



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΔΡΟΜΟΛΟΓΗΣΗ ΣΕ ΥΠΕΡΚΕΙΜΕΝΑ ΔΙΚΤΥΑ
ROUTING IN OVERLAY NETWORKS**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Π. Συκαράς

**Επιβλέπων : Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.**

Αθήνα, Δεκέμβριος 2008



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΔΡΟΜΟΛΟΓΗΣΗ ΣΕ ΥΠΕΡΚΕΙΜΕΝΑ ΔΙΚΤΥΑ
ROUTING IN OVERLAY NETWORKS**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Δημήτριος Π. Συκαράς

**Επιβλέπων : Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.**

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την

.....
**Ευστάθιος Συκάς
Καθηγητής Ε.Μ.Π.**

.....
**Μιχαήλ Θεολόγου
Καθηγητής Ε.Μ.Π.**

.....
**Γεώργιος Στασινόπουλος
Καθηγητής Ε.Μ.Π.**

Αθήνα, Δεκέμβριος 2008

.....

Δημήτριος Π. Συκαράς

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

Copyright © Δημήτριος Π. Συκαράς, 2008

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας είναι η μελέτη της δρομολόγησης σε υπερκείμενα δίκτυα. Αρχικά παρουσιάζεται μια περιγραφή της δομής και της λειτουργίας των υπερκείμενων δικτύων, τα οποία χρησιμοποιούν την ήδη υπάρχουσα δομή του υποκείμενου δικτύου για μια πιο αποκεντρωμένη λειτουργία και τα πλεονεκτήματα που αυτή παρουσιάζει. Γίνεται μια περιγραφή της αρχιτεκτονικής που χρησιμοποιούν τα υπερκείμενα δίκτυα CAN, Chord, Pastry, Tapestry, Mithos και συγκρίνονται τα μεταξύ τους πλεονεκτήματα και μειονεκτήματα.

Ακόμα γίνεται και μια πρόταση νέας τεχνικής δρομολόγησης υπερκείμενων δικτύων και σύγκρισή της με τα προηγούμενα. Τέλος γίνεται προσομοίωσή της λειτουργίας της με το λογισμικό ανοιχτού κώδικα Oversim και του OMNeT++.

Λέξεις-κλειδιά

Δρομολόγηση, υπερκείμενα δίκτυα, CAN, Chord, Pastry, Tapestry, Mithos, OverSim, OMNeT++

Abstract

The scope of this diploma thesis is the study of routing in overlay networks. Initially a description of the structure and function of overlay networks is presented. Overlay networks use the preexisting structure of the underlying network for a decentralized function that has many advantages. The architecture being used by the overlay networks CAN, Chord, Pastry, Tapestry and Mithos is described and the advantages and disadvantages between them are presented.

A technique for routing in overlay networks is being proposed, and compared with the previously described overlay networks. Finally, a simulation is being run with the open source code OverSim and OMNeT++.

Keywords

Routing, overlay networks, CAN, Chord, Pastry, Tapestry, Mithos, OverSim, OMNeT++.

Πίνακας Περιεχομένων

Πίνακας Περιεχομένων.....	7
1.Εισαγωγή.....	11
2.Υπερκείμενο δίκτυο CAN(Content Addressable Network).....	13
2.1 Σχεδιασμός υπερκείμενου δικτύου CAN.....	13
2.2 Δρομολόγηση σε ένα υπερκείμενο δίκτυο CAN.....	14
2.3 Δημιουργία του υπερκείμενου δικτύου CAN.....	15
2.4 Αναχώρηση κόμβων, επαναφορά και διατήρηση του δικτύου CAN.....	17
3.Chord: Μια κλιμακούμενη υπηρεσία αναζήτησης για εφαρμογές στο διαδίκτυο.....	19
3.1 Τα χαρακτηριστικά του πρωτοκόλλου Chord.....	20
3.2 Το βασικό πρωτόκολο Chord.....	22
3.2.1 Επισκόπηση.....	22
3.2.2 Συνεπής ανάμειξη.....	22
3.2.3 Κλιμακούμενη εύρεση κλειδιού	23
3.2.4 Είσοδος κόμβου.....	24
3.2.5 Αναχώρηση κόμβων και αναπαραγωγή.....	26
4. Pastry: Ένα κλιμακούμενο, αποκεντρωμένο υπερκείμενο δίκτυο για δρομολόγηση σε μεγάλης κλίμακας συστήματα.....	27
4.1 Σχεδιασμός του υπερκείμενου δικτύου Pastry.....	28
4.2 Κατάσταση κόμβων στο Pastry.....	29
4.3 Δρομολόγηση στο Pastry.....	30
4.4 Άφιξη και αναχώρηση κόμβων.....	31
4.5 Τοποθεσία.....	34
4.6 Τυχαία κατάρρευση κόμβων και διαχωρισμός του δικτύου.....	35

5. Tapestry: Μια δομή για εντοπισμό και δρομολόγηση ευρείας περιοχής.....	36
5.1 Το υπόβαθρο του αλγόριθμου Plaxton.....	36
5.1.1 Δρομολόγηση στο Plaxton.....	37
5.1.2 Εντοπισμός.....	38
5.1.3 Πλεονεκτήματα-Μειονεκτήματα.....	39
5.2 Οι λειτουργίες του υπερκείμενου δικτύου Tapestry.....	40
5.2.1 Δρομολόγηση στο Tapestry.....	40
5.2.2 Διαχείριση σφαλμάτων.....	41
5.2.2.1 Δρομολόγηση ανεκτική σε σφάλματα.....	41
5.2.2.2 Εντοπισμός ανεκτικός σε σφάλματα.....	42
5.2.3 Εναλλακτική δρομολόγηση.....	42
5.3 Δυναμικοί αλγόριθμοι.....	43
5.3.1 Δυναμική είσοδος κόμβου.....	43
5.3.1.1 Συμπλήρωση πίνακα γειτνίασης.....	44
5.3.1.2 Ενημέρωση γειτονικών κόμβων.....	44
5.3.2 Φορτίο εύρους ζώνης.....	45
6. Υπερκείμενο δίκτυο Mithos.....	46
6.1 Σχεδιασμός του Mithos.....	46
6.2 Εύρεση γειτονικών κόμβων.....	47
6.3 Ανάθεση αναγνωριστικού.....	47
6.4 Εγκατάσταση συνδέσμων.....	48
6.5 Επίτευξη συνδέσεων με βάση τα τεταρτημόρια.....	49
7. Σύνοψη και σύγκριση υπερκείμενων δικτύων.....	50
7.1 Σύνοψη και σύγκριση Tapestry-Pastry.....	50
7.2 Σύνοψη και σύγκριση CAN-Chord.....	52
7.3 Σύγκριση Chord-Pastry.....	53
7.4 Σύνοψη Mithos.....	53

7.4.1 Σύγκριση CAN-Mithos.....	53
8.Προτεινόμενο υπερκείμενο δίκτυο.....	54
8.1 Επικοινωνία με βάση σημείο συνάντησης.....	54
8.1.1 Λειτουργία της παραλλαγής του υπερκείμενου δικτύου Pastry.....	55
8.2 Δημόσιες και ιδιωτικές σκανδάλεις.....	57
8.3 Στοιβα αναγνωριστικών.....	58
8.4 Πλεονεκτήματα προτεινόμενου υπερκείμενου δικτύου.....	59
8.4.1 Κινητικότητα.....	59
8.4.2 Πολυεκπομπή(Multicast).....	60
8.5 Αποτελεσματικότητα δρομολόγησης.....	60
8.5.1 Αποφυγή σημείων συμφόρησης.....	61
8.5.2 Ανεκτικότητα σε σφάλματα.....	62
8.6 Ασφάλεια.....	63
8.6.1 Ανωνυμία.....	64
8.7 Εφαρμογές προτεινόμενου υπερκείμενου δικτύου.....	65
8.7.1 Σύθεση υπηρεσιών.....	65
8.7.2 Ετερογενής πολυεκπομπή.....	66
8.7.3 Πολυεκπομπή μεγάλης κλίμακας.....	66
9.Σύγκριση προτεινόμενου υπερκείμενου δικτύου.....	68
9.1 Ομοιότητες με Pastry.....	68
10.Περιβάλλον προσομοίωσης.....	70
10.1 OMNeT++.....	70
10.2 Πρόγραμμα προσομοίωσης OverSim.....	71
11.Προσομοίωση προτεινόμενου υπερκείμενου δικτύου με το OverSim.....	75
11.1 Υποκείμενη τοπολογία SimpleNetwork.....	75
11.2 Λογισμικές μονάδες.....	76

12.Προσομοίωση.....	78
12.1 Εντολές προσομοίωσης.....	78
12.2 Αποτελέσματα προσομοίωσης.....	80
12.2.1 Προσομοίωση για μεγάλο αριθμό κόμβων.....	80
12.2.2 Προσομοίωση με τη χρήση RandomChurn.....	83
12.2.3 Προσομοίωση με τη χρήση LifetimeChurn.....	86
12.2.4 Προσομοίωση με πολυεκπομπή.....	88
12.3 Σύγκριση αποτελεσμάτων προσομοίωσης.....	91
12.3.1 Προσομοίωση για μεγάλο αριθμό κόμβων.....	91
12.3.2 Προσομοίωση με τη χρήση RandomChurn.....	94
12.3.3 Προσομοίωση με τη χρήση LifetimeChurn.....	96
12.3.4 Προσομοίωση με πολυεκπομπή.....	99
12.4 Μεταβολή των δρομολογητών.....	107
12.5 Μεταβολή ρυθμού δεδομένων.....	113
13.Επίλογος.....	120
14.Βιβλιογραφία.....	121

1.Εισαγωγή

Υπερκείμενα δίκτυα(Overlay Networks)

Ένα υπερκείμενο δίκτυο είναι ένα δίκτυο κόμβων και λογικών συνδέσεων που δημιουργείται με βάση το υπάρχον δίκτυο με σκοπό την εφαρμογή μιας υπηρεσίας που δεν είναι διαθέσιμη στο υπάρχον δίκτυο.

Τα υπερκείμενα δίκτυα μπορούν να κατασκευαστούν για τη δρομολόγηση μηνυμάτων που δεν προσδιορίζονται από διεύθυνση IP με τη χρήση των κατανεμημένων πινάκων ανάμειξης. Για την υλοποίησή τους χρειάζεται η χρήση του κατάλληλου λογισμικού από τους κόμβους που το επιθυμούν και όχι από όλους τους κόμβους του υπάρχοντος δικτύου.

Πλεονεκτήματα

- 1) Δεν χρειάζεται χρησιμοποίηση νέου εξοπλισμού. Χρησιμοποιείται το υπάρχον δίκτυο. Χρειάζεται μόνο η εφαρμογή νέου λογισμικού στο ήδη υπάρχον λογισμικό.
- 2) Δεν χρειάζεται συνεχής ενεργοποίηση σε κάθε κόμβο. Μπορεί ένας κόμβος να μην μπορεί να αντιμετωπίσει τη μνήμη ή το εύρος ζώνης που απαιτείται από το υπερκείμενο δίκτυο.

Μειονεκτήματα

- 1)Προσθέτει επιβάρυνση. Χρειάζονται επιπλέον δεδομένα που μπορεί να μην είναι πάντα απαραίτητα.
- 2)Προσθέτει πολυπλοκότητα. Υπάρχουν πολλά επίπεδα λειτουργίας και συνεπώς μεγαλύτερη ανάγκη για επικοινωνία μεταξύ τους.

Κατανεμημένοι πίνακες ανάμειξης

Ένας κατανεμημένος πίνακας ανάμειξης είναι ένας πίνακας ανάμειξης, ο οποίος είναι κατανεμημένος σε μια ομάδα συνδεδεμένων κόμβων. Όπως ένας πίνακας ανάμειξης, περιέχει ζεύγη κλειδιών-τιμής. Η κύρια λειτουργία ενός κατανεμημένου πίνακα ανάμειξης είναι, δεδομένου ενός κλειδιού να επιστρέφει τη συσχετιζόμενη τιμή. Ακόμα, εκτελεί λειτουργίες όπως η εισαγωγή και η διαγραφή καταχωρήσεων.

Μια σημαντική ιδιότητα των κατανεμημένων πινάκων ανάμειξης είναι η διαχείριση ζευγών κλειδιών-τιμών σε μεγάλη κλίμακα. Έτσι μπορεί να είναι μεγάλος και ο αριθμός των κόμβων που χρησιμοποιούνται. Λόγω της περιορισμένης χωρητικότητας μνήμης και αποθήκευσης του κάθε κόμβου και σε συνδυασμό με το κόστος της εισαγωγής και της ανανέωσης των ζευγών κλειδιών – τιμών, δεν είναι δυνατή η αποθήκευση των τελευταίων σε κάθε κόμβο. Κάθε κόμβος είναι υπεύθυνος για ένα μέρος του συνόλου των ζευγών τα οποία και αποθηκεύει.

Κάθε κόμβος έχει τη δυνατότητα να αναζητά την τιμή που σχετίζεται για κάθε κλειδί. Αφού κάθε κόμβος περιέχει μόνο μέρος των ζευγών, οι αναζητήσεις δρομολογούνται κάθε φορά που ένας κόμβος λαμβάνει μια αίτηση για την οποία δεν είναι υπεύθυνος. Για αυτό το σκοπό κάθε κόμβος έχει έναν πίνακα δρομολόγησης που περιέχει δείκτες σε υπόλοιπους κόμβους του δικτύου. Έτσι μια αναζήτηση δρομολογείται στους κόμβους του δικτύου έως ότου να φτάσει στον κόμβο που είναι υπεύθυνος για ένα συγκεκριμένο κλειδί .

Οι κατανεμημένοι πίνακες ανάμειξης μπορούν να χρησιμοποιηθούν στην υλοποίηση των υπερκείμενων δικτύων. Τα υπερκείμενα δίκτυα αυτά ονομάζονται δομημένα υπερκείμενα δίκτυα. Υπάρχουν και υπερκείμενα δίκτυα που δεν χρησιμοποιούν κατανεμημένους πίνακες ανάμειξης και ονομάζονται μη δομημένα. Από τα υπερκείμενα που χρησιμοποιούν κατανεμημένους πίνακες ανάμειξης, τα κυριότερα είναι τα CAN, Chord, Pastry, Tapestry, Mithos τα οποία παρουσιάζονται και αναλύονται στη συνέχεια.

2. Υπερκείμενο δίκτυο CAN(Content Addressable Network)

Το Content Addressable Network [1] είναι ένα καταναμημένο υπερκείμενο δίκτυο που χρησιμοποιεί τη λειτουργία των καταναμημένων πινάκων ανάμειξης σε επίπεδα της τάξεως του διαδικτύου. Είναι ένα κλιμακούμενο υπερκείμενο δίκτυο , αφού οι κόμβοι διατηρούν ένα μικρό μέρος του ελέγχου που είναι ανεξάρτητο από τον αριθμό των κόμβων στο υπερκείμενο δίκτυο. Είναι ανεκτικό σε λάθη, δηλαδή απώλειες συνδέσεων κόμβων μπορούν να παρακαμφθούν, και καταναμημένο γιατί δεν απαιτείται κάποιος κεντρικός έλεγχος για τη λειτουργία του.

Η βασική λειτουργία που επιτελείται σε ένα Content Addressable Network είναι η εισαγωγή, εύρεση και διαγραφή ζευγών κλειδιών - τιμής. Το CAN αποτελείται από πολλούς εξατομικευμένους κόμβους. Κάθε κόμβος αποθηκεύει ένα μέρος του πίνακα ανάμειξης. Επιπλέον κάθε κόμβος περιέχει την πληροφορία για τους γειτονικούς του κόμβους και του μέρους του πίνακα ανάμειξης που έχουν αυτοί.

2.1 Σχεδιασμός υπερκείμενου δικτύου CAN

Στο CAN σχηματίζεται ένας εικονικός καρτεσιανός χώρος ο οποίος έχει, d -διαστάσεις που δημιουργούν μια σπείρα. Ο χώρος αυτός έχει μόνο λογική σημασία και χρησιμότητα και δεν υπάρχει σε φυσικό επίπεδο. Σε κάθε χρονική στιγμή, ο χώρος αυτός διαμοιράζεται δυναμικά ανάμεσα στους κόμβους του συστήματος έτσι ώστε κάθε κόμβος να έχει το δικό του μέρος του συνολικού χώρου.

Αυτός ο εικονικός καρτεσιανός χώρος χρησιμοποιείται για την αποθήκευση ζευγών κλειδιών - τιμής , σύμφωνα με την εξής διαδικασία : για να αποθηκευθεί ένα ζεύγος (K_1, V_1) , το κλειδί K_1 αντιστοιχίζεται σε ένα σημείο P στον χώρο σύμφωνα με τη συνάρτηση ανάμειξης. Το αντίστοιχο ζευγάρι κλειδιού – τιμής αποθηκεύεται στον κόμβο στον οποίο ανήκει ο χώρος που βρίσκεται το σημείο P . Για την εύρεση της τιμής που αντιστοιχεί στο κλειδί K_1 , κάθε κόμβος χρησιμοποιώντας τη συνάρτηση ανάμειξης μπορεί να φτάσει στο σημείο P από το κλειδί. Αν το σημείο P δεν βρίσκεται στον χώρο του αιτούντος κόμβου ή κάποιου από τους γειτονικούς του κόμβους, τότε η αίτηση δρομολογείται μέσα από την δομή του CAN έως ότου φτάσει στον κόμβο στον οποίο

ανήκει το σημείο P. Η αποτελεσματική δρομολόγηση είναι ένα σημαντικό στοιχείο του CAN.

