

**ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ**



ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

ΣΧΟΛΗ
ΕΠΙΣΤΗΜΩΝ &
ΤΕΧΝΟΛΟΓΙΑΣ
ΤΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ
SCHOOL OF
INFORMATION
SCIENCES &
TECHNOLOGY

ΜΕΤΑΠΤΥΧΙΑΚΟ
ΕΠΙΣΤΗΜΗ ΤΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
MSc IN COMPUTER SCIENCE

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΜΣ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Διπλωματική Εργασία

Μεταπτυχιακού Διπλώματος Ειδίκευσης

**«Ανάλυση και Μοντελοποίηση Αλγόριθμων
Προσωρινής Αποθήκευσης Αρχείων με Δυναμική
Δημοφιλία »**

Καρύδης Στυλιανός

EY1415

Επιβλέπων: Σταύρος Τουμπής

Αθήνα, Ιούνιος 2016



Περίληψη

Η χρήση της προσωρινής μνήμης των κόμβων ενός δικτύου αποσκοπεί στη μείωση της κυκλοφορίας στο δίκτυο, του φόρτου των εξυπηρετητών, καθώς και των καθυστερήσεων ανάκτησης περιεχομένου. Ενώ το κλειδί για την αποτελεσματικότητα της χρήσης προσωρινής μνήμης είναι η χρήση της δημοφιλίας των αρχείων, σπάνια αυτή ενσωματώνεται σε αλγόριθμους αντικατάστασης που χρησιμοποιούνται. Αντίθετα, οι πληροφορίες που χρησιμοποιούνται περιλαμβάνουν συνήθως χρονική τοπικότητα και το μέγεθος του περιεχομένου, τα οποία χρησιμοποιούνται για να συναχθούν έμμεσες πληροφορίες δημοτικότητας και να δημιουργήσουν την πολιτική αντικατάστασης της μνήμης.

Στην εργασία αυτή, αρχικά μοντελοποιούμε πώς η δημοφιλία κάθε αρχείου αλλάζει με το χρόνο, και συγκεκριμένα περιγράφουμε την δημοφιλία ενός αρχείου ως μια μονοδιάστατη αλυσίδα Markov συνεχούς χρόνου (Continuous Time Markov Chain – CTMC). Έπειτα, καθώς μας ενδιαφέρει η αντιγραφή του δημοφιλούς περιεχόμενου στις προσωρινές μνήμες, μοντελοποιούμε τον αριθμό των αντιγράφων ενός αρχείου από κοινού με την δημοφιλία του και κατασκευάζουμε μια δισδιάστατη CTMC, για την οποία, υπολογίζοντας τους ρυθμούς μετάβασής της, μπορούμε να υπολογίσουμε την στατική κατανομή της και να πάρουμε αποτελέσματα. Σε σχέση με προηγούμενες εργασίες, η αλυσίδα μοντελοποιεί περιπτώσεις κατά τις οποίες το πλήθος των αντιγράφων ενός αρχείου μεταβάλλεται κατ' απόλυτο τιμή μεγαλύτερη της μονάδας, μετά από ένα μόνο αίτημα για το αρχείο, κάτι που περιπλέκει σημαντικά την ανάλυση.

Στο Κεφάλαιο 1 θα δούμε κάποια αναφορά στις στρατηγικές αποθήκευσης κάτω από δυναμικό περιεχόμενο δημοτικότητας και κυρίως τα τρία βασικά θέματά τους: α) το μοντέλο κίνησης, β) τις πολιτικές αποθήκευσης, γ) τις στρατηγικές αντιγραφής αρχείων. Στο Κεφάλαιο 2 θα αναφερθούμε στη Δικτύωση με Έμφαση στο Περιεχόμενο και στη στρατηγική Περιεχομένου με την Μεγαλύτερη Δημοφιλία. Στο Κεφάλαιο 3 περιγράφουμε το πρόβλημα και στο Κεφάλαιο 4 περιγράφουμε και αναλύουμε κάποια χαρακτηριστικά τριών τοπολογιών δικτύου. Συνεχίζοντας, στο Κεφάλαιο 5 θα δούμε την δημοτικότητα ενός αρχείου στο χρόνο και πώς την ορίζουμε ως μια μονοδιάστατη αλυσίδα Markov. Στο Κεφάλαιο 6 αναλύουμε και μοντελοποιούμε τη σχέση της δημοφιλίας ενός αρχείου με τον αριθμό των αντιγράφων του ανάμεσα στο χρόνο. Τέλος στο Κεφάλαιο 7 παρουσιάζουμε τα αποτελέσματα της αριθμητική επίλυσης του προβλήματος και στο Κεφάλαιο 8 τα συμπεράσματά μας για την παρούσα εργασία.



Ευχαριστίες

Θα ήθελα ειλικρινά να ευχαριστήσω τον επιβλέποντα καθηγητή μου κύριο Τουμπή Σταύρο για την πολύτιμη βοήθεια και καθοδήγηση που μου παρείχε κατά την διάρκεια της διπλωματικής εργασίας στην εφαρμογή των απαιτούμενων πειραματικών διαδικασιών, και στην συγγραφή του κειμένου. Η βοήθεια και οι γνώσεις του αποτέλεσαν ζωτικής σημασίας παράγοντα για την ανάπτυξη και τη μορφοποίηση της παρούσας διατριβής.

Επιπροσθέτως θα ήθελα να ευχαριστήσω τον κύριο Ξηλωμένο Γεώργιο και τον κύριο Σύρη Βασίλειο για τον χρόνο τους, ως μέλη της επιτροπής εξέτασης της διπλωματικής μου εργασίας.



Λίστα Συντομογραφιών και Συμβόλων

Ελληνική Ορολογία	Αγγλική Ορολογία	Συντομογραφία
Άλμα	Hop	
Δίκτυο Διανομής Περιεχομένου	Content Delivery Network	CDN
Πληροφοριοκεντρικό Δίκτυο	Information Centric Network	ICN
Ανεξάρτητο Μοντέλο Αναφοράς	Independent Reference Model	IRM
Μοντέλο Λήψης Θορύβου	Shot Noise Model	SNM
Λιγότερο Πρόσφατα Χρησιμοποιημένο (αρχείο)	Least Recently Used (file)	LRU
Πρώτο Εισέρχεται, Πρώτο Εξέρχεται	First In, First Out	FIFO
Δίκτυο με Έμφαση στο Περιεχόμενο	Content-Centric Networking	CCN
Περιεχόμενο με την Μεγαλύτερη Δημοφιλία	Most Popular Content	MPC
Προχωρημένη Μηχανή RISC	Advanced RISC Machine	ARM
Μαρκοβιανή Αλυσίδα Συνεχούς Χρόνου	Continuous Time Markov Chain	CTMC



Πίνακας περιεχομένων

Περίληψη	2
Ευχαριστίες	3
Λίστα Συντομογραφιών και Συμβόλων	4
Κεφάλαιο 1^ο - Στρατηγικές αποθήκευσης κάτω από δυναμικό περιεχόμενο δημοτικότητας.....	7
1.1 Μοντέλα κίνησης.....	8
1.2 Πολιτικές αποθήκευσης.....	9
1.3 Στρατηγικές αντιγραφής αρχείων.....	10
Κεφάλαιο 2^ο – Δικτύωση με Έμφαση στο Περιεχόμενο και Περιεχόμενο με την Μεγαλύτερη Δημοφιλία.....	11
2.1 Content Centric Networks	11
2.2 Most Popular Content.....	12
Κεφάλαιο 3^ο – Εισαγωγή στο Πρόβλημα	15
Κεφάλαιο 4^ο – Τοπολογίες Δικτύων Μεταγωγής Πακέτων	17
4.1 Τοπολογία Δακτυλίου	18
4.2 Τοπολογία Πλήρους Διασύνδεσης	21
4.3 Τοπολογία τόρου	21
Κεφάλαιο 5^ο – Δημοτικότητα ενός αρχείου μέσα στο χρόνο	24
Κεφάλαιο 6^ο – Σχέση δημοφιλίας με αντίγραφα ενός αρχείου στο δίκτυο.	32
6.1 Το πρόβλημα	32
6.2 Ανάλυση του προβλήματος.....	33
6.2.1 Υπολογισμός πιθανοτήτων του Δέντρου καταστάσεων.....	37
6.2.2 Υπολογισμός πιθανοτήτων ύπαρξης του σημειωμένο αρχείου στους γειτονικούς κόμβους.....	40
6.2.3 Υπολογισμός πιθανοτήτων μετάβασης, που μειώνουν τα αντίγραφα του σημειωμένο αρχείου.....	43
6.2.4 Υπολογισμός ρυθμών μετάβασης δισδιάστατης CTMC	45
Κεφάλαιο 7^ο – Αριθμητική Επίλυση.....	49
7.1 Steady State	49
7.2 Αποτελέσματα Αριθμητικής Επίλυσης	50
Κεφάλαιο 8^ο – Συμπεράσματα και Περαιτέρω Βελτιώσεις	58
8.1 Συμπεράσματα.....	58
8.2 Περαιτέρω Βελτιώσεις	58



Παράρτημα.....	59
Παραπομπές	63



1

Κεφάλαιο 1^ο - Στρατηγικές αποθήκευσης κάτω από δυναμικό περιεχόμενο δημοτικότητας

Η σύγχρονη χρήση του διαδικτύου έχει προσανατολιστεί σε μεγάλο βαθμό στο περιεχόμενο. Δηλαδή, οι χρήστες δεν νοιάζονται για το πού (από ποιόν εξυπηρετητή) και πώς (με ποιο πρωτόκολλο) θα λάβουν ένα κομμάτι περιεχομένου, αλλά ενδιαφέρονται περισσότερο για το πόσο γρήγορα και αξιόπιστα θα το ανακτήσουν. Εν τω μεταξύ, το μέγεθος και οι διαφορετικοί τύποι των περιεχομένων αυξάνονται, όπως και η κίνηση στο διαδίκτυο, και μάλιστα με πρωτοφανή ρυθμό. Αυτή η εκρηκτική αύξηση της κίνησης στο διαδίκτυο αποτελεί μια σημαντική πρόκληση για το υποκείμενο δίκτυο και έχει δημιουργήσει ενδιαφέροντα ερευνητικά προβλήματα για την ερευνητική κοινότητα.

Η αποθήκευση διαδραματίζει σημαντικό ρόλο στην διανομή του περιεχομένου στο διαδίκτυο. Χάρη στον πολλαπλασιασμό των αποθηκευτικών εξυπηρετητών, με το Δίκτυο Διανομής Περιεχομένου (Content Delivery Networks - CDN), οι πάροχοι περιεχομένου μπορούν να εξυπηρετήσουν μεγάλους πληθυσμούς γεωγραφικά [1]. Με την προσωρινή αποθήκευση του περιεχομένου κοντά στους χρήστες, μερικά από τα οφέλη περιλαμβάνουν τη μείωση του κόστους εύρους ζώνης, τη μείωση του χρόνου φόρτωσης της σελίδας, και την αύξηση της παγκόσμιας διαθεσιμότητας των περιεχομένων.

Ακόμα μια ενδιαφέρουσα προσέγγιση υπάρχει με τις αλλαγές που έχουν γίνει στις επικοινωνίες. Για παράδειγμα, στο παρελθόν επικρατούσε η παραδοσιακή επικοινωνία host-to-host, ενώ σήμερα υπάρχει και η επικοινωνία host-to-content. Μάλιστα, αρκετές αρχιτεκτονικές έχουν προταθεί για την βελτίωση της λειτουργίας διαδικτύου, με μία από αυτή να είναι το Πληροφοριοκεντρικό Δίκτυο (Information Centric Network - ICN) [2]. Σε αυτήν την αρχιτεκτονική, η προσωρινή αποθήκευση γίνεται μια λειτουργία διαθέσιμη για κάθε δρομολογητή. Δηλαδή, επιτρέπεται στους δρομολογητές να μετατρέψουν τους ενταμιευτές ουρών που διαθέτουν σε μνήμες για αποθήκευση πακέτων, εκμεταλλευόμενοι έτσι τους υπάρχοντες αποθηκευτικούς πόρους του δικτύου.

Η έρευνα και η ανάλυση των συστημάτων που κάνουν χρήση αυτής της ενταμίευσης αφορά τρία βασικά θέματα:



- I. Το μοντέλο κίνησης.
- II. Τις πολιτικές αποθήκευσης.
- III. Τις στρατηγικές αντιγραφής αρχείων.

1.1 Μοντέλα κίνησης

Αρχικά, θα αναφέρουμε το Ανεξάρτητο Μοντέλο Αναφοράς (Independent Reference Model - IRM), που είναι μια τυπική προσέγγιση που χρησιμοποιείται στην ανάλυση του συστήματος αποθήκευσης και χαρακτηρίζει τον τρόπο που γίνονται οι αιτήσεις ενός αντικειμένου που φθάνουν σε μια προσωρινή μνήμη. Σύμφωνα με αυτό το μοντέλο, οι αναφορές σε αποθηκευμένα αντικείμενα είναι ανεξάρτητες τυχαίες μεταβλητές. Το IRM βασίζεται στις ακόλουθες παραδοχές

- Οι χρήστες πρέπει να ζητούν αρχεία από ένα σταθερό κατάλογο με M αρχεία,
- η διαδικασία των αιτημάτων ενός δεδομένου αρχείου m μοντελοποιείται με μια ομοιογενή διαδικασία Poisson έντασης $\lambda_m = \lambda p_m$.

Το IRM συνήθως χρησιμοποιείται σε συνδυασμό με τον νόμο πιθανότητας Zipf δηλαδή μια κατανομή που παρατηρήθηκε να ισχύει από μετρήσεις κυκλοφορίας και χρησιμοποιήθηκε σε πολλές μελέτες αξιολόγησης [3],[4]. Εξ ορισμού, το IRM αγνοεί πλήρως όλες τις χρονικές συσχετίσεις στην ακολουθία των αιτημάτων. Συγκεκριμένα, δεν λαμβάνει υπόψιν της ένα βασικό χαρακτηριστικό, που είναι η χρονική τοπικότητα. Δηλαδή, το γεγονός ότι αν ένα αντικείμενο ζητείται σε μια δεδομένη χρονική στιγμή, τότε είναι το πιο πιθανό, το ίδιο αντικείμενο να ζητηθεί πάλι στο άμεσο μέλλον. Είναι γνωστό ότι η χρονική τοπικότητα έχει θετικές επιδράσεις στην επίδοση της αποθήκευσης (αυξάνει την πιθανότητα το ζητούμενο αρχείο να είναι διαθέσιμο) [5].

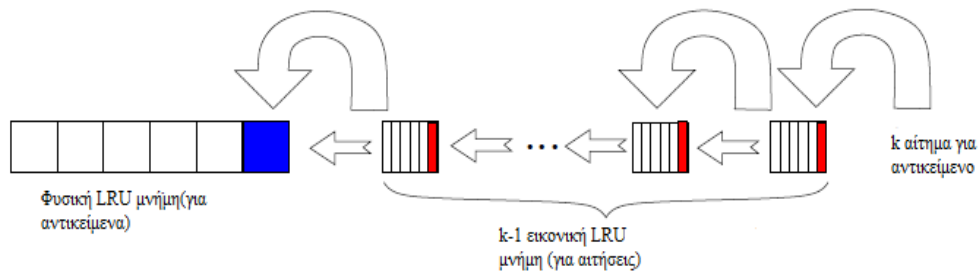
Πρόσφατα ένα νέο μοντέλο κίνησης, που ονομάζεται Μοντέλο Λήψης Θορύβου (Shot Noise Model - SNM), έχει προταθεί ως μια εναλλακτική λύση στα παραδοσιακά μοντέλα κίνησης. Η βασική ιδέα του SNM είναι να παρουσιάσει τη συνολική διαδικασία αιτήσεων ως μια υπέρθεση πολλών ανεξαρτητών διαδικασιών, καθεμία εκ των οποίων αναφέρεται σε ξεχωριστό περιεχόμενο. Συγκεκριμένα, η διαδικασία άφιξης των αιτήσεων για ένα δεδομένο περιεχόμενο σε μια μνήμη, περιγράφεται από μια ανομοιογενή διαδικασία Poisson. Το SMN έχει δείχτει στο [6] ότι παρέχει μια απλή, ευέλικτη και ακριβή περιγραφή της χρονικής και γεωγραφικής τοποθεσίας όπου γίνονται αιτήματα για αρχεία Video-on-Demand. Ένα ενδιαφέρον εύρημα στο [6] είναι ότι η συγκεκριμένη μορφή του «Προφίλ Δημοτικότητας» έχει πολύ μικρή επίδραση στην επίδοση της αποθήκευσης, η οποία ουσιαστικά εξαρτάται μόνο από τη μέση διάρκεια ζωής του περιεχομένου.



1.2 Πολιτικές αποθήκευσης

Πολλές είναι οι στρατηγικές που ελέγχουν την συμπεριφορά ενός κόμβου που αποθηκεύει αρχεία, με τις πιο σημαντικές από αυτές να είναι οι εξής:

- ❖ **Least Recently Used (LRU).** Κατά την άφιξη ενός αιτήματος, το αντικείμενο που δεν είναι αποθηκευμένο στην μνήμη, εισάγεται σε αυτή. Εάν η μνήμη είναι πλήρης, για να δημιουργηθεί χώρος, ώστε να εισέλθει το νέο αντικείμενο, το λιγότερο πρόσφατα χρησιμοποιημένο (Least Recently Used - LRU) αρχείο πετιέται από την μνήμη. Δηλαδή, πετιέται από τη μνήμη το αρχείο που δεν έχει ζητηθεί για μεγάλο χρονικό διάστημα. Αυτός ο αλγόριθμος απαιτεί την παρακολούθηση του πότε τα αρχεία που είναι αποθηκευμένα στη μνήμη χρησιμοποιήθηκαν, η οποία κοστίζει.
- ❖ **q -LRU.** Η διαφορά της στρατηγικής αυτής από την LRU είναι στην πολιτική της εισαγωγής του αρχείου. Δηλαδή, κατά την άφιξη ενός αιτήματος, το αντικείμενο που δεν είναι αποθηκευμένο στην μνήμη, εισάγεται σε αυτήν με πιθανότητα q .
- ❖ **Random.** Διαφέρει από την LRU στην πολιτική της έξωσης. επιλέγεται ένα αρχείο τυχαία και πετιέται για να δημιουργήσει τον απαραίτητο χώρο. Ο αλγόριθμος αυτός δεν απαιτεί τη διατήρηση οποιασδήποτε πληροφορίας σχετικά με το ιστορικό της πρόσβασης. Λόγω της απλότητάς του, είχε χρησιμοποιηθεί σε επεξεργαστές ARM.
- ❖ **FIFO (First In, First Out).** Η διαφορά της στρατηγικής από την στρατηγική LRU είναι στην πολιτική έξωσης. Για να δημιουργηθεί χώρος, ώστε να αποθηκευτεί το νέο αρχείο, το αρχείο το οποίο έχει το μεγαλύτερο χρόνο παραμονής στη μνήμη αποβάλλεται.
- ❖ **k -LRU.** Αυτή η στρατηγική προσφέρει μια έξυπνη πολιτική εισαγωγής ακολουθώντας την εξής ιδέα: πριν φθάσει ένα αρχείο στη φυσική μνήμη, όπου θα αποθηκευτεί πραγματικά, κατατάσσεται σε k αιτήσεις, οι οποίες πρέπει να προχωρήσουν ανάμεσα από μια αλυσίδα με $k-1$ εικονικές μνήμες τοποθετημένες μπροστά του, ενεργώντας ως φίλτρα, στις οποίες αποθηκεύονται μόνο οι δείκτες του αντικειμένου, εκτελώντας την αποθηκευτική λειτουργία πάνω σε αυτές (Σχήμα 1.1). Ειδικότερα, η αλυσίδα αυτή θέλει να ξεχωρίσει και να προωθήσει για αποθήκευση στη φυσική μνήμη μόνο τα δημοφιλή αρχεία. Έτσι, καθώς καταφθάνουν τα αιτήματα, φιλτράρονται στις εικονικές μνήμες, κρατώντας τα δημοφιλή και αποβάλλοντας τα μη δημοφιλή αρχεία με την LRU στρατηγική (μπορεί να εκτελεστεί και στρατηγική FIFO στις εικονικές μνήμες για την επιλογή των δημοφιλή αρχείων). Δηλαδή, όταν ένα αρχείο είναι δημοφιλές, τα αιτήματα για αυτό το αρχείο θα καταφθάνουν συνεχώς και έτσι οι αιτήσεις του συγκεκριμένου αρχείου θα χρησιμοποιούνται αρκετά, με συνέπεια να παραμένουν στις εικονικές μνήμες και να μην αποβάλλονται από την LRU στρατηγική της μνήμης. Άρα μόνο οι αιτήσεις για αντικείμενα, των οποίων η ταυτότητα βρίσκεται στην εικονική μνήμη διαβιβάζεται στην φυσική μνήμη.



Σχήμα 1.1 – Πολιτική k-LRU

Παρατηρούμε ότι η LRU έχει υιοθετηθεί ευρέως, διότι έχει καλή απόδοση και είναι απλή στη χρήση.

