) &= \frac{\hat{\lambda}_i(n) + C\mu_i + \vartheta_i}{C\mu_i} \cdot (S_{i,n}^0 \cdot P(0, 0) + S_{i,n}^1 \cdot P(1, 0)) \\ &\quad - \frac{\hat{\lambda}_i(n-1)}{C\mu_i} \cdot ((S_{i,n-1}^0 \cdot P(0, 0) + S_{i,n-1}^1 \cdot P(1, 0)) \\ &\quad - \frac{\vartheta_j}{C\mu_i} \cdot ((S_{j,n}^0 \cdot P(0, 0) + S_{j,n}^1 \cdot P(1, 0)) \end{aligned} \quad (1.9)$$

Για τον υπολογισμό των τιμών $S_{i,n}^0$, υποθέτουμε ότι $P(0, 0) = 1$ και $P(1, 0) = 0$ και αντικαθιστούμε στις σχέσεις (1.8) και (1.9). Έτσι προκύπτουν οι αναδρομικές σχέσεις:

$$S_{i,n+1}^0 = \frac{\hat{\lambda}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot S_{i,n}^0 - \frac{\hat{\lambda}_i(n-1)}{(n+1)\mu_i} \cdot S_{i,n-1}^0 - \frac{\vartheta_j}{(n+1)\mu_i} \cdot S_{j,n}^0 \quad (1.10)$$

και

$$S_{i,n+1}^0 = \frac{\hat{\lambda}_i(C) + n\mu_i + \vartheta_i}{C\mu_i} \cdot S_{i,n}^0 - \frac{\hat{\lambda}_i(n-1)}{C\mu_i} \cdot S_{i,n-1}^0 - \frac{\vartheta_j}{C\mu_i} \cdot S_{j,n}^0 \quad (1.11)$$

όπου: $S_{0,0}^0 = 1$, $S_{1,0}^0 = 0$ και $S_{0,y}^0 = 0$, $S_{1,y}^0 = 0$ για $y < 0$

Αντίστοιχα, για τον υπολογισμό των τιμών $S_{i,n}^1$, θεωρούμε ότι $P(0,0) = 0$ και $P(1,0) = 1$. Αντικαθιστώντας αυτές τις τιμές στις εξισώσεις (1.8) και (1.9), προκύπτουν οι ακόλουθες αναδρομικές σχέσεις:

$$S_{i,n+1}^1 = \frac{\hat{\lambda}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot S_{i,n}^1 - \frac{\hat{\lambda}_i(n-1)}{(n+1)\mu_i} \cdot S_{i,n-1}^1 - \frac{\vartheta_j}{(n+1)\mu_i} \cdot S_{j,n}^1 \quad (1.12)$$

και

$$S_{i,n+1}^1 = \frac{\hat{\lambda}_i(C) + n\mu_i + \vartheta_i}{C\mu_i} \cdot S_{i,n}^1 - \frac{\hat{\lambda}_i(n-1)}{C\mu_i} \cdot S_{i,n-1}^1 - \frac{\vartheta_j}{C\mu_i} \cdot S_{j,n}^1 \quad (1.13)$$

όπου: $S_{0,0}^1 = 0$, $S_{1,0}^1 = 1$ και $S_{0,y}^1 = 0$, $S_{1,y}^1 = 0$ για $y < 0$

Στη συνέχεια, αφού έχουν πλέον υπολογιστεί οι συντελεστές $S_{i,n}^0$ και $S_{i,n}^1$, πρέπει να προσδιοριστούν οι τιμές των πιθανοτήτων $P(0,0)$ και $P(1,0)$. Για την επίτευξη αυτού του στόχου, δημιουργούμε ένα σύστημα δύο εξισώσεων σύμφωνα με την ακόλουθη διαδικασία:

Στην εξίσωση (1.6), θεωρούμε την περίπτωση όπου $i = 0$ κι έτσι:

$$(C\mu_S + \vartheta_S)P(0, C+K) - \hat{\lambda}_0(C+K-1)P(0, C+K-1) - \vartheta_M P(1, C+K) = 0 \quad (1.14)$$

(Σημειώνουμε ότι και για την περίπτωση $i = 1$ προκύπτουν τα ίδια αποτελέσματα)

Από τον συνδυασμό των σχέσεων (1.14) και (1.7), έχουμε:

$$\begin{aligned} & (C\mu_S + \vartheta_S) \cdot (S_{0,C+K}^0 \cdot P(0,0) + S_{0,C+K}^1 \cdot P(1,0)) \\ & - \hat{\lambda}_0(C+K-1) \cdot (S_{0,C+K-1}^0 \cdot P(0,0) + S_{0,C+K-1}^1 \cdot P(1,0)) \\ & - \vartheta_M \cdot (S_{1,C+K}^0 \cdot P(0,0) + S_{1,C+K}^1 \cdot P(1,0)) \\ & = 0 \Rightarrow \\ & P(0,0) \cdot (S_{0,C+K}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C+K-1}^0 \cdot \lambda_0 - S_{1,C+K}^0 \cdot \vartheta_M) \\ & + P(1,0) \cdot (S_{0,C+K}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C+K-1}^1 \cdot \lambda_0 - S_{1,C+K}^1 \cdot \vartheta_M) \\ & = 0 \Rightarrow \\ & \frac{P(1,0)}{P(0,0)} = - \frac{S_{0,C+K}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C+K-1}^0 \cdot \lambda_0 - S_{1,C+K}^0 \cdot \vartheta_M}{S_{0,C+K}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C+K-1}^1 \cdot \lambda_0 - S_{1,C+K}^1 \cdot \vartheta_M} \end{aligned} \quad (1.15)$$

Θέτοντας στην εξίσωση (1.15):

$$r = -\frac{S_{0,C+K}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C+K-1}^0 \cdot \hat{\rho}_0 - S_{1,C+K}^0 \cdot \vartheta_M}{S_{0,C+K}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C+K-1}^1 \cdot \hat{\rho}_0 - S_{1,C+K}^1 \cdot \vartheta_M} \quad (1.16)$$

προκύπτει η σχέση:

$$\frac{P(1,0)}{P(0,0)} = r \quad (1.17)$$

Για τη δημιουργία της 2ης εξίσωσης, βασιζόμαστε στο γεγονός:

$$\sum_{i=0}^1 \sum_{n=0}^{C+K} P(i,n) = 1 \quad (1.18)$$

Επομένως, λόγω της σχέσης (1.7), η εξίσωση (1.18) γίνεται:

$$\begin{aligned} P(0,0) \cdot \sum_{i=0}^1 \sum_{n=0}^{C+K} S_{i,n}^0 + P(1,0) \cdot \sum_{i=0}^1 \sum_{n=0}^{C+K} S_{i,n}^1 &= 1 \\ \Rightarrow \sum_{i=0}^1 \sum_{n=0}^{C+K} S_{i,n}^0 + \frac{P(1,0) \cdot \sum_{i=0}^1 \sum_{n=0}^{C+K} S_{i,n}^1}{P(0,0)} &= \frac{1}{P(0,0)} \\ \Rightarrow P(0,0) &= \left(\sum_{i=0}^1 \sum_{n=0}^{C+K} S_{i,n}^0 + r \cdot \sum_{i=0}^1 \sum_{n=0}^{C+K} S_{i,n}^1 \right)^{-1} \end{aligned} \quad (1.19)$$

Με τον υπολογισμό των πιθανοτήτων $P(i,n)$ των καταστάσεων του συστήματος, έχουμε τη δυνατότητα να υπολογίσουμε τις πιθανότητες απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής αντίστοιχα.

Η πιθανότητα απώλειας των νέων κλήσεων, όταν το όχημα βρίσκεται στη φάση στάσης, μπορεί να υπολογιστεί από τη σχέση:

$$B_S^N = \sum_{n=S}^{C+K} P(0,n) \quad (1.20)$$

καθώς μια νέα κλήση απορρίπτεται, όταν ο αριθμός των κατειλημμένων καναλιών είναι μεγαλύτερος ή ίσος του αριθμού των κοινών καναλιών των νέων κλήσεων και των κλήσεων μεταπομπής.

Στη φάση κατά την οποία το όχημα βρίσκεται σε κίνηση, η πιθανότητα απώλειας των νέων κλήσεων καθορίζεται από τη σχέση [1]:

$$B_M^N = \sum_{n=C}^{C+K} P(1,n) \quad (1.21)$$

καθώς σε αυτή τη φάση μία νέα κλήση απορρίπτεται μόνο όταν όλα τα κανάλια του συστήματος είναι απασχολημένα.

Επομένως, η συνολική πιθανότητα απώλειας των νέων κλήσεων του συστήματος, προκύπτει από τη σχέση:

$$P_{NCBP} = B_S^N + B_M^N \quad (1.22)$$

Όσον αφορά την πιθανότητα απόρριψης των κλήσεων μεταπομπής, λαμβάνουμε υπόψη το γεγονός ότι το φαινόμενο αυτό πραγματοποιείται μόνο κατά τη διάρκεια της φάσης στάσης του οχήματος και επιπλέον, μόνο όταν όλα τα κανάλια του συστήματος είναι κατειλημμένα. Επομένως, είναι μια δεσμευμένη πιθανότητα και για τον υπολογισμό της χρησιμοποιούμε τον παρακάτω τύπο [1]:

$$P_{HCDP} = P(n = C + K/i = 0) = \frac{P(0, C + K)}{\sum_{n=0}^{C+K} P(0, n)} \quad (1.23)$$

Ο υπολογισμός του ποσοστού χρήσης των καναλιών (channel Utilization - U), προκύπτει από τον λόγο του αναμενόμενου αριθμού των καναλιών που χρησιμοποιούνται προς τη συνολική χωρητικότητα του συστήματος. Επομένως, έχουμε [1]:

$$U = \frac{\sum_{i=0}^1 \sum_{n=0}^C n \cdot P(i, n) + \sum_{i=0}^1 \sum_{n=C+1}^{C+K} C \cdot P(i, n)}{C} \quad (1.24)$$

Τέλος, ο χρόνος παραμονής των κλήσεων στην ουρά μεταπομπής μπορεί να υπολογιστεί σύμφωνα με τον νόμο του Little [17], ως εξής:

$$W_Q = \frac{N_Q}{\hat{\lambda}_e} \quad (1.25)$$

όπου με N_Q συμβολίζεται ο μέσος αριθμός των κλήσεων στην ουρά μεταπομπής, ενώ με $\hat{\lambda}_e$ ο πραγματικός ρυθμός άφιξης των κλήσεων στο σύστημα. Τα N_Q και $\hat{\lambda}_e$ υπολογίζονται με τη χρήση των ακόλουθων τύπων [1]:

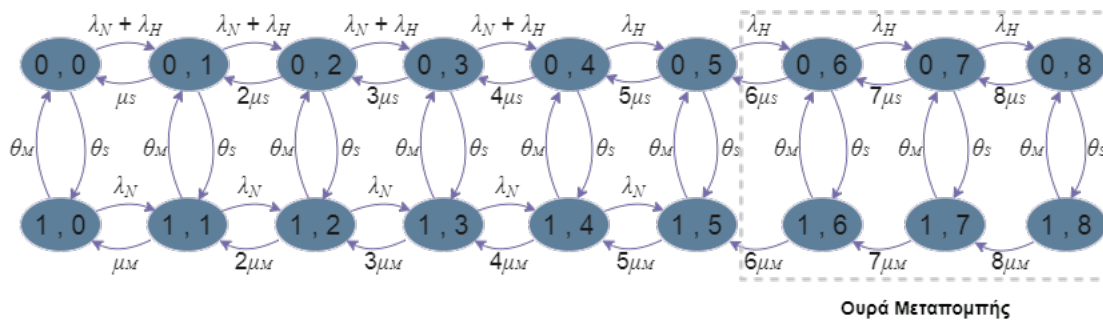
$$N_Q = \sum_{i=0}^1 \sum_{n=C+1}^{C+K} (n - C) \cdot P(i, n) \quad (1.26)$$

και

$$\hat{\lambda}_e = \sum_{n=0}^{S-1} (\hat{\lambda}_N + \hat{\lambda}_H) \cdot P(0, n) + \sum_{n=S}^{C+K-1} \hat{\lambda}_H \cdot P(0, n) + \sum_{n=0}^{C-1} \hat{\lambda}_N \cdot P(1, n) \quad (1.27)$$

1.4 Παράδειγμα

Σε αυτή την ενότητα, παρουσιάζεται ένα παράδειγμα του μοντέλου MA-CAC με ουρά μεταπομπής. Η χωρητικότητα του συστήματος είναι $C = 5$, ο αριθμός των κοινών καναλιών που χρησιμοποιούνται τόσο από νέες όσο και από κλήσεις μεταπομπής είναι $t_S = 4$ και το μέγεθος της ουράς μεταπομπής είναι $K = 3$. Όσον αφορά τους ρυθμούς άφιξης των κλήσεων μεταπομπής και των νέων κλήσεων, ακολουθούν κατανομή Poisson με τιμές $\hat{\lambda}_H = 3$ και $\hat{\lambda}_N = 2$ αντίστοιχα. Η μέση διάρκεια των κλήσεων, τόσο στη φάση στάσης όσο στη φάση κίνησης του οχήματος, ακολουθεί εκθετική κατανομή με ρυθμό $\mu_S = 0,75$ για τη φάση στάσης και $\mu_M = 0,5$ για τη φάση κίνησης. Τέλος, οι μέσοι



Σχήμα 1.3: Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για τον αλγόριθμο MA-CAC με HQ του Παραδείγματος 1.4.

χρόνοι παραμονής και στις δύο φάσεις ακολουθούν εκθετική κατανομή και οι ρυθμοί τους έχουν τιμές $\vartheta_S = 0, 1$ για τη φάση στάσης και $\vartheta_M = 0, 05$ για τη φάση κατά την οποία το όχημα βρίσκεται σε κίνηση.

Στο Σχήμα 1.3, απεικονίζεται η Μαρκοβιανή αλυσίδα συνεχούς χρόνου των καταστάσεων μετάβασης του μοντέλου. Για τη φάση στάσης, ο ρυθμός άφιξης των κλήσεων από τη κατάσταση $(0, 0)$ έως την κατάσταση $(0, 4)$, είναι $\lambda_H + \lambda_N$, ενώ από την κατάσταση $(0, 4)$ έως την κατάσταση $(0, 8)$ είναι λ_H , καθώς ένα κανάλι δεσμεύεται υπέρ των κλήσεων μεταπομπής και οι διαθέσιμες θέσεις στην ουρά μεταπομπής είναι τρεις. Επιπλέον, όσο το όχημα κινείται, οι κλήσεις έως την κατάσταση $(1, 5)$ αφικνούνται με ρυθμό λ_N , δηλαδή έως ότου καλυφθεί η χωρητικότητα του συστήματος, για το λόγο ότι σε αυτή τη φάση η ουρά μεταπομπής παραμένει ανενεργή. Σε ότι αφορά τους ρυθμούς εξυπηρέτησης των κλήσεων από την κάθε κατάσταση, οι τιμές τους είναι $n \cdot \mu_S$ για τις αναχωρήσεις στη φάση στάσης και $n \cdot \mu_M$ για εκείνες της φάσης κίνησης, με την προϋπόθεση ότι ο αριθμός των ενεργών κλήσεων είναι μικρότερος ή ίσος της χωρητικότητας του συστήματος (δηλαδή για $0 \leq n \leq 5$). Στην περίπτωση των καταστάσεων εντός ουράς μεταπομπής, οι προαναφερθέντες ρυθμοί έχουν τιμές $C \cdot \mu_S$ και $C \cdot \mu_M$ για τις φάσεις στάσης και κίνησης αντιστοίχως (δηλαδή για $5 < n \leq 8$).

Σύμφωνα με τα παραπάνω στοιχεία, οι εξισώσεις ισορροπίας των καταστάσεων του συστήματος MA-CAC με ουρά μεταπομπής, είναι:

- Για την κατάσταση $(0, 0)$:

$$\mu_S \cdot P(0, 1) + \vartheta_M \cdot P(1, 0) = (\lambda_H + \lambda_N + \vartheta_S) \cdot P(0, 0)$$

$$\Rightarrow 0, 75 \cdot P(0, 1) + 0, 05 \cdot P(1, 0) = 5, 1 \cdot P(0, 0)$$

$$\Rightarrow 0, 75 \cdot P(0, 1) + 0, 05 \cdot P(1, 0) - 5, 1 \cdot P(0, 0) = 0 \quad (1)$$

- Για την κατάσταση (1, 0):

$$\begin{aligned}\mu_M \cdot P(1, 1) + \vartheta_S \cdot P(0, 0) &= (\hat{\lambda}_N + \vartheta_M) \cdot P(1, 0) \\ \Rightarrow 0,5 \cdot P(1, 1) + 0,1 \cdot P(0, 0) &= 2,05 \cdot P(1, 0) \\ \Rightarrow 0,5 \cdot P(1, 1) + 0,1 \cdot P(0, 0) - 2,05 \cdot P(1, 0) &= 0\end{aligned}\tag{2}$$

- Για την κατάσταση (0, 1):

$$\begin{aligned}(\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 0) + 2\mu_S \cdot P(0, 2) + \vartheta_M \cdot P(1, 1) &= (\hat{\lambda}_H + \hat{\lambda}_N + \mu_S + \vartheta_S) \cdot P(0, 1) \\ \Rightarrow 5 \cdot P(0, 0) + 1,5 \cdot P(0, 2) + 0,05 \cdot P(1, 1) &= 5,85 \cdot P(0, 1) \\ \Rightarrow 5 \cdot P(0, 0) + 1,5 \cdot P(0, 2) + 0,05 \cdot P(1, 1) - 5,85 \cdot P(0, 1) &= 0\end{aligned}\tag{3}$$

- Για την κατάσταση (1, 1):

$$\begin{aligned}\hat{\lambda}_N \cdot P(1, 0) + 2\mu_M \cdot P(1, 2) + \vartheta_S \cdot P(0, 1) &= (\hat{\lambda}_N + \mu_M + \vartheta_M) \cdot P(1, 1) \\ \Rightarrow 2 \cdot P(1, 0) + 1 \cdot P(1, 2) + 0,1 \cdot P(0, 1) &= 2,55 \cdot P(1, 1) \\ \Rightarrow 2 \cdot P(1, 0) + 1 \cdot P(1, 2) + 0,1 \cdot P(0, 1) - 2,55 \cdot P(1, 1) &= 0\end{aligned}\tag{4}$$

- Για την κατάσταση (0, 2):

$$\begin{aligned}(\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 1) + 3\mu_S \cdot P(0, 3) + \vartheta_M \cdot P(1, 2) &= (\hat{\lambda}_H + \hat{\lambda}_N + 2\mu_S + \vartheta_S) \cdot P(0, 2) \\ \Rightarrow 5 \cdot P(0, 1) + 2,25 \cdot P(0, 3) + 0,05 \cdot P(1, 2) &= 6,6 \cdot P(0, 2) \\ \Rightarrow 5 \cdot P(0, 1) + 2,25 \cdot P(0, 3) + 0,05 \cdot P(1, 2) - 6,6 \cdot P(0, 2) &= 0\end{aligned}\tag{5}$$

- Για την κατάσταση (1, 2):

$$\begin{aligned}\hat{\lambda}_N \cdot P(1, 1) + 3\mu_M \cdot P(1, 3) + \vartheta_S \cdot P(0, 2) &= (\hat{\lambda}_N + 2\mu_M + \vartheta_M) \cdot P(1, 2) \\ \Rightarrow 2 \cdot P(1, 1) + 1,5 \cdot P(1, 3) + 0,1 \cdot P(0, 2) &= 3,05 \cdot P(1, 2) \\ \Rightarrow 2 \cdot P(1, 1) + 1,5 \cdot P(1, 3) + 0,1 \cdot P(0, 2) - 3,05 \cdot P(1, 2) &= 0\end{aligned}\tag{6}$$

- Για την κατάσταση (0, 3):

$$(\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 2) + 4\mu_S \cdot P(0, 4) + \vartheta_M \cdot P(1, 3) = (\hat{\lambda}_H + \hat{\lambda}_N + 3\mu_S + \vartheta_S) \cdot P(0, 3)$$

$$\Rightarrow 5 \cdot P(0, 2) + 3 \cdot P(0, 4) + 0,05 \cdot P(1, 3) = 7,35 \cdot P(0, 3)$$

$$\Rightarrow 5 \cdot P(0, 2) + 3 \cdot P(0, 4) + 0,05 \cdot P(1, 3) - 7,35 \cdot P(0, 3) = 0 \quad (7)$$

- Για την κατάσταση (1, 3):

$$\hat{\lambda}_N \cdot P(1, 2) + 4\mu_M \cdot P(1, 4) + \vartheta_S \cdot P(0, 3) = (\hat{\lambda}_N + 3\mu_M + \vartheta_M) \cdot P(1, 3)$$

$$\Rightarrow 2 \cdot P(1, 2) + 2 \cdot P(1, 4) + 0,1 \cdot P(0, 3) = 3,55 \cdot P(1, 3)$$

$$\Rightarrow 2 \cdot P(1, 2) + 2 \cdot P(1, 4) + 0,1 \cdot P(0, 3) - 3,55 \cdot P(1, 3) = 0 \quad (8)$$

- Για την κατάσταση (0, 4):

$$(\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 3) + 5\mu_S \cdot P(0, 5) + \vartheta_M \cdot P(1, 4) = (\hat{\lambda}_H + 4\mu_S + \vartheta_S) \cdot P(0, 4)$$

$$\Rightarrow 5 \cdot P(0, 3) + 3,75 \cdot P(0, 5) + 0,05 \cdot P(1, 4) = 6,1 \cdot P(0, 4)$$

$$\Rightarrow 5 \cdot P(0, 3) + 3,75 \cdot P(0, 5) + 0,05 \cdot P(1, 4) - 6,1 \cdot P(0, 4) = 0 \quad (9)$$

- Για την κατάσταση (1, 4):

$$\hat{\lambda}_N \cdot P(1, 3) + 5\mu_M \cdot P(1, 5) + \vartheta_S \cdot P(0, 4) = (\hat{\lambda}_N + 4\mu_M + \vartheta_M) \cdot P(1, 4)$$

$$\Rightarrow 2 \cdot P(1, 3) + 2,5 \cdot P(1, 5) + 0,1 \cdot P(0, 4) = 4,05 \cdot P(1, 4)$$

$$\Rightarrow 2 \cdot P(1, 3) + 2,5 \cdot P(1, 5) + 0,1 \cdot P(0, 4) - 4,05 \cdot P(1, 4) = 0 \quad (10)$$

- Για την κατάσταση (0, 5):

$$\hat{\lambda}_H \cdot P(0, 4) + 5\mu_S \cdot P(0, 6) + \vartheta_M \cdot P(1, 5) = (\hat{\lambda}_H + 5\mu_S + \vartheta_S) \cdot P(0, 5)$$

$$\Rightarrow 3 \cdot P(0, 4) + 3,75 \cdot P(0, 6) + 0,05 \cdot P(1, 5) = 6,85 \cdot P(0, 5)$$

$$\Rightarrow 3 \cdot P(0, 4) + 3,75 \cdot P(0, 6) + 0,05 \cdot P(1, 5) - 6,85 \cdot P(0, 5) = 0 \quad (11)$$

- Για την κατάσταση (1, 5):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 4) + 5\mu_M \cdot P(1, 6) + \vartheta_S \cdot P(0, 5) &= (5\mu_M + \vartheta_M) \cdot P(1, 5) \\ \Rightarrow 2 \cdot P(1, 4) + 2,5 \cdot P(1, 6) + 0,1 \cdot P(0, 5) &= 2,55 \cdot P(1, 5) \\ \Rightarrow 2 \cdot P(1, 4) + 2,5 \cdot P(1, 6) + 0,1 \cdot P(0, 5) - 2,55 \cdot P(1, 5) &= 0 \end{aligned} \quad (12)$$

- Για την κατάσταση (0, 6):

$$\begin{aligned} \hat{\lambda}_H \cdot P(0, 5) + 5\mu_S \cdot P(0, 7) + \vartheta_M \cdot P(1, 6) &= (\hat{\lambda}_H + 5\mu_S + \vartheta_S) \cdot P(0, 6) \\ \Rightarrow 3 \cdot P(0, 5) + 3,75 \cdot P(0, 7) + 0,05 \cdot P(1, 6) &= 6,85 \cdot P(0, 6) \\ \Rightarrow 3 \cdot P(0, 5) + 3,75 \cdot P(0, 7) + 0,05 \cdot P(1, 6) - 6,85 \cdot P(0, 6) &= 0 \end{aligned} \quad (13)$$

- Για την κατάσταση (1, 6):

$$\begin{aligned} 5\mu_M \cdot P(1, 7) + \vartheta_S \cdot P(0, 6) &= (5\mu_M + \vartheta_M) \cdot P(1, 6) \\ \Rightarrow 2,5 \cdot P(1, 7) + 0,1 \cdot P(0, 6) &= 2,55 \cdot P(1, 6) \\ \Rightarrow 2,5 \cdot P(1, 7) + 0,1 \cdot P(0, 6) - 2,55 \cdot P(1, 6) &= 0 \end{aligned} \quad (14)$$

- Για την κατάσταση (0, 7):

$$\begin{aligned} \hat{\lambda}_H \cdot P(0, 6) + 5\mu_S \cdot P(0, 8) + \vartheta_M \cdot P(1, 7) &= (\hat{\lambda}_H + 5\mu_S + \vartheta_S) \cdot P(0, 7) \\ \Rightarrow 3 \cdot P(0, 6) + 3,75 \cdot P(0, 8) + 0,05 \cdot P(1, 7) &= 6,85 \cdot P(0, 7) \\ \Rightarrow 3 \cdot P(0, 6) + 3,75 \cdot P(0, 8) + 0,05 \cdot P(1, 7) - 6,85 \cdot P(0, 7) &= 0 \end{aligned} \quad (15)$$

- Για την κατάσταση (1, 7):

$$\begin{aligned} 5\mu_M \cdot P(1, 8) + \vartheta_S \cdot P(0, 7) &= (5\mu_M + \vartheta_M) \cdot P(1, 7) \\ \Rightarrow 2,5 \cdot P(1, 8) + 0,1 \cdot P(0, 7) &= 2,55 \cdot P(1, 7) \\ \Rightarrow 2,5 \cdot P(1, 8) + 0,1 \cdot P(0, 7) - 2,55 \cdot P(1, 7) &= 0 \end{aligned} \quad (16)$$

- Για την κατάσταση (0, 8):

$$\begin{aligned}
 \hat{\lambda}_H \cdot P(0, 7) + \vartheta_M \cdot P(1, 8) &= (5\mu_S + \vartheta_S) \cdot P(0, 8) \\
 \Rightarrow 3 \cdot P(0, 7) + 0,05 \cdot P(1, 8) &= 3,85 \cdot P(0, 8) \\
 \Rightarrow 3 \cdot P(0, 7) + 0,05 \cdot P(1, 8) - 3,85 \cdot P(0, 8) &= 0
 \end{aligned} \tag{17}$$

- Για την κατάσταση (1, 8):

$$\begin{aligned}
 \vartheta_S \cdot P(0, 8) &= (5\mu_M + \vartheta_M) \cdot P(1, 8) \\
 \Rightarrow 2,5 \cdot P(0, 8) &= 2,55 \cdot P(1, 8) \\
 \Rightarrow 2,5 \cdot P(0, 8) - 2,55 \cdot P(1, 8) &= 0
 \end{aligned} \tag{18}$$

Για να υπολογίσουμε τους συντελεστές $S_{i,n}^0$, χρησιμοποιούμε τις αναδρομικές σχέσεις (1.10) και (1.11), θεωρώντας ότι $S_{0,0}^0 = 1$, $S_{1,0}^0 = 0$ και $S_{0,y}^0 = 0$, $S_{1,y}^0 = 0$ για $y < 0$. Έτσι έχουμε τα ακόλουθα αποτελέσματα:

- Για $i = 0$ και $n = 0$:

$$\begin{aligned}
 S_{0,1}^0 &= \frac{\hat{\lambda}_0(0) + 0 \cdot \mu_0 + \vartheta_0}{\mu_0} \cdot S_{0,0}^0 - \frac{\hat{\lambda}_0(-1)}{\mu_0} \cdot S_{0,-1}^0 - \frac{\vartheta_1}{\mu_0} \cdot S_{1,0}^0 \\
 &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + \vartheta_S}{\mu_S} \cdot S_{0,0}^0 = 6,8
 \end{aligned}$$

- Για $i = 1$ και $n = 0$:

$$\begin{aligned}
 S_{1,1}^0 &= \frac{\hat{\lambda}_1(0) + 0 \cdot \mu_1 + \vartheta_1}{\mu_1} \cdot S_{1,0}^0 - \frac{\hat{\lambda}_1(-1)}{\mu_1} \cdot S_{1,-1}^0 - \frac{\vartheta_0}{\mu_1} \cdot S_{0,0}^0 \\
 &= -\frac{\vartheta_S}{\mu_M} \cdot S_{0,0}^0 = -0,2
 \end{aligned}$$

- Για $i = 0$ και $n = 1$:

$$S_{0,2}^0 = \frac{\hat{\lambda}_0(1) + 1 \cdot \mu_0 + \vartheta_0}{2\mu_0} \cdot S_{0,1}^0 - \frac{\hat{\lambda}_0(0)}{2\mu_0} \cdot S_{0,0}^0 - \frac{\vartheta_1}{2\mu_0} \cdot S_{1,1}^0$$

$$\begin{aligned}
&= \frac{\hat{\lambda}_H + \hat{\lambda}_N + \mu_S + \vartheta_S}{2\mu_S} \cdot S_{0,1}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{2\mu_S} \cdot S_{0,0}^0 - \frac{\vartheta_M}{2\mu_S} \cdot S_{1,1}^0 \\
&= 23,19333333
\end{aligned}$$

- Για $i = 1$ και $n = 1$:

$$\begin{aligned}
S_{1,2}^0 &= \frac{\hat{\lambda}_1(1) + 1 \cdot \mu_1 + \vartheta_1}{2\mu_1} \cdot S_{1,1}^0 - \frac{\hat{\lambda}_1(0)}{2\mu_1} \cdot S_{1,0}^0 - \frac{\vartheta_0}{2\mu_1} \cdot S_{0,1}^0 \\
&= \frac{\hat{\lambda}_N + \mu_M + \vartheta_M}{2\mu_M} \cdot S_{1,1}^0 - \frac{\hat{\lambda}_N}{2\mu_M} \cdot S_{1,0}^0 - \frac{\vartheta_S}{2\mu_M} \cdot S_{0,1}^0 \\
&= -1,19
\end{aligned}$$

- Για $i = 0$ και $n = 2$:

$$\begin{aligned}
S_{0,3}^0 &= \frac{\hat{\lambda}_0(2) + 2 \cdot \mu_0 + \vartheta_0}{3\mu_0} \cdot S_{0,2}^0 - \frac{\hat{\lambda}_0(1)}{3\mu_0} \cdot S_{0,1}^0 - \frac{\vartheta_1}{3\mu_0} \cdot S_{1,2}^0 \\
&= \frac{\hat{\lambda}_H + \hat{\lambda}_N + 2\mu_S + \vartheta_S}{3\mu_S} \cdot S_{0,2}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{3\mu_S} \cdot S_{0,1}^0 - \frac{\vartheta_M}{3\mu_S} \cdot S_{1,2}^0 \\
&= 52,9491111
\end{aligned}$$

- Για $i = 1$ και $n = 2$:

$$\begin{aligned}
S_{1,3}^0 &= \frac{\hat{\lambda}_1(2) + 2 \cdot \mu_1 + \vartheta_1}{3\mu_1} \cdot S_{1,2}^0 - \frac{\hat{\lambda}_1(1)}{3\mu_1} \cdot S_{1,1}^0 - \frac{\vartheta_0}{3\mu_1} \cdot S_{0,2}^0 \\
&= \frac{\hat{\lambda}_N + 2\mu_M + \vartheta_M}{3\mu_M} \cdot S_{1,2}^0 - \frac{\hat{\lambda}_N}{3\mu_M} \cdot S_{1,1}^0 - \frac{\vartheta_S}{3\mu_M} \cdot S_{0,2}^0 \\
&= -3,69922222
\end{aligned}$$

- Για $i = 0$ και $n = 3$:

$$\begin{aligned}
S_{0,4}^0 &= \frac{\hat{\lambda}_0(3) + 3 \cdot \mu_0 + \vartheta_0}{4\mu_0} \cdot S_{0,3}^0 - \frac{\hat{\lambda}_0(2)}{4\mu_0} \cdot S_{0,2}^0 - \frac{\vartheta_1}{4\mu_0} \cdot S_{1,3}^0 \\
&= \frac{\hat{\lambda}_H + \hat{\lambda}_N + 3\mu_S + \vartheta_S}{4\mu_S} \cdot S_{0,3}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{4\mu_S} \cdot S_{0,2}^0 - \frac{\vartheta_M}{4\mu_S} \cdot S_{1,3}^0 \\
&= 91,13142035
\end{aligned}$$

- Για $i = 1$ και $n = 3$:

$$\begin{aligned} S_{1,4}^0 &= \frac{\hat{\lambda}_1(3) + 3 \cdot \mu_1 + \vartheta_1}{4\mu_1} \cdot S_{1,3}^0 - \frac{\hat{\lambda}_1(2)}{4\mu_1} \cdot S_{1,2}^0 - \frac{\vartheta_0}{4\mu_1} \cdot S_{0,3}^0 \\ &= \frac{\hat{\lambda}_N + 3\mu_M + \vartheta_M}{4\mu_M} \cdot S_{1,3}^0 - \frac{\hat{\lambda}_N}{4\mu_M} \cdot S_{1,2}^0 - \frac{\vartheta_S}{4\mu_M} \cdot S_{0,3}^0 \\ &= -8,023574999 \end{aligned}$$

- Για $i = 0$ και $n = 4$:

$$\begin{aligned} S_{0,5}^0 &= \frac{\hat{\lambda}_0(4) + 4 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,4}^0 - \frac{\hat{\lambda}_0(3)}{5\mu_0} \cdot S_{0,3}^0 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,4}^0 \\ &= \frac{\hat{\lambda}_H + 4\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,4}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{5\mu_S} \cdot S_{0,3}^0 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,4}^0 \\ &= 77,74861 \end{aligned}$$

- Για $i = 1$ και $n = 4$:

$$\begin{aligned} S_{1,5}^0 &= \frac{\hat{\lambda}_1(4) + 4 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,4}^0 - \frac{\hat{\lambda}_1(3)}{5\mu_1} \cdot S_{1,3}^0 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,4}^0 \\ &= \frac{\hat{\lambda}_N + 4\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,4}^0 - \frac{\hat{\lambda}_N}{5\mu_M} \cdot S_{1,3}^0 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,4}^0 \\ &= -13,68407054 \end{aligned}$$

- Για $i = 0$ και $n = 5$:

$$\begin{aligned} S_{0,6}^0 &= \frac{\hat{\lambda}_0(5) + 5 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,5}^0 - \frac{\hat{\lambda}_0(4)}{5\mu_0} \cdot S_{0,4}^0 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,5}^0 \\ &= \frac{\hat{\lambda}_H + 5\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,5}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{5\mu_S} \cdot S_{0,4}^0 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,5}^0 \\ &= 69,29811226 \end{aligned}$$

- Για $i = 1$ και $n = 5$:

$$\begin{aligned} S_{1,6}^0 &= \frac{5 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,5}^0 - \frac{\hat{\lambda}_1(4)}{5\mu_1} \cdot S_{1,4}^0 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,5}^0 \\ &= \frac{5\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,5}^0 - \frac{\hat{\lambda}_N}{5\mu_M} \cdot S_{1,4}^0 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,5}^0 \\ &= -10,64883635 \end{aligned}$$

- Για $i = 0$ και $n = 6$:

$$\begin{aligned} S_{0,7}^0 &= \frac{\hat{\lambda}_0(6) + 5 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,6}^0 - \frac{\hat{\lambda}_0(5)}{5\mu_0} \cdot S_{0,5}^0 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,6}^0 \\ &= \frac{\hat{\lambda}_H + 5\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,6}^0 - \frac{\hat{\lambda}_H}{5\mu_S} \cdot S_{0,5}^0 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,6}^0 \\ &= 64,52764819 \end{aligned}$$

- Για $i = 1$ και $n = 6$:

$$\begin{aligned} S_{1,7}^0 &= \frac{5 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,6}^0 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,6}^0 \\ &= \frac{5\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,6}^0 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,6}^0 \\ &= -13,63373757 \end{aligned}$$

- Για $i = 0$ και $n = 7$:

$$\begin{aligned} S_{0,8}^0 &= \frac{\hat{\lambda}_0(7) + 5 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,7}^0 - \frac{\hat{\lambda}_0(6)}{5\mu_0} \cdot S_{0,6}^0 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,7}^0 \\ &= \frac{\hat{\lambda}_H + 5\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,7}^0 - \frac{\hat{\lambda}_H}{5\mu_S} \cdot S_{0,6}^0 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,7}^0 \\ &= 62,61379736 \end{aligned}$$

- Για $i = 1$ και $n = 7$:

$$\begin{aligned} S_{1,8}^0 &= \frac{5 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,7}^0 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,7}^0 \\ &= \frac{5\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,7}^0 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,7}^0 \\ &= -16,48751825 \end{aligned}$$

Αντίστοιχα, για τον υπολογισμό των συντελεστών $S_{i,n}^1$, χρησιμοποιούμε τις αναδρομικές σχέσεις (1.12) και (1.13), θεωρώντας ότι $S_{0,0}^1 = 0$, $S_{1,0}^1 = 1$ και $S_{0,y}^1 = 0$, $S_{1,y}^1 = 0$ για $y < 0$. Έτσι προκύπτουν:

- Για $i = 0$ και $n = 0$:

$$\begin{aligned} S_{0,1}^1 &= \frac{\hat{\lambda}_0(0) + 0 \cdot \mu_0 + \vartheta_0}{\mu_0} \cdot S_{0,0}^1 - \frac{\hat{\lambda}_0(-1)}{\mu_0} \cdot S_{0,-1}^1 - \frac{\vartheta_1}{\mu_0} \cdot S_{1,0}^1 \\ &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + \vartheta_S}{\mu_S} \cdot S_{0,0}^1 - 0 \cdot S_{0,-1}^1 - \frac{\vartheta_M}{\mu_S} \cdot S_{1,0}^1 = -0,066666667 \end{aligned}$$

- Για $i = 1$ και $n = 0$:

$$\begin{aligned} S_{1,1}^1 &= \frac{\hat{\lambda}_1(0) + 0 \cdot \mu_1 + \vartheta_1}{\mu_1} \cdot S_{1,0}^1 - \frac{\hat{\lambda}_1(-1)}{\mu_1} \cdot S_{1,-1}^1 - \frac{\vartheta_0}{\mu_1} \cdot S_{0,0}^1 \\ &= \frac{\hat{\lambda}_N + 0 \cdot \mu_M + \vartheta_M}{\mu_M} \cdot S_{1,0}^1 - 0 \cdot S_{1,-1}^1 - \frac{\vartheta_S}{\mu_M} \cdot S_{0,0}^1 = 4,1 \end{aligned}$$

- Για $i = 0$ και $n = 1$:

$$\begin{aligned} S_{0,2}^1 &= \frac{\hat{\lambda}_0(1) + 1 \cdot \mu_0 + \vartheta_0}{2\mu_0} \cdot S_{0,1}^1 - \frac{\hat{\lambda}_0(0)}{2\mu_0} \cdot S_{0,0}^1 - \frac{\vartheta_1}{2\mu_0} \cdot S_{1,1}^1 \\ &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + \mu_S + \vartheta_S}{2\mu_S} \cdot S_{0,1}^1 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{2\mu_S} \cdot S_{0,0}^1 - \frac{\vartheta_M}{2\mu_S} \cdot S_{1,1}^1 \\ &= -0,396666667 \end{aligned}$$

- Για $i = 1$ και $n = 1$:

$$\begin{aligned} S_{1,2}^1 &= \frac{\hat{\lambda}_1(1) + 1 \cdot \mu_1 + \vartheta_1}{2\mu_1} \cdot S_{1,1}^1 - \frac{\hat{\lambda}_1(0)}{2\mu_1} \cdot S_{1,0}^1 - \frac{\vartheta_0}{2\mu_1} \cdot S_{0,1}^1 \\ &= \frac{\hat{\lambda}_N + \mu_M + \vartheta_M}{2\mu_M} \cdot S_{1,1}^1 - \frac{\hat{\lambda}_N}{2\mu_M} \cdot S_{1,0}^1 - \frac{\vartheta_S}{2\mu_M} \cdot S_{0,1}^1 \\ &= 8,461666667 \end{aligned}$$

- Για $i = 0$ και $n = 2$:

$$\begin{aligned} S_{0,3}^1 &= \frac{\hat{\lambda}_0(2) + 2 \cdot \mu_0 + \vartheta_0}{3\mu_0} \cdot S_{0,2}^1 - \frac{\hat{\lambda}_0(1)}{3\mu_0} \cdot S_{0,1}^1 - \frac{\vartheta_1}{3\mu_0} \cdot S_{1,2}^1 \\ &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + 2\mu_S + \vartheta_S}{3\mu_S} \cdot S_{0,2}^1 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{3\mu_S} \cdot S_{0,1}^1 - \frac{\vartheta_M}{3\mu_S} \cdot S_{1,2}^1 \\ &= -1,203444445 \end{aligned}$$

- Για $i = 1$ και $n = 2$:

$$\begin{aligned} S_{1,3}^1 &= \frac{\hat{\eta}_1(2) + 2 \cdot \mu_1 + \vartheta_1}{3\mu_1} \cdot S_{1,2}^1 - \frac{\hat{\eta}_1(1)}{3\mu_1} \cdot S_{1,1}^1 - \frac{\vartheta_0}{3\mu_1} \cdot S_{0,2}^1 \\ &= \frac{\hat{\eta}_N + 2\mu_M + \vartheta_M}{3\mu_M} \cdot S_{1,2}^1 - \frac{\hat{\eta}_N}{3\mu_M} \cdot S_{1,1}^1 - \frac{\vartheta_S}{3\mu_M} \cdot S_{0,2}^1 \\ &= 11,76516667 \end{aligned}$$

- Για $i = 0$ και $n = 3$:

$$\begin{aligned} S_{0,4}^1 &= \frac{\hat{\eta}_0(3) + 3 \cdot \mu_0 + \vartheta_0}{4\mu_0} \cdot S_{0,3}^1 - \frac{\hat{\eta}_0(2)}{4\mu_0} \cdot S_{0,2}^1 - \frac{\vartheta_1}{4\mu_0} \cdot S_{1,3}^1 \\ &= \frac{\hat{\eta}_H + \hat{\eta}_N + 3\mu_S + \vartheta_S}{4\mu_S} \cdot S_{0,3}^1 - \frac{\hat{\eta}_H + \hat{\eta}_N}{4\mu_S} \cdot S_{0,2}^1 - \frac{\vartheta_M}{4\mu_S} \cdot S_{1,3}^1 \\ &= 91,13142035 \end{aligned}$$

- Για $i = 1$ και $n = 3$:

$$\begin{aligned} S_{1,4}^1 &= \frac{\hat{\eta}_1(3) + 3 \cdot \mu_1 + \vartheta_1}{4\mu_1} \cdot S_{1,3}^1 - \frac{\hat{\eta}_1(2)}{4\mu_1} \cdot S_{1,2}^1 - \frac{\vartheta_0}{4\mu_1} \cdot S_{0,3}^1 \\ &= \frac{\hat{\eta}_N + 3\mu_M + \vartheta_M}{4\mu_M} \cdot S_{1,3}^1 - \frac{\hat{\eta}_N}{4\mu_M} \cdot S_{1,2}^1 - \frac{\vartheta_S}{4\mu_M} \cdot S_{0,3}^1 \\ &= -8,023574999 \end{aligned}$$

- Για $i = 0$ και $n = 4$:

$$\begin{aligned} S_{0,5}^1 &= \frac{\hat{\eta}_0(4) + 4 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,4}^1 - \frac{\hat{\eta}_0(3)}{5\mu_0} \cdot S_{0,3}^1 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,4}^1 \\ &= \frac{\hat{\eta}_H + 4\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,4}^1 - \frac{\hat{\eta}_H + \hat{\eta}_N}{5\mu_S} \cdot S_{0,3}^1 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,4}^1 \\ &= 77,74861 \end{aligned}$$

- Για $i = 1$ και $n = 4$:

$$\begin{aligned} S_{1,5}^1 &= \frac{\hat{\eta}_1(4) + 4 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,4}^1 - \frac{\hat{\eta}_1(3)}{5\mu_1} \cdot S_{1,3}^1 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,4}^1 \\ &= \frac{\hat{\eta}_N + 4\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,4}^1 - \frac{\hat{\eta}_N}{5\mu_M} \cdot S_{1,3}^1 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,4}^1 \\ &= -13,68407054 \end{aligned}$$

- Για $i = 0$ και $n = 5$:

$$\begin{aligned}
 S_{0,6}^0 &= \frac{\hat{\lambda}_0(5) + 5 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,5}^1 - \frac{\hat{\lambda}_0(4)}{5\mu_0} \cdot S_{0,4}^1 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,5}^1 \\
 &= \frac{\hat{\lambda}_H + 5\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,5}^1 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{5\mu_S} \cdot S_{0,4}^1 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,5}^1 \\
 &= 69,29811226
 \end{aligned}$$

- Για $i = 1$ και $n = 5$:

$$\begin{aligned}
 S_{1,6}^1 &= \frac{5 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,5}^1 - \frac{\hat{\lambda}_1(4)}{5\mu_1} \cdot S_{1,4}^1 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,5}^1 \\
 &= \frac{5\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,5}^1 - \frac{\hat{\lambda}_N}{5\mu_M} \cdot S_{1,4}^1 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,5}^1 \\
 &= -10,64883635
 \end{aligned}$$

- Για $i = 0$ και $n = 6$:

$$\begin{aligned}
 S_{0,7}^1 &= \frac{\hat{\lambda}_0(6) + 5 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,6}^1 - \frac{\hat{\lambda}_0(5)}{5\mu_0} \cdot S_{0,5}^1 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,6}^1 \\
 &= \frac{\hat{\lambda}_H + 5\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,6}^1 - \frac{\hat{\lambda}_H}{5\mu_S} \cdot S_{0,5}^1 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,6}^1 \\
 &= 64,52764819
 \end{aligned}$$

- Για $i = 1$ και $n = 6$:

$$\begin{aligned}
 S_{1,7}^1 &= \frac{5 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,6}^1 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,6}^1 \\
 &= \frac{5\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,6}^1 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,6}^1 \\
 &= -13,63373757
 \end{aligned}$$

- Για $i = 0$ και $n = 7$:

$$\begin{aligned}
 S_{0,8}^1 &= \frac{\hat{\lambda}_0(7) + 5 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,7}^1 - \frac{\hat{\lambda}_0(6)}{5\mu_0} \cdot S_{0,6}^1 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,7}^1 \\
 &= \frac{\hat{\lambda}_H + 5\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,7}^1 - \frac{\hat{\lambda}_H}{5\mu_S} \cdot S_{0,6}^1 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,7}^1 \\
 &= 62,61379736
 \end{aligned}$$

- Για $i = 1$ και $n = 7$:

$$\begin{aligned} S_{1,8}^1 &= \frac{5 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,7}^1 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,7}^1 \\ &= \frac{5\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,7}^1 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,7}^1 \\ &= -16,48751825 \end{aligned}$$

Στους πίνακες που ακολουθούν παρατίθενται τα αριθμητικά αποτελέσματα των συντελεστών $S_{i,n}^0$ και $S_{i,n}^1$. Ο πίνακας 1.1 αφορά στα συγκεντρωτικά αποτελέσματα των συντελεστών που προκύπτουν από τη διαδικασία που ακολουθήσαμε στο Παράδειγμα 1.4, ενώ ο Πίνακας 1.2 αναφέρεται στα αποτελέσματα που προκύπτουν από την υλοποίηση του αλγόριθμου MA-CAC με ουρά μεταπομπής σε πρόγραμμα γλώσσας Python.

Πίνακας 1.1: Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 1.4

$S_{i,n}^0$		$S_{i,n}^1$	
$S_{0,0}^0 = 1$	$S_{1,0}^0 = 0$	$S_{0,0}^1 = 0$	$S_{1,0}^1 = 1$
$S_{0,1}^0 = 6,8$	$S_{1,1}^0 = -0,2$	$S_{0,1}^1 = -0,066666667$	$S_{1,1}^1 = 4,1$
$S_{0,2}^0 = 23,19333333$	$S_{1,2}^0 = -1,19$	$S_{0,2}^1 = -0,396666667$	$S_{1,2}^1 = 8,461666667$
$S_{0,3}^0 = 52,94911111$	$S_{1,3}^0 = -3,699222222$	$S_{0,3}^1 = -1,203444445$	$S_{1,3}^1 = 11,76516667$
$S_{0,4}^0 = 91,13142035$	$S_{1,4}^0 = -8,023574999$	$S_{0,4}^1 = -2,483413889$	$S_{1,4}^1 = 12,48167639$
$S_{0,5}^0 = 77,74861$	$S_{1,5}^0 = -13,68407054$	$S_{0,5}^1 = -2,601516351$	$S_{1,5}^1 = 10,90751897$
$S_{0,6}^0 = 69,29811226$	$S_{1,6}^0 = -10,64883635$	$S_{0,6}^1 = -2,910805676$	$S_{1,6}^1 = 1,244388892$
$S_{0,7}^0 = 64,52764819$	$S_{1,7}^0 = -13,63373757$	$S_{0,7}^1 = -3,252450472$	$S_{1,7}^1 = 1,385708897$
$S_{0,8}^0 = 62,61379736$	$S_{1,8}^0 = -16,48751825$	$S_{0,8}^1 = -3,63097444$	$S_{1,8}^1 = 1,543521094$

Πίνακας 1.2: Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 1.4 (Πρόγραμμα σε Python)

n	$S_{0,n}^0$	$S_{1,n}^0$	$S_{0,n}^1$	$S_{1,n}^1$
0	1	0	0	1
1	6,8	-0,2	-0,066666667	4,1
2	23,19333333	-1,19	-0,396666667	8,461666667
3	52,94911111	-3,699222222	-1,203444444	11,76516667
4	91,13142037	-8,023575	-2,483413889	12,48167639
5	77,74860999	-13,68407054	-2,601516352	10,90751897
6	69,29811222	-10,64883635	-2,910805678	1,244388895
7	64,52764815	-13,63373756	-3,252450475	1,3857089
8	62,61379735	-16,48751824	-3,630974445	1,543521097

Από τη σχέση (1.16) υπολογίζουμε το r :

$$r = -\frac{S_{0,8}^0 \cdot (5\mu_S + \vartheta_S) - S_{0,7}^0 \cdot \hat{\lambda}_H - S_{1,8}^0 \cdot \vartheta_M}{S_{0,8}^1 \cdot (5\mu_S + \vartheta_S) - S_{0,7}^1 \cdot \hat{\lambda}_H - S_{1,8}^1 \cdot \vartheta_M} = 11,23603037$$

Έτσι από τη σχέση (1.17) έχουμε:

$$\frac{P(1, 0)}{P(0, 0)} = 11,23603037$$

Επομένως, από τις σχέσεις (1.19) και (1.17), οι τιμές των πιθανοτήτων $P(0, 0)$ και $P(1, 0)$ είναι αντίστοιχα:

$$P(0, 0) = \left(\sum_{i=0}^1 \sum_{n=0}^8 S_{i,n}^0 + r \cdot \sum_{i=0}^1 \sum_{n=0}^8 S_{i,n}^1 \right)^{-1} = 0,001265736121$$

και

$$P(1, 0) = r \cdot P(0, 0) = 0,014221849$$

από τη σχέση (1.7) υπολογίζουμε τις υπόλοιπες πιθανότητες των καταστάσεων του συστήματος, ως εξής:

$$P(0, 1) = S_{0,1}^0 \cdot P(0, 0) + S_{0,1}^1 \cdot P(1, 0) = 0,007658882351$$

$$P(1, 1) = S_{1,1}^0 \cdot P(0, 0) + S_{1,1}^1 \cdot P(1, 0) = 0,058056433$$

$$P(0, 2) = S_{0,2}^0 \cdot P(0, 0) + S_{0,2}^1 \cdot P(1, 0) = 0,023715306$$

$$P(1, 2) = S_{1,2}^0 \cdot P(0, 0) + S_{1,2}^1 \cdot P(1, 0) = 0,118834319$$

$$P(0, 3) = S_{0,3}^0 \cdot P(0, 0) + S_{0,3}^1 \cdot P(1, 0) = 0,049904397$$

$$P(1, 3) = S_{1,3}^0 \cdot P(0, 0) + S_{1,3}^1 \cdot P(1, 0) = 0,162640184$$

$$P(0, 4) = S_{0,4}^0 \cdot P(0, 0) + S_{0,4}^1 \cdot P(1, 0) = 0,080029593$$

$$P(1, 4) = S_{1,4}^0 \cdot P(0, 0) + S_{1,4}^1 \cdot P(1, 0) = 0,167356788$$

$$P(0, 5) = S_{0,5}^0 \cdot P(0, 0) + S_{0,5}^1 \cdot P(1, 0) = 0,061410851$$

$$P(1, 5) = S_{1,5}^0 \cdot P(0, 0) + S_{1,5}^1 \cdot P(1, 0) = 0,137804665$$

$$P(0, 6) = S_{0,6}^0 \cdot P(0, 0) + S_{0,6}^1 \cdot P(1, 0) = 0,046316085$$

$$P(1, 6) = S_{1,6}^0 \cdot P(0, 0) + S_{1,6}^1 \cdot P(1, 0) = 0,004218894105$$

$$P(0, 7) = S_{0,7}^0 \cdot P(0, 0) + S_{0,7}^1 \cdot P(1, 0) = 0,035419115$$

$$P(1, 7) = S_{1,7}^0 \cdot P(0, 0) + S_{1,7}^1 \cdot P(1, 0) = 0,002450628585$$

$$P(0, 8) = S_{0,8}^0 \cdot P(0, 0) + S_{0,8}^1 \cdot P(1, 0) = 0,027613374$$

$$P(1, 8) = S_{1,8}^0 \cdot P(0, 0) + S_{1,8}^1 \cdot P(1, 0) = 0,001082876533$$

Τα συγκεντρωτικά αποτελέσματα των πιθανοτήτων $P(i, n)$ των καταστάσεων του συστήματος παρουσιάζονται στον πίνακα 1.3, για τη διαδικασία που ακολουθήσαμε στο Παράδειγμα 1.4 ενώ στον πίνακα 1.4, παρατίθενται τα αποτελέσματα του προγράμματος MA-CAC με ουρά μεταπομπής σε Python.

Πίνακας 1.3: Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 1.4

$P(i, n)$	
$P(0, 0) = 0,001265736121$	$P(1, 0) = 0,014221849$
$P(0, 1) = 0,007658882351$	$P(1, 1) = 0,058056433$
$P(0, 2) = 0,023715306$	$P(1, 2) = 0,118834319$
$P(0, 3) = 0,049904397$	$P(1, 3) = 0,162640184$
$P(0, 4) = 0,080029593$	$P(1, 4) = 0,167356788$
$P(0, 5) = 0,061410851$	$P(1, 5) = 0,137804665$
$P(0, 6) = 0,046316085$	$P(1, 6) = 0,004218894105$
$P(0, 7) = 0,035419115$	$P(1, 7) = 0,002450628585$
$P(0, 8) = 0,027613374$	$P(1, 8) = 0,001082876533$

Πίνακας 1.4: Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 1.4 (Πρόγραμμα σε Python)

n	$P(0, n)$	$P(1, n)$
0	0,001265736	0,01422185
1	0,007658882	0,058056436
2	0,023715306	0,118834324
3	0,049904397	0,162640191
4	0,080029592	0,167356794
5	0,06141085	0,137804671
6	0,046316083	0,004218895
7	0,035419114	0,002450629
8	0,027613373	0,001082877

Η πιθανότητα απώλειας των νέων κλήσεων, στη φάση στάσης του οχήματος, υπολογίζεται μέσω της σχέσης (1.20):

$$B_S^N = \sum_{n=4}^8 P(0, n) = 0,250789018$$

Η πιθανότητα απώλειας των νέων κλήσεων, όταν το όχημα κινείται, υπολογίζεται μέσω της σχέσης (1.21):

$$B_S^M = \sum_{n=5}^8 P(1, n) = 0,145557064$$

Επομένως, η συνολική πιθανότητα απώλειας των νέων κλήσεων στο σύστημα είναι (από (1.22)):

$$P_{NCBP} = B_S^N + B_M^N = 0,396346082$$

Όσον αφορά την πιθανότητα απόρριψης των κλήσεων μεταπομπής του συστήματος, μέσω της σχέσης (1.23) έχουμε:

$$P_{HCDP} = P(n = 8/i = 0) = \frac{P(0, 8)}{\sum_{n=0}^8 P(0, n)} = 0,08284012$$

Για τον υπολογισμό του ποσοστού χρήσης των καναλιών, χρησιμοποιούμε τη σχέση (1.24):

$$U = \frac{\sum_{i=0}^1 \sum_{n=0}^5 n \cdot P(i, n) + \sum_{i=0}^1 \sum_{n=6}^8 5 \cdot P(i, n)}{5} = 0,711915255$$

Τέλος, για να προσδιορίσουμε το μέσο χρόνο παραμονής των κλήσεων στην ουρά μεταπομπής, υπολογίζουμε πρώτα τον μέσο αριθμό των κλήσεων στην ουρά (N_Q) και τον πραγματικό ρυθμό άφιξης των κλήσεων στο σύστημα (λ_e), από τις σχέσεις (1.26) και (1.27) αντίστοιχα:

$$N_Q = \sum_{i=0}^1 \sum_{n=6}^8 (n - 5) \cdot P(i, n) = 0,212363271$$

και

$$\lambda_e = \sum_{n=0}^3 (\lambda_N + \lambda_H) \cdot P(0, n) + \sum_{n=4}^7 (\lambda_H) \cdot P(0, n) + \sum_{n=0}^4 (\lambda_N) \cdot P(1, n) = 2,124467685$$

Οπότε, από τον τύπο (1.25), ο μέσος χρόνος παραμονής στην ουρά (σε min) είναι:

$$W_Q = \frac{N_Q}{\lambda_e} = 0,099960671$$

και σε sec:

$$W_Q = 0,099960671 \cdot 60 = 5,997640308$$

Τα αποτελέσματα των πιθανοτήτων απώλειας των νέων κλήσεων (P_{NCBP}), απόρριψης των κλήσεων μεταπομπής (P_{HCDP}) αλλά και το ποσοστό χρήσης των καναλιών (Utilization) και ο χρόνος παραμονής των κλήσεων στην ουρά μεταπομπής (Waiting Time in the Queue), συγκεντρώνονται στον πίνακα 1.5:

Πίνακας 1.5: P_{NCBP} , P_{HCDP} , Ποσοστό Χρήσης Καναλιών και Χρόνος Αναμονής στην Ουρά Μεταπομπής του Παραδείγματος 1.4

P_{NCBP}	0,396346082
P_{HCDP}	0,08284012
<i>Utilization</i>	0,711915255
<i>Waiting Time in the Queue(sec)</i>	5,997640308

ενώ τα αποτελέσματα του προγράμματος MA-CAC με ουρά μεταπομπής σε Python, παρουσιάζονται στον πίνακα 1.6:

Πίνακας 1.6: P_{NCBP} , P_{HCDP} , Ποσοστό Χρήσης Καναλιών και Χρόνος Αναμονής στην Ουρά Μεταπομπής του Παραδείγματος 1.4 (Πρόγραμμα σε Python)

P_{NCBP}	0,396346084
P_{HCDP}	0,082840119
<i>Utilization</i>	0,71191527
<i>Waiting Time in the Queue(sec)</i>	5,997640182

Κεφάλαιο 2

Μοντέλο Απωλειών MA-CAC χωρίς Ουρά Μεταπομπής

2.1 Εισαγωγή

Στο 2ο Κεφάλαιο παρουσιάζεται το μοντέλο αποδοχής κλήσεων MA-CAC χωρίς ουρά μεταπομπής. Πρόκειται για το σύστημα που παρουσιάστηκε στο Κεφάλαιο 1, με τη διαφορά ότι στον αλγόριθμο MA-CAC δεν προστίθεται η ουρά στην οποία εισέρχονταν εκείνες οι κλήσεις μεταπομπής, για τις οποίες δεν υπήρχε διαθέσιμο κανάλι ώστε να εξυπηρετηθούν.