Οι κόμβοι στο CAN αυτοοργανώνονται σε ένα υπερκείμενο δίκτυο που αντιπροσωπεύει τον εικονικό συντεταγμένο χώρο. Ένας κόμβος μαθαίνει και διατηρεί τις διευθύνσεις IP των κόμβων των οποίων η περιοχή συντεταγμένων εφάπτεται με τη δική του περιοχή. Αυτή η ομάδα των γειτονικών κόμβων λειτουργεί σαν πίνακας δρομολόγησης που επιτρέπει τη δρομολόγηση μεταξύ τυχαίων σημείων στο χώρο αυτό. Θα περιγραφούν τα τρία πιο σημαντικά κομμάτια του σχεδιασμού. Η δρομολόγηση στο δίκτυο CAN, η δημιουργία του υπερκείμενου συντεταγμένου χώρου και η διατήρηση του υπερκείμενου δικτύου CAN.

2.2 Δρομολόγηση σε ένα υπερκείμενο δίκτυο CAN

Η δρομολόγηση σε ένα δίκτυο CAN υλοποιείται ακολουθώντας το μονοπάτι στον καρτεσιανό χώρο από την πηγή στις συντεταγμένες προορισμού.

Ένας κόμβος στο δίκτυο CAN διατηρεί έναν πίνακα δρομολόγησης που περιέχει τη διεύθυνση IP και τον εικονικό συντεταγμένο χώρο κάθε άμεσου γειτονικού κόμβου. Σε ένα χώρο συντεταγμένων d διαστάσεων δύο κόμβοι είναι γειτονικοί αν οι συντεταγμένες τους ταυτίζονται για $d - 1$ διαστάσεις και είναι εφάπτομενες για μια διάσταση. Ένα μήνυμα στο CAN δίκτυο περιλαμβάνει τις συντεταγμένες προορισμού. Χρησιμοποιώντας τις συντεταγμένες του γειτονικού κόμβου, ένα μήνυμα προωθείται στον προορισμό του απλά ακολουθώντας τις συντεταγμένες που είναι πιο κοντά σε αυτόν.

Για ένα χώρο d διαστάσεων διαμοιρασμένο σε n ίσες ζώνες, το μέσο μήκος δρομολόγησης είναι $(d/4)(n^{1/d})$ μεταπηδήσεις και κάθε κόμβος διατηρεί $2d$ γειτονικούς κόμβους. Αυτό σημαίνει πως για έναν χώρο d διαστάσεων, μπορεί να αυξηθεί ο αριθμός των κόμβων με το μέσο μήκος δρομολόγησης να αυξάνει κατά $O(n^{1/d})$.

Μεταξύ δύο σημείων στο χώρο υπάρχουν πολλά διαφορετικά μονοπάτια. Έτσι, αν κάποιος από τους γειτονικούς κόμβους αποχωρήσει, η δρομολόγηση συνεχίζει από το επόμενο διαθέσιμο μονοπάτι. Αν ένας κόμβος χάσει όλους τους γειτονικούς του κόμβους σε μια συγκεκριμένη κατεύθυνση, τότε η δρομολόγηση μπορεί στιγμιαία να σταματήσει. Σε αυτή την περίπτωση, ένας κόμβος μπορεί να εντοπίσει τον κόμβο που είναι

πλησιέστερος στον προορισμό από τον ίδιο. Το μήνυμα τότε προωθείται στον κόμβο εκείνο, από όπου συνεχίζεται η προώθηση του μηνύματος.

2.3 Δημιουργία του υπερκείμενου δικτύου CAN

Ο χώρος του υπερκείμενου δικτύου CAN διαμοιράζεται ανάμεσα στους κόμβους που βρίσκονται στο σύστημα εκείνη τη χρονική στιγμή. Για να μπορεί να αυξηθεί το δίκτυο CAN, κάθε νέος κόμβος που εισέρχεται στο σύστημα πρέπει να αποκτήσει μέρος του συντεταγμένου χώρου. Η διαδικασία αυτή γίνεται με έναν υπάρχοντα κόμβο να χωρίζει το χώρο που του έχει ανατεθεί στο μισό, κρατώντας τον μισό και δίνοντας τον άλλο μισό στο νέο κόμβο. Η διαδικασία αυτή αποτελείται από τρία βήματα.

1. Ο νέος κόμβος πρέπει να βρει έναν κόμβο ο οποίος βρίσκεται ήδη στο δίκτυο CAN.
2. Στη συνέχεια, χρησιμοποιώντας τους μηχανισμούς δρομολόγησης του δικτύου CAN, πρέπει να βρει έναν κόμβο του οποίου η περιοχή θα διαμοιραστεί.
3. Τελικά, οι γειτονικοί κόμβοι της διαμοιρασμένης περιοχής πρέπει να ενημερωθούν για την αλλαγή, έτσι ώστε η δρομολόγηση να περιλαμβάνει τον νέο κόμβο.

Πρώτα ένας νέος κόμβος βρίσκει τις διευθύνσεις IP των κόμβων που βρίσκονται εκείνη τη στιγμή στο σύστημα. Η λειτουργία του δικτύου CAN δεν εξαρτάται από τις λεπτομέρειες αυτής της διαδικασίας. Ο βασικός μηχανισμός είναι όπως περιγράφεται στη συνέχεια. Ένα δίκτυο CAN έχει μια ονομασία DNS που οδηγεί στις IP διευθύνσεις ενός ή περισσότερων κόμβων οι οποίοι περιέχουν μέρος του συνόλου της λίστας των κόμβων του δικτύου CAN οι οποίοι πιστεύεται ότι βρίσκονται στο δίκτυο εκείνη τη στιγμή.

Έτσι, ένας κόμβος για να εισέλθει στο δίκτυο βρίσκει το όνομα κατά το DNS για να βρει τον κόμβο που έχει το μέρος της λίστας του συνόλου των κόμβων που βρίσκονται συνδεδεμένα στο δίκτυο.

Ο νέος κόμβος διαλέγει τυχαία ένα σημείο P στο χώρο και στέλνει ένα μήνυμα JOIN στο σημείο αυτό. Το μήνυμα αυτό στέλνεται στο δίκτυο CAN διαμέσω οποιουδήποτε

κόμβου CAN. Κάθε κόμβος CAN χρησιμοποιεί τον τρόπο δρομολόγησης του CAN δικτύου για να προωθήσει το μήνυμα έως ότου φτάσει στον κόμβο στην περιοχή του οποίου βρίσκεται το σημείο P.

Ο τρέχων κατειλημένος κόμβος χωρίζει την περιοχή που του έχει ανατεθεί στο μισό και αναθέτει το υπόλοιπο μισό στο νέο κόμβο. Η διαδικασία γίνεται υποθέτοντας μια συγκεκριμένη σειρά διαχωρισμού των διαστάσεων του χώρου που έχει ανατεθεί σε κάθε κόμβο, έτσι ώστε οι διαχωρισμένες περιοχές να μπορούν να επανενωθούν κατά την αποχώρηση των κόμβων. Για ένα χώρο 2 διαστάσεων, μια περιοχή θα χωριζόταν πρώτα κατά μήκος της X διάστασης, στη συνέχεια κατά μήκος της Y διάστασης και ούτω καθεξής. Τα ζεύγη κλειδιού – τιμής που ανήκουν στην περιοχή που χωρίζεται, μεταφέρονται και αυτά στο νέο κόμβο.

Έχοντας αποκτήσει τη δική του περιοχή, ο νέος κόμβος μαθαίνει από τον προηγούμενο κάτοχο, τις διευθύνσεις IP των νέων γειτονικών του κόμβων. Αυτοί είναι οι γειτονικοί κόμβοι του προηγούμενου κατόχου και ο προηγούμενος κάτοχος. Τελικά όλοι οι γειτονικοί κόμβοι των νέων και των προηγούμενων κόμβων πρέπει να ενημερωθούν για την αναδιάταξη του χώρου. Κάθε κόμβος στο σύστημα στέλνει ένα μήνυμα ενημέρωσης, το οποίο περιοδικά ανανεώνεται, με την παρούσα περιοχή που του έχει ανατεθεί σε όλους τους γειτονικούς του κόμβους. Αυτές οι ενημερώσεις εξασφαλίζουν ότι όλοι οι γειτονικοί κόμβοι θα ενημερωθούν έγκαιρα για τις οποιοσδήποτε αλλαγές.

Η προσθήκη ενός νέου κόμβου επηρεάζει μόνο ένα μικρό αριθμό των κόμβων που βρίσκονται στο δίκτυο. Ο αριθμός των γειτονικών κόμβων που διατηρεί ένας κόμβος εξαρτάται από τις διαστάσεις του συντεταγμένου χώρου και είναι ανεξάρτητος από το πλήθος των κόμβων που υπάρχουν ήδη στο σύστημα. Στην επόμενη εικόνα απεικονίζεται ως παράδειγμα, ένας χώρος 2 διαστάσεων με 5 κόμβους.



Εικόνα 1: Χώρος 2 διαστάσεων με 5 κόμβους

2.4 Αναχώρηση κόμβων, επαναφορά και διατήρηση του δικτύου CAN

Όταν ένας κόμβος αποχωρεί από το δίκτυο CAN, χρειάζεται να βεβαιωθεί ότι η περιοχή που καταλάμβανε θα μεταβιβαστεί στους υπόλοιπους κόμβους. Η διαδικασία είναι ένας κόμβος που αποχωρεί να παραδίδει την περιοχή του και τα ζεύγη κλειδιών – τιμές σε έναν από τους γειτονικούς του κόμβους. Αν η περιοχή ενός από τους γειτονικούς κόμβους μπορεί να συγχωνευτεί με την περιοχή ενός από τους κόμβους που αποχωρούν για να σχηματιστεί μια συμπαγής περιοχή, τότε γίνεται η συγχώνευση. Σε άλλη περίπτωση η περιοχή του κόμβου που αποχωρεί, παραδίδεται στον κόμβο με την μικρότερη περιοχή, ο οποίος θα διαχειρίζεται προσωρινά δύο περιοχές.

Το CAN πρέπει να είναι ανθεκτικό σε σφάλματα του δικτύου, όταν ένας ή περισσότεροι κόμβοι δεν είναι προσβάσιμοι. Αυτό το πρόβλημα αντιμετωπίζεται με την απ' ευθείας κατάληψη του χώρου από έναν γειτονικό κόμβο.

Στη φυσιολογική λειτουργία του δικτύου ένας κόμβος στέλνει περιοδικά μηνύματα σε κάθε γειτονικό κόμβο με τις συντεταγμένες της περιοχής του και με τους γειτονικούς του κόμβους και τις συντεταγμένες των περιοχών αυτών. Η παρατεταμένη απουσία μηνύματος ανανέωσης από έναν γειτονικό κόμβο σημαίνει την αποχώρησή του.

Όταν ένας κόμβος αντιληφθεί ότι ο γειτονικός του κόμβος έχει αποχωρήσει ενεργοποιεί τον μηχανισμό κατάληψης της περιοχής και ξεκινά το χρονόμετρο κατάληψης. Κάθε γειτονικός κόμβος θα το κάνει ξεχωριστά, με το χρονόμετρο να ξεκινά ανάλογα με το μέγεθος της περιοχής του. Όταν το χρονόμετρο σταματά, ένας κόμβος

στέλνει μήνυμα κατάληψης με το μέγεθος της περιοχής του στους γειτονικούς κόμβους του κόμβου που αποχώρησε. Αν ένας κόμβος λάβει μήνυμα ότι το μέγεθος της περιοχής άλλου κόμβου είναι μικρότερο από το μέγεθος της δικής του περιοχής σταματά το χρονόμετρό του. Αλλιώς στέλνει το δικό του μήνυμα κατάληψης. Έτσι επιλέγεται ένας γειτονικός κόμβος ο οποίος είναι ενεργός και του έχει ανατεθεί μικρή περιοχή.

Σε συγκεκριμένες περιπτώσεις που περιλαμβάνουν την ταυτόχρονη αποχώρηση πολλών κόμβων, γίνεται η καταγραφή του προβλήματος, αλλά λιγότεροι από τους μισούς γειτονικούς κόμβους είναι προσβάσιμοι. Σε αυτές τις συνθήκες αν ένας κόμβος καταλάβει μια άλλη περιοχή είναι δυνατό το δίκτυο CAN να γίνει ασταθές. Σε αυτές τις περιπτώσεις πριν εκκινήσει ο διορθωτικός μηχανισμός, ο κόμβος κάνει έναν έλεγχο εκτός της προβληματικής περιοχής, ψάχνοντας για ενεργούς κόμβους για να ενωθεί.

Τέλος, τόσο με την απλή διαδικασία αποχώρησης όσο και με τη διαδικασία άμεσης κατάληψης, το αποτέλεσμα είναι ένας κόμβος να έχει αποκτήσει περισσότερες από μια περιοχές. Για να προβλεφθεί περαιτέρω κατάτμηση του χώρου ενεργοποιείται ένας αλγόριθμος επανατοποθέτησης περιοχών έτσι ώστε το δίκτυο CAN τείνει σε κατάσταση μιας περιοχής ανά κόμβο.

3. Chord: Μια κλιμακούμενη υπηρεσία αναζήτησης για εφαρμογές στο Διαδίκτυο

Ένα βασικό πρόβλημα στις εφαρμογές υπερκείμενων δικτύων είναι η αποτελεσματική εύρεση ενός κόμβου που έχει αποθηκεύσει ένα συγκεκριμένο στοιχείο δεδομένων. Το Chord [2] είναι ένα πρωτόκολλο το οποίο αντιμετωπίζει αυτό το πρόβλημα και η λειτουργία του είναι πως δεδομένου ενός κλειδιού, το αντιστοιχίζει στον κατάλληλο κόμβο. Η εύρεση δεδομένων μπορεί να εφαρμοστεί στο Chord αντιστοιχίζοντας ένα κλειδί με κάθε στοιχείο δεδομένων και αποθηκεύοντας το ζευγάρι κλειδί – δεδομένων στον κόμβο στον οποίο αντιστοιχίζεται το κλειδί. Το Chord προσαρμόζεται αποτελεσματικά καθώς οι κόμβοι εισέρχονται και αποχωρούν από το δίκτυο και είναι αποτελεσματικό ακόμα και αν το σύστημα αλλάζει συνεχώς. Το Chord χρησιμοποιεί συνεπή ανάμειξη για να αντιστοιχίσει κλειδιά στους κόμβους του δικτύου.

Η συνεπής ανάμειξη είναι μια μέθοδος που παρέχει λειτουργία με πίνακες ανάμειξης με τέτοιο τρόπο ώστε η προσθήκη ή αφαίρεση ενός κόμβου δεν αλλάζει την αντιστοίχιση ανάμεσα στα κλειδιά και στους κόμβους. Αντίθετα στους περισσότερους πίνακες ανάμειξης, μια αλλαγή στον αριθμό των κόμβων προκαλεί αλλαγή στην αντιστοίχιση ανάμεσα σε όλα σχεδόν τα κλειδιά και στους κόμβους.

Η συνεπής ανάμειξη τείνει να εξισορροπεί το φορτίο, αφού κάθε κόμβος λαμβάνει σχεδόν τον ίδιο αριθμό κλειδιών και μικρές αλλαγές στην αντιστοίχιση τους όταν κόμβοι εισέρχονται ή αποχωρούν από το σύστημα.

Κάθε κόμβος του δικτύου Chord, χρειάζεται πληροφορίες δρομολόγησης για λίγους μόνο κόμβους. Επειδή ο πίνακας δρομολόγησης είναι κατανεμημένος, ένας κόμβος χρησιμοποιεί τη συνάρτηση ανάμειξης επικοινωνώντας με μικρό αριθμό κόμβων. Σε σταθερή κατάσταση, σε σύστημα με N κόμβους, κάθε κόμβος διατηρεί πληροφορία για τους υπόλοιπους κόμβους της τάξης του $O(\log N)$, και κάνει αναζητήσεις διαμέσω $O(\log N)$ μηνυμάτων σε αυτούς. Το Chord διατηρεί τις πληροφορίες δρομολόγησης καθώς κόμβοι εισέρχονται και αναχωρούν από το σύστημα.

Με υψηλή πιθανότητα κάθε τέτοιο γεγονός έχει ως αποτέλεσμα τη χρησιμοποίηση κατά μέγιστο $O(\log^2 N)$ αριθμού μηνυμάτων.

Τρία χαρακτηριστικά που διαχωρίζουν το Chord από άλλα peer-to-peer πρωτόκολλα είναι η απλότητά του, η ορθότητα και η απόδοσή του. Το Chord είναι απλό,

δρομολογώντας ένα κλειδί μέσα από μια διαδομή $O(\log N)$ κόμβων προς τον τελικό προορισμό. Ένας κόμβος απαιτεί πληροφορίες για $O(\log N)$ υπόλοιπους κόμβους για αποτελεσματική δρομολόγηση, αλλά η απόδοση μειώνεται σημαντικά όταν η πληροφορία δεν είναι ανανεωμένη. Αυτό είναι σημαντικό γιατί οι κόμβοι εισέρχονται και αποχωρούν τυχαία, και η διατήρηση της τάξης του $O(\log N)$ είναι δύσκολη. Για να εγγυηθεί σωστή, αν και αργή, δρομολόγηση των αναζητήσεων, αρκεί ένα μέρος της πληροφορίας ανά κόμβο να είναι σωστό.