1.3 Στρατηγικές αντιγραφής αρχείων

Σε ένα σύστημα διασυνδεδεμένων μνημών, θα πρέπει να καθορίσουμε τι γίνεται κατά μήκος της διαδρομής που διανύει το αίτημα, αφού το αίτημα τελικά φτάσει στο στόχο του, δηλαδή στον κόμβο που έχει το πακέτο. Κάποιοι από τους μηχανισμούς είναι οι ακόλουθοι:

- ❖ Αφήνω – αντίγραφο – παντού [leave-copy-everywhere (LCE)]. Το αντικείμενο τοποθετείται σε όλες τις προσωρινές μνήμες προς τα πίσω της διαδρομής.
- ❖ Αφήνω – αντίγραφο – με πιθανότητα [leave-copy-probabilistically (LCP)]. Το αντικείμενο τοποθετείται με πιθανότητα q σε κάθε προσωρινή μνήμη προς τα πίσω της διαδρομής.
- ❖ Αφήνω – αντίγραφο – κάτω [leave-copy-down (LCD)]. Το αντικείμενο αποστέλλεται μόνο στη προσωρινή μνήμη που προηγείται εκείνου στο οποίο βρίσκεται.

Μια σημαντική ιδιότητα στους παραπάνω μηχανισμούς είναι η εξής: Αν συνδυάσουμε την LCP με την LRU από τις πολιτικές αποθήκευσης σε όλες τις μνήμες, παίρνουμε ένα σύστημα προσωρινής αποθήκευσης, ανάλογο με εκείνο στο οποίο έχουμε υιοθετήσει την στρατηγική αντιγραφής LCE σε συνδυασμό με τη στρατηγική q -LRU σε όλες τις μνήμες.



2

Κεφάλαιο 2^ο – Δικτύωση με Έμφαση στο Περιεχόμενο και Περιεχόμενο με την Μεγαλύτερη Δημοφιλία

Το Διαδίκτυο χρησιμοποιείται σήμερα ως επί τον πλείστον για την πρόσβαση σε περιεχόμενο. Πράγματι, τη δεκαετία του 2000, η peer-to-peer (P2P) κυκλοφορία για ανταλλαγή αρχείων υπολογιζόταν γύρω στο 80% της συνολικής κίνησης του διαδικτύου. Σήμερα, οι video streaming εφαρμογές, όπως το YouTube, αποτελούν το πιο σημαντικό μέρος της κίνησης του Διαδικτύου. Αναμένεται ότι το άθροισμα όλων των μορφών βίντεο (TV, VoD (Video on Demand), P2P) θα είναι περίπου το 85% της παγκόσμιας κίνησης το 2020 [7].

2.1 Content Centric Networks

Ενώ το Διαδίκτυο έχει σχεδιαστεί και εξακολουθεί να επικεντρώνεται σε host-to-host επικοινωνίες, οι περισσότεροι χρήστες ενδιαφέρονται πλέον μόνο για το περιεχόμενο και όχι την τοποθεσία προέλευσής του. Έτσι, έχουν οριστεί διάφορες αρχιτεκτονικές που δίνουν προτεραιότητα στην αποτελεσματική διανομή περιεχομένου. Ανάμεσα σε αυτές είναι και η Δικτύωση με Έμφαση στο Περιεχόμενο (Content Centric Networking - CCN). Η CCN είναι μια αρχιτεκτονική δικτύου που βασίζεται σε ονομαστικά δεδομένα, έτσι ώστε για ένα πακέτο περιεχομένου η διεύθυνσή του είναι το όνομά του και δεν εκφράζει την θέση όπου αυτό βρίσκεται. Στην αρχιτεκτονική CCN, το περιεχόμενο δεν ανακτάται από κάποιο ειδικό εξυπηρετητή, όπως γίνεται στο σημερινό διαδίκτυο, αλλά η παράδοση του περιεχομένου γίνεται με την ενσωμάτωση και αποθήκευση χαρακτηριστικών του περιεχομένου ανά κόμβο, κατά το χρόνο που το περιεχόμενο διασχίζει το διαδίκτυο. Έτσι, το περιεχόμενο αντιγράφεται και βρίσκεται σε διαφορετικά σημεία του δικτύου, αυξάνοντας τη διαθεσιμότητα του σε νέες εισερχόμενες αιτήσεις.

Ειδικότερα η CCN αρχιτεκτονική βασίζεται σε δύο αρχέτυπα μηνύματα: Ενδιαφέροντος και Δεδομένων. Ο ενδιαφερόμενος ζητάει ένα περιεχόμενο με την αποστολή ενός Μηνύματος Ενδιαφέροντος στο διαδίκτυο. Όποιος κόμβος άκουσε το αίτημα και έχει τα δεδομένα, μπορεί να απαντήσει με ένα Μήνυμα Δεδομένων. Το περιεχόμενο επομένως διαβιβάζεται στον ενδιαφερόμενο και κάθε κόμβος, όπου



πέρασε το περιεχόμενο, μπορεί να αποθηκεύσει προσωρινά τα δεδομένα. Η CCN στρατηγική αποθήκευσης αποθηκεύει πάντα περιεχόμενο σε όλους τους κόμβους στο μονοπάτι παράδοσης (Always strategy). Αυτή η στρατηγική έχει δείξει τα καλύτερα αποτελέσματα σε σύγκριση με άλλες [12]. Πάντως, αποθηκεύοντας λιγότερο, έχοντας δηλαδή αποθηκεύσει περιεχόμενο σε ένα υποσύνολο κόμβων κατά μήκος του μονοπατιού παράδοσης, επιτυγχάνονται επίσης παρόμοιες αποθηκευτικές επιδόσεις. Όμως στην στρατηγική Always strategy μπορεί να οδηγηθούμε στην αντικατάσταση δημοφιλούς περιεχομένου από μη δημοφιλές, ενώ αποθηκεύοντας περιεχόμενο σε ένα υποσύνολο κόμβων, η περίπτωση αυτή ελαχιστοποιείται. Υποστηρίζεται ότι η αποθήκευση μόνο δημοφιλούς περιεχομένου θα επιφέρει υψηλές αποδόσεις και εξοικονόμηση πόρων ταυτόχρονα [17].

Ένα σημαντικό χαρακτηριστικό της αρχιτεκτονικής CCN, είναι ότι μπορεί να διαχειριστεί τη προσωρινή μνήμη των κόμβων χρησιμοποιώντας στρατηγικές αποθήκευσης και πολιτικές αντικατάστασης, με τις οποίες πρέπει να αποφασίσει, στην περίπτωση που η μνήμη είναι πλήρης, ποιο στοιχείο πρέπει να αντικατασταθεί. Στο πλαίσιο της αρχιτεκτονικής CCN, υπήρξαν αρκετές μελέτες για το πρόβλημα της προσωρινής αποθήκευσης περιεχομένου για καλύτερη απόδοση και αποτελεσματική χρήση των πόρων [8,9,10]. Στο [8] οι E. Rosensweig et al. παρέχουν ένα προσεγγιστικό μοντέλο για την ανάλυση της απόδοσης των CCNs, όπου το περιεχόμενο αποθηκεύεται προσωρινά σε κάθε κόμβο κατά μήκος της διαδρομής της παράδοσης του αιτήματος του περιεχομένου στον προορισμό. Στο [9] οι I. Ψαράς et al. εξετάζουν την μοντελοποίηση και την αξιολόγηση των πολιτικών αποθήκευσης, βασισμένοι σε Μαρκοβιανές αλυσίδες. Οι Rossi και Rossini προτείνουν μια λύση για την προσωρινή αποθήκευση, κατανέμοντας ξεχωριστούς CCN δρομολογητές και χρησιμοποιώντας μετρικές κεντρικότητας, όπως betweenness, εγγύτητα και ο βαθμός κεντρικότητας [10]. Άλλη μελέτη δείχνει ότι μια τεράστια προσωρινή μνήμη θα ήταν αναγκαία για την επίτευξη υψηλών αποθηκευτικών επιδόσεων [11]. Συνεπώς, είναι απαραίτητο να σχεδιάζονται αποτελεσματικές στρατηγικές αποθήκευσης προσαρμοσμένες στα δίκτυα CCN. Μια τέτοια στρατηγική είναι η Στρατηγική του Περιεχομένου με την Μεγαλύτερη δημοφιλία (Most Popular Content - MPC) [16], που θα δούμε στη συνέχεια αναλυτικότερα.

2.2 Most Popular Content

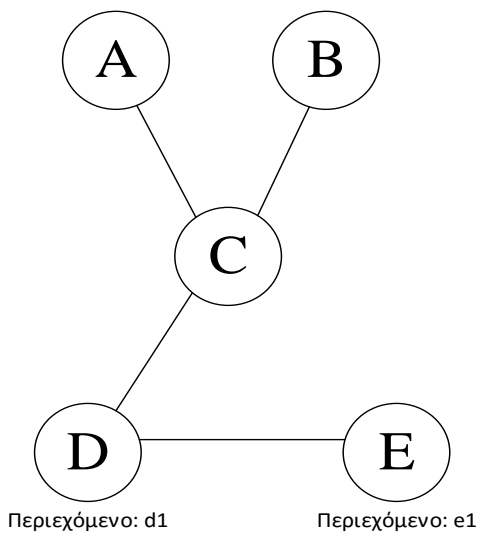
Η MPC είναι μια νέα στρατηγική αποθήκευσης για δίκτυα CCN, όπου οι κόμβοι αποθηκεύουν προσωρινά μόνο δημοφιλές περιεχόμενο. Στην MPC, κάθε κόμβος υπολογίζει τοπικά τους αριθμούς των αιτήσεων για κάθε όνομα του περιεχομένου και αποθηκεύει το ζεύγος (Όνομα Περιεχομένου – Μέτρηση Δημοτικότητας) σε ένα πίνακα δημοτικότητας. Όταν ένα όνομα περιεχομένου φθάσει τοπικά σε ένα κατώφλι δημοτικότητας, στο συγκεκριμένο περιεχόμενο εισάγεται μια ετικέτα δηλώνοντάς το ως δημοφιλές και αν ο κόμβος κατέχει το περιεχόμενο αυτό, συστήνει στους



γειτονικούς κόμβους να αποθηκεύσουν το περιεχόμενο αυτό, στέλνοντας ένα μήνυμα πρότασης. Αυτά τα μηνύματα πρότασης γίνονται αποδεχτά ή όχι, σύμφωνα με τις τοπικές πολιτικές και την διαθεσιμότητα των πόρων.

Δεδομένου ότι η δημοτικότητα του περιεχομένου μπορεί να μειώνεται με το χρόνο μετά το μήνυμα πρότασης, η μέτρηση δημοτικότητας επαναρχικοποιείται σύμφωνα με μια αρχική τιμή, προκειμένου να αποφευχθούν οι πλημμύρες που στέλνουν το ίδιο περιεχόμενο σε πολλούς γείτονες. Ακόμα, εκτός από διαθέσιμο χώρο στη μνήμη που απαιτείται για τη CCN, η MPC χρειάζεται έναν επιπλέον χώρο για να αποθηκεύσει τον πίνακα δημοτικότητας.

(a) Αρχική κατάσταση



Υπόθεση

- A στέλνει αιτήματα για το περιεχόμενο e1
- B στέλνει αιτήματα για το περιεχόμενο d1
- A στέλνει αιτήματα για το περιεχόμενο d1
- C στέλνει αιτήματα για το περιεχόμενο d1

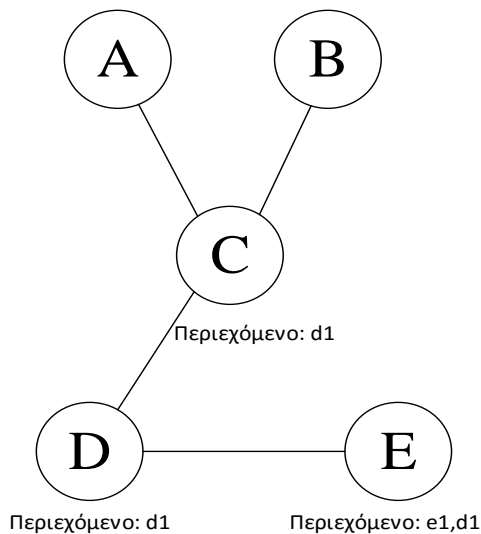
Παρατήρηση

- Η δημοφιλία του e1 στο D: 1
- Η δημοφιλία του d1 στο D: 3

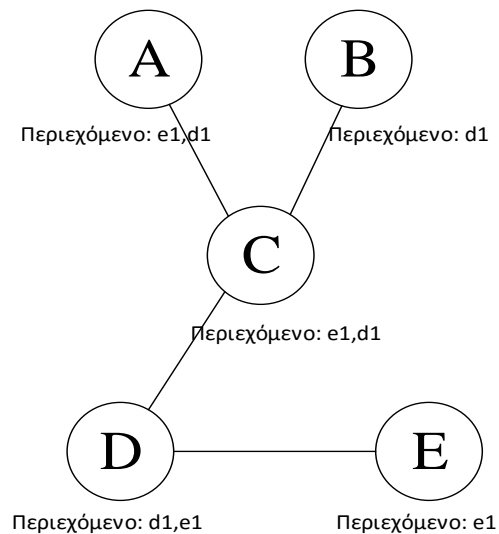
Δραστηριότητα

- D εντοπίζει ότι το περιεχόμενο d1 είναι δημοφιλές
- D ρωτάει τους γείτονες C και E να αποθηκεύσουν το d1 (Τους προτείνει)

(b) Most Popular Caching στρατηγική



(c) CCN – Always Strategy



Σχήμα 2.1

Παράδειγμα MPC και CCN (Always Strategy)(Πηγή [16]).



Στο παραπάνω παράδειγμα παρουσιάζεται πως λειτουργεί η MPC στρατηγική εν συγκρίσει με την στρατηγική CCN (Always Strategy). Αρχικά θεωρούμε ένα σενάριο στο οποίο τρεις κόμβοι (A,B,C) ζητάνε το περιεχόμενο d_1 και ένας κόμβος (ο A) ζητάει το περιεχόμενο e_1 (Σχήμα 2.1.a). Χρησιμοποιώντας την MPC στρατηγική, όταν ο κόμβος A στείλει ένα Μήνυμα Ενδιαφέροντος για το περιεχόμενο e_1 , ο πίνακας δημοφιλίας του κάθε κόμβου που βρίσκεται στην διαδρομή, θα αυξήσει τον αριθμό της δημοφιλίας για το περιεχόμενο e_1 κατά 1. Δηλαδή, η δημοφιλία του e_1 στους κόμβους A,C,D θα είναι 1. Αντίστοιχα, όταν ο κόμβος B στείλει ένα Μήνυμα Ενδιαφέροντος για το περιεχόμενο d_1 , η δημοφιλία του d_1 στους κόμβους D,C θα είναι 1. Όμοια, το Μήνυμα Ενδιαφέροντος για το περιεχόμενο d_1 από τον κόμβο A, θα αυξήσει κατά 1 την δημοφιλία του d_1 στους κόμβους D,C, και τέλος το Μήνυμα Ενδιαφέροντος για το περιεχόμενο d_1 από τον κόμβο C, θα αυξήσει την δημοφιλία του d_1 στον κόμβο D κατά 1. Άρα, παρατηρώντας τους πίνακες δημοφιλίας, βλέπουμε ότι στον κόμβο D, η δημοφιλία του e_1 είναι 1 και η δημοφιλία του d_1 είναι 3. Επομένως με την MPC στρατηγική το περιεχόμενο d_1 χαρακτηρίζεται δημοφιλές περιεχόμενο και ο κόμβος D στέλνει Μήνυμα Πρότασης στους γειτονικούς κόμβους να αποθηκεύσουν το συγκεκριμένο περιεχόμενο, το οποίο δέχονται (Σχήμα 2.1.b).

Αντίθετα, χρησιμοποιώντας την CCN (Always Strategy) στρατηγική, όταν ο κόμβος A στείλει ένα Μήνυμα Ενδιαφέροντος για το περιεχόμενο e_1 , κάθε κόμβος της διαδρομής αποθηκεύει το περιεχόμενο. Δηλαδή, οι κόμβοι A,C,D αποθηκεύουν ένα αντίγραφο του περιεχομένου e_1 . Αντίστοιχα ένα αντίγραφο του περιεχομένου d_1 θα αποθηκευτεί στους κόμβους C,D,B όταν ζητηθεί από τον κόμβο B, στους κόμβους C,A όταν ζητηθεί από τον κόμβο A και στο κόμβο C όταν ζητηθεί από τον κόμβο C (Σχήμα 2.1.c).

Συμπερασματικά, παρατηρούμε ότι με την MPC στρατηγική έχουμε δύο μόνο αντίγραφα του περιεχομένου d_1 , ενώ με την CCN (Always Strategy) στρατηγική έχουμε τρία αντίγραφα του d_1 και τρία αντίγραφα του e_1 . Δηλαδή με την MPC στρατηγική επιτυγχάνεται εξοικονόμηση πόρων στην μνήμη.



3

Κεφάλαιο 3^ο – Εισαγωγή στο Πρόβλημα

Θεωρούμε ένα δίκτυο που αποτελείται από N κόμβους, καθένας εκ των οποίων έχει μια μνήμη που μπορεί να αποθηκεύσει K αρχεία. Οι κόμβοι ζητούν αρχεία τα οποία χρειάζονται οι χρήστες τους. Τα δημοφιλέστερα αρχεία θα πρέπει να αποθηκεύονται στην μνήμη περισσοτέρων του ενός κόμβου, ώστε να μειωθεί η κίνηση στο δίκτυο. Ωστόσο, διατηρώντας πολλαπλά αντίγραφα ενός αρχείου, μεγαλώνει το αποθηκευτικό κόστος. Θα θέλαμε να μελετήσουμε την αντιστάθμιση που προκύπτει, χρησιμοποιώντας αναλυτικά εργαλεία (ειδικότερα αλυσίδες Markov) και αριθμητικές επιλύσεις.

Μια διαδικασία (ή αλυσίδα) Markov είναι μια τυχαία διαδικασία, κατά την οποία το επόμενο βήμα εξαρτάται μόνο από την παρούσα κατάσταση, επομένως, υπό αυτή την έννοια, η διαδικασία δεν έχει καμία μνήμη. Ειδικότερα, θα χρησιμοποιήσουμε τη συνεχούς χρόνου αλυσίδα Markov (Continuous Time Markov Chain - CTMC), που είναι ένα μαθηματικό μοντέλο, που παίρνει τιμές σε κάποιο πεπερασμένο χώρο καταστάσεων και για τις οποίες ο χρόνος που δαπανείται σε κάθε κατάσταση παίρνει μη αρνητικές τυχαίες πραγματικές τιμές και έχει εκθετική κατανομή.

Αρχικά, θα αναφερθούμε σε κάποιες τοπολογίες δικτύων μεταγωγής πακέτων (δακτυλίου, πλήρους διασύνδεσης, τόρος), για να δούμε τον μέσο αριθμό συνδέσμων που πρέπει να διανύσει ένα πακέτο και τις οποίες θα χρησιμοποιήσουμε αργότερα, για να μελετήσουμε την από κοινού κατανομή της δημοτικότητας ενός αρχείου και του πλήθους των αντιγράφων του.

Χρησιμοποιούμε αλυσίδες Markov σε συνδυασμό με την κατανομή Zipf για να μελετήσουμε την εξέλιξη της δημοφιλίας ενός αρχείου. Συγκεκριμένα, θεωρούμε ότι η δημοφιλία ενός αρχείου περιγράφεται από μια μονοδιάστατη CTMC με τον παρακάτω τρόπο. Θεωρούμε ότι υπάρχουν X_i με $i=1, \dots, M$ συνολικά αρχεία, με μειούμενη δημοφιλία καθώς αυξάνει ο δείκτης τους, και η κατανομή της δημοφιλίας τους δίνεται



από την Zipf κατανομή με κάποια παράμετρο α . Επιπλέον, η δημοφιλία κάθε αρχείου αλλάζει με το χρόνο. Συγκεκριμένα, η αλλαγή της σειράς κατάταξης της δημοφιλίας των αρχείων γίνεται με μια διαδικασία Poisson με ρυθμό λ . Όταν συμβαίνει μια άφιξη της άνω διαδικασίας, ένα αρχείο επιλέγεται τυχαία να γίνει το δημοφιλέστερο. Επομένως, τα αρχεία που ήταν πιο δημοφιλή από το επιλεγμένο χάνουν μία θέση στη σειρά κατάταξης, ενώ τα αρχεία μετά από αυτό παραμένουν ως έχουν στη σειρά. Κατόπιν, καθώς μας ενδιαφέρει η αντιγραφή του δημοφιλούς περιεχόμενου στις προσωρινές μνήμες, μοντελοποιούμε τον αριθμό των αντιγράφων ενός αρχείου από κοινού με την δημοφιλία του και κατασκευάζουμε μια δισδιάστατη CTMC.

Μια αρχική προσέγγιση του προβλήματος έχει γίνει από τον Τουμπή Σταύρο και Κατσιόλη Αθανάσιο [15]. Το δικό μας μοντέλο διαφέρει στο ότι επιτρέπονται μεταβάσεις στην δισδιάστατη αλυσίδα τέτοιες ώστε το πλήθος των αντιγράφων ενός αρχείου μεταβάλλεται κατ' απόλυτη τιμή μεγαλύτερη της μονάδας, κάτι που περιπλέκει αισθητά την ανάλυση.