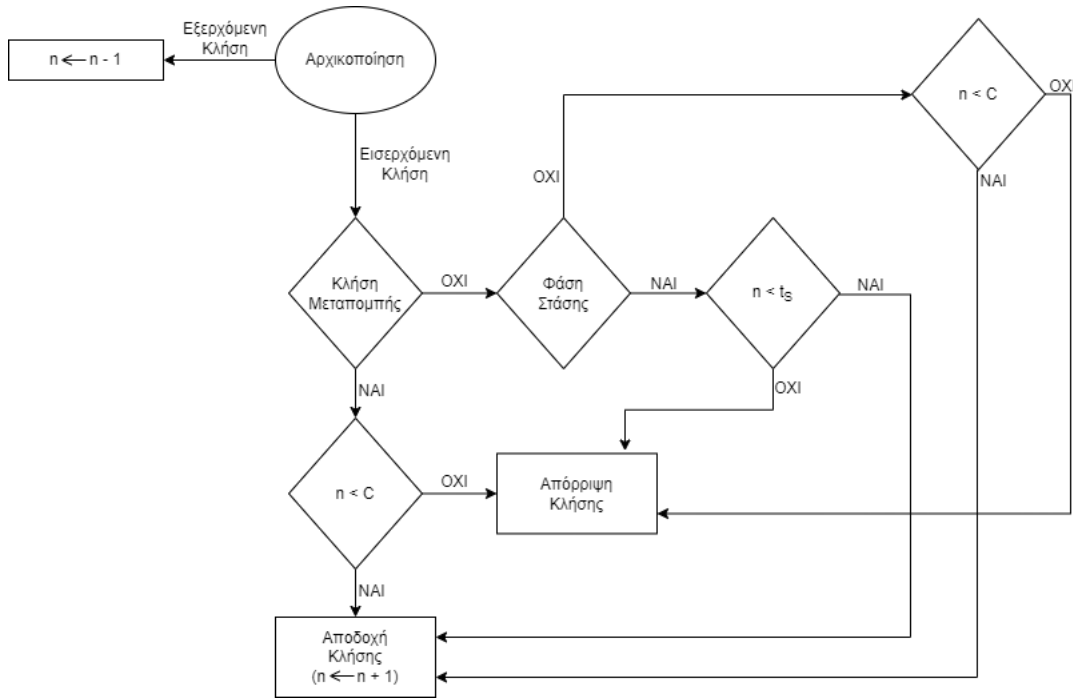
Στην Ενότητα 2.2, περιγράφεται ο αλγόριθμος του μοντέλου MA-CAC χωρίς ουρά μεταπομπής. Στη συνέχεια, στην Ενότητα 2.3, αναλύεται λεπτομερώς το μοντέλο απωλειών και παρουσιάζεται η Μαρκοβιανή αλυσίδα συνεχούς χρόνου του συστήματος. Στην ίδια Ενότητα, παρατίθενται επιπλέον οι εξισώσεις ισορροπίας της Μαρκοβιανής αλυσίδας και επιλύεται το σύστημα των εξισώσεων αυτών, με τη μέθοδο που χρησιμοποιήθηκε στο Κεφάλαιο 1. Έτσι οδηγούμαστε στις σχέσεις που υπολογίζουν τις πιθανότητες απωλειών των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής, καθώς και στον τύπο υπολογισμού του ποσοστού χρήσης των καναλιών του συστήματος. Τέλος, στην Ενότητα 2.4, επιλύεται ένα αριθμητικό παράδειγμα του μοντέλου, χρησιμοποιώντας τις σχέσεις της Ενότητας 2.3.

2.2 Ο Αλγόριθμος MA-CAC χωρίς Ουρά Μεταπομπής

Όπως φαίνεται στο διάγραμμα ροής του Σχήματος 2.1, μόλις μία κλήση εισέλθει στο σύστημα, ο αλγόριθμος MA-CAC χωρίς την ουρά μεταπομπής διακρίνει το είδος της, δηλαδή εξετάζει αν είναι κλήση μεταπομπής ή αν είναι νέα κλήση.

Στην περίπτωση που η κλήση είναι μεταπομπής, ελέγχεται αν ο αριθμός των ενεργών κλήσεων υπερβαίνει το πλήθος των καναλιών του συστήματος. Αν το κάνει η κλήση απορρίπτεται, αλλιώς η κλήση εξυπηρετείται.

Στην άλλη περίπτωση, όπου η κλήση που εισέρχεται στο σύστημα είναι νέα κλήση, ο αλγόριθμος εξετάζει αν το όχημα είναι σε κίνηση ή είναι στατικό. Όταν το όχημα βρίσκεται στη φάση στάσης, ελέγχεται αν ο αριθμός των ενεργών κλήσεων είναι μικρότερος ή ίσος



Σχήμα 2.1: Διάγραμμα Ροής του Αλγορίθμου MA-CAC χωρίς ΗQ.

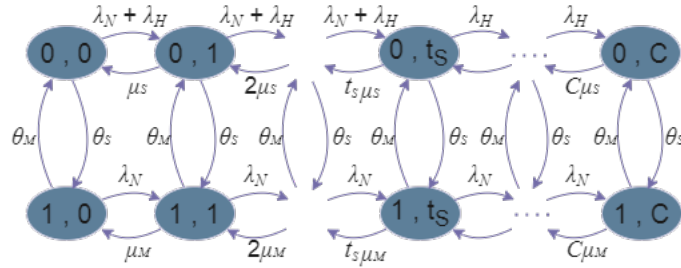
από τον αριθμό των κοινών καναλιών t_s που χρησιμοποιούνται και από τα δύο είδη κλήσεων (κλήσεις μεταπομπής/νέες κλήσεις). Αν είναι τότε η κλήση εξυπηρετείται, διαφορετικά η κλήση απορρίπτεται. Από την άλλη μεριά, όταν ο αριθμός των κλήσεων που ήδη εξυπηρετούνται από το σημείο πρόσβασης του δικτύου WLAN υπερβαίνει το t_s , η κλήση επίσης απορρίπτεται.

2.3 Ανάλυση του μοντέλου

Όπως αναφέρθηκε στην εισαγωγή του Κεφαλαίου, το μοντέλο που αναπτύσσεται εδώ είναι το ίδιο με το μοντέλο που παρουσιάστηκε στο Κεφάλαιο 1 και διαφοροποιείται μόνο ως προς την ύπαρξη της ουράς μεταπομπής. Συνεπώς στο μοντέλο MA-CAC χωρίς την ουρά μεταπομπής λαμβάνεται υπόψη η κατάσταση κίνησης του οχήματος (φάση στάσης/φάση κίνησης). Όπως συμβαίνει και στο σύστημα MA-CAC με ουρά μεταπομπής, στη φάση στάσης μπορούν να εισέλθουν στο σύστημα τόσο κλήσεις που προέρχονται από μεταπομπή όσο και νέες κλήσεις, ενώ κατά τη διάρκεια της φάσης που το όχημα κινείται πραγματοποιούνται μόνο νέες κλήσεις.

Οι διαδικασίες άφιξης λ_H και λ_N , των κλήσεων μεταπομπής και των νέων κλήσεων αντίστοιχα, ακολουθούν κατανομή Poisson. Συνεπώς ο συνολικός ρυθμός άφιξης των κλήσεων, έως ότου κατεληφθούν όλα τα κοινά κανάλια που χρησιμοποιούνται και από τα δύο είδη κλήσεων, είναι $\lambda_H + \lambda_N$. Η μέση διάρκεια των κλήσεων $1/\mu_S$ και $1/\mu_M$ και ο χρόνος παραμονής των κλήσεων στο σύστημα $1/\vartheta_S$ και $1/\vartheta_M$, για τις φάσεις στάσης και κίνησης αντίστοιχα, ακολουθούν εκθετική κατανομή. Επιπλέον, θεωρούμε ότι η κάθε εισερχόμενη στο σύστημα κλήση απαιτεί μόνο μία μονάδα εύρους ζώνης (1 b.u.) για την εξυπηρέτησή της. Τέλος, η τιμή του ποσοστού αναχώρησης μιας κλήσης από το σύστημα είναι $n \cdot \mu_S$ για τη φάση στάσης και $n \cdot \mu_M$ για τη φάση κίνησης αντίστοιχα.

Επομένως, για το ρυθμό μετάβασης από την κατάσταση (i, n) στην κατάσταση (i', n') ,



Σχήμα 2.2: Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για τον αλγόριθμο MA-CAC χωρίς HQ.

έχουμε:

$$\begin{aligned}
 r(0, n; 1, n) &= \vartheta_S, & \text{για } 0 \leq n \leq C \\
 r(1, n; 0, n) &= \vartheta_M, & \text{για } 0 \leq n \leq C \\
 r(0, n; 0, n-1) &= n \cdot \mu_S, & \text{για } 0 < n \leq C \\
 r(1, n; 1, n-1) &= n \cdot \mu_M, & \text{για } 0 < n \leq C \\
 r(0, n; 0, n+1) &= \hat{\lambda}_N + \hat{\lambda}_H, & \text{για } 0 \leq n < t_S \\
 r(0, n; 0, n+1) &= \hat{\lambda}_H, & \text{για } t_S \leq n < C \\
 r(1, n; 1, n+1) &= \hat{\lambda}_N, & \text{για } 0 \leq n < C
 \end{aligned} \tag{2.1}$$

Για τον προσδιορισμό των πιθανοτήτων $P(i, n)$, έχουμε τις ακόλουθες εξισώσεις ισορροπίας:

1) Για $i = 0$ και $n = 0$ έχουμε:

$$((\hat{\lambda}_N + \hat{\lambda}_H) + \vartheta_S) P(i, n) = (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

2) Για $i = 0$ και $0 < n < t_S$ έχουμε:

$$((\hat{\lambda}_N + \hat{\lambda}_H) + n\mu_S + \vartheta_S) P(i, n) = (\hat{\lambda}_N + \hat{\lambda}_H) P(i, n-1) + (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

3) Για $i = 0$ και $n = t_S$ έχουμε:

$$(\hat{\lambda}_H + n\mu_S + \vartheta_S) P(i, n) = (\hat{\lambda}_N + \hat{\lambda}_H) P(i, n-1) + (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

4) Για $i = 0$ και $t_S < n < C$ έχουμε:

$$(\hat{\lambda}_H + n\mu_S + \vartheta_S) P(i, n) = \hat{\lambda}_H P(i, n-1) + (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

5) Για $i = 0$ και $n = C$ έχουμε:

$$(C\mu_S + \vartheta_S) P(i, n) = \hat{\lambda}_H P(i, n-1) + \vartheta_M P(1, n)$$

6) Για $i = 1$ και $n = 0$ έχουμε:

$$(\hat{\lambda}_N + \vartheta_M) P(i, n) = (n+1) \mu_M P(i, n+1) + \vartheta_S P(0, n)$$

7) Για $i = 1$ και $0 < n < C$ έχουμε:

$$(\hat{\lambda}_N + n\mu_M + \vartheta_M)P(i, n) = \hat{\lambda}_N P(i, n-1) + (n+1)\mu_M P(i, n+1) + \vartheta_S P(0, n)$$

8) Για $i = 1$ και $n = C$ έχουμε:

$$(C\mu_M + \vartheta_M)P(i, n) = \hat{\lambda}_N P(i, n-1) + C\mu_M P(i, n+1) + \vartheta_S P(0, n) \quad (2.2)$$

Οι οριακές καταστάσεις $(0, 0)$ και $(1, 0)$ των εξισώσεων (2.2), μπορούν να εκφραστούν μέσω της εξίσωσης:

$$(\hat{\lambda}_i(0) + \vartheta_i)P(i, 0) - \mu_i P(i, 1) - \vartheta_j P(j, 0) = 0 \quad (2.3)$$

Όσον αφορά τις ενδιάμεσες καταστάσεις $(1, 0), (2, 0), \dots, (C-1, 0)$ και $(1, 1), (2, 1), \dots, (C-1, 1)$ των εξισώσεων (2.2) έχουμε:

$$(\hat{\lambda}_i(n) + n\mu_i + \vartheta_i)P(i, n) - \hat{\lambda}_i(n-1)P(i, n-1) - \vartheta_j P(j, n) - (n+1)\mu_i P(i, n+1) = 0 \quad (2.4)$$

Επιπλέον, για την περίπτωση των οριακών καταστάσεων $(0, C)$ και $(1, C)$ των εξισώσεων (2.2), έχουμε:

$$(C\mu_i + \vartheta_i)P(i, C) - \hat{\lambda}_i(C-1)P(i, C-1) - \vartheta_j P(j, C) = 0 \quad (2.5)$$

όπου:

$$i = \begin{cases} 0, & \text{για τη φάση στάσης} \\ 1, & \text{για τη φάση κίνησης} \end{cases} \quad \mu_i = \begin{cases} \mu_S, & \text{όταν } i = 0 \\ \mu_M, & \text{όταν } i = 1 \end{cases}$$

$$\vartheta_i = \begin{cases} \vartheta_S, & \text{όταν } i = 0 \\ \vartheta_M, & \text{όταν } i = 1 \end{cases} \quad \hat{\lambda}_i(n) = \begin{cases} \hat{\lambda}_N + \hat{\lambda}_H, & \text{όταν } i = 0 \text{ και } 0 \leq n < t_S \\ \hat{\lambda}_H, & \text{όταν } i = 0 \text{ και } t_S \leq n < C \\ \hat{\lambda}_N, & \text{όταν } i = 1 \text{ και } 0 \leq n < C \\ 0, & \text{όταν } n < 0 \end{cases}$$

και $j = 1 - i$

Οι πιθανότητες $P(i, n)$ των καταστάσεων του συστήματος, όπως και στο μοντέλο MA-CAC με ουρά μεταπομπής, προκύπτουν από τη σχέση:

$$P(i, n) = S_{i,n}^0 \cdot P(0, 0) + S_{i,n}^1 \cdot P(1, 0) \quad (2.6)$$

Για τον προσδιορισμό των συντελεστών $S_{i,n}^0$ και $S_{i,n}^1$ της σχέσης (2.6), αντικαθιστούμε την εξίσωση (2.6) στη σχέση (2.4) και προκύπτει:

$$\begin{aligned} S_{i,n+1}^0 \cdot P(0, 0) + S_{i,n+1}^1 \cdot P(1, 0) &= \frac{\hat{\lambda}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot (S_{i,n}^0 \cdot P(0, 0) + S_{i,n}^1 \cdot P(1, 0)) \\ &\quad - \frac{\hat{\lambda}_i(n-1)}{(n+1)\mu_i} \cdot ((S_{i,n-1}^0 \cdot P(0, 0) + S_{i,n-1}^1 \cdot P(1, 0)) \\ &\quad - \frac{\vartheta_j}{(n+1)\mu_i} \cdot ((S_{j,n}^0 \cdot P(0, 0) + S_{j,n}^1 \cdot P(1, 0)) \end{aligned} \quad (2.7)$$

Αν θεωρήσουμε στη σχέση (2.7) ότι $P(0,0) = 1$ και $P(1,0) = 0$, μπορούμε να υπολογίσουμε τις τιμές των συντελεστών $S_{i,n}^0$, μέσω της αναδρομικής σχέσης που προκύπτει:

$$S_{i,n+1}^0 = \frac{l_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot S_{i,n}^0 - \frac{\hat{l}_i(n-1)}{(n+1)\mu_i} \cdot S_{i,n-1}^0 - \frac{\vartheta_j}{(n+1)\mu_i} \cdot S_{j,n}^0 \quad (2.8)$$

όπου: $S_{0,0}^0 = 1$, $S_{1,0}^0 = 0$ και $S_{0,y}^0 = 0$, $S_{1,y}^0 = 0$ για $y < 0$

Για τον υπολογισμό των τιμών $S_{i,n}^1$, θεωρούμε στη σχέση (2.7) ότι $P(0,0) = 0$ και $P(1,0) = 1$ και προκύπτει η ακόλουθη αναδρομική σχέση:

$$S_{i,n+1}^1 = \frac{\hat{l}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot S_{i,n}^1 - \frac{\hat{l}_i(n-1)}{(n+1)\mu_i} \cdot S_{i,n-1}^1 - \frac{\vartheta_j}{(n+1)\mu_i} \cdot S_{j,n}^1 \quad (2.9)$$

όπου: $S_{0,0}^1 = 0$, $S_{1,0}^1 = 1$ και $S_{0,y}^1 = 0$, $S_{1,y}^1 = 0$ για $y < 0$

Για τον υπολογισμό των πιθανοτήτων $P(0,0)$ και $P(1,0)$ ακολουθούμε την παρακάτω διαδικασία:

Θεωρούμε στην εξίσωση (2.5) την περίπτωση όπου $i = 0$. Έτσι προκύπτει η σχέση:

$$(C\mu_S + \vartheta_S)P(0,C) - \hat{l}_0(C-1)P(0,C-1) - \vartheta_M P(1,C) = 0 \quad (2.10)$$

(Σημειώνουμε ότι και για την περίπτωση $i = 1$, προκύπτουν τα ίδια αποτελέσματα)

Η εξίσωση (2.10) σε συνδυασμό με την εξίσωση (2.6) γίνεται:

$$\begin{aligned} & (C\mu_S + \vartheta_S) \cdot (S_{0,C}^0 \cdot P(0,0) + S_{0,C}^1 \cdot P(1,0)) \\ & - \hat{l}_0(C-1) \cdot (S_{0,C-1}^0 \cdot P(0,0) + S_{0,C-1}^1 \cdot P(1,0)) \\ & - \vartheta_M \cdot (S_{1,C}^0 \cdot P(0,0) + S_{1,C}^1 \cdot P(1,0)) \\ & = 0 \Rightarrow \\ & P(0,0) \cdot (S_{0,C}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^0 \cdot \hat{l}_0 - S_{1,C}^0 \cdot \vartheta_M) \\ & + P(1,0) \cdot (S_{0,C}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^1 \cdot \hat{l}_0 - S_{1,C}^1 \cdot \vartheta_M) \end{aligned}$$

$$= 0 \Rightarrow$$

$$\frac{P(1, 0)}{P(0, 0)} = - \frac{S_{0,C}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^0 \cdot \lambda_0 - S_{1,C}^0 \cdot \vartheta_M}{S_{0,C}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^1 \cdot \hat{\lambda}_0 - S_{1,C}^1 \cdot \vartheta_M} \quad (2.11)$$

Στην εξίσωση (2.11) θέτουμε:

$$r = - \frac{S_{0,C}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^0 \cdot \hat{\lambda}_0 - S_{1,C}^0 \cdot \vartheta_M}{S_{0,C}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^1 \cdot \hat{\lambda}_0 - S_{1,C}^1 \cdot \vartheta_M} \quad (2.12)$$

και προκύπτει η σχέση:

$$\frac{P(1, 0)}{P(0, 0)} = r \Rightarrow P(1, 0) = r \cdot P(0, 0) \quad (2.13)$$

Επιπλέον γνωρίζουμε ότι:

$$\sum_{i=0}^1 \sum_{n=0}^C P(i, n) = 1 \quad (2.14)$$

Επομένως, λόγω της σχέσης (2.6), η εξίσωση (2.14) γίνεται:

$$\begin{aligned} P(0, 0) \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^0 + P(1, 0) \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^1 &= 1 \\ \Rightarrow \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^0 + \frac{P(1, 0) \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^1}{P(0, 0)} &= \frac{1}{P(0, 0)} \\ \Rightarrow P(0, 0) &= \left(\sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^0 + r \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^1 \right)^{-1} \end{aligned} \quad (2.15)$$

Η πιθανότητα απώλειας των νέων κλήσεων στη φάση στάσης του οχήματος, καθορίζεται από τη σχέση:

$$B_S^N = \sum_{n=S}^C P(0, n) \quad (2.16)$$

Η πιθανότητα απώλειας των νέων κλήσεων στη φάση κίνησης του οχήματος, καθορίζεται από τη σχέση:

$$B_M^N = \sum_{n=C}^C P(1, n) = P(1, C) \quad (2.17)$$

Συνεπώς, η συνολική πιθανότητα απώλειας των νέων κλήσεων στο σύστημά μας υπολογίζεται από τον τύπο :

$$P_{NCBP} = B_S^N + B_M^N \tag{2.18}$$

Για την πιθανότητα απόρριψης των κλήσεων μεταπομπής του μοντέλου MA-CAC χωρίς την ουρά μεταπομπής, δοθέντος του γεγονότος ότι οι κλήσεις μεταπομπής πραγματοποιούνται μόνο όταν το όχημα είναι στατικό, έχουμε :

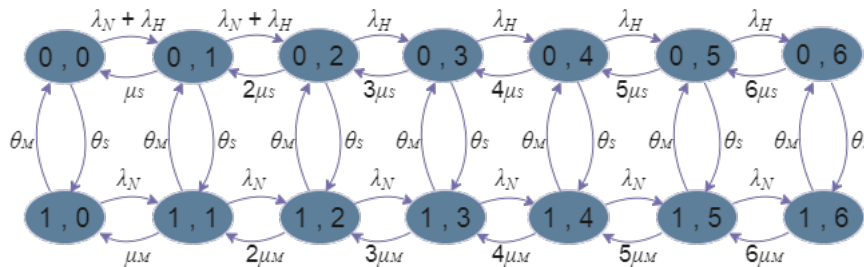
$$P_{HCDP} = P(n = C/i = 0) = \frac{P(0, C)}{\sum_{n=0}^C P(0, n)} \tag{2.19}$$

Τέλος, το ποσοστό χρήσης των καναλιών υπολογίζεται από τη σχέση :

$$U = \frac{\sum_{i=0}^1 \sum_{n=0}^C n \cdot P(i, n)}{C} \tag{2.20}$$

2.4 Παράδειγμα

Στην Ενότητα 2.4 παρουσιάζεται ένα αριθμητικό παράδειγμα του μοντέλου MA-CAC χωρίς ουρά μεταπομπής. Η χωρητικότητα του συστήματος είναι $C = 6$ και ο αριθμός καναλιών που μπορούν να χρησιμοποιηθούν τόσο από νέες κλήσεις όσο και από κλήσεις μεταπομπής είναι $t_s = 2$. Οι τιμές για τους ρυθμούς άφιξης των κλήσεων μεταπομπής και των νέων κλήσεων στο σύστημα, οι οποίες ακολουθούν κατανομή Poisson είναι $\lambda_H = 1$ και $\lambda_N = 2$ αντίστοιχα, ενώ η μέση διάρκεια των κλήσεων στο σύστημα ακολουθεί εκθετική κατανομή με ρυθμό $\mu_s = 0,25$ για τη φάση στάσης και $\mu_M = 0,2$ για τη φάση κίνησης αντίστοιχα. Τέλος, σε ότι αφορά τον ρυθμό του μέσου χρόνου παραμονής των κλήσεων σε καθεμία από τις φάσεις της κινητικότητας του οχήματος, οι τιμές είναι κατ'αντιστοιχία $\theta_s = 0,025$ για τη μετάβαση από τη φάση στάσης στη φάση κίνησης και $\theta_M = 0,1$ για τη μετάβαση από τη φάση κίνησης στη φάση στάσης.



Σχήμα 2.3: Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για τον αλγόριθμο MA-CAC χωρίς HQ του Παραδείγματος 2.4.

Από τη χρήση των σχέσεων (2.3), (2.4) και (2.5), προκύπτουν οι εξισώσεις ισορροπίας των καταστάσεων του συστήματος που ακολουθούν :

- Για την κατάσταση (0, 0):

$$\mu_S \cdot P(0, 1) + \vartheta_M \cdot P(1, 0) = (\lambda_H + \lambda_N + \vartheta_S) \cdot P(0, 0)$$

$$\Rightarrow 0,25 \cdot P(0, 1) + 0,1 \cdot P(1, 0) = 3,025 \cdot P(0, 0)$$

$$\Rightarrow 0,25 \cdot P(0, 1) + 0,1 \cdot P(1, 0) - 3,025 \cdot P(0, 0) = 0 \quad (1)$$

- Για την κατάσταση (1, 0):

$$\mu_M \cdot P(1, 1) + \vartheta_S \cdot P(0, 0) = (\lambda_N + \vartheta_M) \cdot P(1, 0)$$

$$\Rightarrow 0,2 \cdot P(1, 1) + 0,025 \cdot P(0, 0) = 2,1 \cdot P(1, 0)$$

$$\Rightarrow 0,2 \cdot P(1, 1) + 0,025 \cdot P(0, 0) - 2,1 \cdot P(1, 0) = 0 \quad (2)$$

- Για την κατάσταση (0, 1):

$$(\lambda_H + \lambda_N) \cdot P(0, 0) + 2\mu_S \cdot P(0, 2) + \vartheta_M \cdot P(1, 1) = (\lambda_H + \lambda_N + \mu_S + \vartheta_S) \cdot P(0, 1)$$

$$\Rightarrow 3 \cdot P(0, 0) + 0,5 \cdot P(0, 2) + 0,1 \cdot P(1, 1) = 3,275 \cdot P(0, 1)$$

$$\Rightarrow 3 \cdot P(0, 0) + 0,5 \cdot P(0, 2) + 0,1 \cdot P(1, 1) - 3,275 \cdot P(0, 1) = 0 \quad (3)$$

- Για την κατάσταση (1, 1):

$$\lambda_N \cdot P(1, 0) + 2\mu_M \cdot P(1, 2) + \vartheta_S \cdot P(0, 1) = (\lambda_N + \mu_M + \vartheta_M) \cdot P(1, 1)$$

$$\Rightarrow 2 \cdot P(1, 0) + 0,4 \cdot P(1, 2) + 0,025 \cdot P(0, 1) = 2,3 \cdot P(1, 1)$$

$$\Rightarrow 2 \cdot P(1, 0) + 0,4 \cdot P(1, 2) + 0,025 \cdot P(0, 1) - 2,3 \cdot P(1, 1) = 0 \quad (4)$$

- Για την κατάσταση (0, 2):

$$(\lambda_H + \lambda_N) \cdot P(0, 1) + 3\mu_S \cdot P(0, 3) + \vartheta_M \cdot P(1, 2) = (\lambda_H + 2\mu_S + \vartheta_S) \cdot P(0, 2)$$

$$\Rightarrow 3 \cdot P(0, 1) + 0,75 \cdot P(0, 3) + 0,1 \cdot P(1, 2) = 1,525 \cdot P(0, 2)$$

$$\Rightarrow 3 \cdot P(0, 1) + 0,75 \cdot P(0, 3) + 0,1 \cdot P(1, 2) - 1,525 \cdot P(0, 2) = 0 \quad (5)$$

- Για την κατάσταση (1, 2):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 1) + 3\mu_M \cdot P(1, 3) + \vartheta_S \cdot P(0, 2) &= (\hat{\lambda}_N + 2\mu_M + \vartheta_M) \cdot P(1, 2) \\ \Rightarrow 2 \cdot P(1, 1) + 0,6 \cdot P(1, 3) + 0,025 \cdot P(0, 2) &= 2,5 \cdot P(1, 2) \\ \Rightarrow 2 \cdot P(1, 1) + 0,6 \cdot P(1, 3) + 0,025 \cdot P(0, 2) - 2,5 \cdot P(1, 2) &= 0 \end{aligned} \quad (6)$$

- Για την κατάσταση (0, 3):

$$\begin{aligned} \hat{\lambda}_H \cdot P(0, 2) + 4\mu_S \cdot P(0, 4) + \vartheta_M \cdot P(1, 3) &= (\hat{\lambda}_H + 3\mu_S + \vartheta_S) \cdot P(0, 3) \\ \Rightarrow 1 \cdot P(0, 2) + 1 \cdot P(0, 4) + 0,1 \cdot P(1, 3) &= 1,775 \cdot P(0, 3) \\ \Rightarrow 1 \cdot P(0, 2) + 1 \cdot P(0, 4) + 0,1 \cdot P(1, 3) - 1,775 \cdot P(0, 3) &= 0 \end{aligned} \quad (7)$$

- Για την κατάσταση (1, 3):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 2) + 4\mu_M \cdot P(1, 4) + \vartheta_S \cdot P(0, 3) &= (\hat{\lambda}_N + 3\mu_M + \vartheta_M) \cdot P(1, 3) \\ \Rightarrow 2 \cdot P(1, 2) + 0,8 \cdot P(1, 4) + 0,025 \cdot P(0, 3) &= 2,7 \cdot P(1, 3) \\ \Rightarrow 2 \cdot P(1, 2) + 0,8 \cdot P(1, 4) + 0,025 \cdot P(0, 3) - 2,7 \cdot P(1, 3) &= 0 \end{aligned} \quad (8)$$

- Για την κατάσταση (0, 4):

$$\begin{aligned} \hat{\lambda}_H \cdot P(0, 3) + 5\mu_S \cdot P(0, 5) + \vartheta_M \cdot P(1, 4) &= (\hat{\lambda}_H + 4\mu_S + \vartheta_S) \cdot P(0, 4) \\ \Rightarrow 1 \cdot P(0, 3) + 1,25 \cdot P(0, 5) + 0,1 \cdot P(1, 4) &= 2,025 \cdot P(0, 4) \\ \Rightarrow 1 \cdot P(0, 3) + 1,25 \cdot P(0, 5) + 0,1 \cdot P(1, 4) - 2,025 \cdot P(0, 4) &= 0 \end{aligned} \quad (9)$$

- Για την κατάσταση (1, 4):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 3) + 5\mu_M \cdot P(1, 5) + \vartheta_S \cdot P(0, 4) &= (\hat{\lambda}_N + 4\mu_M + \vartheta_M) \cdot P(1, 4) \\ \Rightarrow 2 \cdot P(1, 3) + 1 \cdot P(1, 5) + 0,025 \cdot P(0, 4) &= 2,9 \cdot P(1, 4) \\ \Rightarrow 2 \cdot P(1, 3) + 1 \cdot P(1, 5) + 0,025 \cdot P(0, 4) - 2,9 \cdot P(1, 4) &= 0 \end{aligned} \quad (10)$$

- Για την κατάσταση (0, 5):

$$\begin{aligned} \hat{\lambda}_H \cdot P(0, 4) + 6\mu_S \cdot P(0, 6) + \vartheta_M \cdot P(1, 5) &= (\hat{\lambda}_H + 5\mu_S + \vartheta_S) \cdot P(0, 5) \\ \Rightarrow 1 \cdot P(0, 4) + 1,5 \cdot P(0, 6) + 0,1 \cdot P(1, 5) &= 2,275 \cdot P(0, 5) \\ \Rightarrow 1 \cdot P(0, 4) + 1,5 \cdot P(0, 6) + 0,1 \cdot P(1, 5) - 2,275 \cdot P(0, 5) &= 0 \end{aligned} \quad (11)$$

- Για την κατάσταση (1, 5):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 4) + 6\mu_M \cdot P(1, 6) + j_S \cdot P(0, 5) &= (\hat{\lambda}_N + 5\mu_M + \vartheta_M) \cdot P(1, 5) \\ \Rightarrow 2 \cdot P(1, 4) + 1,2 \cdot P(1, 6) + 0,025 \cdot P(0, 5) &= 3,1 \cdot P(1, 5) \\ \Rightarrow 2 \cdot P(1, 4) + 1,2 \cdot P(1, 6) + 0,025 \cdot P(0, 5) - 3,1 \cdot P(1, 5) &= 0 \end{aligned} \quad (12)$$

- Για την κατάσταση (0, 6):

$$\begin{aligned} \hat{\lambda}_H \cdot P(0, 5) + \vartheta_M \cdot P(1, 6) &= (6\mu_S + \vartheta_S) \cdot P(0, 6) \\ \Rightarrow 1 \cdot P(0, 5) + 0,1 \cdot P(1, 6) &= 1,525 \cdot P(0, 6) \\ \Rightarrow 1 \cdot P(0, 5) + 0,1 \cdot P(1, 6) - 1,525 \cdot P(0, 6) &= 0 \end{aligned} \quad (13)$$

- Για την κατάσταση (1, 6):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 5) + \vartheta_S \cdot P(0, 6) &= (6\mu_M + \vartheta_M) \cdot P(1, 6) \\ \Rightarrow 2 \cdot P(1, 5) + 0,025 \cdot P(0, 6) &= 1,3 \cdot P(1, 6) \\ \Rightarrow 2 \cdot P(1, 5) + 0,025 \cdot P(0, 6) - 1,3 \cdot P(1, 6) &= 0 \end{aligned} \quad (14)$$

Για τον υπολογισμό των συντελεστών $S_{i,n}^0$, χρησιμοποιούμε την σχέση (2.8) με αρχικές τιμές $S_{0,0}^0 = 1$, $S_{1,0}^0 = 0$ και $S_{0,y}^0 = 0$, $S_{1,y}^0 = 0$ για $y < 0$. Έτσι προκύπτουν τα εξής αποτελέσματα:

- Για $i = 0$ και $n = 0$:

$$S_{0,1}^0 = \frac{\hat{\lambda}_0(0) + 0 \cdot \mu_0 + \vartheta_0}{\mu_0} \cdot S_{0,0}^0 - \frac{\hat{\lambda}_0(-1)}{\mu_0} \cdot S_{0,-1}^0 - \frac{\vartheta_1}{\mu_0} \cdot S_{1,0}^0$$

$$= \frac{\hat{\lambda}_H + \hat{\lambda}_N + \vartheta_S}{\mu_S} \cdot S_{0,0}^0 = 12,1$$

- Για $i = 1$ και $n = 0$:

$$\begin{aligned} S_{1,1}^0 &= \frac{\hat{\lambda}_1(0) + 0 \cdot \mu_1 + \vartheta_1}{m_1} \cdot S_{1,0}^0 - \frac{\hat{\lambda}_1(-1)}{\mu_1} \cdot S_{1,-1}^0 - \frac{\vartheta_0}{\mu_1} \cdot S_{0,0}^0 \\ &= -\frac{\vartheta_S}{\mu_M} \cdot S_{0,0}^0 = -0,125 \end{aligned}$$

- Για $i = 0$ και $n = 1$:

$$\begin{aligned} S_{0,2}^0 &= \frac{\hat{\lambda}_0(1) + 1 \cdot \mu_0 + \vartheta_0}{2\mu_0} \cdot S_{0,1}^0 - \frac{\hat{\lambda}_0(0)}{2\mu_0} \cdot S_{0,0}^0 - \frac{\vartheta_1}{2\mu_0} \cdot S_{1,1}^0 \\ &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + \mu_S + \vartheta_S}{2\mu_S} \cdot S_{0,1}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{2\mu_S} \cdot S_{0,0}^0 - \frac{\vartheta_M}{2\mu_S} \cdot S_{1,1}^0 \\ &= 73,28 \end{aligned}$$

- Για $i = 1$ και $n = 1$:

$$\begin{aligned} S_{1,2}^0 &= \frac{\hat{\lambda}_1(1) + 1 \cdot \mu_1 + \vartheta_1}{2\mu_1} \cdot S_{1,1}^0 - \frac{\hat{\lambda}_1(0)}{2\mu_1} \cdot S_{1,0}^0 - \frac{\vartheta_0}{2\mu_1} \cdot S_{0,1}^0 \\ &= \frac{\hat{\lambda}_N + \mu_M + \vartheta_M}{2\mu_M} \cdot S_{1,1}^0 - \frac{\hat{\lambda}_N}{2\mu_M} \cdot S_{1,0}^0 - \frac{\vartheta_S}{2\mu_M} \cdot S_{0,1}^0 \\ &= -1,475 \end{aligned}$$

- Για $i = 0$ και $n = 2$:

$$\begin{aligned} S_{0,3}^0 &= \frac{\hat{\lambda}_0(2) + 2 \cdot \mu_0 + \vartheta_0}{3\mu_0} \cdot S_{0,2}^0 - \frac{\hat{\lambda}_0(1)}{3\mu_0} \cdot S_{0,1}^0 - \frac{\vartheta_1}{3\mu_0} \cdot S_{1,2}^0 \\ &= \frac{\hat{\lambda}_H + 2\mu_S + \vartheta_S}{3\mu_S} \cdot S_{0,2}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{3\mu_S} \cdot S_{0,1}^0 - \frac{\vartheta_M}{3\mu_S} \cdot S_{1,2}^0 \\ &= 100,7993334 \end{aligned}$$

- Για $i = 1$ και $n = 2$:

$$S_{1,3}^0 = \frac{\hat{\lambda}_1(2) + 2 \cdot \mu_1 + \vartheta_1}{3\mu_1} \cdot S_{1,2}^0 - \frac{\hat{\lambda}_1(1)}{3\mu_1} \cdot S_{1,1}^0 - \frac{\vartheta_0}{3\mu_1} \cdot S_{0,2}^0$$

$$\begin{aligned}
&= \frac{\hat{\lambda}_N + 2\mu_M + \vartheta_M}{3\mu_M} \cdot S_{1,2}^0 - \frac{\hat{\lambda}_N}{3\mu_M} \cdot S_{1,1}^0 - \frac{\vartheta_S}{3\mu_M} \cdot S_{0,2}^0 \\
&= -8,7825
\end{aligned}$$

- Για $i = 0$ και $n = 3$:

$$\begin{aligned}
S_{0,4}^0 &= \frac{\hat{\lambda}_0(3) + 3 \cdot \mu_0 + \vartheta_0}{4\mu_0} \cdot S_{0,3}^0 - \frac{\hat{\lambda}_0(2)}{4\mu_0} \cdot S_{0,2}^0 - \frac{\vartheta_1}{4\mu_0} \cdot S_{1,3}^0 \\
&= \frac{\hat{\lambda}_H + 3\mu_S + \vartheta_S}{4\mu_S} \cdot S_{0,3}^0 - \frac{\hat{\lambda}_H}{4\mu_S} \cdot S_{0,2}^0 - \frac{\vartheta_M}{4\mu_S} \cdot S_{1,3}^0 \\
&= 106,5170668
\end{aligned}$$

- Για $i = 1$ και $n = 3$:

$$\begin{aligned}
S_{1,4}^0 &= \frac{\hat{\lambda}_1(3) + 3 \cdot \mu_1 + \vartheta_1}{4\mu_1} \cdot S_{1,3}^0 - \frac{\hat{\lambda}_1(2)}{4\mu_1} \cdot S_{1,2}^0 - \frac{\vartheta_0}{4\mu_1} \cdot S_{0,3}^0 \\
&= \frac{\hat{\lambda}_N + 3\mu_M + \vartheta_M}{4\mu_M} \cdot S_{1,3}^0 - \frac{\hat{\lambda}_N}{4\mu_M} \cdot S_{1,2}^0 - \frac{\vartheta_S}{4\mu_M} \cdot S_{0,3}^0 \\
&= -29,10341667
\end{aligned}$$

- Για $i = 0$ και $n = 4$:

$$\begin{aligned}
S_{0,5}^0 &= \frac{\hat{\lambda}_0(4) + 4 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,4}^0 - \frac{\hat{\lambda}_0(3)}{5\mu_0} \cdot S_{0,3}^0 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,4}^0 \\
&= \frac{\hat{\lambda}_H + 4\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,4}^0 - \frac{\hat{\lambda}_H}{5\mu_S} \cdot S_{0,3}^0 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,4}^0 \\
&= 94,24645481
\end{aligned}$$

- Για $i = 1$ και $n = 4$:

$$\begin{aligned}
S_{1,5}^0 &= \frac{\hat{\lambda}_1(4) + 4 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,4}^0 - \frac{\hat{\lambda}_1(3)}{5\mu_1} \cdot S_{1,3}^0 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,4}^0 \\
&= \frac{\hat{\lambda}_N + 4\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,4}^0 - \frac{\hat{\lambda}_N}{5\mu_M} \cdot S_{1,3}^0 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,4}^0 \\
&= -69,49783501
\end{aligned}$$

- Για $i = 0$ και $n = 5$:

$$\begin{aligned} S_{0,6}^0 &= \frac{\hat{\eta}_0(5) + 5 \cdot \mu_0 + \vartheta_0}{6\mu_0} \cdot S_{0,5}^0 - \frac{\hat{\eta}_0(4)}{6\mu_0} \cdot S_{0,4}^0 - \frac{\vartheta_1}{6\mu_0} \cdot S_{1,5}^0 \\ &= \frac{\hat{\eta}_H + 5\mu_S + \vartheta_S}{6\mu_S} \cdot S_{0,5}^0 - \frac{\hat{\eta}_H}{6\mu_S} \cdot S_{0,4}^0 - \frac{\vartheta_M}{6\mu_S} \cdot S_{1,5}^0 \\ &= 76,56226763 \end{aligned}$$

- Για $i = 1$ και $n = 5$:

$$\begin{aligned} S_{1,6}^0 &= \frac{\hat{\eta}_1(5) + 5 \cdot \mu_1 + \vartheta_1}{6\mu_1} \cdot S_{1,5}^0 - \frac{\hat{\eta}_1(4)}{6\mu_1} \cdot S_{1,4}^0 - \frac{\vartheta_0}{6\mu_1} \cdot S_{0,5}^0 \\ &= \frac{\hat{\eta}_N + 5\mu_M + \vartheta_M}{6\mu_M} \cdot S_{1,5}^0 - \frac{\hat{\eta}_N}{6\mu_M} \cdot S_{1,4}^0 - \frac{\vartheta_S}{6\mu_M} \cdot S_{0,5}^0 \\ &= -132,9938472 \end{aligned}$$

Ο προσδιορισμός των συντελεστών $S_{i,n}^1$ γίνεται μέσω της σχέσης (2.9), θεωρώντας ότι $S_{0,0}^1 = 0$, $S_{1,0}^1 = 1$ και $S_{0,y}^1 = 0$, $S_{1,y}^1 = 0$ για $y < 0$. Έτσι έχουμε :

- Για $i = 0$ και $n = 0$:

$$\begin{aligned} S_{0,1}^1 &= \frac{\hat{\eta}_0(0) + 0 \cdot \mu_0 + \vartheta_0}{\mu_0} \cdot S_{0,0}^1 - \frac{\hat{\eta}_0(-1)}{\mu_0} \cdot S_{0,-1}^1 - \frac{\vartheta_1}{\mu_0} \cdot S_{1,0}^1 \\ &= \frac{\hat{\eta}_H + \hat{\eta}_N + \vartheta_S}{\mu_S} \cdot S_{0,0}^1 - 0 \cdot S_{0,-1}^1 - \frac{\vartheta_M}{\mu_S} \cdot S_{1,0}^1 = -0,4 \end{aligned}$$

- Για $i = 1$ και $n = 0$:

$$\begin{aligned} S_{1,1}^1 &= \frac{\hat{\eta}_1(0) + 0 \cdot \mu_1 + \vartheta_1}{\mu_1} \cdot S_{1,0}^1 - \frac{\hat{\eta}_1(-1)}{\mu_1} \cdot S_{1,-1}^1 - \frac{\vartheta_0}{\mu_1} \cdot S_{0,0}^1 \\ &= \frac{\hat{\eta}_N + 0 \cdot \mu_M + \vartheta_M}{\mu_M} \cdot S_{1,0}^1 - 0 \cdot S_{1,-1}^1 - \frac{\vartheta_S}{\mu_M} \cdot S_{0,0}^1 = 10,5 \end{aligned}$$

- Για $i = 0$ και $n = 1$:

$$\begin{aligned} S_{0,2}^1 &= \frac{\hat{\eta}_0(1) + 1 \cdot \mu_0 + \vartheta_0}{2\mu_0} \cdot S_{0,1}^1 - \frac{\hat{\eta}_0(0)}{2\mu_0} \cdot S_{0,0}^1 - \frac{\vartheta_1}{2\mu_0} \cdot S_{1,1}^1 \\ &= \frac{\hat{\eta}_H + \hat{\eta}_N + \mu_S + \vartheta_S}{2\mu_S} \cdot S_{0,1}^1 - \frac{\hat{\eta}_H + \hat{\eta}_N}{2\mu_S} \cdot S_{0,0}^1 - \frac{\vartheta_M}{2\mu_S} \cdot S_{1,1}^1 \end{aligned}$$

$$= -4,72$$

- Για $i = 1$ και $n = 1$:

$$\begin{aligned} S_{1,2}^1 &= \frac{\hat{\lambda}_1(1) + 1 \cdot \mu_1 + \vartheta_1}{2\mu_1} \cdot S_{1,1}^1 - \frac{\hat{\lambda}_1(0)}{2\mu_1} \cdot S_{1,0}^1 - \frac{\vartheta_0}{2\mu_1} \cdot S_{0,1}^1 \\ &= \frac{\hat{\lambda}_N + \mu_M + \vartheta_M}{2\mu_M} \cdot S_{1,1}^1 - \frac{\hat{\lambda}_N}{2\mu_M} \cdot S_{1,0}^1 - \frac{\vartheta_S}{2\mu_M} \cdot S_{0,1}^1 \\ &= 55,4 \end{aligned}$$

- Για $i = 0$ και $n = 2$:

$$\begin{aligned} S_{0,3}^1 &= \frac{\hat{\lambda}_0(2) + 2 \cdot \mu_0 + \vartheta_0}{3\mu_0} \cdot S_{0,2}^1 - \frac{\hat{\lambda}_0(1)}{3\mu_0} \cdot S_{0,1}^1 - \frac{\vartheta_1}{3\mu_0} \cdot S_{1,2}^1 \\ &= \frac{\hat{\lambda}_H + 2\mu_S + \vartheta_S}{3\mu_S} \cdot S_{0,2}^1 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{3\mu_S} \cdot S_{0,1}^1 - \frac{\vartheta_M}{3\mu_S} \cdot S_{1,2}^1 \\ &= -15,384 \end{aligned}$$

- Για $i = 1$ και $n = 2$:

$$\begin{aligned} S_{1,3}^1 &= \frac{\hat{\lambda}_1(2) + 2 \cdot \mu_1 + \vartheta_1}{3\mu_1} \cdot S_{1,2}^1 - \frac{\hat{\lambda}_1(1)}{3\mu_1} \cdot S_{1,1}^1 - \frac{\vartheta_0}{3\mu_1} \cdot S_{0,2}^1 \\ &= \frac{\hat{\lambda}_N + 2\mu_M + \vartheta_M}{3\mu_M} \cdot S_{1,2}^1 - \frac{\hat{\lambda}_N}{3\mu_M} \cdot S_{1,1}^1 - \frac{\vartheta_S}{3\mu_M} \cdot S_{0,2}^1 \\ &= 196,03 \end{aligned}$$

- Για $i = 0$ και $n = 3$:

$$\begin{aligned} S_{0,4}^1 &= \frac{\hat{\lambda}_0(3) + 3 \cdot \mu_0 + \vartheta_0}{4\mu_0} \cdot S_{0,3}^1 - \frac{\lambda_0(2)}{4\mu_0} \cdot S_{0,2}^1 - \frac{\vartheta_1}{4\mu_0} \cdot S_{1,3}^1 \\ &= \frac{\hat{\lambda}_H + 3\mu_S + \vartheta_S}{4\mu_S} \cdot S_{0,3}^1 - \frac{\hat{\lambda}_H}{4\mu_S} \cdot S_{0,2}^1 - \frac{\vartheta_M}{4\mu_S} \cdot S_{1,3}^1 \\ &= -42,1896 \end{aligned}$$

- Για $i = 1$ και $n = 3$:

$$\begin{aligned}
S_{1,4}^1 &= \frac{\hat{\eta}_1(3) + 3 \cdot \mu_1 + \vartheta_1}{4\mu_1} \cdot S_{1,3}^1 - \frac{\hat{\eta}_1(2)}{4\mu_1} \cdot S_{1,2}^1 - \frac{\vartheta_0}{4\mu_1} \cdot S_{0,3}^1 \\
&= \frac{\hat{\eta}_N + 3\mu_M + \vartheta_M}{4\mu_M} \cdot S_{1,3}^1 - \frac{\hat{\eta}_N}{4\mu_M} \cdot S_{1,2}^1 - \frac{\vartheta_S}{4\mu_M} \cdot S_{0,3}^1 \\
&= 523,582
\end{aligned}$$

- Για $i = 0$ και $n = 4$:

$$\begin{aligned}
S_{0,5}^1 &= \frac{\hat{\eta}_0(4) + 4 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,4}^1 - \frac{\hat{\eta}_0(3)}{5\mu_0} \cdot S_{0,3}^1 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,4}^1 \\
&= \frac{\hat{\eta}_H + 4\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,4}^1 - \frac{\hat{\eta}_H}{5\mu_S} \cdot S_{0,3}^1 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,4}^1 \\
&= -97,926512
\end{aligned}$$

- Για $i = 1$ και $n = 4$:

$$\begin{aligned}
S_{1,5}^1 &= \frac{\hat{\eta}_1(4) + 4 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,4}^1 - \frac{\hat{\eta}_1(3)}{5\mu_1} \cdot S_{1,3}^1 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,4}^1 \\
&= \frac{\hat{\eta}_N + 4\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,4}^1 - \frac{\hat{\eta}_N}{5\mu_M} \cdot S_{1,3}^1 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,4}^1 \\
&= 1127,38254
\end{aligned}$$

- Για $i = 0$ και $n = 5$:

$$\begin{aligned}
S_{0,6}^0 &= \frac{\hat{\eta}_0(5) + 5 \cdot \mu_0 + \vartheta_0}{6\mu_0} \cdot S_{0,5}^1 - \frac{\hat{\eta}_0(4)}{6\mu_0} \cdot S_{0,4}^1 - \frac{\vartheta_1}{6\mu_0} \cdot S_{1,5}^1 \\
&= \frac{\hat{\eta}_H + 5\mu_S + \vartheta_S}{6\mu_S} \cdot S_{0,5}^1 - \frac{\hat{\eta}_H}{6\mu_S} \cdot S_{0,4}^1 - \frac{\vartheta_M}{6\mu_S} \cdot S_{1,5}^1 \\
&= -195,5543125
\end{aligned}$$

- Για $i = 1$ και $n = 5$:

$$\begin{aligned}
S_{1,6}^1 &= \frac{\hat{\eta}_1(5) + 5 \cdot \mu_1 + \vartheta_1}{6\mu_1} \cdot S_{1,5}^1 - \frac{\hat{\eta}_1(4)}{6\mu_1} \cdot S_{1,4}^1 - \frac{\vartheta_0}{6\mu_1} \cdot S_{0,5}^1 \\
&= \frac{\hat{\eta}_N + 5\mu_M + \vartheta_M}{6\mu_M} \cdot S_{1,5}^1 - \frac{\hat{\eta}_N}{6\mu_M} \cdot S_{1,4}^1 - \frac{\vartheta_S}{6\mu_M} \cdot S_{0,5}^1 \\
&= 2041,808364
\end{aligned}$$

Τα αποτελέσματα των συντελεστών $S^0(i, n)$ και $S^1(i, n)$ του παραδείγματος 2.4 συγκεντρώνονται στον πίνακα 2.1. Ο πίνακας 2.2 αφορά στα αποτελέσματα των συντελεστών $S^0(i, n)$ και $S^1(i, n)$ που προκύπτουν από την υλοποίηση του αλγόριθμου MA-CAC χωρίς ουρά μεταπομπής, σε πρόγραμμα γλώσσας Python.

Πίνακας 2.1: Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 2.4

$S_{i,n}^0$		$S_{i,n}^1$	
$S_{0,0}^0 = 1$	$S_{1,0}^0 = 0$	$S_{0,0}^1 = 0$	$S_{1,0}^1 = 1$
$S_{0,1}^0 = 12,1$	$S_{1,1}^0 = -0,125$	$S_{0,1}^1 = -0,4$	$S_{1,1}^1 = 10,5$
$S_{0,2}^0 = 73,28$	$S_{1,2}^0 = -1,475$	$S_{0,2}^1 = -4,72$	$S_{1,2}^1 = 55,4$
$S_{0,3}^0 = 100,7993334$	$S_{1,3}^0 = -8,7825$	$S_{0,3}^1 = -15,384$	$S_{1,3}^1 = 196,03$
$S_{0,4}^0 = 106,5170668$	$S_{1,4}^0 = -29,10341667$	$S_{0,4}^1 = -42,1896$	$S_{1,4}^1 = 523,582$
$S_{0,5}^0 = 94,24645481$	$S_{1,5}^0 = -69,49783501$	$S_{0,5}^1 = -97,926512$	$S_{1,5}^1 = 1127,38254$
$S_{0,6}^0 = 76,56226763$	$S_{1,6}^0 = -132,9938472$	$S_{0,6}^1 = -195,5543125$	$S_{1,6}^1 = 2041,808364$

Πίνακας 2.2: Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 2.4 (Πρόγραμμα σε Python)

n	$S_{0,n}^0$	$S_{1,n}^0$	$S_{0,n}^1$	$S_{1,n}^1$
0	1	0	0	1
1	12,1	-0,125	-0,4	10,5
2	73,28	-1,475	-4,72	55,4
3	100,7993333	-8,7825	-15,384	196,03
4	106,5170667	-29,10341667	-42,1896	523,582
5	94,24645467	-69,497835	-97,926512	1127,38254
6	76,56226747	-132,9938471	-195,5543125	2041,808364

Μέσω της σχέσης (2.12), υπολογίζουμε το r :

$$r = -\frac{S_{0,6}^0 \cdot (6\mu_S + \vartheta_S) - S_{0,5}^0 \cdot \hat{\lambda}_H - S_{1,6}^0 \cdot \vartheta_M}{S_{0,6}^1 \cdot (6\mu_S + \vartheta_S) - S_{0,5}^1 \cdot \hat{\lambda}_H - S_{1,6}^1 \cdot \vartheta_M} = 0,088535555$$

Επομένως, μέσω της σχέσης (2.15), προσδιορίζουμε την τιμή της πιθανότητας $P(0, 0)$:

$$P(0, 0) = \left(\sum_{i=0}^1 \sum_{n=0}^6 S_{i,n}^0 + r \cdot \sum_{i=0}^1 \sum_{n=0}^6 S_{i,n}^1 \right)^{-1} = 0,00184769872$$

Συνειπώς, λόγω της σχέσης (2.13), η τιμή της πιθανότητας $P(1, 0)$ είναι:

$$P(1, 0) = r \cdot P(0, 0) = 0,0001635870316$$

Από τη σχέση (2.6) υπολογίζουμε τις υπόλοιπες πιθανότητες $P(i, n)$, ως εξής:

$$P(0, 1) = S_{0,1}^0 \cdot P(0, 0) + S_{0,1}^1 \cdot P(1, 0) = 0,022291719$$

$$P(1, 1) = S_{1,1}^0 \cdot P(0, 0) + S_{1,1}^1 \cdot P(1, 0) = 0,00148670149$$

$$P(0, 2) = S_{0,2}^0 \cdot P(0, 0) + S_{0,2}^1 \cdot P(1, 0) = 0,134627231$$

$$P(1, 2) = S_{1,2}^0 \cdot P(0, 0) + S_{1,2}^1 \cdot P(1, 0) = 0,0063373659$$

$$P(0, 3) = S_{0,3}^0 \cdot P(0, 0) + S_{0,3}^1 \cdot P(1, 0) = 0,183730176$$

$$P(1, 3) = S_{1,3}^0 \cdot P(0, 0) + S_{1,3}^1 \cdot P(1, 0) = 0,015840551$$

$$P(0, 4) = S_{0,4}^0 \cdot P(0, 0) + S_{0,4}^1 \cdot P(1, 0) = 0,189909776$$

$$P(1, 4) = S_{1,4}^0 \cdot P(0, 0) + S_{1,4}^1 \cdot P(1, 0) = 0,031876879$$

$$P(0, 5) = S_{0,5}^0 \cdot P(0, 0) + S_{0,5}^1 \cdot P(1, 0) = 0,158119546$$

$$P(1, 5) = S_{1,5}^0 \cdot P(0, 0) + S_{1,5}^1 \cdot P(1, 0) = 0,056014102$$

$$P(0, 6) = S_{0,6}^0 \cdot P(0, 0) + S_{0,6}^1 \cdot P(1, 0) = 0,109473854$$

$$P(1, 6) = S_{1,6}^0 \cdot P(0, 0) + S_{1,6}^1 \cdot P(1, 0) = 0,0882808$$

Τα αποτελέσματα των πιθανοτήτων $P(i, n)$ των καταστάσεων του συστήματος, για τη διαδικασία που ακολουθήσαμε στο Παράδειγμα 2.4, συνοψίζονται στον πίνακα 2.3. Στον πίνακα 2.4, παρουσιάζονται τα αποτελέσματα των $P(i, n)$ του προγράμματος MA-CAC χωρίς ουρά μεταπομπής σε Python.

Η πιθανότητα απώλειας των νέων κλήσεων στη φάση στάσης του οχήματος, λόγω της σχέσης (2.16), είναι:

$$B_S^N = \sum_{n=2}^6 P(0, n) = 0,775860583$$

Η πιθανότητα απώλειας των νέων κλήσεων, όταν το όχημα κινείται, υπολογίζεται μέσω της σχέσης (2.17):

$$B_S^M = \sum_{n=6}^6 P(1, n) = P(1, 6) = 0,0882808$$

Πίνακας 2.3: Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 2.4

$P(i, n)$	
$P(0, 0) = 0,00184769872$	$P(1, 0) = 0,0001635870316$
$P(0, 1) = 0,022291719$	$P(1, 1) = 0,00148670149$
$P(0, 2) = 0,134627231$	$P(1, 2) = 0,0063373659$
$P(0, 3) = 0,183730176$	$P(1, 3) = 0,015840551$
$P(0, 4) = 0,189909776$	$P(1, 4) = 0,031876879$
$P(0, 5) = 0,158119546$	$P(1, 5) = 0,056014102$
$P(0, 6) = 0,109473854$	$P(1, 6) = 0,0882808$

Πίνακας 2.4: Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 2.4 (Πρόγραμμα σε Python)

n	$P(0, n)$	$P(1, n)$
0	0,001847699	0,000163587
1	0,02229172	0,001486701
2	0,134627231	0,006337366
3	0,183730176	0,015840552
4	0,189909776	0,03187688
5	0,158119546	0,056014103
6	0,109473854	0,08828081

Επομένως η συνολική πιθανότητα απώλειας των νέων κλήσεων στο σύστημα είναι (από (2.18)):

$$P_{NCBP} = B_S^N + B_M^N = 0,864141383$$

Η πιθανότητα απόρριψης των κλήσεων μεταπομπής του συστήματος, υπολογίζεται μέσω της σχέσης (2.19):

$$P_{HCDP} = P(n = 6/i = 0) = \frac{P(0, 6)}{\sum_{n=0}^6 P(0, n)} = 0,136842317$$

Τέλος, για τον υπολογισμό του ποσοστού χρήσης των καναλιών χρησιμοποιούμε τη σχέση (2.20):

$$U = \frac{\sum_{i=0}^1 \sum_{n=0}^6 n \cdot P(i, n)}{6} = 0,674793763$$

Τα αποτελέσματα των πιθανοτήτων απώλειας των νέων κλήσεων (P_{NCBP}) και απόρριψης των κλήσεων μεταπομπής (P_{HCDP}), καθώς και το ποσοστό χρήσης των καναλιών (Utilization), συνοψίζονται στον πίνακα 2.5:

Στον πίνακα 2.6 παρουσιάζονται τα αποτελέσματα των πιθανοτήτων απώλειας των νέων κλήσεων (P_{NCBP}) και απόρριψης των κλήσεων μεταπομπής (P_{HCDP}) καθώς και το ποσοστό

Πίνακας 2.5: P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 2.4

P_{NCBP}	0,864141383
P_{HCDP}	0,136842317
<i>Utilization</i>	0,674793763

χρήσης των καναλιών (Utilization), όπως προέκυψαν από το πρόγραμμα MA-CAC χωρίς ουρά μεταπομπής, γλώσσας Python:

Πίνακας 2.6: P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 2.4 (Πρόγραμμα σε Python)

P_{NCBP}	0,864141392
P_{HCDP}	0,136842317
<i>Utilization</i>	0,674793775

Κεφάλαιο 3

Μοντέλο Απωλειών Non Priority

3.1 Εισαγωγή

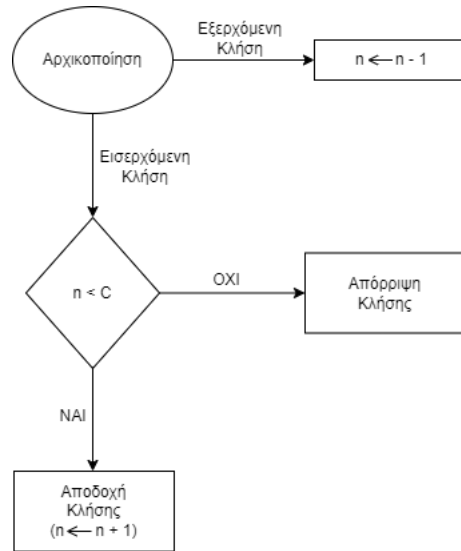
Στο 3ο Κεφάλαιο παρουσιάζεται το τρίτο κατά σειρά μοντέλο απωλειών, το σύστημα Non Priority (NP). Το μοντέλο NP, σε σύγκριση με τα μοντέλα απωλειών MA-CAC που παρουσιάστηκαν στα Κεφάλαια 1 και 2, δεν διατηρεί δεσμευμένα κανάλια για την αποκλειστική χρήση τους από κλήσεις μεταπομπής, σε καμία από τις δύο φάσεις κίνησης του οχήματος. Συνεπώς, τόσο οι κλήσεις μεταπομπής όσο και οι νέες κλήσεις, αντιμετωπίζονται από το σύστημα NP ως απλά εισερχόμενες κλήσεις προς εξυπηρέτηση.

Στην Ενότητα 3.2, γίνεται η παρουσίαση του αλγόριθμου του συστήματος NP, μαζί με το αντίστοιχο διάγραμμα ροής. Στην Ενότητα 3.3, περιγράφεται αναλυτικά το μοντέλο NP και παρατίθενται οι τύποι υπολογισμού των πιθανοτήτων απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής, καθώς και ο τύπος υπολογισμού του ποσοστού χρήσης των καναλιών που προσφέρονται από το σημείο πρόσβασης του δικτύου WLAN. Επιπλέον παρουσιάζεται μία μέθοδος υπολογισμού των πιθανοτήτων των καταστάσεων του συστήματος, οι οποίες είναι απαραίτητες για τον καθορισμό των προαναφεθέντων πιθανοτήτων απόρριψης των κλήσεων αλλά και για τον υπολογισμό του ποσοστού χρήσης των καναλιών. Τέλος, στην Ενότητα 3.4 επιλύεται ένα αριθμητικό παράδειγμα του μοντέλου NP με τη χρήση της μεθοδολογίας της Ενότητας 3.3.

3.2 Ο Αλγόριθμος NP

Στο Σχήμα 3.1, όπου παρουσιάζεται το διάγραμμα ροής του αλγόριθμου NP, παρατηρούμε ότι μόλις εισέλθει μία κλήση στο σύστημα εξετάζεται αν ο αριθμός των ενεργών κλήσεων είναι μικρότερος του αριθμού των διαθέσιμων καναλιών του σημείου πρόσβασης του δικτύου WLAN του οχήματος. Αν είναι, τότε η κλήση γίνεται αποδεκτή από το σύστημα ενώ στην περίπτωση όπου ο αριθμός των κλήσεων υπερβαίνει τη χωρητικότητα του συστήματος, η κλήση απορρίπτεται.