3.1 Τα χαρακτηριστικά του πρωτοκόλου Chord

Το Chord απλοποιεί το σχεδιασμό των υπερκείμενων δικτύων και των εφαρμογών τους αντιμετωπίζοντας τα παρακάτω προβλήματα:

- **Ισορροπία φορτίου:** Το Chord λειτουργεί σαν κατανεμημένη συνάρτηση ανάμειξης, κατανέμοντας τα κλειδιά ομαλά στους κόμβους. Έτσι παρέχεται μια ισορροπία στο φορτίο.
- **Αποκέντρωση:** Το Chord είναι πλήρως κατανεμημένο. Κάθε κόμβος είναι εξίσου σημαντικός με έναν άλλο. Αυτό κάνει το Chord κατάλληλο για εφαρμογές peer-to-peer που είναι χαλαρά οργανωμένες.
- **Κλιμάκωση:** Το κόστος μιας αναζήτησης μεγαλώνει ανάλογα με το λογάριθμο του αριθμού των κόμβων, έτσι ακόμα και μεγάλα συστήματα είναι εφικτά, χωρίς ρύθμιση κάποιας παραμέτρου για να επιτευχθεί αυτή η κλιμάκωση.
- **Διαθεσιμότητα:** Το Chord προσαρμόζει αυτόματα τους πίνακές του ώστε να αντικατοπτρίζουν τους νεοεισερχόμενους κόμβους καθώς και τις αποχωρήσεις κόμβων, διασφαλίζοντας ότι βλάβες στο δίκτυο μπορούν να αντιστοιχιστούν στον κόμβο που είναι υπεύθυνος. Αυτό συμβαίνει ακόμα και σε κατάσταση συνεχούς αλλαγής.
- **Ευέλικτη ονομασία:** Το Chord δεν θέτει περιορισμούς στη δομή των κλειδιών που δρομολογεί. Αυτό δίνει μεγάλη ευελιξία στις εφαρμογές, ως προς τον τρόπο αντιστοίχισης στις ονομασίες.

Το λογισμικό Chord παίρνει τη μορφή βιβλιοθήκης που συνδέεται με τις εφαρμογές πελάτη και εξυπηρετητή που τις χρησιμοποιούν. Η εφαρμογή αλληλεπιδρά με το Chord με δύο τρόπους. Πρώτα το Chord παρέχει έναν αλγόριθμο που εμφανίζει τη διεύθυνση IP του κόμβου που είναι υπεύθυνος για το κλειδί. Έπειτα το λογισμικό σε κάθε κόμβο ειδοποιεί την εφαρμογή για τυχόν αλλαγές στα κλειδιά για τα οποία ο κόμβος είναι υπεύθυνος. Αυτό επιτρέπει το λογισμικό της εφαρμογής, για παράδειγμα, να μετακινήσει τις αντίστοιχες τιμές όταν ένας νέος κόμβος εισέρχεται στο δίκτυο.

Η εφαρμογή που χρησιμοποιεί το Chord είναι υπεύθυνη για την παροχή οποιασδήποτε πιστοποίησης, αντιγραφής ή φιλικής προς τον χρήστη ονομασίας των δεδομένων. Για παράδειγμα, μια εφαρμογή θα μπορούσε να πιστοποιήσει δεδομένα αποθηκεύοντάς τα κάτω από ένα κλειδί που προκύπτει από κρυπτογραφική ανάμειξη των δεδομένων. Παρομοίως, μια εφαρμογή μπορεί να αντιγράψει δεδομένα αποθηκεύοντάς τα κάτω από δύο ξεχωριστά κλειδιά του Chord που προέρχονται από την ταυτοποίηση των δεδομένων.

Υπάρχουν κάποιες εφαρμογές για τις οποίες το Chord μπορεί να προσφέρει σημαντική εξυπηρέτηση.

Αποθήκευση μοιρασμένου χρόνου

Αφορά σε κόμβους με ασυνεχή σύνδεση. Αν κάποιος χρήστης επιθυμεί κάποια δεδομένα του να είναι συνεχώς διαθέσιμα, αλλά η σύνδεσή του δεν είναι πάντα διαθέσιμη, προσφέρουν αποθήκευση δεδομένων άλλων κόμβων όσο ο χρήστης είναι συνδεδεμένος, σε αντάλλαγμα με την αποθήκευση των δικών του δεδομένων όταν ο ίδιος δεν είναι συνδεδεμένος.

Κατανεμημένα περιεχόμενα

Ένα κλειδί σε τέτοιες εφαρμογές θα μπορούσε να προκύψει από επιθυμητές λέξεις, ενώ οι τιμές θα μπορούσαν να είναι λίστες υπολογιστών που προσφέρουν αρχεία με αυτές τις λέξεις.

Συνδυασμένη αναζήτηση μεγάλης κλίμακας

Μια τέτοια εφαρμογή αφορά σε σπάσιμο κώδικα. Σε αυτή την περίπτωση κλειδιά είναι υποψήφιες λύσεις στο πρόβλημα, όπως για παράδειγμα κρυπτογραφικά κλειδιά. Το Chord αντιστοιχίζει τα κλειδιά στους υπολογιστές που είναι υπεύθυνοι για να τα δοκιμάσουν ως πιθανές λύσεις.

3.2 Το βασικό πρωτόκολο Chord

Το πρωτόκολο Chord προσδιορίζει τον τρόπο εύρεσης της τοποθεσίας των κλειδιών, τον τρόπο εισαγωγής νέων κόμβων στο σύστημα και την επαναφορά της ορθής λειτουργίας από σφάλματα στους κόμβους.

3.2.1 Επισκόπηση

Το πρωτόκολο Chord παρέχει γρήγορο, κατανεμημένο υπολογισμό μιας συνάρτησης ανάμειξης που αντιστοιχίζει κλειδιά στους κόμβους που είναι υπεύθυνοι για αυτά. Χρησιμοποιεί συνεπή ανάμειξη που έχει αρκετές θετικές επιπτώσεις. Με υψηλή πιθανότητα η συνάρτηση ανάμειξης εξισορροπεί την κατανομή του φορτίου. Όλοι οι κόμβοι λαμβάνουν περίπου τον ίδιο αριθμό κλειδιών.

Το πρωτόκολο Chord βελτιώνει την κλιμάκωση της συνεπούς ανάμειξης αποφεύγοντας την απαίτηση ότι κάθε κόμβος πρέπει να είναι ενήμερος για κάθε άλλο κόμβο. Ένας κόμβος στο πρωτόκολο Chord χρειάζεται μόνο ένα μέρος της πληροφορίας δρομολόγησης για τους άλλους κόμβους. Επειδή η πληροφορία είναι κατανεμημένη, ένας κόμβος διατηρεί πληροφορία για $O(\log N)$ άλλους κόμβους, και η αναζήτηση απαιτεί $O(\log N)$ μηνύματα.

3.2.2 Συνεπής ανάμειξη

Η συνεπής συνάρτηση ανάμειξης αντιστοιχίζει σε κάθε κόμβο και κλειδί ένα αναγνωριστικό των m -bit χρησιμοποιώντας την βασική συνάρτηση ανάμειξης SHA-1. Το αναγνωριστικό ενός κόμβου επιλέγεται χρησιμοποιώντας τη συνάρτηση ανάμειξης της διεύθυνσης IP του κόμβου και το αναγνωριστικό ενός κλειδιού παρομοίως από το κλειδί. Το μήκος m του αναγνωριστικού πρέπει να είναι αρκετά μεγάλο ώστε να

ελαχιστοποιείται η πιθανότητα για δύο κόμβους ή κλειδιά να έχουν το ίδιο αναγνωριστικό.

Η αντιστοίχιση κλειδιών με τους κόμβους γίνεται ως εξής. Τα αναγνωριστικά κατανέμονται σε έναν κύκλο αναγνωριστικών, υπολοίπου διαίρεσης 2^m . Το κλειδί k αντιστοιχίζεται στον πρώτο κόμβο του οποίου το αναγνωριστικό είναι ίσο ή ακολουθεί το αναγνωριστικό του k στον χώρο των αναγνωριστικών. Αυτός ο κόμβος ονομάζεται διάδοχος κόμβος του κλειδιού k . Αν τα αναγνωριστικά παρουσιάζονται σαν ένας κύκλος αριθμών από το 0 έως το $2^m - 1$, τότε ο διάδοχος κόμβος είναι ο πρώτος κόμβος με ωρολογιακή φορά από το k .

Η συνεπής ανάμειξη έχει σχεδιαστεί ώστε να επιτρέπει στους κόμβους να εισέρχονται και να αναχωρούν από το δίκτυο με τη μικρότερη παρεμπόδιση. Για να διατηρείται ορθή αντιστοίχιση όταν ένας κόμβος n εισέρχεται στο δίκτυο, ορισμένα κλειδιά που είχαν ανατεθεί στο διάδοχο κόμβο του n τώρα ανατίθενται στον κόμβο n . Όταν ένας κόμβος n αναχωρεί, τα κλειδιά που του έχουν ανατεθεί ανατίθενται ξανά. Δεν είναι απαραίτητες άλλες αλλαγές στην αντιστοίχιση.

3.2.3 Κλιμακούμενη εύρεση κλειδιού

Μια μικρή ποσότητα πληροφορίας για τη δρομολόγηση αρκεί για να εφαρμοστεί η συνεπής ανάμειξη σε ένα κατανεμημένο περιβάλλον. Κάθε κόμβος χρειάζεται να ξέρει μόνο τον επόμενο κόμβο από αυτόν στον κύκλο. Αναζητήσεις για ένα δεδομένο αναγνωριστικό μπορούν να προωθηθούν στον κύκλο διαμέσω αυτών των δεικτών των επόμενων κόμβων μέχρι να συναντήσουν τον κόμβο που είναι μετά το αναγνωριστικό. Αυτός είναι ο κόμβος που αντιστοιχίζεται στο κλειδί. Ένα μέρος του πρωτοκόλου Chord διατηρεί τους δείκτες των επόμενων κόμβων. Παρ'όλα αυτά, υπάρχει ένα μειονέκτημα. Είναι πιθανό να περαστούν όλοι οι N κόμβοι, για να βρεθεί η σωστή αντιστοίχιση. Για να επιταχυνθεί η διαδικασία αυτή, το πρωτόκολο Chord διατηρεί επιπρόσθετες πληροφορίες για τη δρομολόγηση. Αυτές οι πληροφορίες δεν είναι απαραίτητες για την ορθότητα, η οποία επιτυγχάνεται αρκεί να διατηρούνται σωστά οι πληροφορίες για τους επόμενους κόμβους.

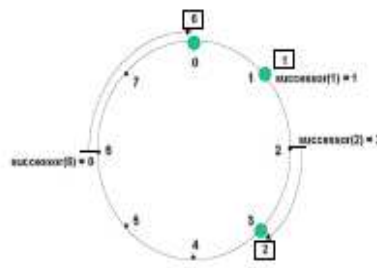
Αν m είναι ο αριθμός των bit στα αναγνωριστικά των κόμβων και των κλειδιών, κάθε κόμβος n διατηρεί έναν πίνακα δρομολόγησης με το πολύ m καταχωρήσεις που

ονομάζεται δακτυλικός πίνακας(finger table). Η είσοδος i του πίνακα αυτού στον κόμβο n περιέχει την ταυτότητα του πρώτου κόμβου, s , που είναι επόμενο του n τουλάχιστον κατά 2^{i-1} , στον κύκλο των αναγνωριστικών. Μια καταχώρηση στον δακτυλικό πίνακα περιέχει τόσο το αναγνωριστικό του Chord, όσο και την διεύθυνση IP του κόμβου.

Αυτή η υλοποίηση περιέχει δύο χαρακτηριστικά. Κάθε κόμβος αποθηκεύει πληροφορία για ένα μικρό αριθμό κόμβων και ξέρει περισσότερα για κόμβους που είναι κοντά του στον κύκλο των αναγνωριστικών, από ότι για κόμβους που βρίσκονται πιο μακριά. Επιπλέον, ο δακτυλικός πίνακας ενός κόμβου δεν περιέχει αρκετά στοιχεία για τον προσδιορισμό του επομένου ενός κλειδιού k .

Αν ένας κόμβος δεν γνωρίζει τον επόμενο ενός κλειδιού k , τότε βρίσκει τον κόμβο του οποίου η θέση είναι κοντινότερη στο k από τη δική του. Αυτός ο κόμβος θα γνωρίζει περισσότερα για τον κύκλο των αναγνωριστικών από τον αρχικό κόμβο.

Στη συνέχεια παρουσιάζεται ένας κύκλος αναγνωριστικών που αποτελείται από τους κόμβους 0, 1 και 3. Σε αυτό το παράδειγμα, το κλειδί 1 βρίσκεται στον κόμβο 1, το κλειδί 2 στον κόμβο 3, και το κλειδί 6 στον κόμβο 0.



Εικόνα 2: Κύκλος αναγνωριστικών αποτελούμενος από τους 3 κόμβους 0, 1, 3.

3.2.4 Είσοδος κόμβου

Σε ένα δυναμικό δίκτυο, κόμβοι μπορούν να εισέρχονται και να αποχωρούν σε οποιαδήποτε στιγμή. Η πρόκληση σε αυτές τις διεργασίες είναι η διατήρηση της ικανότητας εντοπισμού του κάθε κλειδιού στο δίκτυο.

Για να γίνει αυτό το πρωτόκολο Chord πρέπει να διατηρήσει δύο τομείς.

1. Ο διάδοχος κάθε κόμβου διατηρείται αποτελεσματικά
2. Για κάθε κλειδί k , ο διάδοχος κόμβος του κλειδιού k είναι υπεύθυνος για το k .

Για να απλοποιηθούν οι μηχανισμοί για την είσοδο και την αναχώρηση των κόμβων από το δίκτυο, κάθε κόμβος στο πρωτόκολο Chord διατηρεί έναν δείκτη του προκατόχου. Ο δείκτης προκατόχου ενός κόμβου περιέχει το αναγνωριστικό του Chord και τη διεύθυνση IP του άμεσου προκατόχου του κόμβου, και χρησιμοποιείται για την αντιωρολογιακή αναδρομή του κύκλου των αναγνωριστικών.

Το πρωτόκολο Chord πρέπει να κάνει τρεις λειτουργίες όταν ένας κόμβος n εισέρχεται στο δίκτυο:

1. Ενεργοποίηση του προκατόχου και της λειτουργίας του δακτυλικού πίνακα για τον κόμβο n .
2. Ενημέρωση του δακτυλικού πίνακα και των προκατόχων των κόμβων που είναι ήδη στο δίκτυο για την προσθήκη του κόμβου n .
3. Ειδοποίηση του υψηλότερου επιπέδου λογισμικού, για τη μεταφορά χαρακτηριστικών που σχετίζονται με τα κλειδιά για τα οποία είναι υπεύθυνος ο κόμβος n .

Ο νέος κόμβος n χρησιμοποιεί τον υπάρχων κόμβο n' για να προστεθεί στο δίκτυο Chord όπως παρακάτω.

Ενεργοποίηση του προκατόχου και της λειτουργίας του δακτυλικού πίνακα

Ένας νέος κόμβος n , ζητά από έναν άμεσο γείτονα ένα αντίγραφο του πλήρους δακτυλικού πίνακα και του προκατόχου του. Τα περιεχόμενα αυτών των πινάκων μπορούν να χρησιμεύσουν στο νέο κόμβο να βρει τις κατάλληλες τιμές για τους δικούς του πίνακες, αφού οι πίνακες του νέου κόμβου θα έχουν αρκετές ομοιότητες με αυτούς του γειτονικού του κόμβου. Αυτό μπορεί να μειώσει το χρόνο για τη συμπλήρωση του δακτυλικού πίνακα σε $O(\log N)$.

Ενημέρωση των στοιχείων του δακτυλικού πίνακα των υπάρχοντων κόμβων

Ο αριθμός των κόμβων που χρειάζεται να ενημερωθούν όταν ένας κόμβος εισέρχεται στο δίκτυο είναι $O(\log N)$, κατά μεγάλη πιθανότητα. Η εύρεση και ενημέρωση των κόμβων χρειάζεται χρόνο $O(\log^2 N)$.

Μεταφορά κλειδιών

Η τελευταία λειτουργία που χρειάζεται να εκτελεστεί όταν ένας νέος κόμβος εισέρχεται στο σύστημα, είναι να γίνει αντιστοίχιση των κλειδιών για τα οποία ο κόμβος n είναι τώρα ο διάδοχος κόμβος. Αυτό περιλαμβάνει τη μεταφορά των δεδομένων που σχετίζονται με το κάθε κλειδί στον νέο κόμβο.

3.2.5 Αναχώρηση κόμβων και αναπαραγωγή

Όταν ένας κόμβος αναχωρεί, οι κόμβοι των οποίων οι δακτυλικοί πίνακες περιλαμβάνουν τον κόμβο n , πρέπει να βρουν τον διάδοχο κόμβο του n . Ακόμα, η αναχώρηση του κόμβου n , δεν γίνεται να εμποδίσει τις αναζητήσεις που βρίσκονται σε εξέλιξη.

Σημαντικό στοιχείο για την επαναφορά του δικτύου, είναι η διατήρηση των ορθών δεικτών των διαδόχων κόμβων. Κάθε κόμβος διαθέτει μια λίστα με τους κοντινότερους διάδοχους κόμβους στον δακτύλιο Chord. Αν ένας κόμβος ενημερωθεί ότι ο διάδοχος κόμβος του έχει αναχωρήσει, τον αντικαθιστά με τον κοντινότερο από την λίστα αυτή.