4

Κεφάλαιο 4^ο – Τοπολογίες Δικτύων Μεταγωγής Πακέτων

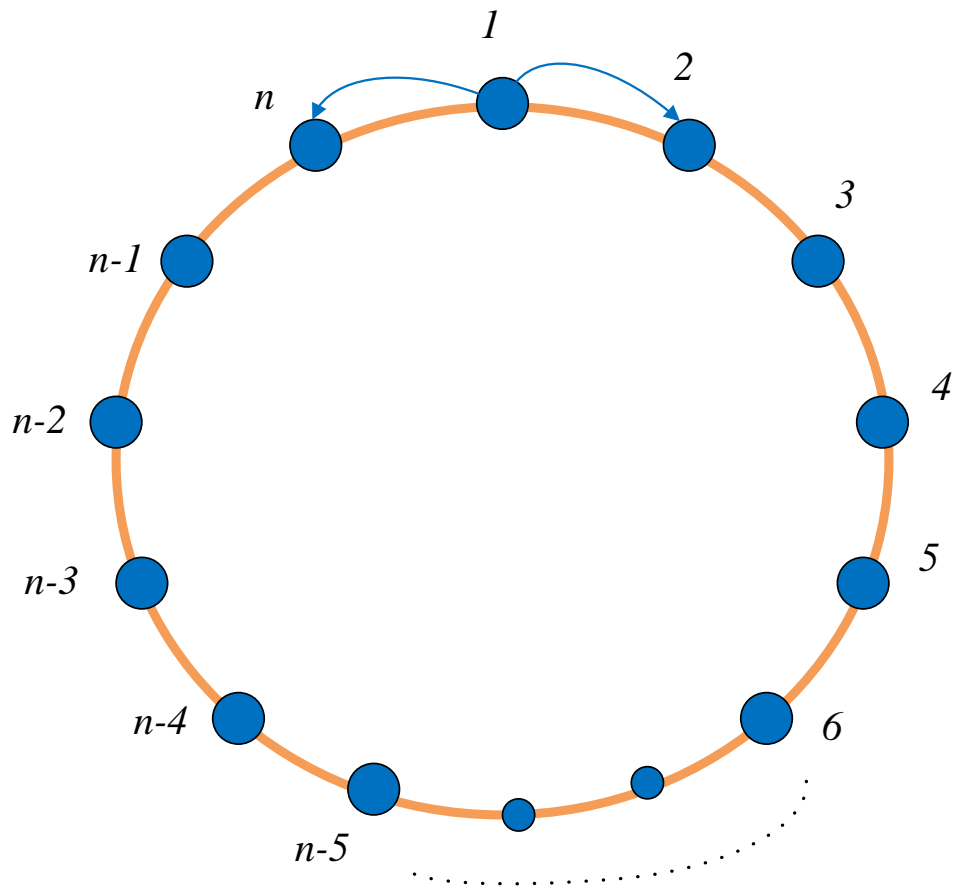
Στο κεφάλαιο αυτό θα δούμε την τοπολογία τριών δικτύων μεταγωγής πακέτων, τα οποία αποτελούνται από n κόμβους ($n \geq 3$). Το πρώτο δίκτυο έχει τοπολογία δακτυλίου (διπλής κατεύθυνσης), το δεύτερο τοπολογία πλήρους διασύνδεσης με ένα φυσικό σύνδεσμο από κάθε κόμβο προς κάθε άλλο, και το τρίτο τοπολογία τόρου.

Για κάθε τοπολογία υπολογίζουμε το μέγιστο, ελάχιστο και μέσο αριθμό συνδέσμων (hops), που θα διανύσει ένα πακέτο για να φτάσει από μια τυχαία πηγή σε ένα τυχαίο προορισμό. Θεωρούμε ότι όλα τα ζεύγη κόμβων πηγής και προορισμού είναι ισοπίθανα, καθώς και ότι ένα πακέτο είναι δυνατό να μεταδίδεται μεταξύ τερματικών συσκευών, που είναι συνδεδεμένες στον ίδιο κόμβο (δηλαδή ο κόμβος πηγής είναι και κόμβος προορισμού).

Αρχικά, θα ορίσουμε κάποιες μεταβλητές, οι οποίες θα χρησιμοποιηθούν παρακάτω. Το d είναι ο αριθμός των αλμάτων που μπορεί να κάνει ένα πακέτο για την μεταφορά του από έναν κόμβο σε έναν άλλο, δηλαδή μας δείχνει τον αριθμό των κόμβων από τους οποίους θα περάσει ταξιδεύοντας από την πηγή στον προορισμό, συμπεριλαμβανομένου του προορισμού. Θεωρούμε πως πάντα η διαδρομή θα είναι το ελάχιστο μονοπάτι και ορίζουμε ως $N(d)$ τον αριθμό των κόμβων στους οποίους μπορεί να αποστείλει το πακέτο ο κόμβος πηγής κάνοντας ακριβώς d άλματα κατά μήκος μιας διαδρομής ελάχιστου μονοπατιού. Ως D ορίζουμε το μέσο αριθμό των συνδέσμων που χωρίζει δύο τυχαία επιλεγμένους κόμβους.



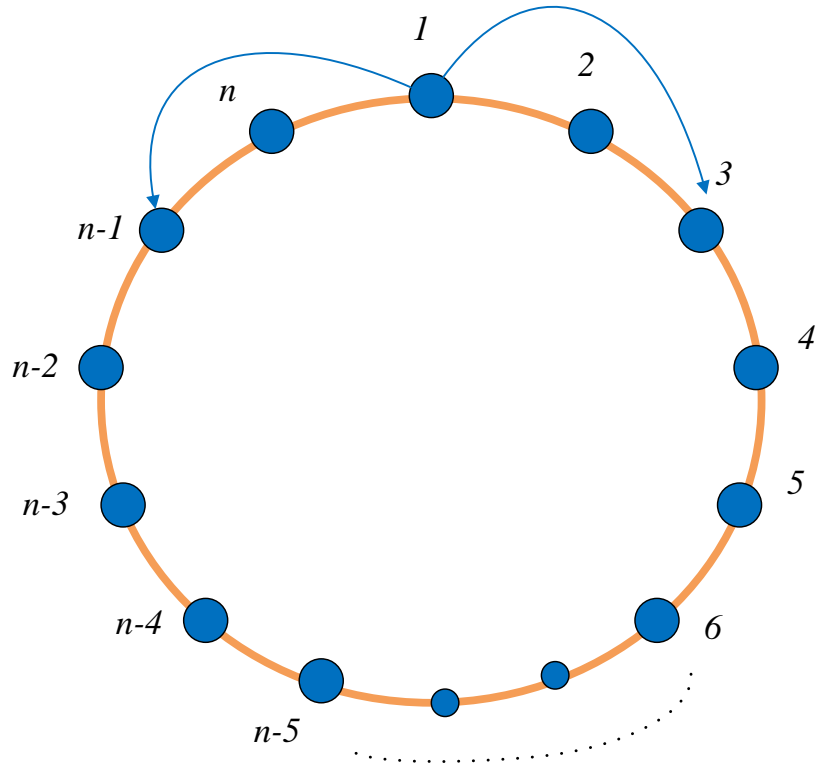
4.1 Τοπολογία Δακτυλίου



Σχήμα 4.1.α: Τοπολογία Δακτυλίου n κόμβων με $d=1$.

Εδώ το πακέτο μπορεί να πάει είτε δεξιά είτε αριστερά. Επομένως,

$$N(d) = 2 \quad \forall d = 1, \dots, n-1.$$



Σχήμα 4.1.β: Τοπολογία Δακτυλίου n κόμβων με $d=2$.

Για τον υπολογισμό του μέγιστου πλήθους διανυθέντων συνδέσμων, η χειρότερη περίπτωση είναι το πακέτο να διανύσει το μισό δακτύλιο, άρα $\frac{n}{2}$ συνδέσμους. Το ελάχιστο θα είναι ίσο με το μηδέν. Για τον υπολογισμό του μέσου πλήθους διανυθέντων συνδέσμων, πρέπει να διακρίνουμε δύο περιπτώσεις:

- Αν το n είναι άρτιος ($n=2k$, $k \in \mathbb{Z}$),
 - τότε με πιθανότητα $\frac{1}{n}$ το πακέτο θα διανύσει 0 συνδέσμους (το τερματικό προορισμού θα είναι στον ίδιο κόμβο με το τερματικό πηγής),
 - με πιθανότητα επίσης $\frac{1}{n}$ θα διανύσει $\frac{n}{2}$ συνδέσμους (θα κατευθύνεται στο διαμετρικά αντίθετο κόμβο),
 - ενώ θα διανύσει από 1 ως $\frac{n}{2}-1$ συνδέσμους με πιθανότητα $\frac{1}{n}$ δεξιόστροφα και $\frac{1}{n}$ αριστερόστροφα (προς τους υπόλοιπους κόμβους). Επομένως,



$$D = 0 \frac{1}{n} + \left(1 + 2 + \dots + \frac{n}{2} - 1\right) \frac{2}{n} + \frac{n-1}{2n} = \frac{\frac{n}{2} \left(\frac{n}{2} - 1\right)}{2} \frac{2}{n} + \frac{1}{2} = \frac{n}{4}.$$

- Αν το n είναι περιττός ($n=2k+1$, $k \in \mathbb{Z}$),
 - τότε με πιθανότητα $\frac{1}{n}$ το πακέτο θα διανύσει 0 συνδέσμους (το τερματικό προορισμού θα είναι στον ίδιο κόμβο με το τερματικό πηγής),
 - ενώ θα διανύσει από 1 ως $[n/2]^1 = k = \frac{n-1}{2}$ συνδέσμους με πιθανότητα $\frac{1}{n}$ δεξιόστροφα και $\frac{1}{n}$ αριστερόστροφα. Οπότε

$$D = 0 \frac{1}{n} + \left(1 + 2 + \dots + \frac{n-1}{2}\right) \frac{2}{n} = \frac{\frac{n-1}{2} \left(\frac{n-1}{2} + 1\right)}{2} \frac{2}{n} = \frac{n^2 - 1}{4n} \quad \text{και} \quad \text{το}$$

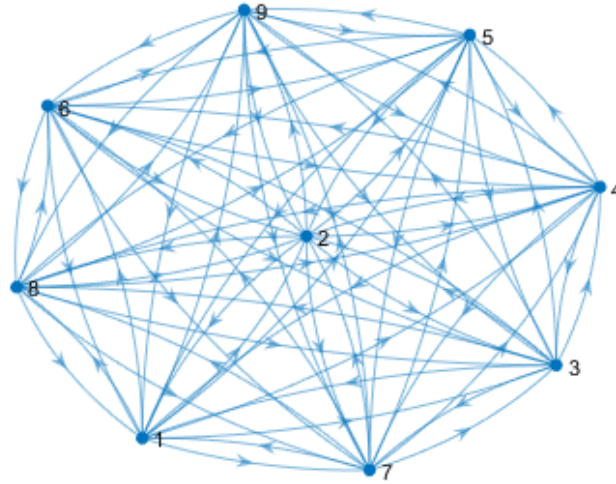
$$\text{μέγιστο είναι } [n/2] = \frac{n-1}{2}.$$

¹ Το σύμβολο $[x]$ δηλώνει το ακέραιο μέρος του αριθμού x , δηλαδή τον μέγιστο ακέραιο που είναι μικρότερος ή ίσος του x .



4.2 Τοπολογία Πλήρους Διασύνδεσης

Για την τοπολογία πλήρους σύνδεσης δίνεται παράδειγμα για $n=9$.



Σχήμα 4.2.α: Τοπολογία πλήρους σύνδεσης με 9 κόμβους.

Εδώ το πακέτο μπορεί να πάει σε όποιο κόμβο θέλει με 1 άλμα. Δηλαδή,

$$N(d) = \begin{cases} n, & d = 1 \\ 0, & d \neq 1 \end{cases}$$

Άρα, για τον υπολογισμό του μέσου πλήθους διανυθέντων συνδέσμων, με πιθανότητα

$\frac{1}{n}$ το πακέτο θα διανύσει 0 συνδέσμους (το τερματικό προορισμού θα είναι στον ίδιο

κόμβο με το τερματικό πηγής), ενώ με πιθανότητα $\frac{1}{n}$ θα κατευθύνεται στους

υπόλοιπους $n-1$ μεταγωγούς διανύοντας 1 σύνδεσμο. Επομένως,

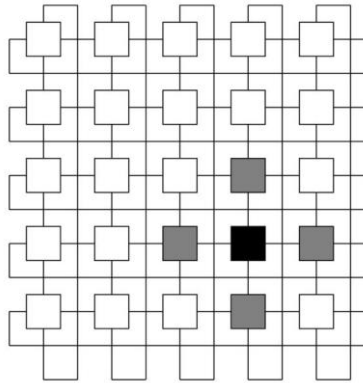
$$D = 0 \cdot \left(\frac{1}{n}\right) + 1 \cdot \left(\frac{1}{n}\right) \cdot (n-1) = \frac{n-1}{n}.$$

Επομένως, το ελάχιστο μονοπάτι που θα μπορεί να διανύσει ένα πακέτο θα είναι 0 και το μέγιστο μονοπάτι θα είναι 1, εφόσον όλοι οι κόμβοι επικοινωνούν μεταξύ τους.

4.3 Τοπολογία τόρου



Για την τοπολογία τόρου θα ασχοληθούμε με ένα $n \times n$ δυαδικό τόρο.



Σχήμα 4.3.1: 5×5 δυαδικός τόρος

Εδώ ο βαθμός του κάθε μεταγωγού είναι 4, άρα ένα πακέτο μπορεί να πάει σε τέσσερις γειτονικούς μεταγωγούς. Για τον υπολογισμό του μέγιστου πλήθους διανυθέντων συνδέσμων, η χειρότερη περίπτωση είναι το πακέτο να διανύσει $2\lfloor n/2 \rfloor$ συνδέσμους. Το ελάχιστο θα είναι ίσο με το μηδέν. Όμως αν αυξηθούν τα άλματα ο αριθμός των κόμβων που μπορεί να πάει το πακέτο αυξάνεται. Δηλαδή,

$$N(d) = \begin{cases} 4, & d = 1, \\ 8, & d = 2, \\ 16, & d = 3, \\ 24, & d = 4, \\ 36, & d = 5, \\ \vdots & \\ \vdots & \end{cases}$$

- Αν το d είναι περιττός ($d=2k+1$, $k \in \mathbb{Z}$),

$$\text{τότε } N(d) = N(d-1) + 4 \frac{n+1}{2}.$$

$$D = 0 \cdot \left(\frac{1}{n^2}\right) + 4 \cdot \left(1 + 2 + \dots + \frac{n-1}{2}\right) \cdot \left(\frac{1}{n^2}\right) = \frac{n-1}{2} \left(\frac{n-1}{2} + 1\right) \cdot \frac{4}{n^2} = \frac{n^2-1}{2n^2}.$$

- Αν το d είναι άρτιος ($d=2k$, $k \in \mathbb{Z}$),

$$\text{τότε } N(d) = N(d-1) + 4 \frac{n}{2}.$$



$$D = 0 \cdot \left(\frac{1}{n^2}\right) + 4 \cdot \left(1 + 2 + \dots + \frac{n}{2} - 1\right) \cdot \left(\frac{1}{n^2}\right) = \frac{\frac{n}{2} \left(\frac{n}{2} - 1\right)}{2} \cdot \frac{4}{n^2} = \frac{n-2}{2n}.$$



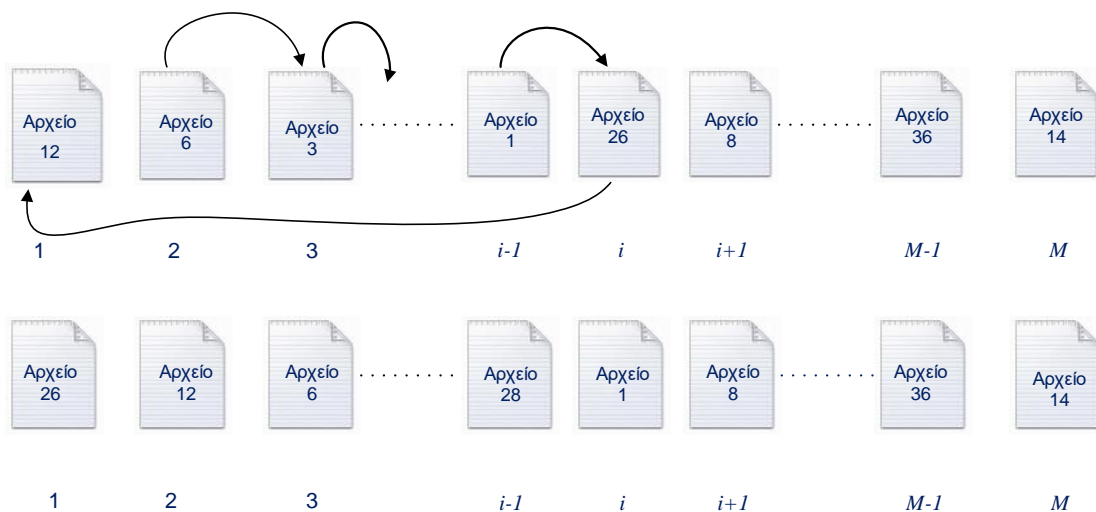
5

Κεφάλαιο 5^ο – Δημοτικότητα ενός αρχείου μέσα στο χρόνο

Σε αυτό το κεφάλαιο θα εξετάσουμε τη δημοτικότητα ενός αρχείου στο χρόνο, χωρίς να αναφερθούμε σε συγκεκριμένη τοπολογία δικτύου.

Αρχικά, έχουμε ένα δίκτυο το οποίο έχει M διαφορετικά αρχεία. Αυτά τα αρχεία είναι τοποθετημένα σε σειρά: πρώτο είναι αυτό με την μεγαλύτερη δημοφιλία και ακολουθούν τα υπόλοιπα αρχεία διατηρώντας φθίνουσα κατάσταση. Η δημοφιλία ενός αρχείου αλλάζει με το χρόνο. Δηλαδή, μπορούμε να εκφράσουμε την αλλαγή της σειράς δημοφιλιών ενός αρχείου ως μια διαδικασία Poisson με ρυθμό λ .

Συγκεκριμένα, θεωρούμε πως όταν ένα αρχείο ζητηθεί, γίνεται άμεσα το πιο δημοφιλές και τοποθετείται πρώτο στην σειρά κατάταξης δημοφιλιών. Τα αρχεία που βρίσκονται ανάμεσα στο αρχείο που ζητήθηκε και το πρώτο στην κατάταξη, θα χάσουν μια θέση καθώς και τα υπόλοιπα, θα παραμείνουν όπως έχουν. (Σχήμα 5.1)



Σχήμα 5.1 Παράδειγμα αλλαγής της δημοφιλιών αρχείων



(Θεωρούμε ότι το αρχείο το οποίο ζητήθηκε και γίνεται άμεσα το πιο δημοφιλές είναι το «Αρχείο 26» που από την θέση i μεταφέρεται στην 1^n).

Ιδιότητα 5.1: Σε μια διαδικασία Poisson με ρυθμό λ , οι χρόνοι μεταξύ δυο διαδοχικών γεγονότων είναι τυχαίοι συνεχείς εκθετικά κατανομημένοι έχοντας πάλι ρυθμό λ .

Ιδιότητα 5.2: Αν θεωρήσουμε μια διαδικασία Poisson με ρυθμό λ , όπου κάθε χρονική στιγμή, ένα γεγονός εμφανίζεται με μια πιθανότητα p_i όπου $i = 1, 2, \dots, n$ τότε η εμφάνιση των γεγονότων $1, 2, \dots, n$ ακολουθεί μια ανεξάρτητη διαδικασία Poisson με σχετικό ρυθμό $\lambda p_1, \lambda p_2, \dots, \lambda p_n$.

Για να συνεχίσουμε την ανάλυσή μας, θα πρέπει να επιλέξουμε ένα αρχείο από την αλυσίδα, το οποίο, από εδώ και πέρα θα ονομάζεται σημαδεμένο (tagged) αρχείο και έστω πως η δημοφιλία του μια χρονική στιγμή είναι η i . Θα μελετήσουμε την πιθανότητα να μεγαλώσει ή να μικρύνει το i όποτε έχουμε άφιξη ενός αιτήματος για αρχείο.

Έστω,

- A το ενδεχόμενο το σημαδεμένο αρχείο να γίνει το δημοφιλέστερο.

Η πιθανότητα του γεγονότος αυτού είναι:

$$P(A) = \frac{1}{M}.$$

- B το ενδεχόμενο το σημαδεμένο αρχείο να διατηρήσει τη δημοφιλία του.