Ο αλγόριθμος NP, σε σύγκριση με τους αλγόριθμους των μοντέλων MA-CAC, με ουρά μεταπομπής και MA-CAC, χωρίς ουρά μεταπομπής που παρουσιάστηκαν στα προηγούμενα Κεφάλαια, δεν λαμβάνει υπόψη του τις καταστάσεις κινητικότητας του οχήματος (φάση στάσης/φάση κίνησης). Αυτή η συμπεριφορά του αλγόριθμου οφείλεται στο γεγονός ότι το σύστημα NP αντιμετωπίζει τις εισερχόμενες κλήσεις με τον ίδιο τρόπο,

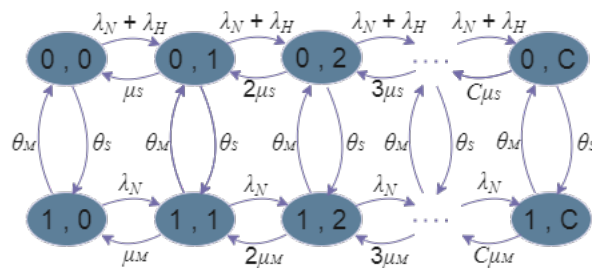


Σχήμα 3.1: Διάγραμμα Ροής του Αλγορίθμου NP.

τόσο κατά τη διάρκεια της φάσης κίνησης όσο και κατά τη διάρκεια της φάσης στην οποία το όχημα είναι στατικό.

3.3 Ανάλυση του μοντέλου

Όπως αναφέρθηκε στην εισαγωγή (Ενότητα 3.1) του παρόντος Κεφαλαίου, το μοντέλο Non Priority δεν διατηρεί κανάλια προς αποκλειστική χρήση από κλήσεις μεταπομπής, σε καμία από τις δύο φάσεις κινητικότητας του οχήματος. Συνεπώς δεν παρέχει κανενός είδους προτεραιότητα στις κλήσεις μεταπομπής και έτσι κάθε κλήση που εισέρχεται στο σύστημα, είτε είναι νέα κλήση είτε προέρχεται από μεταπομπή, αντιμετωπίζεται το ίδιο. Επομένως στο μοντέλο NP όλες οι εισερχόμενες κλήσεις έχουν την ίδια βαρύτητα. Το γεγονός αυτό διαφοροποιεί σημαντικά το μοντέλο NP από τα μοντέλα MA-CAC, είτε με ενσωματωμένη την ουρά μεταπομπής είτε δίχως αυτή, που αναπτύχθηκαν στα προηγούμενα Κεφάλαια.



Σχήμα 3.2: Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου του Μοντέλου Απωλειών NP.

Στο σύστημα NP, όπως και στα προηγούμενα μοντέλα, η διαδικασία άφιξης των κλήσεων μεταπομπής λ_H καθώς και η διαδικασία άφιξης των νέων κλήσεων λ_N , ακολουθούν κατανομή Poisson. Έτσι, ο συνολικός ρυθμός άφιξης των κλήσεων έχει τιμή $\lambda_H + \lambda_N$, καθ'όλη τη διάρκεια της φάσης στάσης του οχήματος. Ο μέσος χρόνος παραμονής των κλήσεων ακολουθεί εκθετική κατανομή με ρυθμούς θ_S για τη φάση στάσης και θ_M για τη φάση κίνησης του οχήματος αντίστοιχα, όπως επίσης και η μέση διάρκεια των

κλήσεων με ρυθμούς μ_S για τη φάση στάσης και μ_M για τη φάση που το όχημα βρίσκεται σε κίνηση. Για το ποσοστό αναχώρησης των κλήσεων από το σύστημα, οι τιμές είναι $n \cdot \mu_S$ και $n \cdot \mu_M$, για τις φάσεις στάσης και κίνησης αντιστοίχως. Τέλος, όπως και στα μοντέλα των Κεφαλαίων 1 και 2, η κάθε κλήση απαιτεί μόλις μία μονάδα εύρους ζώνης (Ib.u.) από τους πόρους του συστήματος, προκειμένου να εξυπηρετηθεί.

Από τη Μαρκοβιανή αλυσίδα συνεχούς χρόνου του συστήματος NP (Σχήμα 3.2), προκύπτει ότι ο ρυθμός μετάβασης από μία κατάσταση του συστήματος σε μία άλλη, είναι [1]:

$$\begin{aligned}
 r(0, n; 1, n) &= \vartheta_S, & \text{για } 0 \leq n \leq C \\
 r(1, n; 0, n) &= \vartheta_M, & \text{για } 0 \leq n \leq C \\
 r(0, n; 0, n-1) &= n \cdot \mu_S, & \text{για } 0 < n \leq C \\
 r(1, n; 1, n-1) &= n \cdot \mu_M, & \text{για } 0 < n \leq C \\
 r(0, n; 0, n+1) &= \hat{\lambda}_N + \hat{\lambda}_H, & \text{για } 0 \leq n < C \\
 r(1, n; 1, n+1) &= \hat{\lambda}_N, & \text{για } 0 \leq n < C
 \end{aligned} \tag{3.1}$$

Οι εξισώσεις ισορροπίας του συστήματος NP, για τον καθορισμό των πιθανοτήτων $P(i, n)$, είναι [1]:

1) Για $i = 0$ και $n = 0$ έχουμε:

$$((\hat{\lambda}_N + \hat{\lambda}_H) + \vartheta_S) P(i, n) = (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

2) Για $i = 0$ και $0 < n < C$ έχουμε:

$$((\hat{\lambda}_N + \hat{\lambda}_H) + n\mu_S + \vartheta_S) P(i, n) = (\hat{\lambda}_N + \hat{\lambda}_H) P(i, n-1) + (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

3) Για $i = 0$ και $n = C$ έχουμε:

$$(C\mu_S + \vartheta_S) P(i, n) = \hat{\lambda}_H P(i, n-1) + \vartheta_M P(1, n)$$

4) Για $i = 1$ και $n = 0$ έχουμε:

$$(\hat{\lambda}_N + \vartheta_M) P(i, n) = (n+1) \mu_M P(i, n+1) + \vartheta_S P(0, n)$$

5) Για $i = 1$ και $0 < n < C$ έχουμε:

$$(\hat{\lambda}_N + n\mu_M + \vartheta_M) P(i, n) = \hat{\lambda}_N P(i, n-1) + (n+1) \mu_M P(i, n+1) + \vartheta_S P(0, n)$$

6) Για $i = 1$ και $n = C$ έχουμε:

$$(C\mu_M + \vartheta_M) P(i, n) = \hat{\lambda}_N P(i, n-1) + C\mu_M P(i, n+1) + \vartheta_S P(0, n) \tag{3.2}$$

Ένας πιο συμπαγής τρόπος για την έκφραση των οριακών καταστάσεων $(0, 0)$ και $(1, 0)$ των εξισώσεων (3.2), είναι η ακόλουθη εξίσωση:

$$(\hat{\lambda}_i(0) + \vartheta_i) P(i, 0) - \mu_i P(i, 1) - \vartheta_j P(j, 0) = 0 \quad (3.3)$$

Οι ενδιαμέσες καταστάσεις $(1, 0), (2, 0), \dots, (C-1, 0)$ και $(1, 1), (2, 1), \dots, (C-1, 1)$ των εξισώσεων (3.2), μπορούν να εκφραστούν από την εξίσωση:

$$(\hat{\lambda}_i(n) + n\mu_i + \vartheta_i) P(i, n) - \hat{\lambda}_i(n-1) P(i, n-1) - \vartheta_j P(j, n) - (n+1)\mu_i P(i, n+1) = 0 \quad (3.4)$$

Τέλος, για τις οριακές καταστάσεις $(0, C)$ και $(1, C)$ των εξισώσεων (3.2), έχουμε:

$$(C\mu_i + \vartheta_i) P(i, C) - \hat{\lambda}_i(C-1) P(i, C-1) - \vartheta_j P(j, C) = 0 \quad (3.5)$$

με:

$$i = \begin{cases} 0, & \text{για τη φάση στάσης} \\ 1, & \text{για τη φάση κίνησης} \end{cases} \quad \mu_i = \begin{cases} \mu_S, & \text{όταν } i = 0 \\ \mu_M, & \text{όταν } i = 1 \end{cases}$$

$$\vartheta_i = \begin{cases} \vartheta_S, & \text{όταν } i = 0 \\ \vartheta_M, & \text{όταν } i = 1 \end{cases} \quad \hat{\lambda}_i(n) = \begin{cases} \hat{\lambda}_N + \hat{\lambda}_H, & \text{όταν } i = 0 \text{ και } 0 \leq n < C \\ \hat{\lambda}_N, & \text{όταν } i = 1 \text{ και } 0 \leq n < C \\ 0, & \text{όταν } n < 0 \end{cases}$$

και $j = 1 - i$

Οι πιθανότητες $P(i, n)$ των καταστάσεων του συστήματος, όπως και στα μοντέλα των προηγούμενων Κεφαλαίων, προκύπτουν από τη σχέση:

$$P(i, n) = S_{i,n}^0 \cdot P(0, 0) + S_{i,n}^1 \cdot P(1, 0) \quad (3.6)$$

Ο προσδιορισμός των συντελεστών $S_{i,n}^0$ και $S_{i,n}^1$ της σχέσης (3.6), γίνεται με τον τρόπο που αναπτύχθηκε και στο μοντέλο MA-CAC χωρίς ουρά μεταπομπής. Επομένως, με την αντικατάσταση της εξίσωσης (3.6) στην εξίσωση (3.4) έχουμε:

$$\begin{aligned} S_{i,n+1}^0 \cdot P(0, 0) + S_{i,n+1}^1 \cdot P(1, 0) &= \frac{\hat{\lambda}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot (S_{i,n}^0 \cdot P(0, 0) + S_{i,n}^1 \cdot P(1, 0)) \\ &\quad - \frac{\hat{\lambda}_i(n-1)}{(n+1)\mu_i} \cdot ((S_{i,n-1}^0 \cdot P(0, 0) + S_{i,n-1}^1 \cdot P(1, 0)) \\ &\quad - \frac{\vartheta_j}{(n+1)\mu_i} \cdot ((S_{j,n}^0 \cdot P(0, 0) + S_{j,n}^1 \cdot P(1, 0)) \end{aligned} \quad (3.7)$$

Για $P(0, 0) = 1$ και $P(1, 0) = 0$, η σχέση (3.7) γίνεται:

$$S_{i,n+1}^0 = \frac{\hat{\lambda}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot S_{i,n}^0 - \frac{\hat{\lambda}_i(n-1)}{(n+1)\mu_i} \cdot S_{i,n-1}^0 - \frac{\vartheta_j}{(n+1)\mu_i} \cdot S_{j,n}^0 \quad (3.8)$$

όπου: $S_{0,0}^0 = 1$, $S_{1,0}^0 = 0$ και $S_{0,y}^0 = 0$, $S_{1,y}^0 = 0$ για $y < 0$

ενώ, για $P(0, 0) = 0$ και $P(1, 0) = 1$, η σχέση (3.7) γίνεται:

$$S_{i,n+1}^1 = \frac{\hat{\mu}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot S_{i,n}^1 - \frac{\hat{\mu}_i(n-1)}{(n+1)\mu_i} \cdot S_{i,n-1}^1 - \frac{\vartheta_j}{(n+1)\mu_i} \cdot S_{j,n}^1 \quad (3.9)$$

όπου: $S_{0,0}^1 = 0$, $S_{1,0}^1 = 1$ και $S_{0,y}^1 = 0$, $S_{1,y}^1 = 0$ για $y < 0$

Για τον υπολογισμό των πιθανοτήτων $P(0, 0)$ και $P(1, 0)$, θεωρούμε στην εξίσωση (3.5) ότι $i = 0$ και προκύπτει:

$$(C\mu_S + \vartheta_S)P(0, C) - \hat{\mu}_0(C-1)P(0, C-1) - \vartheta_M P(1, C) = 0 \quad (3.10)$$

(Σημειώνουμε ότι και για την περίπτωση $i = 1$ προκύπτουν τα ίδια αποτελέσματα)

Η εξίσωση (3.10), μέσω της σχέσης (3.6), γίνεται:

$$\begin{aligned} & (C\mu_S + \vartheta_S) \cdot (S_{0,C}^0 \cdot P(0, 0) + S_{0,C}^1 \cdot P(1, 0)) \\ & - \hat{\mu}_0(C-1) \cdot (S_{0,C-1}^0 \cdot P(0, 0) + S_{0,C-1}^1 \cdot P(1, 0)) \\ & - \vartheta_M \cdot (S_{1,C}^0 \cdot P(0, 0) + S_{1,C}^1 \cdot P(1, 0)) \\ & = 0 \Rightarrow \\ & P(0, 0) \cdot (S_{0,C}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^0 \cdot \hat{\mu}_0 - S_{1,C}^0 \cdot \vartheta_M) \\ & + P(1, 0) \cdot (S_{0,C}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^1 \cdot \hat{\mu}_0 - S_{1,C}^1 \cdot \vartheta_M) \\ & = 0 \Rightarrow \\ & \frac{P(1, 0)}{P(0, 0)} = - \frac{S_{0,C}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^0 \cdot \hat{\mu}_0 - S_{1,C}^0 \cdot \vartheta_M}{S_{0,C}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^1 \cdot \hat{\mu}_0 - S_{1,C}^1 \cdot \vartheta_M} \end{aligned} \quad (3.11)$$

Στην τελευταία εξίσωση θέτουμε:

$$r = - \frac{S_{0,C}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^0 \cdot \hat{\mu}_0 - S_{1,C}^0 \cdot \vartheta_M}{S_{0,C}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^1 \cdot \hat{\mu}_0 - S_{1,C}^1 \cdot \vartheta_M} \quad (3.12)$$

οπότε η εξίσωση (3.11) μετασχηματίζεται σε:

$$\frac{P(1, 0)}{P(0, 0)} = r \Rightarrow P(1, 0) = r \cdot P(0, 0) \quad (3.13)$$

Γενικά ισχύει ότι:

$$\sum_{i=0}^1 \sum_{n=0}^C P(i, n) = 1 \quad (3.14)$$

Άρα, λόγω της εξίσωσης (3.6), η σχέση (3.14) γίνεται:

$$\begin{aligned} P(0, 0) \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^0 + P(1, 0) \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^1 &= 1 \\ \Rightarrow \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^0 + \frac{P(1, 0) \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^1}{P(0, 0)} &= \frac{1}{P(0, 0)} \\ \Rightarrow P(0, 0) &= \left(\sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^0 + r \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^1 \right)^{-1} \end{aligned} \quad (3.15)$$

Στο μοντέλο NP, τόσο στη φάση στάσης όσο και στη φάση κίνησης, οι νέες κλήσεις που εισέρχονται στο σύστημα απορρίπτονται μόνο όταν όλα τα κανάλια του συστήματος είναι απασχολημένα, εξυπηρετώντας προηγούμενες εισερχόμενες κλήσεις. Επομένως για τον υπολογισμό της πιθανότητας απώλειας των νέων κλήσεων, για τη φάση κατά την οποία το όχημα είναι στατικό, έχουμε:

$$B_S^N = P(0, C) \quad (3.16)$$

και για την πιθανότητα απώλειας των νέων κλήσεων, ενώ το όχημα κινείται, έχουμε:

$$B_M^N = P(1, C) \quad (3.17)$$

Συνεπώς, η συνολική πιθανότητα απώλειας των νέων κλήσεων δίνεται από τη σχέση [1]:

$$P_{NCBP} = B_S^N + B_M^N = P(0, C) + P(1, C) \quad (3.18)$$

Επιπλέον στο σύστημα NP, όπως συμβαίνει και στην περίπτωση των νέων κλήσεων, οι κλήσεις που προέρχονται από μεταπομπή απορρίπτονται όταν όλα τα κανάλια του δικτύου είναι κατειλημμένα. Άρα, για τον υπολογισμό της πιθανότητας απόρριψης των κλήσεων μεταπομπής έχουμε [1]:

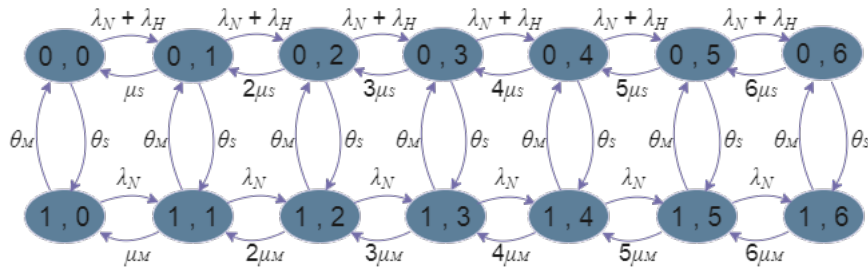
$$P_{HCDP} = P(n = C/i = 0) = \frac{P(0, C)}{\sum_{n=0}^C P(0, n)} \quad (3.19)$$

Τέλος, το ποσοστό χρήσης των καναλιών στο μοντέλο NP, υπολογίζεται με τον ίδιο τύπο που χρησιμοποιείται και στο μοντέλο MA-CAC χωρίς ουρά μεταπομπής, δηλαδή:

$$U = \frac{\sum_{i=0}^1 \sum_{n=0}^C n \cdot P(i, n)}{C} = \frac{\sum_{n=0}^C n \cdot (P(0, n) + P(1, n))}{C} \quad (3.20)$$

3.4 Παράδειγμα

Στην τελευταία Ενότητα του 3ου Κεφαλαίου, παρουσιάζεται μία αριθμητική εφαρμογή του μοντέλου απωλειών NP. Η χωρητικότητα του συστήματος έχει καθοριστεί στην τιμή $C = 6$. Οι κλήσεις μεταπομπής και οι νέες κλήσεις αφικνούνται στο σύστημα με ρυθμούς $\lambda_H = 4$ και $\lambda_N = 2$ αντίστοιχα. Ο ρυθμός της μέσης διάρκειας των κλήσεων στη φάση στάσης έχει τιμή $\mu_S = 0,4$, ενώ στη φάση κίνησης έχει τιμή $\mu_M = 0,25$. Τέλος, ο ρυθμός μετάβασης από τη φάση στάσης στη φάση κίνησης είναι $\vartheta_S = 0,025$ και ο ρυθμός μετάβασης από την κατάσταση κατά την οποία το όχημα κινείται στην κατάσταση όπου το όχημα είναι στατικό, είναι $\vartheta_M = 0,02$.



Σχήμα 3.3: Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για το Μοντέλο Απωλειών NP του Παραδείγματος 3.4.

Οι εξισώσεις ισορροπίας του συστήματος, από τις σχέσεις (3.3), (3.4) και (3.5), είναι:

- Για την κατάσταση (0, 0):

$$\begin{aligned} \mu_S \cdot P(0, 1) + \vartheta_M \cdot P(1, 0) &= (\lambda_H + \lambda_N + \vartheta_S) \cdot P(0, 0) \\ \Rightarrow 0,4 \cdot P(0, 1) + 0,02 \cdot P(1, 0) &= 6,025 \cdot P(0, 0) \\ \Rightarrow 0,4 \cdot P(0, 1) + 0,02 \cdot P(1, 0) - 6,025 \cdot P(0, 0) &= 0 \end{aligned} \quad (1)$$

- Για την κατάσταση (1, 0):

$$\begin{aligned} \mu_M \cdot P(1, 1) + \vartheta_S \cdot P(0, 0) &= (\lambda_N + \vartheta_M) \cdot P(1, 0) \\ \Rightarrow 0,25 \cdot P(1, 1) + 0,025 \cdot P(0, 0) &= 2,02 \cdot P(1, 0) \\ \Rightarrow 0,25 \cdot P(1, 1) + 0,025 \cdot P(0, 0) - 2,02 \cdot P(1, 0) &= 0 \end{aligned} \quad (2)$$

- Για την κατάσταση (0, 1):

$$\begin{aligned}
& (\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 0) + 2\mu_S \cdot P(0, 2) + \vartheta_M \cdot P(1, 1) = (\hat{\lambda}_H + \hat{\lambda}_N + \mu_S + \vartheta_S) \cdot P(0, 1) \\
\Rightarrow & 6 \cdot P(0, 0) + 0,8 \cdot P(0, 2) + 0,02 \cdot P(1, 1) = 6,425 \cdot P(0, 1) \\
\Rightarrow & 6 \cdot P(0, 0) + 0,8 \cdot P(0, 2) + 0,02 \cdot P(1, 1) - 6,425 \cdot P(0, 1) = 0 \tag{3}
\end{aligned}$$

- Για την κατάσταση (1, 1):

$$\begin{aligned}
& \hat{\lambda}_N \cdot P(1, 0) + 2\mu_M \cdot P(1, 2) + \vartheta_S \cdot P(0, 1) = (\hat{\lambda}_N + \mu_M + \vartheta_M) \cdot P(1, 1) \\
\Rightarrow & 2 \cdot P(1, 0) + 0,5 \cdot P(1, 2) + 0,025 \cdot P(0, 1) = 2,27 \cdot P(1, 1) \\
\Rightarrow & 2 \cdot P(1, 0) + 0,5 \cdot P(1, 2) + 0,025 \cdot P(0, 1) - 2,27 \cdot P(1, 1) = 0 \tag{4}
\end{aligned}$$

- Για την κατάσταση (0, 2):

$$\begin{aligned}
& (\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 1) + 3\mu_S \cdot P(0, 3) + \vartheta_M \cdot P(1, 2) = (\hat{\lambda}_H + \hat{\lambda}_N + 2\mu_S + \vartheta_S) \cdot P(0, 2) \\
\Rightarrow & 6 \cdot P(0, 1) + 1,2 \cdot P(0, 3) + 0,02 \cdot P(1, 2) = 6,825 \cdot P(0, 2) \\
\Rightarrow & 6 \cdot P(0, 1) + 1,2 \cdot P(0, 3) + 0,02 \cdot P(1, 2) - 6,825 \cdot P(0, 2) = 0 \tag{5}
\end{aligned}$$

- Για την κατάσταση (1, 2):

$$\begin{aligned}
& \lambda_N \cdot P(1, 1) + 3\mu_M \cdot P(1, 3) + \vartheta_S \cdot P(0, 2) = (\hat{\lambda}_N + 2\mu_M + \vartheta_M) \cdot P(1, 2) \\
\Rightarrow & 2 \cdot P(1, 1) + 0,75 \cdot P(1, 3) + 0,025 \cdot P(0, 2) = 2,52 \cdot P(1, 2) \\
\Rightarrow & 2 \cdot P(1, 1) + 0,75 \cdot P(1, 3) + 0,025 \cdot P(0, 2) - 2,52 \cdot P(1, 2) = 0 \tag{6}
\end{aligned}$$

- Για την κατάσταση (0, 3):

$$\begin{aligned}
& (\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 2) + 4\mu_S \cdot P(0, 4) + \vartheta_M \cdot P(1, 3) = (\hat{\lambda}_H + \hat{\lambda}_N + 3\mu_S + \vartheta_S) \cdot P(0, 3) \\
\Rightarrow & 6 \cdot P(0, 2) + 1,6 \cdot P(0, 4) + 0,02 \cdot P(1, 3) = 7,225 \cdot P(0, 3) \\
\Rightarrow & 6 \cdot P(0, 2) + 1,6 \cdot P(0, 4) + 0,02 \cdot P(1, 3) - 7,225 \cdot P(0, 3) = 0 \tag{7}
\end{aligned}$$

- Για την κατάσταση (1, 3):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 2) + 4\mu_M \cdot P(1, 4) + \vartheta_S \cdot P(0, 3) &= (\hat{\lambda}_N + 3\mu_M + \vartheta_M) \cdot P(1, 3) \\ \Rightarrow 2 \cdot P(1, 2) + 1 \cdot P(1, 4) + 0,025 \cdot P(0, 3) &= 2,77 \cdot P(1, 3) \\ \Rightarrow 2 \cdot P(1, 2) + 1 \cdot P(1, 4) + 0,025 \cdot P(0, 3) - 2,77 \cdot P(1, 3) &= 0 \end{aligned} \quad (8)$$

- Για την κατάσταση (0, 4):

$$\begin{aligned} (\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 3) + 5\mu_S \cdot P(0, 5) + \vartheta_M \cdot P(1, 4) &= (\hat{\lambda}_H + \hat{\lambda}_N + 4\mu_S + \vartheta_S) \cdot P(0, 4) \\ \Rightarrow 6 \cdot P(0, 3) + 2 \cdot P(0, 5) + 0,02 \cdot P(1, 4) &= 7,625 \cdot P(0, 4) \\ \Rightarrow 6 \cdot P(0, 3) + 2 \cdot P(0, 5) + 0,02 \cdot P(1, 4) - 7,625 \cdot P(0, 4) &= 0 \end{aligned} \quad (9)$$

- Για την κατάσταση (1, 4):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 3) + 5\mu_M \cdot P(1, 5) + \vartheta_S \cdot P(0, 4) &= (\hat{\lambda}_N + 4\mu_M + \vartheta_M) \cdot P(1, 4) \\ \Rightarrow 2 \cdot P(1, 3) + 1,25 \cdot P(1, 5) + 0,025 \cdot P(0, 4) &= 3,02 \cdot P(1, 4) \\ \Rightarrow 2 \cdot P(1, 3) + 1,25 \cdot P(1, 5) + 0,025 \cdot P(0, 4) - 3,02 \cdot P(1, 4) &= 0 \end{aligned} \quad (10)$$

- Για την κατάσταση (0, 5):

$$\begin{aligned} (\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 4) + 6\mu_S \cdot P(0, 6) + \vartheta_M \cdot P(1, 5) &= (\hat{\lambda}_H + \hat{\lambda}_N + 5\mu_S + \vartheta_S) \cdot P(0, 5) \\ \Rightarrow 6 \cdot P(0, 4) + 2,4 \cdot P(0, 6) + 0,02 \cdot P(1, 5) &= 8,025 \cdot P(0, 5) \\ \Rightarrow 6 \cdot P(0, 4) + 2,4 \cdot P(0, 6) + 0,02 \cdot P(1, 5) - 8,025 \cdot P(0, 5) &= 0 \end{aligned} \quad (11)$$

- Για την κατάσταση (1, 5):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 4) + 6\mu_M \cdot P(1, 6) + \vartheta_S \cdot P(0, 5) &= (\hat{\lambda}_N + 5\mu_M + \vartheta_M) \cdot P(1, 5) \\ \Rightarrow 2 \cdot P(1, 4) + 1,5 \cdot P(1, 6) + 0,025 \cdot P(0, 5) &= 3,27 \cdot P(1, 5) \\ \Rightarrow 2 \cdot P(1, 4) + 1,5 \cdot P(1, 6) + 0,025 \cdot P(0, 5) - 3,27 \cdot P(1, 5) &= 0 \end{aligned} \quad (12)$$

- Για την κατάσταση (0, 6):

$$\begin{aligned}
 (\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 5) + \vartheta_M \cdot P(1, 6) &= (6\mu_S + \vartheta_S) \cdot P(0, 6) \\
 \Rightarrow 6 \cdot P(0, 5) + 0,02 \cdot P(1, 6) &= 2,425 \cdot P(0, 6) \\
 \Rightarrow 6 \cdot P(0, 5) + 0,02 \cdot P(1, 6) - 2,425 \cdot P(0, 6) &= 0
 \end{aligned} \tag{13}$$

- Για την κατάσταση (1, 6):

$$\begin{aligned}
 \hat{\lambda}_N \cdot P(1, 5) + \vartheta_S \cdot P(0, 6) &= (6\mu_M + \vartheta_M) \cdot P(1, 6) \\
 \Rightarrow 2 \cdot P(1, 5) + 0,025 \cdot P(0, 6) &= 1,52 \cdot P(1, 6) \\
 \Rightarrow 2 \cdot P(1, 5) + 0,025 \cdot P(0, 6) - 1,52 \cdot P(1, 6) &= 0
 \end{aligned} \tag{14}$$

Από την αναδρομική σχέση (3.8), με αρχικές τιμές $S_{0,0}^0 = 1$, $S_{1,0}^0 = 0$ και $S_{0,y}^0 = 0$, $S_{1,y}^0 = 0$ για $y < 0$, υπολογίζουμε τους συντελεστές $S_{i,n}^0$ ως εξής:

- Για $i = 0$ και $n = 0$:

$$\begin{aligned}
 S_{0,1}^0 &= \frac{\hat{\lambda}_0(0) + 0 \cdot \mu_0 + \vartheta_0}{\mu_0} \cdot S_{0,0}^0 - \frac{\hat{\lambda}_0(-1)}{\mu_0} \cdot S_{0,-1}^0 - \frac{\vartheta_1}{\mu_0} \cdot S_{1,0}^0 \\
 &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + \vartheta_S}{\mu_S} \cdot S_{0,0}^0 = 15,0625
 \end{aligned}$$

- Για $i = 1$ και $n = 0$:

$$\begin{aligned}
 S_{1,1}^0 &= \frac{\hat{\lambda}_1(0) + 0 \cdot \mu_1 + \vartheta_1}{\mu_1} \cdot S_{1,0}^0 - \frac{\hat{\lambda}_1(-1)}{\mu_1} \cdot S_{1,-1}^0 - \frac{\vartheta_0}{\mu_1} \cdot S_{0,0}^0 \\
 &= -\frac{\vartheta_S}{\mu_M} \cdot S_{0,0}^0 = -0,1
 \end{aligned}$$

- Για $i = 0$ και $n = 1$:

$$\begin{aligned}
 S_{0,2}^0 &= \frac{\hat{\lambda}_0(1) + 1 \cdot \mu_0 + \vartheta_0}{2\mu_0} \cdot S_{0,1}^0 - \frac{\hat{\lambda}_0(0)}{2\mu_0} \cdot S_{0,0}^0 - \frac{\vartheta_1}{2\mu_0} \cdot S_{1,1}^0 \\
 &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + \mu_S + \vartheta_S}{2\mu_S} \cdot S_{0,1}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{2\mu_S} \cdot S_{0,0}^0 - \frac{\vartheta_M}{2\mu_S} \cdot S_{1,1}^0 \\
 &= 113,4732031
 \end{aligned}$$

- Για $i = 1$ και $n = 1$:

$$\begin{aligned} S_{1,2}^0 &= \frac{\hat{\eta}_1(1) + 1 \cdot \mu_1 + \vartheta_1}{2\mu_1} \cdot S_{1,1}^0 - \frac{\hat{\eta}_1(0)}{2\mu_1} \cdot S_{1,0}^0 - \frac{\vartheta_0}{2\mu_1} \cdot S_{0,1}^0 \\ &= \frac{\hat{\eta}_N + \mu_M + \vartheta_M}{2\mu_M} \cdot S_{1,1}^0 - \frac{\hat{\eta}_N}{2\mu_M} \cdot S_{1,0}^0 - \frac{\vartheta_S}{2\mu_M} \cdot S_{0,1}^0 \\ &= -1,207125 \end{aligned}$$

- Για $i = 0$ και $n = 2$:

$$\begin{aligned} S_{0,3}^0 &= \frac{\hat{\eta}_0(2) + 2 \cdot \mu_0 + \vartheta_0}{3\mu_0} \cdot S_{0,2}^0 - \frac{\hat{\eta}_0(1)}{3\mu_0} \cdot S_{0,1}^0 - \frac{\vartheta_1}{3\mu_0} \cdot S_{1,2}^0 \\ &= \frac{\hat{\eta}_H + \hat{\eta}_N + 2\mu_S + \vartheta_S}{3\mu_S} \cdot S_{0,2}^0 - \frac{\hat{\eta}_H + \hat{\eta}_N}{3\mu_S} \cdot S_{0,1}^0 - \frac{\vartheta_M}{3\mu_S} \cdot S_{1,2}^0 \\ &= 570,0864614 \end{aligned}$$

- Για $i = 1$ και $n = 2$:

$$\begin{aligned} S_{1,3}^0 &= \frac{\hat{\eta}_1(2) + 2 \cdot \mu_1 + \vartheta_1}{3\mu_1} \cdot S_{1,2}^0 - \frac{\hat{\eta}_1(1)}{3\mu_1} \cdot S_{1,1}^0 - \frac{\vartheta_0}{3\mu_1} \cdot S_{0,2}^0 \\ &= \frac{\hat{\eta}_N + 2\mu_M + \vartheta_M}{3\mu_M} \cdot S_{1,2}^0 - \frac{\hat{\eta}_N}{3\mu_M} \cdot S_{1,1}^0 - \frac{\vartheta_S}{3\mu_M} \cdot S_{0,2}^0 \\ &= -7,571713437 \end{aligned}$$

- Για $i = 0$ και $n = 3$:

$$\begin{aligned} S_{0,4}^0 &= \frac{\hat{\eta}_0(3) + 3 \cdot \mu_0 + \vartheta_0}{4\mu_0} \cdot S_{0,3}^0 - \frac{\hat{\eta}_0(2)}{4\mu_0} \cdot S_{0,2}^0 - \frac{\vartheta_1}{4\mu_0} \cdot S_{1,3}^0 \\ &= \frac{\hat{\eta}_H + \hat{\eta}_N + 3\mu_S + \vartheta_S}{4\mu_S} \cdot S_{0,3}^0 - \frac{\hat{\eta}_H + \hat{\eta}_N}{4\mu_S} \cdot S_{0,2}^0 - \frac{\vartheta_M}{4\mu_S} \cdot S_{1,3}^0 \\ &= 2148,866812 \end{aligned}$$

- Για $i = 1$ και $n = 3$:

$$\begin{aligned} S_{1,4}^0 &= \frac{\hat{\eta}_1(3) + 3 \cdot \mu_1 + \vartheta_1}{4\mu_1} \cdot S_{1,3}^0 - \frac{\hat{\eta}_1(2)}{4\mu_1} \cdot S_{1,2}^0 - \frac{\vartheta_0}{4\mu_1} \cdot S_{0,3}^0 \\ &= \frac{\hat{\eta}_N + 3\mu_M + \vartheta_M}{4\mu_M} \cdot S_{1,3}^0 - \frac{\hat{\eta}_N}{4\mu_M} \cdot S_{1,2}^0 - \frac{\vartheta_S}{4\mu_M} \cdot S_{0,3}^0 \\ &= -32,81155776 \end{aligned}$$

- Για $i = 0$ και $n = 4$:

$$\begin{aligned} S_{0,5}^0 &= \frac{\hat{\lambda}_0(4) + 4 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,4}^0 - \frac{\hat{\lambda}_0(3)}{5\mu_0} \cdot S_{0,3}^0 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,4}^0 \\ &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + 4\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,4}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{5\mu_S} \cdot S_{0,3}^0 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,4}^0 \\ &= 6482,623453 \end{aligned}$$

- Για $i = 1$ και $n = 4$:

$$\begin{aligned} S_{1,5}^0 &= \frac{\hat{\lambda}_1(4) + 4 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,4}^0 - \frac{\hat{\lambda}_1(3)}{5\mu_1} \cdot S_{1,3}^0 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,4}^0 \\ &= \frac{\hat{\lambda}_N + 4\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,4}^0 - \frac{\hat{\lambda}_N}{5\mu_M} \cdot S_{1,3}^0 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,4}^0 \\ &= -110,1353183 \end{aligned}$$

- Για $i = 0$ και $n = 5$:

$$\begin{aligned} S_{0,6}^0 &= \frac{\hat{\lambda}_0(5) + 5 \cdot \mu_0 + \vartheta_0}{6\mu_0} \cdot S_{0,5}^0 - \frac{\hat{\lambda}_0(4)}{6\mu_0} \cdot S_{0,4}^0 - \frac{\vartheta_1}{6\mu_0} \cdot S_{1,5}^0 \\ &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + 5\mu_S + \vartheta_S}{6\mu_S} \cdot S_{0,5}^0 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{6\mu_S} \cdot S_{0,4}^0 - \frac{\vartheta_M}{6\mu_S} \cdot S_{1,5}^0 \\ &= 16305,02293 \end{aligned}$$

- Για $i = 1$ και $n = 5$:

$$\begin{aligned} S_{1,6}^0 &= \frac{\hat{\lambda}_1(5) + 5 \cdot \mu_1 + \vartheta_1}{6\mu_1} \cdot S_{1,5}^0 - \frac{\hat{\lambda}_1(4)}{6\mu_1} \cdot S_{1,4}^0 - \frac{\vartheta_0}{6\mu_1} \cdot S_{0,5}^0 \\ &= \frac{\hat{\lambda}_N + 5\mu_M + \vartheta_M}{6\mu_M} \cdot S_{1,5}^0 - \frac{\hat{\lambda}_N}{6\mu_M} \cdot S_{1,4}^0 - \frac{\vartheta_S}{6\mu_M} \cdot S_{0,5}^0 \\ &= -304,3899744 \end{aligned}$$

ενώ, ο υπολογισμός των συντελεστών $S_{i,n}^1$ γίνεται μέσω της αναδρομικής σχέσης (3.9), με αρχικές τιμές $S_{0,0}^1 = 0$, $S_{1,0}^1 = 1$ και $S_{0,y}^1 = 0$, $S_{1,y}^1 = 0$ για $y < 0$:

- Για $i = 0$ και $n = 0$:

$$S_{0,1}^1 = \frac{\hat{\lambda}_0(0) + 0 \cdot \mu_0 + \vartheta_0}{\mu_0} \cdot S_{0,0}^1 - \frac{\hat{\lambda}_0(-1)}{\mu_0} \cdot S_{0,-1}^1 - \frac{\vartheta_1}{\mu_0} \cdot S_{1,0}^1$$

$$= \frac{\hat{\eta}_H + \hat{\eta}_N + \vartheta_S}{\mu_S} \cdot S_{0,0}^1 - 0 \cdot S_{0,-1}^1 - \frac{\vartheta_M}{\mu_S} \cdot S_{1,0}^1 = -0,05$$

- Για $i = 1$ και $n = 0$:

$$\begin{aligned} S_{1,1}^1 &= \frac{\hat{\eta}_1(0) + 0 \cdot \mu_1 + \vartheta_1}{\mu_1} \cdot S_{1,0}^1 - \frac{\hat{\eta}_1(-1)}{\mu_1} \cdot S_{1,-1}^1 - \frac{\vartheta_0}{\mu_1} \cdot S_{0,0}^1 \\ &= \frac{\hat{\eta}_N + 0 \cdot \mu_M + \vartheta_M}{\mu_M} \cdot S_{1,0}^1 - 0 \cdot S_{1,-1}^1 - \frac{\vartheta_S}{\mu_M} \cdot S_{0,0}^1 = 8,08 \end{aligned}$$

- Για $i = 0$ και $n = 1$:

$$\begin{aligned} S_{0,2}^1 &= \frac{\hat{\eta}_0(1) + 1 \cdot \mu_0 + \vartheta_0}{2\mu_0} \cdot S_{0,1}^1 - \frac{\hat{\eta}_0(0)}{2\mu_0} \cdot S_{0,0}^1 - \frac{\vartheta_1}{2\mu_0} \cdot S_{1,1}^1 \\ &= \frac{\hat{\eta}_H + \hat{\eta}_N + \mu_S + \vartheta_S}{2\mu_S} \cdot S_{0,1}^1 - \frac{\hat{\eta}_H + \hat{\eta}_N}{2\mu_S} \cdot S_{0,0}^1 - \frac{\vartheta_M}{2\mu_S} \cdot S_{1,1}^1 \\ &= -0,6035625 \end{aligned}$$

- Για $i = 1$ και $n = 1$:

$$\begin{aligned} S_{1,2}^1 &= \frac{\hat{\eta}_1(1) + 1 \cdot \mu_1 + \vartheta_1}{2\mu_1} \cdot S_{1,1}^1 - \frac{\hat{\eta}_1(0)}{2\mu_1} \cdot S_{1,0}^1 - \frac{\vartheta_0}{2\mu_1} \cdot S_{0,1}^1 \\ &= \frac{\hat{\eta}_N + \mu_M + \vartheta_M}{2\mu_M} \cdot S_{1,1}^1 - \frac{\hat{\eta}_N}{2\mu_M} \cdot S_{1,0}^1 - \frac{\vartheta_S}{2\mu_M} \cdot S_{0,1}^1 \\ &= 32,6857 \end{aligned}$$

- Για $i = 0$ και $n = 2$:

$$\begin{aligned} S_{0,3}^1 &= \frac{\hat{\eta}_0(2) + 2 \cdot \mu_0 + \vartheta_0}{3\mu_0} \cdot S_{0,2}^1 - \frac{\hat{\eta}_0(1)}{3\mu_0} \cdot S_{0,1}^1 - \frac{\vartheta_1}{3\mu_0} \cdot S_{1,2}^1 \\ &= \frac{\hat{\eta}_H + \hat{\eta}_N + 2\mu_S + \vartheta_S}{3\mu_S} \cdot S_{0,2}^1 - \frac{\hat{\eta}_H + \hat{\eta}_N}{3\mu_S} \cdot S_{0,1}^1 - \frac{\vartheta_M}{3\mu_S} \cdot S_{1,2}^1 \\ &= -3,727523386 \end{aligned}$$

- Για $i = 1$ και $n = 2$:

$$S_{1,3}^1 = \frac{\hat{\eta}_1(2) + 2 \cdot \mu_1 + \vartheta_1}{3\mu_1} \cdot S_{1,2}^1 - \frac{\hat{\eta}_1(1)}{3\mu_1} \cdot S_{1,1}^1 - \frac{\vartheta_0}{3\mu_1} \cdot S_{0,2}^1$$

$$\begin{aligned}
&= \frac{\hat{\lambda}_N + 2\mu_M + \vartheta_M}{3\mu_M} \cdot S_{1,2}^1 - \frac{\hat{\lambda}_N}{3\mu_M} \cdot S_{1,1}^1 - \frac{\vartheta_S}{3\mu_M} \cdot S_{0,2}^1 \\
&= 88,29740408
\end{aligned}$$

- Για $i = 0$ και $n = 3$:

$$\begin{aligned}
S_{0,4}^1 &= \frac{\hat{\lambda}_0(3) + 3 \cdot \mu_0 + \vartheta_0}{4\mu_0} \cdot S_{0,3}^1 - \frac{\hat{\lambda}_0(2)}{4\mu_0} \cdot S_{0,2}^1 - \frac{\vartheta_1}{4\mu_0} \cdot S_{1,3}^1 \\
&= \frac{\hat{\lambda}_H + \hat{\lambda}_N + 3\mu_S + \vartheta_S}{4\mu_S} \cdot S_{0,3}^1 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{4\mu_S} \cdot S_{0,2}^1 - \frac{\vartheta_M}{4\mu_S} \cdot S_{1,3}^1 \\
&= -15,67245597
\end{aligned}$$

- Για $i = 1$ και $n = 3$:

$$\begin{aligned}
S_{1,4}^1 &= \frac{\hat{\lambda}_1(3) + 3 \cdot \mu_1 + \vartheta_1}{4\mu_1} \cdot S_{1,3}^1 - \frac{\hat{\lambda}_1(2)}{4\mu_1} \cdot S_{1,2}^1 - \frac{\vartheta_0}{4\mu_1} \cdot S_{0,3}^1 \\
&= \frac{\hat{\lambda}_N + 3\mu_M + \vartheta_M}{4\mu_M} \cdot S_{1,3}^1 - \frac{\hat{\lambda}_N}{4\mu_M} \cdot S_{1,2}^1 - \frac{\vartheta_S}{4\mu_M} \cdot S_{0,3}^1 \\
&= 179,3055974
\end{aligned}$$

- Για $i = 0$ και $n = 4$:

$$\begin{aligned}
S_{0,5}^1 &= \frac{\hat{\lambda}_0(4) + 4 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,4}^1 - \frac{\hat{\lambda}_0(3)}{5\mu_0} \cdot S_{0,3}^1 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,4}^1 \\
&= \frac{\hat{\lambda}_H + \hat{\lambda}_N + 4\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,4}^1 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{5\mu_S} \cdot S_{0,3}^1 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,4}^1 \\
&= -50,3617242
\end{aligned}$$

- Για $i = 1$ και $n = 4$:

$$\begin{aligned}
S_{1,5}^1 &= \frac{\hat{\lambda}_1(4) + 4 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,4}^1 - \frac{\hat{\lambda}_1(3)}{5\mu_1} \cdot S_{1,3}^1 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,4}^1 \\
&= \frac{\hat{\lambda}_N + 4\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,4}^1 - \frac{\hat{\lambda}_N}{5\mu_M} \cdot S_{1,3}^1 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,4}^1 \\
&= 292,2399259
\end{aligned}$$

- Για $i = 0$ και $n = 5$:

$$\begin{aligned} S_{0,6}^0 &= \frac{\hat{\lambda}_0(5) + 5 \cdot \mu_0 + \vartheta_0}{6\mu_0} \cdot S_{0,5}^1 - \frac{\hat{\lambda}_0(4)}{6\mu_0} \cdot S_{0,4}^1 - \frac{\vartheta_1}{6\mu_0} \cdot S_{1,5}^1 \\ &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + 5\mu_S + \vartheta_S}{6\mu_S} \cdot S_{0,5}^1 - \frac{\hat{\lambda}_H + \hat{\lambda}_N}{6\mu_S} \cdot S_{0,4}^1 - \frac{\vartheta_M}{6\mu_S} \cdot S_{1,5}^1 \\ &= -131,6512081 \end{aligned}$$

- Για $i = 1$ και $n = 5$:

$$\begin{aligned} S_{1,6}^1 &= \frac{\hat{\lambda}_1(5) + 5 \cdot \mu_1 + \vartheta_1}{6\mu_1} \cdot S_{1,5}^1 - \frac{\hat{\lambda}_1(4)}{6\mu_1} \cdot S_{1,4}^1 - \frac{\vartheta_0}{6\mu_1} \cdot S_{0,5}^1 \\ &= \frac{\hat{\lambda}_N + 5\mu_M + \vartheta_M}{6\mu_M} \cdot S_{1,5}^1 - \frac{\hat{\lambda}_N}{6\mu_M} \cdot S_{1,4}^1 - \frac{\vartheta_S}{6\mu_M} \cdot S_{0,5}^1 \\ &= 398,8482707 \end{aligned}$$

Τα αριθμητικά αποτελέσματα των συντελεστών $S^0(i, n)$ και $S^1(i, n)$, παρουσιάζονται στον Πίνακα 3.1. Στον Πίνακα 3.2, παρατίθενται τα αποτελέσματα των $S^0(i, n)$ και $S^1(i, n)$ του ίδιου παραδείγματος, που προκύπτουν από την υλοποίηση του αλγόριθμου NP, σε πρόγραμμα γλώσσας Python.

Πίνακας 3.1: Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 3.4

$S_{i,n}^0$		$S_{i,n}^1$	
$S_{0,0}^0 = 1$	$S_{1,0}^0 = 0$	$S_{0,0}^1 = 0$	$S_{1,0}^1 = 1$
$S_{0,1}^0 = 15,0625$	$S_{1,1}^0 = -0,1$	$S_{0,1}^1 = -0,05$	$S_{1,1}^1 = 8,08$
$S_{0,2}^0 = 113,4732031$	$S_{1,2}^0 = -1,207125$	$S_{0,2}^1 = -0,6035625$	$S_{1,2}^1 = 32,6857$
$S_{0,3}^0 = 570,0864614$	$S_{1,3}^0 = -7,571713437$	$S_{0,3}^1 = -3,727523386$	$S_{1,3}^1 = 88,29740408$
$S_{0,4}^0 = 2148,866812$	$S_{1,4}^0 = -32,81155776$	$S_{0,4}^1 = -15,67245597$	$S_{1,4}^1 = 179,3055974$
$S_{0,5}^0 = 6482,623453$	$S_{1,5}^0 = -110,1353183$	$S_{0,5}^1 = -50,3617242$	$S_{1,5}^1 = 292,2399259$
$S_{0,6}^0 = 16305,02293$	$S_{1,6}^0 = -304,3899744$	$S_{0,6}^1 = -131,6512081$	$S_{1,6}^1 = 398,8482707$

Για τον υπολογισμό των πιθανοτήτων $P(0,0)$ και $P(1,0)$, αρχικά υπολογίζουμε τον αριθμό r , μέσω της σχέσης (3.12):

$$r = -\frac{S_{0,6}^0 \cdot (6\mu_S + \vartheta_S) - S_{0,5}^0 \cdot (\hat{\lambda}_H + \hat{\lambda}_N) - S_{1,6}^0 \cdot \vartheta_M}{S_{0,6}^1 \cdot (6\mu_S + \vartheta_S) - S_{0,5}^1 \cdot (\hat{\lambda}_H + \hat{\lambda}_N) - S_{1,6}^1 \cdot \vartheta_M} = 25,93802649$$

Έτσι, μέσω των σχέσεων (3.15) και (3.13), προκύπτουν:

$$P(0,0) = \left(\sum_{i=0}^1 \sum_{n=0}^6 S_{i,n}^0 + r \cdot \sum_{i=0}^1 \sum_{n=0}^6 S_{i,n}^1 \right)^{-1} = 0,00002179190872$$

Πίνακας 3.2: Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 3.4 (Πρόγραμμα σε Python)

n	$S_{0,n}^0$	$S_{1,n}^0$	$S_{0,n}^1$	$S_{1,n}^1$
0	1	0	0	1
1	15,0625	-0,1	-0,05	8,08
2	113,4732031	-1,207125	-0,6035625	32,6857
3	570,0864615	-7,571713438	-3,727523385	88,29740408
4	2148,866813	-32,81155776	-15,67245596	179,3055974
5	6482,623454	-110,1353183	-50,36172418	292,2399259
6	16305,02294	-304,3899744	-131,651208	398,8482707

και

$$P(1, 0) = r \cdot P(0, 0) = 0,0005652391056$$

Οι υπόλοιπες πιθανότητες $P(i, n)$, υπολογίζονται μέσω της σχέσης (3.6):

$$P(0, 1) = S_{0,1}^0 \cdot P(0, 0) + S_{0,1}^1 \cdot P(1, 0) = 0,0002999786698$$

$$P(1, 1) = S_{1,1}^0 \cdot P(0, 0) + S_{1,1}^1 \cdot P(1, 0) = 0,004564952782$$

$$P(0, 2) = S_{0,2}^0 \cdot P(0, 0) + S_{0,2}^1 \cdot P(1, 0) = 0,002131640556$$

$$P(1, 2) = S_{1,2}^0 \cdot P(0, 0) + S_{1,2}^1 \cdot P(1, 0) = 0,01844893$$

$$P(0, 3) = S_{0,3}^0 \cdot P(0, 0) + S_{0,3}^1 \cdot P(1, 0) = 0,01031633$$

$$P(1, 3) = S_{1,3}^0 \cdot P(0, 0) + S_{1,3}^1 \cdot P(1, 0) = 0,049744143$$

$$P(0, 4) = S_{0,4}^0 \cdot P(0, 0) + S_{0,4}^1 \cdot P(1, 0) = 0,037969224$$

$$P(1, 4) = S_{1,4}^0 \cdot P(0, 0) + S_{1,4}^1 \cdot P(1, 0) = 0,100635509$$

$$P(0, 5) = S_{0,5}^0 \cdot P(0, 0) + S_{0,5}^1 \cdot P(1, 0) = 0,112802322$$

$$P(1, 5) = S_{1,5}^0 \cdot P(0, 0) + S_{1,5}^1 \cdot P(1, 0) = 0,162785373$$

$$P(0, 6) = S_{0,6}^0 \cdot P(0, 0) + S_{0,6}^1 \cdot P(1, 0) = 0,28090360$$

$$P(1, 6) = S_{1,6}^0 \cdot P(0, 0) + S_{1,6}^1 \cdot P(1, 0) = 0,218811401$$

Στους Πίνακες που ακολουθούν παρουσιάζονται τα αποτελέσματα των πιθανοτήτων $P(i, n)$ του Παραδείγματος 3.4, που προκύπτουν από τη μέθοδο επίλυσης που εφαρμόσαμε σε αυτή την Ενότητα (Πίνακας 3.3) αλλά και του προγράμματος σε γλώσσα Python, που προκύπτει από την υλοποίηση του αλγόριθμου NP (Πίνακας 3.4).

Πίνακας 3.3: Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 3.4

$P(i, n)$	
$P(0, 0) = 0,00002179190872$	$P(1, 0) = 0,0005652391056$
$P(0, 1) = 0,0002999786698$	$P(1, 1) = 0,004564952782$
$P(0, 2) = 0,002131640556$	$P(1, 2) = 0,01844893$
$P(0, 3) = 0,01031633$	$P(1, 3) = 0,049744143$
$P(0, 4) = 0,037969224$	$P(1, 4) = 0,100635509$
$P(0, 5) = 0,112802322$	$P(1, 5) = 0,162785375$
$P(0, 6) = 0,28090316$	$P(1, 6) = 0,218811401$

Πίνακας 3.4: Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 3.4 (Πρόγραμμα σε Python)

n	$P(0, n)$	$P(1, n)$
0	2,17919E-05	0,000565239
1	0,000299979	0,004564953
2	0,002131641	0,01844893
3	0,01031633	0,049744144
4	0,037969224	0,10063551
5	0,112802322	0,162785377
6	0,280903158	0,218811403

Από τις σχέσεις (3.16) και (3.17) υπολογίζουμε τις πιθανότητες απώλειας των νέων κλήσεων του Παραδείγματος 3.4, όταν το όχημα είναι στατικό και όταν κινείται αντίστοιχα:

$$B_S^N = P(0, 6) = 0,28090316$$

και

$$B_S^M = P(1, 6) = 0,218811401$$

Συνεπώς, η συνολική πιθανότητα απώλειας των νέων κλήσεων του Παραδείγματος 3.4, είναι (από τη σχέση (3.18)):

$$P_{NCBP} = B_S^N + B_M^N = 0,499714561$$

Για τον υπολογισμό της πιθανότητας απόρριψης των κλήσεων που προέρχονται από μεταπομπή, χρησιμοποιούμε τη σχέση (3.19) και προκύπτει:

$$P_{HCDP} = P(n = 6/i = 0) = \frac{P(0, 6)}{\sum_{n=0}^6 P(0, n)} = 0,632032106$$

Τέλος, για τον προσδιορισμό του ποσοστού χρήσης των καναλιών, χρησιμοποιούμε τη σχέση (3.20):

$$U = \frac{\sum_{n=0}^6 n \cdot (P(0, n) + (P(1, n)))}{6} = 0,859475379$$

Οι τιμές των πιθανοτήτων απώλειας των νέων κλήσεων (P_{NCBP}) και απόρριψης των κλήσεων μεταπομπής (P_{HCDP}), αλλά και του ποσοστού χρήσης των καναλιών (Utilization) του Παραδείγματος 3.4, οι οποίες προέκυψαν από τη μεθοδολογία που εφαρμόστηκε στην παρούσα Ενότητα, συγκεντρώνονται στον Πίνακα 3.5.

Πίνακας 3.5: P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 3.4

P_{NCBP}	0,499714561
P_{HCDP}	0,632032106
<i>Utilization</i>	0,859475379

Ο Πίνακας 3.6, αναφέρεται στις τιμές των πιθανοτήτων απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής όπως επίσης και στο ποσοστό χρήσης των καναλιών του προγράμματος στο οποίο έγινε η υλοποίηση του αλγόριθμου NP, σε γλώσσα Python.

Πίνακας 3.6: P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 3.4 (Πρόγραμμα σε Python)

P_{NCBP}	0,49971456
P_{HCDP}	0,632032105
<i>Utilization</i>	0,859475381

Κεφάλαιο 4

Μοντέλο Απωλειών Cutoff Priority

4.1 Εισαγωγή

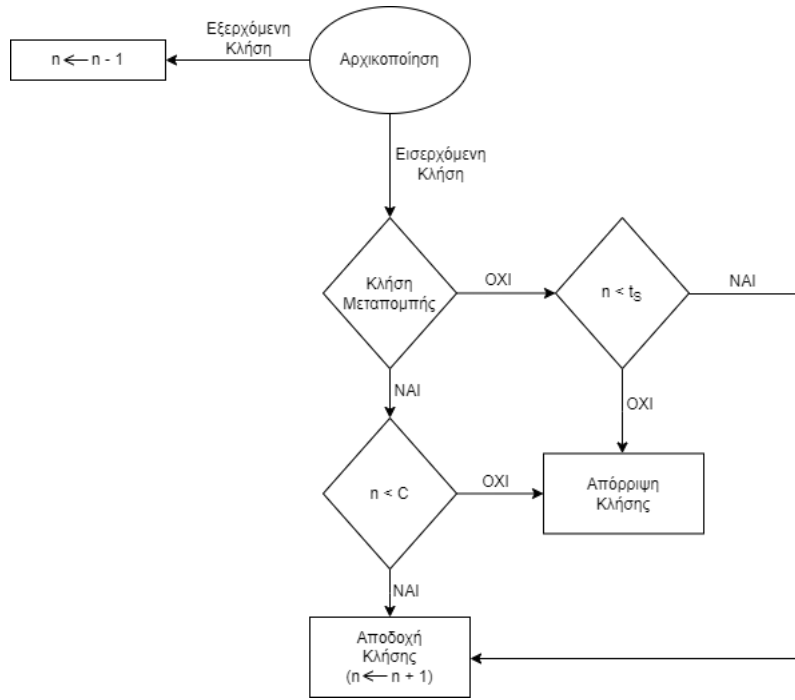
Σε αυτό το Κεφάλαιο παρουσιάζεται το τέταρτο και τελευταίο σύστημα αποδοχής κλήσεων της παρούσας εργασίας, το μοντέλο απωλειών Cutoff Priority (CP). Πρόκειται για ένα σύστημα το οποίο διατηρεί δεσμευμένο έναν σταθερό αριθμό καναλιών για την εξυπηρέτηση αποκλειστικά κλήσεων μεταπομπής, τόσο κατά τη διάρκεια της φάσης στάσης του οχήματος όσο και κατά τη διάρκεια της φάσης στην οποία το όχημα κινείται.

Στις Ενότητες του Κεφαλαίου που ακολουθούν, γίνεται μια πλήρης περιγραφή του μοντέλου CP. Συγκεκριμένα, στην Ενότητα 4.2 παρουσιάζεται ο αλγόριθμος CP. Στην Ενότητα 4.3 γίνεται η λεπτομερής ανάλυση του μοντέλου CP και παρατίθενται οι τύποι υπολογισμού των πιθανοτήτων απόρριψης των κλήσεων μεταπομπής και απώλειας των νέων κλήσεων, καθώς και ο τύπος υπολογισμού του ποσοστού χρήσης των καναλιών του συστήματος. Σε ότι αφορά τον προσδιορισμό των πιθανοτήτων των καταστάσεων μετάβασης του συστήματος CP, γίνεται χρήση της ίδιας μεθοδολογίας που εφαρμόστηκε και στα μοντέλα των Κεφαλαίων 1, 2 και 3. Τέλος, στην Ενότητα 4.4 επιλύεται ένα αριθμητικό παράδειγμα του μοντέλου CP.

4.2 Ο Αλγόριθμος CP

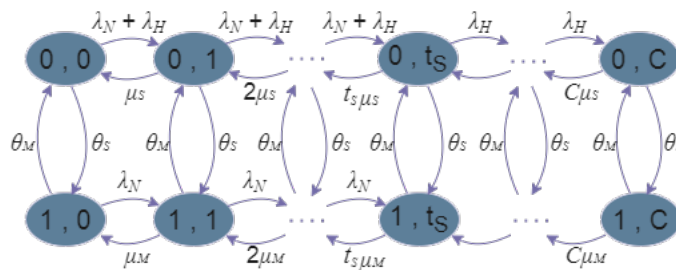
Ο αλγόριθμος του συστήματος CP (Σχήμα 4.1) λειτουργεί όπως ο αλγόριθμος MA-CAC χωρίς ουρά μεταπομπής στη φάση, κατά τη διάρκεια της οποίας, το όχημα είναι στατικό. Πιο συγκεκριμένα, μόλις μια κλήση εισέλθει στο σύστημα ελέγχεται αν είναι νέα κλήση ή αν είναι κλήση μεταπομπής. Αν έχουμε νέα κλήση, εξετάζεται αν ο αριθμός των ενεργών κλήσεων του συστήματος είναι μικρότερος του αριθμού των δεσμευμένων υπέρ των κλήσεων μεταπομπής καναλιών, προκειμένου η κλήση να γίνει αποδεκτή αλλιώς η κλήση απορρίπτεται.

Στην περίπτωση που η εισερχόμενη στο σύστημα κλήση προέρχεται από μεταπομπή, ο αλγόριθμος CP ελέγχει αν ο αριθμός των κλήσεων που ήδη εξυπηρετούνται από το σημείο πρόσβασης του δικτύου WLAN είναι μικρότερος της χωρητικότητας του συστήματος. Αν είναι, τότε η κλήση γίνεται αποδεκτή, διαφορετικά απορρίπτεται. Οι φάσεις στάσης και κίνησης του οχήματος, δεν επηρεάζουν τον τρόπο λειτουργίας του αλγορίθμου CP.



Σχήμα 4.1: Διάγραμμα Ροής του Αλγορίθμου CP.

4.3 Ανάλυση του μοντέλου



Σχήμα 4.2: Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου του Μοντέλου Απωλειών CP.

Στο μοντέλο αποδοχής κλήσεων Cutoff Priority, όπως και στα μοντέλα MA-CAC είτε με ουρά μεταπομπής είτε άνευ ουράς μεταπομπής και στο μοντέλο Non Priority, οι διαδικασίες άφιξης των κλήσεων ακολουθούν κατανομή Poisson, με ρυθμούς λ_H για τις κλήσεις μεταπομπής και λ_N για τις νέες κλήσεις που αφικνούνται στο σύστημα. Όπως αναφέρθηκε στην Εισαγωγή του παρόντος Κεφαλαίου, βασικό χαρακτηριστικό του μοντέλου απωλειών CP είναι το γεγονός ότι διατηρείται ένας σταθερός αριθμός κατειλημμένων καναλιών και στις δύο φάσεις κίνησης του οχήματος, προκειμένου να εξυπηρετούν αποκλειστικά κλήσεις μεταπομπής. Επομένως στη φάση κατά την οποία το όχημα είναι στατικό, ο συνολικός ρυθμός άφιξης των κλήσεων στο σύστημα έχει τιμή $\lambda_H + \lambda_N$, όσο ο αριθμός των ενεργών κλήσεων είναι μικρότερος του αριθμού των προστατευόμενων καναλιών, ενώ στη φάση κίνησης το σύστημα εξυπηρετεί λιγότερες κλήσεις, από τη στιγμή που δεν μπορούν να πραγματοποιηθούν κλήσεις μεταπομπής όσο το όχημα κινείται (Σχήμα 4.2).