Ο μηχανισμός της λίστας διαδόχων βοηθά λογισμικό υψηλότερου επιπέδου στην αντιγραφή δεδομένων. Μια εφαρμογή που χρησιμοποιεί το πρωτόκολο Chord, μπορεί να αποθηκεύει τα αντίγραφα των δεδομένων που σχετίζονται με ένα κλειδί στους k κόμβους που διαδέχονται το κλειδί. Όταν ένας κόμβος χρησιμοποιεί τη λίστα διαδόχων μπορεί να ενημερώσει το λογισμικό για την αποχώρηση ή άφιξη κόμβων και επομένως αν χρειάζεται η αντιγραφή των δεδομένων.

4.Pastry: Ένα κλιμακούμενο, αποκεντρωμένο υπερκείμενο δίκτυο για δρομολόγηση σε μεγάλης κλίμακας συστήματα

Το Pastry [3] είναι ένα κλιμακούμενο, κατανεμημένο υπερκείμενο δίκτυο, ικανό για μεγάλης κλίμακας εφαρμογές με υπερκείμενα δίκτυα. Είναι πλήρως αποκεντρωμένο, ανθεκτικό σε λάθη, έχει χαρακτηριστικά μεγάλης αξιοπιστίας, με θετικές ιδιότητες δρομολόγησης.

Το Pastry λειτουργεί ως υπόστρωμα για την κατασκευή ποικιλίας εφαρμογών peer-to-peer μέσω του διαδικτύου όπως διαμοίρασμα αρχείων, αποθήκευση αρχείων, επικοινωνίες ομάδων, και ονομασία συστημάτων.

Κάθε κόμβος στο δίκτυο έχει ένα μοναδικό αριθμητικό αναγνωριστικό. Όταν σε έναν κόμβο φτάνει ένα μήνυμα και ένα κλειδί, ο κόμβος αυτός δρομολογεί αποτελεσματικά το μήνυμα στον κόμβο με το αναγνωριστικό που είναι πλησιέστερο στο κλειδί, ανάμεσα στους ενεργούς κόμβους του δικτύου. Ο αναμενόμενος αριθμός των βημάτων είναι $O(\log N)$, όπου N είναι ο αριθμός των κόμβων στο δίκτυο. Σε κάθε κόμβο από τον οποίο δρομολογείται το μήνυμα, η εφαρμογή ενημερώνεται και μπορεί να εκτελεί συγκεκριμένες λειτουργίες σχετικά με το μήνυμα.

Το Pastry λαμβάνει υπόψη την τοπολογία του δικτύου. Επιδιώκει να ελαχιστοποιήσει την απόσταση που διανύουν τα μηνύματα, με βάση κάποιο σύστημα διαβάθμισης όπως των αριθμό των μεταβάσεων IP. Κάθε κόμβος στο δίκτυο Pastry γνωρίζει τους άμεσους γείτονές του στον χώρο των αναγνωριστικών, και ενημερώνει τις εφαρμογές για νέες αφίξεις ή αναχωρήσεις. Τα αναγνωριστικά ανατίθενται τυχαία και υπάρχει η περίπτωση κόμβοι με συνεχόμενα αναγνωριστικά να είναι απομακρυσμένοι γεωγραφικά. Οι εφαρμογές μπορούν να ρυθμίσουν αυτό το μειονέκτημα, γιατί το Pastry μπορεί να δρομολογήσει σε έναν από τους k κόμβους που είναι αριθμητικά πλησιέστεροι στο κλειδί.

Υπάρχουν εφαρμογές που χρησιμοποιούν αυτές τις δυνατότητες με διαφορετικό τρόπο. Μπορούν να χρησιμοποιηθούν για την αποθήκευση αρχείων με τη χρήση των μηχανισμών ειδοποίησης του Pastry, όπως το PAST και σε συστήματα εκδόσεων-συνδρομής όπως το SCRIBE. Συγκεκριμένα διατηρούνται αντίγραφα των αρχείων στους k κόμβους που είναι πλησιέστεροι στο κλειδί, ανεξαρτήτως των αφίξεων ή αναχωρήσεων

των κόμβων, με τη χρήση μόνο τοπικού συντονισμού ανάμεσα στους κόμβους που έχουν συνεχόμενα αναγνωριστικά.

4.1 Σχεδιασμός του υπερκείμενου δικτύου Pastry

Το Pastry είναι ένα αυτο-οργανούμενο υπερκείμενο δίκτυο, στο οποίο κάθε κόμβος δρομολογεί αιτήσεις των πελατών και αλληλεπιδρά με μια ή περισσότερες εφαρμογές. Κάθε υπολογιστής που είναι συνδεδεμένος στο διαδίκτυο και χρησιμοποιεί το λογισμικό Pastry, μπορεί να λειτουργήσει σαν κόμβος του Pastry και να υπόκειται στους περιορισμούς ασφαλείας των εφαρμογών.

Σε κάθε κόμβο στο υπερκείμενο δίκτυο Pastry ανατίθεται ένα αναγνωριστικό κόμβου των 128 bit. Το αναγνωριστικό χρησιμοποιείται για να υποδείξει τη θέση του κόμβου σε έναν κυκλικό χώρο, ο οποίος εκτείνεται από 0 έως $2^{128} - 1$. Το αναγνωριστικό ανατίθεται τυχαία μόλις ένας κόμβος εισέρχεται στο δίκτυο. Τα αναγνωριστικά παράγονται έτσι ώστε να κατανέμονται ομοιόμορφα στον κυκλικό χώρο. Για παράδειγμα τα αναγνωριστικά μπορούν να παράγονται από υπολογισμό της συνάρτησης ανάμειξης της διεύθυνσης IP. Με αυτό τον τρόπο κόμβοι με συνεχόμενα αναγνωριστικά είναι ποικίλοι στις τοποθεσίες ή στις δικαιοδοσίες καθώς και σε άλλα χαρακτηριστικά.

Σε ένα δίκτυο με N κόμβους, το Pastry μπορεί να κάνει δρομολόγηση στον αριθμητικά πλησιέστερο κόμβο, δοσμένου ενός κλειδιού, σε λιγότερο από $\log_2 N$ βήματα σε φυσιολογικές συνθήκες. Το b είναι μια παράμετρος με τυπική τιμή ίση με 4. Παρά τις ταυτόχρονες αποχωρήσεις κόμβων, η ορθή δρομολόγηση είναι εξασφαλισμένη, εκτός αν $|L|/2$ κόμβοι με συνεχόμενα αναγνωριστικά αποχωρήσουν ταυτόχρονα. Το $|L|$ είναι μια παράμετρος με τυπική τιμή 16 ή 32.

Για τη διαδικασία της δρομολόγησης τα αναγνωριστικά και τα κλειδιά εκφράζονται ως αλληλουχίες ψηφίων με βάση 2^b . Το Pastry δρομολογεί τα μηνύματα στον κόμβο του οποίου το αναγνωριστικό είναι πλησιέστερο στο δεδομένο κλειδί. Σε κάθε βήμα δρομολόγησης ένας κόμβος προωθεί το μήνυμα στον κόμβο του οποίου το αναγνωριστικό έχει κοινό πρόθεμα με το κλειδί που είναι τουλάχιστον 1 bit μεγαλύτερο από το πρόθεμα που είναι κοινό με το κλειδί στον τρέχων κόμβο. Αν δεν είναι γνωστός ένας τέτοιος κόμβος, το μήνυμα προωθείται στον κόμβο του οποίου το αναγνωριστικό

έχει κοινό πρόθεμα με το κλειδί ίσο με αυτό του τρέχοντος κόμβου, αλλά είναι αριθμητικά πλησιέστερο από αυτόν.

4.2 Κατάσταση κόμβων στο Pastry

Κάθε κόμβος στο Pastry διατηρεί έναν πίνακα δρομολόγησης, μια ομάδα γειτνίασης και μια ομάδα φύλλων. Ο πίνακας δρομολόγησης, R , είναι οργανωμένος σε $\lceil \log_{2^b} N \rceil$ σειρές με $2^b - 1$ καταχωρήσεις η κάθε μια. Οι $2^b - 1$ καταχωρήσεις σε μια σειρά n , αναφέρονται στον κόμβο του οποίου το αναγνωριστικό έχει το ίδιο αναγνωριστικό του παρόντος κόμβου στα πρώτα n ψηφία, αλλά το $n + 1$ ψηφίο, έχει μια από τις $2^b - 1$ δυνατές τιμές, που είναι διαφορετικές από αυτές του $n + 1$ ψηφίου στον παρόντα κόμβο.

Κάθε καταχώρηση στον πίνακα δρομολόγησης περιέχει την διεύθυνση IP ενός από τους πολλούς πιθανούς κόμβους οι οποίοι έχουν το απαραίτητο πρόθεμα. Στην εφαρμογή επιλέγεται ο κόμβος που είναι πλησιέστερος στον παρόντα κόμβο. Αν δεν υπάρχει κάποιος κόμβος με το κατάλληλο αναγνωριστικό, τότε η καταχώρηση μένει άδεια. Η ομοιόμορφη κατανομή των αναγνωριστικών διασφαλίζει την ομαλή κατάληψη του χώρου. Έτσι μόνο $\lceil \log_{2^b} N \rceil$ σειρές καταλαμβάνονται στον πίνακα δρομολόγησης.

Η επιλογή του b γίνεται με υπολογισμό μεταξύ του μεγέθους του κατειλημένου μέρους του πίνακα δρομολόγησης και του μέγιστου αριθμού βημάτων που απαιτούνται για τη δρομολόγηση μεταξύ οποιονδήποτε κόμβων.

Η ομάδα γειτνίασης M περιέχει τα αναγνωριστικά και τις διευθύνσεις IP των $|M|$ κόμβων που είναι πλησιέστερα στον τοπικό κόμβο. Η ομάδα γειτνίασης συνήθως δεν χρησιμοποιείται στη δρομολόγηση μηνυμάτων. Είναι χρήσιμη στη διατήρηση των τοπικών χαρακτηριστικών. Η ομάδα φύλλων L είναι μια ομάδα κόμβων με τους $|L|/2$ μεγαλύτερους αριθμητικά πλησιέστερους κόμβους, και τους $|L|/2$ μικρότερους αριθμητικά πλησιέστερους κόμβους σχετικά με το αναγνωριστικό του παρόντος κόμβου.

4.3 Δρομολόγηση στο Pastry

Η διαδικασία δρομολόγησης περιγράφεται αναλυτικά στη συνέχεια. Με δεδομένο ένα μήνυμα, ο κόμβος ελέγχει πρώτα αν το κλειδί είναι στην ακτίνα των αναγνωριστικών

που καλύπτονται από την ομάδα φύλλων του. Τότε, το μήνυμα προωθείται κατευθείαν στον κόμβο προορισμού, δηλαδή στον κόμβο του οποίου το αναγνωριστικό είναι πλησιέστερο στο κλειδί.

Αν το κλειδί δεν καλύπτεται από την ομάδα φύλλων, τότε χρησιμοποιείται ο πίνακας δρομολόγησης και το μήνυμα προωθείται στον κόμβο που έχει κοινό πρόθεμα με το κλειδί κατά τουλάχιστον ένα ψηφίο. Σε μερικές περιπτώσεις είναι πιθανό ότι η κατάλληλη καταχώρηση στον πίνακα δρομολόγησης είναι άδεια ή ο συσχετιζόμενος κόμβος δεν είναι προσβάσιμος. Σε αυτή την περίπτωση το μήνυμα προωθείται στον κόμβο που έχει κοινό πρόθεμα με το κλειδί τουλάχιστον τόσο μεγάλο όσο στον παρόντα κόμβο, και είναι αριθμητικά πλησιέστερο στο κλειδί από το αναγνωριστικό του παρόντος κόμβου.

Ένας τέτοιος κόμβος πρέπει να είναι στην ομάδα φύλλων εκτός αν το μήνυμα έχει φτάσει στον κόμβο με το πλησιέστερα αριθμητικό αναγνωριστικό. Και εκτός αν $|L|/2$ διαδοχικοί κόμβοι αποχωρήσουν ταυτόχρονα, τουλάχιστον ένας κόμβος πρέπει να είναι ενεργός.

Απόδοση δρομολόγησης

Ο αναμενόμενος αριθμός βημάτων είναι $(\log_{2^b} N)$, με ιδανικές συνθήκες. Αν ένα μήνυμα προωθείται με τη χρήση του πίνακα δρομολόγησης, τότε η ομάδα των κόμβων των οποίων το αναγνωριστικό έχει μεγαλύτερο πρόθεμα από το κλειδί, μειώνεται κατά 2^b σε κάθε βήμα, που σημαίνει ότι το μήνυμα φτάνει στον προορισμό σε $(\log_{2^b} N)$ βήματα. Αν το κλειδί είναι στην ακτίνα της ομάδας φύλλων τότε ο προορισμός είναι το πολύ ένα βήμα μακριά.

Η τρίτη περίπτωση είναι όταν το κλειδί δεν καλύπτεται από την ομάδα φύλλων, αλλά δεν υπάρχει καταχώρηση στον πίνακα δρομολόγησης. Υποθέτοντας ακριβείς πίνακες δρομολόγησης και χωρίς πρόσφατες καταρρεύσεις κόμβων, αυτό σημαίνει ότι ο κόμβος με το κατάλληλο πρόθεμα δεν υπάρχει. Η πιθανότητα αυτού του ενδεχομένου εξαρτάται από το $|L|$. Όταν ισχύει ότι $|L| = 2^b$ και $|L| = 2 \times 2^b$, τότε η πιθανότητα είναι μικρότερη από 0.02 και 0.006 αντίστοιχα.

Στην περίπτωση πολλών και ταυτόχρονων καταρρεύσεων των κόμβων, ο αριθμός των βημάτων δρομολόγησης που απαιτούνται μπορεί να είναι στη χειρότερη περίπτωση γραμμικός με το N , όσο οι κόμβοι ενημερώνονται για τη νέα κατάσταση. Στην πράξη η απόδοση της δρομολόγησης μειώνεται σταδιακά όσο οι κόμβοι καταρρέουν. Η τελική παράδοση του μηνύματος είναι εξασφαλισμένη, εκτός αν $|L|/2$ κόμβοι με διαδοχικά αναγνωριστικά καταρρεύσουν ταυτόχρονα. Εξαιτίας της αναμενόμενης διασποράς των κόμβων με διαδοχικά αναγνωριστικά η πιθανότητα αυτού του ενδεχομένου είναι πολύ μικρή.

4.4 Άφιξη και αναχώρηση κόμβων

Στη συνέχεια θα γίνει περιγραφή του πρωτοκόλου Pastry για τη διαχείριση των αφίξεων και των αναχωρήσεων των κόμβων στο δίκτυο.

Άφιξη κόμβου

Με την άφιξη ενός νέου κόμβου γίνεται αρχικοποίηση του πίνακα κατάστασης και έπειτα ενημέρωση των υπόλοιπων κόμβων για την παρουσία του. Στην αρχή γίνεται η υπόθεση ότι ο νέος κόμβος γνωρίζει για έναν κόμβο A , που είναι ήδη στο δίκτυο.

Έστω ότι το αναγνωριστικό του νέου κόμβου συμβολίζεται με X . Το αναγνωριστικό υπολογίζεται με τη συνάρτηση ανάμειξης της διεύθυνσης IP . Ο κόμβος X ζητά από τον κόμβο A ένα μήνυμα 'join' με το κλειδί ίσο με το X . Όπως σε κάθε μήνυμα, το Pastry δρομολογεί το μήνυμα 'join' στον κόμβο Z του οποίου το αναγνωριστικό είναι πλησιέστερο στο X .

Οι κόμβοι A, Z αλλά και όλοι οι κόμβοι που συναντούνται από το A στο Z , αφού λάβουν το μήνυμα 'join' στέλνουν τους πίνακες κατάστασής τους στον κόμβο X . Ο νέος κόμβος X λαμβάνει και επεξεργάζεται την πληροφορία και ξεκινά τους δικούς του πίνακες κατάστασης. Τελικά, ο κόμβος X ενημερώνει όσους κόμβους χρειάζεται, για την άφιξή του.

Ο κόμβος A , υποτίθεται ότι είναι κοντινός με τον κόμβο X , η ομάδα γειτνίασης του A αρχικοποιεί αυτήν του κόμβου X . Επιπλέον, ο κόμβος Z έχει το πλησιέστερο αναγνωριστικό με το X , επομένως η ομάδα φύλλων του είναι η βάση για την ομάδα φύλλων του X . Στη συνέχεια σχεδιάζεται ο πίνακας δρομολόγησης ξεκινώντας από τη

μηδενική σειρά. Στην πιο γενική περίπτωση τα αναγνωριστικά του κόμβου A δεν έχουν κανένα κοινό πρόθεμα με αυτά του κόμβου X . Με A_i να συμβολίζεται η σειρά του πίνακα δρομολόγησης στο επίπεδο i . Οι καταχωρήσεις στη μηδενική σειρά του πίνακα δρομολόγησης είναι ανεξάρτητες με τα αναγνωριστικά ενός κόμβου. Έτσι, η A_0 περιέχει σχετικές τιμές για το X_0 . Τα άλλα επίπεδα του πίνακα δρομολόγησης για τον πίνακα A δεν έχουν χρησιμότητα για το X .

Παρ' όλα αυτά σχετικές τιμές για το X_1 μπορούν να εξαχθούν από τον B_1 , όπου B είναι ο πρώτος κόμβος που συναντάται στη διαδρομή από τον κόμβο A στον κόμβο Z . Οι καταχωρήσεις στην B_1 και στην X_1 έχουν κοινό πρόθεμα, γιατί οι κόμβοι X και B έχουν το ίδιο πρώτο ψηφίο στο αναγνωριστικό. Ομοίως ο κόμβος X ανακτά καταχωρήσεις για την X_2 από τον κόμβο C , τον επόμενο κόμβο που συναντάται από τη διαδρομή από τον κόμβο A στον κόμβο Z και ούτω καθεξής.