Για να διατηρήσει τη δημοφιλία του, θα πρέπει το ζητούμενο αρχείο να βρίσκεται πιο ψηλά στη θέση κατάταξης από το σημαδεμένο, που βρίσκεται στη θέση i . Αν το σημαδεμένο αρχείο είναι το $i=1$ τότε η πιθανότητα να διατηρήσει τη θέση του είναι η $P(A)$.

$$P(B) = \begin{cases} \frac{1}{M}, & \alpha \nu \quad i=1 \\ \frac{i-1}{M}, & \alpha \nu \quad i=2,3,\dots,M \end{cases}.$$



- C το ενδεχόμενο το σημαδεμένο αρχείο να χάσει μια θέση στον πίνακα δημοφιλίας.

Για να χάσει μια θέση στη σειρά δημοφιλίας το σημαδεμένο αρχείο, θα πρέπει να βρίσκεται ανάμεσα στο δημοφιλέστερο αρχείο και το αρχείο που επιλέχθηκε να πάρει την πρώτη θέση στον πίνακα. Επομένως, αν το αρχείο που επιλέχθηκε είναι το i , η πιθανότητα είναι:

$$P(C) = \frac{M-i}{M}, \quad i = 1, 2, \dots, M-1.$$

Συνέπεια 5.1: Η δημοφιλία ενός αρχείου στο χρόνο αλλάζει ως μια διαδικασία Poisson έχοντας ως ρυθμό λ και πιθανότητα $P(A)$. Έτσι από την ιδιότητα 5.2, έχουμε ότι ένα αρχείο γίνεται το δημοφιλέστερο με ρυθμό

$$\lambda_A = \frac{\lambda}{M}$$

και χάνει μια θέση με ρυθμό

$$\lambda_C = \frac{\lambda(M-i)}{M}.$$

Παρατήρηση: και στις δυο διαδικασίες Poisson διατηρείται η ανεξαρτησία μεταξύ τους.

Σύμφωνα με την Ιδιότητα 5.1, κατά την οποία, μεταξύ δυο διαδοχικών γεγονότων η διαδικασία Poisson ακολουθεί εκθετικές μεταβλητές, οι χρόνοι μετά από τους οποίους ένα αρχείο γίνεται το δημοφιλέστερο ή χάνει μια θέση κατάταξης στον πίνακα δημοφιλίας, είναι τυχαίες εκθετικές μεταβλητές με ρυθμούς λ_A και λ_C .

Επομένως, όσο ο χρόνος μεταξύ των αλλαγών θέσεων στη σειρά δημοφιλίας ενός αρχείου είναι εκθετικός και προσμετράται ως κατανομημένος, το μοντέλο της δημοφιλίας ανάμεσα στο χρόνο είναι CTMC.

Ορισμός 5.1: Έστω P_{ij} η πιθανότητα μετάβασης από την κατάσταση i στην κατάσταση j .



Ιδιότητα 5.3: Όταν η CTMC αφήνει την κατάσταση i , μεταβαίνει στην κατάσταση j με κάποια πιθανότητα P_{ij} . Οι πιθανότητες P_{ij} πρέπει να ικανοποιούν τις ακόλουθες εξισώσεις:

$$P_{ii} = 0, \text{ για όλα τα } i$$

$$\sum_j P_{ij} = 1 \text{ για όλα τα } i.$$

Ορισμός 5.2: Για κάθε ζεύγος καταστάσεων i και j , έχουμε

$$q_{ij} = v_i P_{ij}$$

όπου v_i είναι ο ρυθμός με τον οποίο η αλυσίδα κάνει μια μετάβαση όταν βρίσκεται στην κατάσταση i , P_{ij} η πιθανότητα αυτής της μετάβασης να είναι στην κατάσταση j και q_{ij} να είναι ο ρυθμός μετάβασης από την κατάσταση i στην κατάσταση j . Ισχύει, επίσης, κατά τα γνωστά για τις αλυσίδες Markov συνεχούς χρόνου,

$$v_i = \sum_j v_i P_{ij} = \sum_j q_{ij} \text{ και } P_{ij} = \frac{q_{ij}}{v_i} = \frac{q_{ij}}{\sum_j q_{ij}}.$$

Υπολογίζοντας τους ρυθμούς μετάβασης και δημιουργώντας τον πίνακα ρυθμών μετάβασης \mathbb{Q} με βάσει τη συνέπεια 5.1, πρέπει να ορίσουμε το εξής:

$$q_{ii} = -v_i = -\sum_{j \neq i} q_{ij}.$$

Δηλαδή, το άθροισμα των στοιχείων κάθε γραμμής του \mathbb{Q} είναι ίσο με 0.

Επομένως,

- Στην κατάσταση 1, η μόνη επιλογή της αλυσίδας είναι η μετάβασή της στην κατάσταση 2. Επομένως, ο ρυθμός μετάβασής της θα είναι:

$$q_{12} = v_1 = \frac{\lambda(M-1)}{M}.$$

- Στην κατάσταση i , $i = 2, 3, \dots, M-1$ της αλυσίδας, το αρχείο μπορεί είτε να βρεθεί στην κατάσταση 1 είτε να βρεθεί στην κατάσταση $i+1$. Οπότε, θα βρεθεί στην κατάσταση 1 με ρυθμό



$$q_{i1} = \frac{\lambda}{M}.$$

Επίσης, θα βρεθεί στην κατάσταση $i+1$ με ρυθμό

$$q_{i,i+1} = \frac{\lambda(M-i)}{M}.$$

Επίσης ο ρυθμός στην οποία η CTMC κάνει μια μετάβαση όταν βρίσκεται στην κατάσταση i :

$$v_i = \frac{\lambda(1+M-i)}{M}.$$

- Στην κατάσταση M , η μόνη επιλογή που έχει το αρχείο, είναι να γίνει το δημοφιλέστερο και μεταβεί στην κατάσταση 1. Έτσι,

$$q_{M1} = v_M = \frac{\lambda}{M}.$$

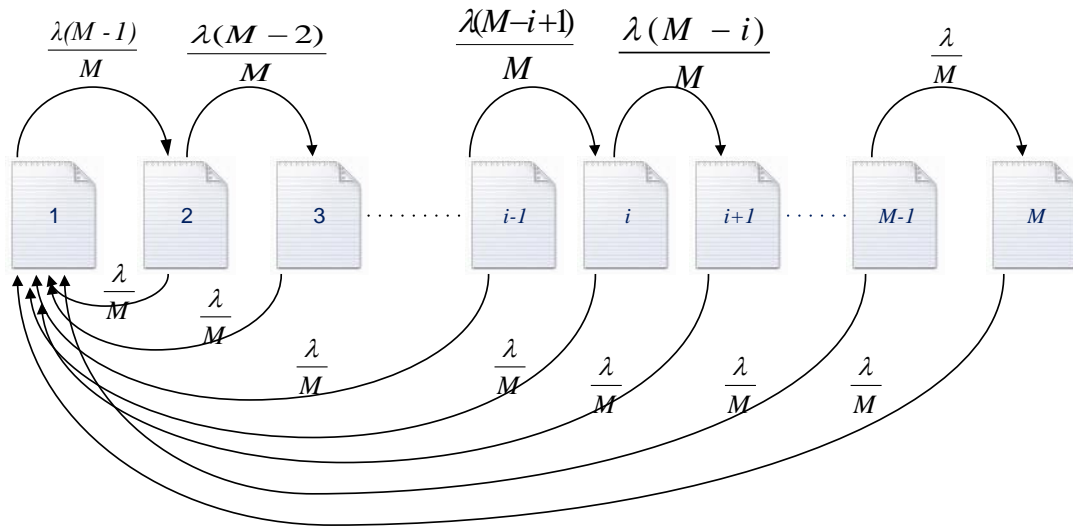
Συγκεντρωτικά, ο πίνακας ρυθμών μετάβασης είναι ο εξής:

$$Q = \begin{bmatrix} \frac{\lambda(M-1)}{M} & \frac{\lambda(M-1)}{M} & 0 & \dots & \dots & \dots & \dots & 0 & 0 \\ \frac{\lambda}{M} & -\frac{\lambda(M-1)}{M} & \frac{\lambda(M-2)}{M} & 0 & \dots & \dots & \dots & \dots & 0 \\ \frac{\lambda}{M} & 0 & -\frac{\lambda(M-2)}{M} & \frac{\lambda(M-3)}{M} & 0 & \dots & \dots & \dots & \vdots \\ \frac{\lambda}{M} & \vdots & 0 & 0 & \ddots & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & 0 & 0 & 0 & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & 0 & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \dots & \dots & \vdots \\ \frac{\lambda}{M} & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ \frac{\lambda}{M} & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ \frac{\lambda}{M} & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \\ \frac{\lambda}{M} & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \vdots \end{bmatrix}$$

Πίνακας 5.1



Παρακάτω περιγράφεται το διάγραμμα ρυθμών μετάβασης της αλυσίδας (Διάγραμμα 5.1).



Διάγραμμα 5.1

Για να υπολογίσουμε τις πιθανότητες μετάβασης P_{ij} από την κατάσταση i στην κατάσταση j έχουμε τα εξής:

- Όπως αναφέραμε και πριν στην κατάσταση 1, η μόνη επιλογή της CTMC είναι η μετάβασή της στην κατάσταση 2. Άρα,

$$P_{12} = 1$$

- Στην κατάσταση i , $i = 2, 3, \dots, M-1$ στην αλυσίδα, έχουμε,

$$P_{i1} = \frac{1}{1+M-i} \text{ και } P_{i,i+1} = \frac{M-i}{1+M-i}.$$

- Στην κατάσταση M ,

$$P_{M1} = 1$$

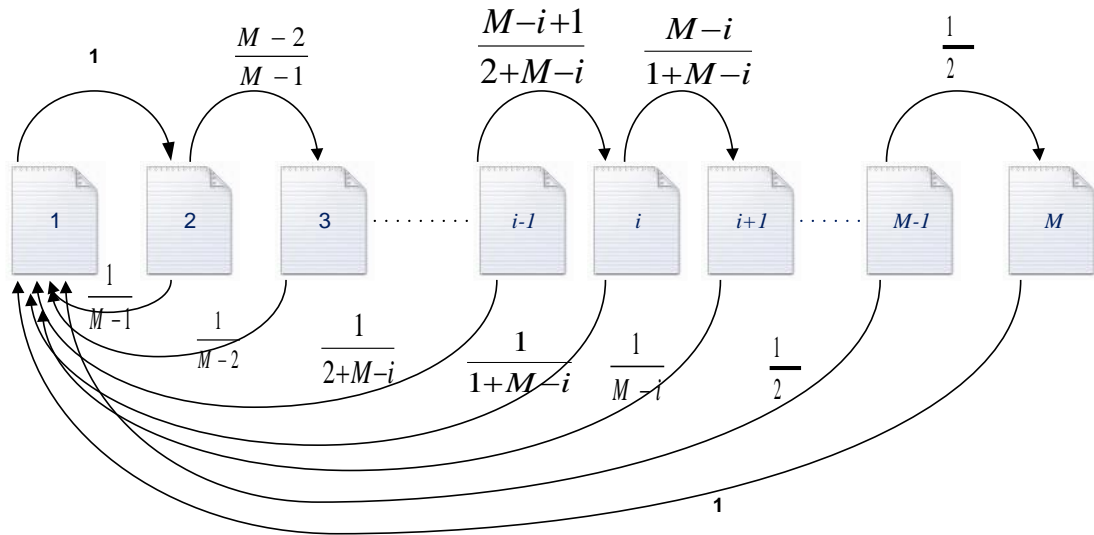
Επομένως, λαμβάνοντας τα παραπάνω και σύμφωνα με την ιδιότητα 3, ο πίνακας πιθανοτήτων μετάβασης είναι ο εξής:



$$P = \begin{bmatrix} 0 & 1 & 0 & \dots & & & \dots & 0 & 0 \\ \frac{1}{M-1} & 0 & \frac{M-2}{M-1} & 0 & \dots & & & & 0 \\ \frac{1}{M-2} & 0 & 0 & \frac{M-3}{M-2} & 0 & & & & \vdots \\ \frac{1}{M-3} & \vdots & 0 & 0 & \ddots & & & & \\ \vdots & & 0 & 0 & 0 & & & & \\ & & \vdots & \vdots & 0 & & & & \\ \vdots & & & & \ddots & 0 & \dots & & \\ \frac{1}{1+M-i} & & & & 0 & \frac{M-i}{1+M-i} & 0 & \dots & \\ \vdots & & & & \ddots & 0 & \frac{M-i-1}{M-i} & 0 & \dots \\ & & & & & 0 & 0 & 0 & \dots \\ & & & & & & 0 & \ddots & \\ & & & & & & & & \ddots & \ddots & 0 & \vdots \\ \vdots & \vdots & & & & & & \dots & 0 & 0 & \frac{2}{3} & 0 \\ \frac{1}{2} & 0 & \dots & & & & & & \dots & 0 & \frac{1}{2} \\ 1 & 0 & \dots & & & & & & & \dots & 0 \end{bmatrix}$$

Πίνακας 5.2

Παρακάτω περιγράφεται το διάγραμμα πιθανοτήτων μετάβασης της αλυσίδας (Διάγραμμα 5.2), στη πρώτη θέση της οποίας, βρίσκεται το αρχείο με τη μεγαλύτερη δημοφιλία και συνεχίζεται με φθίνουσα σειρά.



Διάγραμμα 5.2



6

Κεφάλαιο 6^ο – Σχέση δημοφιλίας με αντίγραφα ενός αρχείου στο δίκτυο.

Στο προηγούμενο κεφάλαιο είδαμε πώς η δημοφιλία του σημαδεμένου αρχείου αλλάζει με το χρόνο και την μοντελοποιήσαμε ως μια μονοδιάστατη αλυσίδα Markov. Στο παρόν κεφάλαιο μελετάμε, αναλύουμε και μοντελοποιούμε τη σχέση της δημοφιλίας ενός αρχείου, με τον αριθμό των αντιγράφων του μέσα στο χρόνο.

6.1 Το πρόβλημα

Υποθέτουμε ότι όλοι οι κόμβοι είναι όμοιοι, δηλαδή έχουν την ίδια χωρητικότητα μνήμης και την ίδια κατανομή αιτημάτων. Εδώ θα εξηγήσουμε αυτές τις παραδοχές, προτού μεταβούμε στην ανάλυση του προβλήματος.

1. Ικανότητα αποθήκευσης. Αν η χωρητικότητα μιας μνήμης στον κόμβο είναι απεριόριστη, όλα τα αρχεία του δικτύου θα μπορούν να αποθηκευτούν στην μνήμη αυτή. Αυτό όμως δεν είναι ρεαλιστικό. Έτσι, συνήθως η μνήμη αυτή, έχει περιορισμένη χωρητικότητα και δεν μπορεί να αποθηκεύσει τον μεγάλο όγκο των αρχείων του δικτύου. Η ικανότητα της μνήμης θα τεθεί στο μοντέλο μας.
2. Η κατανομή των αιτήσεων. Η κατανομή των αιτήσεων επηρεάζει άμεσα την στρατηγική της αποθήκευσης. Ένα δημοφιλές αρχείο γίνεται πιο περιζήτητο και έτσι προωθείται πιο συχνά στο δίκτυο. Διάφορες έρευνες έχουν δείξει ότι η δημοφιλία των αρχείων ακολουθεί μια Zipf κατανομή, στην οποία ο ρυθμός αιτήσεων φθίνει πολυωνυμικά.

Θεωρώντας τους παραπάνω παράγοντες, θέτουμε το πρόβλημα: Έχουμε M διαφορετικά αρχεία στο δίκτυο, αριθμώντας από το 1 έως το M . Για να το



απλουστεύσουμε, θεωρούμε ότι όλοι οι κόμβοι κάνουν τον ίδιο αριθμό αιτημάτων και κάθε κόμβος γεννάει αιτήματα ανεξάρτητα από τους άλλους και ανεξάρτητα από τα προηγούμενα αιτήματα. Ακόμα, όπως αναφέραμε και από το προηγούμενο κεφάλαιο, τα αρχεία είναι ταξινομημένα με βάση τη δημοτικότητα. Επιλέγουμε τυχαία ένα αρχείο, το οποίο, για να το αναγνωρίζουμε τοποθετούμε μια ετικέτα, όπως και στο προηγούμενο κεφάλαιο, και από εδώ και πέρα θα το ονομάζουμε «σημειωμένο» (tagged) αρχείο. Επομένως, θα μελετήσουμε τη συμπεριφορά του σημειωμένου αρχείου με βάση την δημοφιλία του και των αριθμών των αντιγράφων του ανάμεσα στο χρόνο.

6.2 Ανάλυση του προβλήματος

Για να επιτευχθεί αυτό, δημιουργούμε μια δισδιάστατη αλυσίδα, που παίρνει διακριτές τιμές, όπου στις οριζόντιες γραμμές αναγράφονται ο αριθμός των αντιγράφων του σημειωμένου αρχείου και στις κάθετες στήλες η δημοφιλία. Επομένως, οι τιμές που παίρνει η οριζόντια διάσταση είναι από το 1 έως το N και η κάθετη από το 1 έως το M . Για παράδειγμα, αν η αλυσίδα βρίσκεται στην κατάσταση (m,n) , σημαίνει πως το σημειωμένο αρχείο έχει τάξη δημοφιλίας m (βρίσκεται στη m θέση της σειράς δημοφιλίας) και περιέχει n αντίγραφα στο δίκτυο. (Σχήμα 6.1)

Για να απλουστεύσουμε την ανάλυση θα κάνουμε τις παρακάτω υποθέσεις.

Υπόθεση 1: Όλα τα αρχεία είναι το ίδιο μέγεθος και όλοι οι κόμβοι μπορούν να αποθηκεύσουν K αρχεία στην μνήμη τους.

Υπόθεση 2: Τα αιτήματα ακολουθούν την Zipf κατανομή με παράμετρο α .

Υπόθεση 3: Είναι εξίσου πιθανό όλοι οι κόμβοι να έχουν οποιοδήποτε αρχείο αποθηκευμένο.

Υπόθεση 4: Όταν ένας κόμβος ζητήσει ένα αρχείο και δεν το έχει, γίνεται τυχαία επιλογή ενός αρχείου από την μνήμη του για να πεταχτεί, ώστε να πάρει τη θέση του του ζητούμενου αρχείου.

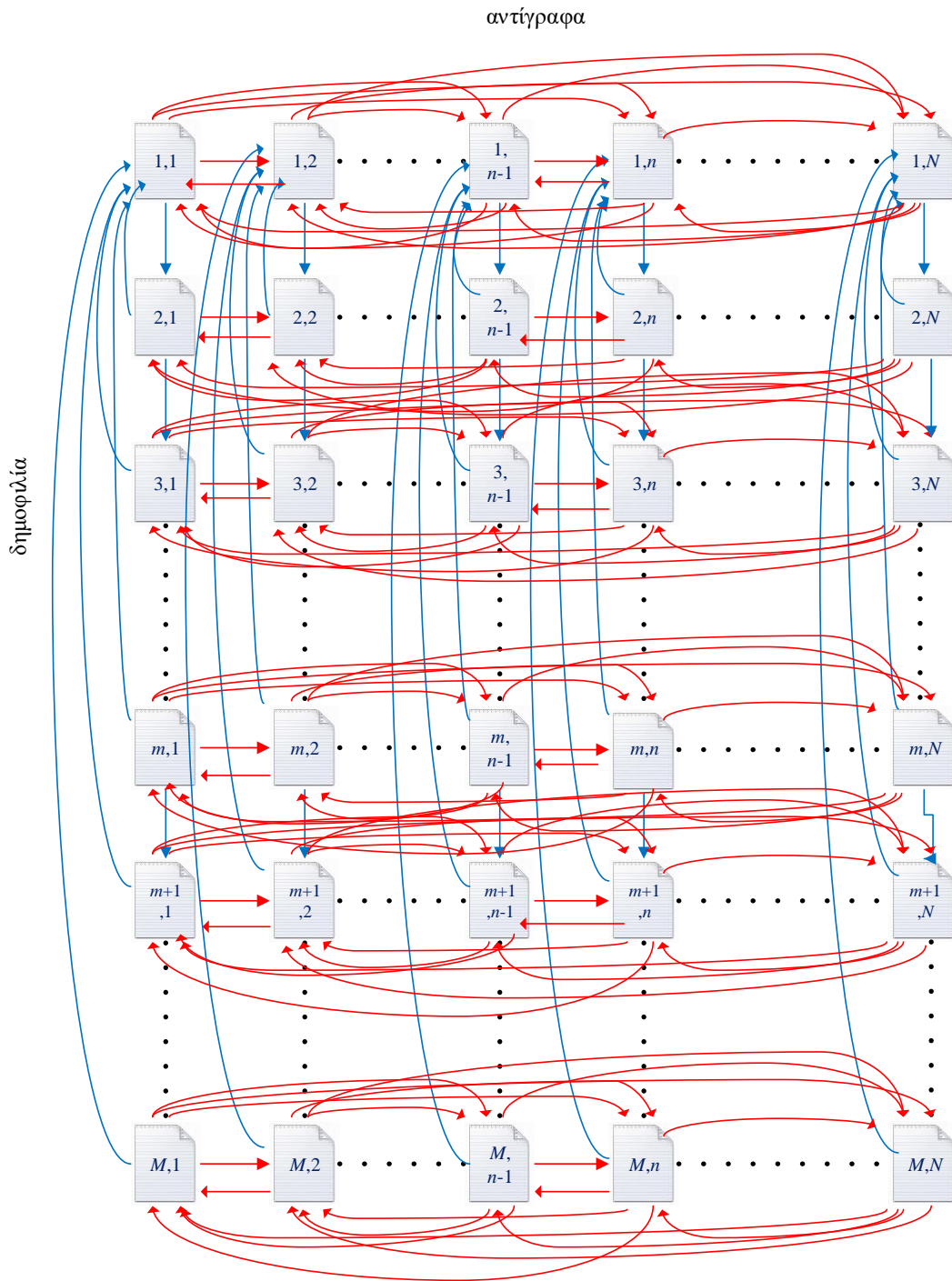
Υπόθεση 5: Στην διαδρομή ενός ζητούμενου αρχείου από την πηγή στον προορισμό, όλοι οι κόμβοι από τους οποίους περνάει, αποθηκεύουν πάντα ένα αντίγραφο.