Ο μέσος χρόνος παραμονής των κλήσεων ακολουθεί εκθετική κατανομή και στις δύο φάσεις κινητικότητας του οχήματος, με ρυθμούς μ_s για τη φάση στάσης και μ_M για

τη φάση κατά την οποία το όχημα κινείται ενώ τα ποσοστά εξυπηρέτησης των κλήσεων από το σύστημα έχουν τιμές $n \cdot \mu_S$ και $n \cdot \mu_M$ για τις φάσεις στάσης και κίνησης αντίστοιχα.

Οι χρόνοι παραμονής στη φάση στάσης $1/\vartheta_S$ και στη φάση κίνησης $1/\vartheta_M$ ακολουθούν εκθετική κατανομή. Να σημειωθεί ότι κάθε κλήση στο μοντέλο CP χρησιμοποιεί 1 b.u. από τους πόρους του συστήματος για την εξυπηρέτησή της.

Τέλος, σε ότι αφορά τον ρυθμό μετάβασης από μία κατάσταση του συστήματος σε μία άλλη, έχουμε τις ακόλουθες σχέσεις [1]:

$$\begin{aligned}
 r(0, n; 1, n) &= \vartheta_S, & \text{για } 0 \leq n \leq C \\
 r(1, n; 0, n) &= \vartheta_M, & \text{για } 0 \leq n \leq C \\
 r(0, n; 0, n-1) &= n \cdot \mu_S, & \text{για } 0 < n \leq C \\
 r(1, n; 1, n-1) &= n \cdot \mu_M, & \text{για } 0 < n \leq C \\
 r(0, n; 0, n+1) &= \hat{\lambda}_N + \hat{\lambda}_H, & \text{για } 0 \leq n < t_S \\
 r(0, n; 0, n+1) &= \hat{\lambda}_H, & \text{για } t_S \leq n < C \\
 r(1, n; 1, n+1) &= \hat{\lambda}_N, & \text{για } 0 \leq n < t_S
 \end{aligned} \tag{4.1}$$

Στο Σχήμα 4.2, παρουσιάζεται η Μαρκοβιανή αλυσίδα συνεχούς χρόνου του μοντέλου απωλειών Cutoff Priority. Οι εξισώσεις ισορροπίας, που προκύπτουν από την αλυσίδα Μαρκοβ, για τον προσδιορισμό των πιθανοτήτων των καταστάσεων $P(i, n)$ είναι οι εξής [1]:

1) Για $i = 0$ και $n = 0$ έχουμε:

$$((\hat{\lambda}_N + \hat{\lambda}_H) + \vartheta_S) P(i, n) = (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

2) Για $i = 0$ και $0 < n < t_S$ έχουμε:

$$((\hat{\lambda}_N + \hat{\lambda}_H) + n\mu_S + \vartheta_S) P(i, n) = (\hat{\lambda}_N + \hat{\lambda}_H) P(i, n-1) + (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

3) Για $i = 0$ και $n = t_S$ έχουμε:

$$(\hat{\lambda}_H + n\mu_S + \vartheta_S) P(i, n) = (\hat{\lambda}_N + \hat{\lambda}_H) P(i, n-1) + (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

4) Για $i = 0$ και $t_S < n < C$ έχουμε:

$$(\hat{\lambda}_H + n\mu_S + \vartheta_S) P(i, n) = \hat{\lambda}_H P(i, n-1) + (n+1) \mu_S P(i, n+1) + \vartheta_M P(1, n)$$

5) Για $i = 0$ και $n = C$ έχουμε:

$$(C\mu_S + \vartheta_S) P(i, n) = \hat{\lambda}_H P(i, n-1) + \vartheta_M P(1, n)$$

6) Για $i = 1$ και $n = 0$ έχουμε:

$$(\hat{\lambda}_N + \vartheta_M) P(i, n) = (n+1) \mu_M P(i, n+1) + \vartheta_S P(0, n)$$

7) Για $i = 1$ και $0 < n < t_S$ έχουμε:

$$(\hat{\lambda}_N + n\mu_M + \vartheta_M)P(i, n) = \hat{\lambda}_N P(i, n-1) + (n+1)\mu_M P(i, n+1) + \vartheta_S P(0, n)$$

8) Για $i = 1$ και $n = t_S$ έχουμε:

$$(n\mu_M + \vartheta_M)P(i, n) = \hat{\lambda}_N P(i, n-1) + (n+1)\mu_M P(i, n+1) + \vartheta_S P(0, n)$$

9) Για $i = 1$ και $t_S < n < C$ έχουμε:

$$(n\mu_M + \vartheta_M)P(i, n) = (n+1)\mu_M P(i, n+1) + \vartheta_S P(0, n)$$

10) Για $i = 1$ και $n = C$ έχουμε:

$$(C\mu_M + \vartheta_M)P(i, n) = \vartheta_S P(0, n) \quad (4.2)$$

Ένας πιο συνοπτικός τρόπος για την έκφραση των εξισώσεων (4.2) είναι μέσω των εξισώσεων που ακολουθούν:

- Για τις οριακές καταστάσεις $(0, 0)$ και $(1, 0)$:

$$(\hat{\lambda}_i(0) + \vartheta_i)P(i, 0) - \mu_i P(i, 1) - \vartheta_j P(j, 0) = 0 \quad (4.3)$$

- Για τις ενδιάμεσες καταστάσεις $(1, 0), (2, 0), \dots, (C-1, 0)$ και $(1, 1), (2, 1), \dots, (C-1, 1)$:

$$(\hat{\lambda}_i(n) + n\mu_i + \vartheta_i)P(i, n) - \hat{\lambda}_i(n-1)P(i, n-1) - \vartheta_j P(j, n) - (n+1)\mu_i P(i, n+1) = 0 \quad (4.4)$$

- Για τις οριακές καταστάσεις $(0, C)$ και $(1, C)$:

$$(C\mu_i + \vartheta_i)P(i, C) - \lambda_i(C-1)P(i, C-1) - \vartheta_j P(j, C) = 0 \quad (4.5)$$

όπου:

$$i = \begin{cases} 0, & \text{για τη φάση στάσης} \\ 1, & \text{για τη φάση κίνησης} \end{cases} \quad \mu_i = \begin{cases} \mu_S, & \text{όταν } i = 0 \\ \mu_M, & \text{όταν } i = 1 \end{cases}$$

$$\vartheta_i = \begin{cases} \vartheta_S, & \text{όταν } i = 0 \\ \vartheta_M, & \text{όταν } i = 1 \end{cases} \quad \hat{\lambda}_i(n) = \begin{cases} \hat{\lambda}_N + \hat{\lambda}_H, & \text{όταν } i = 0 \text{ και } 0 \leq n < t_S \\ \hat{\lambda}_H, & \text{όταν } i = 0 \text{ και } t_S \leq n < C \\ \hat{\lambda}_N, & \text{όταν } i = 1 \text{ και } 0 \leq n < t_S \\ 0, & \text{όταν } i = 1 \text{ και } t_S \leq n < C \\ 0, & \text{όταν } n < 0 \end{cases}$$

και $j = 1 - i$

Οι τιμές των πιθανοτήτων $P(i, n)$ μπορούν να προσδιοριστούν μέσω των πιθανοτήτων των οριακών καταστάσεων $P(0, 0)$ και $P(1, 0)$, μέσω της σχέσης:

$$P(i, n) = S_{i,n}^0 \cdot P(0, 0) + S_{i,n}^1 \cdot P(1, 0) \quad (4.6)$$

Για τον υπολογισμό των τιμών των συντελεστών $S_{i,n}^0$ και $S_{i,n}^1$ της σχέσης (4.6), ακολουθούμε την εξής διαδικασία:

Αρχικά, αντικαθιστούμε τη σχέση (4.6) στην εξίσωση (4.4) των ενδιάμεσων καταστάσεων του συστήματος και προκύπτει η σχέση:

$$\begin{aligned} S_{i,n+1}^0 \cdot P(0,0) + S_{i,n+1}^1 \cdot P(1,0) &= \frac{\hat{\eta}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot (S_{i,n}^0 \cdot P(0,0) + S_{i,n}^1 \cdot P(1,0)) \\ &\quad - \frac{\hat{\eta}_i(n-1)}{(n+1)\mu_i} \cdot (S_{i,n-1}^0 \cdot P(0,0) + S_{i,n-1}^1 \cdot P(1,0)) \\ &\quad - \frac{\vartheta_j}{(n+1)\mu_i} \cdot (S_{j,n}^0 \cdot P(0,0) + S_{j,n}^1 \cdot P(1,0)) \quad (4.7) \end{aligned}$$

Στη συνέχεια, υποθέτουμε ότι στην τελευταία σχέση (Σχέση (4.7)) οι τιμές των πιθανοτήτων των οριακών καταστάσεων είναι $P(0,0) = 1$ και $P(1,0) = 0$ και προκύπτει ο ακόλουθος τύπος, μέσω του οποίου υπολογίζονται οι συντελεστές $S_{i,n}^0$:

$$S_{i,n+1}^0 = \frac{\hat{\eta}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot S_{i,n}^0 - \frac{\hat{\eta}_i(n-1)}{(n+1)\mu_i} \cdot S_{i,n-1}^0 - \frac{\vartheta_j}{(n+1)\mu_i} \cdot S_{j,n}^0 \quad (4.8)$$

όπου: $S_{0,0}^0 = 1$, $S_{1,0}^0 = 0$ και $S_{0,y}^0 = 0$, $S_{1,y}^0 = 0$ για $y < 0$

Τέλος, για να προκύψει ο τύπος υπολογισμού των συντελεστών $S_{i,n}^1$, υποθέτουμε στη σχέση (4.7) ότι $P(0,0) = 0$ και $P(1,0) = 1$:

$$S_{i,n+1}^1 = \frac{\hat{\eta}_i(n) + n\mu_i + \vartheta_i}{(n+1)\mu_i} \cdot S_{i,n}^1 - \frac{\hat{\eta}_i(n-1)}{(n+1)\mu_i} \cdot S_{i,n-1}^1 - \frac{\vartheta_j}{(n+1)\mu_i} \cdot S_{j,n}^1 \quad (4.9)$$

όπου: $S_{0,0}^1 = 0$, $S_{1,0}^1 = 1$ και $S_{0,y}^1 = 0$, $S_{1,y}^1 = 0$ για $y < 0$

Σε αυτό το στάδιο πρέπει να καθοριστούν οι τιμές των πιθανοτήτων $P(0,0)$ και $P(1,0)$, ώστε να μπορέσουμε μέσω της σχέσης (4.6) να υπολογίσουμε όλες τις πιθανότητες $P(i,n)$.

Για τον προσδιορισμό των $P(0,0)$ και $P(1,0)$ θεωρούμε στην εξίσωση (4.5) των οριακών καταστάσεων $(0,C)$ και $(1,C)$ του συστήματος, την περίπτωση όπου $i = 0$ και προκύπτει:

$$(C\mu_S + \vartheta_S)P(0,C) - \hat{\eta}_0(C-1)P(0,C-1) - \vartheta_M P(1,C) = 0 \quad (4.10)$$

(Αν στην εξίσωση (4.5) θεωρούσαμε την περίπτωση $i = 1$, θα προέκυπτε το ίδιο αποτέλεσμα)

Συνδυάζοντας τις εξισώσεις (4.10) και (4.6), έχουμε :

$$\begin{aligned}
& (C\mu_S + \vartheta_S) \cdot (S_{0,C}^0 \cdot P(0,0) + S_{0,C}^1 \cdot P(1,0)) \\
& - \hat{\lambda}_0 (C-1) \cdot (S_{0,C-1}^0 \cdot P(0,0) + S_{0,C-1}^1 \cdot P(1,0)) \\
& - \vartheta_M \cdot (S_{1,C}^0 \cdot P(0,0) + S_{1,C}^1 \cdot P(1,0)) \\
& = 0 \Rightarrow \\
& P(0,0) \cdot (S_{0,C}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^0 \cdot \hat{\lambda}_0 - S_{1,C}^0 \cdot \vartheta_M) \\
& + P(1,0) \cdot (S_{0,C}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^1 \cdot \hat{\lambda}_0 - S_{1,C}^1 \cdot \vartheta_M) \\
& = 0 \Rightarrow \\
& \frac{P(1,0)}{P(0,0)} = - \frac{S_{0,C}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^0 \cdot \hat{\lambda}_0 - S_{1,C}^0 \cdot \vartheta_M}{S_{0,C}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^1 \cdot \hat{\lambda}_0 - S_{1,C}^1 \cdot \vartheta_M} \tag{4.11}
\end{aligned}$$

Θέτουμε στη σχέση (4.11):

$$r = - \frac{S_{0,C}^0 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^0 \cdot \hat{\lambda}_0 - S_{1,C}^0 \cdot \vartheta_M}{S_{0,C}^1 \cdot (C\mu_S + \vartheta_S) - S_{0,C-1}^1 \cdot \hat{\lambda}_0 - S_{1,C}^1 \cdot \vartheta_M} \tag{4.12}$$

και προκύπτει η σχέση :

$$\frac{P(1,0)}{P(0,0)} = r \Rightarrow P(1,0) = r \cdot P(0,0) \tag{4.13}$$

Από τη Θεωρία Πιθανοτήτων γνωρίζουμε ότι το άθροισμα όλων των πιθανοτήτων $P(i, n)$ του συστήματος ισούται με τη μονάδα. Επομένως :

$$\sum_{i=0}^1 \sum_{n=0}^C P(i, n) = 1 \tag{4.14}$$

Άρα, η εξίσωση (4.14) μέσω της σχέσης (4.6) γίνεται :

$$\begin{aligned}
& P(0,0) \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^0 + P(1,0) \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^1 = 1 \\
& \Rightarrow \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^0 + \frac{P(1,0) \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^1}{P(0,0)} = \frac{1}{P(0,0)} \\
& \Rightarrow P(0,0) = \left(\sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^0 + r \cdot \sum_{i=0}^1 \sum_{n=0}^C S_{i,n}^1 \right)^{-1} \tag{4.15}
\end{aligned}$$

Σε αυτή τη φάση, έχοντας πλέον καθορίσει τις τιμές όλων των πιθανοτήτων $P(i, n)$, έχουμε τη δυνατότητα να υπολογίσουμε τις πιθανότητες απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής, καθώς και το ποσοστό χρήσης των καναλιών του συστήματος.

Για να υπολογίσουμε την πιθανότητα απώλειας των νέων κλήσεων του μοντέλου CP, λαμβάνουμε υπόψη μας το γεγονός ότι μία νέα κλήση απορρίπτεται από το σύστημα όταν το άθροισμα των ενεργών κλήσεων ξεπεράσει τον αριθμό των καναλιών που είναι διαθέσιμα, τόσο για τις νέες κλήσεις όσο και για τις κλήσεις μεταπομπής και αυτό συμβαίνει και στις δύο φάσεις κινητικότητας του οχήματος. Επομένως, έχουμε τις ακόλουθες σχέσεις:

- Για την φάση στάσης:

$$B_S^N = \sum_{n=t_S}^C P(0, n) \quad (4.16)$$

- Για την φάση κίνησης:

$$B_M^N = \sum_{n=t_S}^C P(1, n) \quad (4.17)$$

Οπότε, η συνολική πιθανότητα απώλειας των νέων κλήσεων του συστήματος (P_{NCBP}), προκύπτει από τη σχέση [1]:

$$P_{NCBP} = B_S^N + B_M^N = \sum_{n=t_S}^C (P(0, n) + P(1, n)) \quad (4.18)$$

Για το υπολογισμό της πιθανότητας απόρριψης των κλήσεων μεταπομπής του συστήματος (P_{HCDP}), λαμβάνουμε υπόψη το γεγονός ότι κλήσεις από μεταπομπή πραγματοποιούνται μόνο όσο το όχημα είναι στατικό και μέχρι να καλυφθεί η χωρητικότητα του συστήματος. Επομένως έχουμε [1]:

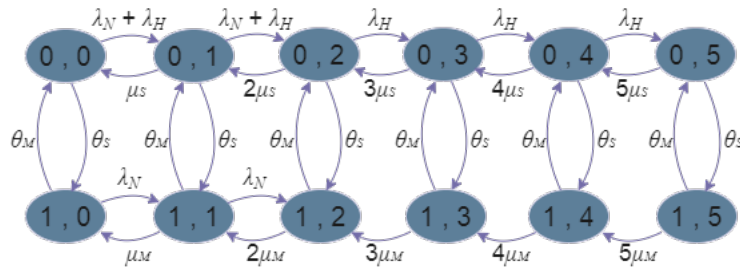
$$P_{HCDP} = P(n = C/i = 0) = \frac{P(0, C)}{\sum_{n=0}^C P(0, n)} \quad (4.19)$$

Τέλος, το ποσοστό χρήσης των καναλιών στο σύστημα CP δίνεται από τον τύπο [1]:

$$U = \frac{\sum_{n=0}^C n \cdot (P(0, n) + P(1, n))}{C} \quad (4.20)$$

4.4 Παράδειγμα

Στην παρούσα Ενότητα παρουσιάζεται ένα αριθμητικό παράδειγμα του μοντέλου απωλειών Cutoff Priority. Η χωρητικότητα του συστήματος στο συγκεκριμένο παράδειγμα είναι $C = 5$ ενώ τα κοινά κανάλια, που χρησιμοποιούνται τόσο από κλήσεις μεταπομπής όσο και από νέες κλήσεις, είναι $t_S = 2$. Οι ρυθμοί άφιξης των κλήσεων μεταπομπής και των νέων κλήσεων είναι $\lambda_H = 3$ και $\lambda_N = 1$, αντίστοιχα. Ο ρυθμός του μέσου χρόνου παραμονής των κλήσεων έχει τιμή $\mu_S = 0,45$ για τη φάση στάσης και $\mu_M = 0,3$ για τη



Σχήμα 4.3: Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου για το Μοντέλο Απωλειών CP του Παραδείγματος 4.4.

φάση κατά την οποία το όχημα κινείται. Τέλος, ο ρυθμός του μέσου χρόνου παραμονής στη φάση στάσης είναι $\vartheta_S = 0,5$ ενώ στη φάση κίνησης έχει τιμή $\vartheta_M = 0,4$.

Το Παράδειγμα 4.4, επιλύεται με τη χρήση της μεθοδολογίας που παρουσιάστηκε στην Ενότητα 4.3 και τα αποτελέσματά του συγκεντρώνονται σε πίνακες που συναντώνται στις σελίδες που ακολουθούν. Επιπλέον, σε άλλους πίνακες της Ενότητας 4.4, παρουσιάζονται τα αποτελέσματα του ίδιου παραδείγματος που προκύπτουν από την υλοποίηση του αλγορίθμου CP σε πρόγραμμα γλώσσας Python.

Οι εξισώσεις ισορροπίας για τις πιθανότητες $P(i, n)$, για το Σχήμα 4.3, προκύπτουν από τις σχέσεις (4.3), (4.4) και (4.5):

- Για την κατάσταση (0, 0):

$$\begin{aligned} \mu_S \cdot P(0, 1) + \vartheta_M \cdot P(1, 0) &= (\lambda_H + \lambda_N + \vartheta_S) \cdot P(0, 0) \\ \Rightarrow 0,45 \cdot P(0, 1) + 0,4 \cdot P(1, 0) &= 4,5 \cdot P(0, 0) \\ \Rightarrow 0,45 \cdot P(0, 1) + 0,4 \cdot P(1, 0) - 4,5 \cdot P(0, 0) &= 0 \end{aligned} \quad (1)$$

- Για την κατάσταση (1, 0):

$$\begin{aligned} \mu_M \cdot P(1, 1) + \vartheta_S \cdot P(0, 0) &= (\lambda_N + \vartheta_M) \cdot P(1, 0) \\ \Rightarrow 0,3 \cdot P(1, 1) + 0,5 \cdot P(0, 0) &= 1,4 \cdot P(1, 0) \\ \Rightarrow 0,3 \cdot P(1, 1) + 0,5 \cdot P(0, 0) - 1,4 \cdot P(1, 0) &= 0 \end{aligned} \quad (2)$$

- Για την κατάσταση (0, 1):

$$\begin{aligned} (\lambda_H + \lambda_N) \cdot P(0, 0) + 2\mu_S \cdot P(0, 2) + \vartheta_M \cdot P(1, 1) &= (\lambda_H + \lambda_N + \mu_S + \vartheta_S) \cdot P(0, 1) \\ \Rightarrow 4 \cdot P(0, 0) + 0,9 \cdot P(0, 2) + 0,4 \cdot P(1, 1) &= 4,95 \cdot P(0, 1) \\ \Rightarrow 4 \cdot P(0, 0) + 0,9 \cdot P(0, 2) + 0,4 \cdot P(1, 1) - 4,95 \cdot P(0, 1) &= 0 \end{aligned} \quad (3)$$

- Για την κατάσταση (1, 1):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 0) + 2\mu_M \cdot P(1, 2) + \vartheta_S \cdot P(0, 1) &= (\hat{\lambda}_N + \mu_M + \vartheta_M) \cdot P(1, 1) \\ \Rightarrow 1 \cdot P(1, 0) + 0,6 \cdot P(1, 2) + 0,5 \cdot P(0, 1) &= 1,7 \cdot P(1, 1) \\ \Rightarrow 1 \cdot P(1, 0) + 0,6 \cdot P(1, 2) + 0,5 \cdot P(0, 1) - 1,7 \cdot P(1, 1) &= 0 \end{aligned} \quad (4)$$

- Για την κατάσταση (0, 2):

$$\begin{aligned} (\hat{\lambda}_H + \hat{\lambda}_N) \cdot P(0, 1) + 3\mu_S \cdot P(0, 3) + \vartheta_M \cdot P(1, 2) &= (\hat{\lambda}_H + 2\mu_S + \vartheta_S) \cdot P(0, 2) \\ \Rightarrow 4 \cdot P(0, 1) + 1,35 \cdot P(0, 3) + 0,4 \cdot P(1, 2) &= 4,4 \cdot P(0, 2) \\ \Rightarrow 4 \cdot P(0, 1) + 1,35 \cdot P(0, 3) + 0,4 \cdot P(1, 2) - 4,4 \cdot P(0, 2) &= 0 \end{aligned} \quad (5)$$

- Για την κατάσταση (1, 2):

$$\begin{aligned} \hat{\lambda}_N \cdot P(1, 1) + 3\mu_M \cdot P(1, 3) + \vartheta_S \cdot P(0, 2) &= (2\mu_M + \vartheta_M) \cdot P(1, 2) \\ \Rightarrow 1 \cdot P(1, 1) + 0,9 \cdot P(1, 3) + 0,5 \cdot P(0, 2) &= 1 \cdot P(1, 2) \\ \Rightarrow 1 \cdot P(1, 1) + 0,9 \cdot P(1, 3) + 0,5 \cdot P(0, 2) - 1 \cdot P(1, 2) &= 0 \end{aligned} \quad (6)$$

- Για την κατάσταση (0, 3):

$$\begin{aligned} \hat{\lambda}_H \cdot P(0, 2) + 4\mu_S \cdot P(0, 4) + \vartheta_M \cdot P(1, 3) &= (\hat{\lambda}_H + 3\mu_S + \vartheta_S) \cdot P(0, 3) \\ \Rightarrow 3 \cdot P(0, 2) + 1,8 \cdot P(0, 4) + 0,4 \cdot P(1, 3) &= 4,85 \cdot P(0, 3) \\ \Rightarrow 3 \cdot P(0, 2) + 1,8 \cdot P(0, 4) + 0,4 \cdot P(1, 3) - 4,85 \cdot P(0, 3) &= 0 \end{aligned} \quad (7)$$

- Για την κατάσταση (1, 3):

$$\begin{aligned} 4\mu_M \cdot P(1, 4) + \vartheta_S \cdot P(0, 3) &= (3\mu_M + \vartheta_M) \cdot P(1, 3) \\ \Rightarrow 1,2 \cdot P(1, 4) + 0,5 \cdot P(0, 3) &= 1,3 \cdot P(1, 3) \\ \Rightarrow 1,2 \cdot P(1, 4) + 0,5 \cdot P(0, 3) - 1,3 \cdot P(1, 3) &= 0 \end{aligned} \quad (8)$$

- Για την κατάσταση (0, 4):

$$\begin{aligned} \hat{\lambda}_H \cdot P(0, 3) + 5\mu_S \cdot P(0, 5) + \vartheta_M \cdot P(1, 4) &= (\hat{\lambda}_H + 4\mu_S + \vartheta_S) \cdot P(0, 4) \\ \Rightarrow 3 \cdot P(0, 3) + 2,25 \cdot P(0, 5) + 0,4 \cdot P(1, 4) &= 5,3 \cdot P(0, 4) \\ \Rightarrow 3 \cdot P(0, 3) + 2,25 \cdot P(0, 5) + 0,4 \cdot P(1, 4) - 5,3 \cdot P(0, 4) &= 0 \end{aligned} \quad (9)$$

- Για την κατάσταση (1, 4):

$$\begin{aligned} 5\mu_M \cdot P(1, 5) + \vartheta_S \cdot P(0, 4) &= (4\mu_M + \vartheta_M) \cdot P(1, 4) \\ \Rightarrow 1,5 \cdot P(1, 5) + 0,5 \cdot P(0, 4) &= 1,6 \cdot P(1, 4) \\ \Rightarrow 1,5 \cdot P(1, 5) + 0,5 \cdot P(0, 4) - 1,6 \cdot P(1, 4) &= 0 \end{aligned} \quad (10)$$

- Για την κατάσταση (0, 5):

$$\begin{aligned} \hat{\lambda}_H \cdot P(0, 4) + \vartheta_M \cdot P(1, 5) &= (5\mu_S + \vartheta_S) \cdot P(0, 5) \\ \Rightarrow 3 \cdot P(0, 4) + 0,4 \cdot P(1, 5) &= 2,75 \cdot P(0, 5) \\ \Rightarrow 3 \cdot P(0, 4) + 0,4 \cdot P(1, 5) - 2,75 \cdot P(0, 5) &= 0 \end{aligned} \quad (11)$$

- Για την κατάσταση (1, 5):

$$\begin{aligned} \vartheta_S \cdot P(0, 5) &= (5\mu_M + \vartheta_M) \cdot P(1, 5) \\ \Rightarrow 0,5 \cdot P(0, 5) &= 1,9 \cdot P(1, 5) \\ \Rightarrow 0,5 \cdot P(0, 5) - 1,9 \cdot P(1, 5) &= 0 \end{aligned} \quad (12)$$

Μέσω της σχέσης (4.8), με αρχικές τιμές $S_{0,0}^0 = 1$, $S_{1,0}^0 = 0$ και $S_{0,y}^0 = 0$, $S_{1,y}^0 = 0$ για $y < 0$, υπολογίζουμε τους συντελεστές $S_{i,n}^0$:

- Για $i = 0$ και $n = 0$:

$$\begin{aligned} S_{0,1}^0 &= \frac{\hat{\lambda}_0(0) + 0 \cdot \mu_0 + \vartheta_0}{\mu_0} \cdot S_{0,0}^0 - \frac{\hat{\lambda}_0(-1)}{\mu_0} \cdot S_{0,-1}^0 - \frac{\vartheta_1}{\mu_0} \cdot S_{1,0}^0 \\ &= \frac{\hat{\lambda}_H + \hat{\lambda}_N + \vartheta_S}{\mu_S} \cdot S_{0,0}^0 = 10 \end{aligned}$$

- Για $i = 1$ και $n = 0$:

$$\begin{aligned} S_{1,1}^0 &= \frac{\hat{\eta}_1(0) + 0 \cdot \mu_1 + \vartheta_1}{\mu_1} \cdot S_{1,0}^0 - \frac{\hat{\eta}_1(-1)}{\mu_1} \cdot S_{1,-1}^0 - \frac{\vartheta_0}{\mu_1} \cdot S_{0,0}^0 \\ &= -\frac{\vartheta_S}{\mu_M} \cdot S_{0,0}^0 = -1,666666667 \end{aligned}$$

- Για $i = 0$ και $n = 1$:

$$\begin{aligned} S_{0,2}^0 &= \frac{\hat{\eta}_0(1) + 1 \cdot \mu_0 + \vartheta_0}{2\mu_0} \cdot S_{0,1}^0 - \frac{\hat{\eta}_0(0)}{2\mu_0} \cdot S_{0,0}^0 - \frac{\vartheta_1}{2\mu_0} \cdot S_{1,1}^0 \\ &= \frac{\hat{\eta}_H + \hat{\eta}_N + \mu_S + \vartheta_S}{2\mu_S} \cdot S_{0,1}^0 - \frac{\hat{\eta}_H + \hat{\eta}_N}{2m_S} \cdot S_{0,0}^0 - \frac{\vartheta_M}{2\mu_S} \cdot S_{1,1}^0 \\ &= 51,2962963 \end{aligned}$$

- Για $i = 1$ και $n = 1$:

$$\begin{aligned} S_{1,2}^0 &= \frac{\hat{\eta}_1(1) + 1 \cdot \mu_1 + \vartheta_1}{2\mu_1} \cdot S_{1,1}^0 - \frac{\hat{\eta}_1(0)}{2\mu_1} \cdot S_{1,0}^0 - \frac{\vartheta_0}{2\mu_1} \cdot S_{0,1}^0 \\ &= \frac{\hat{\eta}_N + \mu_M + \vartheta_M}{2\mu_M} \cdot S_{1,1}^0 - \frac{\hat{\eta}_N}{2\mu_M} \cdot S_{1,0}^0 - \frac{\vartheta_S}{2\mu_M} \cdot S_{0,1}^0 \\ &= -13,05555556 \end{aligned}$$

- Για $i = 0$ και $n = 2$:

$$\begin{aligned} S_{0,3}^0 &= \frac{\hat{\eta}_0(2) + 2 \cdot \mu_0 + \vartheta_0}{3\mu_0} \cdot S_{0,2}^0 - \frac{\hat{\eta}_0(1)}{3\mu_0} \cdot S_{0,1}^0 - \frac{\vartheta_1}{3\mu_0} \cdot S_{1,2}^0 \\ &= \frac{\hat{\eta}_H + 2\mu_S + \vartheta_S}{3\mu_S} \cdot S_{0,2}^0 - \frac{\hat{\eta}_H + \hat{\eta}_N}{3\mu_S} \cdot S_{0,1}^0 - \frac{\vartheta_M}{3\mu_S} \cdot S_{1,2}^0 \\ &= 141,4266118 \end{aligned}$$

- Για $i = 1$ και $n = 2$:

$$\begin{aligned} S_{1,3}^0 &= \frac{\hat{\eta}_1(2) + 2 \cdot \mu_1 + \vartheta_1}{3\mu_1} \cdot S_{1,2}^0 - \frac{\hat{\eta}_1(1)}{3\mu_1} \cdot S_{1,1}^0 - \frac{\vartheta_0}{3\mu_1} \cdot S_{0,2}^0 \\ &= \frac{0 + 2\mu_M + \vartheta_M}{3\mu_M} \cdot S_{1,2}^0 - \frac{\hat{\eta}_N}{3\mu_M} \cdot S_{1,1}^0 - \frac{\vartheta_S}{3\mu_M} \cdot S_{0,2}^0 \\ &= -41,15226338 \end{aligned}$$

- Για $i = 0$ και $n = 3$:

$$\begin{aligned} S_{0,4}^0 &= \frac{\hat{\lambda}_0(3) + 3 \cdot \mu_0 + \vartheta_0}{4\mu_0} \cdot S_{0,3}^0 - \frac{\hat{\lambda}_0(2)}{4\mu_0} \cdot S_{0,2}^0 - \frac{\vartheta_1}{4\mu_0} \cdot S_{1,3}^0 \\ &= \frac{\hat{\lambda}_H + 3\mu_S + \vartheta_S}{4\mu_S} \cdot S_{0,3}^0 - \frac{\hat{\lambda}_H}{4\mu_S} \cdot S_{0,2}^0 - \frac{\vartheta_M}{4\mu_S} \cdot S_{1,3}^0 \\ &= 304,7172687 \end{aligned}$$

- Για $i = 1$ και $n = 3$:

$$\begin{aligned} S_{1,4}^0 &= \frac{\hat{\lambda}_1(3) + 3 \cdot \mu_1 + \vartheta_1}{4\mu_1} \cdot S_{1,3}^0 - \frac{\hat{\lambda}_1(2)}{4\mu_1} \cdot S_{1,2}^0 - \frac{\vartheta_0}{4\mu_1} \cdot S_{0,3}^0 \\ &= \frac{0 + 3\mu_M + \vartheta_M}{4\mu_M} \cdot S_{1,3}^0 - \frac{0}{4\mu_M} \cdot S_{1,2}^0 - \frac{\vartheta_S}{4\mu_M} \cdot S_{0,3}^0 \\ &= -103,5093736 \end{aligned}$$

- Για $i = 0$ και $n = 4$:

$$\begin{aligned} S_{0,5}^0 &= \frac{\hat{\lambda}_0(4) + 4 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,4}^0 - \frac{\hat{\lambda}_0(3)}{5\mu_0} \cdot S_{0,3}^0 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,4}^0 \\ &= \frac{\hat{\lambda}_H + 4\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,4}^0 - \frac{\hat{\lambda}_H}{5\mu_S} \cdot S_{0,3}^0 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,4}^0 \\ &= 547,6113059 \end{aligned}$$

- Για $i = 1$ και $n = 4$:

$$\begin{aligned} S_{1,5}^0 &= \frac{\hat{\lambda}_1(4) + 4 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,4}^0 - \frac{\hat{\lambda}_1(3)}{5\mu_1} \cdot S_{1,3}^0 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,4}^0 \\ &= \frac{0 + 4\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,4}^0 - \frac{0}{5\mu_M} \cdot S_{1,3}^0 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,4}^0 \\ &= -211,9824214 \end{aligned}$$

Μέσω της σχέσης (4.9), με αρχικές τιμές $S_{0,0}^1 = 0$, $S_{1,0}^1 = 1$ και $S_{0,y}^1 = 0$, $S_{1,y}^1 = 0$ για $y < 0$, υπολογίζουμε τους συντελεστές $S_{i,n}^1$:

- Για $i = 0$ και $n = 0$:

$$S_{0,1}^1 = \frac{\hat{\lambda}_0(0) + 0 \cdot \mu_0 + \vartheta_0}{\mu_0} \cdot S_{0,0}^1 - \frac{\hat{\lambda}_0(-1)}{\mu_0} \cdot S_{0,-1}^1 - \frac{\vartheta_1}{\mu_0} \cdot S_{1,0}^1$$

$$= \frac{\hat{\eta}_H + \hat{\eta}_N + \vartheta_S}{\mu_S} \cdot S_{0,0}^1 - 0 \cdot S_{0,-1}^1 - \frac{\vartheta_M}{\mu_S} \cdot S_{1,0}^1 = -0,888888888$$

- Για $i = 1$ και $n = 0$:

$$\begin{aligned} S_{1,1}^1 &= \frac{\hat{\eta}_1(0) + 0 \cdot \mu_1 + \vartheta_1}{\mu_1} \cdot S_{1,0}^1 - \frac{\hat{\eta}_1(-1)}{\mu_1} \cdot S_{1,-1}^1 - \frac{\vartheta_0}{\mu_1} \cdot S_{0,0}^1 \\ &= \frac{\hat{\eta}_N + 0 \cdot \mu_M + \vartheta_M}{\mu_M} \cdot S_{1,0}^1 - 0 \cdot S_{1,-1}^1 - \frac{\vartheta_S}{\mu_M} \cdot S_{0,0}^1 = 4,666666667 \end{aligned}$$

- Για $i = 0$ και $n = 1$:

$$\begin{aligned} S_{0,2}^1 &= \frac{\hat{\eta}_0(1) + 1 \cdot \mu_0 + \vartheta_0}{2\mu_0} \cdot S_{0,1}^1 - \frac{\hat{\eta}_0(0)}{2\mu_0} \cdot S_{0,0}^1 - \frac{\vartheta_1}{2\mu_0} \cdot S_{1,1}^1 \\ &= \frac{\hat{\eta}_H + \hat{\eta}_N + \mu_S + \vartheta_S}{2\mu_S} \cdot S_{0,1}^1 - \frac{\hat{\eta}_H + \hat{\eta}_N}{2\mu_S} \cdot S_{0,0}^1 - \frac{\vartheta_M}{2\mu_S} \cdot S_{1,1}^1 \\ &= -6,962962958 \end{aligned}$$

- Για $i = 1$ και $n = 1$:

$$\begin{aligned} S_{1,2}^1 &= \frac{\hat{\eta}_1(1) + 1 \cdot \mu_1 + \vartheta_1}{2\mu_1} \cdot S_{1,1}^1 - \frac{\hat{\eta}_1(0)}{2\mu_1} \cdot S_{1,0}^1 - \frac{\vartheta_0}{2\mu_1} \cdot S_{0,1}^1 \\ &= \frac{\hat{\eta}_N + \mu_M + \vartheta_M}{2\mu_M} \cdot S_{1,1}^1 - \frac{\hat{\eta}_N}{2\mu_M} \cdot S_{1,0}^1 - \frac{\vartheta_S}{2\mu_M} \cdot S_{0,1}^1 \\ &= 12,29629629 \end{aligned}$$

- Για $i = 0$ και $n = 2$:

$$\begin{aligned} S_{0,3}^1 &= \frac{\hat{\eta}_0(2) + 2 \cdot \mu_0 + \vartheta_0}{3\mu_0} \cdot S_{0,2}^1 - \frac{\hat{\eta}_0(1)}{3\mu_0} \cdot S_{0,1}^1 - \frac{\vartheta_1}{3\mu_0} \cdot S_{1,2}^1 \\ &= \frac{\hat{\eta}_H + 2\mu_S + \vartheta_S}{3\mu_S} \cdot S_{0,2}^1 - \frac{\hat{\eta}_H + \hat{\eta}_N}{3\mu_S} \cdot S_{0,1}^1 - \frac{\vartheta_M}{3\mu_S} \cdot S_{1,2}^1 \\ &= -23,70370369 \end{aligned}$$

- Για $i = 1$ και $n = 2$:

$$S_{1,3}^1 = \frac{\hat{\eta}_1(2) + 2 \cdot \mu_1 + \vartheta_1}{3\mu_1} \cdot S_{1,2}^1 - \frac{\hat{\eta}_1(1)}{3\mu_1} \cdot S_{1,1}^1 - \frac{\vartheta_0}{3\mu_1} \cdot S_{0,2}^1$$

$$\begin{aligned}
&= \frac{0 + 2\mu_M + \vartheta_M}{3\mu_M} \cdot S_{1,2}^1 - \frac{\hat{\eta}_N}{3\mu_M} \cdot S_{1,1}^1 - \frac{\vartheta_S}{3\mu_M} \cdot S_{0,2}^1 \\
&= 12,345679
\end{aligned}$$

- Για $i = 0$ και $n = 3$:

$$\begin{aligned}
S_{0,4}^1 &= \frac{\hat{\eta}_0(3) + 3 \cdot \mu_0 + \vartheta_0}{4\mu_0} \cdot S_{0,3}^1 - \frac{\hat{\eta}_0(2)}{4\mu_0} \cdot S_{0,2}^1 - \frac{\vartheta_1}{4\mu_0} \cdot S_{1,3}^1 \\
&= \frac{\hat{\eta}_H + 3\mu_S + \vartheta_S}{4\mu_S} \cdot S_{0,3}^1 - \frac{\hat{\eta}_H}{4\mu_S} \cdot S_{0,2}^1 - \frac{\vartheta_M}{4\mu_S} \cdot S_{1,3}^1 \\
&= -55,00685868
\end{aligned}$$

- Για $i = 1$ και $n = 3$:

$$\begin{aligned}
S_{1,4}^1 &= \frac{\hat{\eta}_1(3) + 3 \cdot \mu_1 + \vartheta_1}{4\mu_1} \cdot S_{1,3}^1 - \frac{\hat{\eta}_1(2)}{4\mu_1} \cdot S_{1,2}^1 - \frac{\vartheta_0}{4\mu_1} \cdot S_{0,3}^1 \\
&= \frac{0 + 3\mu_M + \vartheta_M}{4\mu_M} \cdot S_{1,3}^1 - \frac{0}{4\mu_M} \cdot S_{1,2}^1 - \frac{\vartheta_S}{4\mu_M} \cdot S_{0,3}^1 \\
&= 23,25102878
\end{aligned}$$

- Για $i = 0$ και $n = 4$:

$$\begin{aligned}
S_{0,5}^1 &= \frac{\hat{\eta}_0(4) + 4 \cdot \mu_0 + \vartheta_0}{5\mu_0} \cdot S_{0,4}^1 - \frac{\hat{\eta}_0(3)}{5\mu_0} \cdot S_{0,3}^1 - \frac{\vartheta_1}{5\mu_0} \cdot S_{1,4}^1 \\
&= \frac{\hat{\eta}_H + 4\mu_S + \vartheta_S}{5\mu_S} \cdot S_{0,4}^1 - \frac{\hat{\eta}_H}{5\mu_S} \cdot S_{0,3}^1 - \frac{\vartheta_M}{5\mu_S} \cdot S_{1,4}^1 \\
&= -102,1002896
\end{aligned}$$

- Για $i = 1$ και $n = 4$:

$$\begin{aligned}
S_{1,5}^1 &= \frac{\hat{\eta}_1(4) + 4 \cdot \mu_1 + \vartheta_1}{5\mu_1} \cdot S_{1,4}^1 - \frac{\hat{\eta}_1(3)}{5\mu_1} \cdot S_{1,3}^1 - \frac{\vartheta_0}{5\mu_1} \cdot S_{0,4}^1 \\
&= \frac{0 + 4\mu_M + \vartheta_M}{5\mu_M} \cdot S_{1,4}^1 - \frac{0}{5\mu_M} \cdot S_{1,3}^1 - \frac{\vartheta_S}{5\mu_M} \cdot S_{0,4}^1 \\
&= 43,13671693
\end{aligned}$$

Τα αποτελέσματα των συντελεστών $S^0(i, n)$ και $S^1(i, n)$ του παραδείγματος, συγκεντρώνονται στους παρακάτω πίνακες:

Πίνακας 4.1: Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 4.4

$S_{i,n}^0$		$S_{i,n}^1$	
$S_{0,0}^0 = 1$	$S_{1,0}^0 = 0$	$S_{0,0}^1 = 0$	$S_{1,0}^1 = 1$
$S_{0,1}^0 = 10$	$S_{1,1}^0 = -1,666666667$	$S_{0,1}^1 = -0,888888888$	$S_{1,1}^1 = 4,666666667$
$S_{0,2}^0 = 51,2962963$	$S_{1,2}^0 = -13,05555556$	$S_{0,2}^1 = -6,962962958$	$S_{1,2}^1 = 12,29629629$
$S_{0,3}^0 = 141,4266118$	$S_{1,3}^0 = -41,15226338$	$S_{0,3}^1 = -23,70370369$	$S_{1,3}^1 = 12,345679$
$S_{0,4}^0 = 304,7172687$	$S_{1,4}^0 = -103,5093736$	$S_{0,4}^1 = -55,00685868$	$S_{1,4}^1 = 23,25102878$
$S_{0,5}^0 = 547,6113059$	$S_{1,5}^0 = -211,9824214$	$S_{0,5}^1 = -102,1002896$	$S_{1,5}^1 = 43,13671693$

Πίνακας 4.2: Αριθμητικά Αποτελέσματα των $S_{i,n}^0$ και $S_{i,n}^1$ του Παραδείγματος 4.4 (Πρόγραμμα σε Python)

n	$S_{0,n}^0$	$S_{1,n}^0$	$S_{0,n}^1$	$S_{1,n}^1$
0	1	0	0	1
1	10	-1,666666667	-0,888888889	4,666666667
2	51,2962963	-13,05555556	-6,962962963	12,2962963
3	141,4266118	-41,15226337	-23,7037037	12,34567901
4	304,7172687	-103,5093736	-55,00685871	23,25102881
5	547,6113059	-211,9824214	-102,1002896	43,13671696

Για τον υπολογισμό των πιθανοτήτων $P(0, 0)$ και $P(1, 0)$, υπολογίζουμε πρώτα τον αριθμό r , μέσω της σχέσης (4.12):

$$r = -\frac{S_{0,5}^0 \cdot (5\mu_S + \vartheta_S) - S_{0,4}^0 \cdot \lambda_H - S_{1,5}^0 \cdot \vartheta_M}{S_{0,5}^1 \cdot (5\mu_S + \vartheta_S) - S_{0,4}^1 \cdot \lambda_H - S_{1,5}^1 \cdot \vartheta_M} = 5,086630527$$

Έτσι, μέσω της σχέσης (4.15), η τιμή της πιθανότητας $P(0, 0)$ είναι:

$$P(0, 0) = \left(\sum_{i=0}^1 \sum_{n=0}^5 S_{i,n}^0 + r \cdot \sum_{i=0}^1 \sum_{n=0}^5 S_{i,n}^1 \right)^{-1} = 0,00461070586$$

ενώ, μέσω της σχέσης (4.13), η πιθανότητα $P(1, 0)$ έχει τιμή:

$$P(1, 0) = r \cdot P(0, 0) = 0,0234529572$$

Σε αυτό το στάδιο, έχοντας υπολογίσει τις τιμές των πιθανοτήτων $P(0, 0)$ και $P(1, 0)$, χρησιμοποιούμε την αναδρομική σχέση (4.6) ώστε να υπολογίσουμε τις τιμές και των υπόλοιπων πιθανοτήτων $P(i, n)$ του παραδείγματος. Έτσι, προκύπτουν τα ακόλουθα αποτελέσματα:

$$P(0, 1) = S_{0,1}^0 \cdot P(0, 0) + S_{0,1}^1 \cdot P(1, 0) = 0,0252599856$$

$$P(1, 1) = S_{1,1}^0 \cdot P(0, 0) + S_{1,1}^1 \cdot P(1, 0) = 0,101762624$$

$$P(0, 2) = S_{0,2}^0 \cdot P(0, 0) + S_{0,2}^1 \cdot P(1, 0) = 0,0732100617$$

$$P(1, 2) = S_{1,2}^0 \cdot P(0, 0) + S_{1,2}^1 \cdot P(1, 0) = 0,228189184$$

$$P(0, 3) = S_{0,3}^0 \cdot P(0, 0) + S_{0,3}^1 \cdot P(1, 0) = 0,0961545597$$

$$P(1, 3) = S_{1,3}^0 \cdot P(0, 0) + S_{1,3}^1 \cdot P(1, 0) = 0,0998016993$$

$$P(0, 4) = S_{0,4}^0 \cdot P(0, 0) + S_{0,4}^1 \cdot P(1, 0) = 0,114888194$$

$$P(1, 4) = S_{1,4}^0 \cdot P(0, 0) + S_{1,4}^1 \cdot P(1, 0) = 0,0680541074$$

$$P(0, 5) = S_{0,5}^0 \cdot P(0, 0) + S_{0,5}^1 \cdot P(1, 0) = 0,130320935$$

$$P(1, 5) = S_{1,5}^0 \cdot P(0, 0) + S_{1,5}^1 \cdot P(1, 0) = 0,0342949833$$

Τα αποτελέσματα των πιθανοτήτων των καταστάσεων $P(i, n)$ του παραδείγματος, συγκριτώνται στους πίνακες που ακολουθούν:

Πίνακας 4.3: Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 4.4

$P(i, n)$	
$P(0, 0) = 0,00461070586$	$P(1, 0) = 0,0234529572$
$P(0, 1) = 0,0252599856$	$P(1, 1) = 0,101762624$
$P(0, 2) = 0,0732100617$	$P(1, 2) = 0,228189184$
$P(0, 3) = 0,0961545597$	$P(1, 3) = 0,0998016993$
$P(0, 4) = 0,114888194$	$P(1, 4) = 0,0680541074$
$P(0, 5) = 0,130320935$	$P(1, 5) = 0,0342949833$

Μέσω της σχέσης (4.18), υπολογίζουμε την πιθανότητα απώλειας των νέων κλήσεων του παραδείγματος:

$$P_{NCBP} = \sum_{n=2}^5 (P(0, n) + P(1, n)) = 0,844913724$$

Πίνακας 4.4: Αριθμητικά Αποτελέσματα των $P(i, n)$ του Παραδείγματος 4.4 (Πρόγραμμα σε Python)

n	$P(0, n)$	$P(1, n)$
0	0,004610706	0,023452957
1	0,025259986	0,101762624
2	0,073210062	0,228189184
3	0,09615456	0,099801699
4	0,114888194	0,068054108
5	0,130320937	0,034294983

Η πιθανότητα απόρριψης των κλήσεων που προέρχονται από μεταπομπή του παραδείγματος, υπολογίζονται μέσω της σχέσης (4.19):

$$P_{HCDP} = P(n = 5/i = 0) = \frac{P(0, 5)}{\sum_{n=0}^5 P(0, n)} = 0,293222105$$

Τέλος, το ποσοστό χρήσης των καναλιών προκύπτει από τη σχέση (4.20):

$$U = \frac{\sum_{n=0}^5 n \cdot (P(0, n) + P(1, n))}{5} = 0,574507735$$

Τα αποτελέσματα των πιθανοτήτων απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής, καθώς και το ποσοστό χρήσης των καναλιών του Παραδείγματος 4.4, συγκεντρώνονται στους κάτωθι πίνακες:

Πίνακας 4.5: P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 4.4

P_{NCBP}	0,844913724
P_{HCDP}	0,293222105
<i>Utilization</i>	0,574507735

Πίνακας 4.6: P_{NCBP} , P_{HCDP} και Ποσοστό Χρήσης Καναλιών του Παραδείγματος 4.4 (Πρόγραμμα σε Python)

P_{NCBP}	0,844913728
P_{HCDP}	0,293222108
<i>Utilization</i>	0,574507738

Κεφάλαιο 5

Επιδράσεις των Παραμέτρων στα Μοντέλα Απωλειών/Ουρών Κλήσεων

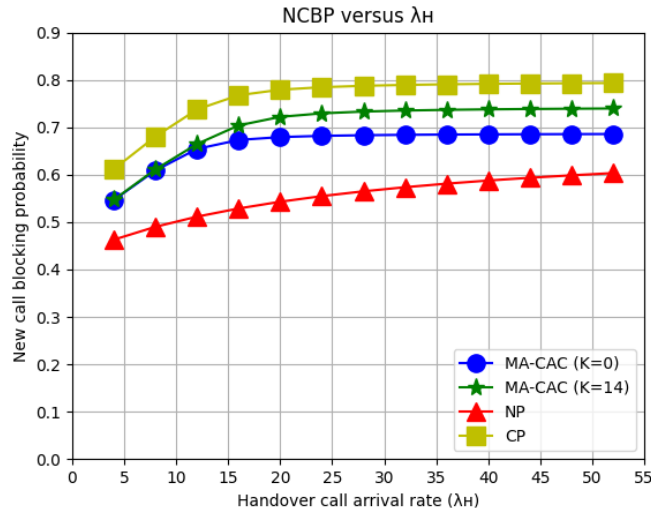
5.1 Εισαγωγή

Σε αυτό το Κεφάλαιο μελετώνται οι επιδράσεις κάποιων από τις παραμέτρους των συστημάτων αποδοχής κλήσεων, στις πιθανότητες απώλειας των νέων κλήσεων, στις πιθανότητες απόρριψης των κλήσεων μεταπομπής, στο ποσοστό χρήσης των καναλιών και στον χρόνο παραμονής των κλήσεων μεταπομπής στην ουρά αναμονής. Πιο συγκεκριμένα, στις Ενότητες 5.2 και 5.3 εξετάζονται οι επιδράσεις των ρυθμών άφιξης λ_H των κλήσεων μεταπομπής και λ_N των νέων κλήσεων αντίστοιχα, στην Ενότητα 5.4 μελετώνται οι επιδράσεις του αριθμού t_S των καναλιών που διατίθενται προς χρήση τόσο από κλήσεις μεταπομπής όσο και από νέες κλήσεις και στην Ενότητα 5.5 ερευνάται η επιρροή του μεγέθους της ουράς K . Τέλος, στις Ενότητες 5.6 και 5.7 εξετάζονται οι επιδράσεις των μέσων χρόνων παραμονής $1/\theta_S$ και $1/\theta_M$ στις φάσεις στάσης και κίνησης αντίστοιχα.

Οι παράμετροι, με εξαίρεση την Ενότητα στην οποία η καθεμία από αυτές μεταβάλλεται, έχουν σταθερές τιμές. Για την ακρίβεια, ο ρυθμός άφιξης των κλήσεων μεταπομπής έχει τιμή $\lambda_H = 28$, ο ρυθμός άφιξης των νέων κλήσεων είναι $\lambda_N = 24$ και ο αριθμός των κοινών καναλιών των νέων κλήσεων και των κλήσεων μεταπομπής είναι $t_S = 56$. Ο αριθμός των θέσεων της ουράς μεταπομπής έχει την τιμή $K = 0$ για το μοντέλο MA-CAC χωρίς ουρά μεταπομπής και $K = 14$ για το μοντέλο MA-CAC με ουρά μεταπομπής ενώ στην Ενότητα 5.6 εξετάζεται και η περίπτωση του μοντέλου με μέγεθος ουράς $K = 6$. Όσον αφορά τους χρόνους παραμονής στην φάση στάσης και στη φάση κίνησης του οχήματος, οι ρυθμοί τους έχουν τιμές $\theta_S = 1/4$ και $\theta_M = 1/5$ αντίστοιχα. Επιπλέον, η χωρητικότητα των συστημάτων για όλες τις περιπτώσεις καθορίζεται στην τιμή $C = 70$.

5.2 Επιδράσεις του λ_H

Σε αυτή την Ενότητα θα δούμε πως ο ρυθμός άφιξης των κλήσεων μεταπομπής επηρεάζει τις πιθανότητες απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής, καθώς και το ποσοστό χρήσης των καναλιών των συστημάτων MA-CAC με ουρά μεταπομπής, MA-CAC χωρίς ουρά μεταπομπής, Cutoff Priority και Non Priority.



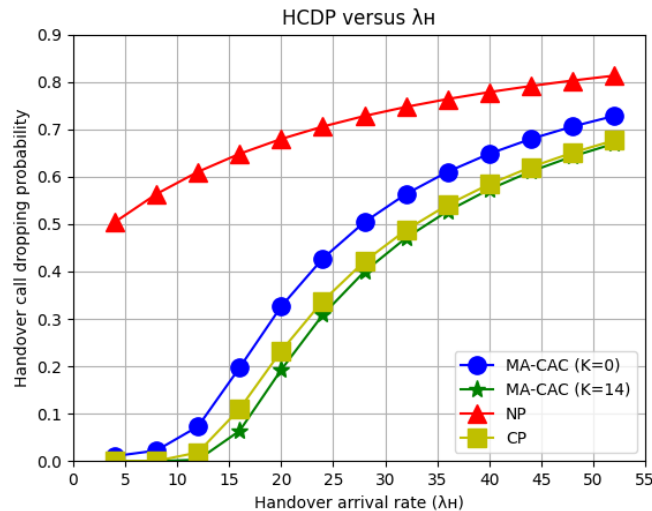
Σχήμα 5.1: Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσει του λ_H .

Στο Σχήμα 5.1 παρουσιάζεται η πιθανότητα απώλειας των νέων κλήσεων (P_{NCBP}) των μοντέλων απωλειών/ουρών κλήσεων, ως συνάρτηση του ρυθμού άφιξης των κλήσεων μεταπομπής (λ_H). Όσο ο ρυθμός λ_H αυξάνεται, οι μικρότερες τιμές της πιθανότητας απώλειας των νέων κλήσεων εμφανίζονται στο μοντέλο NP, συγκριτικά με τα υπόλοιπα μοντέλα απωλειών. Αυτό οφείλεται στο γεγονός ότι τα μοντέλα MA-CAC, με ουρά ή χωρίς ουρά καθώς και το μοντέλο CP, διατηρούν έναν σταθερό αριθμό καναλιών που εξυπηρετούν αποκλειστικά κλήσεις μεταπομπής, ενώ στο μοντέλο NP όλα τα κανάλια του συστήματος είναι διαθέσιμα και για τα δύο είδη κλήσεων (νέες κλήσεις/κλήσεις μεταπομπής) κατά τη διάρκεια και των δύο φάσεων κινητικότητας (φάση στάσης/φάση κίνησης) του οχήματος.

Σε ότι αφορά τα υπόλοιπα μοντέλα παρατηρούμε ότι οι υψηλότερες τιμές της πιθανότητας P_{NCBP} , καθώς το λ_H αυξάνεται, εμφανίζονται στο σύστημα CP. Το γεγονός αυτό οφείλεται στην αρχιτεκτονική του μοντέλου CP σύμφωνα με την οποία το σύστημα διατηρεί προστατευόμενα κανάλια υπέρ των κλήσεων μεταπομπής, τόσο κατά τη διάρκεια της φάσης στην οποία το όχημα είναι στατικό όσο και κατά τη διάρκεια της φάσης κίνησης. Αντιθέτως, τα μοντέλα MA-CAC με ουρά μεταπομπής και MA-CAC χωρίς ουρά μεταπομπής, διατηρούν δεσμευμένα κανάλια για χρήση από κλήσεις μεταπομπής μόνο κατά τη διάρκεια της φάσης στάσης του οχήματος. Επομένως στη φάση κίνησης του συστήματος CP υπάρχουν λιγότερα κανάλια διαθέσιμα για τις νέες κλήσεις, σε σχέση με τα άλλα μοντέλα και έτσι η πιθανότητα απώλειας των νέων κλήσεων αυξάνεται.

Μια τελευταία παρατήρηση για το Σχήμα 5.1, που προκύπτει από τη σύγκριση των μοντέλων απωλειών MA-CAC μεταξύ τους, είναι ότι η πιθανότητα P_{NCBP} του συστήματος που είναι εφοδιασμένο με την ουρά μεταπομπής είναι υψηλότερη σε σχέση με το

σύστημα χωρίς ουρά, καθώς ο ρυθμός άφιξης των κλήσεων μεταπομπής αυξάνεται. Αυτό συμβαίνει λόγω της ύπαρξης της ουράς μεταπομπής στο ένα από τα δύο μοντέλα, η οποία τοποθετεί σε αναμονή έναν αριθμό κλήσεων που προέρχονται από μεταπομπή, προκειμένου να εξυπηρετηθούν μόλις απελευθερωθεί κάποιο κανάλι. Συνέπεια αυτού του γεγονότος είναι η αύξηση του αριθμού των αποσχολημένων καναλιών από κλήσεις μεταπομπής του συστήματος με την ουρά, σε σχέση με το σύστημα χωρίς ουρά.



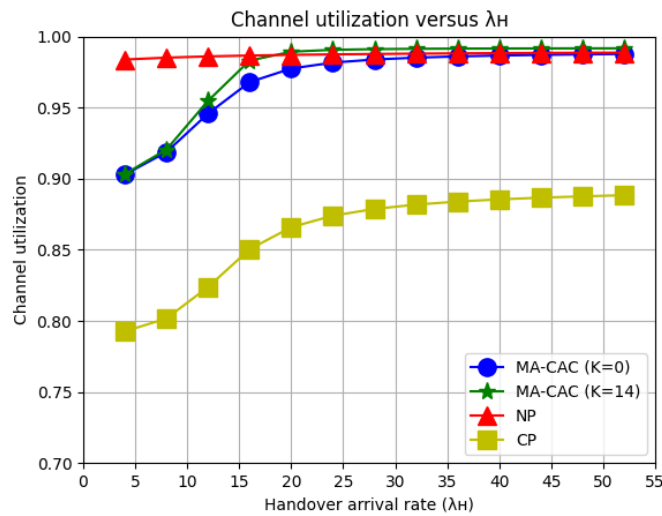
Σχήμα 5.2: Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσει του λ_H

Στο Σχήμα 5.2 παρουσιάζεται η πιθανότητα απόρριψης των κλήσεων μεταπομπής (P_{HCDP}), ως συνάρτηση του ρυθμού λ_H , για τα μοντέλα αποδοχής κλήσεων της παρούσας εργασίας.

Για το σύστημα NP παρατηρούμε ότι, καθώς ο ρυθμός άφιξης των κλήσεων μεταπομπής αυξάνεται, οι τιμές των πιθανοτήτων P_{HCDP} είναι μεγαλύτερες από τις αντίστοιχες τιμές των υπόλοιπων μοντέλων. Αυτό συμβαίνει εξαιτίας της ύπαρξης του δεσμευμένου αριθμού καναλιών υπέρ των κλήσεων μεταπομπής, τόσο στα μοντέλα MA-CAC (είτε με ουρά είτε άνευ ουράς μεταπομπής) όσο και στο σύστημα CP. Έτσι στα συστήματα που διατηρούν δεσμευμένα κανάλια υπέρ των κλήσεων μεταπομπής, εξυπηρετείται πολύ μεγαλύτερος αριθμός κλήσεων μεταπομπής απ' ό,τι στο μοντέλο NP, στο οποίο όλα τα κανάλια διατίθενται για όλα τα είδη κλήσεων, χωρίς καμία διάκριση.