Τελικά, ο κόμβος X μεταδίδει ένα αντίγραφο της τελικής του κατάστασης σε κάθε κόμβο που βρίσκεται στην ομάδα γειτνίασης, ομάδα φύλλων και πίνακα δρομολόγησης. Σε αυτό το σημείο ο νέος κόμβος μπορεί να δρομολογεί και να λαμβάνει μηνύματα, και να συμμετέχει στο υπερκείμενο δίκτυο Pastry. Το συνολικό κόστος για την άφιξη ενός κόμβου σχετικά με τον αριθμό μηνυμάτων που ανταλλάχθηκαν, είναι $O(\log_{2^b} N)$.

Αναχώρηση κόμβου

Οι κόμβοι στο δίκτυο Pastry μπορεί να αναχωρούν ή να καταρρεύσουν χωρίς προειδοποίηση. Ένας κόμβος στο δίκτυο Pastry θεωρείται ότι έχει καταρρεύσει όταν οι άμεσοι γειτονικοί του κόμβοι στον χώρο των αναγνωριστικών δεν μπορούν να επικοινωνήσουν μαζί του.

Για την αντικατάσταση ενός κόμβου που έχει καταρρεύσει, ο γειτονικός του κόμβος στον χώρο των αναγνωριστικών επικοινωνεί με τον ενεργό κόμβο με το μεγαλύτερο πρόθεμα στην πλευρά του κόμβου που έχει καταρρεύσει και ζητά τον πίνακα φύλλων, που έστω ότι ονομάζεται L' . Αυτή η ομάδα επικαλύπτει την ομάδα φύλλων του παρόντος κόμβου και περιέχει κόμβους με αναγνωριστικό που βρίσκονται στην L . Από τους νέους αυτούς κόμβους επιλέγεται ο κατάλληλος για την εισαγωγή στην ομάδα L , μετά την επιβεβαίωση ότι ο κόμβος είναι ενεργός μετά από επικοινωνία με αυτόν. Με

αυτή τη διαδικασία εγγυάται ότι κάθε κόμβος διορθώνει την ομάδα φύλλων, εκτός αν $|L|/2$ κόμβοι με διαδοχικά αναγνωριστικά καταρρεύσουν ταυτόχρονα. Η πιθανότητα για ένα τέτοιο ενδεχόμενο είναι πολύ μικρή.

Η κατάρρευση ενός κόμβου που εμφανίζεται στον πίνακα δρομολόγησης ενός άλλου κόμβου, διαπιστώνεται όταν ο κόμβος προσπαθεί να επικοινωνήσει με τον κόμβο που έχει καταρρεύσει και δεν λαμβάνει απάντηση. Αυτό το γεγονός δεν καθυστερεί τη δρομολόγηση ενός μηνύματος, διότι το μήνυμα μπορεί να προωθηθεί σε έναν άλλο κόμβο. Παρ'όλα αυτά πρέπει να βρεθεί μια καταχώρηση για να αντικαταστήσει την παλιά και να διατηρηθεί η ακεραιότητα του πίνακα δρομολόγησης.

Για τη διόρθωση μιας καταχώρησης R_1^d , ενός πίνακα δρομολόγησης ένας κόμβος επικοινωνεί πρώτα με τον κόμβο που αναφέρεται μια άλλη καταχώρηση $R_1^i, i \neq d$ της ίδιας σειράς και ζητά την καταχώρηση εκείνου του κόμβου για την R_1^d . Στην περίπτωση όπου καμμία από τις καταχωρήσεις στη σειρά 1, δεν έχουν ένα δείκτη σε έναν κόμβο με το κατάλληλο πρόθεμα, ο κόμβος επικοινωνεί με την καταχώρηση $R_{i+1}^i, i \neq d$, δημιουργώντας έτσι ένα ευρύτερο δίκτυο. Με αυτή τη διαδικασία είναι πολύ πιθανό να βρεθεί ο κατάλληλος κόμβος, αν αυτός υπάρχει.

Ο πίνακας γειτνίασης δε χρησιμοποιείται συνήθως στη δρομολόγηση των μηνυμάτων, είναι όμως σημαντικό να διατηρείται ενημερωμένος γιατί έχει σημαντικό ρόλο στην ανταλλαγή πληροφοριών για κοντινούς κόμβους. Για αυτό το λόγο, ένας κόμβος επιχειρεί να επικοινωνεί με κάθε μέλος της ομάδας γειτνίασης τακτικά για να ελέγχει αν είναι ακόμα ενεργό. Αν ένα μέλος δεν απαντά, τότε ο κόμβος ζητά από τα άλλα μέλη τους πίνακες γειτνίασής τους, ελέγχει την απόσταση κάθε νέου κόμβου και ενημερώνει τη δική του ομάδα γειτνίασης.

4.5 Τοποθεσία

Οι ιδιότητες του πρωτόκολου Pastry σχετικά με την τοποθεσία είναι σημαντικές. Δηλαδή, η πορεία που διαλέγεται για ένα μήνυμα είναι πιθανό να είναι καλή σε σχέση με την μέτρηση της εγγύτητας των κόμβων.

Η εγγύτητα των κόμβων βασίζεται στον αριθμό των IP βημάτων δρομολόγησης ή της γεωγραφικής απόστασης. Υποτίθεται ότι η εφαρμογή παρέχει μια συνάρτηση που επιτρέπει σε κάθε κόμβο να προσδιορίζει την απόσταση ενός κόμβου με μια δεδομένη διεύθυνση IP για τον ίδιο. Ένας κόμβος με μικρότερη τιμή απόστασης θεωρείται πιο κατάλληλος.

Εγγύτητα στον πίνακα δρομολόγησης

Η ιδιότητα αυτή είναι πως όλες οι καταχωρήσεις στους πίνακες δρομολόγησης αναφέρονται στον κόμβο ο οποίος βρίσκεται πιο κοντά στον παρόντα κόμβο, από όλους τους κόμβους που έχουν το κατάλληλο πρόθεμα για την καταχώρηση.

Εγγύτητα δρομολόγησης

Οι καταχωρήσεις στον πίνακα δρομολόγησης κάθε κόμβου επιλέγονται να είναι κοντά στον παρόντα κόμβο. Σαν αποτέλεσμα είναι σε κάθε βήμα δρομολόγησης, ένα μήνυμα προωθείται σε έναν σχετικά κοντινό κόμβο με αναγνωριστικό που έχει μεγαλύτερο κοινό πρόθεμα ή είναι αριθμητικά πλησιέστερο στο κλειδί από τον τοπικό κόμβο. Δηλαδή, με κάθε βήμα το μήνυμα πλησιάζει τον προορισμό στον χώρο του αναγνωριστικού, κάνοντας την ελάχιστη δυνατή απόσταση σε σχέση με την εγγύτητα.

Έτσι το Pastry, ελαχιστοποιεί την απόσταση του επόμενου βήματος δρομολόγησης χωρίς αίσθηση καθολικής κατεύθυνσης. Ακόμα, η αναμενόμενη απόσταση που διανύεται από τα μηνύματα κατά τη διάρκεια κάθε διαδοχικού βήματος δρομολόγησης αυξάνεται εκθετικά. Από αυτές τις ιδιότητες προκύπτει ότι ένα μήνυμα τείνει να κάνει αυξανόμενα βήματα χωρίς πιθανότητα επιστροφής σε κόμβο που πέρασε προηγουμένως. Έτσι το μήνυμα δεν έχει να πάει πουθενά αλλού, παρά στον προορισμό του.

Εντοπισμός του κοντινότερου από k κόμβους

Με τη χρήση του Pastry μερικές εφαρμογές αντιγράφουν την πληροφορία που βρίσκεται σε έναν κόμβο, στους k κόμβους με το αριθμητικά πλησιέστερο αναγνωριστικό. Έτσι φτάνοντας σε έναν από αυτούς τους k κόμβους, είναι δυνατή η

ανάκτηση της απαιτούμενης πληροφορίας χωρίς την απευθείας πρόσβαση στο συγκεκριμένο κόμβο. Έτσι ελαχιστοποιείται το φορτίο του δικτύου.

4.6 Τυχαία κατάρρευση κόμβων και διαχωρισμός του δικτύου

Σε τυχαίες καταρρεύσεις κόμβων, όταν ένας κόμβος που έχει καταρρεύσει συνεχίζει να είναι εν μέρει ενεργός αλλά λειτουργεί λανθασμένα, οι αναζητήσεις ακολουθώντας την ίδια διαδρομή, θα αποτυγχάνουν.

Σε εφαρμογές όπου οι τυχαίες καταρρεύσεις κόμβων πρέπει να είναι ανεκτές, η δρομολόγηση μπορεί να γίνεται με τυχαία κατανομή. Έτσι μια αναζήτηση μπορεί να χρειαστεί αρκετές επαναλήψεις, ώστε να επιλεγεί η διαδρομή που θα αποφεύγει τον κόμβο που έχει καταρρεύσει.

Ένα άλλο σημείο που χρειάζεται προσοχή είναι τα προβλήματα της IP δρομολόγησης στο διαδίκτυο, που προκαλούν μερικούς IP hosts να μην είναι προσβάσιμοι από συγκεκριμένους IP hosts, αλλά όχι από άλλους. Η δρομολόγηση στο Pastry είναι ανεκτική σε αυτά τα προβλήματα. Οι κόμβοι θεωρούνται ενεργοί και προσβάσιμοι στο υπερκείμενο δίκτυο όσο μπορούν να επικοινωνήσουν με τους γειτονικούς τους κόμβους στον χώρο των αναγνωριστικών. Με αυτό τον τρόπο όμως μπορεί να δημιουργηθούν απομονωμένα υπερκείμενα δίκτυα, ακόμα και μετά την αποκατάσταση του προβλήματος. Ένας τρόπος αντιμετώπισης είναι η χρήση της IP πολυεκπομπής. Οι κόμβοι κάνουν μια περιοδική έρευνα πολυεκπομπής επεκτεινόμενου δακτυλίου για άλλους κόμβους στην κοντινή τους περιοχή. Αν τα απομονωμένα υπερκείμενα δίκτυα Pastry εξακολουθούν να υπάρχουν, τότε τελικά ανακαλύπτονται και ενσωματώνονται.

5. Tapestry: Μια δομή για εντοπισμό και δρομολόγηση ευρείας περιοχής

Το Tapestry [4] είναι ένα κλιμακούμενο, ανθεκτικό πρωτόκολο δρομολόγησης ευρείας περιοχής που προωθεί αποτελεσματικά αιτήσεις στα κατάλληλα περιεχόμενα, παρουσία μεγάλου φορτίου, δικτύου και σφάλματα κόμβων. Παρέχει δρομολόγηση μηνυμάτων ανεξαρτήτως τοποθεσίας, απευθείας στον πλησιέστερο κόμβο μιας υπηρεσίας χρησιμοποιώντας σύνδεση από σημείο σε σημείο, χωρίς κεντρικό έλεγχο. Το Tapestry χρησιμοποιεί τυχαίες κατανομές για την επίτευξη κατανομής φορτίου και τοπική δρομολόγηση. Έχει τις ρίζες του κατανεμημένου αλγόριθμου Plaxton, βελτιωμένου με μηχανισμούς για την παροχή διαθεσιμότητας, κλιμάκωσης, και προσαρμογής σε περίπτωση κατάρρευσης κόμβων, ή επιθέσεων.

Το πλεονεκτήματά του είναι ότι προσφέρει σταθερότητα συστήματος μέσω στατιστικής. Τα προβληματικά μέρη εντοπίζονται, διαδρομές που έχουν καταρρεύσει παρακάμπτονται, κόμβοι στους οποίους γίνεται επίθεση απομακρύνονται από υπηρεσίες, και οι τοπολογίες επικοινωνίας προσαρμόζονται γρήγορα στις αλλαγές των συνθηκών.

5.1 Το υπόβαθρο του αλγόριθμου Plaxton

Το Plaxton είναι μια κατανεμημένη δομή δεδομένων, που είναι κατάλληλη για τον εντοπισμό ονομαστικών αντικειμένων και τη δρομολόγηση μηνυμάτων σε αυτά τα αντικείμενα. Επιτρέπει στα μηνύματα να εντοπίζουν τα αντικείμενα και να δρομολογούν σε αυτά, μέσα από ένα δίκτυο τυχαίου μεγέθους, με τη χρήση ενός μικρού χάρτη δρομολόγησης σε κάθε βήμα. Επιπλέον, με το συνδυασμό της εύρεσης του αντικειμένου και της δρομολόγησης σε αυτό, εγγυάται ότι ο χρόνος παράδοσης του μηνύματος τείνει στον βέλτιστο χρόνο παράδοσης, από οποιοδήποτε σημείο στο δίκτυο. Στο Plaxton κάθε κόμβος μπορεί να αναλάβει το ρόλο του εξυπηρετητή, του δρομολογητή, και πελάτη.

5.1.1 Δρομολόγηση στο Plaxton

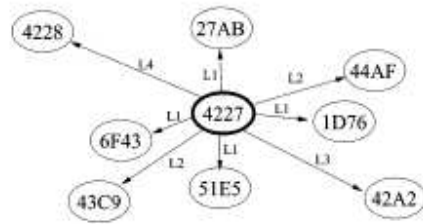
Το Plaxton χρησιμοποιεί τοπικούς χάρτες δρομολόγησης σε κάθε κόμβο, για την συνεχόμενη δρομολόγηση στην ταυτότητα του προορισμού, ψηφίο με ψηφίο. Ο εντοπισμός των ψηφίων γίνεται από δεξιά προς αριστερά, αλλά η απόφαση λαμβάνεται με τυχαίο τρόπο. Ένας κόμβος N έχει έναν χάρτη γειτνίασης πολλαπλών επιπέδων, όπου

κάθε επίπεδο αντιπροσωπεύει ένα σύνολο ψηφίων που ταιριάζει στον συνολικό αριθμό ψηφίων του αναγνωριστικού ID. Ένα δεδομένο επίπεδο του χάρτη γειτνίασης, περιέχει τον αριθμό των καταχωρήσεων ίσους με τη βάση του ID, όπου η καταχώρηση i , στο επίπεδο j , είναι το ID και η τοποθεσία του πλησιέστερου κόμβου που τελειώνει σε 'i'+επίθεμα(N,j-1). Σαν παράδειγμα, η 9^η καταχώρηση του 4^{ου} επιπέδου για τον κόμβο 325AE είναι ο κόμβος πλησιέστερος στο 325AE για απόσταση κόμβων που τελειώνει με 95AE.

Εξ'ορισμού, ο κόμβος n , στον οποίο φτάνει ένα μήνυμα έχει κοινό επίθεμα τουλάχιστον μήκους n , με το ID του προορισμού. Για να βρεθεί ο επόμενος δρομολογητής, γίνεται αναζήτηση στο $n+1$ επίπεδο και βρίσκεται η καταχώρηση που ταιριάζει με την τιμή του επόμενου ψηφίου στο ID του προορισμού. Αυτός ο τρόπος δρομολόγησης εγγυάται ότι κάθε κόμβος στο σύστημα μπορεί να βρεθεί με το πολύ $\log_b N$ βήματα, σε ένα σύστημα με N μέγεθος κόμβων και χρησιμοποίηση ID με βάση το b . Επειδή κάθε χάρτης γειτνίασης σε έναν κόμβο υποθέτει ότι τα προηγούμενα ψηφία ταιριάζουν με το επίθεμα του παρόντος κόμβου, χρειάζεται μικρό αριθμό καταχωρήσεων σε κάθε επίπεδο δρομολόγησης, περάγοντας τοπικούς χάρτες δρομολόγησης μεγέθους $b \cdot \log_b N$.

Ένας τρόπος απεικόνισης αυτού του μηχανισμού δρομολόγησης είναι ότι κάθε κόμβος προορισμού είναι κόμβος-ρίζα ενός δικού του δέντρου που είναι ένα μοναδικό δέντρο που διατρέχει όλους τους κόμβους. Κάθε φύλλο μπορεί να διασχίσει αρκετούς ενδιάμεσους κόμβους κατά τη δρομολόγησή του στον κόμβο-ρίζα. Δηλαδή, ο αλγόριθμος Plaxton χαρτών γειτνίασης είναι ένα ομαδοποιημένο σύνολο ενσωματωμένων δέντρων του δικτύου.

Στη συνέχεια παρουσιάζεται η δρομολόγηση από την πλευρά ενός μόνο κόμβου. Οι εξερχόμενοι γειτονικοί σύνδεσμοι δείχνουν σε κόμβους με κοινό πρόθεμα. Καταχωρήσεις υψηλότερου επιπέδου ταιριάζουν σε περισσότερα ψηφία. Αυτοί οι σύνδεσμοι, συγκεντρωμένοι αποτελούν τον τοπικό πίνακα δρομολόγησης.



Εικόνα 3: Δρομολόγηση από την πλευρά ενός κόμβου

5.1.2 Εντοπισμός

Ο μηχανισμός εντοπισμού επιτρέπει σε έναν πελάτη να εντοπίζει και να αποστέλλει μηνύματα σε ένα αντικείμενο που βρίσκεται σε έναν εξυπηρετητή. Ένας εξυπηρετητής S, ανακοινώνει ότι έχει ένα αντικείμενο O, δρομολογώντας ένα μήνυμα στον κόμβο-ρίζα του O. Ο κόμβος-ρίζα είναι ένας μοναδικός κόμβος στο δίκτυο που χρησιμοποιείται για την τοποθέτηση της ρίζας του ενσωματωμένου δέντρου του αντικειμένου O. Η διαδικασία προβλέπει την αποστολή μηνύματος στον κόμβο-ρίζα. Σε κάθε βήμα κατά τη διαδρομή του, αποθηκεύει πληροφορίες της μορφής (Αντικείμενο-ID(O), Εξυπηρετητής-ID(S)). Οι πληροφορίες αυτές είναι απλώς δείκτες και όχι αντίγραφα του αντικειμένου A.