Υπόθεση 6: Πάντα η διαδρομή ενός αρχείου χρησιμοποιεί το συντομότερο μονοπάτι.



Σύμβολο	Επεξήγηση
N	Ο αριθμός των κόμβων του δικτύου
K	Το μέγεθος της μνήμης κάθε κόμβου
M	Ο αριθμός των αρχείων
α	Παράμετρος της Zipf κατανομής
λ	Ρυθμός αλλαγών της δημοφιλίας
r	Ρυθμός αιτημάτων των κόμβων
m	Τρέχουσα δημοφιλία του σημειωμένου αρχείου
n	Τρέχουσα αντίγραφα του σημειωμένου αρχείου

Πίνακας 6.1 - Επεξήγηση Συμβόλων



Σχήμα 6.1: Διάγραμμα Δημοφιλίας – Αντιγράφων σε δισδιάστατη αλυσίδα CTMC

Όπως αναφέραμε, η δημοφιλία των αρχείων του δικτύου ακολουθεί την Zipf κατανομή. Η κατανομή αυτή ορίζει ότι, αν έχουμε M αρχεία τα οποία είναι διατεταγμένα σύμφωνα με την δημοτικότητα τους, τότε η πιθανότητα επιλογής του m -πιο δημοφιλούς αρχείου ισούται με

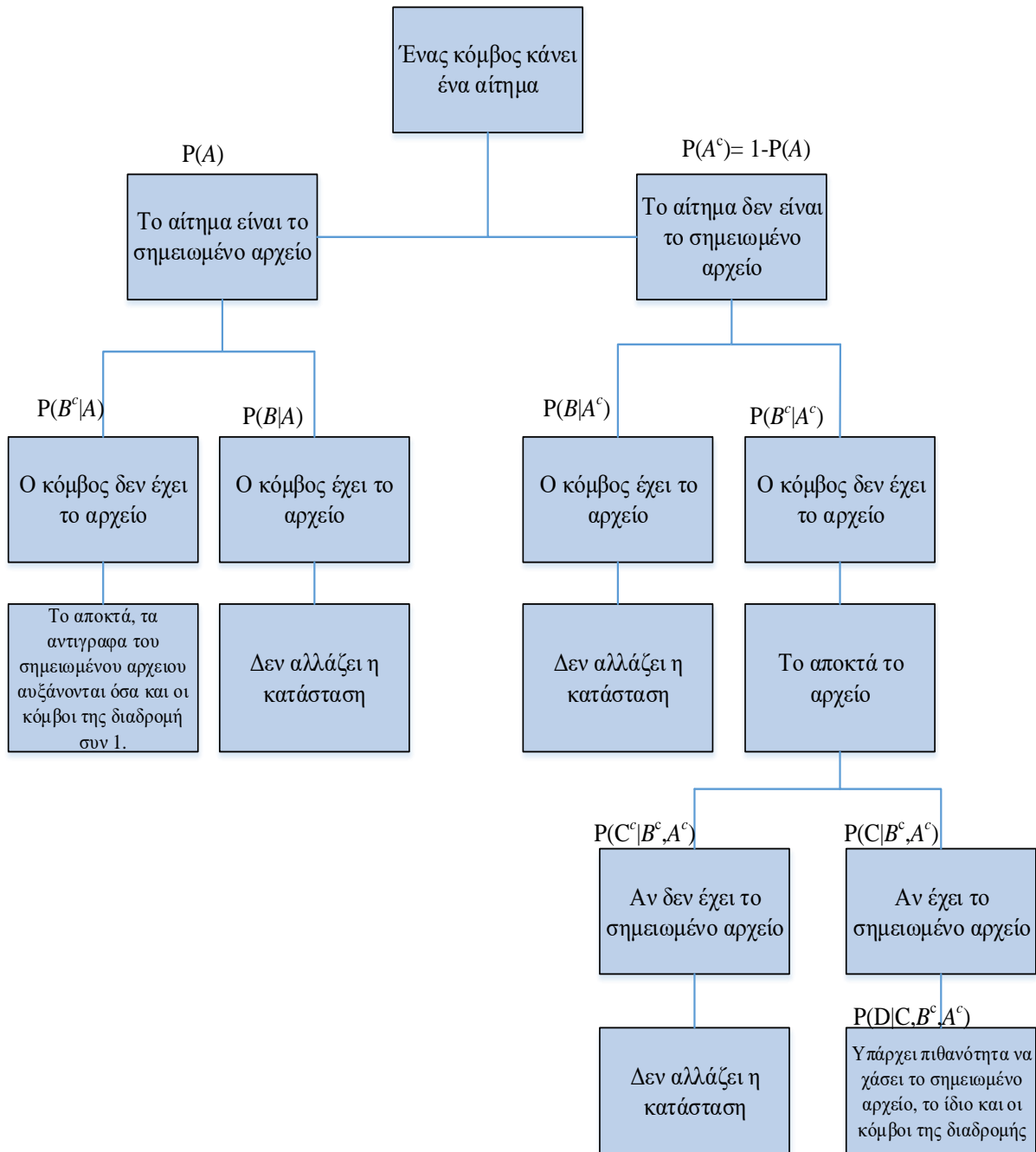


$$P(m) = \frac{\frac{1}{m^a}}{\sum_{j=1}^M \frac{1}{j^a}}$$

όπου a η σταθερά του Zipf.

Η αλλαγή του αριθμού των αντιγράφων ενός αρχείου προκύπτει από διάφορα ενδεχόμενα τα οποία θα εξηγήσουμε και απεικονίζονται στο Σχήμα 4.2. Αρχικά, θεωρούμε ότι ο ρυθμός με τον οποίο οι κόμβοι ζητάνε ένα αρχείο στο δίκτυο είναι r . Όταν ένας κόμβος κάνει ένα αίτημα, οι επιλογές του είναι δυο: είτε το αίτημα αφορά το σημειωμένο αρχείο, είτε όχι. Αν το αίτημα αφορά το σημειωμένο αρχείο και ο κόμβος έχει το αρχείο, τότε δεν αλλάζει η κατάσταση της αλυσίδας. Αλλά, αν ο κόμβος δεν έχει το αρχείο, το αποκτά και καθώς το αποθηκεύει στη προσωρινή του μνήμη, το αποθηκεύουν και οι κόμβοι από τους οποίους πέρασε το αρχείο (Υπόθεση 5). Έτσι ο αριθμός των αντιγράφων αυξάνεται, όσο και οι κόμβοι της διαδρομής, συν 1 του προορισμού.

Αντίθετα, αν το αίτημα δεν αφορά το σημειωμένο αρχείο και ο κόμβος έχει το ζητούμενο αρχείο, τότε πάλι δεν αλλάζει η κατάσταση της αλυσίδας. Αλλά, αν ο κόμβος δεν έχει το ζητούμενο αρχείο, το αποκτά και δημιουργούνται δυο ενδεχόμενα. Πρώτον, ο κόμβος να μην έχει το σημειωμένο αρχείο, οπότε πάλι δεν αλλάζει η κατάσταση του. Δεύτερον, αν ο κόμβος το έχει, τότε υπάρχει πιθανότητα να χάσει το σημειωμένο αρχείο (Υπόθεση 4) και να μειωθεί ο αριθμός των αντιγράφων του.



Σχήμα 6.2: Δέντρο καταστάσεων των ενδεχόμενων που μπορεί να προκύψουν μετά από μια αίτηση για αρχείο.

6.2.1 Υπολογισμός πιθανοτήτων του Δέντρου καταστάσεων

Στο δισδιάστατο πίνακα Δημοφιλίας – Αντιγράφων (Σχήμα 6.1) για τον κάθετο άξονα, οι πιθανότητες μετάβασης είναι γνωστές από το προηγούμενο κεφάλαιο. Δηλαδή, περιγράφονται από τη μονοδιάστατη αλυσίδα και αφορούν τη δημοφιλία του αρχείου.



Σχετικά με τον οριζόντιο άξονα, για να υπολογίσουμε τις πιθανότητες μετάβασης έχουμε τα εξής:

Έστω A το ενδεχόμενο το αίτημα να είναι για το σημειωμένο αρχείο, B το ενδεχόμενο ο κόμβος να έχει το ζητούμενο αρχείο, C το ενδεχόμενο ο κόμβος να έχει το σημειωμένο αρχείο και D το ενδεχόμενο ο κόμβος να χάσει το σημειωμένο αρχείο. Στη συνέχεια υπολογίζουμε τις πιθανότητες τις οποίες αναγράφουμε στο δέντρο καταστάσεων (Σχήμα 6.2).

Έστω ότι βρισκόμαστε στην κατάσταση (m,n) στην αλυσίδα.

- Η πιθανότητα το αίτημα ενός κόμβου να αφορά το σημειωμένο αρχείο υπολογίζεται από την Zipf κατανομή. Δηλαδή, στην κατάσταση που βρισκόμαστε, στην (m,n) , γνωρίζουμε ότι το αρχείο είναι το m πιο δημοφιλές αρχείο, άρα η πιθανότητα το αίτημα να αφορά το σημειωμένο αρχείο είναι :

$$P(A) = \frac{\frac{1}{m^a}}{\sum_{j=1}^M \frac{1}{j^a}}.$$

- Η πιθανότητα ο κόμβος να έχει το ζητούμενο αρχείο, δεδομένου ότι το αίτημα αφορά το σημειωμένο αρχείο είναι:

$$P(B|A) = \frac{n}{N}.$$

Αυτό ισχύει γιατί τα αντίγραφα του σημειωμένου αρχείου στην κατάσταση (m,n) είναι n από N κόμβους (Υπόθεση 3).

- Η πιθανότητα ο κόμβος να μην έχει το ζητούμενο αρχείο με δεδομένο ότι το αίτημα να αφορά το σημειωμένο αρχείο είναι:

$$P(B^c|A) = 1 - P(B|A) = 1 - \frac{n}{N} = \frac{N-n}{N}.$$

- Η πιθανότητα στην οποία ο κόμβος κάνει το αίτημα και το αίτημα δεν είναι για το σημειωμένο αρχείο είναι:

$$P(A^c) = 1 - P(A) = 1 - \frac{\frac{1}{m^a}}{\sum_{j=1}^M \frac{1}{j^a}}.$$



- Ακόμα, η πιθανότητα ο κόμβος να έχει το ζητούμενο αρχείο, δεδομένου ότι το ζητούμενο αρχείο δεν είναι το σημειωμένο αρχείο είναι:

$$P(B|A^c) = \frac{K}{M}.$$

Αυτό ισχύει εφόσον ο κάθε κόμβος μπορεί να αποθηκεύσει στην προσωρινή του μνήμη K αρχεία από τα M συνολικά (Υπόθεση 1 και 3).

- Η πιθανότητα ο κόμβος να μην έχει το ζητούμενο αρχείο και το ζητούμενο αρχείο να μην είναι το σημειωμένο αρχείο, τότε έχουμε:

$$P(B^c|A^c) = 1 - P(B|A^c) = 1 - \frac{K}{M} = \frac{M - K}{M}.$$

- Η πιθανότητα ο κόμβος να χάσει το σημειωμένο αρχείο λόγω της Υπόθεσης 4 είναι:

$$P(D) = \frac{1}{K}.$$

Επομένως, η πιθανότητα ο κόμβος να αποκτήσει το σημειωμένο αρχείο και να αυξήσει κατά ένα τον αριθμό των αντιγράφων του σημειωμένου αρχείου είναι: (το ορίζουμε ως P_S)

$$P_S = P(A) \cdot P(B^c|A) = \frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \cdot \frac{N - n}{N}.$$

Επίσης, η πιθανότητα ο κόμβος να χάσει το σημειωμένο αρχείο και να μειώσει τον αριθμό των αντιγράφων είναι: (το ορίζουμε ως P_C)

$$P_C = P(A^c) \cdot P(B^c|A^c) \cdot P(C|B^c, A^c) = \left(1 - \frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{M - K}{M} \right) \cdot \left(\frac{n}{N} \right).$$

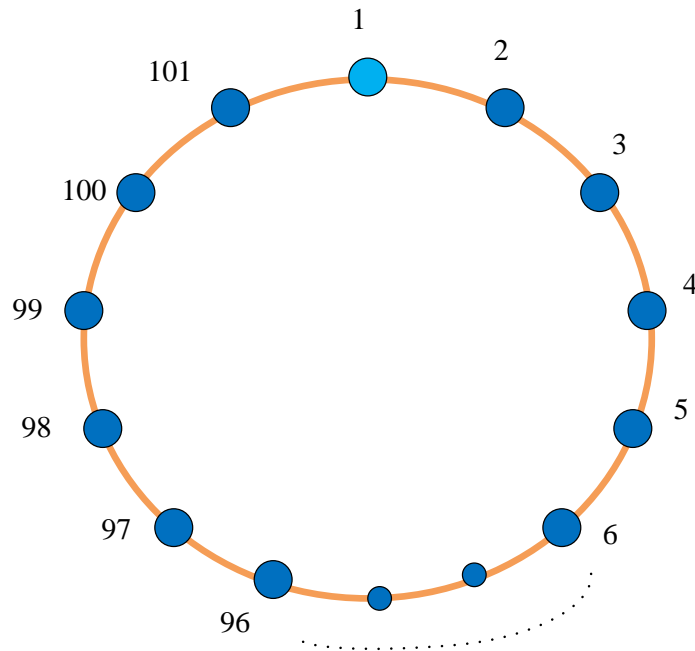
Για να συνεχίσουμε να υπολογίζουμε τις πιθανότητες μετάβασης και κυρίως τις πιθανότητες μετάβασης στον οριζόντιο άξονα, από μια κατάσταση σε μια οποιαδήποτε άλλη, θα πρέπει να υπολογίσουμε τους ρυθμούς μετάβασης μεταξύ καταστάσεων των οποίων τα πλήθη αντιγράφων διαφέρουν περισσότερο από μια μονάδα. Οι ρυθμοί αυτοί



μετάβασης εξαρτώνται από την τοπολογία του δικτύου. Στη συνέχεια θα μελετήσουμε μια συγκεκριμένη τοπολογία, αλλά τονίζουμε ότι με παρόμοια μέθοδο μπορούμε να υπολογίσουμε τους ρυθμούς μετάβασης και για άλλες τοπολογίες.

6.2.2 Υπολογισμός πιθανοτήτων ύπαρξης του σημειωμένο αρχείου στους γειτονικούς κόμβους

Το παράδειγμα αυτό θα μας βοηθήσει στον υπολογισμό της πιθανότητας αύξησης των αντιγράφων από ένα κόμβο. Έστω ότι έχουμε ένα δίκτυο τοπολογίας δακτυλίου (Σχήμα 6.3), που αποτελείται από $N=100$ κόμβους (Ο κόμβος 1 δεν προσμετράται, εφόσον έχουμε θέσει απ' αρχής ότι δεν έχει το σημειωμένο αρχείο). Ο αριθμός των αντιγράφων του σημειωμένο αρχείου είναι $n=10$, ο κόμβος 1 κάνει το αίτημα για το αρχείο και δεν το έχει το σημειωμένο αρχείο και ορίζουμε ως X την απόσταση του κόμβου 1 από τον πιο κοντινό κόμβο που έχει το σημειωμένο αρχείο (στο παράδειγμα, λόγω δακτυλίου το πλήθος των κόμβων για κάθε τιμή του X αφορά 2 κόμβους). Παρατηρούμε ότι η ζητούμενη πιθανότητα μπορεί να προκύψει με χρήση της υπεργεωμετρικής κατανομής. Συγκεκριμένα, έχουμε



Σχήμα 6.3: Δακτύλιος



$P(X=1) = P(\text{οι δύο γείτονες να έχουν το αρχείο}) = 1 - P(\text{οι δύο γείτονες να μην έχουν$

$$\text{το αρχείο}) = 1 - \frac{\binom{N-2}{n}}{\binom{N}{n}}$$

Θέλουμε να βρούμε την πιθανότητα το αρχείο να υπάρχει στους δυο γειτονικούς κόμβους (Σχήμα 6.4.a). Αυτή η πιθανότητα θα είναι ίση με τη διαφορά ένα μείον τη πιθανότητα οι δυο γειτονικοί κόμβοι να μην έχουν το αρχείο. Η πιθανότητα, οι δυο γειτονικοί κόμβοι να μην έχουν το αρχείο, είναι τα n αντίγραφα να βρίσκονται στους $N-2$ κόμβους από το N σύνολο. Για το παράδειγμα, οι 98 κόμβοι πρέπει να έχουν και τα 10 αντίγραφα του σημειωμένου αρχείου.

Συνεχίζοντας σύμφωνα με τη θεωρία των πιθανοτήτων θα έχω,

$$P(X \leq 2) = 1 - P(X \geq 3) = 1 - \frac{\binom{N-4}{n}}{\binom{N}{n}}$$

(Σχήμα 6.4.c)

$$P(X = 2) = P(X \leq 2) - P(X \leq 1)$$

(Σχήμα 6.4.b)

$$P(X \geq 3) = \frac{\binom{N-4}{n}}{\binom{N}{n}}$$

(Σχήμα 6.4.d)

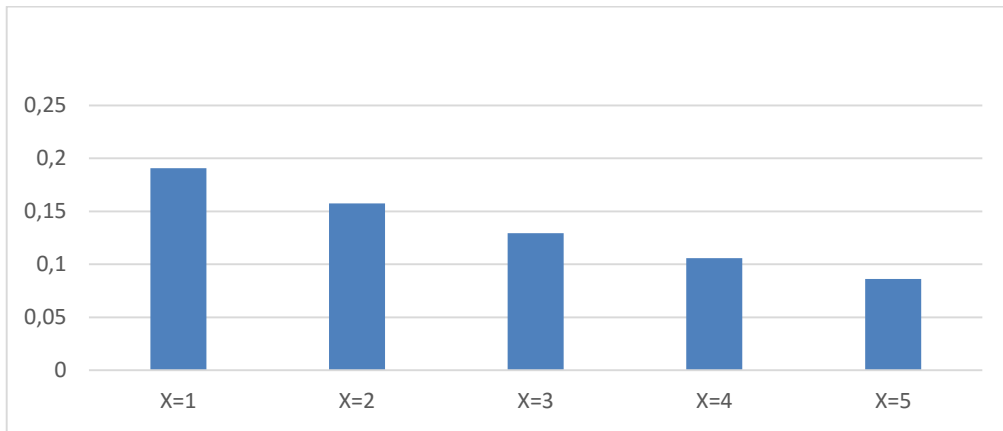
$$P(X = 3) = P(X \geq 3) - P(\geq 4) = \frac{\binom{N-4}{n}}{\binom{N}{n}} - \frac{\binom{N-6}{n}}{\binom{N}{n}}$$

⋮
⋮

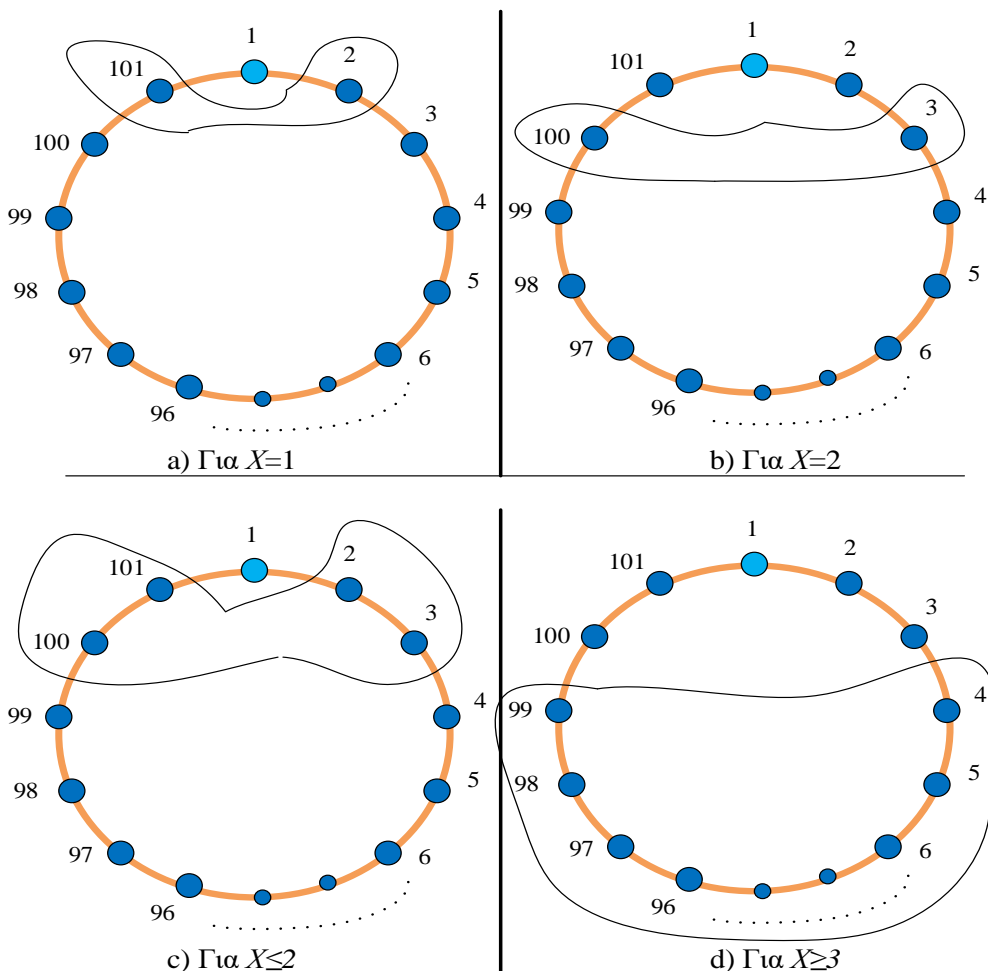
$$P(X = k) \quad \mu\epsilon \quad k = 1, 2, \dots, \frac{N}{2}$$



Παρακάτω παρουσιάζεται το ιστόγραμμα της κατανομής που αποτελείται από 100 κόμβους και 10 αντίγραφα του σημειωμένου αρχείου (Σχήμα 6.5).



