

ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

ΣΧΟΛΗ ΕΠΙΣΤΗΜΗ ΤΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



Πτυχιακή εργασία

Έλεγχος Αποδοχής Κλήσεων σε Δυναμικά Δικτυακά Περιβάλλοντα με
Ανάλυση Ποιότητας Υπηρεσιών

Ρίζου Αναστασία Α.Μ.: Π2019156

Επιβλέπων: Κωνσταντίνος Οικονόμου

27 Ιουνίου 2023

Επιβλέπων

Κωνσταντίνος Οικονόμου, Καθηγητής,
Ιόνιο Πανεπιστήμιο

Τριμελής Επιτροπή

Κωνσταντίνος Οικονόμου, Καθηγητής,
Ιόνιο Πανεπιστήμιο

Θεόδωρος Ανδρόνικος, Αναπληρωτής Καθηγητής,
Ιόνιο Πανεπιστήμιο

Μιχαήλ Στεφανιδάκης, Αναπληρωτής Καθηγητής,
Ιόνιο Πανεπιστήμιο

Η προσπάθεια αυτή είναι αφιερωμένη στην οικογένειά μου.

Περίληψη

Τα δίκτυα Jackson είναι συστήματα αναμονής που ουσιαστικά απαρτίζονται από πολλά μικρότερα συστήματα αναμονής. Δηλαδή μια συλλογή απλών συστημάτων αναμονής, στην οποία κάθε «πελάτης» που αναχωρεί από ένα απλό σύστημα, μετακινείται σε κάποιο άλλο (πιθανώς το ίδιο), ή αναχωρεί από το δίκτυο, αν αυτό επιτρέπεται, όπως και ένα σύστημα αναμονής $M/M/1$. [4]. Η αποδοχή μιας κλήσης μετά από έλεγχο συσχετίζεται άμεσα με την καθυστέρηση και τον χρόνο επεξεργασίας σε ένα σύστημα αναμονής όταν σε αυτό υπάρχουν ήδη «πακέτα - πελάτες - συμβάντα». Σκοπός της συγκεκριμένης πτυχιακής εργασίας αποτέλεσε η μελέτη του πειραματικού δικτύου που υλοποιήθηκε στην πλατφόρμα OMNET++. Εκεί παρατηρήθηκε και αναλύθηκε η συμπεριφορά του δικτύου δεδομένων των διαφόρων παρογόντων όπως το προγραμματιστικό περιβάλλον και το μέγεθος του δικτύου. Τα τρία εσωτερικά μικρότερα συστήματα αναμονής που αποτελούν το ευρύτερο δίκτυο αποτελούνται από μια γεννήτρια συμβάντων, μια ουρά αναμονής και έναν επεξεργαστή. Ο κώδικας για την υλοποίηση και εξαγωγή συμπερασμάτων είναι στην γλώσσα προγραμματισμού C++ όπως η πλατφόρμα υποστηρίζει. Χρησιμοποιώντας κλάσεις αυτής της γλώσσας προγραμματισμού, μηνύματα, ονομασία και δίνοντας μοναδικό αριθμό σε κάθε κόμβο αλλά και χρησιμοποιώντας την ιεραρχική δομή που προσφέρει η πλατφόρμα οπτικοποιήθηκε και ελέγχθηκε η ορθότητα του δικτύου, η συλλογή των δεδομένων καθώς και η εξαγωγή των συμπερασμάτων. Για την καλύτερη κατανόηση της συμπεριφοράς του δικτύου η αναπαράσταση των συμπερασμάτων υλοποιήθηκε στην ανοικτού κώδικα πλατφόρμα γραφικών GNUplot.

Abstract

Jackson networks are queuing systems that are essentially made up of many smaller queuing systems. That is, a collection of simple ones waiting systems, in which each "customer" who departs from a simple system, moves to one another (probably the same), or leaves the network, if it is allowed, as is a M/M/1 waiting system.[4]. The acceptance of a call after screening is directly related to the delay and processing time in a waiting system when there are already "packages - customers - events" in it. The purpose of this thesis was the study of the experimental network implemented on the OMNET++ platform. There the behavior of the network given various factors such as the programming environment and the size of the network was observed and analyzed. The three internal smaller queuing systems that make up the wider network consist of an event generator, a queue, and a processor. The code for implementation and inference is in the programming language C++ as supported by the platform. Using classes of this programming language, messages, naming and giving a unique number to each node, but also using the hierarchical structure offered by the platform, the correctness of the network was visualized and checked, the collection of the data as well as the drawing of the conclusions. For a better understanding of the behavior of the network, the representation of the results was implemented in the open source graphics platform GNUplot.

Πρόλογος και Ευχαριστίες

Η προσπάθεια υλοποίησης της συγκεκριμένης πτυχιακής εργασίας αποτέλεσε μια προσπάθεια με αρκετές δυσκολίες, αλλά και ένα όμορφο ταξίδι που ολοκλήρωσε το κεφάλαιο των σπουδών μου με τον καλύτερο δυνατό τρόπο. Όπως και σε κάθε σημαντικό στάδιο της ζωής μας έτσι και σε αυτό δεν έλειπαν οι δυσκολίες και οι απογοητεύσεις, οι οποίες σου δίνουν, με τον καιρό, θέληση και πείσμα για να συνεχίσεις και να δώσεις τον καλύτερο σου εαυτό, γνωρίζοντας πως οι κόποι σου αυτοί θα ανταμειφθούν και θα σε βοηθήσουν στο να δημιουργήσεις κάτι όμορφο και ουσιαστικό. Έτσι ήταν το ταξίδι της εκπόνησης της πτυχιακής μου εργασίας και ελπίζω ότι έδωσα τον καλύτερό μου εαυτό για ένα αποτέλεσμα που να με αντιπροσωπεύει. Ωστόσο, το αποτέλεσμα δεν ήταν αποκλειστικά δική μου υπόθεση. Αντίθετα ήταν αποτέλεσμα συλλογικής προσπάθειας όλων εκείνων που πίστεψαν και συνέβαλαν με τον δικό τους τρόπο στην υλοποίησης του. Το αποτέλεσμα δεν θα ήταν το ίδιο χωρίς την βοήθεια και τη στήριξη της οικογένειάς μου και των φίλων μου. Ένα ιδιαίτερο ευχαριστώ θα ήθελα να πω στο φίλο και πλέον συναδέλφο που με βοήθησε και με στήριξε σε όλη την διάρκεια της υλοποίησης της πτυχιακής εργασίας με τον δικό του τρόπο, στον Βασίλειο Δραγώνα. Για τη στήριξη και τη βοήθεια που μου προσέφερε όλο αυτόν τον καιρό, θα ήθελα να ευχαριστήσω τον καθηγητή μου κ.Κωνσταντίνο Οικονόμου, αλλά και όλους τους καθηγητές που γνώρισα και συνάντησα σε όλη την διάρκεια των σπουδών μου, για τις γνώσεις, τις πληροφορίες και τις εμπειρίες που μου χάρισαν. Επίσης θα ήθελα να ευχαριστήσω όλους όσων η στήριξη και η πίστη σε εμένα με έκαναν να συνεχίσω να προσπαθώ και να ανταπεξέρχομαι σε όποια δυσκολία αντιμετώπιζα.

Κέρκυρα, 27 Ιουνίου 2023

Αναστασία Ρίζου

Περιεχόμενα

A'	Εισαγωγή	1
A'.1	ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΡΕΥΝΑ	1
A'.2	ΑΠΟΤΕΛΕΣΜΑΤΑ ΚΑΙ ΣΥΝΕΙΣΦΟΡΑ	2
A'.3	ΑΝΤΙΣΤΟΙΧΟΥΜΕΝΑ ΚΕΦΑΛΑΙΑ	2
B'	Θεωρία Ουρών	3
B'.1	ΘΕΩΡΙΑ ΟΥΡΩΝ	3
B'.2	ΤΟ ΘΕΩΡΗΜΑ ΤΟΥ LITTLE	6
B'.3	ΣΥΣΤΗΜΑ ΑΝΑΜΟΝΗΣ M/M/1	7
Γ'	Δίκτυα Jackson	9
Γ'.1	ΔΙΚΤΥΑ JACKSON	9
Γ'.2	ΒΑΣΙΚΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΝΟΣ ΔΙΚΤΥΟΥ JACKSON	10
Γ'.3	ΔΙΚΤΥΑ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ	10
Γ'.4	CALL ADMISSION CONTROL (CAC)	22
Γ'.5	ΠΕΡΙΓΡΑΦΗ ΠΕΙΡΑΜΑΤΙΚΟΥ ΔΙΚΤΥΟΥ	24
Δ'	Υλοποίηση Δικτύου	26
Δ'.1	ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΠΕΡΙΓΡΑΦΗ ΔΙΚΤΥΟΥ	26
Δ'.1.1	Πλατφόρμα OMNET++	26
Δ'.1.1.1	Το NED αρχείο	27
Δ'.1.1.2	Το .h αρχείο	27
Δ'.1.1.3	Το .cc αρχείο	27
Δ'.1.1.4	Το omnetpp.ini αρχείο	28

Περιεχόμενα	v
Δ'.1.2 Περιγραφή πειραματικού δικτύου	28
Δ'.2 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ ΣΥΛΛΟΓΗΣ	28
Δ'.3 ΔΥΣΚΟΛΙΕΣ ΚΑΙ ΠΡΟΒΛΗΜΑΤΑ ΚΑΤΑ ΤΗΝ ΥΛΟΠΟΙΗΣΗ	29
Ε' Σύνοψη και Συμπεράσματα	33
Βιβλιογραφία	35
Παράρτημα 'Α'	36
Συντμήσεις	53
Γλωσσάρι Ξενικών Όρων	54

Κατάλογος Σχημάτων

B'.1	Σειριακή σύνδεση M ουρών	8
B'.2	Πειραματικό δίκτυο M/M/1	8
Γ'.3	(α)Πολυγωνική σύνδεση και (β) ακτινωτή σύνδεση τηλεφωνικών κέντρων [8] .	24
Γ'.4	Πειραματικό δίκτυο Jackson	25
Δ'.5	Η πλατφόρμα Omnet - Αρχική σελίδα	29
Δ'.6	Η πλατφόρμα Omnet - Επιλογή Workspace	30
Δ'.7	Η πλατφόρμα Omnet - Workspace	30
Δ'.8	Πειραματικό δίκτυο Jackson	30
Δ'.9	Αφίξεις και Αναχωρήσεις σε κατανομή Poisson του δικτύου Jackson	31
Δ'.10	Χρόνος αναμονής στο σύστημα απο τον πρώτο server ως τον τελικο	31
Δ'.11	Χρόνος αναμονής στο σύστημα απο εσωτερικό κόμβο - επεξεργαστή(server3) ως τον τελικο	32

Κεφάλαιο Α΄

Εισαγωγή

Ζούμε στην εποχή του Διαδικτύου και των διάφορων Ενσύρματων και Ασύρματων δικτύων. Ο όρος δίκτυο δεν αναφέρεται μόνο σε τεχνολογικά υλικά, υπολογιστές και κυκλώματα αλλά και σε ομάδες ανθρώπων, όπως για παράδειγμα το Facebook ή το δίκτυο αστικών λεοφορειών μιας πόλης. Τα τηλεφωνικά δίκτυα είναι το πιο αντιπροσωπευτικό παράδειγμα για να μπορέσουμε να αντιληφθούμε τις συνδέσεις, ζεύξεις μεταξύ μας, μέσω μιας κλήσης. Στη παρούσα εργασία μελετάται ένα τηλεφωνικό δίκτυο ή αλλιώς ένα δίκτυο Jackson και η καθυστέρηση της κλήσης αλλά και η αποδοχή αυτής, μέσω ενός πειραματικού δικτύου στην πλατφόρμα OMNET++ .

Α΄.1 Βιβλιογραφική Ερευνα

Η εκπόνηση της εργασίας υλοποιήθηκε για την μελέτη αποδοχής μιας κλήσης, (call admission) αλλά και πώς απαντάμε μετά απο έλεγχο (control-, Call Admission Control (CAC)). Αρχικά, αναζητήθηκαν παρόμοιες μελέτες αλλά κι μελέτες πάνω στα δίκτυα Jackson, καθώς τα τηλεφωνικά δίκτυα είναι τέτοιου τύπου δίκτυα. Αναζητήθηκαν έρευνες για την συμπεριφορά των δικτύων αυτών αλλά και έρευνες παράγοντες που πιθανόν να επηρεάζουν την συμπεριφορά του δικτύου. Σκοπός και στόχος της βιβλιογραφικής έρευνας ήταν η γενικότερη και καλύτερη κατανόηση των δικτύων Jackson αλλά και των παραγόντων που πιθανόν το επηρεάζουν. Οι παρατηρήσεις και οι πειραματισμοί των αντίστοιχων εργασιών-μελετών αποτέλεσαν επιβεβαίωση των καταγεγραμμένων μελετών, όπως άλλωστε ήταν αναμενόμενο. Συγκεκριμένα, πολλές έρευνες διεξήγαγαν τα ίδια πειράματα, τόσο ως προς τη συμπεριφορά των δικτύων Jackson

όσο και ως προς τα μετρήσιμα μεγέθη, ωστόσο η εκπόνηση της παρούσας εργασίας έχει ως στόχο να παρουσιάσει τόσο τα δίκτυα Jackson όσο και την αποδοχή μια κλήσης ή όχι μετά από έλεγχο.

Α'.2 Αποτελεσματα και Συνεισφορά

Το πρόβλημα της αποδοχής μιας κλήσης (call admission) και πώς απαντάμε μετά από έλεγχο (control) λέγεται Call Admission Control και αυτό είναι ουσιαστικά το κύριο αντικείμενο της παρούσας εργασίας. Για την εκπόνηση της πτυχιακής υλοποιήθηκε ένα πειραματικό δίκτυο Jackson στην πλατφόρμα εξομοίωσης Omnet++, ώστε μέσω από της συμπεριφοράς του και των εξαγόμενων αποτελεσμάτων να μπορέσουμε να πλησιάσουμε το παραπάνω πρόβλημα. Το πειραματικό δίκτυο αποτελείται από τρία (ή και περισσότερα $m/m/1$) και καταγράφει αφίξεις και αναχωρήσεις πακέτων - δεδομένων από όλους τους κόμβους ώστε να χρησιμοποιηθούν για την εξαγωγή συμπερασμάτων. Τα αποτελέσματα που προέκυψαν ως προς τη συμπεριφορά του δικτύου Jackson επιβεβαιώνουν τις εργασίες και μελέτες που το θέλουν να συμπεριφέρεται όπως ένα δίκτυο αναμονής $m/m/1$. Όσο για τα αποτελέσματα των καταγραφών και το πώς αντιμετωπίζουμε το πρόβλημα αποδοχής μιας κλήσης, η προσέγγιση έγινε τόσο με μαθηματικό τρόπο όσο και με σύγκριση των αποτελεσμάτων - τιμών που εξήγαγε το πειραματικό δίκτυο της εργασίας.

Α'.3 Αντιστοιχούμενα Κεφάλαια

Η δομή της παρούσας πτυχιακής εργασίας είναι τέτοια ώστε να εισάγει τον αναγνώστη ομαλά στο κύριο θέμα της. Αρχίζοντας από τον ορισμό των δικτύων και τη θεωρία ουρών στο Κεφάλαιο Β' αλλά και με παραδείγματα, ο αναγνώστης μπορεί να κατανοήσει βασικές έννοιες και να εισαχθεί στον ορισμό και την θεωρία των δικτύων Jackson, όπου και είναι το κύριο θέμα, στο Κεφάλαιο Γ', περιγράφοντας τα κύρια και βασικά χαρακτηριστικά των δικτύων αλλά και εφαρμογές αυτού. Στο Κεφάλαιο Δ' παρουσιάζεται η υλοποίηση και η ανάλυση του πειραματικού δικτύου για την εκπόνηση της εργασίας, καθώς και η πλατφόρμα Omnet++ που χρησιμοποιήθηκε για την εξαγωγή συμπερασμάτων και αποτελεσμάτων, τα οποία και παρουσιάζονται στο Κεφάλαιο Ε'.

Κεφάλαιο Β΄

Θεωρία Ουρών

Β΄.1 Θεωρία Ουρών

Η θεωρία ουράς είναι το πρωταρχικό μεθοδολογικό πλαίσιο για την ανάλυση της καθυστέρησης του δικτύου. Η χρήση του απαιτεί συχνά απλοποιητικές υποθέσεις, καθώς, δυστυχώς, πιο ρεαλιστικές παραδοχές καθιστούν εξαιρετικά δύσκολη την ουσιαστική ανάλυση. Για το λόγο αυτόμερικές φορές είναι αδύνατο να ληφθούν ακριβείς ποσοτικές προβλέψεις καθυστέρησης με βάση μοντέλα σε ουρά. Ωστόσο, αυτά τα μοντέλα συχνά παρέχουν μια βάση για επαρκή προσεγγίσεις καθυστέρησης, καθώς και πολύτιμα ποιοτικά αποτελέσματα και αξιόλογες γνώσεις. Η εργασία αυτή επικεντρώνεται κυρίως στην καθυστέρηση πακέτων εντός της επικοινωνίας, υποδίκτυο, (δηλαδή το επίπεδο δικτύου). Αυτή η καθυστέρηση είναι το άθροισμα των καθυστερήσεων σε κάθε σύνδεσμο υποδικτύου που διασχίζεται από το πακέτο. Κάθε καθυστέρηση σύνδεσης αποτελείται από τέσσερα στοιχεία.