Κατά τη διάρκεια της διαδικασίας εντοπισμού, οι πελάτες στέλνουν μηνύματα στα αντικείμενα. Ένα μήνυμα που προορίζεται για το αντικείμενο O, δρομολογείται αρχικά προς τη ρίζα του. Σε κάθε βήμα, αν το μήνυμα συναντήσει έναν κόμβο που περιέχει την τοποθεσία του αντικειμένου O, προωθείται στον εξυπηρετητή που περιέχει το αντικείμενο. Σε άλλη περίπτωση προωθείται ένα βήμα πλησιέστερα στη ρίζα. Αν το μήνυμα φτάσει στη ρίζα, είναι εγγυημένη η εύρεση της τοποθεσίας του αντικειμένου O.

Ο κόμβος-ρίζα ενός αντικειμένου, έχει τη σημαντική λειτουργία να παρέχει έναν εγγυημένο κόμβο όπου μπορεί να βρεθεί η τοποθεσία του αντικειμένου. Δεν υπάρχει κάποιος ειδικός συσχετισμός μεταξύ ενός αντικειμένου και του κόμβου-ρίζα που του έχει ανατεθεί. Το Plaxton χρησιμοποιεί έναν ντετερμινιστικό αλγόριθμο για την επιλογή κόμβων-ριζών, με έναν περιορισμό, η ολική ενημέρωση χρησιμοποιείται για την επιλογή ενός κόμβου από μια μεγάλη βάση. Ενώ τα ενδιάμεσα βήματα δεν είναι απολύτως

απαραίτητα, ο κόμβος-ρίζα εξυπηρετεί έναν σημαντικό σκοπό. Επειδή είναι ο μοναδικός με τα χαρακτηριστικά του, μπορεί να γίνει ένα μοναδικό σημείο αποτυχίας.

5.1.3 Πλεονεκτήματα-Μειονεκτήματα

Υπάρχουν μερικές ιδιότητες στον εντοπισμό και τη δρομολόγηση που συμμετέχουν στην αποτελεσματικότητα του αλγόριθμου Plaxton.

- **Διαχείριση απλών λαθών:** Επειδή η δρομολόγηση γίνεται σε κόμβους με ένα συγκεκριμένο επίθεμα, υπάρχει η δυνατότητα εναλλακτικής δρομολόγησης, γύρω από ένα σφάλμα, επιλέγοντας έναν κόμβο με παρόμοιο επίθεμα.
- **Κλιμάκωση:** Είναι δομικά αποκεντρωμένο, και η δρομολόγηση γίνεται χρησιμοποιώντας τα διαθέσιμα τοπικά δεδομένα. Χωρίς κάποιο κεντρικό σημείο, το μόνο πιθανό σημείο για καθυστέρηση υπάρχει στον κόμβο-ρίζα.
- **Εκμετάλλευση τοπικότητας:** Η εύρεση κάθε επιπλέον ψηφίου του επιθέματος, μειώνει τον αριθμό των υποψηφίων κόμβων του ID κατά b . Ο δρόμος που επιλέγεται για τον κόμβο-ρίζα από τον εξυπηρετητή S που αποθηκεύει το αντικείμενο O , και αυτόν που επιλέγεται από τον πελάτη C , συγκλίνουν με ταχύτητα, γιατί ο αριθμός των υποψηφίων κόμβων μειώνεται με αυξανόμενο ρυθμό σε κάθε βήμα. Έτσι αναζητήσεις για αντικείμενα είναι πιθανό να φτάσουν σε έναν εξυπηρετητή με δείκτη στην τοποθεσία του αντικειμένου, πολύ γρήγορα.
- **Αναλογική απόσταση δρομολόγησης:** Για το Plaxton έχει αποδειχθεί ότι η συνολική απόσταση δικτύου που διανύεται από ένα μήνυμα είναι ανάλογη με την απόσταση του υπάρχοντος δικτύου, διαβεβαιώνοντας έτσι για μια λογική επιβάρυνση.

Υπάρχουν όμως και μερικοί περιορισμοί στον αλγόριθμο Plaxton.

- **Ολική ενημέρωση:** Για την επίτευξη μοναδικής αντιστοίχισης μεταξύ αναγνωριστικών κειμένων και κόμβων-ριζών, ο αλγόριθμος Plaxton απαιτεί ολική ενημέρωση τη στιγμή της αντιστοίχισης. Με αυτό τον τρόπο επιβαρύνεται η διαδικασία πρόσθεσης και απομάκρυνσης κόμβων από το δίκτυο.
- **Ευαισθησία κόμβου-ρίζας:** Ο κόμβος-ρίζα ενός αντικειμένου είναι ένα μοναδικό σημείο αποτυχίας, γιατί είναι ο κόμβος για τον οποίο κάθε πελάτης στηρίζεται για την παροχή πληροφοριών για την τοποθεσία του αντικειμένου. Ενώ οι ενδιάμεσοι κόμβοι μπορούν να παρακαμφθούν, ένας κόμβος-ρίζα χωρίς πρόσβαση, δεν επιτρέπει την πρόσβαση των αντικειμένων από απομακρυσμένους πελάτες.
- **Έλλειψη ικανότητας προσαρμογής:** Ο αλγόριθμος Plaxton δεν έχει την ικανότητα να προσαρμόζεται σε δυναμικά σχήματα αναζήτησης, όπως

επαναλαμβανόμενα κλειδιά. Με τον ίδιο τρόπο πιθανά προβληματικά σημεία δεν διορθώνονται πριν δημιουργηθεί υπερφόρτωση.

5.2 Οι λειτουργίες του υπερκείμενου δικτύου Tapestry

Το Tapestry είναι μια δομή υπερκείμενου δικτύου, που εξυπηρετεί τη δημιουργία κλιμακούμενων εφαρμογών, ανεκτικών σε λάθη σε ένα δυναμικό δίκτυο ευρείας περιοχής.

5.2.1 Δρομολόγηση στο Tapestry

Οι βασικές λειτουργίες των μηχανισμών εντοπισμού και δρομολόγησης είναι παρόμοιες με αυτές του Plaxton. Κάθε πίνακας γειτνίασης οργανώνεται σε επίπεδα δρομολόγησης, με κάθε επίπεδο να περιέχει καταχωρήσεις που δείχνουν σε μια ομάδα κόμβων που είναι κοντινή σε απόσταση δικτύου, η οποία ταιριάζει με το επίθεμα για αυτό το επίπεδο. Κάθε κόμβος διατηρεί μια εναλλακτική λίστα δεικτών, που δείχνει σε κόμβους στους οποίους αναφέρεται σαν γείτονας.

Ο μηχανισμός εντοπισμού του Tapestry είναι παρόμοιος με αυτόν του Plaxton. Κάθε κόμβος στη διαδρομή προς τον κόμβο-ρίζα αποθηκεύει την τοποθεσία του κοντινότερου αντιγράφου του, όταν υπάρχουν πολλαπλά αντίγραφα στο Plaxton. Στο Tapestry, αποθηκεύονται οι τοποθεσίες όλων των αντιγράφων για να αυξηθεί η ευελιξία στη σημασιολογία. Έτσι το Tapestry επιτρέπει στην εφαρμογή να προσδιορίσει κάποιο συγκεκριμένο αντίγραφο, ενώ το Plaxton επιστρέφει πάντα το πλησιέστερο. Κάθε αντικείμενο μπορεί να περιέχει μια επιπλέον τιμή που θα καθορίζεται από την εφαρμογή. Έτσι η εφαρμογή μπορεί έτσι να προσδιορίσει από το επιπλέον χαρακτηριστικό και την τιμή του, το κατάλληλο αντικείμενο.

5.2.2 Διαχείριση σφαλμάτων

Η ικανότητα του εντοπισμού, της παράκαμψης και της επαναφοράς από σφάλματα είναι ένας βασικός στόχος του Tapestry. Η επιλογή σχεδιασμού γίνεται ώστε τα μέρη του Tapestry να απευθύνονται στο θέμα της προσαρμογής σε σφάλματα, με τη χρήση των

προσφάτως χρησιμοποιημένων δεδομένων για την επαναφορά από τα σφάλματα. Αυτό επιτρέπει στο Tapestry να χειρίζεται τα σφάλματα σαν μέρος της λειτουργίας του, αντί να τα αντιμετωπίζει σαν ξεχωριστές περιπτώσεις. Τα σφάλματα είναι ένα αναμενόμενο γεγονός στη λειτουργία σε ευρεία περιοχή. Στη συνέχεια επεξηγείται, πώς εντοπίζονται, παρακάμπτονται και επαναφέρεται το σύστημα από τα σφάλματα στη λειτουργία του εντοπισμού και της δρομολόγησης.

5.2.2.1 Δρομολόγηση ανεκτική σε σφάλματα

Τα είδη των αναμενόμενων σφαλμάτων είναι διακοπές λειτουργίας του εξυπηρετητή, διακοπές συνδέσεων, και λάθη στους πίνακες γειτνίασης των εξυπηρετητών. Τα σφάλματα εντοπίζονται γρήγορα, διορθώνονται και στη συνέχεια αποκαθίσταται η λειτουργία των δρομολογητών.

Για τον εντοπισμό της διακοπής λειτουργίας του εξυπηρετητή και τις διακοπές συνδέσεων, το Tapestry στηρίζεται στην εκπονή TCP. Επιπλέον, κάθε κόμβος στέλνει περιοδικά, πακέτα που διαβεβαιώνουν τους γειτονικούς κόμβους ότι είναι αξιόπιστοι για δρομολόγηση. Ελέγχοντας το ID του κάθε κόμβου στον οποίο φτάνει ένα μήνυμα, μπορούν να εντοπιστούν εσφαλμένοι πίνακες γειτνίασης.

Για τη λειτουργία κατά τη διάρκεια των σφαλμάτων, κάθε καταχώρηση στον χάρτη γειτνίασης διατηρεί δύο εναλλακτικούς γειτονικούς κόμβους επιπρόσθετα με τον πλησιέστερο γειτονικό κόμβο. Αυτοί αναφέρονται ως δευτερεύοντες γειτονικοί κόμβοι. Με την κατάρρευση του πρωτεύοντος γειτονικού κόμβου, γίνεται η προσαρμογή στον δευτερεύοντα γειτονικό κόμβο.

Τέλος, αποφεύγονται οι επανασυνδέσεις των κόμβων όταν έχει επιδιορθωθεί μια βλάβη, λόγω κόστους. Όταν ένας κόμβος αντιλαμβάνεται ότι δεν έχει πρόσβαση σε έναν γειτονικό του κόμβο, συνεχίζει τη δρομολόγηση από εναλλακτική διαδρομή. Επειδή οι περισσότερες καταρρεύσεις κόμβων και οι διακοπές συνδέσεων εντοπίζονται και διορθώνονται σε σχετικά μικρό χρονικό διάστημα, δίνεται χρονικό διάστημα δεύτερης ευκαιρίας κατά τη διάρκεια του οποίου στέλνονται μηνύματα στον εξυπηρετητή. Αν το μήνυμα δεν είναι επιτυχημένο τότε χρησιμοποιείται η εναλλακτική διαδρομή. Αλλιώς, το μήνυμα είναι επιτυχημένο, θεωρείται ότι το σφάλμα διορθώθηκε και επιλέγεται η αρχική διαδρομή. Αν λήξει η χρονική περίοδος χωρίς τη διόρθωση του σφάλματος, τότε ο

αρχικός γειτονικός κόμβος απομακρύνεται από τον χάρτη γεινίασης, προωθείται ο εναλλακτικός και επιλέγεται ένας άλλος για εναλλακτικός.

5.2.2.2 Εντοπισμός ανεκτικός σε σφάλματα

Όπως διατυπώθηκε προηγουμένως ο κόμβος-ρίζα ενός αντικειμένου είναι μοναδικό σημείο αποτυχίας στον αλγόριθμο Plaxton. Στο Tapestry αυτό διορθώνεται αντιστοιχίζοντας πολλαπλούς κόμβους-ρίζες για για κάθε αντικείμενο. Για να επιτευχθεί αυτό, συνδέεται μια αλληλουχία ειδικών τιμών στο ID του κάθε αντικειμένου και στη συνέχεια με χρήση συνάρτησης ανάμειξης προκύπτουν οι κατάλληλοι κόμβοι-ρίζες. Αυτοί χρησιμοποιούνται για την εισαγωγή πληροφοριών εντοπισμού στο δίκτυο Tapestry. Κατά την αναζήτηση ενός αντικειμένου, το Tapestry υλοποιεί την ίδια συνάρτηση ανάμειξης με στόχο το ID του αντικειμένου, λαμβάνοντας σαν αποτέλεσμα μια ομάδα κόμβων-ριζών για αναζήτηση.

5.2.3 Εναλλακτική δρομολόγηση

Στον αλγόριθμο Plaxton, ο κόμβος-ρίζα επιλέγεται ως ο κόμβος ο οποίος ταιριάζει με το ID(έστω I), με το μεγαλύτερο αριθμό bit. Αφού μπορεί να υπάρχουν πολλοί κόμβοι που πληρούν τις προϋποθέσεις, επιλέγεται ένας μοναδικός κόμβος-ρίζα με την πρόσκληση όλων των κόμβων του δικτύου. Ο κόμβος με το καλύτερο αποτέλεσμα επιλέγεται σαν κόμβος-ρίζα. Ο εντοπισμός στο Plaxton λειτουργεί αναλύοντας το ID του αντικειμένου ψηφίο προς ψηφίο έως ότου συναντήσει μια κενή καταχώρηση για τους γειτονικούς κόμβους. Σε αυτό το σημείο κάνει ένα τελευταίο βήμα στον κόμβο-ρίζα.

Το Tapestry χρησιμοποιεί τον αλγόριθμο εναλλακτικής δρομολόγησης για να υπολογίσει τον μοναδικό κόμβο-ρίζα. Ο αλγόριθμος είναι ντετερμινιστικός, κλιμακούμενος, και αποτελεσματικός από κάθε σημείο στο δίκτυο. Η εναλλακτική δρομολόγηση επιλέγει δοκιμαστικά τον κόμβο-ρίζα ενός αντικειμένου να έχει την ίδια ονομασία με το ID, I. Με δεδομένο το αραιό πλήθος των πιθανών ονομασιών είναι απίθανο αυτός να υπάρχει. Παρ'όλα αυτά το Tapestry συνεχίζει τη δρομολόγηση σαν να υπήρχε. Η δρομολόγηση σε έναν ανύπαρκτο δείκτη, θα συναντήσει κενές καταχωρήσεις για τους γειτονικούς κόμβους. Ο στόχος είναι η επιλογή ενός κόμβου σαν εναλλακτικού συνδέσμου. Αυτή η επιλογή γίνεται με ντετερμινιστική επιλογή ανάμεσα στους

γειτονικούς δείκτες. Η δρομολόγηση τερματίζεται όταν σε έναν χάρτη γειτνίασης, η μοναδική μη κενή καταχώρηση ανήκει στον τρέχοντα κόμβο. Αυτός ο κόμβος καθορίζεται ως ο εναλλακτικός κόμβος-ρίζα για το αντικείμενο.

Ένας σύνδεσμος σε γειτονικό κόμβο μπορεί να είναι κενός μόνο αν δεν υπάρχουν υποψήφιοι κόμβοι σε ολόκληρο το δίκτυο. Έτσι, γειτονικοί κόμβοι σε ολόκληρο το δίκτυο θα έχουν κενές καταχωρήσεις στον χάρτη γειτνίασης αν και μόνο αν όλοι οι κόμβοι με αυτό το επίθεμα έχουν ακριβώς τις ίδιες κενές καταχωρήσεις. Ένας ντετερμινιστικός αλγόριθμος θα έφτανε στον ίδιο μοναδικό εναλλακτικό κόμβο-ρίζα από οποιαδήποτε τοποθεσία στο δίκτυο Tapestry. Η εναλλακτική δρομολόγηση παρέχει μια τεχνική με την οποία κάθε δείκτης μπορεί να αντιστοιχιστεί μοναδικά σε έναν κόμβο του δικτύου.

Ενώ η εναλλακτική δρομολόγηση μπορεί να χρειαστεί μερικά βήματα παραπάνω σε σύγκριση με τον αλγόριθμο Plaxton, ο αριθμός των επιπρόσθετων βημάτων όμως παραμένει μικρός.

5.3 Δυναμικοί αλγόριθμοι

Ο μόνος περιορισμός στην πρόταση του Plaxton είναι η στατική φύση των αλγορίθμων του. Στο Tapestry προτείνονται αλγόριθμοι που επικεντρώνονται σε δυναμικές λειτουργίες με αποκεντρωμένη μέθοδο. Έτσι επιτυγχάνονται αρκετές από τις επιθυμητές ιδιότητες που έχουν αναφερθεί.

5.3.1 Δυναμική είσοδος κόμβου

Ο αλγόριθμος για τη δυναμική είσοδο κόμβου δεν εγγυάται ιδανική τοπολογία, αλλά είναι μια καλή προσέγγιση που θα μπορεί να συγκλίνει σε αυτήν.