Μία επιπλέον παρατήρηση που προκύπτει από το Σχήμα 5.2 αφορά τα μοντέλα MA-CAC με ουρά μεταπομπής και MA-CAC χωρίς ουρά μεταπομπής όπου φαίνεται ότι η πιθανότητα απόρριψης των κλήσεων μεταπομπής του συστήματος που είναι εφοδιασμένο με την ουρά, είναι μικρότερη από την αντίστοιχη πιθανότητα του μοντέλου άνευ ουράς, για πολλές τιμές του ρυθμού λ_H . Εδώ είναι ξεκάθαρη η επίδραση της ουράς μεταπομπής στις τιμές του P_{HCDP} , καθώς στο σύστημα χωρίς την ουρά οι κλήσεις μεταπομπής που δεν μπορούν να εξυπηρετηθούν άμεσα απορρίπτονται ενώ στο σύστημα MA-CAC με ΗQ οι κλήσεις μεταπομπής που δεν μπορούν να εξυπηρετηθούν εισέρχονται στην ουρά, μέχρι να απελευθερωθεί κάποιο από τα απασχολημένα κανάλια και να αποσταλούν σε αυτό. Παρ' όλα αυτά παρατηρούμε ότι όταν οι τιμές του λ_H ξεπεράσουν το 48, οι καμπύλες των δύο μοντέλων αρχίζουν να συγκλίνουν. Αυτό συμβαίνει διότι οι, εισερχόμενες στο σύστημα, κλήσεις μεταπομπής είναι πλέον αριθμητικά πολλές για το

συγκεκριμένο μέγεθος της ουράς (14 θέσεις) κι έτσι δεν μπορούν να φιλοξενηθούν από αυτή και κατά συνέπεια απορρίπτονται.



Σχήμα 5.3: Ποσοστό Χρήσης των Καναλιών συναρτήσει του λ_H

Τέλος, στο Σχήμα 5.3 παρουσιάζεται το ποσοστό χρήσης των καναλιών (channel utilization) του κάθε μοντέλου, συναρτήσει του ρυθμού άφιξης των κλήσεων που προέρχονται από μεταπομπή.

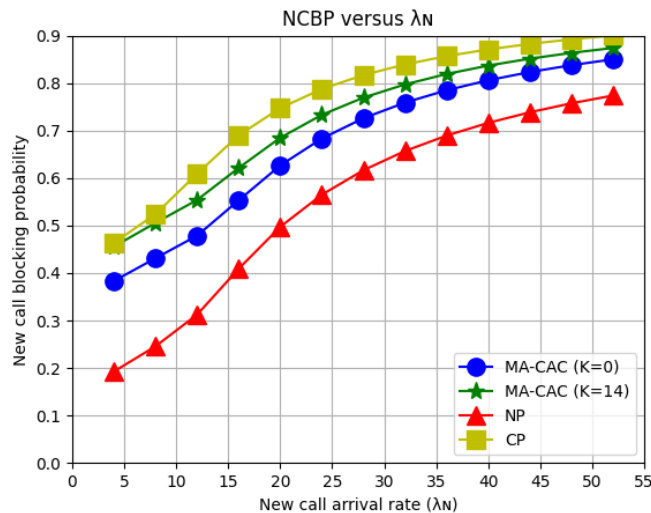
Παρατηρούμε ότι η καμπύλη του συστήματος CP για το ποσοστό χρήσης των καναλιών, ξεκινάει από την τιμή 79,26% για $\lambda = 4$ και έχει ανοδική πορεία, καθώς οι τιμές του ρυθμού λ_H αυξάνονται. Όμως η μέγιστη τιμή του utilization φαίνεται να ισορροπεί σε ένα ποσοστό της τάξης του 89%. Αυτό σημαίνει ότι το σύστημα CP δεν μπορεί να εκμεταλλευτεί πλήρως τα κανάλια που έχει στη διάθεσή του. Το φαινόμενο αυτό οφείλεται στο γεγονός ότι το μοντέλο CP διατηρεί κανάλια δεσμευμένα για να χρησιμοποιηθούν από κλήσεις μεταπομπής, τόσο στη φάση στάσης όσο και στη φάση κίνησης του οχήματος. Έτσι όσο το όχημα κινείται και από τη στιγμή που δεν μπορούν να πραγματοποιηθούν κλήσεις μεταπομπής στη φάση αυτή, ο αριθμός των καναλιών υπέρ των κλήσεων μεταπομπής παραμένει αχρησιμοποίητος και επηρεάζει το utilization των καναλιών.

Σε αντίθεση με το σύστημα CP, τα μοντέλα MA-CAC με ουρά μεταπομπής και MA-CAC χωρίς ουρά μεταπομπής, διατηρούν κανάλια για αποκλειστική χρήση από κλήσεις μεταπομπής μόνο κατά τη διάρκεια της φάσης στάσης του οχήματος ενώ στη συνέχεια (φάση κίνησης) τα αποδεσμεύουν και τα διαθέτουν υπέρ των νέων κλήσεων. Έτσι, επιτυγχάνουν την σχεδόν πλήρη χρήση των καναλιών του συστήματός τους, ξεκινώντας από ένα ποσοστό της τάξης του 90% και φτάνοντας σε ποσοστά άνω του 99%.

Σε ότι αφορά το μοντέλο NP, παρατηρούμε ότι χρησιμοποιεί σχεδόν όλους τους πόρους του συστήματός του ανεξάρτητα από τον ρυθμό άφιξης των κλήσεων μεταπομπής. Το γεγονός αυτό δικαιολογείται από τον σχεδιασμό του μοντέλου που είναι τέτοιος ώστε να διαθέτει όλα τα κανάλια του για όλες τις κλήσεις, τόσο στην περίπτωση όπου το όχημα είναι στατικό όσο και στην φάση κατά την οποία κινείται.

5.3 Επιδράσεις του λ_N

Στην Ενότητα 5.3 εξετάζονται οι συνέπειες του ρυθμού άφιξης των νέων κλήσεων στις πιθανότητες απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής καθώς και στο ποσοστό χρήσης των καναλιών των τεσσάρων μοντέλων απωλειών.

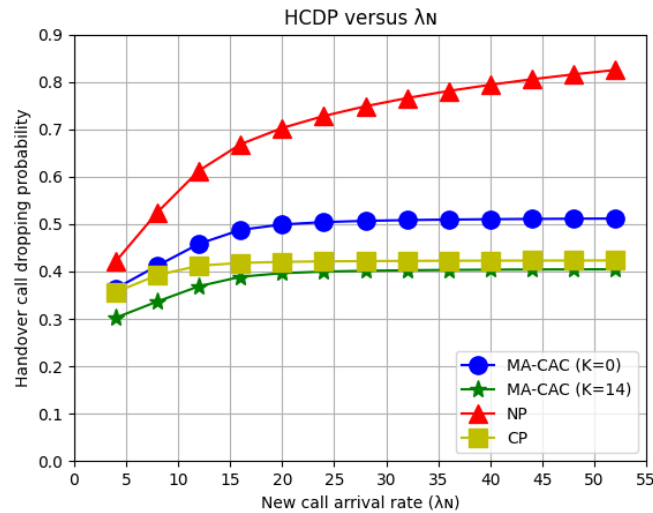


Σχήμα 5.4: Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσει του λ_N

Το πρώτο Σχήμα (5.4) της Ενότητας αφορά την πιθανότητα απώλειας των νέων κλήσεων. Εδώ παρατηρούμε ότι οι τιμές των πιθανοτήτων P_{NCBP} του μοντέλου NP είναι σαφώς μικρότερες, σε σχέση με τα άλλα μοντέλα, καθώς ο ρυθμός άφιξης των νέων κλήσεων αυξάνεται. Στα υπόλοιπα μοντέλα φαίνεται πως η διατήρηση δεσμευμένων καναλιών υπέρ των κλήσεων μεταπομπής, επηρεάζει τις πιθανότητες απώλειας των νέων κλήσεων που εισέρχονται στα συστήματά τους, καθώς το λ_N αυξάνεται. Οι υψηλότερες τιμές των P_{NCBP} σημειώνονται από το σύστημα CP, καθώς διατηρεί τον μεγαλύτερο αριθμό καναλιών υπέρ των κλήσεων μεταπομπής σε σχέση με τα υπόλοιπα μοντέλα. Ακολουθούν τα μοντέλα με τα λιγότερα δεσμευμένα κανάλια που είναι κατά σειρά το σύστημα MA-CAC με ουρά και στη συνέχεια το μοντέλο MA-CAC άνευ ουράς μεταπομπής.

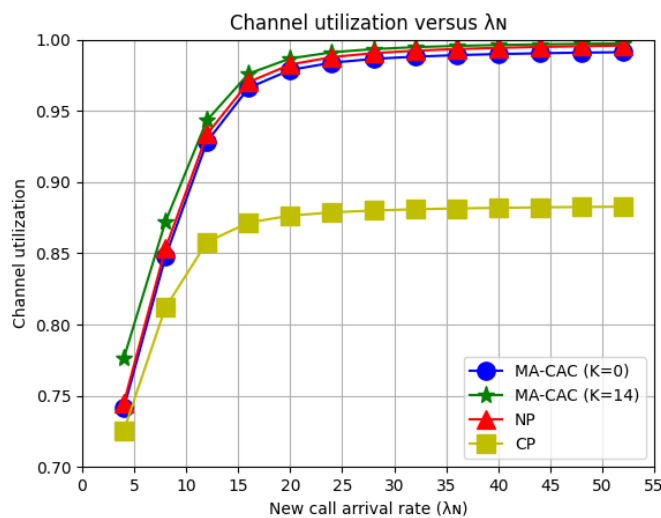
Στο Σχήμα (5.5) παρουσιάζεται η πιθανότητα απόρριψης των κλήσεων μεταπομπής όταν το λ_N μεταβάλλεται. Εδώ φαίνεται πως η ύπαρξη του αριθμού δεσμευμένων καναλιών υπέρ των κλήσεων μεταπομπής διατηρεί τις πιθανότητες απόρριψης των κλήσεων μεταπομπής των μοντέλων CP και MA-CAC, είτε με ουρά είτε δίχως ουρά, σε σχετικά χαμηλά επίπεδα όσο το λ_N αυξάνεται. Αντιθέτως, η διάθεση των καναλιών του μοντέλου NP για όλα τα είδη κλήσεων φαίνεται ότι ανεβάζει διαρκώς τις τιμές των πιθανοτήτων του P_{HCDP} , όσο ο αριθμός των εισερχόμενων νέων κλήσεων στο σύστημα αυξάνεται.

Στο τελευταίο Σχήμα αυτής της Ενότητας (Σχήμα 5.6) παρουσιάζεται το ποσοστό χρήσης των μοντέλων απωλειών/ουρών σε σχέση με τον ρυθμό λ_N . Όσο ο ρυθμός άφιξης των νέων κλήσεων αυξάνεται, τόσο ανεβαίνουν και τα ποσοστά χρήσης των καναλιών των μοντέλων NP και MA-CAC και μάλιστα με σχεδόν ίδιες τιμές. Αυτό σημαίνει ότι η ύπαρξη ή η έλλειψη της ουράς μεταπομπής δεν επηρεάζει το μοντέλο MA-CAC, σε ότι αφορά την εκμετάλλευση των πόρων του συστήματος. Για το μοντέλο CP παρατηρούμε ότι το ποσοστό χρήσης των καναλιών του είναι αρκετά χαμηλό, λόγω των καναλιών



Σχήμα 5.5: Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσει του λN

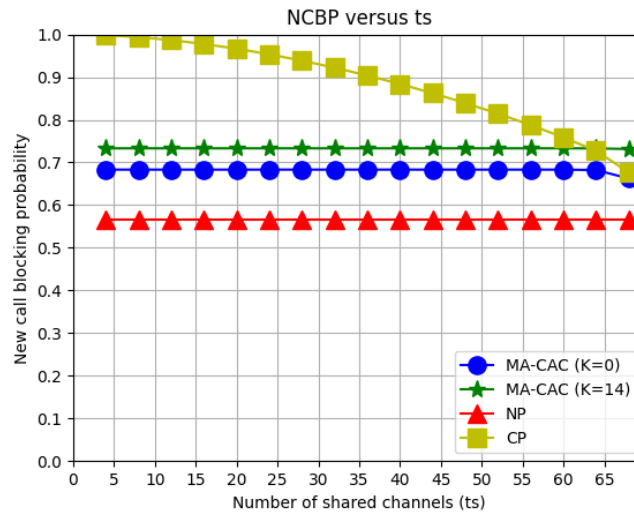
που δεν χρησιμοποιούνται από καμία κλήση στη διάρκεια της φάσης κίνησης του οχήματος.



Σχήμα 5.6: Ποσοστό Χρήσης των Καναλιών συναρτήσει του λN

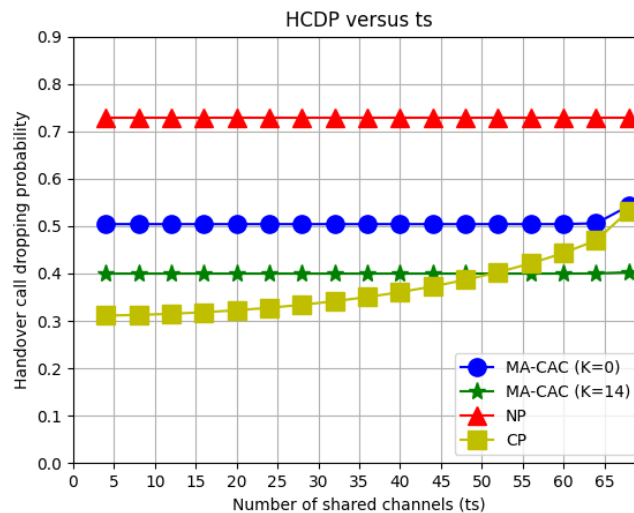
5.4 Επιδράσεις του t_S

Στην Ενότητα 5.4 παρουσιάζεται η επίδραση που έχει ο αριθμός των καναλιών που χρησιμοποιούνται τόσο από νέες κλήσεις όσο και από κλήσεις μεταπομπής (t_S), στις πιθανότητες απώλειας των νέων κλήσεων, στις πιθανότητες απόρριψης των κλήσεων μεταπομπής και στο ποσοστό χρήσης των καναλιών των μοντέλων αποδοχής κλήσεων. Στο Σχήμα 5.7 βλέπουμε τη γραφική παράσταση της πιθανότητας απώλειας των νέων κλήσεων των συστημάτων απωλειών, σε σχέση με τον αριθμό των κοινόχρηστων καναλιών t_S . Παρά το γεγονός ότι τα μοντέλα MA-CAC με ουρά και MA-CAC χωρίς ουρά μεταπομπής διατηρούν κανάλια για την εξυπηρέτηση αποκλειστικά των κλήσεων μετα-



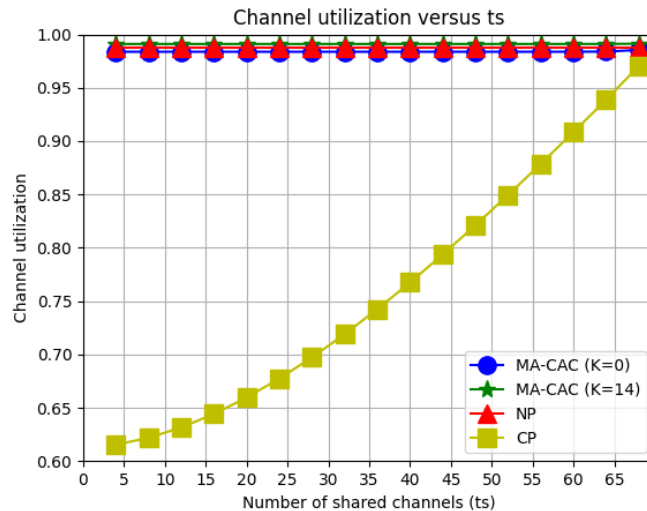
Σχήμα 5.7: Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσει του t_S

πομπής, φαίνεται ότι ο αριθμός αυτών των καναλιών ($C - t_S$), δεν επηρεάζει την πορεία των πιθανοτήτων P_{NCBP} , παρά μόνο όταν το $C - t_S$ μικρύνει υπερβολικά. Το μοντέλο που επηρεάζεται δραστικά από το t_S είναι το Cutoff Priority. Συγκεκριμένα, καθώς οι τιμές του t_S αυξάνονται, το P_{NCBP} του CP εμφανίζει φθίνουσα πορεία. Αυτό συμβαίνει διότι όσο μεγαλύτερος είναι αριθμός καναλιών $C - t_S$ υπέρ των κλήσεων μεταπομπής, τόσο λιγότερες θα είναι οι νέες κλήσεις που θα εξυπηρετηθούν από το σύστημα, στη φάση κίνησης του οχήματος. Το μοντέλο NP, όπως αναμενόταν, παραμένει ανεπηρέαστο από το t_S καθώς δεν προβλέπεται η ύπαρξη δεσμευμένων καναλιών στο σύστημά του.



Σχήμα 5.8: Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσει του t_S

Στο Σχήμα 5.8 παρουσιάζεται η πιθανότητα απόρριψης των κλήσεων μεταπομπής ως συνάρτηση του αριθμού t_S . Παρατηρούμε ότι η ιδιότητα του μοντέλου CP να διατηρεί τα δεσμευμένα υπέρ των κλήσεων μεταπομπής κανάλια του και στις δύο καταστάσεις κίνησης του οχήματος αυξάνει συνεχώς την τιμή της πιθανότητάς του P_{HCDP} όσο ο αριθμός των δεσμευμένων καναλιών μειώνεται. Για τα μοντέλα MA-CAC παρατηρείται



Σχήμα 5.9: Ποσοστό Χρήσης των Καναλιών συναρτήσει του t_s

ότι, όπως και προηγουμένως οι τιμές των P_{NCBP} παρέμεναν σταθερές στη μεταβολή του αριθμού t_s (Σχήμα 5.7), έτσι και εδώ οι τιμές των πιθανοτήτων P_{HCDP} παραμένουν επίσης σταθερές, ανεξάρτητα από την τιμή του t_s , με εξαίρεση την περίπτωση όπου ο αριθμός των κοινών καναλιών αγγίζει την χωρητικότητα του συστήματος. Το γεγονός αυτό καθιστά σχετικά εύκολη την επιλογή του κατάλληλου αριθμού t_s για τα συστήματα MA-CAC.

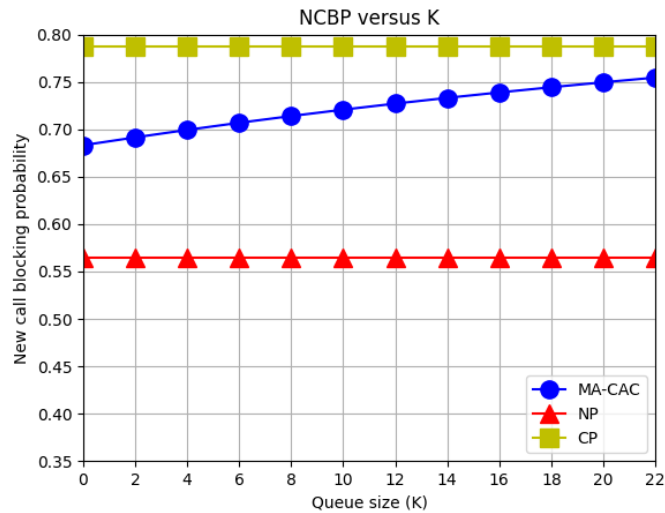
Το τελευταίο Σχήμα της παρούσας Ενότητας (Σχήμα 5.9) αναφέρεται στην επίδραση του t_s στο ποσοστό χρήσης των καναλιών των συστημάτων απωλειών. Τα μοντέλα MA-CAC και NP χρησιμοποιούν τους πόρους τους στον μέγιστο βαθμό. Δε συμβαίνει όμως το ίδιο και με το σύστημα CP καθώς φαίνεται ξεκάθαρα ότι για να επιτευχθεί ένα ικανοποιητικό ποσοστό χρήσης των καναλιών του, πρέπει και ο αριθμός t_s να είναι αρκετά μεγάλος ενώ όσο το πλήθος των κοινών καναλιών μικραίνει, μειώνεται αντίστοιχα και το ποσοστό χρήσης των καναλιών.

5.5 Επιδράσεις του K

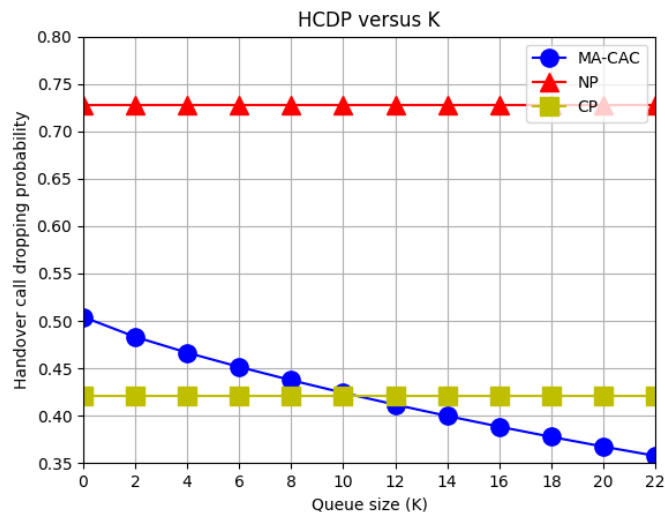
Σε αυτή την Ενότητα παρουσιάζεται η επίδραση του μεγέθους της ουράς μεταπομπής (K) στις πιθανότητες απώλειας των νέων κλήσεων και απόρριψης των κλήσεων μεταπομπής, στο ποσοστό χρήσης των καναλιών και στον χρόνο αναμονής των κλήσεων στην ουρά μεταπομπής (Waiting Time in Queue - WQ).

Τα Σχήματα 5.10 και 5.11 αναφέρονται στις πιθανότητες P_{NCBP} και P_{HCDP} , των νέων κλήσεων και των κλήσεων μεταπομπής αντίστοιχα. Στο μοντέλο MA-CAC οι τιμές του P_{NCBP} αυξάνονται καθώς συγκεντρώνεται μεγαλύτερος αριθμός κλήσεων μεταπομπής στην ουρά αναμονής, οι οποίες θα έχουν προτεραιότητα στην εξυπηρέτησή τους, αμέσως μόλις απελευθερωθεί ο αντίστοιχος αριθμός καναλιών. Για τον ίδιο λόγο, το P_{HCDP} του MA-CAC (Σχήμα 5.11) μειώνεται όσο οι τιμές του K αυξάνονται. Όπως είναι φυσικό, το μέγεθος της ουράς μεταπομπής δεν μπορεί να επηρεάσει τα μοντέλα Non Priority και Cutoff Priority.

Οι τιμές του χρόνου αναμονής των κλήσεων στην ουρά μεταπομπής (Σχήμα 5.12) είναι



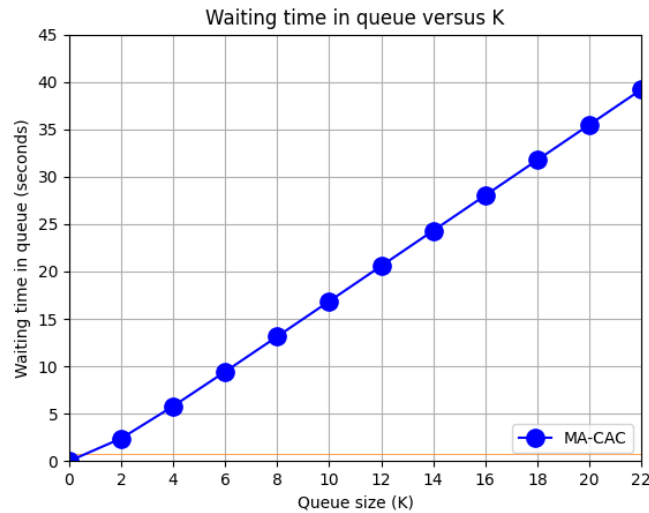
Σχήμα 5.10: Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσει του K



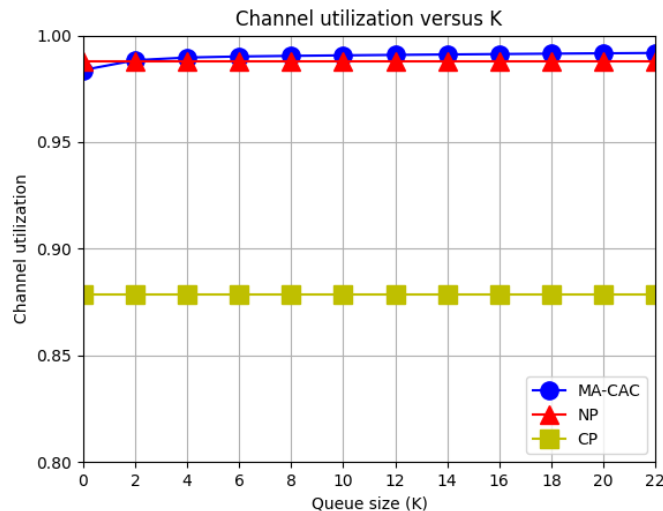
Σχήμα 5.11: Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσει του K

ανάλογες των τιμών του μεγέθους της ουράς. Πιο συγκεκριμένα, όταν το K αυξάνεται μεγαλώνει ο χρόνος παραμονής στην ουρά ενώ όταν οι τιμές του K μειώνονται, ελαχιστοποιούνται και οι τιμές του WQ . Αυτό δικαιολογείται από το γεγονός ότι όσο μεγαλύτερος είναι ο αριθμός των κλήσεων που συγκεντρώνεται στην ουρά τόσο πιο πολύς είναι ο χρόνος που απαιτείται, προκειμένου να εξυπηρετηθούν οι κλήσεις αυτές. Σίγουρα, για την κατάλληλη επιλογή του αριθμού K , θα πρέπει να ληφθεί υπόψη το χρονικό όριο για το οποίο μια κλήση δύναται να παραμείνει σε αναμονή, χωρίς να δημιουργείται πρόβλημα επικοινωνίας.

Τέλος, το ποσοστό χρήσης των καναλιών του μοντέλου MA-CAC (Σχήμα 5.13) έχει πολύ υψηλή τιμή, η οποία διατηρείται σταθερή όσο το μέγεθος της ουράς αυξάνεται. Μόνο στην περίπτωση που δεν υπάρχει ουρά έχουμε μια μικρή μείωση στο ποσοστό χρήσης των καναλιών του MA-CAC. Συνεπώς, το μέγεθος της ουράς στο σύστημα MA-CAC με ουρά μεταπομπής δεν έχει επίδραση στο ποσοστό χρήσης των καναλιών αλλά έχει με-



Σχήμα 5.12: Ποσοστό Χρήσης των Καναλιών συναρτήσει του K

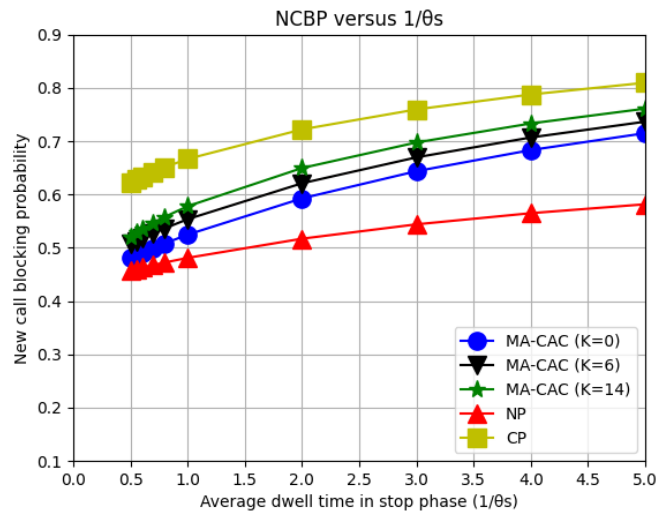


Σχήμα 5.13: Χρόνος Αναμονής των Κλήσεων στην Ουρά συναρτήσει του K

γαλύτερη τιμή συγκριτικά με το μοντέλο MA-CAC χωρίς ουρά. Φυσικά το μέγεθος της ουράς δεν έχει κανενός είδους επίδραση στο ποσοστό χρήσης των καναλιών των μοντέλων NP και CP, καθώς δεν συμπεριλαμβάνεται στον αρχιτεκτονικό σχεδιασμό των συστημάτων αυτών.

5.6 Επιδράσεις του $1/\theta_S$

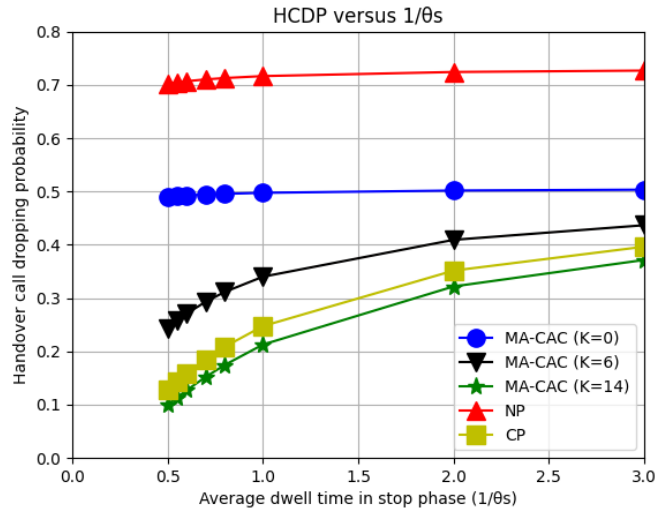
Σε αυτή την Ενότητα μελετάται η επίδραση που έχει ο χρόνος παραμονής στη φάση στάσης του οχήματος στις πιθανότητες απώλειας των νέων κλήσεων, αλλά και απόρριψης των κλήσεων μεταπομπής, των συστημάτων απωλειών/ουρών. Για το μοντέλο MA-CAC έχουν μελετηθεί τρεις περιπτώσεις. Το σύστημα MA-CAC χωρίς ουρά μεταπομπής, το MA-CAC με ουρά μεγέθους 6 θέσεων και το MA-CAC με ουρά 14 θέσεων. Στο Σχήμα 5.14 παρουσιάζεται η πιθανότητα απώλειας των νέων κλήσεων ως συνάρτηση



Σχήμα 5.14: Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσει του θ_S

του χρόνου παραμονής στη φάση στάσης $1/\theta_S$. Στη διάρκεια της φάσης κατά την οποία το όχημα είναι στατικό, το σύστημα Cutoff Priority διατηρεί δεσμευμένο έναν αριθμό από τα κανάλια του, προκειμένου να εξυπηρετήσουν κλήσεις μεταπομπής. Έτσι όσο μεγαλώνει ο χρόνος παραμονής του οχήματος στη φάση στάσης τόσο αυξάνεται και η πιθανότητα απώλειας των νέων κλήσεων, καθώς τα κανάλια που μπορούν να εξυπηρετήσουν τις κλήσεις αυτές περιορίζονται. Για τον ίδιο λόγο, τα μοντέλα MA-CAC παρουσιάζουν αντίστοιχη συμπεριφορά, στις τιμές του P_{NCBP} , με εκείνη του συστήματος CP. Μια επιπλέον παρατήρηση που αφορά το Σχήμα 5.14 είναι ότι όσο το $1/\theta_S$ μειώνεται, οι τιμές των πιθανοτήτων P_{NCBP} του MA-CAC χωρίς ουρά μεταπομπής και του μοντέλου Non Priority φαίνεται να συγκλίνουν. Αυτό δικαιολογείται από το γεγονός ότι όταν ο χρόνος παραμονής στη φάση στάσης είναι μικρός, στο σύστημα κυριαρχεί ακόμα η φάση κίνησης και όπως γνωρίζουμε, όταν το όχημα κινείται, τα συστήματα MA-CAC χωρίς ουρά μεταπομπής και NP λειτουργούν με τον ίδιο τρόπο. Μια τελευταία παρατήρηση για το Σχήμα 5.14 αφορά το γεγονός ότι οι τιμές των πιθανοτήτων απώλειας των νέων κλήσεων των μοντέλων MA-CAC δεν έχουν μεγάλες αποκλίσεις για μικρές τιμές του $1/\theta_S$. Συνεπώς το μέγεθος της ουράς μεταπομπής δεν επηρεάζει το P_{NCBP} του συστήματος MA-CAC, όσο η παραμονή στη φάση στάσης είναι σύντομη.

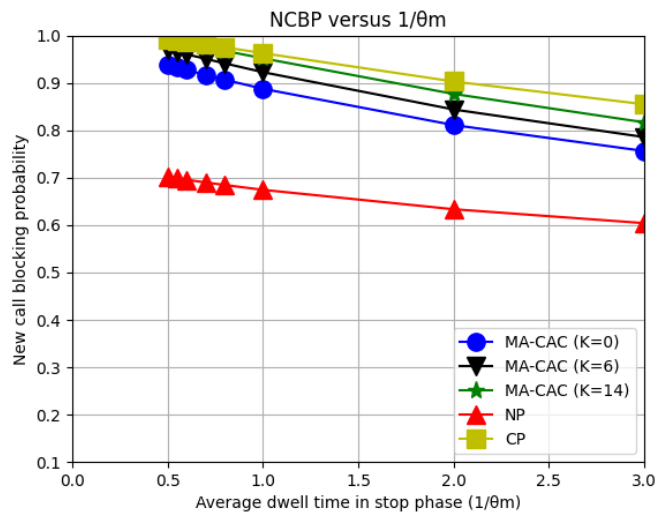
Στο Σχήμα 5.15 ερευνάται η επίδραση του χρόνου παραμονής στη φάση στάσης του οχήματος στην πιθανότητα απόρριψης των κλήσεων που προέρχονται από μεταπομπή. Εδώ παρατηρούμε ότι οι τιμές των πιθανοτήτων P_{HCDP} των μοντέλων MA-CAC χωρίς ουρά μεταπομπής και NP είναι σταθερές σχεδόν καθ' όλη τη διάρκεια της φάσης στάσης. Το μοντέλο CP, όσο ο χρόνος παραμονής στη φάση στάσης είναι σύντομος, φαίνεται να μειώνει σε μεγάλο βαθμό τις τιμές των πιθανοτήτων απόρριψης των κλήσεων μεταπομπής, γεγονός που οφείλεται στα κανάλια που διαθέτει για την αποκλειστική χρήση τους από κλήσεις μεταπομπής. Επιπλέον, για μικρές τιμές του $1/\theta_S$, παρατηρούμε μεγάλη διαφορά στις πιθανότητες απόρριψης των κλήσεων μεταπομπής των μοντέλων MA-CAC. Αυτό οφείλεται στο γεγονός ότι όσο μεγαλώνει το μέγεθος της ουράς μεταπομπής τόσο αριθμός των κλήσεων μεταπομπής που εξυπηρετούνται από το σύστημα. Συνεπώς, αν ο χρόνος παραμονής του οχήματος στη φάση στάσης του είναι σύντομος, το μέγεθος της ουράς επιδρά σε μεγάλο βαθμό στις πιθανότητες P_{HCDP} του συστήματος MA-CAC.



Σχήμα 5.15: Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσει του θ_S

5.7 Επιδράσεις του $1/\theta_M$

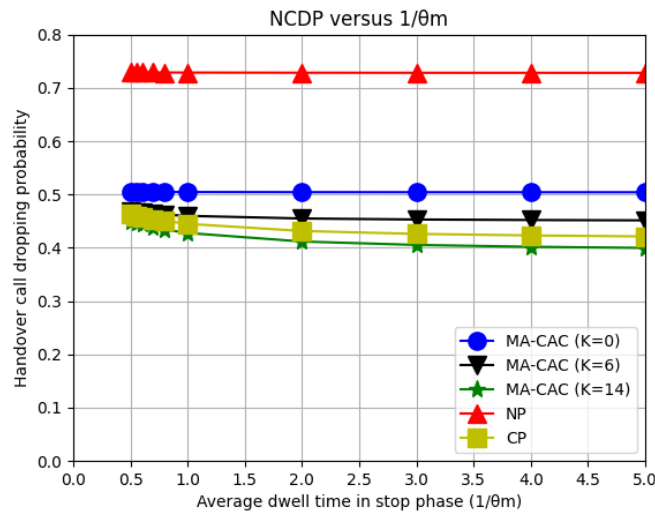
Στην τελευταία Ενότητα του παρόντος Κεφαλαίου παρουσιάζεται η πιθανότητα απώλειας των νέων κλήσεων και η πιθανότητα απόρριψης των κλήσεων μεταπομπής, συναρτήσει του χρόνου παραμονής του οχήματος στην φάση κίνησης $1/\theta_M$. Για τα μοντέλα MA-CAC μελετήθηκαν οι περιπτώσεις του μοντέλου χωρίς ουρά μεταπομπής και των μοντέλων με ουρά μεγέθους 6 θέσεων και 14 θέσεων, αντίστοιχα.



Σχήμα 5.16: Πιθανότητα Απώλειας των Νέων Κλήσεων συναρτήσει του θ_M

Στο Σχήμα 5.16 ερευνάται η επίδραση του χρόνου παραμονής του οχήματος στη φάση κίνησης στις πιθανότητες απώλειας των νέων κλήσεων. Τα αριθμητικά αποτελέσματα για την πιθανότητα P_{NCBP} του συστήματος MA-CAC με την ουρά μεταπομπής μεγέθους 14 θέσεων και του μοντέλου CP, όσο ο χρόνος παραμονής στην φάση κίνησης ελαχιστοποιείται, συγκλίνουν σχεδόν στην ίδια τιμή. Δε συμβαίνει όμως το ίδιο με τα μοντέλα MA-CAC με την ουρά μεταπομπής χωρητικότητας 6 θέσεων και MA-CAC χωρίς ουρά

μεταπομπής, όπου παρατηρούμε ότι όσο μειώνεται το μέγεθος της ουράς, οι τιμές για την πιθανότητά τους P_{NCBP} διαφοροποιούνται από εκείνες του συστήματος CP. Όπως γνωρίζουμε το σύστημα CP διατηρεί και στις δύο φάσεις κίνησης του οχήματος έναν αριθμό δεσμευμένων καναλιών για την εξυπηρέτηση των κλήσεων μεταπομπής. Αυτή την ιδιότητα τη συναντούμε και στο σύστημα MA-CAC, αλλά μόνο όσο το όχημα είναι στατικό. Συνεπώς, αν το μέγεθος της ουράς είναι κατάλληλο και οι τιμές του $1/\theta_M$ είναι μικρές, το μοντέλο MA-CAC συμπεριφέρεται με τον ίδιο τρόπο με το μοντέλο Cutoff Priority καθώς όταν ο χρόνος παραμονής στη φάση κίνησης είναι σύντομος, κυριαρχεί στο σύστημα η φάση στάσης.



Σχήμα 5.17: Πιθανότητα Απόρριψης των Κλήσεων Μεταπομπής συναρτήσει του θ_M

Η κυριαρχία της φάσης στάσης επί της φάσης κίνησης παρατηρείται και στο Σχήμα 5.17 όπου όσο οι τιμές του $1/\theta_M$ ελαχιστοποιούνται, οι πιθανότητες απόρριψης των κλήσεων μεταπομπής των συστημάτων MA-CAC και του συστήματος CP καταγράφουν χαμηλές τιμές και μάλιστα όσο πιο μεγάλος είναι ο αριθμός των κλήσεων που δύναται να φιλοξενήσει η ουρά μεταπομπής, τόσο οι τιμές του CP και του MA-CAC συγκλίνουν. Επιπλέον παρατηρούμε ότι, όσο μεγαλώνει η διάρκεια παραμονής στη φάση στάσης, οι τιμές όλων των μοντέλων απωλειών κλήσεων διαγράφουν σταθερή πορεία. Αυτό οφείλεται στο γεγονός ότι όσο το όχημα βρίσκεται σε κίνηση δεν μπορούν να πραγματοποιηθούν επιπλέον κλήσεις μεταπομπής, από εκείνες που είναι ήδη ενεργές από το τέλος της φάσης κατά την οποία το όχημα βρισκόταν σε στάση.

Ανακεφαλαίωση - Συμπεράσματα

Στην παρούσα διπλωματική εργασία ερευνήθηκαν τέσσερα μοντέλα απωλειών/ουρών κλήσεων στα mobile hotspots οχημάτων. Συγκεκριμένα, μελετήθηκαν τα μοντέλα MA-CAC με ουρά μεταπομπής και MA-CAC χωρίς ουρά μεταπομπής καθώς και τα συστήματα Non Priority (NP) και Cutoff Priority (CP). Τα μοντέλα αυτά, είναι μονοδιάστατης τηλεπικοινωνιακής κίνησης και αναλύθηκαν ως Μαρκοβιανές αλυσίδες συνεχούς χρόνου, ενώ εξυπηρετούν κλήσεις δύο ειδών: τις κλήσεις που προέρχονται από μεταπομπή, η οποία διενεργείται από έναν εξωτερικό σταθμό βάσης προς το σημείο πρόσβασης του δικτύου WLAN του οχήματος και τις νέες κλήσεις. Επίσης παρουσιάστηκε για το κάθε μοντέλο ένα αριθμητικό παράδειγμα, όπου υπολογίστηκαν η πιθανότητα απόρριψης των κλήσεων μεταπομπής, η πιθανότητα απώλειας των νέων κλήσεων και το ποσοστό χρήσης των καναλιών του κάθε συστήματος αντίστοιχα. Επιπροσθέτως, στο παράδειγμα του μοντέλου MA-CAC με ουρά μεταπομπής, υπολογίστηκε και ο μέσος χρόνος αναμονής των κλήσεων στην ουρά μεταπομπής. Τα αριθμητικά αποτελέσματα των συγκεκριμένων παραδειγμάτων επαληθεύτηκαν από προγράμματα γλώσσας Python, στα οποία υλοποιήθηκαν τα τέσσερα μοντέλα απωλειών που ερευνήθηκαν στην παρούσα εργασία. Τέλος, εξετάστηκαν οι επιδράσεις μερικών από τις παραμέτρους εισόδου των συστημάτων απωλειών κλήσεων στις πιθανότητες απόρριψης των κλήσεων μεταπομπής και απώλειας των νέων κλήσεων, στο ποσοστό χρήσης των καναλιών και στον χρόνο αναμονής στην ουρά μεταπομπής. Οι επιδράσεις αυτές των παραμέτρων οπτικοποιήθηκαν με τη βοήθεια ενός προγράμματος σε γλώσσα Python.

Τα συμπεράσματα που προκύπτουν από τη μελέτη αυτών των τεσσάρων μοντέλων απωλειών κλήσεων είναι ξεκάθαρα. Το σύστημα MA-CAC μπορεί να μειώσει την πιθανότητα απώλειας των νέων κλήσεων και την πιθανότητα απόρριψης των κλήσεων μεταπομπής, ενώ συγχρόνως επιτυγχάνεται υψηλό ποσοστό χρήσης των καναλιών του δικτύου. Ιδιαίτερα το μοντέλο MA-CAC που είναι εφοδιασμένο με την ουρά μεταπομπής, μειώνει ακόμα περισσότερο την πιθανότητα απόρριψης των κλήσεων μεταπομπής, συγκριτικά με την εκδοχή του χωρίς την ουρά. Το μοντέλο Cutoff Priority δεν επιτυγχάνει υψηλά ποσοστά χρήσης των καναλιών και αυτό οφείλεται στο γεγονός ότι διατηρεί τα δεσμευμένα κανάλια υπέρ των κλήσεων μεταπομπής τόσο στη φάση στάσης όσο και στη φάση κίνησης του οχήματος, παρόλο που στην τελευταία περίπτωση δεν μπορούν να πραγματοποιηθούν κλήσεις μεταπομπής. Έτσι σε όλη τη διάρκεια της φάσης κατά την οποία το όχημα βρίσκεται σε κίνηση, ένας σταθερός αριθμός καναλιών παραμένει αχρησιμοποίητος. Για το μοντέλο Non Priority παρατηρούμε ότι παρουσιάζει μεγαλύτερες τιμές στις πιθανότητες απόρριψης των κλήσεων μεταπομπής, συγκριτικά με τα υπόλοιπα μοντέλα. Μια τελευταία παρατήρηση αφορά το μέγεθος της ουράς μεταπομπής του οποίου όσο η τιμή αυξάνεται τόσο μεγαλώνει και ο χρόνος παραμονής των κλήσεων μεταπομπής στην ουρά.

Σε μελλοντικές εργασίες μπορεί να μελετηθεί περαιτέρω το ζήτημα της επιλογής του κατάλληλου μεγέθους της ουράς μεταπομπής, προκειμένου ο χρόνος παραμονής των κλήσεων σε αυτή να διατηρείται σε ένα επίπεδο τέτοιο ώστε να μην υπερβαίνεται το χρονικό όριο που δύναται μία κλήση να παρεμείνει στην ουρά, χωρίς να δημιουργηθεί πρόβλημα επικοινωνίας του χρήστη. Επιπλέον μπορούν να διεξαχθούν προσομοιώσεις τόσο των συστημάτων απωλειών/ουρών κλήσεων όσο και των παραδειγμάτων που παρουσιάστηκαν στις Ενότητες 1.4, 2.4, 3.4 και 4.4, προκειμένου να επαληθευτούν τα αποτελέσματα που προέκυψαν από τα αναλυτικά μοντέλα της παρούσας εργασίας.

Παράρτημα Α΄

Ακρωνύμια και συντομογραφίες

AP Access Point

CP Cutoff Priority

HCDP Handover-Call Dropping Probability

HQ Handover Queue

IVN Intelligent Vehicular Network

KTX Korean Train eXpress

LAP Logical Access Point

MA-CAC Mobility-Aware Call Admission Control

MPTCP Multipath Transmission Control Protocol

NCBP New-Call Blocking Probability

NEMO Network Mobility Protocol

NP Non Priority

QoS Quality of Service

SDN Software Defined Network

SIP Session Initiation Protocol

SGP Société du Grand Paris

TCP Transmission Control Protocol

TFRC Transmission Control Protocol Friendly Rate Control

V2I Vehicle to Infrastructure

WiFi Wireless Fidelity

WLAN Wireless Local Area Network

WQ Waiting Time in Queue

WWAN Wireless Wide Area Network

Παράρτημα Β΄

Προγράμματα σε Γλώσσα Python των Μοντέλων Απωλειών για τα Παραδείγματα των Κεφαλαίων 1, 2, 3 και 4

Β΄.1 Πρόγραμμα MA-CAC με ΗΘ του Παραδείγματος 1.4

```
# Rate of the arrival process of New calls:
λn = 2
# Rate of the arrival process of Handover calls:
λh = 3
# Capacity of WLAN:
C = 5
# Number of shared channels:
ts = 4
# Rate of the call duration time in stop phase:
μs = 0.75
# Rate of the call duration time in moving phase:
μm = 0.5
# Rate of the dwell time in stop phase:
θs = 0.1
# Rate of the dwell time in moving phase:
θm = 0.05
# Handover queue size:
K = 3

l1 = 0
k1 = 0

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
```

```

S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ1 = Decimal(0)
Summ2 = Decimal(0)

import xlswriter

workbook = xlswriter.Workbook('K_3_C_5_tnew_4__S0_in.xlsx')
bold_format = workbook.add_format({'bold': True})
bold_format.set_align('top')
bold_format.set_align('center')
bold_format.set_bg_color('green')
bold_format.set_border()
bold_format_2 = workbook.add_format({'bold': True})
bold_format_2.set_align('top')
bold_format_2.set_align('center')
bold_format_2.set_bg_color('blue')
bold_format_2.set_border()
bold_format_3 = workbook.add_format({'bold': True})
bold_format_3.set_align('top')
bold_format_3.set_align('center')
bold_format_3.set_bg_color('red')
bold_format_3.set_border()
bold_format_4 = workbook.add_format({'bold': True})
bold_format_4.set_align('top')
bold_format_4.set_align('center')
bold_format_4.set_bg_color('yellow')
bold_format_4.set_border()

# creation of the excel worksheet of S(i,n) contributors:
worksheet = workbook.add_worksheet('S(i,n)')
# creation of the excel worksheet of P(i,n) steady-state probabilities:
worksheet_2 = workbook.add_worksheet('P(i,n)')
# creation of the excel worksheet of NCBP, HCDP , Utilization & Waiting
  ↪ Time in the queue:
worksheet_3 = workbook.add_worksheet('Results')

worksheet.write('A1', 'n', bold_format)
worksheet.write('B1', 'S0(0,n)', bold_format)
worksheet.write('C1', 'S0(1,n)', bold_format)
worksheet.write('D1', 'S1(0,n)', bold_format)
worksheet.write('E1', 'S1(1,n)', bold_format)
worksheet_2.write('A1', 'n', bold_format)
worksheet_2.write('B1', 'P(0,n)', bold_format)
worksheet_2.write('C1', 'P(1,n)', bold_format)
worksheet_3.write('A1', 'P(NCBP)', bold_format_3)

```



```

worksheet_3.write('B1', 'P(HCDP)', bold_format)
worksheet_3.write('C1', 'Utilization', bold_format_2)
worksheet_3.write('D1', 'Waiting Time in the Queue(sec)', bold_format_4)
rowIndex = 2
worksheet.write('A' + str(rowIndex), 0)
worksheet.write('B' + str(rowIndex), S1)
worksheet.write('C' + str(rowIndex), L1)
worksheet.write('D' + str(rowIndex), F1)
worksheet.write('E' + str(rowIndex), Q1)
worksheet_2.write('A' + str(rowIndex), 0)
rowIndex = 3

# calculation of S(i,n) contributors:
for n in range(0, C + K):
    worksheet.write('A' + str(rowIndex), n + 1)
    for i in range(0, 2):
        if i == 0 and n == 0:
            print("S\u2070" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  "\u2192" + str(S1))
            print("S\u00B9" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  "\u2192" + str(F1))
            Summ1 += S1
            Summ2 += F1
        elif i == 1 and n == 0:
            print("S\u2070" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  "\u2192" + str(L1))
            print("S\u00B9" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  "\u2192" + str(Q1))
            Summ1 += L1
            Summ2 += Q1
        if i == 0 and n <= ts - 1:
            \theta_i = \theta_s
            j = 1 - i
            if j == 0:
                \theta_j = \theta_s
            else:
                \theta_j = \theta_m
            \mu_i = \mu_s
            \lambda = \lambda_h + \lambda_n
            S = (((Decimal(\lambda) + Decimal(n) * Decimal(\mu_i) + Decimal(\theta_i)) /
                  "\u2192" + ((Decimal(n + 1)) * Decimal(\mu_i)))
                  * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                  "\u2192" + Decimal(\mu_i))) * Decimal(S2)) - (
                  (Decimal(\theta_j) / ((Decimal(n + 1)) * Decimal(\mu_i))) *
                  "\u2192" + Decimal(L1))
            F = (((Decimal(\lambda) + Decimal(n) * Decimal(\mu_i) + Decimal(\theta_i)) /
                  "\u2192" + ((Decimal(n + 1)) * Decimal(\mu_i)))
                  * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                  "\u2192" + Decimal(\mu_i))) * Decimal(F2)) - (
                  (Decimal(\theta_j) / ((Decimal(n + 1)) * Decimal(\mu_i))) *
                  "\u2192" + Decimal(Q1))
            Summ1 += S
            Summ2 += F
            worksheet.write('B' + str(rowIndex), S)
            worksheet.write('D' + str(rowIndex), F)
            print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " = " +
                  "\u2192" + str(S))
            if n == C - 1:

```

```

λ = λh + λn
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(F))
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= ts and n < C:
ϑi = ϑs
j = 1 - i
if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μs
λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ ((Decimal(n + 1)) * Decimal(μi))) * Decimal(

```

```

S1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
↪ * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
↪ ((Decimal(n + 1)) * Decimal(μi))) * Decimal(
F1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
↪ * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(Q1))
print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
↪ " + str(S))
Summ1 += S
Summ2 += F
worksheet.write('B' + str(rowIndex), S)
worksheet.write('D' + str(rowIndex), F)
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
S2 = S1

```

```

S1 = S
print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(F))
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
    ↪ ((Decimal(C)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
    ↪ ((Decimal(C)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(Q1))
    print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(S))
    Summ1 += S
    Summ2 += F
    worksheet.write('B' + str(rowIndex), S)
    worksheet.write('D' + str(rowIndex), F)
    if n == K + C - 1:
        λ = λh
        S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(S1)) -
        ↪ (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi)))
        ↪ * Decimal(L1))
        F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(F1)) -
        ↪ (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi)))
        ↪ * Decimal(Q1))

        A = S
        D = F
        X = λ
    elif n == K + C - 2:
        λ = λh
        S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (

```

```

        (Decimal(C)) * Decimal(μi))) * Decimal(S1)) -
        ↪ (
            (Decimal(l1) / ((Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(θj) / ((Decimal(C)) * Decimal(μi)))
                ↪ * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(F1)) -
    ↪ (
        (Decimal(l1) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(θj) / ((Decimal(C)) * Decimal(μi)))
                ↪ * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(F))
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
    ↪ ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(L1)) - ((Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((n + 1)) * Decimal(μi))) *
        ↪ Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
    ↪ ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(Q1)) - ((Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F2))

Summ1 += L
Summ2 += Q
worksheet.write('C' + str(rowIndex), L)
worksheet.write('E' + str(rowIndex), Q)
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
                (Decimal(θj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))

```

```

Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↳ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↳ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
            ↳ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
                ↳ Decimal(μi))) * Decimal(F2))

P = L
G = Q
print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↳ " + str(L))
L2 = L1
L1 = L
print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↳ " + str(Q))
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(C)) * Decimal(μi)))
    ↳ * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((C)) * Decimal(μi))) *
        ↳ Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(C)) * Decimal(μi)))
    ↳ * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(C)) * Decimal(μi))) *
        ↳ Decimal(F2))

Summ1 += L
Summ2 += Q
worksheet.write('C' + str(rowIndex), L)
worksheet.write('E' + str(rowIndex), Q)
print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↳ " + str(L))
L2 = L1
L1 = L
print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↳ " + str(Q))
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs

```

```

else:
     $\vartheta_j = \vartheta_m$ 
     $\mu_i = \mu_m$ 
     $k = \lambda_n$ 
    L = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
        (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(L1)) - (
        (Decimal( $\vartheta_j$ ) / (Decimal(C) * Decimal( $\mu_i$ ))) *
         $\hookrightarrow$  Decimal(S2))
    Q = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
        (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) - (
        (Decimal( $\vartheta_j$ ) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
         $\hookrightarrow$  Decimal(F2))

    Summ1 += L
    Summ2 += Q
    worksheet.write('C' + str(rowIndex), L)
    worksheet.write('E' + str(rowIndex), Q)
    if n == K + C - 1:
        L = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
            (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(L1)) - (
            (Decimal( $\vartheta_j$ ) / (Decimal(C) * Decimal( $\mu_i$ ))) *
             $\hookrightarrow$  Decimal(S2))
        Q = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
            (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) - (
            (Decimal( $\vartheta_j$ ) / ((Decimal(C)) * Decimal( $\mu_i$ )))
             $\hookrightarrow$  * Decimal(F2))

        P = L
        G = Q
        print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
             $\hookrightarrow$  " + str(L))
        L2 = L1
        L1 = L
        print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
             $\hookrightarrow$  " + str(Q))
        Q2 = Q1
        Q1 = Q
        k1 = k

    rowIndex += 1
    worksheet.set_column(1, 1, width=15)
    worksheet.set_column(2, 2, width=15)
    worksheet.set_column(3, 3, width=15)
    worksheet.set_column(4, 4, width=15)
    print("-----")

    # calculation of P(0,0) & P(1,0) probabilities:
    r = (Decimal(C) * Decimal( $\mu_s$ ) * A + Decimal( $\vartheta_s$ ) * A - X * B - Decimal( $\vartheta_m$ )
         $\hookrightarrow$  * P) / - (Decimal(C)
        * Decimal( $\mu_s$ ) * D + Decimal( $\vartheta_s$ ) * D - X * E - Decimal( $\vartheta_m$ ) * G)
    print("r = " + str(r))
    print("-----")
    P00 = 1 / (Summ1 + r * Summ2)
    P10 = r * P00
    rowIndex = 2
    worksheet_2.write('B' + str(rowIndex), P00)
    worksheet_2.write('C' + str(rowIndex), P10)
    rowIndex = 3

    # calculation of the rest of P(i,n) probabilities:
    S1 = Decimal(1)

```

```

L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new = 0
Bm_new = 0
Summ_P0C = 0
Summ_P1C = 0
Summ_P0C_K = 0
Summ_P1C_K = 0
Summ_P0_K_U = 0
Summ_P1_K_U = 0
Summ_NQ_0 = 0
Summ_NQ_1 = 0
Summ_λe_1 = 0
Summ_λe_2 = 0
Summ_λe_3 = 0

for n in range(0, C + K):
    worksheet_2.write('A' + str(rowIndex), n + 1)
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0C += n * Pin
            Summ_P0C_K += Pin
            Summ_λe_1 += (λh + λn) * Pin
            print("P(" + str(i) + ", " + str(n) + ") = " + str(Pin))
            if i == 0 and n >= ts - 1 and n < C + K:
                Bs_new += Pin
            if i == 0 and n == C - 1:
                Bs_handoff = Pin
            if i == 0 and n == C + K - 1:
                P0_C_K = Pin
            if i == 0 and n >= C and n < C + K:
                P0_C_K_U = Pin
                Summ_P0_K_U += C * P0_C_K_U
                Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P1C_K += Pin
            Summ_P1C += n * Pin
            print("P(" + str(i) + ", " + str(n) + ") = " + str(Pin))
            Summ_λe_3 += λn * Pin
            if i == 1 and n >= C - 1:
                Bm_new += Pin
            if i == 1 and n >= C and n < C + K:
                P1_C_K_U = Pin
                Summ_P1_K_U += C * P1_C_K_U
                Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
        if i == 0 and n <= ts - 1:
            ϑi = ϑs
            j = 1 - i
            if j == 0:
                ϑj = ϑs
            else:

```



```

     $\vartheta_j = \vartheta_m$ 
 $\mu_i = \mu_s$ 
 $\lambda = \lambda_h + \lambda_n$ 
S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
↪ ((Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
S1)) - (Decimal(l1) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
↪ * Decimal(S2)) - (
    Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
↪ * Decimal(L1))
F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
↪ ((Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
F1)) - (Decimal(l1) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
↪ * Decimal(F2)) - (
    Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
↪ * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
worksheet_2.write('B' + str(rowIndex), Pin)
print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
Summ_P0C += (n + 1) * Pin
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n < ts - 1:
    Summ_λe_1 += λ * Pin
if i==0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= ts and n < C:
     $\vartheta_i = \vartheta_s$ 
    j = 1 - i
    if j == 0:
         $\vartheta_j = \vartheta_s$ 
    else:
         $\vartheta_j = \vartheta_m$ 
 $\mu_i = \mu_s$ 
 $\lambda = \lambda_h$ 
S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
↪ ((Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
S1)) - (Decimal(l1) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
↪ * Decimal(S2)) - (
    Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
↪ * Decimal(L1))
F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
↪ ((Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
F1)) - (Decimal(l1) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
↪ * Decimal(F2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
worksheet_2.write('B' + str(rowIndex), Pin)
print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
Summ_P0C += (n + 1) * Pin
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
    ↪ ((Decimal(C)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
    ↪ ((Decimal(C)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
worksheet_2.write('B' + str(rowIndex), Pin)
print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U

```

```

        Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
    if i == 0 and n >= ts - 1 and n < C + K - 1:
        Summ_λe_2 += λh * Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
    L = ((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
        ↪ (
            (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(θj) / (Decimal((n + 1)) * Decimal(μi)))
                ↪ * Decimal(S2))
    Q = ((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
        ↪ (
            (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
                ↪ * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
    Summ_P1C += (n + 1) * Pin
    Summ_P1C_K += Pin
    worksheet_2.write('C' + str(rowIndex), Pin)
    if i == 1 and n >= C - 1 and n < C + K:
        Bm_new += Pin
    if i == 1 and n >= C and n < C + K:
        P1_C_K_U = Pin
        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
    if i == 1 and n < C - 1:
        Summ_λe_3 += λn * Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm

```

```

μi = μm
k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C) * Decimal(μi))) * Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(C) * Decimal(μi))) *
    ↪ Decimal(L2)) - (
    (Decimal(ϑj) / (Decimal((C) * Decimal(μi))) *
    ↪ Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C) * Decimal(μi))) * Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(C) * Decimal(μi))) *
    ↪ Decimal(Q2)) - (
    (Decimal(ϑj) / ((Decimal(C) * Decimal(μi))) *
    ↪ Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
worksheet_2.write('C' + str(rowIndex), Pin)
print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
Summ_P1C_K += Pin
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(ϑj) / (Decimal(C) * Decimal(μi))) *
        ↪ Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(ϑj) / ((Decimal(C) * Decimal(μi))) *
        ↪ Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    worksheet_2.write('C' + str(rowIndex), Pin)
    print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
    Summ_P1C_K += Pin
    if i == 1 and n >= C - 1 and n < C + K:
        Bm_new += Pin
    if i == 1 and n >= C and n < C + K:
        P1_C_K_U = Pin
        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U

```

```

        if i == 1 and n < C - 1:
            Summ_λe_3 += λn * Pin
        L2 = L1
        L1 = L
        Q2 = Q1
        Q1 = Q
        k1 = k
    rowIndex += 1
print ("-----")
print ("Summ P(0,n) = " + str(Summ_P0C_K))
print ("Summ P(1,n) = " + str(Summ_P1C_K))
print ("-----")

# calculation of total new call blocking probability:
Bnew = Bs_new + Bm_new
print ("Bs_new = " + str(Bs_new))
print ("Bm_new = " + str(Bm_new))
print ("Bnew = " + str(Bnew))
print ("-----")

# calculation of total handover call dropping probability:
P_HCDP = P0_C_K / Summ_P0C_K
print ("P_HCDP = " + str(P_HCDP))
print ("-----")

# calculation of channel utilization:
U = Summ_P0C + Summ_P1C
U2 = Summ_P0_K_U + Summ_P1_K_U
Utilization = (U + U2) / C
print ("Utilization = " + str(Utilization))
print ("-----")

# calculation of the average number of calls in the handover queue:
NQ = Summ_NQ_0 + Summ_NQ_1
print ("NQ = " + str(NQ))
print ("-----")

# calculation of the effective call arrival rate:
λe = Summ_λe_1 + Summ_λe_2 + Summ_λe_3
print ("λe = " + str(λe))
print ("-----")

# calculation of the average handover call waiting time in queue:
WQ = (NQ / λe) * 60
print ("WQ = " + str(WQ))
rowIndex = 2
worksheet_3.write('A' + str(rowIndex), Bnew)
worksheet_3.write('B' + str(rowIndex), P_HCDP)
worksheet_3.write('C' + str(rowIndex), Utilization)
worksheet_3.write('D' + str(rowIndex), WQ)
print ("-----")
worksheet_2.set_column(1, 1, width=15)
worksheet_2.set_column(2, 2, width=15)
worksheet_3.set_column(0, 0, width=23)
worksheet_3.set_column(1, 1, width=23)
worksheet_3.set_column(2, 2, width=23)
worksheet_3.set_column(3, 3, width=28)