1. Η καθυστέρηση της επεξεργασίας, processing delay, μεταξύ της ώρας λήψης του πακέτου στην αρχή του κόμβου σύνδεσμου και του χρόνου που το πακέτο εκχωρείται σε μια ουρά εξερχόμενης σύνδεσης για μετάδοση. (Σε ορισμένα συστήματα, πρέπει να προσθέσουμε σε αυτήν την καθυστέρηση κάποια επιπλέον χρόνος επεξεργασίας στο DLC και στα φυσικά επίπεδα.)
2. Η καθυστέρηση ουράς, queuing delay, μεταξύ της ώρας που το πακέτο εκχωρείται σε μια ουρά για μετάδοση και της ώρας που αρχίζει να μεταδίδεται. Κατά τη διάρκεια αυτής της περιόδου, το πακέτο περιμένει ενώ άλλα πακέτα στην ουρά μετάδοσης μεταδίδονται.
3. Η καθυστέρηση μετάδοσης, transmission delay, μεταξύ των χρόνων που το πρώτο και το τελευταίο βιτς του πακέτου μεταδίδονται.

4. Η καθυστέρηση διάδοσης, propagation delay, μεταξύ του χρόνου μετάδοσης του τελευταίου bit στον αρχικό κόμβο του συνδέσμου και ο χρόνος λήψης του τελευταίου bit στον τελευταίο κόμβο. Αυτό είναι ανάλογο με τη φυσική απόσταση μεταξύ πομπού και δέκτη· μπορεί να είναι σχετικά σημαντικό, ιδιαίτερα για μια δορυφορική σύνδεση ή μια σύνδεση πολύ υψηλής ταχύτητας.

Ο σύνδεσμος επικοινωνίας που εξετάζεται θεωρείται ως ένας αγωγός από bit πάνω από τον οποίο ένας δεδομένος αριθμός bit μπορούν να μεταδοθούν ανά δευτερόλεπτο. Αυτός ο αριθμός ονομάζεται transmission capacity χωρητικότητα μετάδοσης. Εξαρτάται τόσο από το φυσικό κανάλι όσο και από τη διεπαφή (π.χ. μόντεμ) και είναι απλά ο ρυθμός με τον οποίο η διεπαφή δέχεται bits. Η χωρητικότητα του συνδέσμου μπορεί να εξυπηρετεί πολλές ροές κυκλοφορίας (π.χ. εικονικά κυκλώματα ή ομάδες εικονικών κυκλωμάτων) που πολυπλέκονται στη σύνδεση. Ο τρόπος κατανομής της χωρητικότητας μεταξύ αυτών των ροών κυκλοφορίας έχει βαθιά επίδραση σε καθυστέρηση πακέτων. Στο πιο κοινό σχήμα, στη στατιστική πολυπλεξία statistical multiplexing, τα πακέτα από όλες τις ροές συγχωνεύονται σε μια ενιαία ουρά και μεταδίδονται με σειρά προτεραιότητας.

Μια παραλλαγή αυτού του σχήματος, που έχει περίπου την ίδια μέση καθυστέρηση ανά πακέτο, διατηρείται μια ξεχωριστή ουρά για κάθε ροή κυκλοφορίας και εξυπηρετεί τις ουρές με συχνότητα ένα πακέτο κάθε φορά. Ωστόσο, εάν η ουρά μιας ροής κυκλοφορίας είναι κενή, η επόμενη ροή εξυπηρετείται και δεν σπαταλάται κανένας πόρος επικοινωνίας. Δεδομένου ότι ολόκληρη η ικανότητα μετάδοσης C (bits/sec) εκχωρείται σε ένα μόνο πακέτο κάθε φορά, χρειάζονται L / C δευτερόλεπτα για να μεταδοθεί ένα πακέτο που έχει μήκος L bit.

Σε πολυπλεξία διαίρεσης χρόνου (TDM) και πολυπλεξία διαίρεσης συχνότητας (FDM) με m ροές, η χωρητικότητα ζεύξης υποδιαιρείται ουσιαστικά σε m τμήματα, ένα ανά ροή. Στο FDM, το εύρος ζώνης καναλιού W υποδιαιρείται σε m κανάλια που το καθένα έχει εύρος ζώνης W/m (στην πραγματικότητα ελαφρώς μικρότερο λόγω της ανάγκης για προστατευτικές ζώνες μεταξύ καναλιών). Η ικανότητα μετάδοσης κάθε καναλιού είναι περίπου C/m , όπου το C είναι η χωρητικότητα που θα αποκτούσε εάν ολόκληρο το εύρος ζώνης κατανεμόταν σε ένα μόνο κανάλι. Ο χρόνος μετάδοσης ενός πακέτου που έχει μήκος L bit είναι Lm/C ή m φορές μεγαλύτερο από το αντίστοιχο σχήμα στατιστικής πολυπλεξίας. Στο TDM, η κατανομή γίνεται με διαίρεση του άξονα χρόνου σε υποδοχές σταθερού μήκους (π.χ. μήκους ενός bit ή ενός byte, ή ίσως ένα πακέτο μήκους για πακέτα σταθερού μήκους). Και πάλι, εννοιολογικά, η ζεύξη επικοινωνίας αποτελείται από m ξεχωριστές ζεύξεις έχει χωρητικότητα C / m . Στην περίπτωση όπου οι υποδοχές είναι μικρές σε σχέση με το μήκος του πακέτου, μπορεί και πάλι να θεωρήσουμε τη μετάδοση χρόνου ενός πακέτου L bits ίσο με Lm/C . Στην

περίπτωση που οι υποδοχές έχουν ίσο μήκος πακέτου, ο χρόνος μετάδοσης ενός πακέτου L bit είναι L/C , αλλά υπάρχει αναμονή $(m - 1)$ πακέτου χρόνους μετάδοσης μεταξύ πακέτων της ίδιας ροής. Ένα από τα θέματα που έχουν προκύψει κατά την ανάλυσή στην ουρά είναι αυτό το στατιστικό. Η πολυπλεξία (multiplexing) έχει μικρότερη μέση καθυστέρηση ανά πακέτο από το TDM ή το FDM. Αυτό ιδιαίτερα ισχύει όταν οι πολυπλεξικές ροές κυκλοφορίας έχουν σχετικά χαμηλό κύκλο λειτουργίας. Ο κύριος λόγος για την κακή απόδοση καθυστέρησης του TDM και του FDM είναι όταν κατά την επικοινωνία οι πόροι σπαταλούνται όταν κατανέμονται σε μια ροή κίνησης με μια στιγμιαία άδεια ουρά, ενώ άλλα ρεύματα κυκλοφορίας έχουν πακέτα που περιμένουν στην ουρά τους. Για μια αναλογία κυκλοφορίας, ας φανταστούμε έναν αυτοκινητόδρομο με μια λωρίδα m και δύο περιπτώσεις. Σε μια περίπτωση δεν επιτρέπεται η διέλευση αυτοκινήτων σε άλλες λωρίδες (αυτό αντιστοιχεί σε TDM ή FDM), ενώ στην άλλη περίπτωση, τα αυτοκίνητα μπορούν να αλλάξουν λωρίδα (αυτό αντιστοιχεί περίπου σε στατιστική πολυπλεξία). Αυτός ο περιορισμός crossover αυξάνει τον χρόνο ταξιδιού για τον ίδιο λόγο που τα χαρακτηριστικά καθυστέρησης του TDM ή του FDM είναι συγκεκριμένα, ορισμένοι πόροι του συστήματος (λωρίδες αυτοκινητοδρόμων ή κανάλια επικοινωνίας) μπορεί να μην αξιοποιηθούν, ενώ άλλοι στιγμιαία συγχωνεύονται.

Στο πλαίσιο ενός δικτύου δεδομένων, οι πελάτες αντιπροσωπεύουν πακέτα που έχουν εκχωρηθεί σε μια επικοινωνία για μετάδοση. Ο χρόνος υπηρεσίας αντιστοιχεί στο χρόνο μετάδοσης του πακέτου και είναι ίσος με το L/C , όπου L είναι το μήκος του πακέτου σε bit και C είναι η χωρητικότητα μετάδοσης ζεύξης σε bit/sec. Επομένως, τα μήκη πακέτων λαμβάνονται για να περιλαμβάνουν κεφαλίδες πλαισίου και ρυμουλκούμενα. Σε ένα κάπως διαφορετικό πλαίσιο (που δεν θα τονίσουμε ιδιαίτερα), πελάτες αντιπροσωπεύουν συνεχείς συνομιλίες (ή εικονικά κυκλώματα) μεταξύ σημείων σε ένα δίκτυο και ο χρόνος υπηρεσίας αντιστοιχεί στη διάρκεια μιας συνομιλίας. Σε ένα σχετικό πλαίσιο, οι πελάτες αντιπροσωπεύουν ενεργές κλήσεις σε τηλεφωνικό ή δίκτυο μεταγωγής κυκλώματος και ξανά χρόνο εξυπηρέτησης αντιστοιχεί στη διάρκεια της κλήσης. Συνήθως θα μας ενδιαφέρει να εκτιμήσουμε ποσότητες όπως π.χ

1. Ο μέσος αριθμός πελατών στο σύστημα (δηλ. ο «τυπικός» αριθμός των πελάτες είτε περιμένουν στην ουρά είτε υποβάλλονται σε σέρβις).
2. Η μέση καθυστέρηση ανά πελάτη (δηλαδή, ο «τυπικός» χρόνος που αφιερώνει ένας πελάτης σε αναμονή στην ουρά συν το χρόνο εξυπηρέτησης).

Αυτές οι ποσότητες θα εκτιμηθούν με βάση γνωστές πληροφορίες όπως:

1. Το ποσοστό άφιξης πελατών (δηλαδή, ο «τυπικός» αριθμός πελατών που εισέρχονται στο σύστημα ανά μονάδα χρόνου).
2. Το ποσοστό εξυπηρέτησης πελατών (δηλαδή, ο «τυπικός» αριθμός πελατών που εξυπηρετεί

το σύστημα ανά μονάδα χρόνου όταν είναι συνεχώς απασχολημένο). Σε πολλές περιπτώσεις, οι τιμές άφιξης πελατών και εξυπηρέτησης δεν επαρκούν για τον προσδιορισμό τα χαρακτηριστικά καθυστέρησης του συστήματος. Για παράδειγμα, εάν οι πελάτες τείνουν να φτάνουν μέσα ομάδες, η μέση καθυστέρηση πελατών θα τείνει να είναι μεγαλύτερη από ό,τι κατά την άφιξή τους βρίσκονται σε τακτική απόσταση μεταξύ τους. Έτσι, για να προβλέψουμε τη μέση καθυστέρηση, θα χρειαστούμε συνήθως περισσότερα λεπτομερείς (στατιστικές) πληροφορίες σχετικά με τους χρόνους άφιξης και εξυπηρέτησης πελατών.[1].

Β'.2 Το θεώρημα του Little

. Προχωράμε στην αποσαφήνιση της σημασίας των όρων «μέσος όρος» και «τυπικός» που χρησιμοποιήσαμε κάπως ελεύθερα παραπάνω σε σχέση με τον αριθμό των πελατών στο σύστημα, το καθυστέρηση πελάτη και ούτω καθεξής. Κάνοντας αυτό θα βγάλουμε ένα σημαντικό αποτέλεσμα γνωστό ως Θεώρημα Little. Ας υποθέσουμε ότι παρατηρούμε ένα δείγμα ιστορικού του συστήματος από το χρόνο $t = 0$ έως το αόριστο μέλλον και καταγράφουμε τις αξίες διαφόρων ποσοτήτων ενδιαφέροντος ως χρόνο προοδεύει. Συγκεκριμένα, ας ορίσουμε:

$N(t)$ = Αριθμός πελατών στο σύστημα τη στιγμή t

aCt) = Αριθμός πελατών που έφτασαν στο διάστημα $[0, t]$

T = Χρόνος που δαπανάται στο σύστημα από το i^{th} πελάτη που φτάνει. Η διαισθητική μας αντίληψη για τον «τυπικό» αριθμό πελατών στο σύστημα που παρατηρήθηκε μέχρι ώρα t είναι:

$$\lambda_i = 0$$

$N(t) = \frac{1}{t} * \int_0^t N(t), dt$ που ονομάζουμε μέσο όρο χρόνου του $N(T)$ μέχρι τη στιγμή t . Φυσικά, το Nt αλλάζει με το χρόνο t , αλλά σε πολλά συστήματα ενδιαφέροντος, το Nt τείνει σε μια σταθερή κατάσταση N καθώς το t αυξάνεται, που είναι $N =$

$$\lim_{t \rightarrow \infty} N(t)$$

Σε αυτή την περίπτωση, ονομάζουμε N τον μέσο όρο χρόνου σταθερής κατάστασης (ή απλώς μέσο όρο χρόνου) του $N(T)$. Είναι επίσης φυσικό να το βλέπεις $\lambda_t = \frac{a(t)}{t}$ ως ο μέσος χρόνος άφιξης στο διάστημα $[0, t]$. Το ποσοστό άφιξης σε σταθερή κατάσταση είναι ορίζεται ως $\lambda = \lim_{t \rightarrow \infty} (\lambda_t)$ (υποθέτοντας ότι υπάρχει το όριο). Ο μέσος όρος χρόνου της καθυστέρησης του πελάτη μέχρι το χρόνο t είναι ορίζεται ομοίως ως $T_t = \frac{\sum_{i=1}^{a(t)} (T_i)}{a(t)}$ ο μέσος χρόνος που δαπανάται στο σύστημα ανά πελάτη μέχρι το χρόνο t . Η σταθερή κατάσταση H μέση χρονική

καθυστερήση πελάτη ορίζεται ως (υποθέτοντας ότι υπάρχει το όριο).

Αποδεικνύεται ότι οι ποσότητες N , A και T παραπάνω σχετίζονται με έναν απλό τύπο που καθιστά δυνατό τον προσδιορισμό του ενός δεδομένου του άλλου. Αυτό το αποτέλεσμα, γνωστό ως Little's θεώρημα, έχει τη μορφή

$$N = \lambda * T$$

Το Θεώρημα του Little εκφράζει τη φυσική ιδέα ότι συσχετίζονται συνωστισμένα συστήματα (μεγάλο N), με μεγάλες καθυστερήσεις πελατών (μεγάλο T) και αντίστροφα. Για παράδειγμα, μια βροχερή μέρα, κίνηση σε μια ώρα αιχμής κινείται πιο αργά από το μέσο όρο (μεγάλο T), ενώ οι δρόμοι είναι πιο γεμάτοι (μεγάλο N). Ομοίως, ένα fastfood (μικρό T) χρειάζεται μια μικρότερη αίθουσα αναμονής (μικρό N) από ένα κανονικό εστιατόριο για την ίδια τιμή άφιξης πελατών.[1]

Β'.3 Σύστημα Αναμονής M/M/1

Ως σύστημα αναμονής ορίζουμε μια συλλογή απλών συστημάτων αναμονής στην οποία κάθε "πελάτης" που αποχωρεί από το σύστημα, μετακινείται σε κάποιο άλλο (πιθανώς το ίδιο), ή αναχωρεί από το δίκτυο, αν αυτό επιτρέπεται. Το M/M/1 είναι το απλούστερο σύστημα αναμονής στο οποίο υπάρχει ένας εξυπηρετητής με εκθετικούς χρόνους εξυπηρέτησης μέσης τιμής $\frac{1}{\mu}$ και τροφοδοτείται από ένα σύστημα αφίξεων. Το διάστημα μεταξύ δύο διαδοχικών αφίξεων είναι εκθετικά κατανομημένο με μέση τιμή $\frac{1}{\lambda}$.

Έτσι,

μ είναι ο μέσος ρυθμός εξυπηρέτησης και

λ ο μέσος ρυθμός αφίξεων πελατών στο σύστημα.

Η κατάσταση n του συστήματος ισούται με το πλήθος πελατών στο σύστημα (ουρά + εξυπηρέτηση). Από την κατάσταση n πηγαίνουμε στην κατάσταση $n+1$ με μία άφιξη και στην κατάσταση $n-1$ με μία εξυπηρέτηση και άναχωρηση από το σύστημα.

Η πιθανότητα να συμβεί μία άφιξη στο διάστημα $[t, t+x]$ δοθέντος ότι θα συμβεί σε χρόνο $\geq t$ είναι

$1 - e^{-\lambda x}$. Όταν στο σύστημα δεν υπάρχουν πελάτες, δηλαδή $n=0$ τότε έχουμε την εξίσωση $\lambda P_0 = \mu P_1$.