Σχήμα 6.5: Ιστόγραμμα της κατανομής του παραπάνω παραδείγματος.



Σχήμα 6.4: Απεικόνιση περιπτώσεων παραδείγματος



6.2.3 Υπολογισμός πιθανοτήτων μετάβασης, που μειώνουν τα αντίγραφα του σημειωμένου αρχείου

Συνεχίζοντας την ανάλυσή μας θα πρέπει να υπολογίσουμε τις πιθανότητες μετάβασης που μειώνουν τα αντίγραφα του σημειωμένου αρχείου κατά αυθαίρετες τιμές. Εδώ ορίζουμε:

t - το πλήθος των κόμβων διαμέσου των οποίων θα περάσει το αρχείο καθώς πάει προς το προορισμό.

Y - το πλήθος των κόμβων από τους t κόμβους που έχουν το σημειωμένο αρχείο.

Έστω πως είμαστε στη κατάσταση (m, n) , $t = 1, 2, \dots, N-1$ και $Y \leq n$.

Άρα, η πιθανότητα θα υπολογισθεί με το νόμο της ολικής πιθανότητας και θα έχω,

P (να πέσει το σημειωμένο αρχείο s θέσεις | $X = t$) =

$$= \sum_{y=1}^t P(\text{να πέσει το σημειωμένο αρχείο } s \text{ θέσεις} \mid X = t, Y = y) \cdot P(Y = y) \quad (6.2.3.1)$$

με $s \leq n-1$ και $s \leq t$.

Το X όπως αναφέραμε στο παράδειγμα είναι η απόσταση του κόμβου που βρισκόμαστε από τον πιο κοντινό κόμβο που έχει το σημειωμένο αρχείο.

Έτσι το (6.2.3.1) γίνεται:

$$(6.2.3.1) = \sum_{t=1}^{N-1} \left(\binom{t}{s} \cdot \left(\frac{1}{K}\right)^s \cdot \left(1 - \frac{1}{K}\right)^{t-s} \right) \cdot \left(\frac{\binom{n}{s} \cdot \binom{N-n}{t-s}}{\binom{N}{t}} \right).$$

Εδώ χρησιμοποιούμε αρχικά μια διωνυμική κατανομή στην οποία θεωρούμε ως επιτυχία την αποβολή του σημειωμένου αρχείου. Δηλαδή, θέλουμε την πιθανότητα να έχουμε s επιτυχίες σε t κόμβους και θεωρώντας ως πιθανότητα επιτυχίας να είναι ίση με $\frac{1}{K}$, αφού αυτή είναι η πιθανότητα η μνήμη να αποβάλλει ένα αρχείο. Έπειτα

χρησιμοποιούμε μια υπεργεωμετρική κατανομή στην οποία αφού έχουμε επιλέξει t κόμβους, θέλουμε να βρούμε την πιθανότητα οι s κόμβοι από τους t να έχουν το σημειωμένο αρχείο. Ο συνολικός αριθμός των σημειωμένων αρχείων στο δίκτυο είναι n .

Επομένως, η πιθανότητα να πέσει το σημειωμένο αρχείο μια θέση είναι:



$$\begin{aligned}
 & P(\text{να πέσει το σημειωμένο αρχείο 1 θέση} \mid X = t) = \\
 & = \sum_{y=1}^t P(\text{να πέσει το σημειωμένο αρχείο 1 θέση} \mid X = t, Y = y) \cdot P(Y = y) = \\
 & = \binom{1}{1} \cdot \left(\frac{1}{K}\right) \cdot \left(1 - \frac{1}{K}\right)^0 \cdot \left(\frac{\binom{n}{1} \cdot \binom{N-n}{0}}{\binom{N}{1}}\right) + \binom{2}{1} \cdot \left(\frac{1}{K}\right) \cdot \left(1 - \frac{1}{K}\right) \cdot \left(\frac{\binom{n}{1} \cdot \binom{N-n}{1}}{\binom{N}{2}}\right) + \dots \\
 & \dots + \binom{N-1}{1} \cdot \left(\frac{1}{K}\right) \cdot \left(1 - \frac{1}{K}\right)^{N-2} \cdot \left(\frac{\binom{n}{1} \cdot \binom{N-n}{N-2}}{\binom{N}{N-1}}\right). \quad (1)
 \end{aligned}$$

Όμοια η πιθανότητα να πέσει το σημειωμένο αρχείο 2 θέσεις είναι:

$$\begin{aligned}
 & P(\text{να πέσει το σημειωμένο αρχείο 2 θέσεις} \mid X = t) = \\
 & = \sum_{y=1}^t P(\text{να πέσει το σημειωμένο αρχείο 2 θέσεις} \mid X = t, Y = y) \cdot P(Y = y) = \\
 & = \binom{2}{2} \cdot \left(\frac{1}{K}\right)^2 \cdot \left(1 - \frac{1}{K}\right)^0 \cdot \left(\frac{\binom{n}{2} \cdot \binom{N-n}{0}}{\binom{N}{2}}\right) + \binom{3}{2} \cdot \left(\frac{1}{K}\right)^2 \cdot \left(1 - \frac{1}{K}\right)^1 \cdot \left(\frac{\binom{n}{2} \cdot \binom{N-n}{1}}{\binom{N}{3}}\right) + \dots \\
 & \dots + \binom{N-1}{2} \cdot \left(\frac{1}{K}\right)^2 \cdot \left(1 - \frac{1}{K}\right)^{N-3} \cdot \left(\frac{\binom{n}{2} \cdot \binom{N-n}{N-3}}{\binom{N}{N-1}}\right). \quad (2)
 \end{aligned}$$

Συνεχίζουμε όμοια μέχρι $s = n-1$. Έτσι θα έχουμε ότι η πιθανότητα να πέσει το σημειωμένο αρχείο $n-1$ θέσεις είναι:

$$P(\text{να πέσει το σημειωμένο αρχείο } n-1 \text{ θέσεις} \mid X = t) =$$



$$\begin{aligned}
 &= \sum_{y=1}^t P(\text{να πέσει το σημειωμένο αρχείο } n-1 \text{ θέσεις} \mid X = t, Y = y) \cdot P(Y = y) = \\
 &= \binom{n-1}{n-1} \cdot \left(\frac{1}{K}\right)^{n-1} \cdot \left(1 - \frac{1}{K}\right)^0 \cdot \frac{\binom{n}{n-1} \cdot \binom{N-n}{0}}{\binom{N}{n-1}} + \binom{n}{n-1} \cdot \left(\frac{1}{K}\right)^{n-1} \cdot \left(1 - \frac{1}{K}\right)^1 \cdot \frac{\binom{n}{n-1} \cdot \binom{N-n}{1}}{\binom{N}{n}} + \dots \\
 &\dots + \binom{N-1}{n-1} \cdot \left(\frac{1}{K}\right)^{n-1} \cdot \left(1 - \frac{1}{K}\right)^{N-n} \cdot \frac{\binom{n}{n-1} \cdot \binom{N-n}{N-n}}{\binom{N}{n-1}}. \quad (n-1)
 \end{aligned}$$

6.2.4 Υπολογισμός ρυθμών μετάβασης δισδιάστατης CTMC

Στη συνέχεια, σύμφωνα με τους Ορισμούς 5.1 και 5.2 με τη διαφορά ότι η κάθε κατάσταση ορίζεται από δυο διαστάσεις (m,n) και ότι ο ρυθμός αιτημάτων των κόμβων είναι r , ορίζουμε ως ρυθμό μετάβασης από την κατάσταση (m,n) στην κατάσταση (m,n') όταν τα αντίγραφα θα αυξάνονται

$$q_{(m,n),(m,n')} = rP_S \cdot P(X = t) \text{ για } t = |n-n'| \text{ και } t=1,2,\dots,N-1 \quad (6.2.4.1)$$

και ως ρυθμό μετάβασης από την κατάσταση (m,n) στην κατάσταση (m,n') όταν τα αντίγραφα θα μειώνονται

$$q_{(m,n),(m,n')} = r \cdot P(A^C) \cdot P(B^C | A^C) \cdot P(C | B^C, A^C). \quad (6.2.4.2)$$

Άρα, σύμφωνα με τον Ορισμό 5.2 αλλά για δισδιάστατη CMTC έχουμε:

$$v_{m,n} = \sum_{m',n'} v_{m,n} P_{(m,n),(m',n')} = \sum_{m',n'} q_{(m,n),(m',n')}$$

Επομένως, ξεκινώντας να βρούμε τους ρυθμούς μετάβασης έχουμε:

- Στην κατάσταση $(1,1)$, η μόνη μετάβαση που μπορεί να κάνει εξαρτώμενη από τη δημοφιλία είναι να μεταβεί στην κατάσταση $(2,1)$. Οι μεταβάσεις που μπορεί να κάνει αυξάνοντας τα αντίγραφα, είναι στις καταστάσεις $(1,w)$ με $w=2,3,\dots,N$. Επομένως ο ρυθμός μετάβασης της θα είναι:

$$v_{1,1} = q_{(1,1),(2,1)} + \sum_{w=2}^N q_{(1,1),(1,w)}$$



$$v_{1,1} = \lambda \frac{(M-1)}{M} + \sum_{t=1}^{N-1} r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \cdot \frac{N-n}{N} \right) \cdot P(X=t).$$

- Στην κατάσταση $(1, N)$, όμοια η μόνη μετάβαση που μπορεί να κάνει η αλυσίδα εξαρτώμενη από τη δημοφιλία είναι να μεταβεί στην κατάσταση $(2, N)$. Όμως, οι μεταβάσεις που μπορεί να κάνει και να επηρεάσουν τον αριθμό των αντιγράφων, είναι μόνο για να μειωθούν τα αντίγραφα. Οι καταστάσεις αυτές είναι οι $(1, z)$ με $z = 1, 2, \dots, N-1$. Επομένως, ο ρυθμός μετάβασης της θα είναι:

$$v_{1,N} = q_{(1,N),(2,N)} + \sum_{z=1}^{N-1} q_{(1,N),(1,z)}$$

$$v_{1,N} = \lambda \cdot \left(\frac{M-1}{M} \right) + r \cdot \left(1 - \frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{M-K}{M} \right) \cdot \left(\frac{n}{N} \right) \sum_{s=1}^{N-1} P(\text{να πέσει το σημειωμένο αρχείο } s \text{ θέσεις} \mid X=t)$$

Όπου, $s \leq t$ και $t = 1, 2, \dots, N-1$.

Δηλαδή,

$$v_{1,N} = \lambda \cdot \left(\frac{M-1}{M} \right) + r \cdot \left(1 - \frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{M-K}{M} \right) \cdot \left(\frac{n}{N} \right) \sum_{s=1}^{N-1} \cdot \left(\sum_{t=1}^{N-1} \left(\binom{t}{s} \cdot \left(\frac{1}{K} \right)^s \cdot \left(1 - \frac{1}{K} \right)^{t-s} \right) \cdot \left(\frac{\binom{N}{s} \cdot \binom{N-N}{t-s}}{\binom{N}{t}} \right) \right)$$

Το άθροισμα στην παραπάνω ισότητα έχει αναλυθεί πιο πάνω.

- Στην κατάσταση (m, n) , οι μεταβάσεις που μπορεί να γίνουν επηρεασμένες από την δημοφιλία είναι προς τις καταστάσεις $(m+1, n)$ και η $(1, n)$. Ακόμα, στις μεταβάσεις όπου η αιτία είναι η αλλαγή του αριθμού των αντιγράφων, έχουμε και δεξιά μετακίνηση και αριστερή. Δηλαδή, μπορεί να η αλυσίδα μεταβεί στη κατάσταση (m, z) με $z = 1, 2, \dots, n-1$ και στη κατάσταση (m, w) με $w = n+1, n+2, \dots, N$. Έτσι, ο ρυθμός μετάβασης της θα είναι:

$$v_{m,n} = q_{(m,n),(m+1,n)} + q_{(m,n),(1,n)} + \sum_{z=1}^{n-1} q_{(m,n),(m,z)} + \sum_{w=n+1}^N q_{(m,n),(m,w)}$$



$$v_{m,n} = \lambda \cdot \left(\frac{M-m}{M} \right) + \lambda \cdot \left(\frac{1}{M} \right) + r \cdot \left(1 - \frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{M-K}{M} \right) \cdot \left(\frac{n}{N} \right) \cdot \sum_{s=1}^{n-1} \left(\sum_{t=1}^{n-1} \left(\binom{t}{s} \cdot \left(\frac{1}{K} \right)^s \cdot \left(1 - \frac{1}{K} \right)^{t-s} \right) \cdot \frac{\binom{n}{s} \cdot \binom{N-n}{t-s}}{\binom{N}{t}} \right) + \sum_{k=1}^{N-n} r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \cdot \frac{N-n}{N} \cdot P(X=k) \right)$$

Στη παραπάνω ισότητα ισχύει $s \leq t$.

- Στην κατάσταση $(M,1)$, η μόνη μετάβαση που μπορεί να κάνει η αλυσίδα εξαρτώμενη από τη δημοφιλία είναι να μεταβεί στην κατάσταση $(1,1)$. Αντίθετα, οι μεταβάσεις που μπορεί να κάνει στον οριζόντιο άξονα είναι για κάθε κατάσταση, ανάλογα πόσο αυξάνεται ο αριθμός των αντιγράφων. Έτσι, ο ρυθμός μετάβασης της θα είναι:

$$v_{M,1} = q_{(M,1),(1,1)} + \sum_{w=2}^N q_{(M,1),(M,w)}$$
$$v_{M,1} = \frac{\lambda}{M} + \sum_{t=1}^{N-1} r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \cdot \frac{N-n}{N} \right) \cdot P(X=t)$$



$$\begin{aligned}
 v_{M,1} &= \frac{\lambda}{M} + r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{N-1}{N} \right) \cdot \left(1 - \frac{\binom{N-1}{1}}{\binom{N}{1}} \right) + r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{N-1}{N} \right) \cdot \left(\frac{\binom{N-1}{1} \binom{N-2}{1}}{\binom{N}{1}} \right) + \\
 &+ \dots + r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{N-1}{N} \right) \cdot \left(1 - \frac{\binom{N-1}{1}}{\binom{N}{1}} \right) = \\
 &= \frac{\lambda}{M} + r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{N-1}{N} \right) \cdot \left(1 - \frac{N-1}{N} \right) + r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{N-1}{N} \right) \cdot \left(\frac{N-1}{N} - \frac{N-2}{N} \right) + \\
 &+ \dots + r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{N-1}{N} \right) \cdot \left(\frac{1}{N} \right) = \\
 &= \frac{\lambda}{M} + (N-1) \cdot \left[r \cdot \left(\frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{N-1}{N} \right) \cdot \left(\frac{1}{N} \right) \right]
 \end{aligned}$$

- Στην κατάσταση (M,N) , η μετάβαση που γίνεται στον κάθετο άξονα και αφορά την δημοφιλία είναι μόνο στην κατάσταση $(1,N)$, ενώ στον οριζόντιο άξονα, οι μεταβάσεις είναι στις καταστάσεις (M,z) με $z = 1,2..N-1$. Άρα, ο ρυθμός μετάβασης της θα είναι:

$$v_{M,N} = q_{(M,N),(1,N)} + \sum_{z=1}^{N-1} q_{(M,N),(M,z)}$$

$$v_{1,N} = \frac{\lambda}{M} + r \cdot \left(1 - \frac{1}{\sum_{j=1}^M \frac{1}{j^a}} \right) \cdot \left(\frac{M-K}{M} \right) \cdot \left(\frac{n}{N} \right) \cdot \sum_{s=1}^{N-1} \cdot \left(\sum_{t=1}^{N-1} \left(\binom{t}{s} \cdot \left(\frac{1}{K} \right)^s \cdot \left(1 - \frac{1}{K} \right)^{t-s} \right) \cdot \left(\frac{\binom{N}{s} \cdot \binom{N-N}{t-s}}{\binom{N}{t}} \right) \right)$$



7

Κεφάλαιο 7^ο – Αριθμητική Επίλυση

Προκειμένου να αξιολογήσουμε το μοντέλο μας, έχοντας υπολογίσει τους ρυθμούς μετάβασης της δισδιάστατης CTMC, θα υπολογίσουμε την στατική κατανομή της αλυσίδας και θα την απεικονίσουμε δισδιάστατα. Προτού αναλύσουμε τα αποτελέσματα αυτής της αριθμητικής επίλυσης, θα αναφερθούμε λίγο στον ορισμό της στατικής κατανομής (steady state) της CTMC.

7.1 Steady State

Σε μια αλυσίδα Markov συνεχούς χρόνου, οι μεταβάσεις από μια κατάσταση σε μια άλλη μπορούν να συμβούν ανά πάσα στιγμή, και ο χρόνος μεταξύ των μεταβάσεων αυτών είναι εκθετικά κατανομημένος. Δεδομένου ότι η εκθετική κατανομή είναι χωρίς μνήμη, η μελλοντική έκβαση της διαδικασίας εξαρτάται μόνο από την παρούσα κατάσταση και δεν εξαρτάται από το πότε συνέβη η τελευταία μετάβαση ή ποια ήταν.

Ορισμός 7.1: Για μια αμείωτη CTMC με θετικές επαναλαμβανόμενες καταστάσεις, οι πιθανότητες να βρεθεί η αλυσίδα στις καταστάσεις τις συγκλίνουν σε ένα διάνυσμα σταθερών πιθανοτήτων (p_1, p_2, \dots) το οποίο είναι ανεξάρτητο από την αρχική κατάσταση. Περαιτέρω, το διάνυσμα αυτό είναι η μοναδική λύση του ακόλουθου συστήματος εξισώσεων:

$$\begin{cases} p \cdot Q = 0 \\ \sum_i p_i = 1 \end{cases} \Leftrightarrow \begin{cases} \forall j, \sum_i p_i q_{ij} = 0 \\ \sum_i p_i = 1 \end{cases} \quad (7.1.1)$$

Συγκεκριμένα, Q ορίζουμε τον πίνακα ρυθμών μετάβασης. Κάθε στοιχείο q_{ij} του Q δείχνει τον ρυθμό μετάβασης από την κατάσταση i στην j για $i \neq j$. Ο χρόνος που χρειάζεται για μια μετάβαση στην κατάσταση j , δεδομένου ότι βρισκόμαστε στην κατάσταση i , είναι εκθετικά κατανομημένη με ρυθμό q_{ij} .

Ακόμα, για την επίτευξη της αριθμητικής επίλυσης χρειαζόμαστε και την εξίσωση ροής ισορροπίας (flow balance equation).