```

```
workbook.close()
```

B.2 Πρόγραμμα MA-CAC χωρίς ΗΘ του Παραδείγματος 2.4

```
# Rate of the arrival process of New calls:
λn = 2
# Rate of the arrival process of Handover calls:
λh = 1
# Capacity of WLAN:
C = 6
# Number of shared channels:
ts = 2
# Rate of the call duration time in stop phase:
μs = 1/4
# Rate of the call duration time in moving phase:
μm = 1/5
# Rate of the dwell time in stop phase:
ϑs = 1/40
# Rate of the dwell time in moving phase:
ϑm = 1/10
# Handover queue size:
K = 0

l1 = 0
k1 = 0

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ1 = Decimal(0)
Summ2 = Decimal(0)

import xlswriter

workbook = xlswriter.Workbook('K_0_C_6__S0_in.xlsx')
bold_format = workbook.add_format({'bold': True})
bold_format.set_align('top')
```

```

bold_format.set_align('center')
bold_format.set_bg_color('green')
bold_format.set_border()
bold_format_2 = workbook.add_format({'bold': True})
bold_format_2.set_align('top')
bold_format_2.set_align('center')
bold_format_2.set_bg_color('blue')
bold_format_2.set_border()
bold_format_3 = workbook.add_format({'bold': True})
bold_format_3.set_align('top')
bold_format_3.set_align('center')
bold_format_3.set_bg_color('red')
bold_format_3.set_border()

# creation of the excel worksheet of S(i,n) contributors:
worksheet = workbook.add_worksheet('S(i,n)')
# creation of the excel worksheet of P(i,n) steady-state probabilities:
worksheet_2 = workbook.add_worksheet('P(i,n)')
# creation of the excel worksheet of NCBP, HCDP , Utilization & Waiting
↳ Time in the queue:
worksheet_3 = workbook.add_worksheet('Results')

worksheet.write('A1', 'n', bold_format)
worksheet.write('B1', 'S0(0,n)', bold_format)
worksheet.write('C1', 'S0(1,n)', bold_format)
worksheet.write('D1', 'S1(0,n)', bold_format)
worksheet.write('E1', 'S1(1,n)', bold_format)
worksheet_2.write('A1', 'n', bold_format)
worksheet_2.write('B1', 'P(0,n)', bold_format)
worksheet_2.write('C1', 'P(1,n)', bold_format)
worksheet_3.write('A1', 'P(NCBP)', bold_format_3)
worksheet_3.write('B1', 'P(HCDP)', bold_format)
worksheet_3.write('C1', 'Utilization', bold_format_2)
worksheet_3.write('D1', 'Waiting Time in the Queue(sec)', bold_format)
rowIndex = 2
worksheet.write('A' + str(rowIndex), 0)
worksheet.write('B' + str(rowIndex), S1)
worksheet.write('C' + str(rowIndex), L1)
worksheet.write('D' + str(rowIndex), F1)
worksheet.write('E' + str(rowIndex), Q1)
worksheet_2.write('A' + str(rowIndex), 0)
rowIndex = 3

# calculation of S(i,n) contributors:
for n in range(0, C + K):
    worksheet.write('A' + str(rowIndex), n + 1)
    for i in range(0, 2):
        if i == 0 and n == 0:
            print("S\u2070" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  ↳ str(S1))
            print("S\u00B9" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  ↳ str(F1))
            Summ1 += S1
            Summ2 += F1
        elif i == 1 and n == 0:
            print("S\u2070" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  ↳ str(L1))

```

```

print("S\u00B9" + "(" + str(i) + "," + str(n) + ")" + " = " +
      ↪ str(Q1))
Summ1 += L1
Summ2 += Q1
if i == 0 and n <= ts - 1:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
          ↪ ((Decimal(n + 1)) * Decimal(μi))) *
          Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
          ↪ Decimal(μi))) * Decimal(S2)) - (
          (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi))) *
          ↪ Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
          ↪ ((Decimal(n + 1)) * Decimal(μi))) *
          Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
          ↪ Decimal(μi))) * Decimal(F2)) - (
          (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi))) *
          ↪ Decimal(Q1))
    Summ1 += S
    Summ2 += F
    worksheet.write('B' + str(rowIndex), S)
    worksheet.write('D' + str(rowIndex), F)
    print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " = " +
          ↪ str(S))
    if n == C - 1:
        λ = λh + λn
        S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
              ↪ Decimal(ϑi)) / (
              (Decimal(n + 1)) * Decimal(μi))) *
              ↪ Decimal(S1)) - (
              (Decimal(l1) / ((Decimal(n + 1)) *
              ↪ Decimal(μi))) * Decimal(S2)) - (
              (Decimal(ϑj) / ((Decimal(n + 1)) *
              ↪ Decimal(μi))) * Decimal(L1))
        F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
              ↪ Decimal(ϑi)) / (
              (Decimal(n + 1)) * Decimal(μi))) *
              ↪ Decimal(F1)) - (
              (Decimal(l1) / ((Decimal(n + 1)) *
              ↪ Decimal(μi))) * Decimal(F2)) - (
              (Decimal(ϑj) / ((Decimal(n + 1)) *
              ↪ Decimal(μi))) * Decimal(Q1))

        A = S
        D = F
        X = λ
    elif n == C - 2:
        S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
              ↪ Decimal(ϑi)) / (
              (Decimal(n + 1)) * Decimal(μi))) *
              ↪ Decimal(S1)) - (

```



```

        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(F))
    F2 = F1
    F1 = F
    l1 = λ
    elif i == 0 and n >= ts and n < C:
        θi = θs
        j = 1 - i
        if j == 0:
            θj = θs
        else:
            θj = θm
        μi = μs
        λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
    ↪ ((Decimal(n + 1)) * Decimal(μi))) * Decimal(
        S1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
    ↪ ((Decimal(n + 1)) * Decimal(μi))) * Decimal(
        F1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(Q1))
    print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(S))
    Summ1 += S
    Summ2 += F
    worksheet.write('B' + str(rowIndex), S)
    worksheet.write('D' + str(rowIndex), F)
    if n == C - 1:
        λ = λh
        S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))

```

```

F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↳ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↳ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
            ↳ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↳ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↳ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↳ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
                ↳ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                    ↳ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↳ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↳ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
                ↳ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                    ↳ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
S2 = S1
S1 = S
print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↳ " + str(F))
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
        ↳ ((Decimal(C)) * Decimal(μi))) * Decimal(
            S1)) - ((Decimal(l1) / ((Decimal(C)) * Decimal(μi))) *
                ↳ Decimal(S2)) - (
                    (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi))) *
                        ↳ Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
        ↳ ((Decimal(C)) * Decimal(μi))) * Decimal(
            F1)) - ((Decimal(l1) / ((Decimal(C)) * Decimal(μi))) *
                ↳ Decimal(F2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(Q1))
print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(S))
Summ1 += S
Summ2 += F
worksheet.write('B' + str(rowIndex), S)
worksheet.write('D' + str(rowIndex), F)
if n == K + C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(C)) * Decimal(μi))) * Decimal(S1)) -
        ↪ (
            (Decimal(l1) / ((Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi)))
                ↪ * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(C)) * Decimal(μi))) * Decimal(F1)) -
        ↪ (
            (Decimal(l1) / ((Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi)))
                ↪ * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == K + C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(C)) * Decimal(μi))) * Decimal(S1)) -
        ↪ (
            (Decimal(l1) / ((Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi)))
                ↪ * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(C)) * Decimal(μi))) * Decimal(F1)) -
        ↪ (
            (Decimal(l1) / ((Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi)))
                ↪ * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(F))
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm

```

```

j = 1 - i
if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μm
k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ ((Decimal(n + 1)) * Decimal(μi))) *
    Decimal(L1)) - ((Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (
    Decimal(ϑj) / (Decimal((n + 1)) * Decimal(μi))) *
    ↪ Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ ((Decimal(n + 1)) * Decimal(μi))) *
    Decimal(Q1)) - ((Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q2)) - (
    Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F2))
Summ1 += L
Summ2 += Q
worksheet.write('C' + str(rowIndex), L)
worksheet.write('E' + str(rowIndex), Q)
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
                Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
            Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
                Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
    print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(L))
    L2 = L1
    L1 = L
    print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(Q))
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm

```

```

μi = μm
k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C) * Decimal(μi))) * Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(C) * Decimal(μi)))
    ↪ * Decimal(L2)) - (
    (Decimal(ϑj) / (Decimal(C) * Decimal(μi))) *
    ↪ Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C) * Decimal(μi))) * Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(C) * Decimal(μi)))
    ↪ * Decimal(Q2)) - (
    (Decimal(ϑj) / ((Decimal(C) * Decimal(μi))) *
    ↪ Decimal(F2))

Summ1 += L
Summ2 += Q
worksheet.write('C' + str(rowIndex), L)
print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(L))
L2 = L1
L1 = L
print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(Q))
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(ϑj) / (Decimal(C) * Decimal(μi))) *
        ↪ Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(ϑj) / ((Decimal(C) * Decimal(μi))) *
        ↪ Decimal(F2))

    Summ1 += L
    Summ2 += Q
    worksheet.write('C' + str(rowIndex), L)
    worksheet.write('E' + str(rowIndex), Q)
if n == K + C - 1:
    L = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(ϑj) / (Decimal(C) * Decimal(μi))) *
        ↪ Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(ϑj) / ((Decimal(C) * Decimal(μi)))
        ↪ * Decimal(F2))

    P = L
    G = Q

```

```

        print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
            ↳ " + str(L))
        L2 = L1
        L1 = L
        print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
            ↳ " + str(Q))
        Q2 = Q1
        Q1 = Q
        k1 = k
    rowIndex += 1
worksheet.set_column(1, 1, width=15)
worksheet.set_column(2, 2, width=15)
worksheet.set_column(3, 3, width=15)
worksheet.set_column(4, 4, width=15)
print("-----")
# calculation of P(0,0) & P(1,0) probabilities:
r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B - Decimal(ϑm)
    ↳ * P) / - (Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
    ↳ Decimal(ϑm) * G)
print("r = " + str(r))
print("-----")
P00 = 1 / (Summ1 + r * Summ2)
P10 = r * P00
rowIndex = 2
worksheet_2.write('B' + str(rowIndex), P00)
worksheet_2.write('C' + str(rowIndex), P10)
rowIndex = 3

# calculation of the rest of P(i,n) probabilities:
S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new = 0
Bm_new = 0
Summ_P0C = 0
Summ_P1C = 0
Summ_P0C_K = 0
Summ_P0_K_U = 0
Summ_P1_K_U = 0
Summ_NQ_0 = 0
Summ_NQ_1 = 0
Summ_λe_1 = 0
Summ_λe_2 = 0
Summ_λe_3 = 0

for n in range(0, C + K):
    worksheet_2.write('A' + str(rowIndex), n + 1)
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0C += n * Pin
            Summ_P0C_K += Pin
            Summ_λe_1 += (λh + λn) * Pin

```

```

print("P(" + str(i) + "," + str(n) + ") = " + str(Pin))
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
elif i == 1 and n == 0:
    Pin = L1 * P00 + Q1 * P10
    Summ_P1C += n * Pin
    print("P(" + str(i) + "," + str(n) + ") = " + str(Pin))
    Summ_λe_3 += λn * Pin
if i == 1 and n >= C - 1:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 0 and n <= ts - 1:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
    ↪ ((Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
    ↪ ((Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    worksheet_2.write('B' + str(rowIndex), Pin)
    print("P(" + str(i) + "," + str(n + 1) + ") = " + str(Pin))
    Summ_P0C += (n + 1) * Pin
    Summ_P0C_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U

```

```

    if i == 0 and n < ts - 1:
        Summ_λe_1 += λ * Pin
    if i==0 and n >= ts - 1 and n < C + K - 1:
        Summ_λe_2 += λh * Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    → ((Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
    → * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
    → * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    → ((Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
    → * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
    → * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    worksheet_2.write('B' + str(rowIndex), Pin)
    print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
    Summ_POC += (n + 1) * Pin
    Summ_POC_K += Pin
    if i == 0 and n >= ts - 1:
        Bs_new += Pin
    if i == 0 and n == C - 1:
        Bs_handoff = Pin
    if i == 0 and n == C + K - 1:
        P0_C_K = Pin
    if i == 0 and n >= C and n < C + K:
        P0_C_K_U = Pin
        Summ_P0_K_U += C * P0_C_K_U
        Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
    if i == 0 and n >= ts - 1 and n < C + K - 1:
        Summ_λe_2 += λh * Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= C and n < C + K:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:

```



```

     $\vartheta_j = \vartheta_m$ 
 $\mu_i = \mu_s$ 
 $\lambda = \lambda_h$ 
S = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
↪ ((Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(
S1)) - (Decimal(l1) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
↪ Decimal(S2)) - (
    (Decimal( $\vartheta_j$ ) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(L1))
F = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
↪ ((Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(
F1)) - (Decimal(l1) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
↪ Decimal(F2)) - (
    (Decimal( $\vartheta_j$ ) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
worksheet_2.write('B' + str(rowIndex), Pin)
print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
     $\vartheta_i = \vartheta_m$ 
    j = 1 - i
    if j == 0:
         $\vartheta_j = \vartheta_s$ 
    else:
         $\vartheta_j = \vartheta_m$ 
 $\mu_i = \mu_m$ 
k = λn
L = (((Decimal(k) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
↪ (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(L2)) - (
            (Decimal( $\vartheta_j$ ) / (Decimal((n + 1)) * Decimal( $\mu_i$ )))
            ↪ * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
↪ (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
    ↪ (

```

```

        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
print("P(" + str(i) + "," + str(n + 1) + ") = " + str(Pin))
Summ_P1C += (n + 1) * Pin
worksheet_2.write('C' + str(rowIndex), Pin)
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(L2)) - (
        (Decimal(θj) / (Decimal((C)) * Decimal(μi))) *
        ↪ Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
worksheet_2.write('C' + str(rowIndex), Pin)
print("P(" + str(i) + "," + str(n + 1) + ") = " + str(Pin))
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k

```

```

elif i == 1 and n > C and n < C + K:
     $\theta_i = \theta_m$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_m$ 
    k =  $\lambda_n$ 
    L = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(L1)) - (
        (Decimal( $\theta_j$ ) / (Decimal(C) * Decimal( $\mu_i$ ))) *
         $\leftrightarrow$  Decimal(S2))
    Q = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) - (
        (Decimal( $\theta_j$ ) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
         $\leftrightarrow$  Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    worksheet_2.write('C' + str(rowIndex), Pin)
    print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
    if i == 1 and n >= C - 1 and n < C + K:
        Bm_new += Pin
    if i == 1 and n >= C and n < C + K:
        P1_C_K_U = Pin
        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
    if i == 1 and n < C - 1:
        Summ_ $\lambda_e_3$  +=  $\lambda_n$  * Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
    rowIndex += 1
print("-----")

# calculation of total new call blocking probability:
Bnew = Bs_new + Bm_new
print("Bnew = " + str(Bnew))
print("Bs_new = " + str(Bs_new))
print("Bm_new = " + str(Bm_new))
print("-----")

# calculation of total handover call dropping probability:
P_HCDP = P0_C_K / Summ_P0C_K
print("P_HCDP = " + str(P_HCDP))
print("-----")

# calculation of channel utilization:
U = Summ_P0C + Summ_P1C
U2 = Summ_P0_K_U + Summ_P1_K_U
Utilization = (U + U2) / C
print("Utilization = " + str(Utilization))
print("-----")

# calculation of the average number of calls in the handover queue:
NQ = Summ_NQ_0 + Summ_NQ_1
print("NQ = " + str(NQ))

```

```

print("-----")

# calculation of the effective call arrival rate:
λe = Summ_λe_1 + Summ_λe_2 + Summ_λe_3
print("λe = " + str(λe))
print("-----")

# calculation of the average handover call waiting time in queue:
WQ = (NQ / λe) * 60
print("WQ = " + str(WQ))
rowIndex = 2
worksheet_3.write('A' + str(rowIndex), Bnew)
worksheet_3.write('B' + str(rowIndex), P_HCDP)
worksheet_3.write('C' + str(rowIndex), Utilization)
worksheet_3.write('D' + str(rowIndex), WQ)
print("-----")
worksheet_2.set_column(1, 1, width=15)
worksheet_2.set_column(2, 2, width=15)
worksheet_3.set_column(0, 0, width=23)
worksheet_3.set_column(1, 1, width=23)
worksheet_3.set_column(2, 2, width=23)
worksheet_3.set_column(3, 3, width=28)
workbook.close()

```

B.3 Πρόγραμμα Non Priority του Παραδείγματος 3.4

```

# Rate of the arrival process of New calls:
λn = 2
# Rate of the arrival process of Handover calls:
λh = 4
# Capacity of WLAN:
C = 6
# Rate of the call duration time in stop phase:
μs = 0.4
# Rate of the call duration time in moving phase:
μm = 0.25
# Rate of the dwell time in stop phase:
ϑs = 0.025
# Rate of the dwell time in moving phase:
ϑm = 0.02

l1 = 0
k1 = 0

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)

```

```

F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_NP = Decimal(0)
Summ_S1_in_NP = Decimal(0)

import xlswriter

workbook = xlswriter.Workbook('NP_C_6.xlsx')
bold_format = workbook.add_format({'bold': True})
bold_format.set_align('top')
bold_format.set_align('center')
bold_format.set_bg_color('green')
bold_format.set_border()
bold_format_2 = workbook.add_format({'bold': True})
bold_format_2.set_align('top')
bold_format_2.set_align('center')
bold_format_2.set_bg_color('blue')
bold_format_2.set_border()
bold_format_3 = workbook.add_format({'bold': True})
bold_format_3.set_align('top')
bold_format_3.set_align('center')
bold_format_3.set_bg_color('red')
bold_format_3.set_border()

# creation of the excel worksheet of S(i,n) contributors:
worksheet = workbook.add_worksheet('S(i,n)')
# creation of the excel worksheet of P(i,n) steady-state probabilities:
worksheet_2 = workbook.add_worksheet('P(i,n)')
# creation of the excel worksheet of NCBP, HCDP & Utilization:
worksheet_3 = workbook.add_worksheet('Results')

worksheet.write('A1', 'n', bold_format)
worksheet.write('B1', 'S0(0,n)', bold_format)
worksheet.write('C1', 'S0(1,n)', bold_format)
worksheet.write('D1', 'S1(0,n)', bold_format)
worksheet.write('E1', 'S1(1,n)', bold_format)
worksheet_2.write('A1', 'n', bold_format)
worksheet_2.write('B1', 'P(0,n)', bold_format)
worksheet_2.write('C1', 'P(1,n)', bold_format)
worksheet_3.write('A1', 'P(NCBP)', bold_format_3)
worksheet_3.write('B1', 'P(HCDP)', bold_format)
worksheet_3.write('C1', 'Utilization', bold_format_2)
rowIndex = 2
worksheet.write('A' + str(rowIndex), 0)
worksheet.write('B' + str(rowIndex), S1)
worksheet.write('C' + str(rowIndex), L1)
worksheet.write('D' + str(rowIndex), F1)
worksheet.write('E' + str(rowIndex), Q1)
worksheet_2.write('A' + str(rowIndex), 0)
rowIndex = 3

```

```

# calculation of S(i,n) contributors:
for n in range(0, C):
    worksheet.write('A' + str(rowIndex), n + 1)
    for i in range(0, 2):
        if i == 0 and n == 0:
            print("S\u2070" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  ↪ str(S1))
            print("S\u00B9" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  ↪ str(F1))
            Summ_S0_in_NP += S1
            Summ_S1_in_NP += F1
        elif i == 1 and n == 0:
            print("S\u2070" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  ↪ str(L1))
            print("S\u00B9" + "(" + str(i) + "," + str(n) + ")" + " = " +
                  ↪ str(Q1))
            Summ_S0_in_NP += L1
            Summ_S1_in_NP += Q1
        if i == 0 and n < C:
             $\vartheta_i = \vartheta_s$ 
            j = 1 - i
            if j == 0:
                 $\vartheta_j = \vartheta_s$ 
            else:
                 $\vartheta_j = \vartheta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
                  ↪ (
                    (Decimal(n + 1) * Decimal( $\mu_i$ )) * Decimal(S1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                    ↪ * Decimal(S2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                    ↪ * Decimal(L1))
                ))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
                  ↪ (
                    (Decimal(n + 1) * Decimal( $\mu_i$ )) * Decimal(F1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                    ↪ * Decimal(F2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                    ↪ * Decimal(Q1))
                ))
            Summ_S0_in_NP += S
            Summ_S1_in_NP += F
            worksheet.write('B' + str(rowIndex), S)
            worksheet.write('D' + str(rowIndex), F)
            print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " = " +
                  ↪ str(S))
        if n == C - 1:
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                  ↪ Decimal( $\vartheta_i$ )) / (
                    (Decimal(n + 1) * Decimal( $\mu_i$ )) * Decimal(S1)) -
                    ↪ (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                    ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
                    ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
                ))

```

```

F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
      (Decimal(n + 1)) * Decimal(μi))) * Decimal(F1)) -
↪ (
      (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2)) - (
      (Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
      (Decimal(n + 1)) * Decimal(μi))) * Decimal(S1)) -
↪ (
      (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(S2)) - (
      (Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
      (Decimal(n + 1)) * Decimal(μi))) * Decimal(F1)) -
↪ (
      (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2)) - (
      (Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
print ("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
↪ " + str(F))
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
θi = θm
j = 1 - i
if j == 0:
    θj = θs
else:
    θj = θm
μi = μm
k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
↪ (
      (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) - (
      (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
↪ * Decimal(L2)) - (
      (Decimal(θj) / (Decimal((n + 1)) * Decimal(μi)))
↪ * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
↪ (
      (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) - (
      (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
↪ * Decimal(Q2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(F2))
    Summ_S0_in_NP += L
    Summ_S1_in_NP += Q
    worksheet.write('C' + str(rowIndex), L)
    worksheet.write('E' + str(rowIndex), Q)
    if n == C - 1:
        L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
            ↪ Decimal(ϑi)) / (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
            ↪ (
                (Decimal(k1) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(L2)) - (
                    (Decimal(ϑj) / (Decimal((n + 1)) *
                        ↪ Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
            ↪ Decimal(ϑi)) / (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
            ↪ (
                (Decimal(k1) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(Q2)) - (
                    (Decimal(ϑj) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(F2))

        P = L
        G = Q
    print("S\u2070" + "(" + str(i) + ", " + str(n + 1) + ")" + " =
        ↪ " + str(L))
    L2 = L1
    L1 = L
    print("S\u00B9" + "(" + str(i) + ", " + str(n + 1) + ")" + " =
        ↪ " + str(Q))
    Q2 = Q1
    Q1 = Q
    k1 = k

    rowIndex += 1
worksheet.set_column(1, 1, width=15)
worksheet.set_column(2, 2, width=15)
worksheet.set_column(3, 3, width=15)
worksheet.set_column(4, 4, width=15)
print("-----")
# calculation of P(0,0) & P(1,0) probabilities:
r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B - Decimal(ϑm)
    ↪ * P) / - (
        Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
        ↪ Decimal(ϑm) * G)
print("r = " + str(r))
print("-----")
P00 = 1 / (Summ_S0_in_NP + r * Summ_S1_in_NP)
P10 = r * P00
rowIndex = 2
worksheet_2.write('B' + str(rowIndex), P00)
worksheet_2.write('C' + str(rowIndex), P10)
rowIndex = 3

# calculation of the rest of P(i,n) probabilities:
S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)

```



```

Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_NP = 0
Bm_new_NP = 0
Summ_P0n = 0
Summ_P1n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0

for n in range(0, C):
    worksheet_2.write('A' + str(rowIndex), n + 1)
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            print("P(" + str(i) + ", " + str(n) + ") = " + str(Pin))
            if i == 0 and n == C - 1:
                Bs_new_NP = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            Summ_P1n += Pin
            print("P(" + str(i) + ", " + str(n) + ") = " + str(Pin))
            if i == 1 and n == C - 1:
                Bm_new_NP = Pin
        if i == 0 and n < C:
            θi = θs
            j = 1 - i
            if j == 0:
                θj = θs
            else:
                θj = θm
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
                ↪ (
                    (Decimal(n + 1)) * Decimal(μi))) * Decimal(
                S1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
                ↪ * Decimal(S2)) - (
                    (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
                    ↪ * Decimal(L1))
            F = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
                ↪ (
                    (Decimal(n + 1)) * Decimal(μi))) * Decimal(
                F1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
                ↪ * Decimal(F2)) - (
                    (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
                    ↪ * Decimal(Q1))
            Pin = Decimal(S) * P00 + Decimal(F) * P10
            worksheet_2.write('B' + str(rowIndex), Pin)
            print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
            Summ_P0n += Pin
            Summ_P_n_0n += (n + 1) * Pin
            if i == 0 and n == C - 1:

```

```

        Bs_new_NP = Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
    elif i == 1 and n < C:
        ϑi = ϑm
        j = 1 - i
        if j == 0:
            ϑj = ϑs
        else:
            ϑj = ϑm
        μi = μm
        k = λn
        L = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
            → (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) - (
                (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
                → * Decimal(L2)) - (
                (Decimal(ϑj) / ((Decimal((n + 1)) * Decimal(μi)))
                → * Decimal(S2))
        Q = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
            → (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) - (
                (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
                → * Decimal(Q2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
                → * Decimal(F2))
        Pin = Decimal(L) * P00 + Decimal(Q) * P10
        print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
        Summ_P_n_1n += (n + 1) * Pin
        Summ_P1n += Pin
        worksheet_2.write('C' + str(rowIndex), Pin)
        if i == 1 and n == C - 1:
            Bm_new_NP = Pin
            L2 = L1
            L1 = L
            Q2 = Q1
            Q1 = Q
            k1 = k
    rowIndex += 1
print("-----")
print("Summ P(0,n) = " + str(Summ_P0n))
print("Summ P(1,n) = " + str(Summ_P1n))
print("-----")

# calculation of total new call blocking probability:
print("Bs_new(NP) = " + str(Bs_new_NP))
print("Bm_new(NP) = " + str(Bm_new_NP))
Bnew_NP = Bs_new_NP + Bm_new_NP
print("P_NCBP_NP = " + str(Bnew_NP))
print("-----")

# calculation of total handover call dropping probability:
P_HCDP_NP = Bs_new_NP / Summ_P0n
print("P_HCDP_NP = " + str(P_HCDP_NP))
print("-----")

```

```

# calculation of channel utilization:
U = Summ_P_n_0n + Summ_P_n_1n
Utilization_NP = U / C
print("Utilization = " + str(Utilization_NP))
print("-----")

rowIndex = 2
worksheet_3.write('A' + str(rowIndex), Bnew_NP)
worksheet_3.write('B' + str(rowIndex), P_HCDP_NP)
worksheet_3.write('C' + str(rowIndex), Utilization_NP)
worksheet_2.set_column(1, 1, width=15)
worksheet_2.set_column(2, 2, width=15)
worksheet_3.set_column(0, 0, width=23)
worksheet_3.set_column(1, 1, width=23)
worksheet_3.set_column(2, 2, width=23)
workbook.close()

```

B.4 Πρόγραμμα Cutoff Priority του Παραδείγματος 4.4

```

# Rate of the arrival process of New calls:
λn = 1
# Rate of the arrival process of Handover calls:
λh = 3
# Capacity of WLAN:
C = 5
# Number of shared channels:
ts = 2
# Rate of the call duration time in stop phase:
μs = 0.45
# Rate of the call duration time in moving phase:
μm = 0.3
# Rate of the dwell time in stop phase:
θs = 0.5
# Rate of the dwell time in moving phase:
θm = 0.4

l1 = 0
k1 = 0

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0

```

```

B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_CP = Decimal(0)
Summ_S1_in_CP = Decimal(0)

import xlswriter

workbook = xlswriter.Workbook('CP_C_5.xlsx')
bold_format = workbook.add_format({'bold': True})
bold_format.set_align('top')
bold_format.set_align('center')
bold_format.set_bg_color('green')
bold_format.set_border()
bold_format_2 = workbook.add_format({'bold': True})
bold_format_2.set_align('top')
bold_format_2.set_align('center')
bold_format_2.set_bg_color('blue')
bold_format_2.set_border()
bold_format_3 = workbook.add_format({'bold': True})
bold_format_3.set_align('top')
bold_format_3.set_align('center')
bold_format_3.set_bg_color('red')
bold_format_3.set_border()

# creation of the excel worksheet of S(i,n) contributors:
worksheet = workbook.add_worksheet('S(i,n)')
# creation of the excel worksheet of P(i,n) steady-state probabilities:
worksheet_2 = workbook.add_worksheet('P(i,n)')
# creation of the excel worksheet of NCBP, HCDP & Utilization:
worksheet_3 = workbook.add_worksheet('Results')

worksheet.write('A1', 'n', bold_format)
worksheet.write('B1', 'S0(0,n)', bold_format)
worksheet.write('C1', 'S0(1,n)', bold_format)
worksheet.write('D1', 'S1(0,n)', bold_format)
worksheet.write('E1', 'S1(1,n)', bold_format)
worksheet_2.write('A1', 'n', bold_format)
worksheet_2.write('B1', 'P(0,n)', bold_format)
worksheet_2.write('C1', 'P(1,n)', bold_format)
worksheet_3.write('A1', 'P(NCBP)', bold_format_3)
worksheet_3.write('B1', 'P(HCDP)', bold_format)
worksheet_3.write('C1', 'Utilization', bold_format_2)
rowIndex = 2
worksheet.write('A' + str(rowIndex), 0)
worksheet.write('B' + str(rowIndex), S1)
worksheet.write('C' + str(rowIndex), L1)
worksheet.write('D' + str(rowIndex), F1)
worksheet.write('E' + str(rowIndex), Q1)
worksheet_2.write('A' + str(rowIndex), 0)
rowIndex = 3

# calculation of S(i,n) contributors:
for n in range(0, C):
    worksheet.write('A' + str(rowIndex), n + 1)
    for i in range(0, 2):

```

```

if i == 0 and n == 0:
    print("S\u2070" + "(" + str(i) + "," + str(n) + ")" + " = " +
          ↪ str(S1))
    print("S\u00B9" + "(" + str(i) + "," + str(n) + ")" + " = " +
          ↪ str(F1))
    Summ_S0_in_CP += S1
    Summ_S1_in_CP += F1
elif i == 1 and n == 0:
    print("S\u2070" + "(" + str(i) + "," + str(n) + ")" + " = " +
          ↪ str(L1))
    print("S\u00B9" + "(" + str(i) + "," + str(n) + ")" + " = " +
          ↪ str(Q1))
    Summ_S0_in_CP += L1
    Summ_S1_in_CP += Q1
if i == 0 and n <= ts - 1:
     $\theta_i = \theta_s$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_s$ 
     $\lambda = \lambda_h + \lambda_n$ 
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) /
          ↪ (
                (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(S1)) - (
                (Decimal(l1) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                ↪ * Decimal(S2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                ↪ * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) /
          ↪ (
                (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(F1)) - (
                (Decimal(l1) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                ↪ * Decimal(F2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                ↪ * Decimal(Q1))
    Summ_S0_in_CP += S
    Summ_S1_in_CP += F
    worksheet.write('B' + str(rowIndex), S)
    worksheet.write('D' + str(rowIndex), F)
    print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " = "
          ↪ = " + str(S))
    if n == C - 1:
         $\lambda = \lambda_h + \lambda_n$ 
        S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
          ↪ Decimal( $\theta_i$ )) / (
                (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(S1)) -
          ↪ (
                (Decimal(l1) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
        F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
          ↪ Decimal( $\theta_i$ )) / (
                (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(F1)) -
          ↪ (

```

```

        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(S1)) -
        (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(F1)) -
        (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(F))
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(Q1))
    print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + "
    ↪ = " + str(S))

```

```

Summ_S0_in_CP += S
Summ_S1_in_CP += F
worksheet.write('B' + str(rowIndex), S)
worksheet.write('D' + str(rowIndex), F)
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    ↪ (Decimal(n + 1)) * Decimal(μi))) * Decimal(S1)) -
    ↪ (
    ↪ (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ ↪ Decimal(μi))) * Decimal(S2)) - (
    ↪ ↪ (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ ↪ ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    ↪ (Decimal(n + 1)) * Decimal(μi))) * Decimal(F1)) -
    ↪ (
    ↪ (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ ↪ Decimal(μi))) * Decimal(F2)) - (
    ↪ ↪ (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ ↪ ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    ↪ (Decimal(n + 1)) * Decimal(μi))) * Decimal(S1)) -
    ↪ (
    ↪ (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ ↪ Decimal(μi))) * Decimal(S2)) - (
    ↪ ↪ (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ ↪ ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    ↪ (Decimal(n + 1)) * Decimal(μi))) * Decimal(F1)) -
    ↪ (
    ↪ (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ ↪ Decimal(μi))) * Decimal(F2)) - (
    ↪ ↪ (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ ↪ ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(F))
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n <= ts - 1:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:

```

```

    ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) * Decimal(μi)))
        ↪ * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(F2))
    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
    worksheet.write('C' + str(rowIndex), L)
    worksheet.write('E' + str(rowIndex), Q)
    if n == C - 1:
        L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2))

        P = L
        G = Q
    print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(L))
    L2 = L1
    L1 = L
    print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(Q))
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == ts:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn

```



```

L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(L2)) - (
    (Decimal(θj) / (Decimal((n + 1)) * Decimal(μi)))
    ↪ * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(Q2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(F2))
Summ_S0_in_CP += L
Summ_S1_in_CP += Q
worksheet.write('C' + str(rowIndex), L)
worksheet.write('E' + str(rowIndex), Q)
print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(L))
L2 = L1
L1 = L
print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
    ↪ " + str(Q))
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > ts and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(θj) / (Decimal((n + 1)) * Decimal(μi)))
        ↪ * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(F2))
    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
    worksheet.write('C' + str(rowIndex), L)
    worksheet.write('E' + str(rowIndex), Q)
if n == C - 1:
    L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
    P = L

```

```

        G = Q
        print("S\u2070" + "(" + str(i) + "," + str(n + 1) + ")" + " =
        → " + str(L))
        L2 = L1
        L1 = L
        print("S\u00B9" + "(" + str(i) + "," + str(n + 1) + ")" + " =
        → " + str(Q))
        Q2 = Q1
        Q1 = Q
        k1 = k

    rowIndex += 1
    worksheet.set_column(1, 1, width=15)
    worksheet.set_column(2, 2, width=15)
    worksheet.set_column(3, 3, width=15)
    worksheet.set_column(4, 4, width=15)
print("-----")
# calculation of P(0,0) & P(1,0) probabilities:
r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B - Decimal(ϑm)
    → * P) / - (
        Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
        → Decimal(ϑm) * G)
print("r = " + str(r))
print("-----")
P00 = 1 / (Summ_S0_in_CP + r * Summ_S1_in_CP)
P10 = r * P00
rowIndex = 2
worksheet_2.write('B' + str(rowIndex), P00)
worksheet_2.write('C' + str(rowIndex), P10)
rowIndex = 3

# calculation of the rest of P(i,n) probabilities:
S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Summ_POS = 0
Summ_P1S = 0
Summ_P0n = 0
Summ_P1n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
POC = 0

for n in range(0, C):
    worksheet_2.write('A' + str(rowIndex), n + 1)
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            print("P(" + str(i) + "," + str(n) + ") = " + str(Pin))
            if i == 0 and n >= ts - 1 and n < C:
                Summ_POS += Pin
            if i == 0 and n == C - 1:

```

```

        POC = Pin
    elif i == 1 and n == 0:
        Pin = L1 * P00 + Q1 * P10
        Summ_P_n_1n += n * Pin
        Summ_P1n += Pin
        print("P(" + str(i) + ", " + str(n) + ") = " + str(Pin))
        if i == 1 and n >= ts - 1 and n < C:
            Summ_P1S += Pin
    if i == 0 and n <= ts - 1:
        θi = θs
        j = 1 - i
        if j == 0:
            θj = θs
        else:
            θj = θm
        μi = μs
        λ = λh + λn
        S = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
            ↪ (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(
                S1)) - (Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
            ↪ * Decimal(S2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
                ↪ * Decimal(L1))
        F = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
            ↪ (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(
                F1)) - (Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
            ↪ * Decimal(F2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
                ↪ * Decimal(Q1))
        Pin = Decimal(S) * P00 + Decimal(F) * P10
        worksheet_2.write('B' + str(rowIndex), Pin)
        print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
        Summ_P0n += Pin
        Summ_P_n_0n += (n + 1) * Pin
        if i == 0 and n >= ts - 1 and n < C:
            Summ_P0S += Pin
        if i == 0 and n == C - 1:
            POC = Pin
        S2 = S1
        S1 = S
        F2 = F1
        F1 = F
        l1 = λ
    elif i == 0 and n >= ts and n < C:
        θi = θs
        j = 1 - i
        if j == 0:
            θj = θs
        else:
            θj = θm
        μi = μs
        λ = λh
        S = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(θi)) /
            ↪ (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(

```

```

        S1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
            ↪ * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
        F1)) - ((Decimal(l1) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
            ↪ * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
worksheet_2.write('B' + str(rowIndex), Pin)
print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n >= ts - 1 and n < C:
    Summ_P0S += Pin
if i == 0 and n == C - 1:
    P0C = Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n <= ts - 1:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(L2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
            ↪ * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) + Decimal(ϑi)) /
    ↪ (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
            ↪ * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
Summ_P_n_1n += (n + 1) * Pin
Summ_P1n += Pin
worksheet_2.write('C' + str(rowIndex), Pin)
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1

```

```

Q1 = Q
k1 = k
elif i == 1 and n == ts:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((n + 1)) * Decimal(μi)))
        ↪ * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
    Summ_P_n_1n += (n + 1) * Pin
    Summ_P1n += Pin
    worksheet_2.write('C' + str(rowIndex), Pin)
    if i == 1 and n >= ts - 1 and n < C:
        Summ_P1S += Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n > ts and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(θj) / (Decimal((n + 1)) * Decimal(μi)))
        ↪ * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
        ↪ * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    print("P(" + str(i) + ", " + str(n + 1) + ") = " + str(Pin))
    Summ_P_n_1n += (n + 1) * Pin
    Summ_P1n += Pin
    worksheet_2.write('C' + str(rowIndex), Pin)
    if i == 1 and n >= ts - 1 and n < C:

```

```

        Summ_P1S += Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
    rowIndex += 1
print("-----")
print("Summ P(0,n) = " + str(Summ_P0n))
print("Summ P(1,n) = " + str(Summ_P1n))
print("-----")

# calculation of total new call blocking probability:
Bnew_CP = Summ_P0S + Summ_P1S
print("P_NCBP_CP = " + str(Bnew_CP))
print("-----")

# calculation of total handover call dropping probability:
P_HCDP_CP = P0C / Summ_P0n
print("P_HCDP_CP = " + str(P_HCDP_CP))
print("-----")

# calculation of channel utilization:
U = Summ_P_n_0n + Summ_P_n_1n
Utilization_CP = U / C
print("Utilization = " + str(Utilization_CP))
print("-----")
rowIndex = 2
worksheet_3.write('A' + str(rowIndex), Bnew_CP)
worksheet_3.write('B' + str(rowIndex), P_HCDP_CP)
worksheet_3.write('C' + str(rowIndex), Utilization_CP)
worksheet_2.set_column(1, 1, width=15)
worksheet_2.set_column(2, 2, width=15)
worksheet_3.set_column(0, 0, width=23)
worksheet_3.set_column(1, 1, width=23)
worksheet_3.set_column(2, 2, width=23)
workbook.close()

```

Παράρτημα Γ΄

Προγράμματα σε Γλώσσα Python των Επιδράσεων των Παραμέτρων στα Μοντέλα Απωλειών

Γ΄.1 Πρόγραμμα των Επιδράσεων του λ_H

```
list_λh = []
list_λh_2 = []
list_Bnew = []
list_Bnew_2 = []
list_Bnew_NP = []
list_Bnew_CP = []
list_HC DP = []
list_HC DP_2 = []
list_HC DP_NP = []
list_HC DP_CP = []
list_Util = []
list_Util_2 = []
list_Util_NP = []
list_Util_CP = []

cnt = 1
while cnt <= 13:
    # Rate of the arrival process of Handover calls:
    λh = cnt * 4
    cnt = cnt + 1
    # Rate of the arrival process of New calls:
    λn = 24
    # Capacity of WLAN:
    C = 70
    # Number of shared channels:
    ts = 56
    # Rate of the call duration time in stop phase:
    μs = 1 / 5
    # Rate of the call duration time in moving phase:
    μm = 1 / 5
    # Rate of the dwell time in stop phase:
    ϑs = 1 / 4
```

```

# Rate of the dwell time in moving phase:
θm = 1 / 5
# Handover queue size:
for K in range(0, 14 + 1, 14):
    l1 = 0
    k1 = 0
    S2 = 0
    S1 = 1
    l1 = 0
    k1 = 0
# MA-CAC Model
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ1 = Decimal(0)
Summ2 = Decimal(0)

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ1 += S1
            Summ2 += F1
        elif i == 1 and n == 0:
            Summ1 += L1
            Summ2 += Q1
        if i == 0 and n <= ts - 1:
            θi = θs
            j = 1 - i
            if j == 0:
                θj = θs
            else:
                θj = θm
            μi = μs
            λ = λh + λn

```



```

S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
↪ *
    Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
↪ *
    Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
Summ1 += S
Summ2 += F
if n == C - 1:
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

```

```

        B = S
        E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ *
        Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ *
        Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

    Summ1 += S
    Summ2 += F
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) *
    ↪ Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) *
    ↪ Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    λ = λh

```

```

S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
↪ Decimal( $\theta$ i)) / (
    (Decimal(n + 1)) * Decimal( $\mu$ i))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal( $\mu$ i))) *
    ↪ Decimal(S2)) - (
    (Decimal( $\theta$ j) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu$ i))) * Decimal(L1))
F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
↪ Decimal( $\theta$ i)) / (
    (Decimal(n + 1)) * Decimal( $\mu$ i))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal( $\mu$ i))) *
    ↪ Decimal(F2)) - (
    (Decimal( $\theta$ j) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu$ i))) * Decimal(Q1))

B = S
E = F

S2 = S1
S1 = S
F2 = F1
F1 = F
l1 =  $\lambda$ 
elif i == 0 and n >= C and n < C + K:
     $\theta$ i =  $\theta$ s
    j = 1 - i
    if j == 0:
         $\theta$ j =  $\theta$ s
    else:
         $\theta$ j =  $\theta$ m
     $\mu$ i =  $\mu$ s
     $\lambda$  =  $\lambda$ h
    S = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu$ i) +
    ↪ Decimal( $\theta$ i)) / ((Decimal(C)) * Decimal( $\mu$ i))) *
        Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal( $\mu$ i))) * Decimal(S2)) - (
        (Decimal( $\theta$ j) / ((Decimal(C)) *
        ↪ Decimal( $\mu$ i))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu$ i) +
    ↪ Decimal( $\theta$ i)) / ((Decimal(C)) * Decimal( $\mu$ i))) *
        Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal( $\mu$ i))) * Decimal(F2)) - (
        (Decimal( $\theta$ j) / ((Decimal(C)) *
        ↪ Decimal( $\mu$ i))) * Decimal(Q1))

Summ1 += S
Summ2 += F
if n == K + C - 1:
     $\lambda$  =  $\lambda$ h
    S = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu$ i) +
    ↪ Decimal( $\theta$ i)) / (
        (Decimal(C)) * Decimal( $\mu$ i))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal( $\mu$ i))) * Decimal(S2))
        ↪ - (

```

```

        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))
        ↪ - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == K + C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2))
        ↪ - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))
        ↪ - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ *
        Decimal(L1)) - ((Decimal(k1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2))

```

```

Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
↪ *
    Decimal(Q1)) - ((Decimal(k1) / ((Decimal(n + 1))
↪ * Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2))
Summ1 += L
Summ2 += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1))
↪ * Decimal(μi))) *
↪ Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1))
↪ * Decimal(μi))) *
↪ Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C)) * Decimal(μi))) *
↪ Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(C)) *
↪ Decimal(μi))) * Decimal(L2)) - (
    (Decimal(ϑj) / (Decimal((C)) *
↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C)) * Decimal(μi))) *
↪ Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(C)) *
↪ Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(ϑj) / ((Decimal(C)) *
↪ Decimal(μi))) * Decimal(F2))

```

```

Summ1 += L
Summ2 += Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
        (Decimal(ϑj) / (Decimal(C) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))

    Summ1 += L
    Summ2 += Q
    if n == K + C - 1:
        L = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
        ↪ (
            (Decimal(C)) * Decimal(μi))) *
            ↪ Decimal(L1)) - (
            (Decimal(ϑj) / (Decimal(C) *
            ↪ Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
        ↪ (
            (Decimal(C)) * Decimal(μi))) *
            ↪ Decimal(Q1)) - (
            (Decimal(ϑj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(F2))

        P = L
        G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k

r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    ↪ Decimal(ϑm) * P) / - (Decimal(C) * Decimal(μs) *
    D + Decimal(ϑs) * D - X * E - Decimal(ϑm) * G)
P00 = 1 / (Summ1 + r * Summ2)
P10 = r * P00

from decimal import *

ctx = getcontext()
ctx.prec = 50

```

```

ctx.rounding = ROUND_UP
S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new = 0
Bm_new = 0
Summ_P0C = 0
Summ_P1C = 0
Summ_P0C_K = 0
Summ_P0_K_U = 0
Summ_P1_K_U = 0
Summ_NQ_0 = 0
Summ_NQ_1 = 0
Summ_λe_1 = 0
Summ_λe_2 = 0
Summ_λe_3 = 0

for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0C += n * Pin
            Summ_P0C_K += Pin
            Summ_λe_1 += (λh + λn) * Pin
            if i == 0 and n >= ts - 1 and n < C + K:
                Bs_new += Pin
            if i == 0 and n == C - 1:
                Bs_handoff = Pin
            if i == 0 and n == C + K - 1:
                P0_C_K = Pin
            if i == 0 and n >= C and n < C + K:
                P0_C_K_U = Pin
                Summ_P0_K_U += C * P0_C_K_U
                Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P1C += n * Pin
            Summ_λe_3 += λn * Pin
            if i == 1 and n >= C - 1:
                Bm_new += Pin
            if i == 1 and n >= C and n < C + K:
                P1_C_K_U = Pin
                Summ_P1_K_U += C * P1_C_K_U
                Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
        if i == 0 and n <= ts - 1:
            ϑi = ϑs
            j = 1 - i
            if j == 0:
                ϑj = ϑs
            else:
                ϑj = ϑm
            μi = μs
            λ = λh + λn

```

```

S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
→ *
    Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
→ * Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
→ *
    Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
→ * Decimal(μi))) * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0C += (n + 1) * Pin
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n < ts - 1:
    Summ_λe_1 += λ * Pin
if i==0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
→ *
    Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
→ * Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
→ *
    Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
→ * Decimal(μi))) * Decimal(F2)) - (

```



```

                                (Decimal( $\theta_j$ ) / ((Decimal( $n + 1$ )) *
                                 $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal( $Q_1$ ))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0C += ( $n + 1$ ) * Pin
Summ_P0C_K += Pin
if  $i == 0$  and  $n \geq ts - 1$ :
    Bs_new += Pin
if  $i == 0$  and  $n == C - 1$ :
    Bs_handoff = Pin
if  $i == 0$  and  $n == C + K - 1$ :
    P0_C_K = Pin
if  $i == 0$  and  $n \geq C$  and  $n < C + K$ :
    P0_C_K_U = Pin
    Summ_P0_K_U +=  $C * P0_C_K_U$ 
    Summ_NQ_0 += ( $n + 1 - C$ ) * P0_C_K_U
if  $i == 0$  and  $n \geq ts - 1$  and  $n < C + K - 1$ :
    Summ_λe_2 +=  $\lambda_h * Pin$ 
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 =  $\lambda$ 
elif  $i == 0$  and  $n \geq C$  and  $n < C + K$ :
     $\theta_i = \theta_s$ 
     $j = 1 - i$ 
    if  $j == 0$ :
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_s$ 
     $\lambda = \lambda_h$ 
    S = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu_i$ ) +
     $\hookrightarrow$  Decimal( $\theta_i$ ) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
    Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
     $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(S2)) - (
    (Decimal( $\theta_j$ ) / ((Decimal(C)) *
     $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(l1))
    F = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu_i$ ) +
     $\hookrightarrow$  Decimal( $\theta_i$ ) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
    Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
     $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(F2)) - (
    (Decimal( $\theta_j$ ) / ((Decimal(C)) *
     $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal( $Q_1$ ))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    Summ_P0C_K += Pin
    if  $i == 0$  and  $n \geq ts - 1$  and  $n < C + K$ :
        Bs_new += Pin
    if  $i == 0$  and  $n == C - 1$ :
        Bs_handoff = Pin
    if  $i == 0$  and  $n == C + K - 1$ :
        P0_C_K = Pin
    if  $i == 0$  and  $n \geq C$  and  $n < C + K$ :
        P0_C_K_U = Pin
        Summ_P0_K_U +=  $C * P0_C_K_U$ 
        Summ_NQ_0 += ( $n + 1 - C$ ) * P0_C_K_U
    if  $i == 0$  and  $n \geq ts - 1$  and  $n < C + K - 1$ :
        Summ_λe_2 +=  $\lambda_h * Pin$ 
S2 = S1

```

```

S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    Summ_P1C += (n + 1) * Pin
    if i == 1 and n >= C - 1 and n < C + K:
        Bm_new += Pin
    if i == 1 and n >= C and n < C + K:
        P1_C_K_U = Pin
        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
    if i == 1 and n < C - 1:
        Summ_λe_3 += λn * Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(L2)) - (

```

```

        (Decimal( $\theta_j$ ) / (Decimal(C)) *
        ↪ Decimal( $\mu_i$ )) * Decimal(S2))
Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(C)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(Q2)) - (
        (Decimal( $\theta_j$ ) / ((Decimal(C)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
     $\theta_i$  =  $\theta_m$ 
    j = 1 - i
if j == 0:
     $\theta_j$  =  $\theta_s$ 
else:
     $\theta_j$  =  $\theta_m$ 
 $\mu_i$  =  $\mu_m$ 
k = λn
L = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
    ↪ (
        (Decimal( $\theta_j$ ) / (Decimal(C)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
Q = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
    ↪ (
        (Decimal( $\theta_j$ ) / ((Decimal(C)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
Bnew = Bs_new + Bm_new
U = Summ_P0C + Summ_P1C

```

```

P_HCDP = P0_C_K / Summ_P0C_K
U2 = Summ_P0_K_U + Summ_P1_K_U
Utilization = (U + U2) / C
NQ = Summ_NQ_0 + Summ_NQ_1
λe = Summ_λe_1 + Summ_λe_2 + Summ_λe_3
WQ = (NQ / λe) * 60
if K == 0:
    list_λh.append(λh)
    list_Bnew.append(Bnew)
    list_HCDP.append(P_HCDP)
    list_Util.append(Utilization)
    print("H lista λh einai: " + str(list_λh))
    print("H lista Bnew_MA-CAC (K=0) einai: " + str(list_Bnew))
    print("H lista HCDP_MA-CAC (K=0) einai: " + str(list_HCDP))
    print("H lista Util_MA-CAC (K=0) einai: " + str(list_Util))
elif K == 14:
    list_λh_2.append(λh)
    list_Bnew_2.append(Bnew)
    list_HCDP_2.append(P_HCDP)
    list_Util_2.append(Utilization)
    print("H lista λh einai: " + str(list_λh))
    print("H lista Bnew_MA-CAC (K=14) einai: " +
        → str(list_Bnew_2))
    print("H lista HCDP_MA-CAC (K=14) einai: " +
        → str(list_HCDP_2))
    print("H lista Util_MA-CAC (K=14) einai: " +
        → str(list_Util_2))
    → print("-----")
# NP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_NP = Decimal(0)
Summ_S1_in_NP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ_S0_in_NP += S1
            Summ_S1_in_NP += F1

```

```

elif i == 1 and n == 0:
    Summ_S0_in_NP += L1
    Summ_S1_in_NP += Q1
if i == 0 and n < C:
     $\theta_i = \theta_s$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_s$ 
     $\lambda = \lambda_h + \lambda_n$ 
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
        (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
        (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
    Summ_S0_in_NP += S
    Summ_S1_in_NP += F
if n == C - 1:
     $\lambda = \lambda_h + \lambda_n$ 
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
        (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
        (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))

    A = S
    D = F
    X =  $\lambda$ 
elif n == C - 2:
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

Summ_S0_in_NP += L
Summ_S1_in_NP += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (

```

```

(Decimal(k1) / ((Decimal(n + 1)) *
↳ Decimal( $\mu_i$ ))) * Decimal(Q2)) - (
(Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
↳ Decimal( $\mu_i$ ))) * Decimal(F2))

P = L
G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k

r = (Decimal(C) * Decimal( $\mu_s$ ) * A + Decimal( $\vartheta_s$ ) * A - X * B -
↳ Decimal( $\vartheta_m$ ) * P) / - (
Decimal(C) * Decimal( $\mu_s$ ) * D + Decimal( $\vartheta_s$ ) * D - X * E -
↳ Decimal( $\vartheta_m$ ) * G)

P00 = 1 / (Summ_S0_in_NP + r * Summ_S1_in_NP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_NP = 0
Bm_new_NP = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n == C - 1:
                Bs_new = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n == C - 1:
                Bm_new = Pin
        if i == 0 and n < C:
             $\vartheta_i$  =  $\vartheta_s$ 
            j = 1 - i
            if j == 0:
                 $\vartheta_j$  =  $\vartheta_s$ 
            else:
                 $\vartheta_j$  =  $\vartheta_m$ 
             $\mu_i$  =  $\mu_s$ 
             $\lambda$  =  $\lambda_h$  +  $\lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
↳ Decimal( $\vartheta_i$ )) / (
(Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↳ Decimal( $\mu_i$ ))) * Decimal(S2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n == C - 1:
    Bs_new_NP = Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    Summ_P_n_1n += (n + 1) * Pin
    if i == 1 and n == C - 1:
        Bm_new_NP = Pin
        L2 = L1
        L1 = L
        Q2 = Q1
        Q1 = Q
        k1 = k
Bnew_NP = Bs_new_NP + Bm_new_NP
U = Summ_P_n_0n + Summ_P_n_1n
P_HCDP_NP = Bs_new_NP / Summ_P0n
Utilization_NP = U / C
list_Bnew_NP.append(Bnew_NP)

```



```

list_HCDP_NP.append(P_HCDP_NP)
list_Util_NP.append(Utilization_NP)
print("H lista λh einai: " + str(list_λh))
print("H lista Bnew_NP einai: " + str(list_Bnew_NP))
print("H lista HCDP_NP einai: " + str(list_HCDP_NP))
print("H lista Util_NP einai: " + str(list_Util_NP))
print("-----")
# CP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_CP = Decimal(0)
Summ_S1_in_CP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ_S0_in_CP += S1
            Summ_S1_in_CP += F1
        elif i == 1 and n == 0:
            Summ_S0_in_CP += L1
            Summ_S1_in_CP += Q1
        if i == 0 and n <= ts - 1:
            θi = θs
            j = 1 - i
            if j == 0:
                θj = θs
            else:
                θj = θm
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(θi)) / (
                    (Decimal(n + 1)) * Decimal(μi))) *
                ↪ Decimal(S1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(S2)) - (
                    (Decimal(θj) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(L1))
            F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(θi)) / (

```

```

        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Summ_S0_in_CP += S
Summ_S1_in_CP += F
if n == C - 1:
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs

```

```

else:
     $\vartheta_j = \vartheta_m$ 
     $\mu_i = \mu_s$ 
     $\lambda = \lambda_h$ 
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\vartheta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
    S1)) - (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\vartheta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
    F1)) - (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
    Summ_S0_in_CP += S
    Summ_S1_in_CP += F
    if n == C - 1:
         $\lambda = \lambda_h$ 
        S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
        ↪ Decimal( $\vartheta_i$ )) / (
            (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
            ↪ Decimal(S1)) - (
                (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
        F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
        ↪ Decimal( $\vartheta_i$ )) / (
            (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
            ↪ Decimal(F1)) - (
                (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))

        A = S
        D = F
        X =  $\lambda$ 
    elif n == C - 2:
         $\lambda = \lambda_h$ 
        S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
        ↪ Decimal( $\vartheta_i$ )) / (
            (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
            ↪ Decimal(S1)) - (
                (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
        F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
        ↪ Decimal( $\vartheta_i$ )) / (
            (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
            ↪ Decimal(F1)) - (
                (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))

```

```

                                (Decimal(ϑj) / ((Decimal(n + 1)) *
                                ↪ Decimal(μi))) * Decimal(Q1))
    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n <= ts - 1:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
                (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))
    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
    if n == C - 1:
        L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
            ↪ Decimal(L1)) - (
                (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L2)) - (
                    (Decimal(ϑj) / (Decimal((n + 1)) *
                    ↪ Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
            ↪ Decimal(Q1)) - (
                (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q2)) - (
                    (Decimal(ϑj) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
    L2 = L1
    L1 = L
    Q2 = Q1

```

```

Q1 = Q
k1 = k
elif i == 1 and n == ts:
     $\theta_i = \theta_m$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_m$ 
    k =  $\lambda_n$ 
    L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(L2)) - (
                (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
                    ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(Q2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                    ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n > ts and n < C:
     $\theta_i = \theta_m$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_m$ 
    k =  $\lambda_n$ 
    L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
        ↪ (
            (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
        ↪ (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
if n == C - 1:
    L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(L1)) - (

```

```

        (Decimal(ϑj) / (Decimal((n + 1)) *
        → Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        → Decimal(Q1)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        → Decimal(μi))) * Decimal(F2))

P = L
G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k

r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    → Decimal(ϑm) * P) / - (
        Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
        → Decimal(ϑm) * G)
P00 = 1 / (Summ_S0_in_CP + r * Summ_S1_in_CP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_CP = 0
Bm_new_CP = 0
Summ_POS = 0
Summ_P1S = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
POC = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n >= ts - 1 and n < C:
                Summ_POS += Pin
            if i == 0 and n == C - 1:
                POC = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n >= ts - 1 and n < C:
                Summ_P1S += Pin
        if i == 0 and n <= ts - 1:
            ϑi = ϑs
            j = 1 - i
            if j == 0:
                ϑj = ϑs
            else:

```

```

         $\vartheta_j = \vartheta_m$ 
 $\mu_i = \mu_s$ 
 $\lambda = \lambda_h + \lambda_n$ 
S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
↪ Decimal( $\vartheta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(L1))
F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
↪ Decimal( $\vartheta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n >= ts - 1 and n < C:
    Summ_POS += Pin
if i == 0 and n == C - 1:
    P0C = Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 =  $\lambda$ 
elif i == 0 and n >= ts and n < C:
 $\vartheta_i = \vartheta_s$ 
j = 1 - i
if j == 0:
     $\vartheta_j = \vartheta_s$ 
else:
     $\vartheta_j = \vartheta_m$ 
 $\mu_i = \mu_s$ 
 $\lambda = \lambda_h$ 
S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
↪ Decimal( $\vartheta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(L1))
F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
↪ Decimal( $\vartheta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n >= ts - 1 and n < C:
    Summ_POS += Pin
if i == 0 and n == C - 1:

```

```

        P0C = Pin
        S2 = S1
        S1 = S
        F2 = F1
        F1 = F
        l1 = λ
    elif i == 1 and n <= ts - 1:
        ϑi = ϑm
        j = 1 - i
        if j == 0:
            ϑj = ϑs
        else:
            ϑj = ϑm
        μi = μm
        k = λn
        L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
            → Decimal(ϑi)) / (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
            → (
                (Decimal(k1) / ((Decimal(n + 1)) *
                    → Decimal(μi))) * Decimal(L2)) - (
                    (Decimal(ϑj) / (Decimal((n + 1)) *
                        → Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
            → Decimal(ϑi)) / (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
            → (
                (Decimal(k1) / ((Decimal(n + 1)) *
                    → Decimal(μi))) * Decimal(Q2)) - (
                    (Decimal(ϑj) / ((Decimal(n + 1)) *
                        → Decimal(μi))) * Decimal(F2))
        Pin = Decimal(L) * P00 + Decimal(Q) * P10
        Summ_P_n_1n += (n + 1) * Pin
        if i == 1 and n >= ts - 1 and n < C:
            Summ_P1S += Pin
        L2 = L1
        L1 = L
        Q2 = Q1
        Q1 = Q
        k1 = k
    elif i == 1 and n == ts:
        ϑi = ϑm
        j = 1 - i
        if j == 0:
            ϑj = ϑs
        else:
            ϑj = ϑm
        μi = μm
        k = λn
        L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
            → (
                (Decimal(k1) / ((Decimal(n + 1)) *
                    → Decimal(μi))) * Decimal(L2)) - (
                    (Decimal(ϑj) / (Decimal((n + 1)) *
                        → Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (

```



```

        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(Q2)) - (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > ts and n < C:
     $\theta_i$  =  $\theta_m$ 
    j = 1 - i
    if j == 0:
         $\theta_j$  =  $\theta_s$ 
    else:
         $\theta_j$  =  $\theta_m$ 
     $\mu_i$  =  $\mu_m$ 
    k =  $\lambda_n$ 
L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
    ↪ (
        (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
    ↪ (
        (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
Bnew_CP = Summ_P0S + Summ_P1S
U = Summ_P_n_0n + Summ_P_n_1n
P_HCDP_CP = P0C / Summ_P0n
Utilization_CP = U / C
list_Bnew_CP.append(Bnew_CP)
list_HCDP_CP.append(P_HCDP_CP)
list_Util_CP.append(Utilization_CP)
print("H lista  $\lambda_h$  einai: " + str(list_ $\lambda_h$ ))
print("H lista Bnew_CP einai: " + str(list_Bnew_CP))
print("H lista HCDP_CP einai: " + str(list_HCDP_CP))
print("H lista Util_CP einai: " + str(list_Util_CP))

print("-----")
import matplotlib.pyplot as plt
import numpy as np