Ένα σύστημα λέγεται σταθερό όταν ο ρυθμός εισόδων σε αυτό είναι μικρότερος από τον ρυθμό εξυπηρέτησης $\lambda < \mu$ και μας δίνεται από τη σχέση $\rho = \frac{\lambda}{\mu}$. Το μέσο πλήθος πελατών στο σύστημα είναι $N = \sum_{n=0}^{\infty} n P_n = (1 - \rho) \sum_{n=0}^{\infty} n \rho^n = \frac{\rho}{1-\rho}$

Σε κάθε σύστημα αναμονής ισχύει ο νόμος του Little

(μέσο πλήθος πελατών) = (μέσος ρυθμός εισόδου) * (μέσος χρόνος παραμονής στο σύστημα).

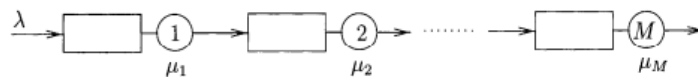
Για το M/M/1, όπου δεν υπάρχει περιορισμός στην ουρά, ο μέσος ρυθμός αφίξεων πελατών λ ισούται με το μέσο ρυθμό εισόδου δηλαδή, όσοι φτάνουν γίνονται πάντα δεκτοί από το σύστημα και κατά συνέπεια,

ο μέσος χρόνος παραμονής στο σύστημα (ουρά + εξυπηρέτηση) είναι $T = \frac{N}{\lambda}$,

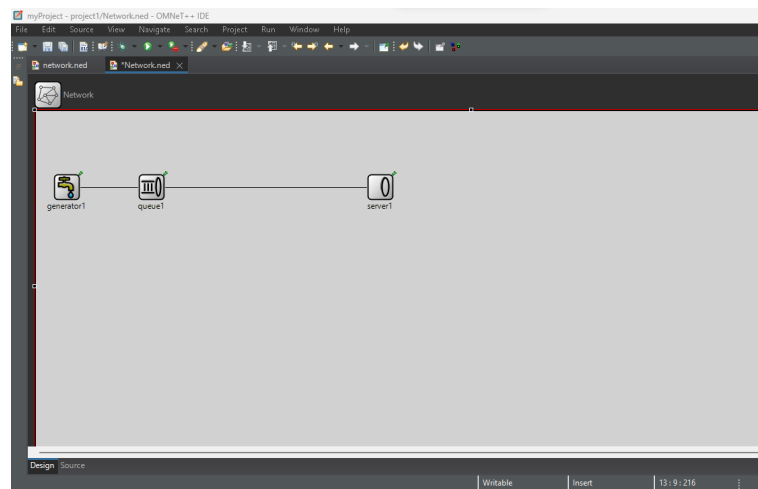
ο μέσος χρόνος αναμονής στην ουρά μόνον είναι $W = T - \frac{1}{\mu}$.

Εφαρμόζοντας το νόμο του Little για το σύστημα 'ουρά' προκύπτει το μέσο πλήθος πελατών στην ουρά [1] [2] [3]

$$N_q = \lambda W = \frac{\rho}{1-\rho}$$



Σχήμα Β'.1: Σειριακή σύνδεση M ουρών



Σχήμα Β'.2: Πειραματικό δίκτυο M/M/1

Κεφάλαιο Γ΄

Δίκτυα Jackson

Γ΄.1 Δίκτυα Jackson

Γενικά, ένα δίκτυο συστημάτων αναμονής, μπορεί να περιγραφεί από ένα σύνολο M σταθμών (απλά συστήματα αναμονής) όπου

1. Κάθε σταθμός αναπαριστά κάποια παροχή υπηρεσίας με τη βοήθεια c_i σερερς στο σταθμό i , $i = 1, 2, 3, \dots, M$.
2. Γενικά, οι πελάτες μπορούν να αφιχθούν από το περιβάλλον στο δίκτυο σε οποιοδήποτε σταθμό και να εγκαταλείψουν το σύστημα από οποιοδήποτε σταθμό.
3. Μπορούν να διασχίσουν το σύστημα από διαφορετικά μονοπάτια κάθε φορά. Και
4. Μπορούν να επιστρέψουν σε σταθμούς που έχουν προηγουμένως επισκεφτεί, να παραλείψουν εντελώς κάποιους από αυτούς, ή ακόμη να επιλέξουν να παραμείνουν στο σύστημα για πάντα.

Η παραπάνω περιγραφή δεν άλλο από ένα δίκτυο Jackson δηλαδή μια συλλογή απλών συστημάτων αναμονής, στην οποία κάθε «πελάτης» που αναχωρεί από ένα απλό σύστημα, μετακινείται σε κάποιο άλλο (πιθανώς το ίδιο), ή αναχωρεί από το δίκτυο, αν αυτό επιτρέπεται, όπως και ένα σύστημα αναμονής $M/M/1$.

Διακρίνονται σε δύο κατηγορίες στο

Ανοιχτό δίκτυο συστημάτων αναμονής και είναι αυτό στο οποίο επιτρέπονται αφίξεις από το περιβάλλον του δικτύου και αναχωρήσεις προς το περιβάλλον, και στο

Κλειστό δίκτυο που είναι αυτό στο οποίο δεν επιτρέπονται αφίξεις και αναχωρήσεις, αλλά διατηρείται ένας σταθερός αριθμός N μετακινούμενων «πελατών». Στην παρούσα εργασία θα

ασχοληθούμε με του ανοιχτού τύπου δίκτυο συστημάτων αναμονής. [4]

Γ'.2 Βασικά Χαρακτηριστικά ενός δικτύου Jackson

Είτε πρόκειται για Ανοιχτό είτε για Κλειστό δίκτυο, έχουν τα βασικά χαρακτηριστικά ενός δικτύου ή συστήματος αναμονής δηλαδή,

Οι αφίξεις από το περιβάλλον στο σταθμό i (ουρές μήκους ∞) ακολουθούν μία διαδικασία Poisson με μέσο ρυθμό λ_i .

Οι χρόνοι εξυπηρέτησης είναι ίδιοι στους servers κάθε σταθμού, ανεξάρτητοι και εκθετικά κατανομημένοι με παράμετρο μ_i (θεωρούμε 1 server ανά σταθμό αλλά εύκολα > 1 όπως στην περίπτωση των δικτύων Jackson).

Η πιθανότητα ένας πελάτης που τελείωσε την εξυπηρέτηση του στον σταθμό i , να πάει στο σταθμό j (routing probability), είναι ρ_{ij} (ανεξάρτητη από την κατάσταση του συστήματος) με $i = 1, 2, \dots, M$, $j = 1, 2, \dots, M$.

Η πιθανότητα ρ_{i0} υποδεικνύει την πιθανότητα ο πελάτης να αναχωρήσει από το δίκτυο, από το σταθμό i .

Ιδιαίτερα, για τα Κλειστά συστήματα Jackson ισχύει ότι

$\lambda_i = 0$ για όλα τα i (κανένας πελάτης δεν μπορεί να εισέλθει στο δίκτυο) και

$\rho_{i0} = 0$ για όλα τα i (κανένας πελάτης δεν μπορεί να αναχωρήσει από το δίκτυο).

Το δίκτυο συμπεριφέρεται σαν κάθε σταθμός του να είναι ένα ανεξάρτητο $M/M/1$ με παραμέτρους λ_i και μ_i

Η συνολική κατανομή πιθανότητας μπορεί να γραφεί σαν γινόμενο των λύσεων των επί μέρους $M/M/1$

Στην πραγματικότητα δεν είναι $M/M/1$ τα επί μέρους συστήματα λόγω των πιθανών αναδράσεων στη δρομολόγηση, ωστόσο συμπεριφέρονται ακριβώς όπως τα $M/M/1$. [4]

Γ'.3 Δίκτυα Τηλεπικοινωνιών

Ένα δίκτυο τηλεπικοινωνιών είναι ένα σύστημα επικοινωνιών το οποίο διαθέτει: α) πολύπλοκες τηλεπικοινωνιακές συσκευές που χρησιμοποιούνται από κοινού με όλους τους συνδρομητές του δικτύου, β) τηλεπικοινωνιακούς κόμβους και γ) τα φυσικά μέσα διάδοσης της πληροφορίας (γραμμές επικοινωνίας). Επίσης περιλαμβάνει και τις διατάξεις πρόσβασης στο δίκτυο (τη-

λέφωνα, υπολογιστές κλπ.). Βασική ιδιότητα του κάθε δικτύου είναι η παροχή ικανοποιητικής επικοινωνίας με τον ελάχιστο δυνατό αριθμό διασυνδέσεων των κόμβων του. Με το δίκτυο, κάθε συνδρομητής μπορεί να χρησιμοποιεί από κοινού με άλλους συνδρομητές διάφορες τηλεπικοινωνιακές συσκευές που δεν μπορεί να διαθέσει μόνος του. Με βάση τις τεχνικές που χρησιμοποιούνται για τη μετάδοση των δεδομένων σε ένα δίκτυο, τα δίκτυα διακρίνονται σε δύο μεγάλες κατηγορίες: τα **Δίκτυα μεταγωγής** και τα **Δίκτυα πολλαπλής πρόσβασης στο κανάλι διάδοσης**. Ειδικότερα,

Στα δίκτυα μεταγωγής, τα δεδομένα που εισέρχονται στο δίκτυο από κάποια πηγή πληροφορίας (τερματική διάταξη), μεταφέρονται μέσω ενδιάμεσων κόμβων στον προκαθορισμένο δέκτη. Οι κόμβοι διακινούν τα δεδομένα προς τον προορισμό τους αποφασίζοντας ή όχι για την αποτελεσματική διακίνησή τους. Για την αύξηση της αξιοπιστίας του δικτύου, οι κόμβοι συνδέονται με τέτοιο τρόπο ώστε να υπάρχει εναλλακτικός δρόμος μεταξύ των τερματικών σημείων. Οι τρεις τεχνικές που χρησιμοποιούνται για τη μετάδοση δεδομένων στα δίκτυα μεταγωγής είναι οι εξής:

Μεταγωγή κυκλώματος (Circuit switching)

Μεταγωγή πακέτων (Packet switching)

Μεταγωγή μηνύματος (Message switching)

Στη μεταγωγή κυκλώματος ένα φυσικό κανάλι προσφέρεται στους συνδρομητές αποκλειστικά σε όλη τη διάρκεια της επικοινωνίας τους και καταργείται μόνο με τον τερματισμό της επικοινωνίας αυτής. Χαρακτηριστικό παράδειγμα της τεχνικής μεταγωγής κυκλώματος αποτελεί το τηλεφωνικό δίκτυο. Στη μεταγωγή πακέτων τα δεδομένα που πρόκειται να μεταφερθούν τεμαχίζονται σε πακέτα ομοίου μήκους. Στην τεχνική αυτή δεν υπάρχει εκ των προτέρων σχηματιζόμενο φυσικό κανάλι για τη συγκεκριμένη επικοινωνία των δύο συνδρομητών. Οι ενδιάμεσοι κόμβοι του δικτύου αποφασίζουν για τη διαδρομή που θα διανύσει το κάθε πακέτο ώστε να φτάσει στον προορισμό του με τον πιο αποτελεσματικό τρόπο. Συνεπώς οι κόμβοι του δικτύου θα πρέπει να έχουν επεξεργαστική ικανότητα για την προώθηση των πακέτων. Δύο διαφορετικές μέθοδοι χρησιμοποιούνται για την προώθηση των πακέτων: α) τα αυτοδύναμα πακέτα (datagram) και β) τα εικονικά κυκλώματα (virtual circuits). Με τη μέθοδο datagram, κάθε πακέτο αντιμετωπίζεται από τους κόμβους του δικτύου σαν ένα ολοκληρωμένο μήνυμα. Κάθε κόμβος που παραλαμβάνει το πακέτο επιλέγει ποιος θα είναι ο επόμενος έτσι ώστε το δίκτυο να λειτουργεί με τον καλύτερο δυνατό τρόπο. Για τον λόγο αυτόν οι κόμβοι οφείλουν να διαθέτουν αρκετές πληροφορίες για τη δομή και την κατάσταση του δικτύου κάθε χρονική στιγμή. Τα πακέτα πληροφορίας ενώ έχουν τον ίδιο προορισμό δεν ακολουθούν όλα τον ίδιο δρόμο γι' αυτό υπάρχει πιθανότητα να φτάσουν με διαφορετική σειρά από αυτήν που

στάλθηκαν. Έτσι θα πρέπει να υπάρχει κατάλληλη διάταξη που να τα τοποθετεί στην αρχική τους σειρά. Στη μέθοδο εικονικού κυκλώματος (virtual circuit), πριν αρχίσει η αποστολή των πακέτων αποκαθίσταται μία σταθερή νοητή σύνδεση μεταξύ των δύο συνδρομητών από όπου στη συνέχεια θα περάσουν όλα τα πακέτα του μηνύματος. Δηλαδή ο δρόμος που θα ακολουθήσουν τα πακέτα καθορίζεται μια φορά στην αρχή και παραμένει ο ίδιος μέχρι να διακοπεί η επικοινωνία των δύο συνδρομητών. Η εξασφάλιση της ύπαρξης ελεύθερου δρόμου γίνεται με την ανταλλαγή μηνυμάτων μεταξύ των δύο συνδρομητών που πρόκειται να επικοινωνήσουν. Στην τεχνική αυτή οι ενδιαμέσοι κόμβοι δεν απαιτείται να έχουν πληροφορίες για την κατάσταση του δικτύου, γιατί δεν αποφασίζουν για τη δρομολόγηση των μηνυμάτων αλλά απλά τα διακινούν στον προορισμό τους.

Στη μεταγωγή μηνύματος τα δεδομένα αποστέλλονται με τη μορφή μηνύματος που μεταδίδεται ολόκληρο ανεξάρτητα από το μέγεθός του. Το δίκτυο προωθεί το μήνυμα από κόμβο σε κόμβο μέχρι να φτάσει στον προορισμό του. Κάθε κόμβος αποφασίζει ποιος θα είναι ο επόμενος που θα παραλάβει το μήνυμα, δηλαδή οι κόμβοι αποφασίζουν για τη δρομολόγηση του μηνύματος. Για τον λόγο αυτόν θα πρέπει στους κόμβους να υπάρχει αρκετή πληροφορία για τη δομή και την κατάσταση του δικτύου κάθε χρονική στιγμή. Παρά τα πλεονεκτήματα της τεχνικής αυτής, στην πράξη η μεταγωγή μηνύματος έχει αντικατασταθεί από τη μεταγωγή πακέτων.

Τα δίκτυα πολλαπλής πρόσβασης στο κανάλι διάδοσης έχουν εντελώς διαφορετικά χαρακτηριστικά από τα δίκτυα μεταγωγής που μελετήσαμε παραπάνω. Τα κύρια χαρακτηριστικά αυτών των δικτύων είναι τα εξής:

1. Δεν υπάρχουν ενδιαμέσοι κόμβοι μεταγωγής μεταξύ πομπού και δέκτη.
2. Το μέσο επικοινωνίας είναι κοινό για όλους τους συνδρομητές.
3. Σε μια συγκεκριμένη περιοχή του μέσου επικοινωνίας μόνο ένας συνδρομητής μπορεί κάθε στιγμή να εκπέμπει.
4. Το εκπεμπόμενο σήμα μπορεί να λαμβάνεται από όλους τους συνδρομητές αρκεί να έχουν την κατάλληλη συσκευή πρόσβασης.
5. Μπορεί να υπάρχουν πολλοί πομποί που μοιράζονται χρονικά το ίδιο μέσο μετάδοσης.

Οι πιο γνωστοί τύποι τέτοιων δικτύων είναι:

Τα επίγεια ραδιοδίκτυα.

Τα δορυφορικά δίκτυα.

Τα τοπικά δίκτυα (LAN).

Οι δύο πρώτοι τύποι δικτύων αποτελούν αντικείμενο των ασυρμάτων τηλεπικοινωνιών. Στα δίκτυα αυτά κάθε σταθμός πρέπει να διαθέτει κεραιές εκπομπής και λήψεως και να βρίσκεται μέσα στην εμβέλεια των υπολοίπων. Τα τοπικά δίκτυα LAN είναι εντελώς διαφορετικά από

τους άλλους τύπους δικτύων αλλά και εδώ το βασικό χαρακτηριστικό είναι η ύπαρξη κοινού μέσου επικοινωνίας για όλους τους συνδρομητές, το οποίο είναι ένα καλώδιο ή μία οπτική ίνα. Ανάλογα με τη γεωγραφική έκταση που καλύπτουν, τα δίκτυα χωρίζονται σε αστικά δίκτυα (etropolitan Area Networks-MAN), σε τοπικά δίκτυα (Local Area Networks – LAN) και σε δίκτυα ευρείας περιοχής (Wide Area Networks – WAN).