Η εξίσωση ισορροπίας ισούται με:



$$\sum_{i \neq j} p_j q_{ji} = \sum_{i \neq j} p_i q_{ij} \quad (7.1.2)$$

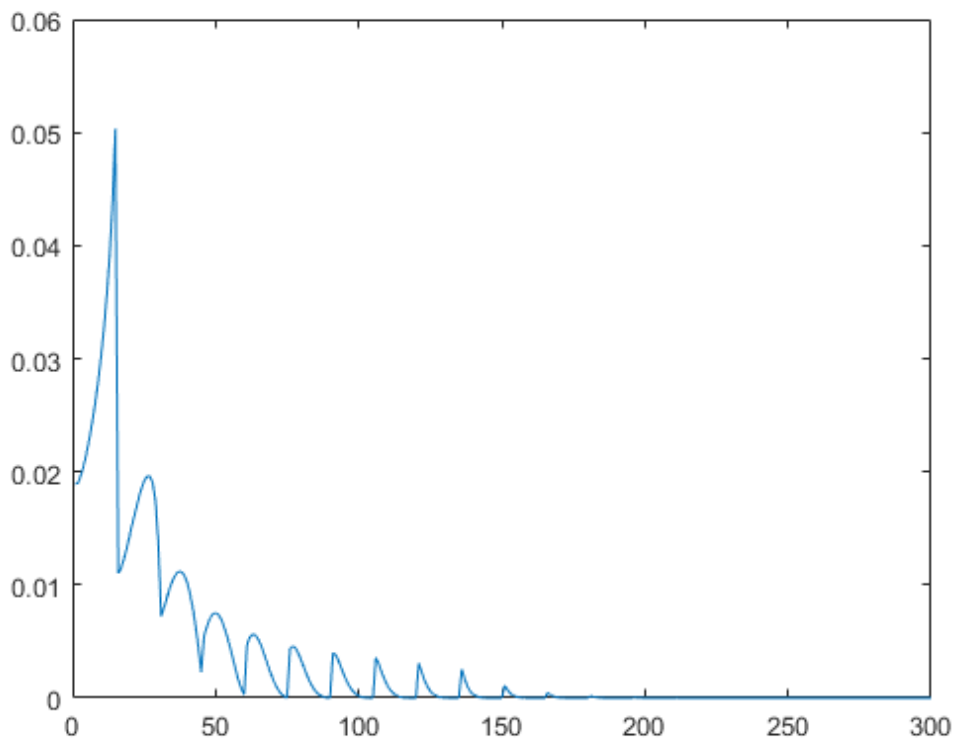
όπου $\sum_{i \neq j} p_j q_{ji}$ είναι η συνολική ροή που εισέρχεται στην κατάσταση i και

και $\sum_{i \neq j} p_i q_{ij}$ είναι η συνολική ροή που εξέρχεται από την κατάσταση i .

Η παραπάνω εξίσωση εξασφαλίζει ότι η συνολική ροή που εισέρχεται να είναι ίση με την συνολική ροή που εξέρχεται.

7.2 Αποτελέσματα Αριθμητικής Επίλυσης

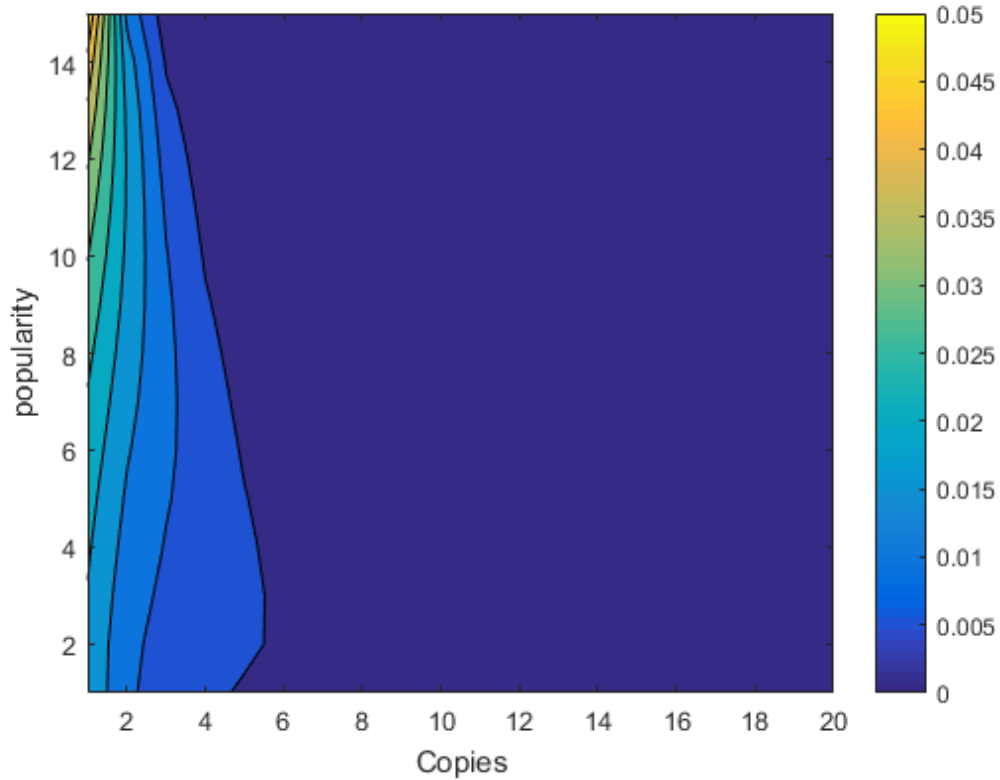
Στην αριθμητική επίλυση της αλυσίδας στη MATLAB, αρχικά ορίσαμε ο αριθμός των κόμβων να είναι $N=20$, τα αρχεία να είναι $M=15$ και το μέγεθος την μνήμης κάθε κόμβου να είναι $K=2$. Η επιλογή ήταν αυθαίρετη. Έτσι λύνοντας τις (7.1.1 και 7.1.2) με τις παραπάνω μεταβλητές δημιουργήσαμε ένα πίνακα με 300 μεταβάσεις στο οποίο παρουσιάζουμε τις ιδιοτιμές (p_i) (Σχήμα 7.1).



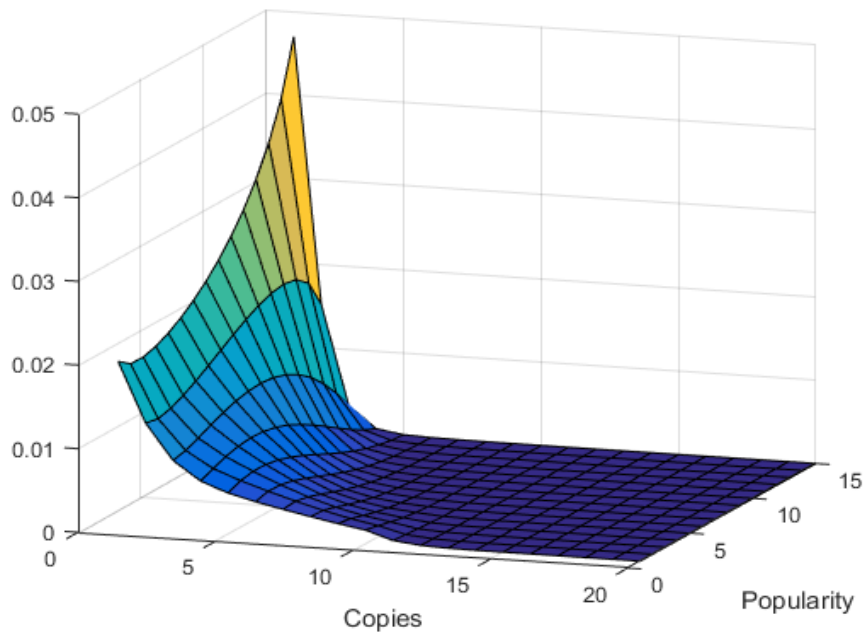
Σχήμα 7.1: Steady State της δισδιάστατης CTMC



Έπειτα απεικονίσαμε τον πίνακα αυτόν αρχικά σε δισδιάστατη μορφή έχοντας ως κάθετο άξονα την δημοφιλία και οριζόντιο τον αριθμό των αντιγράφων, και στην συνέχεια σε τρισδιάστατο (Σχήμα 7.2 και Σχήμα 7.3 αντίστοιχα).



Σχήμα 7.2: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων.



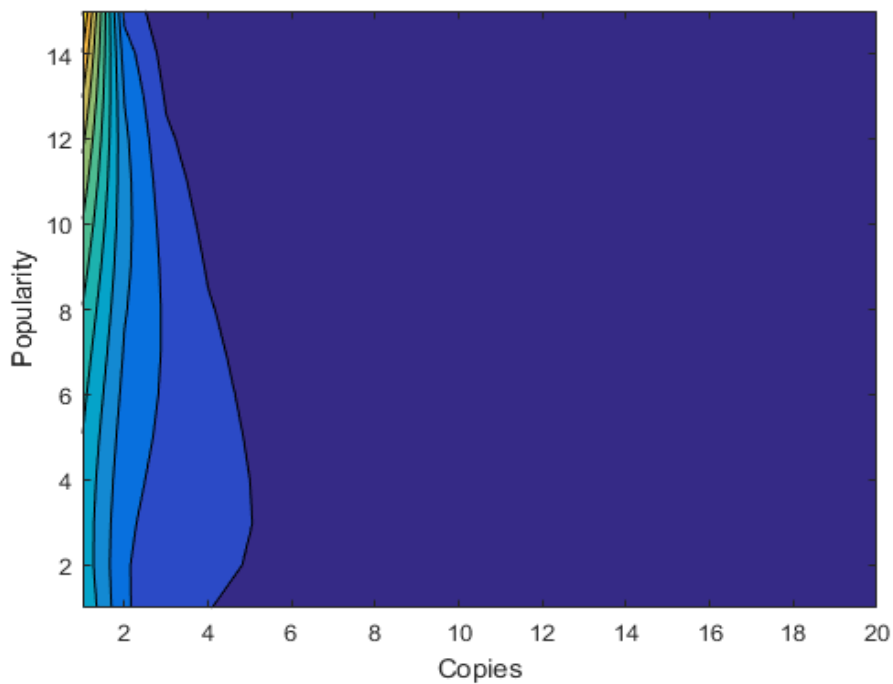
Σχήμα 7.3: Τρισδιάστατη μορφή Δημοφιλίας – Αντιγράφων



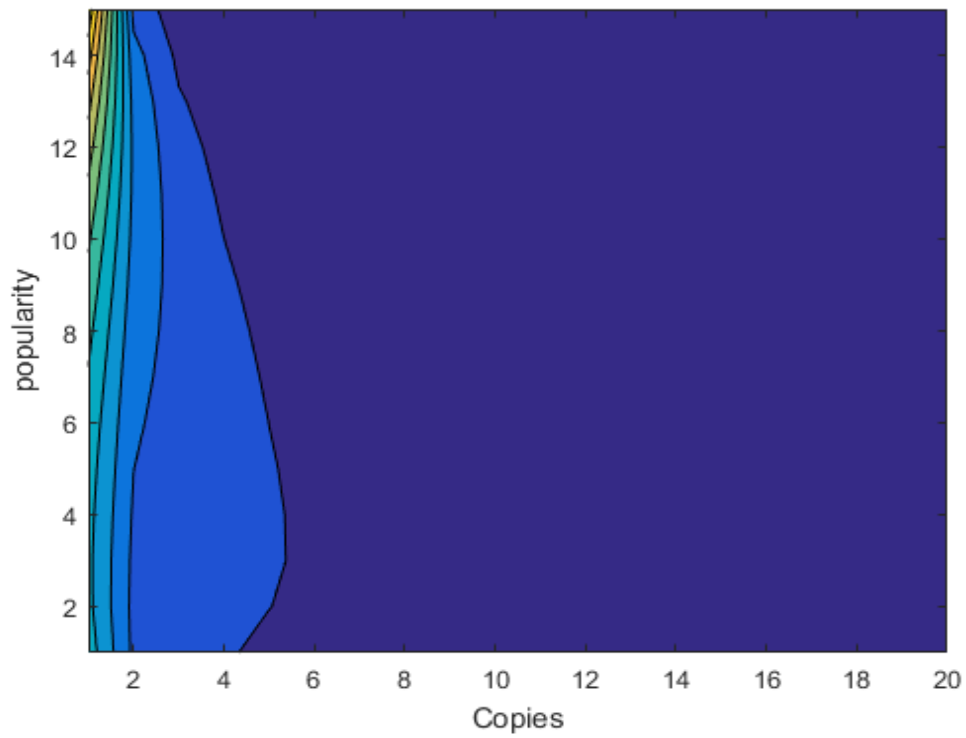
Το κύριο συμπέρασμα που παίρνουμε από τα παραπάνω σχήματα είναι ότι, όταν το αρχείο γίνεται δημοφιλές (πηγαίνει στον άξονα της δημοφιλίας προς το 1), ο αριθμός των αντιγράφων αυξάνεται και καθώς προχωράμε σε αρχεία με λιγότερη δημοφιλία, τα αντίγραφα μειώνονται. Δηλαδή παρατηρούμε στο Σχήμα 7.3 πως όταν ένα αρχείο είναι δημοφιλές (στην 1 ή 2 ή 3 θέση) η πιθανότητα τα αντίγραφά του να είναι 5 ως 6 είναι μεγάλη, ενώ τα αρχεία που βρίσκονται στις τελευταίες θέσεις δημοφιλίας, έχουν μεγάλη πιθανότητα να παραμείνουν ελάχιστα. Υπενθυμίζουμε πως αυτά αποτελούν αποτελέσματα με $K=2$.

Αλλάζοντας το μέγεθος της μνήμης κάθε κόμβου παρατηρούμε αρκετές αλλαγές.

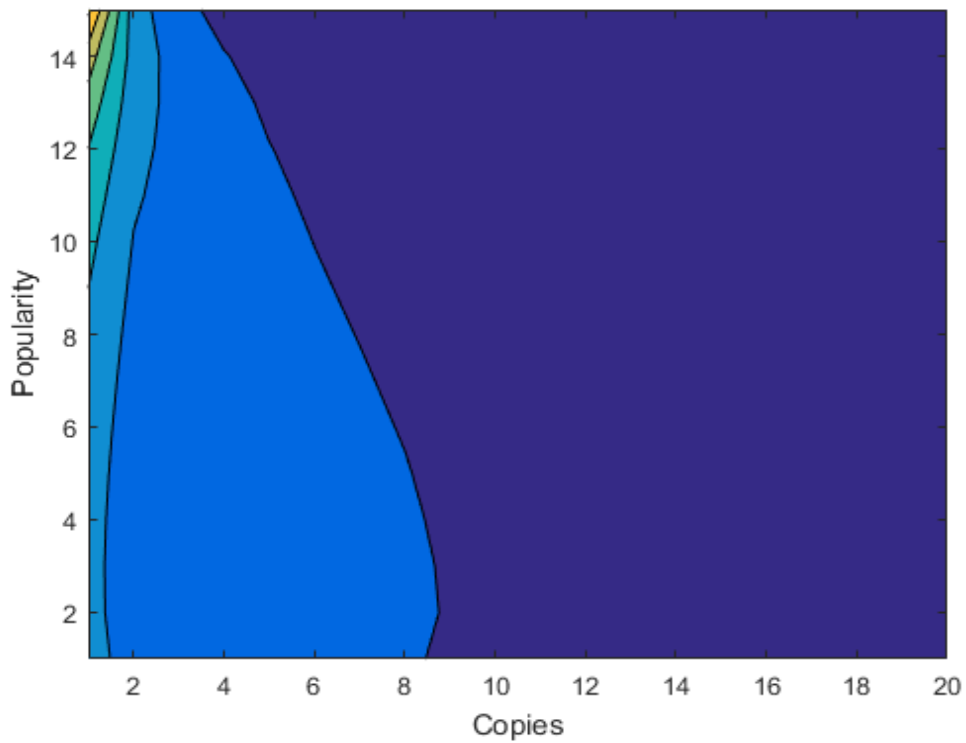
Στα παρακάτω σχήματα βλέπουμε τα αποτελέσματα της αριθμητικής επίλυσης για $N=20$, $M=15$ αλλά $K=3$ (Σχήμα7.4), 5 (Σχήμα7.5), 8 (Σχήμα7.6), 10 (Σχήμα7.7), 12 (Σχήμα7.8), 14 (Σχήμα7.9).



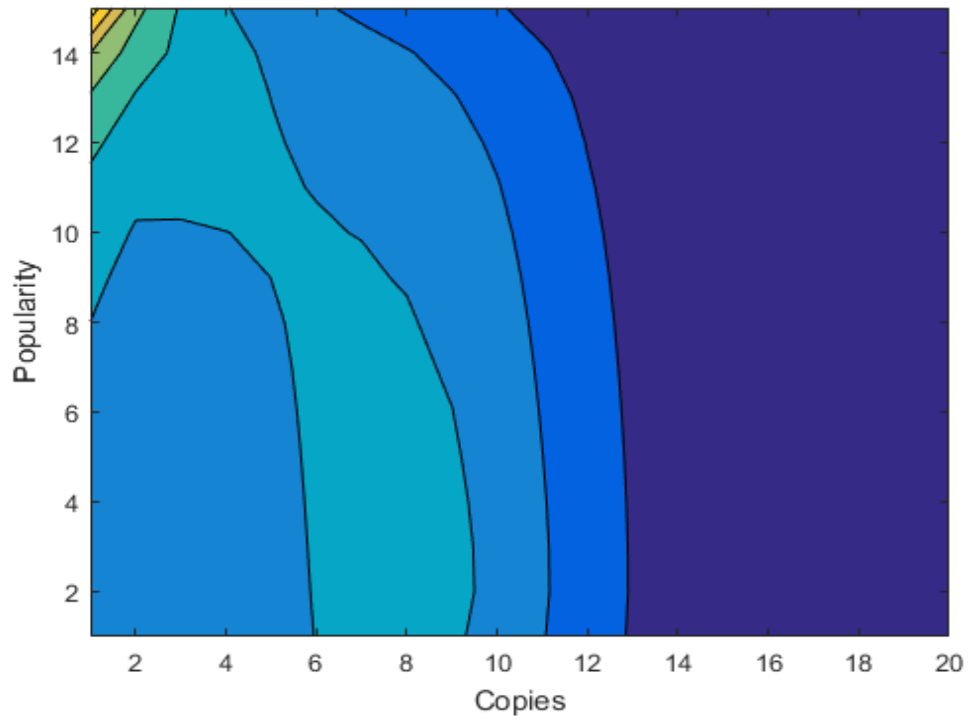
Σχήμα 7.4: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων με $K=3$.



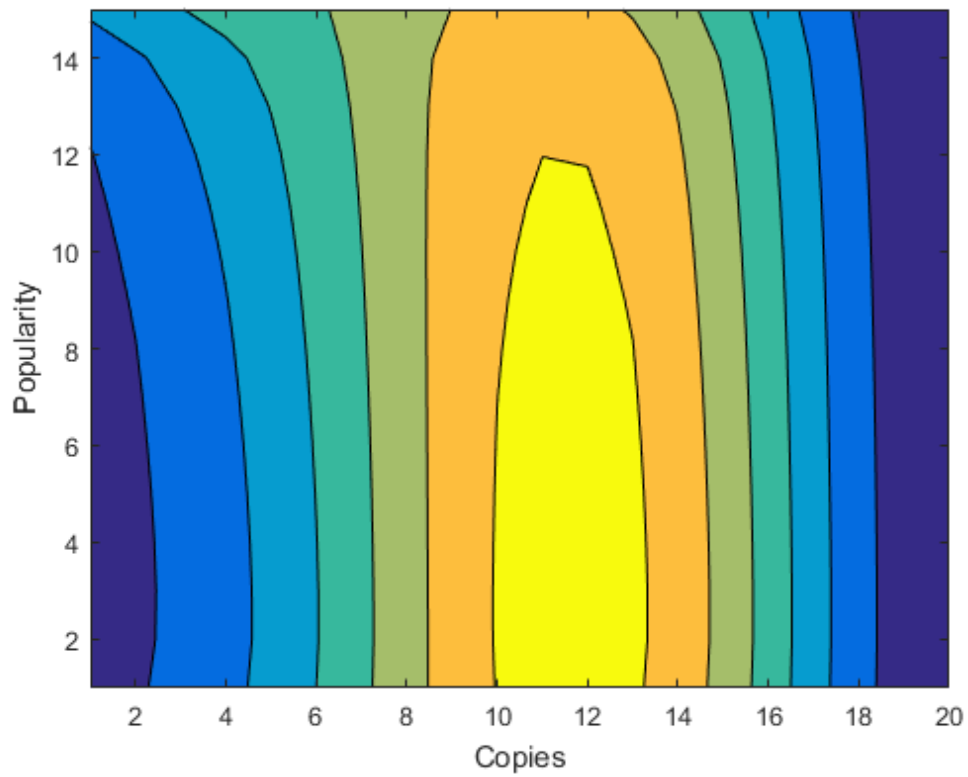
Σχήμα 7.5: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων με $K=5$.