Το βασικό μέρος του αλγορίθμου είναι το παρακάτω. Αρχικά, συμπληρώνονται οι πίνακες γειτνίασης του νέου κόμβου με δρομολόγηση στο ID. Στη συνέχεια αντιγράφονται και βελτιστοποιούνται σε κάθε βήμα από το δρομολογητή. Ενημερώνονται οι γειτονικοί κόμβοι για την άφιξη του νέου κόμβου, ώστε να ενημερώσουν τους δικούς τους πίνακες γειτνίασης.

5.3.1.1 Συμπλήρωση πίνακα γειτνίασης

Ο κόμβος N γνωρίζει τον κόμβο εξόδου G, που είναι ένας κοντινός του κόμβος σε απόσταση στο δίκτυο. Ξεκινώντας από τον κόμβο G, ο κόμβος N επιχειρεί να δρομολογήσει προς το new_id, που είναι το νέο αναγνωριστικό ID, και αντιγράφει έναν προσεγγιστικό πίνακα γειτνίασης από το βήμα i , H_i με τον αρχικό κόμβο να είναι $G = H_0$. Με τη δρομολόγηση στο δικό του ID, ο κόμβος N γνωρίζει ότι μοιράζεται με το H_i ένα επίθεμα μήκους i . Αντιγράφει τον πίνακα γειτνίασης και προσπαθεί να βελτιστοποιήσει κάθε καταχώρηση. Η βελτιστοποίηση γίνεται συγκρίνοντας τις αποστάσεις μεταξύ του N και κάθε γειτονικής καταχώρησης και των δευτερευόντων γειτονικών κόμβων. Ο νέος κόμβος σταματά να αντιγράφει πίνακες γειτνίασης, όταν η αναζήτηση δείχνει σε κενή καταχώρηση στο επόμενο βήμα. Τότε μετακινεί τα δεδομένα που προορίζονται για το new_id στον κόμβο N.

5.3.1.2 Ενημέρωση γειτονικών κόμβων

Το επόμενο βήμα είναι η ενημέρωση των κόμβων για την άφιξη του νέου κόμβου N. Για τους κόμβους που έχουν άδεια καταχώρηση στη θέση που έπρεπε να είναι ο κόμβος N, διασχίζονται οι εναλλακτικοί δείκτες μέχρι το επίπεδο όπου η εναλλακτική δρομολόγηση ήταν απαραίτητη. Για την ενημέρωση των τοπικών κόμβων που μπορεί να επωφεληθούν από τον κόμβο N σαν νέο κόμβο δρομολόγησης, στέλνονται μηνύματα σε όλους τους γειτονικούς κόμβους σε κάθε επίπεδο. Οι κόμβοι που έχουν ειδοποιηθεί έχουν την επιλογή να υπολογίσουν την απόσταση με τον κόμβο N, και αν είναι κατάλληλος να αντικαταστήσουν την καταχώρηση με έναν γειτονικό κόμβο με το N.

Με τη διαδικασία ενημέρωσης των κόμβων για την αλλαγή των καταχωρήσεών τους για τον κόμβο N, μπορεί ακούσια να αλλαχθεί ο εναλλακτικός τρόπος δρομολόγησης, λόγω της συμπλήρωσης κενής θέσης. Αυτό το πρόβλημα λύνεται με τον κόμβο N να αποθηκεύει αντιστοιχίσεις αντικειμένων με τις τοποθεσίες τους. Έτσι μπορούν να διατηρηθούν οι δρομολογήσεις προς τα αντικείμενα χωρίς διακοπές.

Κάθε είσοδος κόμβου απαιτεί ένα μη αμελητέο χρονικό διάστημα. Αυτό είναι μέρος της λογικής για την εισαγωγή των διορθωμένων κόμβων. Η αναχώρηση κόμβων απαιτεί μικρότερο χρονικό διάστημα. Ένας κόμβος μπορεί να ενημερώσει τους υπόλοιπους για

την αναχώρησή του. Παρ'όλο που το δίκτυο αναμένεται να είναι δυναμικό, μόνο μικρό μέρος του δικτύου εισέρχεται ή εξέρχεται ταυτόχρονα.

5.3.2 Φορτίο εύρους ζώνης

Για τη διατήρηση ενημερωμένων δεικτών που δείχνουν σε αντικείμενα, η ανανέωση σε τακτά χρονικά διαστήματα είναι μια ικανή λύση, επιφέρει βάρος στο εύρος ζώνης της γραμμής. Το ίδιο ισχύει και για την ενημέρωση των πληροφοριών σχετικά με τους κόμβους.

Έτσι, χρειάζεται μια προσέγγιση με ειδικές ανανεώσεις, τόσο για τους κόμβους όσο και για τα αντικείμενα. Η αντιστοίχιση των αντικειμένων στις τοποθεσίες χρειάζονται τα τρία χαρακτηριστικά ObjectID, ServerID, LastHopID. Αυτά αναφέρονται στα αντικείμενα, στον εξυπηρετητή και στο τελευταίο βήμα. Για κάθε βήμα προς τον κόμβο-ρίζα, κάθε εξυπηρετητής αποθηκεύει το ID του προηγούμενου κόμβου ως το LastHopID. Με αυτό τον τρόπο γίνεται καλύτερη κατανομή του φορτίου στο εύρος ζώνης.

6. Υπερκείμενο δίκτυο Mithos

Το υπερκείμενο δίκτυο Mithos [5] συνδυάζει την ελαχιστοποίηση της πληροφορίας δρομολόγησης, τη χρήση των κατανεμημένων πινάκων ανάμειξης, καθώς και αρκετά πλεονεκτήματα όπως τη χρήση του πρωτοκόλου IPv6 σαν μηχανισμό μεταφοράς. Σε αντίθεση με άλλα προτεινόμενα υπερκείμενα δίκτυα που χρησιμοποιούν καρτεσιανές συντεταγμένες, το Mithos χρησιμοποιεί κάθε κόμβο σε ολόκληρο το δίκτυο σαν σημείο αναφοράς της τοπολογίας. Έτσι επιτυγχάνεται ακρίβεια και αποτελεσματικότητα χωρίς την επιβάρυνση πολλών διαστάσεων.

Στο Mithos, ο πίνακας δρομολόγησης ελαχιστοποιείται γιατί κάθε κόμβος χρειάζεται να γνωρίζει μόνο τους γειτονικούς του κόμβους. Για να επιτευχθεί η ελαχιστοποίηση χρησιμοποιείται μια προσέγγιση στη δρομολόγηση με ακανόνιστο δίκτυο.

6.1 Σχεδιασμός του Mithos

Η βασική ιδέα του Mithos είναι η ενσωμάτωση του δικτύου σε ένα πολυδιάστατο χώρο, με κάθε κόμβο να αντιστοιχίζεται σε συγκεκριμένες συντεταγμένες σε αυτόν. Είναι παρόμοια με διασυνδέσεις που χρησιμοποιούνται σε πολλούς παράλληλους υπολογιστές υψηλής απόδοσης, δίνοντας τη δυνατότητα δρομολόγησης με απλή ενημέρωση για τις τοπικές συντεταγμένες. Αντίθετα με τους παράλληλους υπολογιστές όμως, το πλεγματοειδές δίκτυο που χρησιμοποιείται στο Mithos δεν είναι κανονικό, έτσι ώστε να προσαρμόζεται στη δυναμικότητα του δικτύου.

Για κάθε νέο κόμβο που εισέρχεται στο δίκτυο ακολουθούνται τρεις φάσεις.

1. Έυρεση γειτονικών κόμβων και δημιουργία γειτονικής περιοχής
2. Ανάθεση αναγνωριστικού στο νέο κόμβο με βάση τη γειτονική περιοχή που ανήκει
3. Εγκατάσταση συνδέσμων με τη γειτονική περιοχή

6.2 Έυρεση γειτονικών κόμβων

Για την εξασφάλιση ότι οι γειτονικοί κόμβοι στο υπερκείμενο δίκτυο είναι σε κοντινή απόσταση και στο υπάρχον δίκτυο, χρειάζεται να οριστούν ένας τρόπος απόστασης και η διαδικασία εντοπισμού. Επιλέγεται η καθυστέρηση ανάμεσα σε δύο κόμβους σαν μέτρο υπολογισμού της απόστασης.

Η συνδεσιμότητα και οι παράμετροι σύνδεσης δεν είναι πάντα συμμετρικοί στο διαδίκτυο, ειδικά όταν εμπλέκονται αυτόνομα συστήματα. Παρ'όλα αυτά, αυτός ο τρόπος παρέχει την αρχική βάση για το υπερκείμενο δίκτυο. Κατά την οργάνωση ενός πυκνού υπερκείμενου δικτύου που στόχος του είναι η ελαχιστοποίηση των παραμέτρων σύνδεσης για κάθε σύνδεσμο, το υπερκείμενο δίκτυο θα προσαρμοστεί προσπαθώντας να μεγιστοποιήσει το βαθμό εξυπηρέτησης από το ήδη υπάρχον δίκτυο.

Κατά την αναζήτηση γειτονικών κόμβων, η διαδικασία γίνεται με απλή εκπομπή. Για την είσοδο ενός κόμβου απαιτείται ο υποψήφιος κόμβος να γνωρίζει πώς να επικοινωνήσει με έναν κόμβο που είναι μέρος του δικτύου. Μια μη κενή υποομάδα του δικτύου επιλέγεται σαν πρώτη υποψήφια γειτονικών κόμβων. Στη συνέχεια χρησιμοποιούνται δεδομένα από το υπερκείμενο δίκτυο της πραγματικής γειτονικής περιοχής. Κάθε υποψήφιος γειτονικός κόμβος ελέγχεται για τους δικούς του άμεσους γειτονικούς κόμβους, στη συνέχεια μετράται η απόσταση από αυτούς τους κόμβους. Ο καλύτερος κόμβος είναι ο νέος υποψήφιος γειτονικός κόμβος. Αυτή η διαδικασία ακολουθείται μέχρι το σημείο όπου βελτιστοποιείται η απόσταση των κόμβων. Η διαδικασία τείνει να τερματίζει στο τοπικό ελάχιστο και όχι στο γενικό ελάχιστο του δικτύου. Για αυτό το λόγο στο Mithos ελέγχονται οι κόμβοι που βρίσκονται δύο βήματα μακριά πριν τερματιστεί η διαδικασία. Αν βρεθεί καταλληλότερος υποψήφιος η διαδικασία συνεχίζεται.

6.3 Ανάθεση αναγνωριστικού

Σε αυτή τη φάση γίνεται ανάθεση ταυτότητας αναγνωριστικού στον υποψήφιο κόμβο. Η διαδικασία αυτή είναι σημαντική γιατί μια ακατάλληλη ονοματοδοσία μπορεί να δημιουργήσει πολλά τοπικά ελάχιστα, αποτρέποντας μια αποτελεσματική γειτονική περιοχή κόμβων στο μέλλον.

Το Mithos χρησιμοποιεί την μέτρηση της απόστασης στο τελευταίο βήμα της δημιουργίας γειτονικής περιοχής σαν βάση για την ανάθεση αναγνωριστικού. Οι δύο πλησιέστεροι κόμβοι που βρίσκονται στη διαδικασία, οι γειτονικοί τους κόμβοι και οι σχετικές τους αποστάσεις χρησιμοποιούνται στον υπολογισμό.

Για τον υπολογισμό του αναγνωριστικού, δημιουργούνται εικονικές ‘ελαστικές’ συνδέσεις μεταξύ του υποψήφιου μέλους και τους γειτονικούς κόμβους που του έχουν ανατεθεί. Το βάρος των συνδέσεων είναι αντιστρόφως ανάλογο με τη μετρούμενη απόσταση. Η θέση του υποψήφιου κόμβου με το ελάχιστο βάρος χρησιμοποιείται άμεσα για το αναγνωριστικό.

Τώρα που το αναγνωριστικό κατοχυρώθηκε, οι αποστάσεις υπολογίζονται στο χώρο των αναγνωριστικών ανάλογα με τον τρόπο υπολογισμού της απόστασης, χωρίς την ανάγκη επιπλέον ανταλλαγής μηνυμάτων.

6.4 Εγκατάσταση συνδέσμων

Η τελευταία φάση είναι η εγκατάσταση συνδέσμων μεταξύ γειτονικών κόμβων. Για την αξιολόγηση των πιθανών επιλογών για τη διασύνδεση των γειτονικών κόμβων υπάρχουν τρία κριτήρια.

1. Ελάχιστο μέγεθος πίνακα δρομολόγησης
2. Αποτελεσματική συνδεσιμότητα, πλήρης πρόσβαση
3. Απλός και γρήγορος αλγόριθμος προώθησης.

Αυτοί οι στόχοι επιτυγχάνονται από τις διασυνδέσεις που χρησιμοποιούνται σε πολλούς παράλληλους υπολογιστές. Στα δυναμικά δίκτυα όμως, με το κριτήριο της απόστασης μεταξύ γειτονικών κόμβων αυτό αλλάζει. Εναλλακτικοί τρόποι σύνδεσης σε δυναμικά περιβάλλοντα περιγράφονται στη συνέχεια.

Πλησιέστερα στον άξονα: Κατά μήκος κάθε άξονα σε κάθε κατεύθυνση, βρίσκεται ο κόμβος που είναι πλησιέστερος στον άξονα και επιτυγχάνεται η σύνδεση. Στη συνέχεια εκτελείται η προώθηση των μηνυμάτων.

Με βάση τα τεταρτημόρια: Κάθε κόμβος επιτυγχάνει σύνδεση στον πλησιέστερο γειτονικό κόμβο σε κάθε τεταρτημόριο. Κατά την προώθηση, για το επόμενο βήμα επιλέγεται ο γειτονικός κόμβος στο ίδιο τεταρτημόριο ως τελικός προορισμός.

Ορθογώνια υποδιαίρεση: Σε κάθε κόμβο αντιστοιχίζεται ένα πολυδιάστατο ορθογώνιο, παράλληλο με τους άξονες. Η προώθηση γίνεται στο σημείο του ορθογωνίου όπου το διάνυσμα προορισμού τέμνει τα ορθογώνια όρια του τρέχοντος κόμβου.

Τριγωνοποίηση Delauney: Η επίτευξη της σύνδεσης γίνεται ανάλογα με την τριγωνοποίηση Delauney των κόμβων. Η προώθηση γίνεται στον κόμβο που είναι γωνιακά πλησιέστερος στο διάνυσμα προορισμού.

Όλες αυτές οι προσεγγίσεις επιτυγχάνουν μικρούς πίνακες δρομολόγησης και στη χειρότερη περίπτωση ένας κόμβος μπορεί να έχει όλους τους άλλους κόμβους στο σύστημα σαν γειτονικούς κόμβους.

Η συνδεσιμότητα είναι αποτελεσματική, εκτός από την προσέγγιση για τον πλησιέστερο στον άξονα κόμβο, που δεν επιτυγχάνει την αναγνώριση κόμβων εκτός του άξονα που είναι πλησιέστεροι από τον επόμενο κόμβο εντός του άξονα. Η ορθογώνια υποδιαίρεση και η τριγωνοποίηση Delauney απαιτούν πολύπλοκους υπολογισμούς και συγκρίσεις. Έτσι ο μηχανισμός με βάση τα τεταρτημόρια είναι ο μοναδικός που πληροί τις προϋποθέσεις.

6.5 Επίτευξη συνδέσεων με βάση τα τεταρτημόρια

Όταν ο εισερχόμενος κόμβος J έχει αποκτήσει το αναγνωριστικό του, οι υπόλοιποι γειτονικοί κόμβοι που τον βοήθησαν να την αποκτήσει μπορεί να μην έχουν πληροφορίες για τον πλησιέστερο γειτονικό κόμβο του στο τεταρτημόριο. Αυτό μπορεί να συμβαίνει επειδή οι γειτονικοί κόμβοι δεν έχουν τις πληροφορίες για την τελική θέση του κόμβου J . Επιπλέον, επειδή ο κόμβος J ξέρει έναν κόμβο στο τεταρτημόριο δεν σημαίνει αυτός ο κόμβος είναι και ο πλησιέστερος στον J . Έτσι ο κόμβος J πρέπει να προσδιορίσει τους πλησιέστερους γειτονικούς κόμβους στην περιοχή.

Τώρα που ολόκληρη η γειτονική περιοχή έχει οριοθετηθεί, πρέπει να διασφαλιστεί ότι οι συνδέσεις επιτυγχάνονται στους πλησιέστερους γειτονικούς κόμβους για να είναι εγγυημένη η σωστή προώθηση. Για αυτό το λόγο στη δεύτερη φάση προσπαθεί να εντοπίσει έναν πλησιέστερο γειτονικό κόμβο ξεκινώντας από τον γνωστό κόμβο και ψάχνοντας προς τα όρια του τεταρτημόριου.

Σε αυτή τη δεύτερη φάση χρησιμοποιείται η επεξεργασία παράλληλων μονοπατιών, που μπορεί να βελτιωθεί λαμβάνοντας υπ'όψιν τα γεωμετρικά χαρακτηριστικά των κόμβων.

7. Σύνοψη και σύγκριση υπερκείμενων δικτύων

Τα υπερκείμενα δίκτυα που περιγράφηκαν προηγουμένως παρουσιάζουν αρκετές ομοιότητες αλλά και διαφορές. Η βασική τους ομοιότητα είναι η χρήση των κατανεμημένων πινάκων ανάμειξης για τη διαδικασία της δρομολόγησης. Στη συνέχεια παρουσιάζονται περισσότερες ομοιότητες και διαφορές μεταξύ τους.