Ως αστικά χαρακτηρίζονται εκείνα τα δίκτυα των οποίων το μέγεθος δεν ξεπερνάει τα όρια μιας πόλης. Τοπικά ονομάζονται εκείνα τα δίκτυα όπου όλα τα στοιχεία που τα απαρτίζουν βρίσκονται στον ίδιο γεωγραφικό χώρο. Τα δίκτυα ευρείας περιοχής καλύπτουν μια μεγάλη γεωγραφική περιοχή ή ολόκληρη χώρα και διασυνδεδεμένα αποτελούν διεθνή δίκτυα καλύπτοντας ηπείρους. Η διαφοροποίηση των δικτύων στο μέγεθος δημιουργεί εντελώς διαφορετικά προβλήματα σχεδιασμού και λειτουργίας και επιβάλλει διαφορετικές τεχνικές επίλυσής τους. Για παράδειγμα, στα δίκτυα ευρείας περιοχής, το ζήτημα του τοπολογικού σχεδιασμού είναι πολύ σύνθετο και δύσκολα επιλύεται, ενώ στα τοπικά δίκτυα οι επιλογές τοπολογίας έχουν καθοριστεί. Επίσης, στα εθνικά δίκτυα ένα σημαντικό ζήτημα είναι το πρόβλημα διακίνησης μηνυμάτων, ενώ αντίστοιχο πρόβλημα στα τοπικά δίκτυα μπορεί να θεωρηθεί το πρόβλημα πρόσβασης στο κοινό κανάλι των συνδρομητών. Ωστόσο, παρά τη γεωγραφική διαίρεση των δικτύων η ανάγκη διασύνδεσης των δικτύων μεταξύ τους έχει σαν αποτέλεσμα πολλά χαρακτηριστικά των δικτύων να ομογενοποιηθούν και πολλές διαφορές μεταξύ τοπικών δικτύων και δικτύων ευρείας περιοχής να εξαλειφθούν.

κτός από τις τερματικές συσκευές και τους συνδρομητικούς βρόχους, όλες οι υπόλοιπες διατάξεις του τηλεπικοινωνιακού δικτύου χρησιμοποιούνται από κοινού από τους συνδρομητές για να μειωθεί το κόστος των παρεχόμενων υπηρεσιών. Αν κάθε συνδρομητής έπρεπε να διαθέτει τον δικό του τηλεπικοινωνιακό εξοπλισμό για να επικοινωνεί με άλλους συνδρομητές, η χρήση των δυνατοτήτων των τηλεπικοινωνιών θα ήταν αντιοικονομική από το μεγαλύτερο ποσοστό του πληθυσμού. Η κοινή χρήση των διατάξεων του τηλεπικοινωνιακού δικτύου δεν είναι τυχαία αλλά απορρέει από το γεγονός ότι ποτέ οι συνδρομητές δεν ζητούν εξυπηρέτηση από το δίκτυο όλοι ταυτόχρονα. Δεχόμαστε δηλαδή ότι ο τρόπος άφιξης των κλήσεων στο τηλεπικοινωνιακό κέντρο και η χρονική διάρκεια της κάθε κλήσης είναι μία τυχαία διαδικασία. Επομένως, για τη σωστή λειτουργία του τηλεφωνικού δικτύου είναι απαραίτητη η γνώση της τηλεπικοινωνιακής κίνησης, δηλαδή του αριθμού των κλήσεων που καλείται να διεκπεραιώσει το δίκτυο κατά τη διάρκεια της ημέρας και πιο συγκεκριμένα κατά την ώρα αιχμής, όπου και παρατηρείται η μεγαλύτερη τηλεπικοινωνιακή κίνηση. Συνεπώς, για την καλή εξυπηρέτηση των συνδρομητών καθώς και για τον οικονομικό σχεδιασμό του δικτύου, θα πρέπει να μελετηθεί η τηλεπικοινωνιακή κίνηση και να προσδιοριστούν τα βασικά χαρακτηριστικά και οι παράγοντες

που την καθορίζουν. Για το σκοπό αυτόν η τηλεπικοινωνιακή κίνηση πρέπει να καταγράφεται σε καθημερινή βάση. Με τη βοήθεια των μετρήσεων αυτών είναι δυνατή η πρόβλεψη της τηλεπικοινωνιακής κίνησης που καλείται να εξυπηρετήσει το τηλεπικοινωνιακό σύστημα.

Η λήψη, επεξεργασία και αξιοποίηση των μετρήσεων της τηλεπικοινωνιακής κίνησης, αποτελούν για κάθε τηλεπικοινωνιακό οργανισμό, χρήσιμο εργαλείο που χρησιμοποιείται για την εξαγωγή κάποιων συμπερασμάτων που αφορούν τον προσδιορισμό της δομής των κέντρων, την παρακολούθηση της παρεχόμενης ποιότητας επικοινωνίας κλπ. Λαμβάνοντας υπόψη τις μετρήσεις της τηλεπικοινωνιακής κίνησης, το τηλεπικοινωνιακό σύστημα σχεδιάζεται έτσι ώστε να μπορεί να αντιμετωπίσει την κίνηση κατά την ώρα αιχμής, δηλαδή την ώρα που παρατηρείται η μεγαλύτερη τηλεπικοινωνιακή κίνηση. Εφ' όσον το τηλεπικοινωνιακό δίκτυο σχεδιάζεται με αυτόν τον τρόπο, αυτό σημαίνει ότι τις περισσότερες από τις υπόλοιπες ώρες το σύστημα υπολειτουργεί. Με βάση τα παραπάνω, καταλαβαίνουμε τον λόγο για τον οποίον οι τηλεπικοινωνιακοί οργανισμοί προσφέρουν φθηνές τιμές κλήσεων κάποιες συγκεκριμένες ώρες της ημέρας.

Αυτό που συμβαίνει συχνά σε ένα τηλεπικοινωνιακό δίκτυο είναι, όλες οι γραμμές που καλούνται να εξυπηρετήσουν την τηλεπικοινωνιακή κίνηση να είναι κατειλημμένες και επομένως το σύστημα να μη μπορεί να δεχθεί άλλες κλήσεις. Η κατάσταση αυτή ονομάζεται συμφόρηση. Στην κατάσταση συμφόρησης, πολύ συχνά χάνονται κλήσεις και γι' αυτό η πραγματική κίνηση (κλήσεις που πραγματοποιούνται) είναι μικρότερη από την προσφερόμενη κίνηση (συνολικός αριθμός κλήσεων που παρέχεται στο δίκτυο) κατά το ποσό της κίνησης που χάνεται, δηλαδή τις απώλειες. Επομένως ισχύει:

$$\text{Διεκπεραιούμενη κίνηση} = \text{Προσφερόμενη κίνηση} - \text{Απώλειες}$$

Στην κατάσταση της συμφόρησης μπορούμε να έχουμε τις εξής δύο περιπτώσεις:

1. Η εισερχόμενη κλήση μπλοκάρει και εγκαταλείπει το σύστημα, οπότε το σύστημα καλείται σύστημα απωλειών. Στα συστήματα απωλειών (όπως π.χ. στα συμβατικά τηλεφωνικά κέντρα), κριτήριο αξιολόγησης της απόδοσης του συστήματος είναι η πιθανότητα απώλειας κλήσης ή πιθανότητα μπλοκαρίσματος. Η σχεδίαση του συστήματος στην περίπτωση αυτή γίνεται με τέτοιον τρόπο, έτσι ώστε ακόμα και στις ώρες που παρουσιάζεται η μέγιστη τηλεπικοινωνιακή κίνηση (ώρες αιχμής) μόνο ένα μικρό προκαθορισμένο ποσοστό των συνδέσεων να μην μπορεί να αποκατασταθεί.
2. Η κλήση μπορεί να περιμένει για να γίνει η σύνδεση οπότε το σύστημα ονομάζεται σύστημα αναμονής. Στα συστήματα αναμονής (π.χ. δίκτυα υπολογιστικών συστημάτων ή ψηφιακά τηλεφωνικά κέντρα), κριτήριο αξιολόγησης αποτελεί ο μέσος χρόνος αναμονής ή η πιθανότητα αναμονής. Σε αυτή την περίπτωση το τηλεπικοινωνιακό σύστημα κατασχευάζεται έτσι ώστε

τις ώρες της μέγιστης τηλεπικοινωνιακής κίνησης, η αναμονή να είναι ανεκτή.

Η παραπάνω περιγραφή της τηλεπικοινωνιακής κίνησης είναι ακριβώς ίδια με την κυκλοφοριακή κίνηση που είδαμε στο κεφάλαιο Β' Θεωρία Ουρών.

Τα βασικά μεγέθη τηλεπικοινωνιακής κίνησης είναι: α) **Βαθμός εξυπηρέτησης**: Το ποσοστό των κλήσεων που χάνονται ή καθυστερούν να διεκπεραιωθούν λόγω συμφόρησης είναι ένας δείκτης της ποιότητας εξυπηρέτησης που παρέχεται από το τηλεπικοινωνιακό σύστημα. Το ποσοστό των χαμένων κλήσεων ονομάζεται βαθμός εξυπηρέτησης, B . Για τα συστήματα απωλειών, ο βαθμός εξυπηρέτησης B , ορίζεται ως εξής: $B = (\text{αριθμός κλήσεων που χάνονται}) / (\text{συνολικό αριθμό κλήσεων που προσφέρονται στο σύστημα})$, ή

$B = (\text{κίνηση που χάνθηκε}) / \text{κίνηση που προσφέρθηκε}$, ή

$B = \text{Πιθανότητα συμφόρησης}$, ή

$B = \text{Πιθανότητα ότι μία κλήση θα χαθεί λόγω συμφόρησης}$ Ο βαθμός εξυπηρέτησης καθορίζεται για την κίνηση κατά την ώρα αιχμής. Φυσικά, κατά τις ώρες μη αιχμής οι κλήσεις εξυπηρετούνται με τρόπο καλύτερο από αυτόν που δείχνει ο βαθμός εξυπηρέτησης. Αν γνωρίζουμε τον βαθμό εξυπηρέτησης B και το φορτίο τηλεπικοινωνιακής κίνησης a , τότε η κίνηση που θα χαθεί είναι $a * B$ και η κίνηση που θα διεκπεραιωθεί είναι $a * (1 - B)$.

Η επιλογή ενός συγκεκριμένου βαθμού εξυπηρέτησης είναι θέμα που πρέπει να προσεχθεί ιδιαίτερα, για τους εξής λόγους:

α) Αν ο βαθμός εξυπηρέτησης τεθεί μεγάλος (π.χ. 25%), οι συνδρομητές θα παραπονούνται ότι έχουν πολλές ανεπιτυχείς κλήσεις.

β) Αντίθετα, αν τεθεί πολύ μικρός, τότε το τηλεπικοινωνιακό σύστημα υπολειτουργεί τις περισσότερες ώρες, που σημαίνει ότι έχουμε κάνει σπατάλη επενδύσεων και υπάρχει πλεονάζων εξοπλισμός. Για τους παραπάνω λόγους είναι σκόπιμο ο βαθμός εξυπηρέτησης να μην είναι ενιαίος για ολόκληρο το τηλεπικοινωνιακό σύστημα αλλά να μεταβάλλεται, ανάλογα με τις απαιτήσεις του συγκεκριμένου τμήματος του συστήματος. Οι μαθηματικές σχέσεις της τηλεπικοινωνιακής κίνησης έχουν ως σκοπό τον υπολογισμό των γραμμών N που πρέπει να χρησιμοποιηθούν, δεδομένης της προσφερόμενης κινήσεως a και ενός επιθυμητού βαθμού εξυπηρέτησης B .

β) **Φορτίο κίνησης**: Το φορτίο κίνησης a , ορίζεται ως η συνολική διάρκεια όλων των κλήσεων εντός ενός χρονικού διαστήματος που λαμβάνεται ως μονάδα. Από αυτόν τον ορισμό καταλαβαίνουμε ότι το φορτίο κίνησης είναι ένα μέγεθος χωρίς διαστάσεις. Παρ' όλα αυτά το φορτίο κίνησης μετράται σε Erlang ή σε CCS ως εξής: Φορτίο κίνησης a (σε Erlang) = (συνολική διάρκεια όλων των κλήσεων σε sec) / (3600 sec).

Φορτίο κίνησης a (σε CCS) = (συνολική διάρκεια όλων των κλήσεων σε sec) / (100 sec)

Από τις παραπάνω σχέσεις προκύπτει ότι η αναλογία rlang και CCS είναι: 1 Erlang = 36 CCS.

Το φορτίο κίνησης αναφέρεται μερικές φορές και ως ένταση κίνησης και έχει τις ακόλουθες ιδιότητες:

1. Αν c είναι ο αριθμός των κλήσεων που φθάνουν σε ένα τηλεπικοινωνιακό σύστημα και h η μέση διάρκειά τους, τότε το φορτίο κίνησης a δίνεται από τη σχέση:

$$a = c * h(\text{Erlang})$$

2. Το φορτίο κίνησης ισούται προς τον αριθμό των κλήσεων που φθάνουν σε ένα τηλεπικοινωνιακό σύστημα εντός χρονικού διαστήματος ίσον προς τη μέση τιμή της διάρκειάς των. 3. Το φορτίο κίνησης που διεκπεραιώνεται από μία γραμμή μόνο, είναι ισοδύναμο με την πιθανότητα ότι η γραμμή χρησιμοποιείται (ποσοστό του χρόνου που η γραμμή είναι κατειλημμένη). Επομένως, μία γραμμή δεν μπορεί να μεταφέρει παρά μόνον 1 Erlang το πολύ (αφού η μέγιστη τιμή πιθανότητας είναι 1). 4. Το φορτίο κίνησης που διεκπεραιώνεται από μία δέσμη γραμμών είναι ισοδύναμο με τον μέσο αριθμό κατειλημμένων γραμμών της δέσμης. Οι βασικότερες διαδικασίες που χαρακτηρίζουν οποιοδήποτε τηλεπικοινωνιακό σύστημα είναι:

α) Η διαδικασία άφιξης πελατών στο σύστημα δηλαδή η άφιξη πακέτων δεδομένων ή τηλεφωνικών κλήσεων.

β) Η διαδικασία εξυπηρέτησης των πελατών, δηλαδή η διάρκεια των τηλεφωνικών συνδιαλέξεων ή γενικότερα η διάρκεια επικοινωνίας.

Η κατανομή που χρησιμοποιείται περισσότερο από όλες για τη περιγραφή τυχαίων αφίξεων είναι η κατανομή Poisson. Για να χαρακτηριστεί μία διαδικασία άφιξης κλήσεων ως τυχαία θα πρέπει να ικανοποιούνται οι εξής συνθήκες: Αν θεωρήσουμε ένα πολύ μικρό χρονικό διάστημα ($\Delta t \rightarrow 0$) τότε:

1. Η πιθανότητα άφιξης μιας κλήσεως στο χρονικό διάστημα Δt είναι ίση με $\lambda * \Delta t$, όπου λ είναι ένας σταθερός αριθμός.

2. Η πιθανότητα άφιξης δύο ή και περισσότερων κλήσεων στο ίδιο χρονικό διάστημα Δt τείνει στο μηδέν.

3. Οι κλήσεις που φθάνουν στο σύστημα είναι ανεξάρτητες μεταξύ τους.

Όταν πληρούνται οι παραπάνω συνθήκες, τότε δεχόμαστε ότι οι αφίξεις των κλήσεων ακολουθούν κατανομή Poisson.

Με βάση τα παραπάνω, η πιθανότητα k κλήσεις να φθάσουν στο σύστημα μέσα σε κάποιο χρονικό διάστημα $[0, t]$ δίνεται από τη σχέση:

$$P_k(t) = \frac{(\lambda * t)^k}{k!} * e^{-\lambda * t}, \text{ όπου } k = 0, 1, 2, \dots$$

Η παραπάνω σχέση μας δίνει την κατανομή Poisson, όπου το λ ονομάζεται ρυθμός άφιξης των

κλήσεων και το γινόμενο $\lambda * t$ μέση τιμή της κατανομής Poisson.

Από την παραπάνω σχέση προκύπτει ότι η πιθανότητα μηδέν κλήσεις να φθάσουν στο σύστημα στο διάστημα $[0, t]$ είναι: $P_0(t) = e^{-\lambda * t}$.

Αντίθετα, η πιθανότητα να πραγματοποιηθεί άφιξη στο χρονικό διάστημα $[0, t]$ δίνεται από τη σχέση:

$$A(t) = 1 - e^{-\lambda * t}$$

Η παραπάνω σχέση μας δίνει την πιθανότητα ότι ο χρόνος μεταξύ δύο διαδοχικών αφίξεων δεν θα υπερβεί την τιμή t .