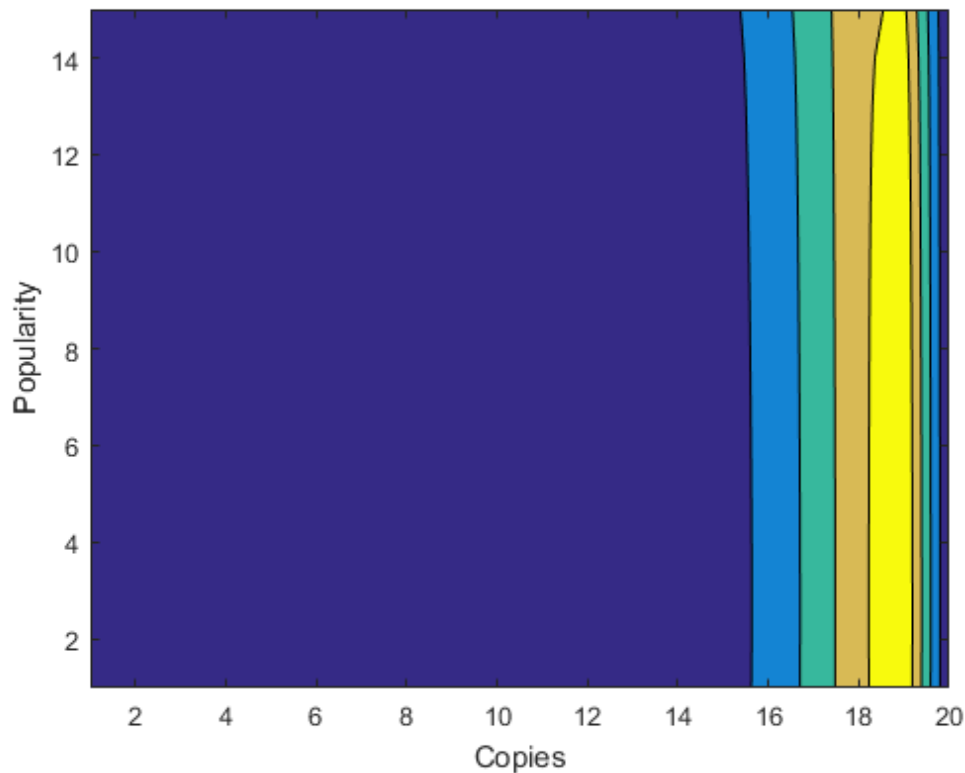
Σχήμα 7.6: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων με $K=8$.



Σχήμα 7.7: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων με $K=10$.



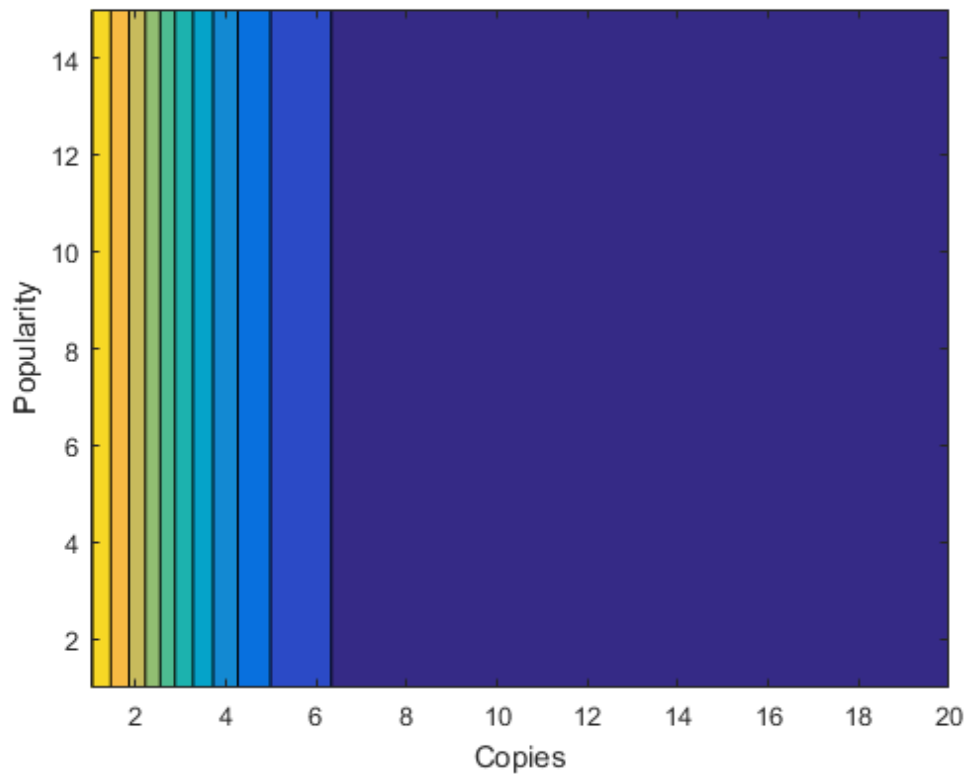
Σχήμα 7.8: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων με $K=12$.



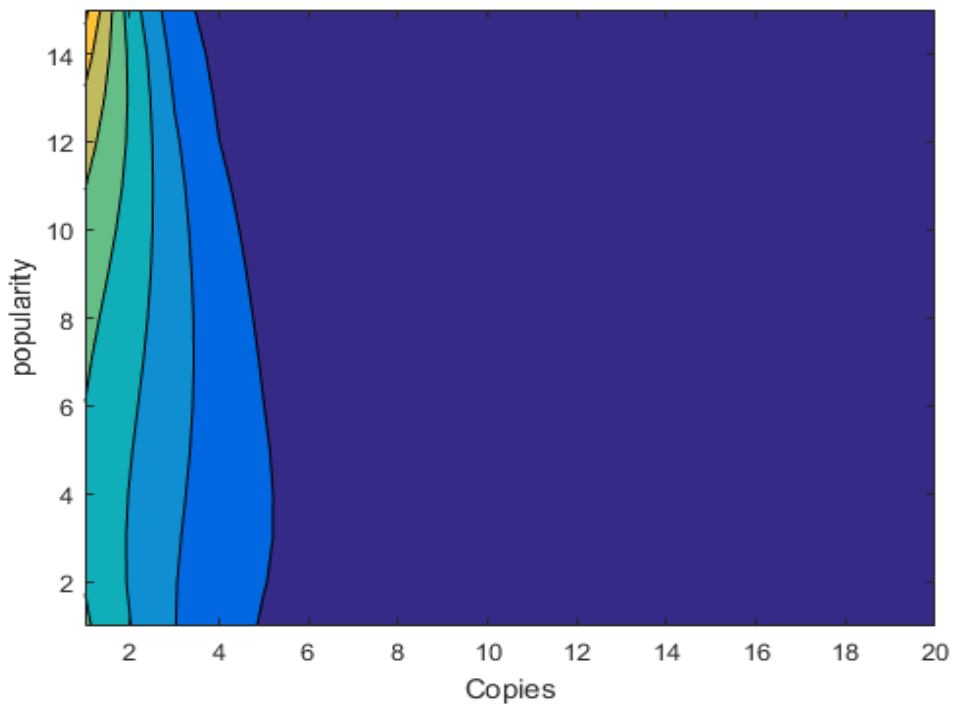
Σχήμα 7.9: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων με $K=14$.

Παρατηρώντας τα παραπάνω σχήματα, βλέπουμε ότι καθώς αυξάνουμε το μέγεθος της μνήμης του κάθε κόμβου, οι πιθανότητες να αυξηθεί ο αριθμός των αντιγράφων των δημοφιλών αρχείων αυξάνεται και αυτός αναλογικά με το K . Δηλαδή, στο Σχήμα 7.7 όπου το $K=10$, παρατηρούμε ότι στα δημοφιλή αρχεία, ο αριθμός των αντιγράφων κινείται από 11 ως 13 αντίγραφα, κατάσταση την οποία θα περιμέναμε.

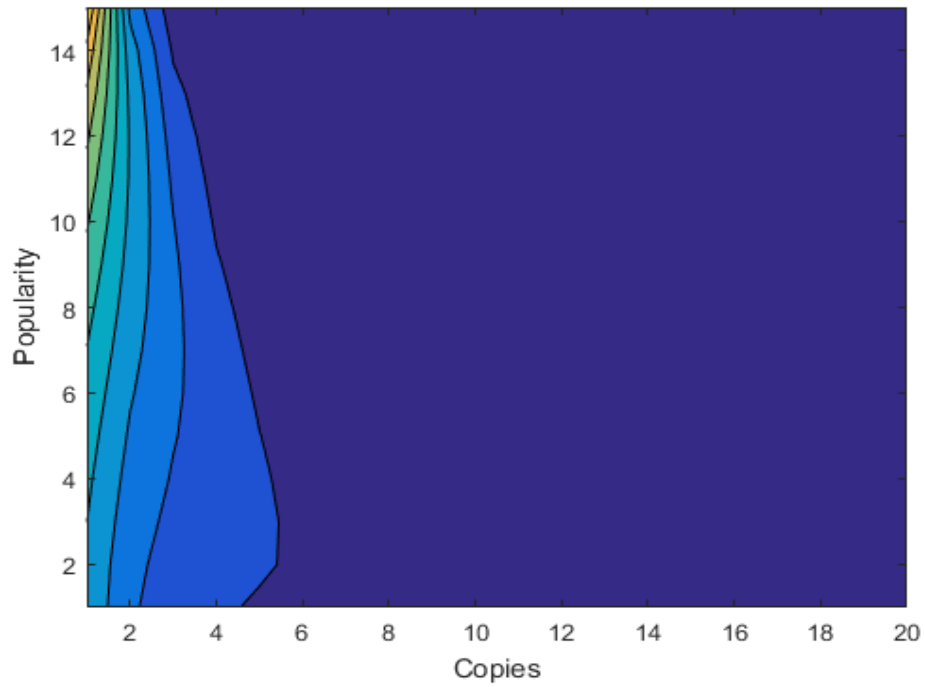
Ακόμα μια παρατήρηση που πρέπει να αναφέρουμε, είναι η συμπεριφορά της αλυσίδας για διαφορετικές τιμές του a (η παράμετρος της Zipf κατανομής). Έτσι, αν πάρουμε $a=0$, παρατηρούμε ότι η αλυσίδα χάνει την εκθετική της ιδιότητα, δηλαδή η πιθανότητα να επιλεγεί το m – πιο δημοφιλές αρχείο είναι ίση με το να επιλεγεί οποιαδήποτε άλλο αρχείο (Σχήμα 7.10). Αντίθετα, όσο πιο μεγάλο είναι το a , τόσο πιο μεγάλη είναι η συσχέτιση μεταξύ της δημοφιλίας και του πλήθους των αντιγράφων (Σχήμα 7.11 & 7.12).



Σχήμα 7.10: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων με $a=0$ ($N=20$, $M=15$, $K=2$).



Σχήμα 7.11: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων με $a=1$ ($N=20$, $M=15$, $K=2$).



Σχήμα 7.12: Δισδιάστατη μορφή Δημοφιλίας – Αντιγράφων με $a=5(N=20, M=15, K=2)$.



Κεφάλαιο 8^ο – Συμπεράσματα και Περαιτέρω Βελτιώσεις

8.1 Συμπεράσματα

Η παρούσα εργασία είχε ως στόχο τη μοντελοποίηση του αριθμού των αντιγράφων ενός αρχείου από κοινού με την δημοφιλία του. Έτσι κατασκευάσαμε μια δισδιάστατη CTMC, στην οποία κάνοντας την αριθμητική επίλυσή της σε MATLAB βγάλαμε ενδιαφέροντα συμπεράσματα. Κυρίως είδαμε, ότι η αλυσίδα μοντελοποιεί περιπτώσεις κατά τις οποίες το πλήθος των αντιγράφων ενός αρχείου μεταβάλλεται κατ' απόλυτο τιμή μεγαλύτερη της μονάδας.

Όπως αποδείχθηκε από τα αποτελέσματα της αριθμητικής επίλυσής, καθώς αυξήσαμε το μέγεθος της μνήμης ενός κόμβου, τα δημοφιλής αρχεία έχουν τη τάση να αυξάνουν τα αντίγραφα τους σε σχέση με τα μη δημοφιλή. Επίσης παρατηρήσαμε τη συμπεριφορά της αλυσίδας για διαφορετικές τιμές του a (η παράμετρος της Zipf κατανομής). Δηλαδή, όσο πιο μεγάλο είναι το a , τόσο πιο μεγάλη είναι η συσχέτιση μεταξύ της δημοφιλίας και του πλήθους των αντιγράφων.

Πάρα ταύτα, υπάρχουν πολλά περιθώρια για βελτίωση των αποτελεσμάτων και δοκιμή επιπλέον αλγόριθμων. Επίσης, ο δικός μας αλγόριθμος υλοποιήθηκε σε τοπολογία δακτυλίου.

8.2 Περαιτέρω Βελτιώσεις

Δεδομένου ότι η υλοποίηση μας βασίζεται σε τοπολογία δακτυλίου, ο αλγόριθμος θα μπορούσε να δοκιμαστεί και σε άλλες τοπολογίες και αφού εξαχθούν τα αποτελέσματα, να συγκριθούν και να αξιολογηθούν. Επίσης, θα μπορούσε να διαμορφωθεί η μοντελοποίηση χωρίς να υπάρχουν κάποιες υποθέσεις που την απλοποιούν. Για παράδειγμα, όταν ζητηθεί το σημειωμένο αρχείο να μην το αποθηκεύουν υποχρεωτικά οι κόμβοι της διαδρομής.

Ακόμα, υποθέτουμε ότι τα αρχεία μας δεν χάνονται (δηλαδή όταν υπάρχει μόνο ένα αντίγραφο ενός αρχείου, δεν μπορεί να αφαιρεθεί από την μνήμη), άρα μια εναλλακτική θα ήταν να δοκιμαστεί η μοντελοποίηση σε ρεαλιστικό δίκτυο.

Εντέλει, θα μπορούσαν να εισαχθούν διάφοροι παράγοντες που θα περιέπλεκαν αρκετά τη μοντελοποίηση, όπως όταν χάνεται ένα αρχείο να μπορεί να εισαχθεί πάλι από κάποιο εξωτερικό παράγοντα με κάποιο κόστος.



Παράρτημα

Παρακάτω περιγράφεται ο κώδικας της αριθμητικής επίλυσης της αλυσίδας του προβλήματος σε MATLAB.

```
% INPUT PARAMETERS
% =====
% N: the number of nodes
% M: the number of files
% K: cache size
% l: arrival rate of popularity
% r: arrival rate of request
% a: parameter of Zipf distribution

N=20;
M=15;
K=2;
l=3;
r=2;
a=5;

SS=steady_state( N,M,K,l,r,a);

*****

% function steady_state = steady_state( N,M,K,l,r,a)
function steady_state= steady_state( N,M,K,l,r,a)

A=zeros(M*N,M*N);

for n=1:N,
    for m=1:M,
        i=(n-1)*M+m;
        KK=v(N,M,K,l,r,a,n,m);
        for n1=1:N,
            for m1=1:M,
                j=(n1-1)*M+m1;
                A(i,j)=q(N,M,K,l,r,a,n1,m1,n,m)/KK;
            end;
        end;
    end;
end;

[V,D]=eig(A);
disp(['First eigenvalue is' D(1,1)]);
steady_state=V(:,1)/sum(V(:,1));
plot(steady_state)

*****
```



```
% function q = q( N,M,K,l,r,a,n,m,n1,m1 )
%
% This function calculates the transition rates q from state (m,n) to
state
% (m1,n1)
%
% INPUT PARAMETERS
% =====
% N: the number of nodes
% M: the number of files
% K: cache size
% l: arrival rate of popularity
% r: arrival rate of request
% a: parameter of Zipf distribution
% n: current number of file copies
% m: current popularity of tagged file
% n1: new current number of file copies
% m1: new current popularity of tagged file

function q = q( N,M,K,l,r,a,n,m,n1,m1 )

k=abs(n1-n); %
s=abs(n-n1);%
R=pmfX(k,N,n); % This function calculates the probability that the
copies will be increased by k
D=pmfS(s,N,n,K); % This function calculates the probability that the
copies will be decreased by s
G=zipf(M,a,m); %%This function calculates the Zipf distribution

if (n==n1)&& (m==m1);
    q=0;
elseif (m~=m1) && (n~=n1)
    q=0;
elseif (m1==(m+1)&& m1~=1) && (n==n1);
    q=0;
elseif (n==n1==m==1) && (m1==2);
    q=l*( (M-1)/M );
elseif (m~=1) && (m1==1) && (n==n1);
    q=l*( 1/M );
elseif (m~=M) && (m1==m+1)&& (n==n1);
    q=l*( (M-m)/M );
elseif (m==m1) && (n<n1);
    q=r*G*( (N-n)/N ) *R;
elseif (m==m1) && (n>n1);
    q=r*D*( 1-G ) * ( (M-K)/M ) * (n/N);
end;

end

*****
```



```
% This function calculates the probability that the copies will be
decreased by s
% s the number of positions which it losing
% N the number of nodes
% n the number of nodes which have the tagged file
% K cache size
% THIS IS FOR THE RING
```

```
function D = pmfS( s,N,n,K )
```

```
F=0;
G=0;
if s>n
    D=0;
else
    for t=1:(N/2)
        if (s>t)
            G=0;
            D=G;
        elseif (N-n)>=(t-s);
            b_1=nchoosek(t,s);
            b_2=nchoosek(n,s);
            b_3=nchoosek(N-n,t-s);
            b_4=nchoosek(N,t);

            B(t)=(b_1)*((1/K)^s)*(((K-1)/K)^(t-
s))*((b_2)*(b_3))/(b_4));

            F=sum(B);
            D=F;
        else
            D=G+F;
        end
    end
end
end
```

```
*****
```

```
% This function calculates the probability that the copies will be
increased by k
% N the number of nodes
% n the number of nodes which have the tagged file
% k the distance of node 1 from the nearest node which have the
tagged file
% THIS IS FOR THE RING
```

```
function R = pmfX(k,N,n )
```

```
if (N-2*k)>=n;
    c_1=nchoosek((N-2*k),n);
    c_2=nchoosek(N,n);
    c_3=nchoosek(N-2*k+2,n);

    if k>(N/2)
        disp('k>=N/2');
        R=0;
    end
end
```



```
elseif k==1
    R=1-(c_1/c_2);
elseif (k~=1)
    R=(c_3/c_2)-(c_1/c_2);
end
else
    R=0;
end
end

*****

%This function calculates the Zipf distribution
% M: the number of files
% a: parameter of Zipf distribution
% m: current popularity of tagged file

function G = zipf( M,a,m)

A=0;

for i=1:M
    A=A+1/(i^a);
end
G=(1/(m^a))/A;

end

*****

% function v = v( N,M,K,l,r,a,n,m)
%
% This function calculates the rate with which we leave state (m,n)
%
% INPUT PARAMETERS
% =====
% N: the number of nodes
% M: the number of files
% K: cache size
% l: arrival rate of popularity
% r: arrival rate of request
% a: parameter of Zipf distribution
% n: current number of file copies
% m: current popularity of tagged file

function v = v( N,M,K,l,r,a,n,m)

v=0;

for n1=1:N,
    for m1=1:M,
        v=v+ q(N,M,K,l,r,a,n,m,n1,m1);
    end;
end;
```



Παραπομπές

- [1] W. Jiang, S. Ioannidis, L. Massoulie and F. Picconi, “*Orchestrating massively distributed CDNs*,” in CoNEXT, 2012.
- [2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in CoNEXT 2009.
- [3] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, “Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems,” IEEE/ACM Trans. on Netw., vol. 17(5), pp. 1357–1370, 2009.
- [4] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, “*Impact of traffic mix on caching performance in a content-centric network*,” in NOMEN, 2012.
- [5] E. Coffman and P. Denning, “*Operating Systems Theory*,” Prentice-Hall, 1973.
- [6] S. Traverso, M. Ahmed, M. Garetto, P. Giaccone, E. Leonardi, and S. Niccolini, “*Temporal locality in today’s content caching: Why it matters and how to model it*,” CCR, vol. 43(5), pp. 5–12, Nov. 2013.
- [7] Cisco, “*Cisco visual networking index: Global mobile data traffic forecast update, 2016-2020*,” Cisco, Tech. Rep., Feb. 2016.
- [8] E. Rosensweig, J. Kurose, and D. Towsley. “*Approximate models for general cache networks*.” In Proc. IEEE INFOCOM 2010, pages 1–9, Mar. 2010.
- [9] I. Psaras, R. G. Clegg, R. Landa, W. K. Chai, and G. Pavlou. “*Modelling and evaluation of CCN-caching trees*.” In Proc. IFIP NETWORKING 2011, pages 78–91. Springer-Verlag, 2011.
- [10] D. Rossi and G. Rossini. “*On sizing CCN content stores by exploiting topological information*.” In Proc. 2012 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pages 280–285, Mar. 2012
- [11] C. Fricker, P. Robert, J. Roberts, and N. Sbihi, “*Impact of traffic mix on caching performance in a content-centric network*,” Feb. 2012.
- [12] D. Rossi and G. Rossini, “*Caching performance of content centric networks under multi-path routing (and more)*,” Telecom ParisTech, Tech. Rep., 2011.



- [13] Valentina Martina, Michele Garetto and Emilio Leonardi, “*A unified approach to the performance analysis of caching systems,*” in ACM TOMPECS 20016.
- [14] G. Rossini and D. Rossi, “*Coupling caching and forwarding: Benefits, analysis, and implementation,*” in ICN ’14, (New York, NY, USA), pp. 127–136, ACM, 2014.
- [15] Athanasios Katsiolis “*Markov Chain Modelling of Caching Networks with Time – Varying Popularities,*” B.Sc. Thesis, AUEB, 2015.
- [16] César Bernardini, Thomas Silverston, Olivier Festor, “*Popularity-based caching strategy for content centric networks,*” in 2013 IEEE International Conference on Communications (ICC), pages 3619-3623, 2013
- [17] W. K. Chai, D. He, I. Psaras, and G. Pavlou, “*Cache ”less for more” in information-centric networks,*” in Networking (1), 2012, pp. 27–40.