```

```

plt.figure(1)
plt.title('NCBP versus  $\lambda$ ')
plt.ylabel('New call blocking probability')
plt.xlabel('Handover call arrival rate ( $\lambda$ )')
x1 = np.array(list_λh)
y1 = np.array(list_Bnew)
y2 = np.array(list_Bnew_2)
y3 = np.array(list_Bnew_NP)
y4 = np.array(list_Bnew_CP)
plt.xticks(np.arange(0, 56, step=5))
plt.yticks(np.arange(0, 1, step=0.1))
plt.xlim(0, 55)
plt.ylim(0, 0.9)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y3, '^~r', ms = 11, label = 'NP')
plt.plot(x1, y4, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.figure(2)
plt.title('HCDP versus  $\lambda$ ')
plt.ylabel('Handover call dropping probability')
plt.xlabel('Handover arrival rate ( $\lambda$ )')
x1 = np.array(list_λh)
y1 = np.array(list_HCDP)
y2 = np.array(list_HCDP_2)
y3 = np.array(list_HCDP_NP)
y4 = np.array(list_HCDP_CP)
plt.xticks(np.arange(0, 56, step=5))
plt.yticks(np.arange(0, 1, step=0.1))
plt.xlim(0, 55)
plt.ylim(0, 0.9)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y3, '^~r', ms = 11, label = 'NP')
plt.plot(x1, y4, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.figure(3)
plt.title('Channel utilization versus  $\lambda$ ')
plt.ylabel('Channel utilization')
plt.xlabel('Handover arrival rate ( $\lambda$ )')
x1 = np.array(list_λh)
y1 = np.array(list_Util)
y2 = np.array(list_Util_2)
y3 = np.array(list_Util_NP)
y4 = np.array(list_Util_CP)
plt.xticks(np.arange(0, 56, step=5))
plt.yticks(np.arange(0.7, 1.1, step=0.05))
plt.xlim(0, 55)
plt.ylim(0.7, 1)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)

```

```

plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y3, '^-r', ms = 11, label = 'NP')
plt.plot(x1, y4, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.show()

```

Γ'.2 Πρόγραμμα των Επιδράσεων του λ_N

```

list_lambda_n = []
list_lambda_n_2 = []
list_Bnew = []
list_Bnew_2 = []
list_Bnew_NP = []
list_Bnew_CP = []
list_HC DP = []
list_HC DP_2 = []
list_HC DP_NP = []
list_HC DP_CP = []
list_Util = []
list_Util_2 = []
list_Util_NP = []
list_Util_CP = []

cnt = 1
while cnt <= 13:
    # Rate of the arrival process of New calls:
    lambda_n = cnt * 4
    cnt = cnt + 1
    # Rate of the arrival process of Handover calls:
    lambda_h = 28
    # Capacity of WLAN:
    C = 70
    # Number of shared channels:
    ts = 56
    # Rate of the call duration time in stop phase:
    mu_s = 1/5
    # Rate of the call duration time in moving phase:
    mu_m = 1/5
    # Rate of the dwell time in stop phase:
    theta_s = 1/4
    # Rate of the dwell time in moving phase:
    theta_m = 1/5
    # Handover queue size:
    for K in range(0, 14 + 1, 14):
        l1 = 0
        k1 = 0
        S2 = 0
        S1 = 1
        l1 = 0
        k1 = 0
# MA-CAC Model
from decimal import *

```

```

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ1 = Decimal(0)
Summ2 = Decimal(0)

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ1 += S1
            Summ2 += F1
        elif i == 1 and n == 0:
            Summ1 += L1
            Summ2 += Q1
        if i == 0 and n <= ts - 1:
             $\theta_i = \theta_s$ 
            j = 1 - i
            if j == 0:
                 $\theta_j = \theta_s$ 
            else:
                 $\theta_j = \theta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
            ↪ Decimal( $\theta_i$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
            * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
            ↪ 1)) * Decimal( $\mu_i$ ))) * Decimal(S2)) - (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
            ↪ Decimal( $\theta_i$ ) / ((Decimal(n + 1))
            * Decimal( $\mu_i$ ))) * Decimal(F1)) - ((Decimal(l1) /
            ↪ ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
            * Decimal(F2)) - ((Decimal( $\theta_j$ ) / ((Decimal(n +
            ↪ 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1))
            Summ1 += S
            Summ2 += F

```

```

if n == C - 1:
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal(μi))) *
            ↪ Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal(μi))) *
            ↪ Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal(μi))) *
            ↪ Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal(μi))) *
            ↪ Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh

```

```

S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
→ *
    Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
→ 1)) * Decimal(μi))) * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
→ *
    Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
→ * Decimal(μi))) * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(Q1))

Summ1 += S
Summ2 += F
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
→ * Decimal(μi))) *
→ Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
→ * Decimal(μi))) *
→ Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
→ * Decimal(μi))) *
→ Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
→ * Decimal(μi))) *
→ Decimal(F2)) - (

```

```

                                (Decimal( $\theta_j$ ) / ((Decimal( $n + 1$ )) *
                                ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 =  $\lambda$ 
elif i == 0 and n >= C and n < C + K:
     $\theta_i$  =  $\theta_s$ 
    j = 1 - i
    if j == 0:
         $\theta_j$  =  $\theta_s$ 
    else:
         $\theta_j$  =  $\theta_m$ 
     $\mu_i$  =  $\mu_s$ 
     $\lambda$  =  $\lambda_h$ 
    S = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ ) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
    Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
    (Decimal( $\theta_j$ ) / ((Decimal(C)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ ) / ((Decimal(C)) * Decimal( $\mu_i$ ))) *
    Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
    (Decimal( $\theta_j$ ) / ((Decimal(C)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))

    Summ1 += S
    Summ2 += F
if n == K + C - 1:
     $\lambda$  =  $\lambda_h$ 
    S = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ ) / (
    (Decimal(C)) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
    ↪ - (
    (Decimal( $\theta_j$ ) / ((Decimal(C)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(C) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ ) / (
    (Decimal(C)) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
    ↪ - (
    (Decimal( $\theta_j$ ) / ((Decimal(C)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))

    A = S
    D = F
    X =  $\lambda$ 
elif n == K + C - 2:
     $\lambda$  =  $\lambda_h$ 

```

```

S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↳ Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
    ↳ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(C)) *
            ↳ Decimal(μi))) * Decimal(S2))
    ↳ - (
        (Decimal(θj) / ((Decimal(C)) *
            ↳ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↳ Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
    ↳ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(C)) *
            ↳ Decimal(μi))) * Decimal(F2))
    ↳ - (
        (Decimal(θj) / ((Decimal(C)) *
            ↳ Decimal(μi))) * Decimal(Q1))

B = S
E = F

S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm

μi = μm
k = λn

L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↳ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    ↳ *
        Decimal(L1)) - ((Decimal(k1) / ((Decimal(n + 1))
            ↳ * Decimal(μi))) * Decimal(L2)) - (
            (Decimal(θj) / (Decimal((n + 1)) *
                ↳ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↳ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    ↳ *
        Decimal(Q1)) - ((Decimal(k1) / ((Decimal(n + 1))
            ↳ * Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
                ↳ Decimal(μi))) * Decimal(F2))

Summ1 += L
Summ2 += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↳ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↳ Decimal(L1)) - (

```



```

        (Decimal(k1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(L2)) - (
        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

P = L
G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (
    (Decimal(θj) / (Decimal((C)) *
    ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(θj) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(F2))

Summ1 += L
Summ2 += Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm

```

```

μi = μm
k = λn
L = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C)) * Decimal(μi))) *
    → Decimal(L1)) - (
    (Decimal(ϑj) / (Decimal(C) *
    → Decimal(μi))) * Decimal(S2))
Q = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C)) * Decimal(μi))) *
    → Decimal(Q1)) - (
    (Decimal(ϑj) / ((Decimal(C)) *
    → Decimal(μi))) * Decimal(F2))

Summ1 += L
Summ2 += Q
if n == K + C - 1:
    L = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
        → (
            (Decimal(C)) * Decimal(μi))) *
            → Decimal(L1)) - (
                (Decimal(ϑj) / (Decimal(C) *
                → Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) /
        → (
            (Decimal(C)) * Decimal(μi))) *
            → Decimal(Q1)) - (
                (Decimal(ϑj) / ((Decimal(C)) *
                → Decimal(μi))) * Decimal(F2))

    P = L
    G = Q

    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k

r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    → Decimal(ϑm) * P) / - (Decimal(C) * Decimal(μs) *
    D + Decimal(ϑs) * D - X * E - Decimal(ϑm) * G)
P00 = 1 / (Summ1 + r * Summ2)
P10 = r * P00

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new = 0
Bm_new = 0
Summ_P0C = 0
Summ_P1C = 0

```

```

Summ_P0C_K = 0
Summ_P0_K_U = 0
Summ_P1_K_U = 0
Summ_NQ_0 = 0
Summ_NQ_1 = 0
Summ_λe_1 = 0
Summ_λe_2 = 0
Summ_λe_3 = 0

for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0C += n * Pin
            Summ_P0C_K += Pin
            Summ_λe_1 += (λh + λn) * Pin
            if i == 0 and n >= ts - 1 and n < C + K:
                Bs_new += Pin
            if i == 0 and n == C - 1:
                Bs_handoff = Pin
            if i == 0 and n == C + K - 1:
                P0_C_K = Pin
            if i == 0 and n >= C and n < C + K:
                P0_C_K_U = Pin
                Summ_P0_K_U += C * P0_C_K_U
                Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P1C += n * Pin
            Summ_λe_3 += λn * Pin
            if i == 1 and n >= C - 1:
                Bm_new += Pin
            if i == 1 and n >= C and n < C + K:
                P1_C_K_U = Pin
                Summ_P1_K_U += C * P1_C_K_U
                Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
        if i == 0 and n <= ts - 1:
            θi = θs
            j = 1 - i
            if j == 0:
                θj = θs
            else:
                θj = θm
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
                ↪ *
                ↪ Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal(μi))) * Decimal(S2)) - (
                ↪ (Decimal(θj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
            F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
                ↪ *
                ↪ Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal(μi))) * Decimal(F2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0C += (n + 1) * Pin
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n < ts - 1:
    Summ_λe_1 += λ * Pin
if i==0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= ts and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ *
    Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(S2)) - (
    Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ *
    Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(F2)) - (
    Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0C += (n + 1) * Pin
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin

```

```

        Summ_P0_K_U += C * P0_C_K_U
        Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(θi)) / ((Decimal(C)) * Decimal(μi))) *
        Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(θj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(l1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(θi)) / ((Decimal(C)) * Decimal(μi))) *
        Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(θj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    Summ_P0C_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn

```

```

L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L2)) - (
    (Decimal(θj) / (Decimal((n + 1)) *
→ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P1C += (n + 1) * Pin
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn

L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
→ (
    (Decimal(k1) / ((Decimal(C)) *
→ Decimal(μi))) * Decimal(L2)) - (
    (Decimal(θj) / (Decimal((C)) *
→ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) -
→ (
    (Decimal(k1) / ((Decimal(C)) *
→ Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(θj) / ((Decimal(C)) *
→ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin

```

```

    if i == 1 and n >= C and n < C + K:
        P1_C_K_U = Pin
        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
    if i == 1 and n < C - 1:
        Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(θj) / (Decimal(C) *
            ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(θj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
Bnew = Bs_new + Bm_new
U = Summ_P0C + Summ_P1C
P_HCDP = P0_C_K / Summ_P0C_K
U2 = Summ_P0_K_U + Summ_P1_K_U
Utilization = (U + U2) / C
NQ = Summ_NQ_0 + Summ_NQ_1
λe = Summ_λe_1 + Summ_λe_2 + Summ_λe_3
WQ = (NQ / λe) * 60
if K == 0:
    list_λn.append(λn)
    list_Bnew.append(Bnew)
    list_HCDP.append(P_HCDP)
    list_Util.append(Utilization)
print("H lista λn einai: " + str(list_λn))

```

```

    print("H lista Bnew_MA-CAC (K=0) einai: " + str(list_Bnew))
    print("H lista HCDP_MA-CAC (K=0) einai: " + str(list_HCDP))
    print("H lista Util_MA-CAC (K=0) einai: " + str(list_Util))
elif K == 14:
    list_λn_2.append(λn)
    list_Bnew_2.append(Bnew)
    list_HCDP_2.append(P_HCDP)
    list_Util_2.append(Utilization)
    print("H lista λn einai: " + str(list_λn))
    print("H lista Bnew_MA-CAC (K=14) einai: " +
        → str(list_Bnew_2))
    print("H lista HCDP_MA-CAC (K=14) einai: " +
        → str(list_HCDP_2))
    print("H lista Util_MA-CAC (K=14) einai: " +
        → str(list_Util_2))

    → print("-----")
# NP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_NP = Decimal(0)
Summ_S1_in_NP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ_S0_in_NP += S1
            Summ_S1_in_NP += F1
        elif i == 1 and n == 0:
            Summ_S0_in_NP += L1
            Summ_S1_in_NP += Q1
        if i == 0 and n < C:
            ϑi = ϑs
            j = 1 - i
            if j == 0:
                ϑj = ϑs
            else:
                ϑj = ϑm
            μi = μs
            λ = λh + λn

```



```

S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
↪ Decimal( $\vartheta$ i)) / (
    (Decimal(n + 1)) * Decimal( $\mu$ i))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu$ i))) * Decimal(S2)) - (
    (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu$ i))) * Decimal(L1))
F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
↪ Decimal( $\vartheta$ i)) / (
    (Decimal(n + 1)) * Decimal( $\mu$ i))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu$ i))) * Decimal(F2)) - (
    (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu$ i))) * Decimal(Q1))
Summ_S0_in_NP += S
Summ_S1_in_NP += F
if n == C - 1:
     $\lambda$  =  $\lambda_h$  +  $\lambda_n$ 
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
    ↪ Decimal( $\vartheta$ i)) / (
        (Decimal(n + 1)) * Decimal( $\mu$ i))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu$ i))) * Decimal(S2)) - (
        (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu$ i))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
    ↪ Decimal( $\vartheta$ i)) / (
        (Decimal(n + 1)) * Decimal( $\mu$ i))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu$ i))) * Decimal(F2)) - (
        (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu$ i))) * Decimal(Q1))

    A = S
    D = F
    X =  $\lambda$ 
elif n == C - 2:
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
    ↪ Decimal( $\vartheta$ i)) / (
        (Decimal(n + 1)) * Decimal( $\mu$ i))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu$ i))) * Decimal(S2)) - (
        (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu$ i))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
    ↪ Decimal( $\vartheta$ i)) / (
        (Decimal(n + 1)) * Decimal( $\mu$ i))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu$ i))) * Decimal(F2)) - (
        (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu$ i))) * Decimal(Q1))

    B = S
    E = F

```

```

S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))
    Summ_S0_in_NP += L
    Summ_S1_in_NP += Q
    if n == C - 1:
        L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
            ↪ Decimal(ϑi)) / (
                (Decimal(n + 1)) * Decimal(μi))) *
            ↪ Decimal(L1)) - (
                (Decimal(k1) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(L2)) - (
                (Decimal(ϑj) / (Decimal((n + 1)) *
                    ↪ Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
            ↪ Decimal(ϑi)) / (
                (Decimal(n + 1)) * Decimal(μi))) *
            ↪ Decimal(Q1)) - (
                (Decimal(k1) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(Q2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(F2))

        P = L
        G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    ↪ Decimal(ϑm) * P) / - (

```

```

        Decimal(C) * Decimal( $\mu$ s) * D + Decimal( $\vartheta$ s) * D - X * E -
        ↪ Decimal( $\vartheta$ m) * G)
P00 = 1 / (Summ_S0_in_NP + r * Summ_S1_in_NP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_NP = 0
Bm_new_NP = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n == C - 1:
                Bs_new = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n == C - 1:
                Bm_new = Pin
        if i == 0 and n < C:
             $\vartheta$ i =  $\vartheta$ s
            j = 1 - i
            if j == 0:
                 $\vartheta$ j =  $\vartheta$ s
            else:
                 $\vartheta$ j =  $\vartheta$ m
             $\mu$ i =  $\mu$ s
             $\lambda$  =  $\lambda$ h +  $\lambda$ n
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i)) +
            ↪ Decimal( $\vartheta$ i)) / (
                (Decimal(n + 1)) * Decimal( $\mu$ i))) * Decimal(
            S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu$ i))) * Decimal(S2)) - (
                (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu$ i))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i)) +
            ↪ Decimal( $\vartheta$ i)) / (
                (Decimal(n + 1)) * Decimal( $\mu$ i))) * Decimal(
            F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu$ i))) * Decimal(F2)) - (
                (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu$ i))) * Decimal(Q1))
            Pin = Decimal(S) * P00 + Decimal(F) * P10
            Summ_P0n += Pin
            Summ_P_n_0n += (n + 1) * Pin
            if i == 0 and n == C - 1:

```

```

        Bs_new_NP = Pin
        S2 = S1
        S1 = S
        F2 = F1
        F1 = F
        l1 = λ
    elif i == 1 and n < C:
        ϑi = ϑm
        j = 1 - i
        if j == 0:
            ϑj = ϑs
        else:
            ϑj = ϑm
        μi = μm
        k = λn
        L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
            → Decimal(ϑi)) / (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
            → (
                (Decimal(k1) / ((Decimal(n + 1)) *
                    → Decimal(μi))) * Decimal(L2)) - (
                    (Decimal(ϑj) / (Decimal((n + 1)) *
                        → Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
            → Decimal(ϑi)) / (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
            → (
                (Decimal(k1) / ((Decimal(n + 1)) *
                    → Decimal(μi))) * Decimal(Q2)) - (
                    (Decimal(ϑj) / ((Decimal(n + 1)) *
                        → Decimal(μi))) * Decimal(F2))
        Pin = Decimal(L) * P00 + Decimal(Q) * P10
        Summ_P_n_0n += (n + 1) * Pin
        if i == 1 and n == C - 1:
            Bm_new_NP = Pin
            L2 = L1
            L1 = L
            Q2 = Q1
            Q1 = Q
            k1 = k
    Bnew_NP = Bs_new_NP + Bm_new_NP
    U = Summ_P_n_0n + Summ_P_n_1n
    P_HCDP_NP = Bs_new_NP / Summ_P0n
    Utilization_NP = U / C
    list_Bnew_NP.append(Bnew_NP)
    list_HCDP_NP.append(P_HCDP_NP)
    list_Util_NP.append(Utilization_NP)
    print("H lista λn einai: " + str(list_λn))
    print("H lista Bnew_NP einai: " + str(list_Bnew_NP))
    print("H lista HCDP_NP einai: " + str(list_HCDP_NP))
    print("H lista Util_NP einai: " + str(list_Util_NP))
    print("-----")
# CP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
```

```

ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_CP = Decimal(0)
Summ_S1_in_CP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ_S0_in_CP += S1
            Summ_S1_in_CP += F1
        elif i == 1 and n == 0:
            Summ_S0_in_CP += L1
            Summ_S1_in_CP += Q1
        if i == 0 and n <= ts - 1:
             $\theta_i = \theta_s$ 
            j = 1 - i
            if j == 0:
                 $\theta_j = \theta_s$ 
            else:
                 $\theta_j = \theta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                 $\hookrightarrow$  Decimal( $\theta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                 $\hookrightarrow$  Decimal(S1)) - (
                    (Decimal(L1) / ((Decimal(n + 1)) *
                         $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                    (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                         $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                 $\hookrightarrow$  Decimal( $\theta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                 $\hookrightarrow$  Decimal(F1)) - (
                    (Decimal(L1) / ((Decimal(n + 1)) *
                         $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                    (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                         $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(Q1))
            Summ_S0_in_CP += S
            Summ_S1_in_CP += F
        if n == C - 1:
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                 $\hookrightarrow$  Decimal( $\theta_i$ )) / (

```

```

        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))

```

```

F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
↪ Decimal( $\vartheta$ i)) / (
      (Decimal(n + 1)) * Decimal( $\mu$ i))) * Decimal(
F1)) - (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(F2)) - (
      (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(Q1))
Summ_S0_in_CP += S
Summ_S1_in_CP += F
if n == C - 1:
     $\lambda$  =  $\lambda$ h
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
↪ Decimal( $\vartheta$ i)) / (
      (Decimal(n + 1)) * Decimal( $\mu$ i))) *
↪ Decimal(S1)) - (
      (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(S2)) - (
      (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
↪ Decimal( $\vartheta$ i)) / (
      (Decimal(n + 1)) * Decimal( $\mu$ i))) *
↪ Decimal(F1)) - (
      (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(F2)) - (
      (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(Q1))

    A = S
    D = F
    X =  $\lambda$ 
elif n == C - 2:
     $\lambda$  =  $\lambda$ h
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
↪ Decimal( $\vartheta$ i)) / (
      (Decimal(n + 1)) * Decimal( $\mu$ i))) *
↪ Decimal(S1)) - (
      (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(S2)) - (
      (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu$ i) +
↪ Decimal( $\vartheta$ i)) / (
      (Decimal(n + 1)) * Decimal( $\mu$ i))) *
↪ Decimal(F1)) - (
      (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(F2)) - (
      (Decimal( $\vartheta$ j) / ((Decimal(n + 1)) *
↪ Decimal( $\mu$ i))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 =  $\lambda$ 
elif i == 1 and n <= ts - 1:
     $\vartheta$ i =  $\vartheta$ m
    j = 1 - i

```

```

if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μm
k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))
Summ_S0_in_CP += L
Summ_S1_in_CP += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == ts:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (

```



```

        (Decimal(n + 1)) * Decimal( $\mu_i$ )) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ )) * Decimal(L2)) - (
            (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
            ↪ Decimal( $\mu_i$ )) * Decimal(S2))
Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ )) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ )) * Decimal(Q2)) - (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ )) * Decimal(F2))
Summ_S0_in_CP += L
Summ_S1_in_CP += Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > ts and n < C:
     $\theta_i$  =  $\theta_m$ 
    j = 1 - i
    if j == 0:
         $\theta_j$  =  $\theta_s$ 
    else:
         $\theta_j$  =  $\theta_m$ 
     $\mu_i$  =  $\mu_m$ 
    k =  $\lambda_n$ 
L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ )) * Decimal(L1)) -
        ↪ (
            (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
            ↪ Decimal( $\mu_i$ )) * Decimal(S2))
Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ )) * Decimal(Q1)) -
        ↪ (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ )) * Decimal(F2))
Summ_S0_in_CP += L
Summ_S1_in_CP += Q
if n == C - 1:
    L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ )) *
        ↪ Decimal(L1)) - (
            (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
            ↪ Decimal( $\mu_i$ )) * Decimal(S2))
    Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ )) *
        ↪ Decimal(Q1)) - (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ )) * Decimal(F2))

    P = L
    G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q

```

```

k1 = k
r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    ↪ Decimal(ϑm) * P) / - (
    Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
    ↪ Decimal(ϑm) * G)
P00 = 1 / (Summ_S0_in_CP + r * Summ_S1_in_CP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_CP = 0
Bm_new_CP = 0
Summ_P0S = 0
Summ_P1S = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
POC = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n >= ts - 1 and n < C:
                Summ_P0S += Pin
            if i == 0 and n == C - 1:
                POC = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n >= ts - 1 and n < C:
                Summ_P1S += Pin
        if i == 0 and n <= ts - 1:
            ϑi = ϑs
            j = 1 - i
            if j == 0:
                ϑj = ϑs
            else:
                ϑj = ϑm
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(ϑi)) / (
                    (Decimal(n + 1)) * Decimal(μi))) * Decimal(
                    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(S2)) - (
                    (Decimal(ϑj) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(L1))
            F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(ϑi)) / (
                    (Decimal(n + 1)) * Decimal(μi))) * Decimal(

```

```

        F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n >= ts - 1 and n < C:
    Summ_POS += Pin
if i == 0 and n == C - 1:
    POC = Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh

S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n >= ts - 1 and n < C:
    Summ_POS += Pin
if i == 0 and n == C - 1:
    POC = Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n <= ts - 1:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:

```

```

    ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    Summ_P_n_1n += (n + 1) * Pin
    if i == 1 and n >= ts - 1 and n < C:
        Summ_P1S += Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == ts:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    Summ_P_n_1n += (n + 1) * Pin
    if i == 1 and n >= ts - 1 and n < C:
        Summ_P1S += Pin
    L2 = L1
    L1 = L
    Q2 = Q1

```

```

    Q1 = Q
    k1 = k
    elif i == 1 and n > ts and n < C:
         $\theta_i = \theta_m$ 
        j = 1 - i
        if j == 0:
             $\theta_j = \theta_s$ 
        else:
             $\theta_j = \theta_m$ 
         $\mu_i = \mu_m$ 
        k =  $\lambda_n$ 
        L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
            (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
            ↪ (
                (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
                    ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
        Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
            (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
            ↪ (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                    ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
        Pin = Decimal(L) * P00 + Decimal(Q) * P10
        Summ_P_n_1n += (n + 1) * Pin
        if i == 1 and n >= ts - 1 and n < C:
            Summ_P1S += Pin
        L2 = L1
        L1 = L
        Q2 = Q1
        Q1 = Q
        k1 = k
    Bnew_CP = Summ_P0S + Summ_P1S
    U = Summ_P_n_0n + Summ_P_n_1n
    P_HCDP_CP = P0C / Summ_P0n
    Utilization_CP = U / C
    list_Bnew_CP.append(Bnew_CP)
    list_HCDP_CP.append(P_HCDP_CP)
    list_Util_CP.append(Utilization_CP)
    print("H lista  $\lambda_n$  einai: " + str(list_λn))
    print("H lista Bnew_CP einai: " + str(list_Bnew_CP))
    print("H lista HCDP_CP einai: " + str(list_HCDP_CP))
    print("H lista Util_CP einai: " + str(list_Util_CP))
    print("-----")

```

```

import matplotlib.pyplot as plt
import numpy as np

```

```

plt.figure(4)
plt.title('NCBP versus  $\lambda$ \u0274')
plt.ylabel('New call blocking probability')
plt.xlabel('New call arrival rate ( $\lambda$ \u0274)')
x1 = np.array(list_λn)
y1 = np.array(list_Bnew)
y2 = np.array(list_Bnew_2)
y3 = np.array(list_Bnew_NP)
y4 = np.array(list_Bnew_CP)
plt.xticks(np.arange(0, 56, step=5))
plt.yticks(np.arange(0, 1, step=0.1))
plt.xlim(0, 55)

```

```

plt.ylim(0, 0.9)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y3, '^~r', ms = 11, label = 'NP')
plt.plot(x1, y4, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.figure(5)
plt.title('HCDP versus  $\lambda$ \u0274')
plt.ylabel('Handover call dropping probability')
plt.xlabel('New call arrival rate ( $\lambda$ \u0274)')
x1 = np.array(list_λn)
y1 = np.array(list_HCDP)
y2 = np.array(list_HCDP_2)
y3 = np.array(list_HCDP_NP)
y4 = np.array(list_HCDP_CP)
plt.xticks(np.arange(0, 56, step=5))
plt.yticks(np.arange(0, 1, step=0.1))
plt.xlim(0, 55)
plt.ylim(0, 0.9)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y3, '^~r', ms = 11, label = 'NP')
plt.plot(x1, y4, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.figure(6)
plt.title('Channel utilization versus  $\lambda$ \u0274')
plt.ylabel('Channel utilization')
plt.xlabel('New call arrival rate ( $\lambda$ \u0274)')
x1 = np.array(list_λn)
y1 = np.array(list_Util)
y2 = np.array(list_Util_2)
y3 = np.array(list_Util_NP)
y4 = np.array(list_Util_CP)
plt.xticks(np.arange(0, 56, step=5))
plt.yticks(np.arange(0.7, 1.1, step=0.05))
plt.xlim(0, 55)
plt.ylim(0.7, 1)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y3, '^~r', ms = 11, label = 'NP')
plt.plot(x1, y4, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.show()

```

Γ.3 Πρόγραμμα των Επιδράσεων του t_S

```

list_ts = []
list_ts_2 = []
list_Bnew = []
list_Bnew_2 = []
list_Bnew_NP = []
list_Bnew_CP = []
list_HC DP = []
list_HC DP_2 = []
list_HC DP_NP = []
list_HC DP_CP = []
list_Util = []
list_Util_2 = []
list_Util_NP = []
list_Util_CP = []

cnt = 1
while cnt <= 17:
    # Number of shared channels:
    ts = cnt * 4
    cnt = cnt + 1
    # Rate of the arrival process of New calls:
    λn = 24
    # Capacity of WLAN:
    C = 70
    # Rate of the arrival process of Handover calls:
    λh = 28
    # Rate of the call duration time in stop phase:
    μs = 1 / 5
    # Rate of the call duration time in moving phase:
    μm = 1 / 5
    # Rate of the dwell time in stop phase:
    θs = 1 / 4
    # Rate of the dwell time in moving phase:
    θm = 1 / 5
    # Handover queue size:
    for K in range(0, 14 + 1, 14):
        l1 = 0
        k1 = 0
        S2 = 0
        S1 = 1
        l1 = 0
        k1 = 0
# MA-CAC Model
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)

```

```

Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ1 = Decimal(0)
Summ2 = Decimal(0)

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ1 += S1
            Summ2 += F1
        elif i == 1 and n == 0:
            Summ1 += L1
            Summ2 += Q1
        if i == 0 and n <= ts - 1:
             $\vartheta_i = \vartheta_s$ 
            j = 1 - i
            if j == 0:
                 $\vartheta_j = \vartheta_s$ 
            else:
                 $\vartheta_j = \vartheta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
            ↪ Decimal( $\vartheta_i$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
            ↪ *
            Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
            ↪ * Decimal( $\mu_i$ ))) * Decimal(S2)) - (
            (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
            ↪ Decimal( $\vartheta_i$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
            ↪ *
            Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
            ↪ * Decimal( $\mu_i$ ))) * Decimal(F2)) - (
            (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
            Summ1 += S
            Summ2 += F
        if n == C - 1:
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
            ↪ Decimal( $\vartheta_i$ ) / (
            (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
            ↪ Decimal(S1)) - (

```



```

        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ

elif i == 0 and n >= ts and n < C:
θi = θs
j = 1 - i
if j == 0:
    θj = θs
else:
    θj = θm
μi = μs
λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
↪ *
        Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) * Decimal(S2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
↪ *
    Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Summ1 += S
Summ2 += F
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S

```

```

F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(θi)) / ((Decimal(C)) * Decimal(μi))) *
        Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(θi)) / ((Decimal(C)) * Decimal(μi))) *
        Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))
    Summ1 += S
    Summ2 += F
if n == K + C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2))
        ↪ - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))
        ↪ - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == K + C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2))
        ↪ - (

```

```

        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))
        ↪ - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
↪ *
        Decimal(L1)) - ((Decimal(k1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
↪ *
        Decimal(Q1)) - ((Decimal(k1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

Summ1 += L
Summ2 += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (

```

```

                                (Decimal(k1) / ((Decimal(n + 1))
                                ↪ * Decimal(μi))) *
                                ↪ Decimal(Q2)) - (
                                (Decimal(θj) / ((Decimal(n + 1)) *
                                ↪ Decimal(μi))) * Decimal(F2))
    P = L
    G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((C)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))

    Summ1 += L
    Summ2 += Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n > C and n < C + K:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
    L = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
        (Decimal(θj) / (Decimal(C) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (

```

```

        (Decimal(C)) * Decimal(μi)) *
        ↳ Decimal(Q1)) - (
        (Decimal(θj) / ((Decimal(C)) *
        ↳ Decimal(μi))) * Decimal(F2))
Summ1 += L
Summ2 += Q
if n == K + C - 1:
    L = (((Decimal(C) * Decimal(μi) + Decimal(θi)) /
        ↳ (
            (Decimal(C)) * Decimal(μi)) *
            ↳ Decimal(L1)) - (
                (Decimal(θj) / (Decimal(C) *
                ↳ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(θi)) /
        ↳ (
            (Decimal(C)) * Decimal(μi)) *
            ↳ Decimal(Q1)) - (
                (Decimal(θj) / ((Decimal(C)) *
                ↳ Decimal(μi))) * Decimal(F2))
    P = L
    G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
r = (Decimal(C) * Decimal(μs) * A + Decimal(θs) * A - X * B -
    ↳ Decimal(θm) * P) / - (Decimal(C) *
        Decimal(μs) * D + Decimal(θs) * D - X * E - Decimal(θm) * G)
P00 = 1 / (Summ1 + r * Summ2)
P10 = r * P00

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new = 0
Bm_new = 0
Summ_P0C = 0
Summ_P1C = 0
Summ_P0C_K = 0
Summ_P0_K_U = 0
Summ_P1_K_U = 0
Summ_NQ_0 = 0
Summ_NQ_1 = 0
Summ_λe_1 = 0
Summ_λe_2 = 0
Summ_λe_3 = 0

```

```

for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0C += n * Pin
            Summ_P0C_K += Pin
            Summ_λe_1 += (λh + λn) * Pin
            if i == 0 and n >= ts - 1 and n < C + K:
                Bs_new += Pin
            if i == 0 and n == C - 1:
                Bs_handoff = Pin
            if i == 0 and n == C + K - 1:
                P0_C_K = Pin
            if i == 0 and n >= C and n < C + K:
                P0_C_K_U = Pin
                Summ_P0_K_U += C * P0_C_K_U
                Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P1C += n * Pin
            Summ_λe_3 += λn * Pin
            if i == 1 and n >= C - 1:
                Bm_new += Pin
            if i == 1 and n >= C and n < C + K:
                P1_C_K_U = Pin
                Summ_P1_K_U += C * P1_C_K_U
                Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
        if i == 0 and n <= ts - 1:
            θi = θs
            j = 1 - i
            if j == 0:
                θj = θs
            else:
                θj = θm
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
                ↪ *
                ↪ Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal(μi))) * Decimal(S2)) - (
                ↪ (Decimal(θj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
            F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
                ↪ *
                ↪ Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal(μi))) * Decimal(F2)) - (
                ↪ (Decimal(θj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))
            Pin = Decimal(S) * P00 + Decimal(F) * P10
            Summ_P0C += (n + 1) * Pin
            Summ_P0C_K += Pin
            if i == 0 and n >= ts - 1 and n < C + K:
                Bs_new += Pin
            if i == 0 and n == C - 1:
                Bs_handoff = Pin
            if i == 0 and n == C + K - 1:

```

```

    P0_C_K = Pin
    if i == 0 and n >= C and n < C + K:
        P0_C_K_U = Pin
        Summ_P0_K_U += C * P0_C_K_U
        Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
    if i == 0 and n < ts - 1:
        Summ_λe_1 += λ * Pin
    if i==0 and n >= ts - 1 and n < C + K - 1:
        Summ_λe_2 += λh * Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ *
    ↪ Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(S2)) - (
    ↪ (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ *
    ↪ Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) * Decimal(F2)) - (
    ↪ (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    Summ_P0C += (n + 1) * Pin
    Summ_P0C_K += Pin
    if i == 0 and n >= ts - 1:
        Bs_new += Pin
    if i == 0 and n == C - 1:
        Bs_handoff = Pin
    if i == 0 and n == C + K - 1:
        P0_C_K = Pin
    if i == 0 and n >= C and n < C + K:
        P0_C_K_U = Pin
        Summ_P0_K_U += C * P0_C_K_U
        Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
    if i == 0 and n >= ts - 1 and n < C + K - 1:
        Summ_λe_2 += λh * Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= C and n < C + K:

```



```

 $\vartheta_i = \vartheta_s$ 
 $j = 1 - i$ 
if  $j == 0$ :
     $\vartheta_j = \vartheta_s$ 
else:
     $\vartheta_j = \vartheta_m$ 
 $\mu_i = \mu_s$ 
 $\lambda = \lambda_h$ 
 $S = ((\text{Decimal}(\lambda) + \text{Decimal}(C) * \text{Decimal}(\mu_i) +$ 
     $\hookrightarrow \text{Decimal}(\vartheta_i)) / ((\text{Decimal}(C)) * \text{Decimal}(\mu_i))) *$ 
     $\text{Decimal}(S1)) - ((\text{Decimal}(l1) / ((\text{Decimal}(C)) *$ 
     $\hookrightarrow \text{Decimal}(\mu_i))) * \text{Decimal}(S2)) - ($ 
     $(\text{Decimal}(\vartheta_j) / ((\text{Decimal}(C)) *$ 
     $\hookrightarrow \text{Decimal}(\mu_i))) * \text{Decimal}(L1))$ 
 $F = ((\text{Decimal}(\lambda) + \text{Decimal}(C) * \text{Decimal}(\mu_i) +$ 
     $\hookrightarrow \text{Decimal}(\vartheta_i)) / ((\text{Decimal}(C)) * \text{Decimal}(\mu_i))) *$ 
     $\text{Decimal}(F1)) - ((\text{Decimal}(l1) / ((\text{Decimal}(C)) *$ 
     $\hookrightarrow \text{Decimal}(\mu_i))) * \text{Decimal}(F2)) - ($ 
     $(\text{Decimal}(\vartheta_j) / ((\text{Decimal}(C)) *$ 
     $\hookrightarrow \text{Decimal}(\mu_i))) * \text{Decimal}(Q1))$ 
 $\text{Pin} = \text{Decimal}(S) * P00 + \text{Decimal}(F) * P10$ 
 $\text{Summ\_POC\_K} += \text{Pin}$ 
if  $i == 0$  and  $n \geq ts - 1$  and  $n < C + K$ :
     $\text{Bs\_new} += \text{Pin}$ 
if  $i == 0$  and  $n == C - 1$ :
     $\text{Bs\_handoff} = \text{Pin}$ 
if  $i == 0$  and  $n == C + K - 1$ :
     $\text{P0\_C\_K} = \text{Pin}$ 
if  $i == 0$  and  $n \geq C$  and  $n < C + K$ :
     $\text{P0\_C\_K\_U} = \text{Pin}$ 
     $\text{Summ\_P0\_K\_U} += C * \text{P0\_C\_K\_U}$ 
     $\text{Summ\_NQ\_0} += (n + 1 - C) * \text{P0\_C\_K\_U}$ 
if  $i == 0$  and  $n \geq ts - 1$  and  $n < C + K - 1$ :
     $\text{Summ\_}\lambda e_2 += \lambda_h * \text{Pin}$ 
 $S2 = S1$ 
 $S1 = S$ 
 $F2 = F1$ 
 $F1 = F$ 
 $l1 = \lambda$ 
elif  $i == 1$  and  $n < C$ :
     $\vartheta_i = \vartheta_m$ 
     $j = 1 - i$ 
if  $j == 0$ :
     $\vartheta_j = \vartheta_s$ 
else:
     $\vartheta_j = \vartheta_m$ 
 $\mu_i = \mu_m$ 
 $k = \lambda_n$ 
 $L = ((\text{Decimal}(k) + \text{Decimal}(n) * \text{Decimal}(\mu_i) +$ 
     $\hookrightarrow \text{Decimal}(\vartheta_i)) / ($ 
     $(\text{Decimal}(n + 1)) * \text{Decimal}(\mu_i))) *$ 
     $\hookrightarrow \text{Decimal}(L1)) - ($ 
     $(\text{Decimal}(k1) / ((\text{Decimal}(n + 1)) *$ 
     $\hookrightarrow \text{Decimal}(\mu_i))) * \text{Decimal}(L2)) - ($ 
     $(\text{Decimal}(\vartheta_j) / (\text{Decimal}((n + 1)) *$ 
     $\hookrightarrow \text{Decimal}(\mu_i))) * \text{Decimal}(S2))$ 
 $Q = ((\text{Decimal}(k) + \text{Decimal}(n) * \text{Decimal}(\mu_i) +$ 
     $\hookrightarrow \text{Decimal}(\vartheta_i)) / ($ 

```

```

        (Decimal(n + 1)) * Decimal(μi))) *
        ↳ Decimal(Q1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↳ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↳ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P1C += (n + 1) * Pin
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
    ↳ (
        (Decimal(k1) / ((Decimal(C)) *
        ↳ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((C)) *
        ↳ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) -
    ↳ (
        (Decimal(k1) / ((Decimal(C)) *
        ↳ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↳ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k

```

```

elif i == 1 and n > C and n < C + K:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
    L = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(θj) / (Decimal(C) *
            ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(θj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    if i == 1 and n >= C - 1 and n < C + K:
        Bm_new += Pin
    if i == 1 and n >= C and n < C + K:
        P1_C_K_U = Pin
        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
    if i == 1 and n < C - 1:
        Summ_λe_3 += λn * Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
Bnew = Bs_new + Bm_new
U = Summ_P0C + Summ_P1C
P_HCDP = P0_C_K / Summ_P0C_K
U2 = Summ_P0_K_U + Summ_P1_K_U
Utilization = (U + U2) / C
NQ = Summ_NQ_0 + Summ_NQ_1
λe = Summ_λe_1 + Summ_λe_2 + Summ_λe_3
WQ = (NQ / λe) * 60
if K == 0:
    list_ts.append(ts)
    list_Bnew.append(Bnew)
    list_HCDP.append(P_HCDP)
    list_Util.append(Utilization)
elif K == 14:
    list_ts_2.append(ts)
    list_Bnew_2.append(Bnew)
    list_HCDP_2.append(P_HCDP)
    list_Util_2.append(Utilization)
print("H lista ts einai: " + str(list_ts))
print("H lista Bnew einai: " + str(list_Bnew))
print("H lista Util einai: " + str(list_Util))
print("H lista Util_2 einai: " + str(list_Util_2))
print("-----")

```

NP Scheme

11 = 0


```

λ = λh + λn
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
ϑi = ϑm
j = 1 - i
if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μm
k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (

```

```

        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
Summ_S0_in_NP += L
Summ_S1_in_NP += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    ↪ Decimal(ϑm) * P) / - (
        Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
        ↪ Decimal(ϑm) * G)
P00 = 1 / (Summ_S0_in_NP + r * Summ_S1_in_NP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_NP = 0
Bm_new_NP = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0

```

```

for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n == C - 1:
                Bs_new = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n == C - 1:
                Bm_new = Pin
        if i == 0 and n < C:
             $\vartheta_i = \vartheta_s$ 
            j = 1 - i
            if j == 0:
                 $\vartheta_j = \vartheta_s$ 
            else:
                 $\vartheta_j = \vartheta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\vartheta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
                S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\vartheta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
                F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
            Pin = Decimal(S) * P00 + Decimal(F) * P10
            Summ_P0n += Pin
            Summ_P_n_0n += (n + 1) * Pin
            if i == 0 and n == C - 1:
                Bs_new_NP = Pin
            S2 = S1
            S1 = S
            F2 = F1
            F1 = F
            l1 =  $\lambda$ 
        elif i == 1 and n < C:
             $\vartheta_i = \vartheta_m$ 
            j = 1 - i
            if j == 0:
                 $\vartheta_j = \vartheta_s$ 
            else:
                 $\vartheta_j = \vartheta_m$ 
             $\mu_i = \mu_m$ 
            k =  $\lambda_n$ 
            L = (((Decimal(k) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\vartheta_i$ )) / (

```

```

        (Decimal(n + 1)) * Decimal(μi)) * Decimal(L1)) -
        → (
            (Decimal(k1) / ((Decimal(n + 1)) *
            → Decimal(μi)) * Decimal(L2)) - (
            (Decimal(θj) / (Decimal((n + 1)) *
            → Decimal(μi)) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    → Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi)) * Decimal(Q1)) -
        → (
            (Decimal(k1) / ((Decimal(n + 1)) *
            → Decimal(μi)) * Decimal(Q2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            → Decimal(μi)) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n == C - 1:
    Bm_new_NP = Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
Bnew_NP = Bs_new_NP + Bm_new_NP
U = Summ_P_n_0n + Summ_P_n_1n
P_HCDP_NP = Bs_new_NP / Summ_P0n
Utilization_NP = U / C
list_Bnew_NP.append(Bnew_NP)
list_HCDP_NP.append(P_HCDP_NP)
list_Util_NP.append(Utilization_NP)
print("H lista ts einai: " + str(list_ts))
print("H lista Bnew_NP einai: " + str(list_Bnew_NP))
print("H lista HCDP_NP einai: " + str(list_HCDP_NP))
print("H lista Util_NP einai: " + str(list_Util_NP))
print("-----")
# CP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_CP = Decimal(0)

```



```

Summ_S1_in_CP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ_S0_in_CP += S1
            Summ_S1_in_CP += F1
        elif i == 1 and n == 0:
            Summ_S0_in_CP += L1
            Summ_S1_in_CP += Q1
        if i == 0 and n <= ts - 1:
             $\vartheta_i = \vartheta_s$ 
            j = 1 - i
            if j == 0:
                 $\vartheta_j = \vartheta_s$ 
            else:
                 $\vartheta_j = \vartheta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\vartheta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                ↪ Decimal(S1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
                            ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\vartheta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                ↪ Decimal(F1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
                            ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
            Summ_S0_in_CP += S
            Summ_S1_in_CP += F
        if n == C - 1:
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\vartheta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                ↪ Decimal(S1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
                            ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\vartheta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                ↪ Decimal(F1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
                            ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
            A = S
            D = F
            X =  $\lambda$ 
        elif n == C - 2:

```

```

S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(Q1))

Summ_S0_in_CP += S
Summ_S1_in_CP += F
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L1))

```

```

F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n <= ts - 1:
ϑi = ϑm
j = 1 - i
if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μm
k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (

```

```

        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
Summ_S0_in_CP += L
Summ_S1_in_CP += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == ts:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

Summ_S0_in_CP += L
Summ_S1_in_CP += Q
L2 = L1
L1 = L
Q2 = Q1

```

```

Q1 = Q
k1 = k
elif i == 1 and n > ts and n < C:
     $\theta_i = \theta_m$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_m$ 
    k =  $\lambda_n$ 
    L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
        ↪ (
            (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
        ↪ (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
if n == C - 1:
    L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(L1)) - (
            (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(Q1)) - (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2))

    P = L
    G = Q

    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k

r = (Decimal(C) * Decimal( $\mu_s$ ) * A + Decimal( $\theta_s$ ) * A - X * B -
    ↪ Decimal( $\theta_m$ ) * P) / - (
        Decimal(C) * Decimal( $\mu_s$ ) * D + Decimal( $\theta_s$ ) * D - X * E -
        ↪ Decimal( $\theta_m$ ) * G)
P00 = 1 / (Summ_S0_in_CP + r * Summ_S1_in_CP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_CP = 0
Bm_new_CP = 0

```

```

Summ_POS = 0
Summ_P1S = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
POC = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n >= ts - 1 and n < C:
                Summ_POS += Pin
            if i == 0 and n == C - 1:
                POC = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n >= ts - 1 and n < C:
                Summ_P1S += Pin
        if i == 0 and n <= ts - 1:
             $\vartheta_i = \vartheta_s$ 
            j = 1 - i
            if j == 0:
                 $\vartheta_j = \vartheta_s$ 
            else:
                 $\vartheta_j = \vartheta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                 $\rightarrow$  Decimal( $\vartheta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
                S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                 $\rightarrow$  Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
                 $\rightarrow$  Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                 $\rightarrow$  Decimal( $\vartheta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
                F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                 $\rightarrow$  Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                    (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
                 $\rightarrow$  Decimal( $\mu_i$ ))) * Decimal(Q1))
            Pin = Decimal(S) * P00 + Decimal(F) * P10
            Summ_P0n += Pin
            Summ_P_n_0n += (n + 1) * Pin
            if i == 0 and n >= ts - 1 and n < C:
                Summ_POS += Pin
            if i == 0 and n == C - 1:
                POC = Pin
            S2 = S1
            S1 = S
            F2 = F1
            F1 = F
            l1 =  $\lambda$ 
        elif i == 0 and n >= ts and n < C:
             $\vartheta_i = \vartheta_s$ 

```

```

j = 1 - i
if j == 0:
     $\theta_j = \theta_s$ 
else:
     $\theta_j = \theta_m$ 
 $\mu_i = \mu_s$ 
 $\lambda = \lambda_h$ 
S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
↪ Decimal( $\theta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
S1)) - (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
    (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(L1))
F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
↪ Decimal( $\theta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
F1)) - (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
    (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n >= ts - 1 and n < C:
    Summ_POS += Pin
if i == 0 and n == C - 1:
    POC = Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 =  $\lambda$ 
elif i == 1 and n <= ts - 1:
     $\theta_i = \theta_m$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_m$ 
    k =  $\lambda_n$ 
L = (((Decimal(k) + Decimal(n) * Decimal( $\mu_i$ ) +
↪ Decimal( $\theta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
    (
    (Decimal(k1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(L2)) - (
    (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal( $\mu_i$ ) +
↪ Decimal( $\theta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
    (
    (Decimal(k1) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(Q2)) - (
    (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
↪ Decimal( $\mu_i$ ))) * Decimal(F2))

```

```

Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_ln += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == ts:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_ln += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > ts and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (

```



```

        (Decimal(0j) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    Summ_P_n_1n += (n + 1) * Pin
    if i == 1 and n >= ts - 1 and n < C:
        Summ_P1S += Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
    Bnew_CP = Summ_P0S + Summ_P1S
    U = Summ_P_n_0n + Summ_P_n_1n
    P_HCDP_CP = P0C / Summ_P0n
    Utilization_CP = U / C
    list_Bnew_CP.append(Bnew_CP)
    list_HCDP_CP.append(P_HCDP_CP)
    list_Util_CP.append(Utilization_CP)
    print("H lista ts einai: " + str(list_ts))
    print("H lista Bnew_CP einai: " + str(list_Bnew_CP))
    print("H lista HCDP_CP einai: " + str(list_HCDP_CP))
    print("H lista Util_CP einai: " + str(list_Util_CP))
    print("-----")

```

```

import matplotlib.pyplot as plt
import numpy as np

```

```

plt.figure(7)
plt.title('NCBP versus t\uA731')
plt.ylabel('New call blocking probability')
plt.xlabel('Number of shared channels (t\uA731)')
x1 = np.array(list_ts)
y1 = np.array(list_Bnew)
y2 = np.array(list_Bnew_2)
y3 = np.array(list_Bnew_NP)
y4 = np.array(list_Bnew_CP)
plt.xticks(np.arange(0, 70, step=5))
plt.yticks(np.arange(0, 1.1, step=0.1))
plt.xlim(0, 70)
plt.ylim(0, 1)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y3, '^--r', ms = 11, label = 'NP')
plt.plot(x1, y4, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.figure(8)
plt.title('HCDP versus t\uA731')
plt.ylabel('Handover call dropping probability')
plt.xlabel('Number of shared channels (t\uA731)')
x1 = np.array(list_ts)
y1 = np.array(list_HCDP)
y2 = np.array(list_HCDP_2)
y3 = np.array(list_HCDP_NP)
y4 = np.array(list_HCDP_CP)
plt.xticks(np.arange(0, 70, step=5))

```

```

plt.yticks(np.arange(0, 1, step=0.1))
plt.xlim(0, 70)
plt.ylim(0, 0.9)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y3, '^-r', ms = 11, label = 'NP')
plt.plot(x1, y4, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.figure(9)
plt.title('Channel utilization versus t\A731')
plt.ylabel('Channel utilization')
plt.xlabel('Number of shared channels (t\A731)')
x1 = np.array(list_ts)
y1 = np.array(list_Util)
y2 = np.array(list_Util_2)
y3 = np.array(list_Util_NP)
y4 = np.array(list_Util_CP)
plt.xticks(np.arange(0, 70, step=5))
plt.yticks(np.arange(0.6, 1.1, step=0.05))
plt.xlim(0, 70)
plt.ylim(0.6, 1)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y3, '^-r', ms = 11, label = 'NP')
plt.plot(x1, y4, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.show()

```

Γ'.4 Πρόγραμμα των Επιδράσεων του K

```

list_K = []
list_Bnew_for_K = []
list_HCDP_for_K = []
list_WQ_for_K = []
list_Utilization_for_K = []
list_Bnew_NP_for_K = []
list_HCDP_NP_for_K = []
list_Utilization_NP_for_K = []
list_Bnew_CP_for_K = []
list_HCDP_CP_for_K = []
list_Utilization_CP_for_K = []

cnt = 0
while cnt <= 11:
    # Handover queue size:
    K = cnt * 2
    cnt = cnt + 1
    # Rate of the arrival process of New calls:
    λn = 24

```

```

# Rate of the arrival process of Handover calls:
λh = 28
# Capacity of WLAN:
C = 70
# Number of shared channels:
ts = 56
# Rate of the call duration time in stop phase:
μs = 1/5
# Rate of the call duration time in moving phase:
μm = 1/5
# Rate of the dwell time in stop phase:
θs = 1/4
# Rate of the dwell time in moving phase:
θm = 1/5

l1 = 0
k1 = 0
S2 = 0
S1 = 1
l1 = 0
k1 = 0
# MA-CAC Model
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ1 = Decimal(0)
Summ2 = Decimal(0)

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ1 += S1
            Summ2 += F1
        elif i == 1 and n == 0:
            Summ1 += L1

```

```

Summ2 += Q1
if i == 0 and n <= ts - 1:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(Q1))
    Summ1 += S
    Summ2 += F
if n == C - 1:
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (

```

```

                                (Decimal(l1) / ((Decimal(n + 1)) *
                                ↪ Decimal(μi))) * Decimal(F2)) - (
                                (Decimal(θj) / ((Decimal(n + 1)) *
                                ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

    Summ1 += S
    Summ2 += F
    if n == C - 1:
        λ = λh
        S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
        F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    λ = λh

```

```

S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
→ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
→ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
→ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
→ Decimal(θi)) / ((Decimal(C)) * Decimal(μi)))
    * Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
→ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(C)) * Decimal(μi)))
→ * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
→ Decimal(θi)) / ((Decimal(C)) * Decimal(μi)))
    * Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
→ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(C)) * Decimal(μi)))
→ * Decimal(Q1))

Summ1 += S
Summ2 += F
if n == K + C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
→ Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
→ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(C)) *
→ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(C)) *
→ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
→ Decimal(θi)) / (

```

```

(Decimal(C)) * Decimal(μi))) *
↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
(Decimal(θj) / ((Decimal(C)) *
↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == K + C - 2:
λ = λh
S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
↪ Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
(Decimal(θj) / ((Decimal(C)) *
↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
↪ Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
(Decimal(θj) / ((Decimal(C)) *
↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F

S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
θi = θm
j = 1 - i
if j == 0:
    θj = θs
else:
    θj = θm
μi = μm
k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(L1)) - ((Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(Q1)) - ((Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) * Decimal(μi)))
    ↪ * Decimal(F2))

Summ1 += L
Summ2 += Q
if n == C - 1:

```

```

L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↳ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↳ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
            ↳ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(θj) / (Decimal((n + 1)) *
                ↳ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↳ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↳ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
            ↳ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
                ↳ Decimal(μi))) * Decimal(F2))

P = L
G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
    ↳ (
        (Decimal(k1) / ((Decimal(C)) *
            ↳ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(θj) / (Decimal((C)) * Decimal(μi)))
            ↳ * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) -
    ↳ (
        (Decimal(k1) / ((Decimal(C)) *
            ↳ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(θj) / ((Decimal(C)) * Decimal(μi)))
            ↳ * Decimal(F2))

Summ1 += L
Summ2 += Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs

```



```

else:
     $\theta_j = \theta_m$ 
     $\mu_i = \mu_m$ 
     $k = \lambda_n$ 
    L = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
        ↪ (
        (Decimal( $\theta_j$ ) / (Decimal(C) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(S2))
    Q = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
        ↪ (
        (Decimal( $\theta_j$ ) / ((Decimal(C)) * Decimal( $\mu_i$ )))
        ↪ * Decimal(F2))

    Summ1 += L
    Summ2 += Q
    if n == K + C - 1:
        L = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
            (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
            ↪ (
            (Decimal( $\theta_j$ ) / (Decimal(C) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
        Q = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
            (Decimal(C)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
            ↪ (
            (Decimal( $\theta_j$ ) / ((Decimal(C)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2))

        P = L
        G = Q

        L2 = L1
        L1 = L
        Q2 = Q1
        Q1 = Q
        k1 = k

    r = (Decimal(C) * Decimal( $\mu_s$ ) * A + Decimal( $\theta_s$ ) * A - X * B -
        ↪ Decimal( $\theta_m$ ) * P) / - (Decimal(C)
        * Decimal( $\mu_s$ ) * D + Decimal( $\theta_s$ ) * D - X * E - Decimal( $\theta_m$ ) * G)
    P00 = 1 / (Summ1 + r * Summ2)
    P10 = r * P00

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new = 0
Bm_new = 0
Summ_P0C = 0
Summ_P1C = 0

```

```

Summ_P0C_K = 0
Summ_P0_K_U = 0
Summ_P1_K_U = 0
Summ_NQ_0 = 0
Summ_NQ_1 = 0
Summ_λe_1 = 0
Summ_λe_2 = 0
Summ_λe_3 = 0
for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0C += n * Pin
            Summ_P0C_K += Pin
            Summ_λe_1 += (λh + λn) * Pin
            if i == 0 and n >= ts - 1 and n < C + K:
                Bs_new += Pin
            if i == 0 and n == C - 1:
                Bs_handoff = Pin
            if i == 0 and n == C + K - 1:
                P0_C_K = Pin
            if i == 0 and n >= C and n < C + K:
                P0_C_K_U = Pin
                Summ_P0_K_U += C * P0_C_K_U
                Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P1C += n * Pin
            Summ_λe_3 += λn * Pin
            if i == 1 and n >= C - 1:
                Bm_new += Pin
            if i == 1 and n >= C and n < C + K:
                P1_C_K_U = Pin
                Summ_P1_K_U += C * P1_C_K_U
                Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
        if i == 0 and n <= ts - 1:
            ϑi = ϑs
            j = 1 - i
            if j == 0:
                ϑj = ϑs
            else:
                ϑj = ϑm
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
                * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
            F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
                * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))
            Pin = Decimal(S) * P00 + Decimal(F) * P10
            Summ_P0C += (n + 1) * Pin

```

```

Summ_P0C_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n < ts - 1:
    Summ_λe_1 += λ * Pin
if i==0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(S2)) - (
        Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2)) - (
        Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0C += (n + 1) * Pin
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S

```

```

F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(ϑi)) / ((Decimal(C)) * Decimal(μi)))
        * Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi)))
            ↪ * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
        ↪ Decimal(ϑi)) / ((Decimal(C)) * Decimal(μi)))
        * Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi)))
            ↪ * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    Summ_P0C_K += Pin
    if i == 0 and n >= ts - 1 and n < C + K:
        Bs_new += Pin
    if i == 0 and n == C - 1:
        Bs_handoff = Pin
    if i == 0 and n == C + K - 1:
        P0_C_K = Pin
    if i == 0 and n >= C and n < C + K:
        P0_C_K_U = Pin
        Summ_P0_K_U += C * P0_C_K_U
        Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
    if i == 0 and n >= ts - 1 and n < C + K - 1:
        Summ_λe_2 += λh * Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
        μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (

```

```

        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P1C += (n + 1) * Pin
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(C)) * Decimal(μi)))
        ↪ * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((C)) * Decimal(μi)))
        ↪ * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(C)) * Decimal(μi)))
        ↪ * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(C)) * Decimal(μi)))
        ↪ * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k

```

```

elif i == 1 and n > C and n < C + K:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(L1)) - (
        (Decimal(ϑj) / (Decimal(C) * Decimal(μi))) *
        ↪ Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) - (
        (Decimal(ϑj) / ((Decimal(C)) * Decimal(μi)))
        ↪ * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    if i == 1 and n >= C - 1 and n < C + K:
        Bm_new += Pin
    if i == 1 and n >= C and n < C + K:
        P1_C_K_U = Pin
        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
Bnew = Bs_new + Bm_new
U = Summ_P0C + Summ_P1C
P_HCDP = P0_C_K / Summ_P0C_K
U2 = Summ_P0_K_U + Summ_P1_K_U
Utilization = (U + U2) / C
NQ = Summ_NQ_0 + Summ_NQ_1
λe = Summ_λe_1 + Summ_λe_2 + Summ_λe_3
WQ = (NQ / λe) * 60
list_K.append(K)
list_Bnew_for_K.append(Bnew)
list_HCDP_for_K.append(P_HCDP)
list_WQ_for_K.append(WQ)
list_Utilization_for_K.append(Utilization)
print("H lista K einai: " + str(list_K))
print("H lista Bnew einai: " + str(list_Bnew_for_K))
print("H lista HCDP einai: " + str(list_HCDP_for_K))
print("H lista WQ einai: " + str(list_WQ_for_K))

↪ print("-----")
# NP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)

```

```

L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_NP = Decimal(0)
Summ_S1_in_NP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ_S0_in_NP += S1
            Summ_S1_in_NP += F1
        elif i == 1 and n == 0:
            Summ_S0_in_NP += L1
            Summ_S1_in_NP += Q1
        if i == 0 and n < C:
             $\theta_i = \theta_s$ 
            j = 1 - i
            if j == 0:
                 $\theta_j = \theta_s$ 
            else:
                 $\theta_j = \theta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(S1)) -
                ↪ (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                                ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(F1)) -
                ↪ (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                                ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
            Summ_S0_in_NP += S
            Summ_S1_in_NP += F
        if n == C - 1:
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                ↪ Decimal(S1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm

    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (

```



```

        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2))
Summ_S0_in_NP += L
Summ_S1_in_NP += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q

    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k

r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    ↪ Decimal(ϑm) * P) / - (
    Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
    ↪ Decimal(ϑm) * G)
P00 = 1 / (Summ_S0_in_NP + r * Summ_S1_in_NP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_NP = 0
Bm_new_NP = 0
Summ_P0n = 0
Summ_P1n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin

```

```

Summ_P_n_0n += n * Pin
if i == 0 and n == C - 1:
    Bs_new_NP = Pin
elif i == 1 and n == 0:
    Pin = L1 * P00 + Q1 * P10
    Summ_P_n_1n += n * Pin
    Summ_P1n += Pin
    if i == 1 and n == C - 1:
        Bm_new_NP = Pin
if i == 0 and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    ↪ (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
    ↪ (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    ↪ (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
    ↪ (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    Summ_P0n += Pin
    Summ_P_n_0n += (n + 1) * Pin
    if i == 0 and n == C - 1:
        Bs_new_NP = Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    ↪ (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
    ↪ (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (

```

```

        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
Summ_P1n += Pin
if i == 1 and n == C - 1:
    Bm_new_NP = Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
Bnew_NP = Bs_new_NP + Bm_new_NP
U = Summ_P_n_0n + Summ_P_n_1n
P_HCDP_NP = Bs_new_NP / Summ_P0n
Utilization_NP = U / C
list_Bnew_NP_for_K.append(Bnew_NP)
list_HCDP_NP_for_K.append(P_HCDP_NP)
list_Utilization_NP_for_K.append(Utilization_NP)
print("H lista K einai: " + str(K))
print("H lista Bnew_NP(for K) einai: " + str(list_Bnew_NP_for_K))
print("H lista HCDP_NP(for K) einai: " + str(list_HCDP_NP_for_K))

    ↪ print("-----")
#CP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_CP = Decimal(0)
Summ_S1_in_CP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):

```

```

if i == 0 and n == 0:
    Summ_S0_in_CP += S1
    Summ_S1_in_CP += F1
elif i == 1 and n == 0:
    Summ_S0_in_CP += L1
    Summ_S1_in_CP += Q1
if i == 0 and n <= ts - 1:
     $\vartheta_i = \vartheta_s$ 
    j = 1 - i
    if j == 0:
         $\vartheta_j = \vartheta_s$ 
    else:
         $\vartheta_j = \vartheta_m$ 
     $\mu_i = \mu_s$ 
     $\lambda = \lambda_h + \lambda_n$ 
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\vartheta_i$ )) / (
    ↪ (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(S1)) -
    ↪ (
    ↪ (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
    ↪ (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\vartheta_i$ )) / (
    ↪ (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(F1)) -
    ↪ (
    ↪ (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
    ↪ (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
    Summ_S0_in_CP += S
    Summ_S1_in_CP += F
    if n == C - 1:
         $\lambda = \lambda_h + \lambda_n$ 
        S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
        ↪ Decimal( $\vartheta_i$ )) / (
        ↪ (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(S1)) - (
        ↪ (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
        ↪ (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
        F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
        ↪ Decimal( $\vartheta_i$ )) / (
        ↪ (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(F1)) - (
        ↪ (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
        ↪ (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
        A = S
        D = F
        X =  $\lambda$ 
    elif n == C - 2:
        S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
        ↪ Decimal( $\vartheta_i$ )) / (

```

```

        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

Summ_S0_in_CP += S
Summ_S1_in_CP += F
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (

```

```

        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n <= ts - 1:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (

```

```

                                (Decimal(ϑj) / ((Decimal(n + 1)) *
                                ↪ Decimal(μi))) * Decimal(F2))
Summ_S0_in_CP += L
Summ_S1_in_CP += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == ts:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm

μi = μm
k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

Summ_S0_in_CP += L
Summ_S1_in_CP += Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q

```

```

k1 = k
elif i == 1 and n > ts and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        (
            (Decimal(ϑj) / (Decimal((n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2))
    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
    if n == C - 1:
        L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
            ↪ Decimal(L1)) - (
                (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
            ↪ Decimal(Q1)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))
        P = L
        G = Q
        L2 = L1
        L1 = L
        Q2 = Q1
        Q1 = Q
        k1 = k
    r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
        ↪ Decimal(ϑm) * P) / - (
            Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
            ↪ Decimal(ϑm) * G)
    P00 = 1 / (Summ_S0_in_CP + r * Summ_S1_in_CP)
    P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_NP = 0
Bm_new_NP = 0
Summ_POS = 0

```



```

Summ_P1S = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
POC = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n >= ts - 1 and n < C:
                Summ_POS += Pin
            if i == 0 and n == C - 1:
                POC = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n >= ts - 1 and n < C:
                Summ_P1S += Pin
        if i == 0 and n <= ts - 1:
             $\theta_i = \theta_s$ 
            j = 1 - i
            if j == 0:
                 $\theta_j = \theta_s$ 
            else:
                 $\theta_j = \theta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                 $\hookrightarrow$  Decimal( $\theta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
                S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                 $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                    (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                 $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                 $\hookrightarrow$  Decimal( $\theta_i$ )) / (
                    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(
                F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
                 $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                    (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                 $\hookrightarrow$  Decimal( $\mu_i$ ))) * Decimal(Q1))
            Pin = Decimal(S) * P00 + Decimal(F) * P10
            Summ_P0n += Pin
            Summ_P_n_0n += (n + 1) * Pin
            if i == 0 and n >= ts - 1 and n < C:
                Summ_POS += Pin
            if i == 0 and n == C - 1:
                POC = Pin
            S2 = S1
            S1 = S
            F2 = F1
            F1 = F
            l1 =  $\lambda$ 
        elif i == 0 and n >= ts and n < C:
             $\theta_i = \theta_s$ 
            j = 1 - i

```

```

if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μs
λ = λh
S = ((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
F = ((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n >= ts - 1 and n < C:
    Summ_POS += Pin
if i == 0 and n == C - 1:
    P0C = Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n <= ts - 1:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
L = ((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2))
Q = ((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10

```

```

Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == ts:
     $\vartheta_i = \vartheta_m$ 
    j = 1 - i
    if j == 0:
         $\vartheta_j = \vartheta_s$ 
    else:
         $\vartheta_j = \vartheta_m$ 
     $\mu_i = \mu_m$ 
    k =  $\lambda_n$ 
L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(L2)) - (
        (Decimal( $\vartheta_j$ ) / (Decimal((n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(Q2)) - (
        (Decimal( $\vartheta_j$ ) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > ts and n < C:
     $\vartheta_i = \vartheta_m$ 
    j = 1 - i
    if j == 0:
         $\vartheta_j = \vartheta_s$ 
    else:
         $\vartheta_j = \vartheta_m$ 
     $\mu_i = \mu_m$ 
    k =  $\lambda_n$ 
L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
    ↪ (
        (Decimal( $\vartheta_j$ ) / (Decimal((n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
    (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
    ↪ (

```

```

        (Decimal(θj) / ((Decimal(n + 1)) *
        → Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
Bnew_CP = Summ_P0S + Summ_P1S
U = Summ_P_n_0n + Summ_P_n_1n
P_HCDP_CP = P0C / Summ_P0n
Utilization_CP = U / C
list_Bnew_CP_for_K.append(Bnew_CP)
list_HCDP_CP_for_K.append(P_HCDP_CP)
list_Utilization_CP_for_K.append(Utilization_CP)
print("H lista K einai: " + str(list_K))
print("H lista Bnew_CP(for K) einai: " + str(list_Bnew_CP_for_K))
print("H lista HCDP_CP(for K) einai: " + str(list_HCDP_CP_for_K))
print("H lista Utilization_CP(for K) einai: " +
    → str(list_Utilization_CP_for_K))

→ print("-----")

import matplotlib.pyplot as plt
import numpy as np

plt.figure(10)
plt.title('NCBP versus K')
plt.ylabel('New call blocking probability')
plt.xlabel('Queue size (K)')
x1 = np.array(list_K)
y1 = np.array(list_Bnew_for_K)
y2 = np.array(list_Bnew_NP_for_K)
y3 = np.array(list_Bnew_CP_for_K)
plt.xticks(np.arange(0, 23, step=2))
plt.yticks(np.arange(0.35, 0.85, step=0.05))
plt.xlim(0, 22)
plt.ylim(0.35, 0.8)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC')
plt.plot(x1, y2, '^-r', ms = 11, label = 'NP')
plt.plot(x1, y3, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.figure(11)
plt.title('HCDP versus K')
plt.ylabel('Handover call dropping probability')
plt.xlabel('Queue size (K)')
plt.xticks(np.arange(0, 23, step=2))
plt.yticks(np.arange(0.35, 0.85, step=0.05))
plt.xlim(0, 22)
plt.ylim(0.35, 0.8)
x1 = np.array(list_K)
y1 = np.array(list_HCDP_for_K)

```

```

y2 = np.array(list_HCDP_NP_for_K)
y3 = np.array(list_HCDP_CP_for_K)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC')
plt.plot(x1, y2, '^-r', ms = 11, label = 'NP')
plt.plot(x1, y3, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'upper right')
plt.grid()
plt.figure(12)
plt.title('Waiting time in queue versus K')
plt.ylabel('Waiting time in queue (seconds)')
plt.xlabel('Queue size (K)')
plt.axis([0, 22, 0, 45])
plt.xticks(np.arange(0, 23, step=2))
x1 = np.array(list_K)
y1 = np.array(list_WQ_for_K)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC')
plt.legend(loc = 'lower right')
plt.grid()
plt.figure(13)
plt.title('Channel utilization versus K')
plt.ylabel('Channel utilization')
plt.xlabel('Queue size (K)')
x1 = np.array(list_K)
y1 = np.array(list_Utilization_for_K)
y2 = np.array(list_Utilization_NP_for_K)
y3 = np.array(list_Utilization_CP_for_K)
plt.xticks(np.arange(0, 23, step=2))
plt.yticks(np.arange(0.8, 1.1, step=0.05))
plt.xlim(0, 22)
plt.ylim(0.8, 1)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC')
plt.plot(x1, y2, '^-r', ms = 11, label = 'NP')
plt.plot(x1, y3, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.show()

```

Γ.5 Πρόγραμμα των Επιδράσεων του $1/\theta_S$

```

list_Bnew_1_theta_S_NP = []
list_HCDP_1_theta_S_NP = []
list_Bnew_1_theta_S_CP = []
list_HCDP_1_theta_S_CP = []
list_1_theta_S = [0.5, 0.55, 0.6, 0.7, 0.8, 1, 2, 3, 4, 5]
# Handover queue size:
list_mono_K = [0, 6, 14]
# Rate of the dwell time in stop phase:
list_theta_S = [1 / 0.5, 1 / 0.55, 1 / 0.6, 1 / 0.7, 1 / 0.8, 1, 1 / 2, 1 /
↳ 3, 1 / 4, 1 / 5]

```

```

list_Bnew_1_θs_K_0 = []
list_Bnew_1_θs_K_6 = []
list_Bnew_1_θs_K_14 = []
list_HC DP_1_θs_K_0 = []
list_HC DP_1_θs_K_6 = []
list_HC DP_1_θs_K_14 = []

for θs in list_θs:
    # Rate of the arrival process of New calls:
    λn = 24
    # Rate of the arrival process of Handover calls:
    λh = 28
    # Capacity of WLAN:
    C = 70
    # Number of shared channels:
    ts = 56
    # Rate of the call duration time in stop phase:
    μs = 1/5
    # Rate of the call duration time in moving phase:
    μm = 1/5
    # Rate of the dwell time in moving phase:
    θm = 1/5
    for K in list_mono_K:
        l1 = 0
        k1 = 0
        S2 = 0
        S1 = 1
        l1 = 0
        k1 = 0
#MA-CAC Model
    from decimal import *

    ctx = getcontext()
    ctx.prec = 50
    ctx.rounding = ROUND_UP

    S2 = Decimal(0)
    S1 = Decimal(1)
    L2 = Decimal(0)
    L1 = Decimal(0)
    F2 = Decimal(0)
    F1 = Decimal(0)
    Q2 = Decimal(0)
    Q1 = Decimal(1)
    A = 0
    B = 0
    P = 0
    D = 0
    E = 0
    G = 0
    Summ1 = Decimal(0)
    Summ2 = Decimal(0)

    from decimal import *

    ctx = getcontext()
    ctx.prec = 50
    ctx.rounding = ROUND_UP

```

```

for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ1 += S1
            Summ2 += F1
        elif i == 1 and n == 0:
            Summ1 += L1
            Summ2 += Q1
        if i == 0 and n <= ts - 1:
             $\theta_i = \theta_s$ 
            j = 1 - i
            if j == 0:
                 $\theta_j = \theta_s$ 
            else:
                 $\theta_j = \theta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ )) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
                ↪ 1)) * Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ )) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n +
                ↪ 1)) * Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
            Summ1 += S
            Summ2 += F
        if n == C - 1:
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ )) / (
                (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                ↪ Decimal(S1)) - (
                (Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal( $\mu_i$ ))) *
                ↪ Decimal(S2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ )) / (
                (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                ↪ Decimal(F1)) - (
                (Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal( $\mu_i$ ))) *
                ↪ Decimal(F2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))

            A = S
            D = F
            X =  $\lambda$ 
        elif n == C - 2:
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ )) / (

```

```

        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1))
            ↪ * Decimal(μi))) *
            ↪ Decimal(S2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
        * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
        ↪ 1)) * Decimal(μi))) * Decimal(S2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
        * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n +
        ↪ 1)) * Decimal(μi))) * Decimal(F2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q1))

    Summ1 += S
    Summ2 += F
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1))
            ↪ * Decimal(μi))) *
            ↪ Decimal(S2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))

```



```

F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) *
    ↪ Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
    (Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) *
    ↪ Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) *
    ↪ Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F

S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
ϑi = ϑs
j = 1 - i
if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μs
λ = λh
S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
↪ Decimal(ϑi)) / ((Decimal(C)) * Decimal(μi)))
    * Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
↪ Decimal(ϑi)) / ((Decimal(C)) * Decimal(μi)))
    * Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Summ1 += S
Summ2 += F
if n == K + C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2))
    ↪ - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))
    ↪ - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == K + C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2))
    ↪ - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))
    ↪ - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
if j == 0:

```

```

         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_m$ 
     $k = \lambda_n$ 
    L = (((Decimal(k) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
        * Decimal(L1)) - ((Decimal(k1) / ((Decimal(n +
    ↪ 1)) * Decimal( $\mu_i$ ))) * Decimal(L2)) - (
        (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
        * Decimal(Q1)) - ((Decimal(k1) / ((Decimal(n +
    ↪ 1)) * Decimal( $\mu_i$ ))) * Decimal(Q2)) - (
        (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(F2))
    Summ1 += L
    Summ2 += Q
    if n == C - 1:
        L = (((Decimal(k) + Decimal(n) * Decimal( $\mu_i$ ) +
        ↪ Decimal( $\theta_i$ ) / (
            (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1))
        ↪ * Decimal( $\mu_i$ ))) *
        ↪ Decimal(L2)) - (
            (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
        Q = (((Decimal(k) + Decimal(n) * Decimal( $\mu_i$ ) +
        ↪ Decimal( $\theta_i$ ) / (
            (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(Q1)) - (
            (Decimal(k1) / ((Decimal(n + 1))
        ↪ * Decimal( $\mu_i$ ))) *
        ↪ Decimal(Q2)) - (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
        ↪ Decimal( $\mu_i$ ))) * Decimal(F2))

        P = L
        G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
    elif i == 1 and n == C:
         $\theta_i = \theta_m$ 
        j = 1 - i
        if j == 0:
             $\theta_j = \theta_s$ 
        else:
             $\theta_j = \theta_m$ 
         $\mu_i = \mu_m$ 
        k =  $\lambda_n$ 
        L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
            (Decimal(C)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(L1)) - (

```

```

        (Decimal(k1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((C)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))

Summ1 += L
Summ2 += Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
        (Decimal(θj) / (Decimal(C) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))

Summ1 += L
Summ2 += Q
if n == K + C - 1:
    L = (((Decimal(C) * Decimal(μi) + Decimal(θi)) /
    ↪ (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
        (Decimal(θj) / (Decimal(C) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(θi)) /
    ↪ (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q

```

```

k1 = k
r = (Decimal(C) * Decimal(μs) * A + Decimal(θs) * A - X * B -
↳ Decimal(θm) * P) / - (Decimal(C) * Decimal(μs)
    * D + Decimal(θs) * D - X * E - Decimal(θm) * G)
P00 = 1 / (Summ1 + r * Summ2)
P10 = r * P00

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new = 0
Bm_new = 0
Summ_P0C = 0
Summ_P1C = 0
Summ_P0C_K = 0
Summ_P0_K_U = 0
Summ_P1_K_U = 0
Summ_NQ_0 = 0
Summ_NQ_1 = 0
Summ_λe_1 = 0
Summ_λe_2 = 0
Summ_λe_3 = 0
for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0C += n * Pin
            Summ_P0C_K += Pin
            Summ_λe_1 += (λh + λn) * Pin
            if i == 0 and n >= ts - 1 and n < C + K:
                Bs_new += Pin
            if i == 0 and n == C - 1:
                Bs_handoff = Pin
            if i == 0 and n == C + K - 1:
                P0_C_K = Pin
            if i == 0 and n >= C and n < C + K:
                P0_C_K_U = Pin
                Summ_P0_K_U += C * P0_C_K_U
                Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P1C += n * Pin
            Summ_λe_3 += λn * Pin
            if i == 1 and n >= C - 1:
                Bm_new += Pin
            if i == 1 and n >= C and n < C + K:
                P1_C_K_U = Pin

```

```

        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
    if i == 0 and n <= ts - 1:
        ϑi = ϑs
        j = 1 - i
        if j == 0:
            ϑj = ϑs
        else:
            ϑj = ϑm
        μi = μs
        λ = λh + λn
        S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
            ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
            * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
            ↪ 1)) * Decimal(μi))) * Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))
        F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
            ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
            * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n +
            ↪ 1)) * Decimal(μi))) * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q1))
        Pin = Decimal(S) * P00 + Decimal(F) * P10
        Summ_P0C += (n + 1) * Pin
        Summ_P0C_K += Pin
        if i == 0 and n >= ts - 1 and n < C + K:
            Bs_new += Pin
        if i == 0 and n == C - 1:
            Bs_handoff = Pin
        if i == 0 and n == C + K - 1:
            P0_C_K = Pin
        if i == 0 and n >= C and n < C + K:
            P0_C_K_U = Pin
            Summ_P0_K_U += C * P0_C_K_U
            Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
        if i == 0 and n < ts - 1:
            Summ_λe_1 += λ * Pin
        if i==0 and n >= ts - 1 and n < C + K - 1:
            Summ_λe_2 += λh * Pin
        S2 = S1
        S1 = S
        F2 = F1
        F1 = F
        l1 = λ
    elif i == 0 and n >= ts and n < C:
        ϑi = ϑs
        j = 1 - i
        if j == 0:
            ϑj = ϑs
        else:
            ϑj = ϑm
        μi = μs
        λ = λh
        S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
            ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
            * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
            ↪ 1)) * Decimal(μi))) * Decimal(S2)) - (

```

```

        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n +
    ↪ 1)) * Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_POC += (n + 1) * Pin
Summ_POC_K += Pin
if i == 0 and n >= ts - 1:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(θi)) / ((Decimal(C)) * Decimal(μi)))
    * Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(θi)) / ((Decimal(C)) * Decimal(μi)))
    * Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_POC_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:

```

```

        P0_C_K_U = Pin
        Summ_P0_K_U += C * P0_C_K_U
        Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
    if i == 0 and n >= ts - 1 and n < C + K - 1:
        Summ_λe_2 += λh * Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
        μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    Summ_P1C += (n + 1) * Pin
    if i == 1 and n >= C - 1 and n < C + K:
        Bm_new += Pin
    if i == 1 and n >= C and n < C + K:
        P1_C_K_U = Pin
        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
    if i == 1 and n < C - 1:
        Summ_λe_3 += λn * Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm

```



```

k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(θj) / (Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(θj) / ((Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(θj) / (Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(C) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(θj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k

```

```

Bnew = Bs_new + Bm_new
U = Summ_P0C + Summ_P1C
P_HCDP = P0_C_K / Summ_P0C_K
U2 = Summ_P0_K_U + Summ_P1_K_U
Utilization = (U + U2) / C
NQ = Summ_NQ_0 + Summ_NQ_1
λe = Summ_λe_1 + Summ_λe_2 + Summ_λe_3
WQ = (NQ / λe) * 60
if K == 0:
    list_Bnew_1_0s_K_0.append(Bnew)
    list_HCDP_1_0s_K_0.append(P_HCDP)
elif K == 6:
    list_Bnew_1_0s_K_6.append(Bnew)
    list_HCDP_1_0s_K_6.append(P_HCDP)
elif K == 14:
    list_Bnew_1_0s_K_14.append(Bnew)
    list_HCDP_1_0s_K_14.append(P_HCDP)
print("H lista Bnew gia K=0 (1/0s) einai: " +
      ↪ str(list_Bnew_1_0s_K_0))
print("H lista Bnew gia K=6 (1/0s) einai: " +
      ↪ str(list_Bnew_1_0s_K_6))
print("H lista Bnew gia K=14 (1/0s) einai: " +
      ↪ str(list_Bnew_1_0s_K_14))
print("H lista HCDP gia K=0 (1/0s) einai: " +
      ↪ str(list_HCDP_1_0s_K_0))
print("H lista HCDP gia K=6 (1/0s) einai: " +
      ↪ str(list_HCDP_1_0s_K_6))
print("H lista HCDP gia K=14 (1/0s) einai: " +
      ↪ str(list_HCDP_1_0s_K_14))

↪ print("-----")
#NP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_NP = Decimal(0)
Summ_S1_in_NP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:

```

```

Summ_S0_in_NP += S1
Summ_S1_in_NP += F1
elif i == 1 and n == 0:
    Summ_S0_in_NP += L1
    Summ_S1_in_NP += Q1
if i == 0 and n < C:
     $\theta_i = \theta_s$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_s$ 
     $\lambda = \lambda_h + \lambda_n$ 
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
    Summ_S0_in_NP += S
    Summ_S1_in_NP += F
if n == C - 1:
     $\lambda = \lambda_h + \lambda_n$ 
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
    F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
    A = S
    D = F
    X =  $\lambda$ 
elif n == C - 2:
    S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
    ↪ Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
        ↪ Decimal(S1)) - (

```

```

        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

Summ_S0_in_NP += L
Summ_S1_in_NP += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (

```

```

(Decimal(n + 1)) * Decimal(μi))) *
↳ Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(F2))

P = L
G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k

r = (Decimal(C) * Decimal(μs) * A + Decimal(θs) * A - X * B -
↳ Decimal(θm) * P) / - (
    Decimal(C) * Decimal(μs) * D + Decimal(θs) * D - X * E -
    ↳ Decimal(θm) * G)
P00 = 1 / (Summ_S0_in_NP + r * Summ_S1_in_NP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_NP = 0
Bm_new_NP = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n == C - 1:
                Bs_new = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n == C - 1:
                Bm_new = Pin
        if i == 0 and n < C:
            θi = θs
            j = 1 - i
            if j == 0:
                θj = θs
            else:
                θj = θm
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
            ↳ Decimal(θi)) / (
                (Decimal(n + 1)) * Decimal(μi))) * Decimal(

```

```

S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↳ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(
F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n == C - 1:
    Bs_new_NP = Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↳ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↳ (
    (Decimal(k1) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(L2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↳ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↳ (
    (Decimal(k1) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
    ↳ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n == C - 1:
    Bm_new_NP = Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
Bnew_NP = Bs_new_NP + Bm_new_NP
U = Summ_P_n_0n + Summ_P_n_1n
P_HCDP_NP = Bs_new_NP / Summ_P0n

```

```

Utilization_NP = U / C
list_Bnew_1_θs_NP.append(Bnew_NP)
list_HCDP_1_θs_NP.append(P_HCDP_NP)
print("H lista K einai: " + str(K))
print("H lista Bnew_NP (for θs) einai: " + str(list_Bnew_1_θs_NP))
print("H lista HCDP_NP (for θs) einai: " + str(list_HCDP_1_θs_NP))

↪ print("-----")
#CP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_CP = Decimal(0)
Summ_S1_in_CP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ_S0_in_CP += S1
            Summ_S1_in_CP += F1
        elif i == 1 and n == 0:
            Summ_S0_in_CP += L1
            Summ_S1_in_CP += Q1
        if i == 0 and n <= ts - 1:
            θi = θs
            j = 1 - i
            if j == 0:
                θj = θs
            else:
                θj = θm
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
                    (Decimal(n + 1)) * Decimal(μi))) *
↪ Decimal(S1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(S2)) - (
                    (Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(L1))

```

```

F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))
Summ_S0_in_CP += S
Summ_S1_in_CP += F
if n == C - 1:
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(θj) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i

```



```

if j == 0:
    θj = θs
else:
    θj = θm
μi = μs
λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(
S1)) - (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(S2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(
F1)) - (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(Q1))

Summ_S0_in_CP += S
Summ_S1_in_CP += F
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
↪ Decimal(F1)) - (

```

```

                                (Decimal(l1) / ((Decimal(n + 1)) *
                                ↪ Decimal(μi))) * Decimal(F2)) - (
                                (Decimal(θj) / ((Decimal(n + 1)) *
                                ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n <= ts - 1:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
    if n == C - 1:
        L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(θj) / (Decimal((n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2))
        Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(θi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
    L2 = L1

```

```

L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == ts:
     $\theta_i = \theta_m$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_m$ 
    k =  $\lambda_n$ 
    L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(L2)) - (
            (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(Q2)) - (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2))

    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n > ts and n < C:
     $\theta_i = \theta_m$ 
    j = 1 - i
    if j == 0:
         $\theta_j = \theta_s$ 
    else:
         $\theta_j = \theta_m$ 
     $\mu_i = \mu_m$ 
    k =  $\lambda_n$ 
    L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(L1)) -
        ↪ (
            (Decimal( $\theta_j$ ) / (Decimal((n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (
        (Decimal(n + 1)) * Decimal( $\mu_i$ ))) * Decimal(Q1)) -
        ↪ (
            (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
            ↪ Decimal( $\mu_i$ ))) * Decimal(F2))

    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
if n == C - 1:
    L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\theta_i$ )) / (

```

```

        (Decimal(n + 1)) * Decimal(μi))) *
        ↳ Decimal(L1)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
            ↳ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↳ Decimal(Q1)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↳ Decimal(μi))) * Decimal(F2))

P = L
G = Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k

r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    ↳ Decimal(ϑm) * P) / - (
        Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
        ↳ Decimal(ϑm) * G)
P00 = 1 / (Summ_S0_in_CP + r * Summ_S1_in_CP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_CP = 0
Bm_new_CP = 0
Summ_POS = 0
Summ_P1S = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
POC = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n >= ts - 1 and n < C:
                Summ_POS += Pin
            if i == 0 and n == C - 1:
                POC = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n >= ts - 1 and n < C:
                Summ_P1S += Pin
        if i == 0 and n <= ts - 1:
            ϑi = ϑs
            j = 1 - i
            if j == 0:

```

```

        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    Summ_P0n += Pin
    Summ_P_n_0n += (n + 1) * Pin
    if i == 0 and n >= ts - 1 and n < C:
        Summ_POS += Pin
    if i == 0 and n == C - 1:
        POC = Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    θi = θs
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    Summ_P0n += Pin
    Summ_P_n_0n += (n + 1) * Pin
    if i == 0 and n >= ts - 1 and n < C:

```

```

        Summ_P0S += Pin
    if i == 0 and n == C - 1:
        P0C = Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n <= ts - 1:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L2)) - (
                (Decimal(ϑj) / (Decimal((n + 1)) *
                    ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                    ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    Summ_P_n_1n += (n + 1) * Pin
    if i == 1 and n >= ts - 1 and n < C:
        Summ_P1S += Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == ts:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L2)) - (
                (Decimal(ϑj) / (Decimal((n + 1)) *
                    ↪ Decimal(μi))) * Decimal(S2))

```

```

Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(θj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > ts and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(θj) / (Decimal((n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
    (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(θj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
Bnew_CP = Summ_P0S + Summ_P1S
U = Summ_P_n_0n + Summ_P_n_1n
P_HCDP_CP = P0C / Summ_P0n
Utilization_CP = U / C
list_Bnew_1_θs_CP.append(Bnew_CP)
list_HCDP_1_θs_CP.append(P_HCDP_CP)
print("H lista Bnew_CP(for θs) einai: " + str(list_Bnew_1_θs_CP))
print("H lista HCDP_CP(for θs) einai: " + str(list_HCDP_1_θs_CP))

↪ print("-----")

import matplotlib.pyplot as plt
import numpy as np

```

```

plt.figure(14)
plt.title('NCBP versus  $1/\theta_s$ ')
plt.ylabel('New call blocking probability')
plt.xlabel('Average dwell time in stop phase ( $1/\theta_s$ )')
x1 = np.array(list_1_theta_s)
y1 = np.array(list_Bnew_1_theta_s_K_0)
y2 = np.array(list_Bnew_1_theta_s_K_6)
y3 = np.array(list_Bnew_1_theta_s_K_14)
y4 = np.array(list_Bnew_1_theta_s_NP)
y5 = np.array(list_Bnew_1_theta_s_CP)
plt.xticks(np.arange(0, 5.5, step=0.5))
plt.yticks(np.arange(0.1, 0.95, step=0.1))
plt.xlim(0, 5)
plt.ylim(0.1, 0.9)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, 'v-k', ms = 11, label = 'MA-CAC (K=6)')
plt.plot(x1, y3, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y4, '^-r', ms = 11, label = 'NP')
plt.plot(x1, y5, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.figure(15)
plt.title('HCDP versus  $1/\theta_s$ ')
plt.ylabel('Handover call dropping probability')
plt.xlabel('Average dwell time in stop phase ( $1/\theta_s$ )')
plt.xticks(np.arange(0, 3.5, step=0.5))
plt.yticks(np.arange(0, 0.85, step=0.1))
plt.xlim(0, 3)
plt.ylim(0, 0.8)
x1 = np.array(list_1_theta_s)
y1 = np.array(list_HCDP_1_theta_s_K_0)
y2 = np.array(list_HCDP_1_theta_s_K_6)
y3 = np.array(list_HCDP_1_theta_s_K_14)
y4 = np.array(list_HCDP_1_theta_s_NP)
y5 = np.array(list_HCDP_1_theta_s_CP)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, 'v-k', ms = 11, label = 'MA-CAC (K=6)')
plt.plot(x1, y3, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y4, '^-r', ms = 11, label = 'NP')
plt.plot(x1, y5, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.show()

```

Γ'.6 Πρόγραμμα των Επιδράσεων του $1/\theta_M$

```

list_Bnew_1_theta_m_NP = []
list_HCDP_1_theta_m_NP = []
list_Bnew_1_theta_m_CP = []
list_HCDP_1_theta_m_CP = []
list_1_theta_m = [0.5, 0.55, 0.6, 0.7, 0.8, 1, 2, 3, 4, 5]

```



```

# Handover queue size:
list_mono_K = [0, 6, 14]
# Rate of the dwell time in moving phase:
list_θm = [1 / 0.5, 1 / 0.55, 1 / 0.6, 1 / 0.7, 1 / 0.8, 1 , 1 / 2, 1 /
↪ 3, 1 / 4, 1 / 5]
list_Bnew_1_θm_K_0 = []
list_Bnew_1_θm_K_6 = []
list_Bnew_1_θm_K_14 = []
list_HC DP_1_θm_K_0 = []
list_HC DP_1_θm_K_6 = []
list_HC DP_1_θm_K_14 = []

for θm in list_θm:
    # Rate of the arrival process of New calls:
    λn = 24
    # Rate of the arrival process of Handover calls:
    λh = 28
    # Capacity of WLAN:
    C = 70
    # Number of shared channels:
    ts = 56
    # Rate of the call duration time in stop phase:
    μs = 1/5
    # Rate of the call duration time in moving phase:
    μm = 1/5
    # Rate of the dwell time in stop phase:
    θs = 1/4
    for K in list_mono_K:
        l1 = 0
        k1 = 0
        S2 = 0
        S1 = 1
        l1 = 0
        k1 = 0
# MA-CAC Model
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ1 = Decimal(0)
Summ2 = Decimal(0)

```

```

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ1 += S1
            Summ2 += F1
        elif i == 1 and n == 0:
            Summ1 += L1
            Summ2 += Q1
        if i == 0 and n <= ts - 1:
             $\theta_i = \theta_s$ 
            j = 1 - i
            if j == 0:
                 $\theta_j = \theta_s$ 
            else:
                 $\theta_j = \theta_m$ 
             $\mu_i = \mu_s$ 
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
                ↪ 1)) * Decimal( $\mu_i$ ))) * Decimal(S2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ ) / ((Decimal(n + 1)) * Decimal( $\mu_i$ )))
                * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n +
                ↪ 1)) * Decimal( $\mu_i$ ))) * Decimal(F2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))
            Summ1 += S
            Summ2 += F
        if n == C - 1:
             $\lambda = \lambda_h + \lambda_n$ 
            S = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ ) / (
                (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                ↪ Decimal(S1)) - (
                (Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal( $\mu_i$ ))) *
                ↪ Decimal(S2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(L1))
            F = (((Decimal( $\lambda$ ) + Decimal(n) * Decimal( $\mu_i$ ) +
                ↪ Decimal( $\theta_i$ ) / (
                (Decimal(n + 1)) * Decimal( $\mu_i$ ))) *
                ↪ Decimal(F1)) - (
                (Decimal(l1) / ((Decimal(n + 1))
                ↪ * Decimal( $\mu_i$ ))) *
                ↪ Decimal(F2)) - (
                (Decimal( $\theta_j$ ) / ((Decimal(n + 1)) *
                ↪ Decimal( $\mu_i$ ))) * Decimal(Q1))

A = S

```

```

D = F
X = λ
elif n == C - 2:
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
          (Decimal(n + 1)) * Decimal(μi))) *
      ↪ Decimal(S1)) - (
          (Decimal(l1) / ((Decimal(n + 1))
          ↪ * Decimal(μi))) *
          ↪ Decimal(S2)) - (
          (Decimal(ϑj) / ((Decimal(n + 1)) *
          ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
          (Decimal(n + 1)) * Decimal(μi))) *
      ↪ Decimal(F1)) - (
          (Decimal(l1) / ((Decimal(n + 1))
          ↪ * Decimal(μi))) *
          ↪ Decimal(F2)) - (
          (Decimal(ϑj) / ((Decimal(n + 1)) *
          ↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= ts and n < C:
ϑi = ϑs
j = 1 - i
if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μs
λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
      * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
↪ 1)) * Decimal(μi))) * Decimal(S2)) - (
          (Decimal(ϑj) / ((Decimal(n + 1)) *
          ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
      * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n +
↪ 1)) * Decimal(μi))) * Decimal(F2)) - (
          (Decimal(ϑj) / ((Decimal(n + 1)) *
          ↪ Decimal(μi))) * Decimal(Q1))

Summ1 += S
Summ2 += F
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
          (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (

```

```

        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

A = S
D = F
X = λ
elif n == C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1))
        ↪ * Decimal(μi))) *
        ↪ Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= C and n < C + K:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(C)) * Decimal(μi)))
        * Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (

```

```

        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
↪ Decimal(ϑi)) / ((Decimal(C)) * Decimal(μi)))
    * Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(C)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Summ1 += S
Summ2 += F
if n == K + C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(S2))
        ↪ - (
            (Decimal(ϑj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(F2))
        ↪ - (
            (Decimal(ϑj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == K + C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(S2))
        ↪ - (
            (Decimal(ϑj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(F2))
        ↪ - (
            (Decimal(ϑj) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
S2 = S1
S1 = S
F2 = F1

```

```

F1 = F
l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(L1)) - ((Decimal(k1) / ((Decimal(n +
    ↪ 1)) * Decimal(μi))) * Decimal(L2)) - (
    (Decimal(ϑj) / (Decimal((n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(Q1)) - ((Decimal(k1) / ((Decimal(n +
    ↪ 1)) * Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2))
    Summ1 += L
    Summ2 += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) *
    ↪ Decimal(L2)) - (
    (Decimal(ϑj) / (Decimal((n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(n + 1))
    ↪ * Decimal(μi))) *
    ↪ Decimal(Q2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm

```

```

 $\mu_i = \mu_m$ 
 $k = \lambda_n$ 
L = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
    (Decimal(C) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(C) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(L2)) - (
    (Decimal( $\vartheta_j$ ) / (Decimal(C) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
Q = (((Decimal(n) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
    (Decimal(C) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(C) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(Q2)) - (
    (Decimal( $\vartheta_j$ ) / ((Decimal(C) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(F2))

Summ1 += L
Summ2 += Q
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
 $\vartheta_i = \vartheta_m$ 
j = 1 - i
if j == 0:
     $\vartheta_j = \vartheta_s$ 
else:
     $\vartheta_j = \vartheta_m$ 
 $\mu_i = \mu_m$ 
 $k = \lambda_n$ 
L = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
    (Decimal(C) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(L1)) - (
    (Decimal( $\vartheta_j$ ) / (Decimal(C) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
Q = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) / (
    (Decimal(C) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(Q1)) - (
    (Decimal( $\vartheta_j$ ) / ((Decimal(C) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(F2))

Summ1 += L
Summ2 += Q
if n == K + C - 1:
    L = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
    ↪ (
    (Decimal(C) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(L1)) - (
    (Decimal( $\vartheta_j$ ) / (Decimal(C) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(S2))
    Q = (((Decimal(C) * Decimal( $\mu_i$ ) + Decimal( $\vartheta_i$ )) /
    ↪ (
    (Decimal(C) * Decimal( $\mu_i$ ))) *
    ↪ Decimal(Q1)) - (
    (Decimal( $\vartheta_j$ ) / ((Decimal(C) *
    ↪ Decimal( $\mu_i$ ))) * Decimal(F2))

P = L

```

```

        G = Q
        L2 = L1
        L1 = L
        Q2 = Q1
        Q1 = Q
        k1 = k
r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    ↪ Decimal(ϑm) * P) / - (Decimal(C) * Decimal(μs)
        * D + Decimal(ϑs) * D - X * E - Decimal(ϑm) * G)
P00 = 1 / (Summ1 + r * Summ2)
P10 = r * P00

from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new = 0
Bm_new = 0
Summ_P0C = 0
Summ_P1C = 0
Summ_P0C_K = 0
Summ_P0_K_U = 0
Summ_P1_K_U = 0
Summ_NQ_0 = 0
Summ_NQ_1 = 0
Summ_λe_1 = 0
Summ_λe_2 = 0
Summ_λe_3 = 0
for n in range(0, C + K):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0C += n * Pin
            Summ_P0C_K += Pin
            Summ_λe_1 += (λh + λn) * Pin
            if i == 0 and n >= ts - 1 and n < C + K:
                Bs_new += Pin
            if i == 0 and n == C - 1:
                Bs_handoff = Pin
            if i == 0 and n == C + K - 1:
                P0_C_K = Pin
            if i == 0 and n >= C and n < C + K:
                P0_C_K_U = Pin
                Summ_P0_K_U += C * P0_C_K_U
                Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P1C += n * Pin

```



```

Summ_λe_3 += λn * Pin
if i == 1 and n >= C - 1:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 0 and n <= ts - 1:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
    ↪ 1)) * Decimal(μi))) * Decimal(S2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
    * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n +
    ↪ 1)) * Decimal(μi))) * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    Summ_P0C += (n + 1) * Pin
    Summ_P0C_K += Pin
    if i == 0 and n >= ts - 1 and n < C + K:
        Bs_new += Pin
    if i == 0 and n == C - 1:
        Bs_handoff = Pin
    if i == 0 and n == C + K - 1:
        P0_C_K = Pin
    if i == 0 and n >= C and n < C + K:
        P0_C_K_U = Pin
        Summ_P0_K_U += C * P0_C_K_U
        Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
    if i == 0 and n < ts - 1:
        Summ_λe_1 += λ * Pin
    if i==0 and n >= ts - 1 and n < C + K - 1:
        Summ_λe_2 += λh * Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs

```

```

λ = λh
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
  ↳ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
  * Decimal(S1)) - ((Decimal(l1) / ((Decimal(n +
  ↳ 1)) * Decimal(μi))) * Decimal(S2)) - (
  (Decimal(ϑj) / ((Decimal(n + 1)) *
  ↳ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
  ↳ Decimal(ϑi)) / ((Decimal(n + 1)) * Decimal(μi)))
  * Decimal(F1)) - ((Decimal(l1) / ((Decimal(n +
  ↳ 1)) * Decimal(μi))) * Decimal(F2)) - (
  (Decimal(ϑj) / ((Decimal(n + 1)) *
  ↳ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0C += (n + 1) * Pin
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1:
    Bs_new += Pin
if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 0 and n >= C and n < C + K:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
S = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
  ↳ Decimal(ϑi)) / ((Decimal(C)) * Decimal(μi)))
  * Decimal(S1)) - ((Decimal(l1) / ((Decimal(C)) *
  ↳ Decimal(μi))) * Decimal(S2)) - (
  (Decimal(ϑj) / ((Decimal(C)) *
  ↳ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(C) * Decimal(μi) +
  ↳ Decimal(ϑi)) / ((Decimal(C)) * Decimal(μi)))
  * Decimal(F1)) - ((Decimal(l1) / ((Decimal(C)) *
  ↳ Decimal(μi))) * Decimal(F2)) - (
  (Decimal(ϑj) / ((Decimal(C)) *
  ↳ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0C_K += Pin
if i == 0 and n >= ts - 1 and n < C + K:
    Bs_new += Pin

```

```

if i == 0 and n == C - 1:
    Bs_handoff = Pin
if i == 0 and n == C + K - 1:
    P0_C_K = Pin
if i == 0 and n >= C and n < C + K:
    P0_C_K_U = Pin
    Summ_P0_K_U += C * P0_C_K_U
    Summ_NQ_0 += (n + 1 - C) * P0_C_K_U
if i == 0 and n >= ts - 1 and n < C + K - 1:
    Summ_λe_2 += λh * Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
    μi = μm
k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(ϑj) / (Decimal((n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P1C += (n + 1) * Pin
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
if i == 1 and n < C - 1:
    Summ_λe_3 += λn * Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n == C:
    ϑi = ϑm
    j = 1 - i

```

```

if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μm
k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
    (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) -
    ↪ (
        (Decimal(k1) / ((Decimal(C)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
if i == 1 and n >= C - 1 and n < C + K:
    Bm_new += Pin
if i == 1 and n >= C and n < C + K:
    P1_C_K_U = Pin
    Summ_P1_K_U += C * P1_C_K_U
    Summ_NQ_1 += (n + 1 - C) * P1_C_K_U
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > C and n < C + K:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(ϑj) / (Decimal(C) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(C) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(C)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(ϑj) / ((Decimal(C)) *
                ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    if i == 1 and n >= C - 1 and n < C + K:
        Bm_new += Pin
    if i == 1 and n >= C and n < C + K:
        P1_C_K_U = Pin
        Summ_P1_K_U += C * P1_C_K_U
        Summ_NQ_1 += (n + 1 - C) * P1_C_K_U

```

```

        L2 = L1
        L1 = L
        Q2 = Q1
        Q1 = Q
        k1 = k

Bnew = Bs_new + Bm_new
U = Summ_P0C + Summ_P1C
P_HCDP = P0_C_K / Summ_P0C_K
U2 = Summ_P0_K_U + Summ_P1_K_U
Utilization = (U + U2) / C
NQ = Summ_NQ_0 + Summ_NQ_1
λe = Summ_λe_1 + Summ_λe_2 + Summ_λe_3
WQ = (NQ / λe) * 60
if K == 0:
    list_Bnew_1_θm_K_0.append(Bnew)
    list_HCDP_1_θm_K_0.append(P_HCDP)
elif K == 6:
    list_Bnew_1_θm_K_6.append(Bnew)
    list_HCDP_1_θm_K_6.append(P_HCDP)
elif K == 14:
    list_Bnew_1_θm_K_14.append(Bnew)
    list_HCDP_1_θm_K_14.append(P_HCDP)
print("H lista Bnew gia K=0 (1/θm) einai: " +
      ↪ str(list_Bnew_1_θm_K_0))
print("H lista Bnew gia K=6 (1/θm) einai: " +
      ↪ str(list_Bnew_1_θm_K_6))
print("H lista Bnew gia K=14 (1/θm) einai: " +
      ↪ str(list_Bnew_1_θm_K_14))
print("H lista HCDP gia K=0 (1/θm) einai: " +
      ↪ str(list_HCDP_1_θm_K_0))
print("H lista HCDP gia K=6 (1/θm) einai: " +
      ↪ str(list_HCDP_1_θm_K_6))
print("H lista HCDP gia K=14 (1/θm) einai: " +
      ↪ str(list_HCDP_1_θm_K_14))

      ↪ print("-----")
# NP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0

```

```

Summ_S0_in_NP = Decimal(0)
Summ_S1_in_NP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ_S0_in_NP += S1
            Summ_S1_in_NP += F1
        elif i == 1 and n == 0:
            Summ_S0_in_NP += L1
            Summ_S1_in_NP += Q1
        if i == 0 and n < C:
            ϑi = ϑs
            j = 1 - i
            if j == 0:
                ϑj = ϑs
            else:
                ϑj = ϑm
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(ϑi)) / (
                    (Decimal(n + 1)) * Decimal(μi))) *
                ↪ Decimal(S1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(S2)) - (
                    (Decimal(ϑj) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(L1))
            F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(ϑi)) / (
                    (Decimal(n + 1)) * Decimal(μi))) *
                ↪ Decimal(F1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(F2)) - (
                    (Decimal(ϑj) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(Q1))
            Summ_S0_in_NP += S
            Summ_S1_in_NP += F
        if n == C - 1:
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(ϑi)) / (
                    (Decimal(n + 1)) * Decimal(μi))) *
                ↪ Decimal(S1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(S2)) - (
                    (Decimal(ϑj) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(L1))
            F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(ϑi)) / (
                    (Decimal(n + 1)) * Decimal(μi))) *
                ↪ Decimal(F1)) - (
                    (Decimal(l1) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(F2)) - (
                    (Decimal(ϑj) / ((Decimal(n + 1)) *
                        ↪ Decimal(μi))) * Decimal(Q1))

            A = S
            D = F
            X = λ

```

```

elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
                (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))

    Summ_S0_in_NP += L
    Summ_S1_in_NP += Q
if n == C - 1:
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (

```

```

        (Decimal(ϑj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k

r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    ↪ Decimal(ϑm) * P) / - (
        Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
        ↪ Decimal(ϑm) * G)
P00 = 1 / (Summ_S0_in_NP + r * Summ_S1_in_NP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_NP = 0
Bm_new_NP = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n == C - 1:
                Bs_new = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n == C - 1:
                Bm_new = Pin
        if i == 0 and n < C:
            ϑi = ϑs
            j = 1 - i
            if j == 0:
                ϑj = ϑs
            else:
                ϑj = ϑm
            μi = μs

```



```

λ = λh + λn
S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
      (Decimal(n + 1)) * Decimal(μi))) * Decimal(
S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(S2)) - (
      (Decimal(ϑj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
      (Decimal(n + 1)) * Decimal(μi))) * Decimal(
F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2)) - (
      (Decimal(ϑj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(Q1))
Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n == C - 1:
    Bs_new_NP = Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
      (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
↪ (
      (Decimal(k1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(L2)) - (
      (Decimal(ϑj) / (Decimal((n + 1)) *
↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
      (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
↪ (
      (Decimal(k1) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(Q2)) - (
      (Decimal(ϑj) / ((Decimal(n + 1)) *
↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n == C - 1:
    Bm_new_NP = Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q

```

```

        k1 = k
    Bnew_NP = Bs_new_NP + Bm_new_NP
    U = Summ_P_n_0n + Summ_P_n_1n
    P_HCDP_NP = Bs_new_NP / Summ_P0n
    Utilization_NP = U / C
    list_Bnew_1_0m_NP.append(Bnew_NP)
    list_HCDP_1_0m_NP.append(P_HCDP_NP)
    print("H lista K einai: " + str(K))
    print("H lista Bnew_NP (for 0m) einai: " + str(list_Bnew_1_0m_NP))
    print("H lista HCDP_NP (for 0m) einai: " + str(list_HCDP_1_0m_NP))

    ↪ print("-----")
# CP Scheme
l1 = 0
k1 = 0
from decimal import *

ctx = getcontext()
ctx.prec = 50
ctx.rounding = ROUND_UP
S2 = Decimal(0)
S1 = Decimal(1)
L2 = Decimal(0)
L1 = Decimal(0)
F2 = Decimal(0)
F1 = Decimal(0)
Q2 = Decimal(0)
Q1 = Decimal(1)
A = 0
B = 0
P = 0
D = 0
E = 0
G = 0
Summ_S0_in_CP = Decimal(0)
Summ_S1_in_CP = Decimal(0)
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Summ_S0_in_CP += S1
            Summ_S1_in_CP += F1
        elif i == 1 and n == 0:
            Summ_S0_in_CP += L1
            Summ_S1_in_CP += Q1
        if i == 0 and n <= ts - 1:
            0i = 0s
            j = 1 - i
            if j == 0:
                0j = 0s
            else:
                0j = 0m
            μi = μs
            λ = λh + λn
            S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
                ↪ Decimal(0i)) / (
                    (Decimal(n + 1)) * Decimal(μi))) *
                ↪ Decimal(S1)) - (

```

```

        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L1))
F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q1))
Summ_S0_in_CP += S
Summ_S1_in_CP += F
if n == C - 1:
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(S1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
        ↪ Decimal(ϑi)) / (
            (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(F1)) - (
            (Decimal(l1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q1))

    B = S
    E = F
S2 = S1
S1 = S
F2 = F1
F1 = F

```

```

l1 = λ
elif i == 0 and n >= ts and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
    Summ_S0_in_CP += S
    Summ_S1_in_CP += F
if n == C - 1:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

    A = S
    D = F
    X = λ
elif n == C - 2:
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(S1)) - (
        (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))

```

```

F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(F1)) - (
    (Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

B = S
E = F
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n <= ts - 1:
ϑi = ϑm
j = 1 - i
if j == 0:
    ϑj = ϑs
else:
    ϑj = ϑm
μi = μm
k = λn
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (
    (Decimal(ϑj) / (Decimal((n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q2)) - (
    (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2))

Summ_S0_in_CP += L
Summ_S1_in_CP += Q
if n == C - 1:
L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(L1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L2)) - (
    (Decimal(ϑj) / (Decimal((n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
↪ Decimal(ϑi)) / (
    (Decimal(n + 1)) * Decimal(μi))) *
    ↪ Decimal(Q1)) - (
    (Decimal(k1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q2)) - (

```

```

                                (Decimal(ϑj) / ((Decimal(n + 1)) *
                                ↪ Decimal(μi))) * Decimal(F2))
    P = L
    G = Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == ts:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2))

    Summ_S0_in_CP += L
    Summ_S1_in_CP += Q
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n > ts and n < C:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(ϑj) / (Decimal((n + 1)) *
            ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(F2))

```

```

Summ_S0_in_CP += L
Summ_S1_in_CP += Q
if n == C - 1:
    L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(L1)) - (
            (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) *
        ↪ Decimal(Q1)) - (
            (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))

    P = L
    G = Q

    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k

r = (Decimal(C) * Decimal(μs) * A + Decimal(ϑs) * A - X * B -
    ↪ Decimal(ϑm) * P) / - (
        Decimal(C) * Decimal(μs) * D + Decimal(ϑs) * D - X * E -
        ↪ Decimal(ϑm) * G)
P00 = 1 / (Summ_S0_in_CP + r * Summ_S1_in_CP)
P10 = r * P00

S1 = Decimal(1)
L1 = Decimal(0)
F1 = Decimal(0)
Q1 = Decimal(1)
S2 = Decimal(0)
L2 = Decimal(0)
F2 = Decimal(0)
Q2 = Decimal(0)
Bs_new_CP = 0
Bm_new_CP = 0
Summ_P0S = 0
Summ_P1S = 0
Summ_P0n = 0
Summ_P_n_0n = 0
Summ_P_n_1n = 0
P0C = 0
for n in range(0, C):
    for i in range(0, 2):
        if i == 0 and n == 0:
            Pin = S1 * P00 + F1 * P10
            Summ_P0n += Pin
            Summ_P_n_0n += n * Pin
            if i == 0 and n >= ts - 1 and n < C:
                Summ_P0S += Pin
            if i == 0 and n == C - 1:
                P0C = Pin
        elif i == 1 and n == 0:
            Pin = L1 * P00 + Q1 * P10
            Summ_P_n_1n += n * Pin
            if i == 1 and n >= ts - 1 and n < C:
                Summ_P1S += Pin

```

```

if i == 0 and n <= ts - 1:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh + λn
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))
    Pin = Decimal(S) * P00 + Decimal(F) * P10
    Summ_P0n += Pin
    Summ_P_n_0n += (n + 1) * Pin
    if i == 0 and n >= ts - 1 and n < C:
        Summ_POS += Pin
    if i == 0 and n == C - 1:
        P0C = Pin
    S2 = S1
    S1 = S
    F2 = F1
    F1 = F
    l1 = λ
elif i == 0 and n >= ts and n < C:
    ϑi = ϑs
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μs
    λ = λh
    S = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    S1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(S2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(L1))
    F = (((Decimal(λ) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(
    F1)) - ((Decimal(l1) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(F2)) - (
        (Decimal(ϑj) / ((Decimal(n + 1)) *
    ↪ Decimal(μi))) * Decimal(Q1))

```



```

Pin = Decimal(S) * P00 + Decimal(F) * P10
Summ_P0n += Pin
Summ_P_n_0n += (n + 1) * Pin
if i == 0 and n >= ts - 1 and n < C:
    Summ_P0S += Pin
if i == 0 and n == C - 1:
    P0C = Pin
S2 = S1
S1 = S
F2 = F1
F1 = F
l1 = λ
elif i == 1 and n <= ts - 1:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(L2)) - (
                (Decimal(ϑj) / (Decimal((n + 1)) *
                ↪ Decimal(μi))) * Decimal(S2))
    Q = (((Decimal(k) + Decimal(n) * Decimal(μi) +
    ↪ Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
            (Decimal(k1) / ((Decimal(n + 1)) *
            ↪ Decimal(μi))) * Decimal(Q2)) - (
                (Decimal(ϑj) / ((Decimal(n + 1)) *
                ↪ Decimal(μi))) * Decimal(F2))
    Pin = Decimal(L) * P00 + Decimal(Q) * P10
    Summ_P_n_1n += (n + 1) * Pin
    if i == 1 and n >= ts - 1 and n < C:
        Summ_P1S += Pin
    L2 = L1
    L1 = L
    Q2 = Q1
    Q1 = Q
    k1 = k
elif i == 1 and n == ts:
    ϑi = ϑm
    j = 1 - i
    if j == 0:
        ϑj = ϑs
    else:
        ϑj = ϑm
    μi = μm
    k = λn
    L = (((Decimal(n) * Decimal(μi) + Decimal(ϑi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (

```

```

        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(L2)) - (
        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
        (Decimal(k1) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(Q2)) - (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
elif i == 1 and n > ts and n < C:
    θi = θm
    j = 1 - i
    if j == 0:
        θj = θs
    else:
        θj = θm
    μi = μm
    k = λn
L = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(L1)) -
        ↪ (
        (Decimal(θj) / (Decimal((n + 1)) *
        ↪ Decimal(μi))) * Decimal(S2))
Q = (((Decimal(n) * Decimal(μi) + Decimal(θi)) / (
        (Decimal(n + 1)) * Decimal(μi))) * Decimal(Q1)) -
        ↪ (
        (Decimal(θj) / ((Decimal(n + 1)) *
        ↪ Decimal(μi))) * Decimal(F2))
Pin = Decimal(L) * P00 + Decimal(Q) * P10
Summ_P_n_1n += (n + 1) * Pin
if i == 1 and n >= ts - 1 and n < C:
    Summ_P1S += Pin
L2 = L1
L1 = L
Q2 = Q1
Q1 = Q
k1 = k
Bnew_CP = Summ_P0S + Summ_P1S
U = Summ_P_n_0n + Summ_P_n_1n
P_HCDP_CP = P0C / Summ_P0n
Utilization_CP = U / C
list_Bnew_1_θm_CP.append(Bnew_CP)
list_HCDP_1_θm_CP.append(P_HCDP_CP)
print("H lista Bnew_CP(for θm) einai: " + str(list_Bnew_1_θm_CP))
print("H lista HCDP_CP(for θm) einai: " + str(list_HCDP_1_θm_CP))

↪ print("-----")

```

```

import matplotlib.pyplot as plt
import numpy as np

plt.figure(16)
plt.title('NCBP versus  $1/\vartheta_M$ ')
plt.ylabel('New call blocking probability')
plt.xlabel('Average dwell time in stop phase ( $1/\vartheta_M$ )')
x1 = np.array(list_1_0m)
y1 = np.array(list_Bnew_1_0m_K_0)
y2 = np.array(list_Bnew_1_0m_K_6)
y3 = np.array(list_Bnew_1_0m_K_14)
y4 = np.array(list_Bnew_1_0m_NP)
y5 = np.array(list_Bnew_1_0m_CP)
plt.xticks(np.arange(0, 3.5, step=0.5))
plt.yticks(np.arange(0.1, 1.1, step=0.1))
plt.xlim(0, 3)
plt.ylim(0.1, 1)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, 'v-k', ms = 11, label = 'MA-CAC (K=6)')
plt.plot(x1, y3, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y4, '^~r', ms = 11, label = 'NP')
plt.plot(x1, y5, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()

plt.figure(17)
plt.title('NCDP versus  $1/\vartheta_M$ ')
plt.ylabel('Handover call dropping probability')
plt.xlabel('Average dwell time in stop phase ( $1/\vartheta_M$ )')
plt.xticks(np.arange(0, 5.5, step=0.5))
plt.yticks(np.arange(0, 0.85, step=0.1))
plt.xlim(0, 5)
plt.ylim(0, 0.8)
x1 = np.array(list_1_0m)
y1 = np.array(list_HCDP_1_0m_K_0)
y2 = np.array(list_HCDP_1_0m_K_6)
y3 = np.array(list_HCDP_1_0m_K_14)
y4 = np.array(list_HCDP_1_0m_NP)
y5 = np.array(list_HCDP_1_0m_CP)
lines = plt.plot(x1, y1, x1, y2)
plt.setp(lines, linewidth = 0.5)
plt.plot(x1, y1, 'o-b', ms = 11, label = 'MA-CAC (K=0)')
plt.plot(x1, y2, 'v-k', ms = 11, label = 'MA-CAC (K=6)')
plt.plot(x1, y3, '*-g', ms = 11, label = 'MA-CAC (K=14)')
plt.plot(x1, y4, '^~r', ms = 11, label = 'NP')
plt.plot(x1, y5, 's-y', ms = 11, label = 'CP')
plt.legend(loc = 'lower right')
plt.grid()
plt.show()

```

Βιβλιογραφία

- [1] Y. Kim, H. Ko, S. Pack, W. Lee, and X. Shen, "Mobility-aware call admission control algorithm with handoff queue in mobile hotspots," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 8, pp. 3903–3912, 2013.
- [2] T. Kerdoncuff, T. Galezowski, and X. Lagrange, "Mobile relay for lte: Proof of concept and performance measurements," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, Porto, Portugal, 2018, pp. 1–5.
- [3] D. Niyato and E. Hossain, "Integration of iee 802.11 wlans with iee 802.16-based multihop infrastructure mesh/relay networks: A game-theoretic approach to radio resource management," *Netw. Mag. of Global Internetwkg.*, vol. 21, no. 3, p. 6–14, May 2007. [Online]. Available: <https://doi.org/10.1109/MNET.2007.364253>
- [4] C.-W. Lee, Y. S. Sun, and M. C. Chen, "Himip-nemo: Combining cross-layer network mobility management and resource allocation for fast qos-handovers," *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pp. 2282–2286, 2008.
- [5] Y. Kirsal, G. Mapp, and F. Sardis, "Using advanced handover and localization techniques for maintaining quality-of-service of mobile users in heterogeneous cloud-based environment," *Journal of Network and Systems Management*, vol. 27, pp. 972–997, 2019.
- [6] J.-T. Wang, Y.-Y. Hsu, and C.-C. Tseng, "A bandwidth-sharing reservation scheme to support qos for network mobility," in *2006 IEEE International Conference on Communications*, vol. 2, Istanbul, Turkey, pp. 693–698.
- [7] Y. Li, W. Zhou, and S. Zhou, "Forecast based handover in an extensible multi-layer leo mobile satellite system," *IEEE Access*, vol. 8, pp. 42 768–42 783, 2020.
- [8] S. Pack, X. Shen, J. W. Mark, and L. Cai, "Throughput analysis of tcp-friendly rate control in mobile hotspots," *IEEE Transactions on Wireless Communications*, vol. 7, no. 1, pp. 193–203, 2008.
- [9] W. Song and W. Zhuang, "Resource reservation for self-similar data traffic in cellular/wlan integrated mobile hotspots," in *2010 IEEE International Conference on Communications*. Cape Town, South Africa: IEEE, 2010, pp. 1–5.
- [10] S. Choi, H. Chung, J. Kim, J. Ahn, and I. Kim, "Mobile hotspot network system for high-speed railway communications using millimeter waves," *ETRI Journal*, vol. 38, no. 6, pp. 1052–1063, 2016.

- [11] H. Lee, J. Flinn, and B. Tonshal, "Raven: Improving interactive latency for the connected car," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, New Delhi, India, 2018, pp. 557–572.
- [12] S. Pack, X. Shen, J. W. Mark, and J. Pan, "Mobility management in mobile hotspots with heterogeneous multihop wireless links," *IEEE Communications Magazine*, vol. 45, no. 9, pp. 106–112, 2007.
- [13] W.-K. Chiang, W.-Y. Chang, and L.-Y. Liu, "Simultaneous handover support for mobile networks on vehicles," in *2008 IEEE Wireless Communications and Networking Conference*. Las Vegas, Nevada, USA: IEEE, 2008, pp. 2771–2776.
- [14] Z. Zhang, A. Boukerche, and R. W. Pazzi, "A novel network mobility management scheme for vehicular networks," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. Miami, Florida, USA: IEEE, 2010, pp. 1–5.
- [15] S. M. M. Gilani, T. Hong, W. Jin, G. Zhao, H. M. Heang, and C. Xu, "Mobility management in ieee 802.11 wlan using sdn/nfv technologies," *EURASIP Journal on Wireless Communications and Networking*, vol. 2017, no. 1, pp. 1–14, 2017.
- [16] N. Aljeri and A. Boukerche, "A two-tier machine learning-based handover management scheme for intelligent vehicular networks," *Ad Hoc Networks*, vol. 94, p. 101930, 2019.
- [17] L. Kleinrock, "Queuing systems, vol 1: theory," 1975.