Για να περιγράψουμε τα φαινόμενα εξυπηρέτησης των κλήσεων χρησιμοποιούμε την εκθετική κατανομή. Σύμφωνα με αυτήν η πιθανότητα μία κλήση να μην τερματιστεί σε διάστημα $[0, t]$, δηλαδή η πιθανότητα ο χρόνος εξυπηρέτησης της κλήσεως (έστω x) να είναι μεγαλύτερος από t , δίνεται από τη σχέση: $P(x > t) = e^{-\mu}$

όπου η ποσότητα μ ονομάζεται ρυθμός εξυπηρέτησης, ενώ το μέγεθος $1/\mu$ ονομάζεται μέση τιμή της εκθετικής κατανομής.

Ιδιότητες τηλεπικοινωνιακής κίνησης:

- Νόμος του Little

Ο νόμος του Little ορίζεται ως εξής: Αν σε ένα σύστημα αναμονής, λ είναι ο ρυθμός αφίξεων των κλήσεων, L είναι ο αριθμός των κλήσεων που αναμένουν στην ουρά για να εξυπηρετηθούν και W είναι ο μέσος χρόνος αναμονής στην ουρά, τότε ισχύει:

$$L = \lambda * W$$

Επέκταση του θεωρήματος του Little αποτελεί η παρακάτω σχέση:

$$N = \lambda * T$$

Στην παραπάνω σχέση N είναι ο μέσος αριθμός κλήσεων στο σύστημα (άθροισμα κλήσεων που αναμένουν και κλήσεων που εξυπηρετούνται), λ είναι ο ρυθμός άφιξης των κλήσεων και T είναι η μέση τιμή του συνολικού χρόνου που πρέπει να παραμείνει μια κλήση στο σύστημα, δηλαδή είναι το άθροισμα του χρόνου παραμονής της κλήσης στην ουρά και του χρόνου εξυπηρέτησής της.

- Ιδιότητες κατανομής Poisson:

1. Αν έχουμε M ανεξάρτητες κατανομές Poisson $l_1, l_2, l_3, \dots, l_M$ τότε η κατανομή που προκύπτει από το άθροισμά τους είναι επίσης κατανομή Poisson με μέση τιμή:

$$\lambda = \sum_{i=1}^M \lambda_i$$

2. Το άθροισμα ενός πολύ μεγάλου αριθμού ανεξάρτητων διαδικασιών του οποίου η μέση τιμή τείνει στο άθροισμα των μέσων τιμών των διαδικασιών δίνει κατανομή Poisson. - Μαρκοβιανή ιδιότητα:

Η μαρκοβιανή ιδιότητα είναι μία ιδιότητα που ισχύει για φαινόμενα τα οποία ακολουθούν εκθετική κατανομή (π.χ. η διαδικασία εξυπηρέτησης των κλήσεων). Όπως αναφέρθηκε και παραπάνω σύμφωνα με την εκθετική κατανομή, η πιθανότητα ο χρόνος εξυπηρέτησης της κλήσεως (έστω Q) να είναι μεγαλύτερος από x , δηλαδή η πιθανότητα το φαινόμενο να συνεχίζεται μετά τη χρονική στιγμή x δίνεται από τη σχέση: $P(X > x) = e^{-\mu}$

Με βάση την παραπάνω σχέση, η πιθανότητα ότι το φαινόμενο συνεχίζεται μετά από χρονική περίοδο t δεδομένου ότι έχει διαρκέσει μέχρι τη χρονική στιγμή x , προκύπτει ως εξής: $P(X > x + t | X > x) = \frac{P(X > x+t)}{P(X > x)} = \frac{e^{-\mu(x+t)}}{e^{-\mu x}} = e^{-\mu t} = P(X > t)$ Δηλαδή η πιθανότητα το φαινόμενο να συνεχιστεί μετά από χρονική περίοδο t δεδομένου ότι έχει διαρκέσει μέχρι τη χρονική στιγμή x είναι ανεξάρτητη από το x . Η ιδιότητα αυτή ονομάζεται μαρκοβιανή ιδιότητα και ισχύει για εκθετικές κατανομές.

Διάκριση μοντέλων τηλεπικοινωνιακής κίνησης: Για την ταξινόμηση των μοντέλων τηλεπικοινωνιακής κίνησης χρησιμοποιείται ο συμβολισμός του Kendall: A/B/s. Το A δηλώνει την κατανομή άφιξης των κλήσεων, το B δηλώνει την κατανομή εξυπηρέτησης των κλήσεων και το s τον αριθμό των εξερχόμενων γραμμών και λαμβάνει κάποια ακέραια τιμή. Το A και το B λαμβάνουν σαν τιμές τα αρχικά των κατανομών που χρησιμοποιούνται για την άφιξη ή την εξυπηρέτηση. Έτσι μπορούν να εκφράζονται από τους χαρακτήρες: M (Μαρκοβιανή) ή G (Γενική) ή D (ντετερμινιστική) κατανομή. Για παράδειγμα, ένα σύστημα με Poisson κατανομή άφιξης των κλήσεων και εκθετική κατανομή του χρόνου εξυπηρέτησης δηλώνεται ως //s. Με n πεπερασμένες γραμμές εισόδου δηλώνεται ως M(n)//s ενώ με ουρά m θέσεων δηλώνεται ως M/M/s(m) ή ως M/M/s/s+m.

Μαρκοβιανά συστήματα απωλειών: Τα μαρκοβιανά συστήματα απωλειών είναι συστήματα με χρόνους άφιξης και εξυπηρέτησης εκθετικά κατανεμημένους. Τέτοιου είδους συστήματα χρησιμοποιήθηκαν στο παρελθόν στα παλαιά συμβατικά τηλεφωνικά δίκτυα. Στα συστήματα απωλειών κριτήριο αξιολόγησης της απόδοσης αποτελεί η πιθανότητα απώλειας κλήσεως. Στη συνέχεια εξετάζονται τα ακόλουθα συστήματα απωλειών:

1. Το σύστημα M/M/s(0) (σύστημα με Poisson κατανομή άφιξης των κλήσεων, εκθετική κατανομή του χρόνου εξυπηρέτησης και s αριθμό εξερχόμενων γραμμών (εξυπηρετητών) με μηδενική ουρά.
2. Το σύστημα M(n)/M/s(0) (σύστημα με εκθετική κατανομή άφιξης και εξυπηρέτησης των κλήσεων και s αριθμό εξερχόμενων γραμμών με μηδενική ουρά αλλά με πεπερασμένο αριθμό εισόδων n).

Σύστημα απωλειών M/M/s(0): Στο σύστημα απωλειών M/M/s(0) οι αφίξεις ακολουθούν κατανομή Poisson και ο χρόνος εξυπηρέτησης των κλήσεων εκθετική κατανομή. Θεωρο-

ύμε επίσης ότι το σύστημα αυτής της μορφής διαθέτει άπειρο αριθμό εισερχόμενων γραμμών. Σε περίπτωση που οι αφίξεις βρουν όλους τους εξυπηρετητές απασχολημένους εγκαταλείπουν αμέσως το σύστημα. Συνεπώς, ο αριθμός των κλήσεων που μένουν στο σύστημα και δεν το εγκαταλείπουν, ισούται με τον αριθμό των εξυπηρετητών (εξερχόμενων γραμμών που διαθέτει το σύστημα).

Τα συστήματα απωλειών της μορφής $M/M/s(0)$ προσεγγίζονται ικανοποιητικά αν θεωρήσουμε ότι η προσφερόμενη κίνηση είναι καθαρά τυχαία κίνηση. Σ' αυτή την περίπτωση οι αφίξεις των κλήσεων θα έχουν – όπως προαναφέρθηκε – κατανομή Poisson.

Ο αριθμός των ταυτόχρονων καταλήψεων των εξερχόμενων γραμμών θα έχει την ονομαζόμενη κατανομή ERLANG. Σύμφωνα με αυτήν, η πιθανότητα να βρεθούν r εξερχόμενες γραμμές ταυτόχρονα κατειλημμένες σε μία χρονική στιγμή ισούται με:

$$P_r = \frac{\frac{a^r}{r!}}{\sum_{i=1}^s \frac{a^i}{i!}}$$

$r = 0, 1, 2, \dots, s$ όπου a είναι το προσφερόμενο φορτίο, s ο αριθμός των εξερχόμενων γραμμών ή εξυπηρετητών του συστήματος και r ο αριθμός των κατειλημμένων εξερχόμενων γραμμών. Στην ειδική περίπτωση που ο αριθμός των εξερχόμενων γραμμών γίνει πολύ μεγάλος

$$(s \rightarrow \infty)$$

η κατανομή ERLANG μετατρέπεται στην κατανομή Poisson οπότε:

$$P_r \rightarrow \frac{a^r}{r!} * e^{-a}$$

Η πιθανότητα σε μία χρονική στιγμή να βρεθούν όλες οι εξερχόμενες γραμμές κατειλημμένες δίνεται από τον τύπο απωλειών του ERLANG:

$$B_r = \frac{\frac{a^s}{s!}}{\sum_{i=1}^s \frac{a^i}{i!}} = E_s(a)$$

ο οποίος αναφέρεται και ως RLANG B – formula, που εκφράζει το ποσοστό απωλειών κλήσεων. Από την παραπάνω σχέση προκύπτει ο αναδρομικός τύπος της B – formula ERLANG:

$$E_s(a) = \frac{a * E_s - 1(a)}{s + a * E_s - 1(a)}, E_0(a) = 1$$

Η παραπάνω σχέση είναι ιδιαίτερα χρήσιμη όταν δίνεται το φορτίο κίνησης και η επιθυμητή πιθανότητα απώλειας κλήσεων (βαθμός εξυπηρέτησης) και ζητείται να υπολογιστεί ο αριθμός των εξερχόμενων γραμμών. **Σύστημα απωλειών $(n)/s(0)$** Στο σύστημα απωλειών με

πεπερασμένο αριθμό εισόδων θεωρούμε τα εξής:

1. Ο αριθμός των εισόδων του συστήματος έχει πεπερασμένη τιμή ν .
2. Οι αφίξεις των κλήσεων ακολουθούν εκθετική κατανομή με μέση τιμή $1/\lambda$, όπου λ είναι ο ρυθμός αφίξεων των κλήσεων στο σύστημα.
3. Θεωρούμε ότι κάθε μία από τις εισερχόμενες ν πεπερασμένες γραμμές είναι, όσον αφορά την κίνησή της, ανεξάρτητη από τις άλλες γραμμές. Επίσης θεωρούμε ότι κάθε μία από τις ν γραμμές έχει την ίδια κίνηση.

Στην περίπτωση αυτή έχει διαπιστωθεί ότι ο αριθμός των ταυτόχρονα κατειλημμένων r εξερχόμενων γραμμών ακολουθεί κατανομή ENGSET. Η πιθανότητα μία χρονική στιγμή να υπάρχουν ταυτόχρονα r κατειλημμένες γραμμές ισούται με:

$$P_r = \frac{\binom{n}{r} * \left(\frac{\lambda}{\mu}\right)^r}{\sum_{i=1}^S \binom{n}{i} * \left(\frac{\lambda}{\mu}\right)^i}$$

όπου λ ο ρυθμός αφίξεων των κλήσεων στο σύστημα, m ο ρυθμός εξυπηρέτησης των κλήσεων (πολλές φορές αντί του μ δίνεται ο μέσος χρόνος εξυπηρέτησης των κλήσεων: $h = 1/m$), s ο αριθμός των εξερχόμενων γραμμών και r ο αριθμός των εξερχόμενων κατειλημμένων γραμμών.

Η πιθανότητα σε μία χρονική στιγμή να βρεθούν όλες οι εξερχόμενες γραμμές κατειλημμένες δίνεται από τη σχέση:

$$B_r = \frac{\binom{n}{S} * \left(\frac{\lambda}{\mu}\right)^S}{\sum_{i=1}^S \binom{n}{i} * \left(\frac{\lambda}{\mu}\right)^i}$$

Η πιθανότητα αυτή ονομάζεται πιθανότητα συμφόρησης του συστήματος και δηλώνει το ποσοστό του χρόνου στην ώρα αιχμής κατά το οποίο το σύστημα είναι πλήρως κατειλημμένο. Η πιθανότητα απώλειας κλήσεως δίνεται από τον τύπο απωλειών του ENGSET:

$$B = \frac{\binom{n-1}{S} * \left(\frac{\lambda}{\mu}\right)^S}{\sum_{i=1}^S \binom{n-1}{i} * \left(\frac{\lambda}{\mu}\right)^i}$$

Η παραπάνω πιθανότητα ορίζεται σαν το πηλίκο των μπλοκαρισμένων κλήσεων προς τον συνολικό αριθμό κλήσεων, δηλαδή εκφράζει το ποσοστό απωλειών με το οποίο διαβιβάζεται η κίνηση. Από αυτήν προκύπτει η αναδρομική σχέση για τον τύπο απωλειών του ENGSET:

$$B(s, n, v * h) = \frac{(n-1) * v * h * B(s-1, n, v * h)}{s + (n-s) * v * h * B(s-1, n, v * h)}$$

όπου $B(s, n, v * h)$ είναι η πιθανότητα απώλειας κλήσεων. Το v αντιπροσωπεύει τον ρυθμό άφιξης των κλήσεων από μία πηγή εισόδου. Το συνολικά προσφερόμενο φορτίο κίνησης a δίνεται από τη σχέση: $a = (n-r) * v * h$, από όπου προκύπτει $r = \frac{n*v*h-a}{v*h}$, όπου r είναι ο αριθμός των εξερχόμενων γραμμών που είναι κατειλημμένες. Επειδή το μεταφερόμενο φορτίο κίνησης a_c , δηλαδή το φορτίο που εξυπηρετείται από το σύστημα δίνεται από τη σχέση: $a_c = a * (1 - B_s) = \frac{n*v*h-a}{v*h}$, όπου B_s είναι ο βαθμός εξυπηρέτησης) προκύπτει ότι $a = \frac{n*v*h}{1+v*h(1-B_s)}$, απ' όπου προκύπτει ότι για το γινόμενο $v * h$ ισχύει: $v * h = \frac{a}{n-a*(1-B_s)}$, όπου $v * h$ είναι το φορτίο κίνησης για κάθε ελεύθερη πηγή εισόδου.

Η απόδοση των γραμμών δίνεται από τη σχέση: $Q = \frac{a_c}{s} = \frac{a*(1-B_s)}{s}$ δηλαδή είναι το πηλίκο της κίνησης που εξυπηρετείται προς τον συνολικό αριθμό γραμμών.

Μαρκοβιανά συστήματα αναμονής Σε αντίθεση με τα συστήματα απωλειών όπου σε κατάσταση συμφόρησης η εισερχόμενη κλήση μπλοκάρει και εγκαταλείπει το σύστημα, στα συστήματα αναμονής η κλήση μπορεί να περιμένει για να γίνει η σύνδεση. Στα συστήματα αναμονής, κριτήριο αξιολόγησης της απόδοσης του συστήματος είναι η πιθανότητα αναμονής ή ο μέσος χρόνος αναμονής στην ουρά.

Σύστημα αναμονής M/M/s Το σύστημα αναμονής M/M/s είναι σύστημα με Poisson κατανομή άφιξης κλήσεων, εκθετική κατανομή του χρόνου εξυπηρέτησης, s αριθμό εξερχόμενων γραμμών και άπειρες θέσεις αναμονής στις οποίες όλες οι κλήσεις παραμένουν μέχρι να εξυπηρετηθούν, κατά μέσον όρο επί χρόνο W . Στα συστήματα αυτού του είδους υπάγονται τα δίκτυα υπολογιστών και τα ψηφιακά τηλεφωνικά κέντρα.

Ας θεωρήσουμε ότι l είναι ο ρυθμός άφιξης των κλήσεων στο σύστημα, m ο ρυθμός εξυπηρέτησης των κλήσεων, s ο αριθμός των εξερχόμενων γραμμών και r ο αριθμός των κλήσεων στο σύστημα. Το σύστημα αναμονής //s σχηματικά είναι ακριβώς ίδιο με ένα σύστημα M/M/1 όπως έχουμε ήδη αναφερθεί στην αρχή του κεφαλαίου.

Σε ένα τέτοιο σύστημα αναμονής ισχύει η σχέση:

$a < s \Rightarrow \frac{a}{s} < 1$ Δηλαδή το προσφερόμενο φορτίο κίνησης ανά γραμμή εξυπηρέτησης θα πρέπει να είναι μικρότερο από τη μονάδα. Το μέγεθος $r = \frac{a}{s} < 1$ ονομάζεται απόδοση των γραμμών όπως και στα συστήματα απωλειών.

Ορίζουμε σαν πιθανότητα αναμονής την πιθανότητα ότι μία κλήση θα περιμένει στην ουρά μέχρι να εξυπηρετηθεί και τη συμβολίζουμε με $M(0)$:

$$M(0) = \frac{\frac{a^S}{S!} * \frac{S}{s-a}}{\sum_{r=0}^{S-1} \frac{a^r}{r!} + \frac{a^S}{S!} * \frac{S}{s-a}}$$

Η παραπάνω σχέση αποτελεί την εξίσωση του ERLANG C αλλά λόγω της δυσκολίας που παρουσιάζει στους υπολογισμούς συνήθως γίνεται χρήση της με τη μορφή:

$$M(0) = \frac{s * E_S(a)}{s - a * [1 - E_S(a)]}$$

όπου το $E_S(a)$ δίνεται από τον τύπο απωλειών του ERLANG B.

Η μέση τιμή των κλήσεων αναμονής δίνεται από τη σχέση:

$$L = M(0) * \frac{a}{s - a}$$

και εκφράζει τον αριθμό των κλήσεων που περιμένουν να εξυπηρετηθούν. Κάνοντας χρήση του νόμου του *Little* : $L = \lambda * W$ και της προηγούμενης σχέσης μπορούμε να υπολογίσουμε τον μέσο χρόνο αναμονής στην ουρά:

$$W \frac{L}{\lambda} = \frac{M(0)}{\mu * (s - a)}$$

ο οποίος εκφράζει το χρονικό διάστημα που πρέπει να περιμένει μία κλήση μέχρι να εξυπηρετηθεί.

Τέλος ένα άλλο μέγεθος με πρακτικό ενδιαφέρον είναι η κατανομή του χρόνου αναμονής, που εκφράζει το ποσοστό των κλήσεων που θα καθυστερήσουν περισσότερο από χρόνο t , ως προς όλες τις κλήσεις και δίνεται από τη σχέση:

$$M(t) = M(0) * e^{-(1-\rho)*s*\mu*t}$$

Όλα τα παραπάνω, είναι ακριβώς ίδια με όσα αναφέραμε στην αρχή του κεφαλαίου με ελάχιστες διαφορές στην σηματολογία και επιπλέον και ότι τα δίκτυα τηλεπικοινωνιών χρησιμοποιούν ευρέως την εξίσωση του ERLANG . [6] [7] [8]

Γ'.4 Call Admission Control (CAC)

Ένα δίκτυο Jackson, όπως προαναφέρθηκε, είναι ένα δίκτυο ανοιχτού ή κλειστού τύπου, με μια είσοδο, ενδάμεσους επεξεργαστές και έναν τελικό επεξεργαστή που καταλήγει στην έξοδο ή και όχι αν πρόκειται για κλειστού τύπου δίκτυο. Ουσιαστικά ένα δίκτυο Jackson αποτελείται

απο πολλά εσωτερικά συστήματα αναμονής M/M/1 και κατά συνέπεια συμπεριφέρεται ως ένα M/M/1 αποτελούμενο από πολλά μικρότερα. Η παρούσα εργασία υλοποιήθηκε ώστε δείξει την ακρίβεια του παραπάνω θεωρητικού συμπεράσματος καθώς και ότι ένα τηλεπικοινωνιακό δίκτυο είναι ένα δίκτυο Jackson, δηλαδή M/M/1.

Γενικά,

Ένα δίκτυο Jackson μπορεί να ανήκει σε οποιαδήποτε από τις παραπάνω κατηγορίες δικτύων. Ας υποθέσουμε ότι έχουμε ένα δίκτυο αναμονής m/m/1 και πως υπάρχουν ήδη n χρήστες με Poisson $\lambda_1, \lambda_2, \dots, \lambda_n$ που το χρησιμοποιούν. Οπότε, ο ρυθμός άφιξης $\lambda = \lambda_1 + \lambda_2 + \dots + \lambda_n$ καθώς είναι Poisson. Συνεπώς η καθυστέρηση T είναι ήση με

$$T = N/\lambda = \frac{\rho}{1-\rho} = \frac{\lambda^* \mu}{1-\rho} = \frac{1}{\mu^*(1-\rho)} = \frac{1}{\mu-\lambda}.$$

Έστω ότι οι χρήστες έχουν απαίτηση για χρονική καθυστέρηση μικρότερη ή ίση του d_i .

Υποθέτουμε πως υπάρχει μηχανισμός που λέει NAI/OXI σε κάθε νέο χρήστη προκειμένου να χρησιμοποιήσει το σύστημα.

Έστω ο χρήστης 1 με (λ_1, d_1) . Μπορεί να γίνει δεκτός.

Απάντηση: Αν γίνει δεκτός τότε η καθυστέρηση θα είναι $T = \frac{1}{\mu-\lambda_1}$. Αν ισχύει $\frac{1}{\mu-\lambda_1} \leq d_1$ τότε NAI.

Έστω πως ο χρήστης 1 με λ_1, d_1 έγινε δεκτός (προφανώς $\frac{1}{\mu-\lambda_1} \leq d_1$).

Έστω ένας δεύτερος χρήστης 2 με (λ_2, d_2) . Μπορεί να γίνει δεκτός. Σε αυτήν την περίπτωση το $\frac{1}{\mu-\lambda_2} \leq d_2$ δεν ισχύει.

Καθώς υπάρχει ήδη ο (λ_1, d_1) γνωρίζω πως ο ρυθμός άφιξης είναι πλέον $(\lambda_1 + \lambda_2)$. Οπότε, η καθυστέρηση θα πρέπει να είναι $\frac{1}{\mu-(\lambda_1+\lambda_2)} \leq d_2$. Τί γίνεται όμως με τον χρήστη 1.

Όταν έγινε δεκτός στο σύστημα ο 1 δεν υπήρχε ο 2. Αν μπει ο 2 (υποθέτοντας $\frac{1}{\mu-(\lambda_1+\lambda_2)} \leq d_2$), η επιπλέον κίνηση θα αυξήσει την καθυστέρηση και μπορεί ο 1 να μην ικανοποιεί την απαίτηση $\leq d_1$. Οπότε για να γίνει δεκτός ο 2 θα πρέπει να ισχύουν ΚΑΙ οι δύο συνθήκες,

$$\frac{1}{\mu-\lambda_1} \leq d_1 \text{ που ικανοποιείται ο 1 και}$$

$$\frac{1}{\mu-\lambda_1} \leq d_2 \text{ που ικανοποιείται ο 2.}$$

Σε πιο συνεπτυγμένη μορφή, ο 2 γίνεται δεκτός αν $\frac{1}{\mu-(\lambda_1+\lambda_2)} \leq \min(d_1, d_2)$.

Γενική περίπτωση:

Αν έχουμε n χρήστες ήδη στο σύστημα, ποιά η συνθήκη για να υποδεχτούμε τον $n+1$.

Απάντηση: Εφόσον είναι ήδη n χρήστες στο σύστημα θα ισχύει:

$$\frac{1}{\mu-\sum_{i=1}^n \lambda_i} \leq \min(d_i), i = 1, 2, \dots, n$$

Για να γίνει αποδεκτός ο $n+1$ θα πρέπει

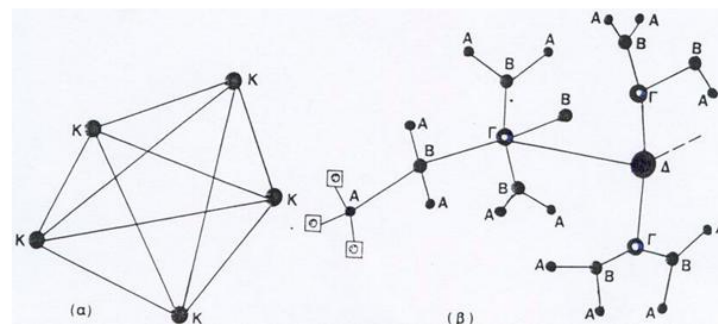
$$\frac{1}{\mu-\sum_{i=1}^{n+1} \lambda_i} \leq \min(d_i), i = 1, 2, \dots, n+1.$$

Η παραπάνω συνθήκες αποτελούν το πρόβλημα αποδοχής μιας κλήσης (Call Admission και

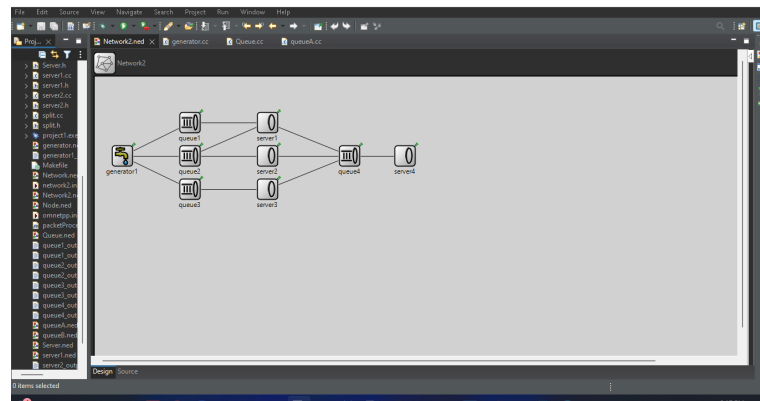
πώς πρέπει να απαντήσουμε μετά απο έλεγχο (Control) - CAC

Γ'.5 Περιγραφή πειραματικού δικτύου

Το πειραματικό δίκτυο που δημιουργήθηκε στα πλαίσια εκπόνησης της πτυχιακής εργασίας υλοποιήθηκε στην πλατφόρμα OMNETT++ εκδόσεως 5.7. Απαρτίζεται από τρία εσωτερικά συστήματα αναμονής M/M/1 και καθώς είναι ανοιχτού τύπου σύστημα αναμονής έχει μία 'κύρια' είσοδο και μία 'κύρια' έξοδο. Συγκεκριμένα, κάθε ένα σύστημα αναμονής M/M/1 έχει δύο επιλογές εισόδου, μία την 'κύρια' είσοδο του συστήματος και μία την είσοδο προερχόμενη από άλλο σύστημα αναμονής M/M/1 του ίδιου δικτύου. Παράλληλα, έχει και τη δυνατότητα δύο εξόδων, μία της 'κύριας' εξόδου του συστήματος και μία προς την είσοδο ενός άλλου συστήματος αναμονής M/M/1 του ίδιου δικτύου. Χάρην του visualization της πλατφόρμας μπορούμε να δούμε την κίνηση των πακέτων.



Σχήμα Γ'.3: (α) Πολυγωνική σύνδεση και (β) ακτινωτή σύνδεση τηλεφωνικών κέντρων [8]



Σχήμα Γ'.4: Πειραματικό δίκτυο Jackson

Κεφάλαιο Δ΄

Υλοποίηση Δικτύου

Δ΄.1 Υλοποίηση και περιγραφή δικτύου

Δ΄.1.1 Πλατφόρμα OMNeT++

Το IDE προσομοίωσης OMNeT++ βασίζεται στην πλατφόρμα Eclipse και το επεκτείνει με νέους επεξεργαστές, προβολές, οδηγούς και άλλες λειτουργίες. Το OMNeT++ προσθέτει λειτουργικότητα για τη δημιουργία και διαμόρφωση μοντέλων (αρχεία NED και INI), εκτέλεση ομαδικών εκτελέσεων και ανάλυση των αποτελεσμάτων προσομοίωσης, ενώ το Eclipse παρέχει επεξεργασία C++, ενοποίηση SVN/GIT και άλλες προαιρετικές λειτουργίες (μοντελοποίηση UML, ενσωμάτωση bug-tracker, πρόσβαση σε βάσεις δεδομένων κ.λπ.) μέσω διαφόρων προσθηκών ανοιχτού κώδικα. Το περιβάλλον είναι άμεσα αναγνωρίσιμο σε όσους με την πλατφόρμα Eclipse.

Ο χώρος εργασίας στο OMNeT++ είναι βασικά ένας κατάλογος όπου βρίσκονται όλα τα έργα. Μπορούμε να δημιουργήσουμε και να χρησιμοποιήσουμε αρκετούς χώρους εργασίας και να τους εναλλάζουμε μεταξύ τους ανάλογα με τις ανάγκες. Κατά την πρώτη διαδρομή, το OMNeT++ IDE προσφέρεται να ανοίξει τον κατάλογο δειγμάτων ως χώρο εργασίας, ώστε να μπορούμε να πειραματιστούμε με τα διαθέσιμα παραδείγματα αμέσως. Μόλις αρχίσουμε να εργαζόμαστε στο δικό μας έργο, συνίσταται να δημιουργήσουμε τον δικό μας χώρο εργασίας επιλέγοντας Αρχείο → Εναλλαγή, Χώρος εργασίας → Άλλο. Μπορούμε να κάνουμε εναλλαγή μεταξύ χώρων εργασίας, αν αυτό απαιτείται. Υπόψη ότι το OMNeT++ IDE επανεκκινείται με κάθε διακοπή στους χώρους εργασίας.

Δ'.1.1.1 Το NED αρχείο

Για να δημιουργήσω ένα ned αρχείο, από το μενού File -> New -> Simple Module. Για να δημιουργήσω το δικό μου δίκτυο επιλέγω σαν template “A simple module” και πατάω Finish. Τη στιγμή εκείνη δημιουργούνται τρία νέα αρχεία στο project folder και το αρχείο .ned είναι ήδη ανοιχτό στον editor. Είναι τα αρχεία .cc και .h και μπορώ να τα ανοίξω με διπλό κλικ. Αφού έχω ήδη ανοιχτό το .ned αρχείο στο κάτω μέρος του παραθύρου του editor, μπορώ να δώ το αρχείο σε Design view και Source view. Επιλέγω Source view και βλέπω ότι ήδη περιέχει την δηλωση module με όποιο όνομα έδωσα παραπάνω και τύπο simple. Στο .ned γίνεται η προσθήκη/δήλωση των κόμβων, η εμφάνισή τους στον καμβά προσομοίωσης, καθώς επίσης και η δήλωση μεταβλητών με αρχικοποίηση, αν αυτό κρίνεται απαραίτητο.

Δ'.1.1.2 Το .h αρχείο

Είμι ένα απο τα αρχεία που δημιουργούνται αυτόματα κατά την δημιουργία του .ned και στο οποίο έχει γίνει include το αρχείο omnetpp.h. Στο αρχείο αυτό έχει δημιουργηθεί μία κλάση με το όνομα που δώσαμε στον κόμβο μας η οποία κληρονομεί την κλάση cSimpleModule. Στην κλάση αυτή έχουν ήδη δηλωθεί δύο μέθοδοι οι οποίες κάνουν override τις ήδη υπάρχουσες της cSimpleModule. Μέσα σε αυτές τις μεθόδους γράφουμε κώδικα περιγράφοντας την συμπεριφορά του κόμβου μας και καλούνται αυτόματα όταν πρέπει.

Δ'.1.1.3 Το .cc αρχείο

Το αρχείο .cc έχει έτοιμες προς υλοποίηση τις δύο παραπάνω μεθόδους οι οποίες αρχικά είναι κενές. Μέσα στην initialize() γράφουμε κώδικα ο οποίος θέλουμε να εκτελεστεί μία και μοναδική φορά κατά την έναρξη της προσομοίωσης. Καλείται αυτόματα με την έναρξη αυτής και μέσα στην handleMessage() γράφουμε κώδικα ο οποίος θα εκτελείται οποτεδήποτε ο κόμβος μας λαμβάνει ένα νέο μήνυμα. Το μήνυμα αυτό θα είναι ένα αντικείμενο της κλάσης cMessage το οποίο θα μας δίνεται στην μεταβλητή-δείκτη msg η οποία είναι παράμετρος της handleMessage(). Αν παρατηρήσουμε θα δούμε επίσης ότι δεν υπάρχει Constructor της κλάσης και δεν πρέπει να υπάρχει καθώς σύμφωνα με το οδηγίες του OMNeT++ αναλαμβάνει αυτό να τον δημιουργήσει και να τον χρησιμοποιήσει όπως πρέπει.

Δ'.1.1.4 Το omnetpp.ini αρχείο

Πρόκειται για ένα αρχείο το οποίο δημιουργήτε κατά την προσομοίωση του δικτύου. Σε αυτό είναι δηλωμένο το όνομα του δικτύου. Επίσης σε αυτό μπορούμε να δηλώσουμε μεταβλητές οι οποίες θα έχουν εμβέλεια σε όλα τα αρχεία του project folder.

Δ'.1.2 Περιγραφή πειραματικού δικτύου

Το πειραματικό δίκτυο Jackson είναι ένα δίκτυο ανοιχτού τύπου, με μια είσοδο, ενδάμεσους επεξεργαστές και έναν τελικό επεξεργαστή που καταλήγει στην έξοδο. Ουσιαστικά ένα δίκτυο Jackson αποτελείται από πολλά εσωτερικά συστήματα αναμονής $M/M/1$ και κατά συνέπεια συμπεριφέρεται ως ένα $M/M/1$ αποτελούμενο από πολλά μικρότερα. Η παρούσα πτυχιακή εργασία υλοποιήθηκε ώστε δείξει την ακρίβεια του παραπάνω θεωρητικού συμπεράσματος καθώς και ότι ένα τηλεπικοινωνιακό δίκτυο είναι ένα δίκτυο Jackson, δηλαδή $M/M/1$. Συγκεκριμένα, το δίκτυο Jackson αποτελείται από τρία εσωτερικά $M/M/1$, μία γεννήτρια τυχαίων συμβάντων και έναν τελικό επεξεργαστή, όπου και η έξοδος από το σύστημα. Μπορούμε να προσθέσουμε όσα εσωτερικά συστήματα αναμονής $M/M/1$ θέλουμε ή χρειαζόμαστε ανάλογα με τις ανάγκες του δικτύου μας. Κάθε εσωτερικό σύστημα αναμονής $M/M/1$ έχει τη δική του εσωτερική επεξεργαστική μονάδα αλλά μία μόνο κοινή γεννήτρια - είσοδο, αυτήν του ευρύτερου συστήματος αναμονής του δικτύου Jackson, όπως και μία μόνο έξοδο από το σύστημα, τον τελικό server του δικτύου Jackson. Στο Σχήμα Δ'9 φαίνεται να υπάρχει μεγαλύτερη καθυστέρηση για την έξοδο του πακέτου από το σύστημα επειδή ο server1 έχει την επιλογή να προωθήσει το πακέτο στην ουρά του εσωτερικού κόμβου queue2 και προς τον τελικό server όπου και η έξοδος του πακέτου από το σύστημα, ενώ στο Σχήμα Δ'10 όπου ο εσωτερικός κόμβος προωθεί τα πακέτα στον τελικό και στην έξοδο έχουμε αναμενόμενο χρόνο αναμονής.

Δ'.2 Περιγραφή των δεδομένων συλλογής

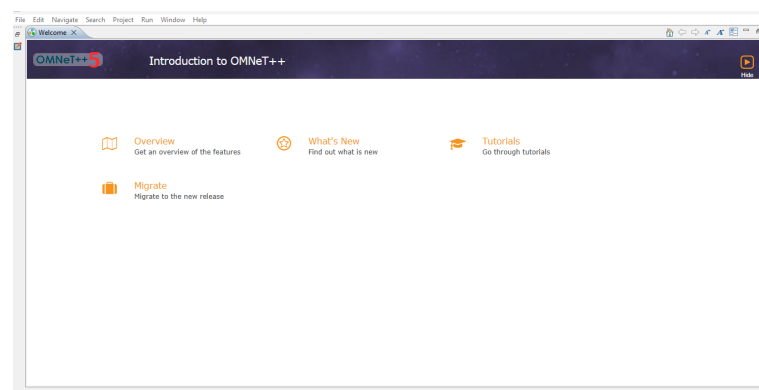
Για την καλύτερη κατανόηση και συλλογή δεδομένων του πειραματικού δικτύου της πτυχιακής εργασίας χρησιμοποιήθηκε η πλατφόρμα GNUplot. Το Gnuplot είναι ένα φορητό βοηθητικό πρόγραμμα γραφικών, ανοικτού κώδικα, που βασίζεται στη γραμμή εντολών για Linux, OS/2, MS Windows, OSX, VMS, και πολλές άλλες πλατφόρμες. (Να ανεβάσω το documentation. Ως δεδομένα εισόδου χρησιμοποιήθηκαν οι χρόνοι άφιξης των γεγονότων και ο χρόνος επεξεργασίας τους. Κάθε ένα από τα γραφήματα αναπαριστάνει την σχέση μεταξύ

αφίξεων - αναχωρήσεων, καθυστέρηση γεγονότων στην ουρά καθώς και τον ρυθμό με τον οποίο δημιουργούνται και επεξεργάζονται τα γεγονότα. [5]

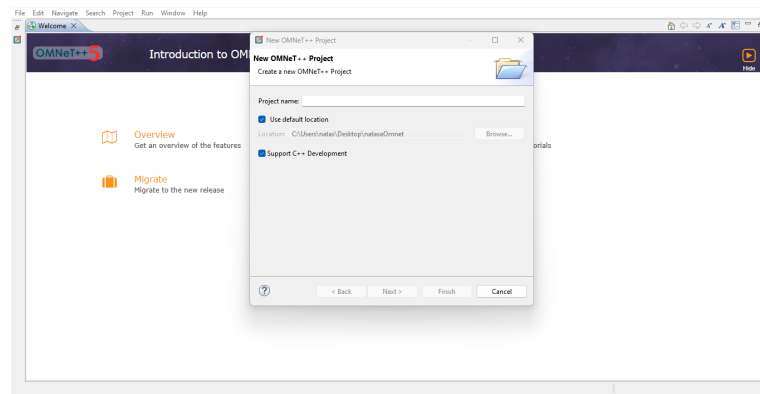
Δ΄.3 Δυσκολίες και προβλήματα κατά την υλοποίηση

Μια από τις δυσκολίες κατά την υλοποίηση του πειραματικού δικτύου ήταν η εξαγωγή του χρόνου καθυστέρησης στην ουρά κάθε πακέτου, στο δίκτυο Jackson, ιδιαίτερα αν το πακέτο αυτό έπρεπε να μεταβεί σε άλλες εσωτερικές ουρές για περαιτέρω επεξεργασία, πριν περάσει στην τελική ουρά και τέλος στην έξοδο. Το πρόβλημα αυτό αντιμετωπίστηκε με την εισαγωγή μοναδικών αριθμών, IDs σε κάθε πακέτο και με μια μεταβλητή στην οποία προστιθεται ο χρόνος αναμονής και επεξεργασίας κάθε πακέτου ξεχωριστά. Δηλαδή, κάθε πακέτο 'κουβαλάει' τον δικό του μοναδικό αριθμό και μια μεταβλητή η οποία απεικονίζει τον χρόνο που το πακέτο αυτό παρέμεινε στο σύστημα από την αρχή της δημιουργίας του ως και τον χρόνο που εξήλθε του συστήματος.

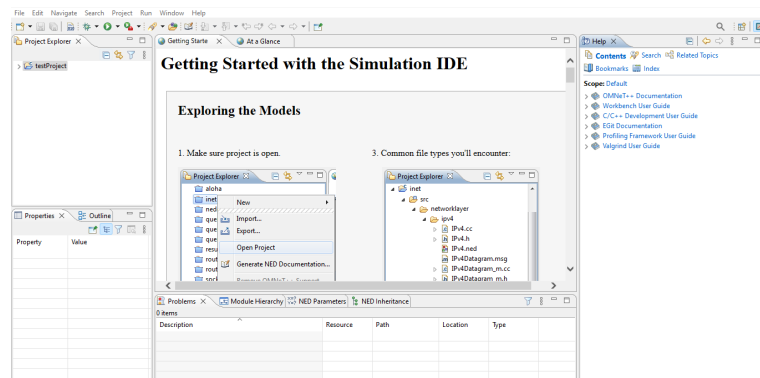
Μια δεύτερη ήταν η απεικόνιση της διαδρομής του πακέτου όταν αυτό έπρεπε να μεταβεί σε διαφορετικούς εσωτερικούς κόμβους πριν την έξοδό του. Αυτό επιλύθηκε με την εισαγωγή ονομάτων στους κόμβους. Κάθε φορά που ένα πακέτο μεταβαίνει από έναν κόμβο σε έναν άλλο εμφανίζεται στο δίκτυο το όνομα του κόμβου στον οποίο μεταβαίνει καθώς και ο μοναδικός αριθμός του πακέτου. Έτσι, ο χρήστης του δικτύου μπορεί να γνωρίζει ποιό πακέτο πάει σε ποιόν κόμβο. Θα μπορούσαμε να εμφανίζουμε και άλλες πληροφορίες του πακέτου, όπως για παράδειγμα τον τρέχον χρόνο παραμονής του στο σύστημα.



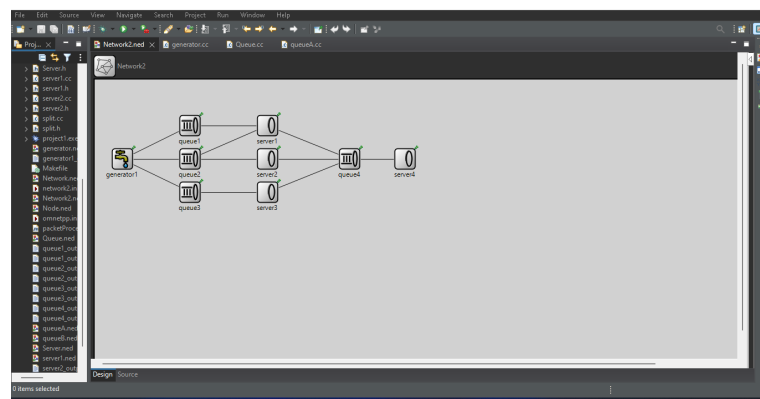
Σχήμα Δ΄.5: Η πλατφόρμα Omnet - Αρχική σελίδα



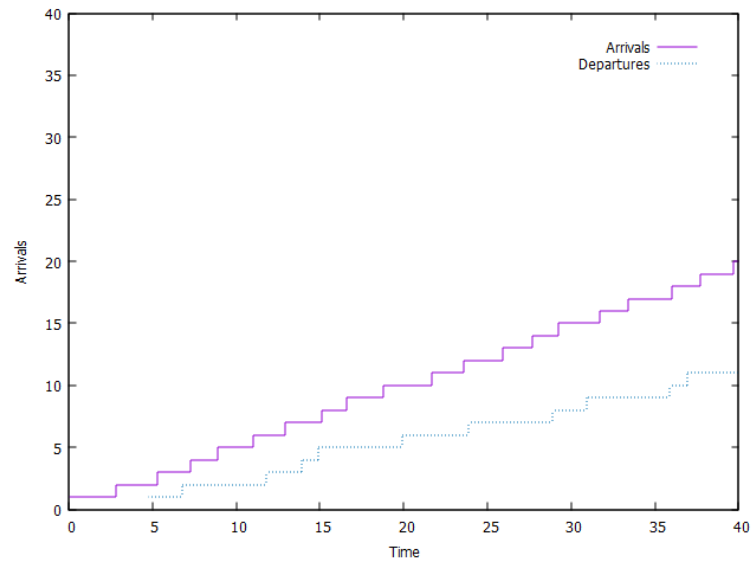
Σχήμα Δ'.6: Η πλατφόρμα Omnet - Επιλογή Workspace



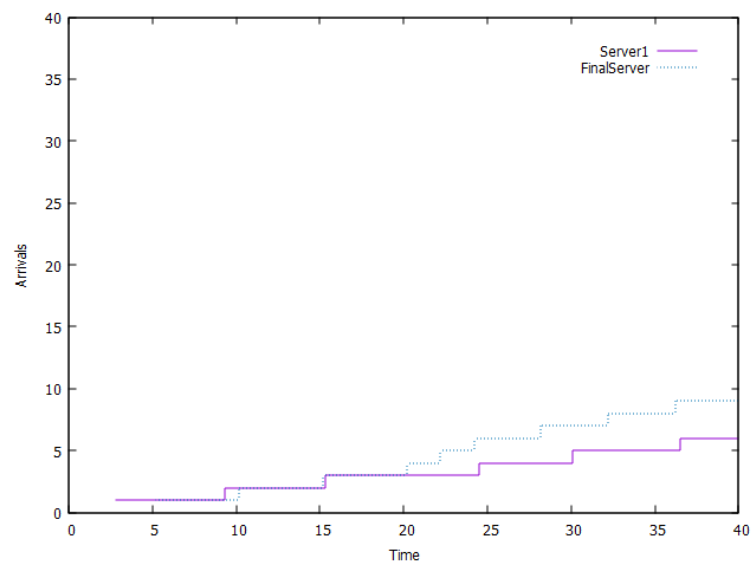
Σχήμα Δ'.7: Η πλατφόρμα Omnet - Workspace



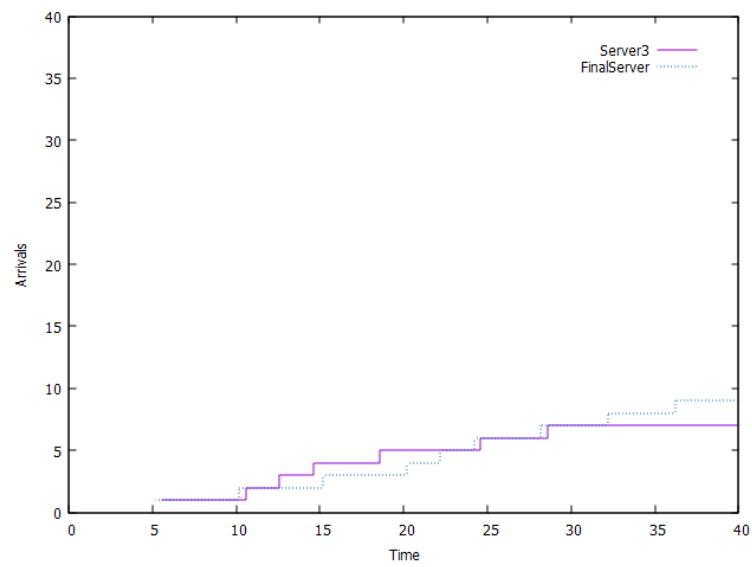
Σχήμα Δ'.8: Πειραματικό δίκτυο Jackson



Σχήμα Δ'.9: Αφίξεις και Αναχωρήσεις σε κατανομή Poisson του δικτύου Jackson



Σχήμα Δ'.10: Χρόνος αναμονής στο σύστημα απο τον πρώτο server ως τον τελικο



Σχήμα Δ'.11: Χρόνος αναμονής στο σύστημα απο εσωτερικό κόμβο - επεξεργαστή(server3) ως τον τελικο

Κεφάλαιο Ε΄

Σύνοψη και Συμπεράσματα

Τα συστήματα αναμονής, ανεξάρτητα από το μέγεθός τους, το πόσους κόμβους και επεξεργαστές περιλαμβάνουν, το ρυθμό με τον οποίο δημιουργούνται και επεξεργάζονται συμβάντα υπάρχουν σε κάθε πτυχή της καθημερινότητάς μας. Από το σούπερ μάρκετ ως την τράπεζα, σε σταθμούς λεοφωρείων, τρένων, αεροπλάνων, τηλεπικοινωνιακά δίκτυα αλλά ακόμα και στην ίδια την ύπαρξή μας, μπορούμε να διακρίνουμε τα τρία βασικά χαρακτηριστικά ενός συστήματος αναμονής:

1. Άφιξη
2. Αναμονή
3. Αναχώρηση

Ο τρόπος επεξεργασίας και ο ρυθμός με τον οποίο δημιουργούνται τα γεγονότα είναι τα χαρακτηριστικά εκείνα που μας επιτρέπουν να κατηγοριοποιήσουμε τα συστήματα αναμονής. Έτσι και τα δίκτυα Jackson είναι ένα σύστημα αναμονής που αποτελείται από πολλά μικρότερα. Τα κύρια συμπεράσματα που προέκυψαν από την παρούσα πτυχιακή εργασία είναι:

- Τα δίκτυα Jackson συμπεριφέρονται όπως τα συστήματα αναμονής $M/M/1$, αφού απορτίζονται από αυτά, με μία μικρή ίσως διαφορά στον χρόνο αναμονής που και πάλι είναι ανάλογο του μεγέθους του κάτι που μπορούμε να διακρίνουμε και από τη γραφική παράσταση (Σχήμα Δ΄ 8). [1] [2] [3]
- Ο χρόνος επεξεργασίας ανάμεσα στους ενδιάμεσους κόμβους ως τον τελικό είναι ανάλο-

γος της εσωτερικής αναμονής στο σύστημα (Σχήμα Δ'9 - Σχήμα Δ'10).

- Η αποδοχή μιας κλήσης μετά απο έλεγχο CAC, προϋποθέτει οτι στο σύστημα θα πρέπει αυτοί που ήδη υπάρχουν να εξυπηρετούνται ή άλλως να εξέρχονται απο το σύστημα με τη μικρότερη δυνατή καθυστέρηση.
- Κάνοντας χρήση του νόμου του *Little* : $L = \lambda * W$ σε συνδυασμό με την εξίσωση ERLANG B (για τις απώλειες τηλεφωνικών συνδέσεων) μπορούμε να υπολογίσουμε τον μέσο χρόνο αναμονής στην ουρά, ο οποίος εκφράζει το χρονικό διάστημα που πρέπει να περιμένει μία κλήση μέχρι να εξυπηρετηθεί [8]

Η μελέτη των δικτύων Jackson, καθώς και η υλοποίηση του πειραματικού δικτύου, οδήγησε τόσο στην εξοικίωση με τα κατανομημένα δίκτυα, όσο και στην λειτουργία αυτών. Η υλοποίηση του δικτύου επιβεβαίωσε όλα όσα μελετήθηκαν στην βιβλιογραφική έρευνα για τη συμπεριφορά των δικτύων Jackson καθώς και για τον έλεγχο αποδοχής μιας κλήσης. Η πτυχιακή εργασία αυτή αποτελεί προοίμιο για περαιτέρω μελέτη, υλοποίηση και εξαγωγή συμπερασμάτων της αποδοχής μιας κλήσης μετά απο έλεγχο.

Βιβλιογραφία

- [1] DIMITRIS BERTSEKAS, ROBERT GALLAGER, *DATA NETWORKS*, Delay models in data networks, 1992.
- [2] BOUDEWIJN R. HAVERKORT, *PERFORMANCE OF COMPUTER COMMUNICATION SYSTEMS*, A Model-Based Approach, 1998.
- [3] PIET VAN MIEGHEM, *PERFORMANCE ANALYSIS OF COMMUNICATIONS NETWORKS AND SYSTEMS*, Queueing models, 2006.
- [4] Γιάννης Γαροφαλάκης Τεχνικές Εκτίμησης Υπολογιστικών Συστημάτων, 2021
- [5] Thomas Williams and Colin Kelley, *An Interactive Plotting Program*, 1986-2004
- [6] Μ. Χ. Λογοθέτης Θεωρία τηλεπικοινωνιακής κινήσεως και εφαρμογές, Εκδόσεις Παπασωτηρίου, 2001
- [7] Ε. Χ. Μπίλλης Τηλεπικοινωνιακά Συστήματα, Τόμοι Ι
II, Εκδόσεις Συμμετρία, 2003
- [8] ΤΕΙ ΚΕΝΤΡΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ ΣΗΜΕΙΩΣΕΙΣ ΜΑΘΗΜΑΤΟΣ ΕΙΣΑΓΩΓΗ ΣΤΑ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΑΚΑ ΔΙΚΤΥΑ

Παράρτημα ‘Α’

Στο Παράρτημα παρατήθεται το αρχείο του κώδικα που υλοποιήθηκε στα πλαίσια εκπόνησης της πτυχιακής εργασίας και συλλογής δεδομένων μεταξύ των κόμβων.

Το αρχείο Network.ned

```
network Network2
{
    parameters:

        @display("bgb=656,279");

    submodules:

        generator1: Generator {
            @display("p=46,113");
            myName = "generator1";
        }
        queue1: Queue {
            @display("p=172,83");
            myName = "queue1";
        }
        server1: Server {
            @display("p=312,83");
            myName = "server1";
        }
}
```

```

queue2: Queue {
    @display("p=172,143");
    myName = "queue2";
}
server2: Server {
    @display("p=312,143");
    myName = "server2";
}

queue3: Queue {
    @display("p=512,113");
    myName = "queue3";
}
server3: Server {
    @display("p=572,113");
    myName = "server3";
}
connections:
    generator1.toQueue++ <—> queue1.in++;
    generator1.toQueue++ <—> queue2.in++;

    queue1.toServer++ <—> server1.fromQueue++;
    queue2.toServer++ <—> server2.fromQueue++;

    server1.out++ <—> queue3.in++;
    server1.out++ <—> queue2.in++;
    server2.out++ <—> queue3.in++;

    queue3.toServer++ <—> server3.fromQueue++;

}

```

To αρχείο generation.ned

```
simple Generator
{
    parameters:

        @display("i=block/source");
        string myName;

    gates:
        inout toQueue[];
}
```

To αρχείο generation.h

```
#ifndef _PROJECT1_GENERATOR_H_
#define _PROJECT1_GENERATOR_H_

#include <omnetpp.h>
#include "packetProcess_m.h"

using namespace omnetpp;

/**
 * TODO - Generated class
 */
class Generator : public cSimpleModule
{
protected:
    virtual void initialize();
    virtual void handleMessage(cMessage *msg);
    virtual void finish();
    int selectRandomOutput();
public:
    int totalCountMessages;
    std::string myName;
```

```

    std::vector<double> *timeX;
    std::vector<int> *arrivalsY;

};

#endif

To αρχείο generation.cc

#include "generator.h"

Define_Module(Generator);

void Generator::initialize()
{
    timeX = new std::vector<double>();
    arrivalsY = new std::vector<int>();

    myName = par("myName").stringValue();
    EV<<myName.c_str();

    cMessage* startMessage = new cMessage("start");
    scheduleAt(simTime()+0.1, startMessage);

    totalCountMessages=0;
}

void Generator::handleMessage(cMessage *msg)
{
    std::string message = msg->getName();
    if (message=="start")
    {
        scheduleAt(simTime(), new cMessage("generate"));
    }
}

```

```

        delete msg;
    }

    if (message=="generate") //generator poisson
    {

        //EV<<myName<<endl;

        packetProcess* pp = new packetProcess("process");
        pp->setProcessId(-1);
        int processingTime = intuniform(1,7);
        pp->setProcessingTime_total(processingTime);
        int randomGateIndex=selectRandomOutput();
        send(pp, gate("toQueue$o", randomGateIndex));

        //double next_event = ((double)poisson(10))/10; // 0, 0.1, 0.2, ...
        double next_event = ((double)poisson(20))/10; // 0, 0.1, 0.2, ... 2
        totalCountMessages+=1;
        std::string bubbleMessage = "generated:_ " + std::to_string(totalCountMessages) + " _";
        bubble(bubbleMessage.c_str());
        EV<<"This_message_was_created_after_"<<next_event<<"_seconds."<<endl;
        scheduleAt(simTime()+next_event, msg);
        timeX->push_back(pp->getCreationTime().dbl());
        arrivalsY->push_back(totalCountMessages);

    }

    EV<<"My_Messages_are:_ "<< totalCountMessages<<endl;

}

```

```

void Generator::finish()
{
    std::string outputFileName = "";
    outputFileName.append(this->myName);
    outputFileName.append("_output_arrivals.txt");

    fopen("output_arrivals.txt","w+");
    FILE* arrivalsTime = fopen(outputFileName.c_str(),"w+");
    fprintf(arrivalsTime, "%lf_",timeX->at(0));
    fprintf(arrivalsTime, "%d_\n",arrivalsY->at(0));
    for (int i=1;i<arrivalsY->size();i++)
    {
        if (timeX->at(i)==timeX->at(i-1) && arrivalsY->at(i)==arrivalsY->at(i-1))
        {
            continue;
        }
        fprintf(arrivalsTime, "%lf_",timeX->at(i));
        fprintf(arrivalsTime, "%d_\n",arrivalsY->at(i));
    }
    fclose(arrivalsTime);
}

int Generator::selectRandomOutput()
{
    return intuniform(0,this->gateSize("toQueue")-1);
}

```

To αρχείο Queue.ned

```

simple Queue
{

```

```

    parameters:
        int totalCountMessages = default(-1);
        @display("i=block/queue");

```

```

        string myName;

    gates:
        inout in [];
        inout toServer [];

}

```

To αρχείο Queue.h

```

#ifndef __PROJECT1_QUEUE_H_
#define __PROJECT1_QUEUE_H_

#include <omnetpp.h>

using namespace omnetpp;

/**
 * TODO - Generated class
 */
class Queue : public cSimpleModule
{
protected:
    virtual void initialize();
    virtual void handleMessage(cMessage *msg);
    virtual void finish();
    std::vector<cMessage*> *vectorQueue;
    bool serverRequestedMessage;

    int totalCountMessages;
    std::string myName;

    std::vector<double> *valuesX;
    std::vector<int> *valuesY;

```

```

        std::vector<double> *valuesX_arrivalsDepartures;
        std::vector<int> *valuesY_arrivalsDepartures;
    };

#endif

To αρχείο Queue.cc

#include "Queue.h"

Define_Module (Queue);

void Queue::initialize ()
{
    totalCountMessages=0;
    serverRequestedMessage = false;
    vectorQueue = new std::vector<cMessage*>();

    valuesX = new std::vector<double>();
    valuesY = new std::vector<int>();

    valuesX_arrivalsDepartures = new std::vector<double>();
    valuesY_arrivalsDepartures = new std::vector<int>();

    myName = par("myName").stringValue();
    EV<<myName.c_str();

    scheduleAt(0, new cMessage("countQueueElements"));
}

void Queue::handleMessage(cMessage *msg)
{
    std::string message = msg->getName();
    if (message=="process")

```

```

{
    totalCountMessages+=1;
    std::string bubbleMessage = "received:_" + std::to_string(totalCountMessages);
    bubble(bubbleMessage.c_str());

    valuesX_arrivalsDepartures->push_back(simTime().dbl());
    valuesY_arrivalsDepartures->push_back(vectorQueue->size());

    vectorQueue->push_back(msg);
    EV<<"queue_size_"<<vectorQueue->size();

    if (serverRequestedMessage==true)
    {

        cMessage *tmp=vectorQueue->at(0);
        vectorQueue->erase(vectorQueue->begin());
        int randomGateIndex=0;
        send(tmp, gate("toServer$o", randomGateIndex));
        serverRequestedMessage=false;
        std::string bubbleMessage = "in_queue:_" + std::to_string(vectorQueue->size());
        bubble(bubbleMessage.c_str());

    }

    valuesX_arrivalsDepartures->push_back(simTime().dbl());
    valuesY_arrivalsDepartures->push_back(vectorQueue->size());

    //send(new cMessage("iAmQueue"), gate("in$o", 0));

}
else if (message=="giveMeNext")

```

```

{
    if (vectorQueue->size()==0)
    {
        valuesX_arrivalsDepartures->push_back(simTime().dbl());
        valuesY_arrivalsDepartures->push_back(vectorQueue->size());

        bubble("Queue_empty!");
        serverRequestedMessage = true;

        valuesX_arrivalsDepartures->push_back(simTime().dbl());
        valuesY_arrivalsDepartures->push_back(vectorQueue->size());
    }
    else
    {
        valuesX_arrivalsDepartures->push_back(simTime().dbl());
        valuesY_arrivalsDepartures->push_back(vectorQueue->size());

        cMessage *tmp=vectorQueue->at(0);
        vectorQueue->erase(vectorQueue->begin());
        int randomGateIndex=0;
        send(tmp, gate("toServer$o", randomGateIndex));

        std::string bubbleMessage = "in_queue:_ " + std::to_string(vectorQueue->size());
        bubble(bubbleMessage.c_str());

        valuesX_arrivalsDepartures->push_back(simTime().dbl());
        valuesY_arrivalsDepartures->push_back(vectorQueue->size());
    }
    delete msg;
}

{
    valuesX->push_back(simTime().dbl());

```

```

        valuesY->push_back(vectorQueue->size());
        scheduleAt(simTime()+1, msg);
    }
}

void Queue::finish()
{

    std::string outputFileName = "";
    outputFileName.append(this->myName);
    outputFileName.append("_output_arrivals.txt");

    FILE* outputFile = fopen(outputFileName.c_str(),"w+");
    for (int i=0;i<valuesY->size();i++)
    {
        fprintf(outputFile, "%d_", valuesY->at(i));
    }
    fprintf(outputFile, "\n");
    for (int i=0;i<valuesX->size();i++)
    {
        fprintf(outputFile, "%lf_", valuesX->at(i));
    }
    fclose(outputFile);

    std::cout<<" finish_was_executed_successfully!";

    fopen("output_arrivalsdepartures.txt","w+");

```

```

std::string outputFileName2 = "";
outputFileName2.append(this→myName);
outputFileName2.append("_output_arrivalsdepartures.txt");

FILE* outputFile_arrivalsdepartures = fopen(outputFileName2.c_str(),"w+");
fprintf(outputFile_arrivalsdepartures, "%lf_", valuesX_arrivalsDepartures);
printf("%lf_", valuesX_arrivalsDepartures→at(0));
fprintf(outputFile_arrivalsdepartures, "%d\n", valuesY_arrivalsDepartures);
printf("%d_", valuesY_arrivalsDepartures→at(0));
for (int i=1;i<valuesY_arrivalsDepartures→size();i++)
{
    if ( valuesX_arrivalsDepartures→at(i)==valuesX_arrivalsDepartures→at(0))
    {
        continue;
    }
    fprintf(outputFile_arrivalsdepartures, "%lf_", valuesX_arrivalsDepartures→at(i));
    printf("%lf_", valuesX_arrivalsDepartures→at(i));
    fprintf(outputFile_arrivalsdepartures, "%d\n", valuesY_arrivalsDepartures→at(i));
    printf("%d_", valuesY_arrivalsDepartures→at(i));
}
fclose(outputFile_arrivalsdepartures);
}

```

To αρχείο Server.h

```

#ifndef _PROJECT1_SERVER_H_
#define _PROJECT1_SERVER_H_

#include <omnetpp.h>
#include "packetProcess_m.h"

using namespace omnetpp;

/**

```

```

* TODO – Generated class
*/
class Server : public cSimpleModule
{
protected:
    virtual void initialize ();
    virtual void handleMessage (cMessage *msg);
    virtual void finish ();

    int selectRandomOutput ();

    int totalCountMessages;
    int defaultProcessingTime;
    std::string myName;

    cMessage *lastProcessedMessage=NULLptr;

    std::vector<double> *timeX;
    std::vector<int> *arrivalsY;
};

#endif

```

To αρχείο Server.ned

```

simple Server
{
    parameters:
        @display("i=block/server");
        string myName;
    gates:

        inout fromQueue [];
        inout out [];

```

```
}
```

To αρχείο Server.cc

```
#include "Server.h"
```

```
Define_Module( Server );
```

```
void Server::initialize()
```

```
{
```

```
    timeX = new std::vector<double>();
```

```
    arrivalsY = new std::vector<int>();
```

```
    totalCountMessages=0;
```

```
    defaultProcessingTime=4;
```

```
    myName = par("myName").stringValue();
```

```
    EV<<myName.c_str();
```

```
    cMessage * startMessage = new cMessage("start");
```

```
    scheduleAt(simTime()+0.2,startMessage);
```

```
}
```

```
void Server::handleMessage(cMessage *msg)
```

```
{
```

```
    std::string message = msg->getName();
```

```
    if (message=="start")
```

```
    {
```

```
        scheduleAt(simTime(),new cMessage("nextFromQueue"));
```

```
        delete msg;
```

```
    }
```

```
    else if (message=="nextFromQueue")
```

```
    {
```

```
        if (lastProcessedMessage==nullptr)
```

```
        {
```

```
            bubble("malakia_egine!");
```

```

    }
    else
    {

        if (gateSize("out")==0)
        {
            delete lastProcessedMessage;
            std::string bubbleMessage = "destroyed:_" + std::to_string(lastProcessedMessage->id);
            bubble(bubbleMessage.c_str());
        }
        else
        {
            int randomGateIndex=selectRandomOutput();
            send(lastProcessedMessage, gate("out$o",randomGateIndex));
            std::string bubbleMessage = "forwarded:_" + std::to_string(lastProcessedMessage->id);
            bubble(bubbleMessage.c_str());
        }
        lastProcessedMessage=nullptr;
    }

    send(new cMessage("giveMeNext"), gate("fromQueue$o",0));
    delete msg;
}
else if (message=="process")
{
    totalCountMessages++;
    timeX->push_back(simTime().dbl());
    arrivalsY->push_back(totalCountMessages);

    lastProcessedMessage = msg;

    packetProcess *currentPacket = (packetProcess*)msg;
    double currentMessageProcessingTime=currentPacket->getProcessingTime();

```

```

        scheduleAt(simTime()+currentMessageProcessingTime,new cMessage("
    })
}

```

```

void Server::finish()
{
    std::string outputFileName = "";
    outputFileName.append(this→myName);
    outputFileName.append("_output_arrivals.txt");

    FILE* departuresTime = fopen(outputFileName.c_str(),"w+");
    fprintf(departuresTime, "%lf_",timeX→at(0));
    fprintf(departuresTime, "%d\n",arrivalsY→at(0));
    for (int i=1;i<arrivalsY→size();i++)
    {
        if (timeX→at(i)==timeX→at(i-1) && arrivalsY→at(i)==arrivalsY→at(i-1))
        {
            continue;
        }
        fprintf(departuresTime, "%lf_",timeX→at(i));
        fprintf(departuresTime, "%d\n",arrivalsY→at(i));
    }
    fclose(departuresTime);
}

```

```

int Server::selectRandomOutput()
{
    return intuniform(0,this→gateSize("out")-1);
}

```

To αρχείο packetProcess.msg

```

message packetProcess
{
    int processId;
}

```

```
        double processingTime_total;  
    }
```

Συντμήσεις

CAC	Call Admission Control
DLC	Data Link Layer
FDM	Frequency-Division Multiplexing
M/M/1	Memoryless Single Queue- Single Server system
TDM	Time Division Multiplexing

Γλωσσάρι Ξενικών Όρων

Access Permissions	επιτρεπτή Πρόσβαση
Call Admission Control	Αποδοχή μιας κλήσης μετά από έλεγχο
Design view	Εικόνα σχεδιασμού
Editor	Κειμενογράφος
Facebook	Πλατφόρμα κοινωνικής δικτύωσης
Multiplexing	Πολυπλεξία
Omnet++	Πλατφόρμα ανάπτυξης κώδικα
Processing delay	Καθυστερήση επεξεργασίας
Propagation delay	Καθυστερήση διάδοσης
Project Folder	Ο φάκελος με την εργασία
Queuing delay	Καθυστερήση ουράς
Routing probability	Πιθανότητα μετάβασης

Statistical multiplexing	Στατιστική πολυπλεξία
Source view	Εικόνα κώδικα
Transmission delay	Καθυστέρηση μετάδοσης
Transmission capacity	Χωρητικότητα μετάδοσης
Visualization	Οπτικοποίηση