7.1 Σύνοψη και σύγκριση Tapestry-Pastry

Το υπερκείμενο δίκτυο Tapestry είναι ένα δίκτυο το οποίο βασίζεται στον αλγόριθμο Plaxton. Στο Tapestry όμως επιδιώκεται η βελτίωση στην ανίχνευση, παράκαμψη και ανάκαμψη από καταρρεύσεις κόμβων διατηρώντας στη μνήμη περιοδικά ανανεωμένο περιεχόμενο. Έχει τα παρακάτω χαρακτηριστικά. Για την ανίχνευση την κατάρρευση εξυπηρετητών και συνδέσεων κατά τη διάρκεια της κανονικής λειτουργίας, το Tapestry χρησιμοποιεί τις εκποές του TCP. Επιπλέον κάθε κόμβος του δικτύου επικοινωνεί με τους γειτονικούς του κόμβους για να δηλώσει την παρουσία του.

Το υπερκείμενο δίκτυο Pastry παρουσιάζει αρκετές ομοιότητες με το υπερκείμενο δίκτυο Tapestry. Οι βασικές ομοιότητες είναι η χρήση κοινών προθεμάτων και υποθεμάτων κατά τη δρομολόγηση, ομοιότητες κατά την εισαγωγή και την αποχώρηση των κόμβων αλλά και στο κόστος επιβάρυνσης για την αποθήκευση, αλλά και την διατήρηση επικοινωνίας μεταξύ των κόμβων. Συγκεκριμένα για τη δρομολόγηση, στο δίκτυο Pastry η δρομολόγηση γίνεται με την προώθηση των μηνυμάτων στον κόμβο με τον οποίο έχει κοινό πρόθεμα τουλάχιστον κατά ένα ψηφίο μεγαλύτερο από τον τρέχοντα κόμβο. Στο δίκτυο Tapestry η δρομολόγηση των μηνυμάτων γίνεται προς τον n κόμβο με τον οποίο έχει κοινό υπόθεμα τουλάχιστον κατά μήκος n .

Υπάρχουν όμως και διαφορές που διαχωρίζουν το υπερκείμενο δίκτυο Pastry από το Tapestry. Στο δίκτυο Pastry μπορεί να υπάρχουν αντικείμενα τα οποία αντιγράφονται χωρίς έλεγχο από τον χρήστη [8]. Με τη χρησιμοποίηση ενός αντικειμένου στο δίκτυο δημιουργούνται αντίγραφα τα οποία τοποθετούνται σε κόμβους με γειτονικά αναγνωριστικά. Στο υπερκείμενο δίκτυο Tapestry δεν υπάρχει ομάδα φύλλων ενώ χρησιμοποιείται και η τεχνική της εναλλακτικής δρομολόγησης η οποία αντιστοιχίζει κλειδιά σε έναν μοναδικό ενεργό κόμβο του δικτύου.

7.2 Σύνοψη και σύγκριση CAN-Chord

Στο υπερκείμενο δίκτυο CAN κάθε κλειδί αντιστοιχίζεται σε ένα σημείο ενός χώρου d διαστάσεων σύμφωνα με μια συνάρτηση ανάμειξης. Όταν ένας νέος κόμβος εισέρχεται στο δίκτυο, επιλέγει τυχαία ένα σημείο στον χώρο των d διαστάσεων. Τότε γίνεται υπεύθυνος για τη μισή περιοχή στην οποία ανήκει το σημείο και για όλα τα κλειδιά τα αναγνωριστικά των οποίων ανήκουν στην περιοχή αυτή. Ο αριθμός των βημάτων είναι της τάξης του $O(n^{1/d})$.

Το υπερκείμενο δίκτυο Chord είναι ένα δίκτυο στο οποίο αποθηκεύονται ζεύγη κλειδιών – τιμών. Όταν δίνεται ένα κλειδί, ο κόμβος ο οποίος είναι υπεύθυνος για την αποθήκευση της τιμής του κλειδιού προσδιορίζεται με τη χρήση μιας συνάρτησης ανάμειξης που αντιστοιχίζει ένα αναγνωριστικό σε κάθε κόμβο και σε κάθε κλειδί. Κάθε κλειδί αποθηκεύεται στον πρώτο κόμβο του οποίου το αναγνωριστικό είναι ίσο ή ακολουθεί το κλειδί στον χώρο των αναγνωριστικών. Κάθε κόμβος διατηρεί πίνακα δρομολόγησης με πληροφορίες για $O(\log N)$ κόμβους.

Ακόμα στο υπερκείμενο δίκτυο CAN, σε αντίθεση με το δίκτυο Chord, κάθε κόμβος διατηρεί πληροφορίες για τους υπόλοιπους κόμβους της τάξης του $O(d)$, που δεν εξαρτάται από το μέγεθος του δικτύου N . Το κόστος αναζήτησης όμως αυξάνεται γρηγορότερα από $\log N$. Για την περίπτωση κατά την οποία $d = \log N$, ο χρόνος αναζήτησης στο δίκτυο CAN είναι παρόμοιος με αυτόν του δικτύου Chord. Αλλά το δίκτυο CAN δε σχεδιάζεται ώστε το d να μεταβάλλεται όπως μεταβάλλεται το N . Έτσι, αυτή η ταύτιση εμφανίζεται μόνο για το κατάλληλο N ανάλογα με κάποιο σταθερά τιμή διαστάσεων d . Στο υπερκείμενο δίκτυο CAN απαιτείται περιοδική ανακατανομή του χώρου των αναγνωριστικών στους διαθέσιμους κόμβους. Το υπερκείμενο δίκτυο Chord, έχει το πλεονέκτημα της διορθωτικής δρομολόγησης, όταν οι πληροφορίες για την αρχική δρομολόγηση είναι ανακριβείς λόγω της αποχώρησης ή κατάρρευσης ενός κόμβου του δικτύου.

7.3 Σύγκριση Chord-Pastry

Το υπερκείμενο δίκτυο Chord παρουσιάζει γενικές ομοιότητες με το υπερκείμενο δίκτυο Pastry. Χρησιμοποιούν κυκλικό λογικό χώρο αναγνωριστικών, ενώ αντίθετα με το Pastry το Chord προωθεί τα μηνύματα μόνο κατά την ωρολογιακή φορά στον κυκλικό χώρο των αναγνωριστικών. Το Pastry εκτελεί τη δρομολόγηση των πακέτων με με βάση το πλησιέστερο κοινό πρόθεμα και χρησιμοποιεί πίνακες δρομολόγησης, πίνακες γειτνίασης και ομάδα φύλλων, ενώ στο Chord οι κόμβοι διατηρούν έναν δακτυλικό πίνακα που αποτελείται από δείκτες σε άλλους ενεργούς κόμβους.

7.4 Σύνοψη Mithos

Το υπερκείμενο δίκτυο Mithos ενσωματώνει το δίκτυο σε έναν πολυδιάστατο χώρο. Σε κάθε κόμβο ανατίθεται ένα μοναδικό αναγνωριστικό. Κάθε νεοεισερχόμενος κόμβος ακολουθεί την εξής διαδικασία. Βρίσκει κοντινούς κόμβους ώστε να επιτευχθεί σύνδεση μεταξύ τους. Στη συνέχεια ανατίθεται αναγνωριστικό στον νεοεισερχόμενο κόμβο με βάση τους γειτονικούς του κόμβους. Το τελευταίο βήμα είναι η σύνδεση του νέου κόμβου με τους γειτονικούς του κόμβους.

7.4.1 Σύγκριση CAN-Mithos

Σύμφωνα με την παραπάνω διαδικασία για το υπερκείμενο δίκτυο Mithos δεν απαιτούνται πληροφορίες για ολόκληρο το δίκτυο, ενώ η δρομολόγηση των μηνυμάτων είναι αρκετά αποτελεσματική. Στο δίκτυο αυτό επιχειρείται ένας συνδυασμός μεταξύ της δρομολόγησης με βάση την κοντινότητα και της γεωγραφικής θέσης των κόμβων. Σε σχέση με τα υπόλοιπα υπερκείμενα δίκτυα, το υπερκείμενο δίκτυο CAN εμφανίζει ομοιότητες σε σχέση με τα χαρακτηριστικά του υπερκείμενου δικτύου Mithos, αλλά χρησιμοποιεί εντελώς διαφορετική προσέγγιση.

8. Προτεινόμενο υπερκείμενο δίκτυο.

Το ενσύρματο διαδίκτυο έχει εξελιχθεί σε μεγάλο βαθμό, ώστε δίνει τη δυνατότητα χρησιμοποίησης πολλών εφαρμογών σε οικίες, επιχειρήσεις αλλά και σε άλλα πεδία της καθημερινότητας. Το επόμενο βήμα είναι η δυνατότητα χρήσης υπηρεσιών διαδικτύου σε χώρους δημόσιας πρόσβασης με χρήση ασύρματων συσκευών. Κάτι τέτοιο απαιτεί την ικανότητα διαχείρισης της κινητικότητας των τερματικών χρηστών των υπηρεσιών αυτών.

Προτείνεται ένα υπερκείμενο δίκτυο το οποίο χρησιμοποιεί τη λογική της επικοινωνίας με βάση κάποιο προκαθορισμένο σημείο συνάντησης. Αντί ένα πακέτο δεδομένων να αποστέλεται απ'ευθείας από τον αποστολέα στον προορισμό, κάθε πακέτο αντιστοιχίζεται με ένα αναγνωριστικό. Αυτό το αναγνωριστικό χρησιμοποιείται στη συνέχεια από τον παραλήπτη για την ολοκλήρωση της παραλαβής του πακέτου [6]. Με αυτόν τον τρόπο διαχωρίζεται η διαδικασία της αποστολής από αυτήν της παραλαβής και δίνεται η δυνατότητα χρησιμοποίησης αρκετών εφαρμογών κάτι που είναι σημαντικό πλεονέκτημα σε σχέση με άλλα υπερκείμενα δίκτυα. Για τη σχεδίαση του υπερκείμενου δικτύου αυτού θα χρησιμοποιηθεί μια τροποποιημένη εκδοχή του υπερκείμενου δικτύου Pastry.

8.1 Επικοινωνία με βάση σημείο συνάντησης

Στη βασική τους μορφή τα πακέτα είναι ζεύγη ($id, data$), όπου το id συμβολίζει ένα αναγνωριστικό των m bit και το $data$ συμβολίζει το φορτίο που μεταφέρεται. Οι κόμβοι δέκτες χρησιμοποιούν σκανδάλεις για να ξεκινήσουν τη σύνδεση με τη μεταφορά πακέτων. Στη βασική τους μορφή οι σκανδάλεις είναι ζεύγη ($id, addr$), όπου το id συμβολίζει το αναγνωριστικό της σκανδάλης και το $addr$ συμβολίζει τη διεύθυνση του κόμβου. Η σκανδάλη ($id, addr$) υποδεικνύει ότι όλα τα πακέτα με αναγνωριστικό id πρέπει να προωθηθούν από το υπερκείμενο δίκτυο στον κόμβο με τη διεύθυνση $addr$.

Η διαδικασία είναι όπως εξηγείται στη συνέχεια. Ένας κόμβος δέκτης με διεύθυνση R που θέλει να λαμβάνει πακέτα που στέλνονται στο id , εισάγει στο δίκτυο μια σκανδάλη (id, R). Όταν ένα πακέτο στέλνεται στο αναγνωριστικό id , η σκανδάλη προκαλεί την προώθησή του στον κόμβο με διεύθυνση R .

Το αναγνωριστικό id αναπαριστά ένα σημείο συνάντησης μεταξύ των πακέτων του κόμβου αποστολέα και της σκανδάλης του κόμβου δέκτη. Με αυτό τον τρόπο διαχωρίζεται η διαδικασία της αποστολής από αυτήν της λήψης. Ο κόμβος αποστολέας δεν χρειάζεται να γνωρίζει τον αριθμό των κόμβων δεκτών, ούτε την τοποθεσία τους. Ομοίως ένας κόμβος δέκτης δεν χρειάζεται να γνωρίζει τον αριθμό ή την τοποθεσία των κόμβων αποστολέων.

Χρήσιμη είναι και η διαδικασία μη ακριβούς αντιστοίχισης μεταξύ αναγνωριστικών. Τα αναγνωριστικά έχουν μήκος m bit και υπάρχει ένα όριο ταυτοποίησης k , όπου $k < m$. Έτσι ένα αναγνωριστικό id_i μιας σκανδάλης ταιριάζει με ένα αναγνωριστικό id ενός πακέτου αν και μόνο αν το id_i είναι το μέγιστο κοινό πρόθεμα και είναι τουλάχιστον ίσο με το όριο ταυτοποίησης k . Η τιμή του k , επιλέγεται να είναι αρκετά μεγάλη ώστε η πιθανότητα δύο τυχαία αναγνωριστικά να ταιριάζουν να είναι αμελητέα.

Σε αυτό το υπερκείμενο δίκτυο σε κάθε κόμβο αποθηκεύονται σκανδάλεις (αναγνωριστικού – διεύθυνσης). Κάθε τερματικός κόμβος γνωρίζει περισσότερους από έναν κόμβους-εξυπηρετητές. Όταν ένας κόμβος θέλει να αποστείλει ένα πακέτο αναγνωριστικού-δεδομένων, το προωθεί σε έναν από τους κόμβους-εξυπηρετητές που γνωρίζει. Εάν αυτός ο κόμβος-εξυπηρετητής του δικτύου δεν έχει αποθηκευμένη τη σκανδάλη που αντιστοιχεί στο πακέτο αναγνωριστικού-δεδομένων, τότε το πακέτο προωθείται σε επόμενο κόμβο-εξυπηρετητή. Αυτή η διαδικασία συνεχίζεται έως ότου το πακέτο αυτό φτάσει σε έναν κόμβο-εξυπηρετητή που έχει αποθηκευμένη την κατάλληλη σκανδάλη.

8.1.1 Λειτουργία της παραλλαγής του υπερκείμενου δικτύου Pastry

Αυτή η διαδικασία εφαρμόζεται σε μια παραλλαγή του υπερκείμενου δικτύου Pastry που παρουσιάστηκε αναλυτικά προηγουμένως και μπορεί να προσφέρει αρκετά πλεονεκτήματα όπως την εκμετάλλευση της κοντινότητας της απόστασης μεταξύ των κόμβων. Όπως έχει αναφερθεί, σε κάθε κόμβο ανατίθεται ένα μοναδικό αναγνωριστικό των 128 bit. Το εύρος των αναγνωριστικών είναι από το 0 έως το $2^{128} - 1$ σε λογική κυκλικής διάταξης ώστε το 0 να έπεται του $2^{128} - 1$. Η ανάθεση των αναγνωριστικών

γίνεται τυχαία. Το υπερκείμενο δίκτυο Pastry δρομολογεί ένα μήνυμα στον κόμβο που είναι αριθμητικά πλησιέστερος με το κλειδί.

Κάθε κόμβος στο υπερκείμενο δίκτυο Pastry χειρίζεται έναν πίνακα δρομολόγησης, μια ομάδα γειτνίασης και μια ομάδα φύλλων. Κάθε κόμβος στο υπερκείμενο δίκτυο Pastry έχει τη δυνατότητα μέτρησης της απόστασής του από οποιονδήποτε άλλο κόμβο του ίδιου δικτύου με βάση την καθυστέρηση Round Trip Time [11]. Χρησιμοποιώντας τους κόμβους αυτούς που βρίσκονται σε κοντινότερη απόσταση με βάση την μέτρηση της καθυστέρησης RTT, και με βάση το κοινό πρόθεμα, συμπληρώνεται ο πίνακας δρομολόγησης ενός τυχαίου κόμβου έτσι ώστε να βρίσκονται σε αυτόν πάντα κοντινοί κόμβοι, ως προς την τοπολογία.

Έτσι κατά τη δρομολόγηση ενός μηνύματος, αυτό προωθείται σε κάθε βήμα σε έναν κοντινό κόμβο από αυτούς με τους οποίους έχει κοινό πρόθεμα, σύμφωνα με τον πίνακα δρομολόγησης. Αυτό είναι ένα χρήσιμο πλεονέκτημα για την γρήγορη εύρεση ενός αντικειμένου στο δίκτυο. Ακόμα, με τη χρήση αντιγράφων ενός αντικειμένου που αποθηκεύονται σε κόμβους, δίνεται η δυνατότητα να εντοπιστεί ο κόμβος που είναι πλησιέστερος με τον κόμβο που κάνει την αναζήτηση, χωρίς να είναι απαραίτητο να φτάσει στον κόμβο που περιέχει το αρχικό αντικείμενο αλλά βρίσκεται σε μεγαλύτερη απόσταση.

Η ομάδα γειτνίασης περιέχει τα αναγνωριστικά των κόμβων που είναι απόλυτα πλησιέστεροι στον αντίστοιχο κόμβο. Δεν χρησιμοποιείται για τη δρομολόγηση μηνυμάτων, και διαφέρει από τον πίνακα δρομολόγησης ως προς το μέγεθος και ως προς το γεγονός ότι ο πίνακας δρομολόγησης περιέχει κόμβους που είναι κοντινοί σύμφωνα με τη μέτρηση του RTT, αλλά μπορεί να μην περιέχονται κόμβοι που είναι οι απόλυτα πλησιέστεροι.

Η ομάδα φύλλων, περιέχει αναγνωριστικά κόμβων που είναι αριθμητικά πλησιέστεροι ως προς το αναγνωριστικό του τρέχοντος κόμβου, τόσο μεγαλύτερους όσο και μικρότερους.

Κατά τη δρομολόγηση ενός μηνύματος, όταν αυτό φτάνει σε έναν κόμβο ελέγχεται αν το αναγνωριστικό του ταυτίζεται με κάποιο από την ομάδα φύλλων του κόμβου. Αν συμβαίνει αυτό τότε προωθείται στον κόμβο προορισμού. Σε αντίθετη περίπτωση χρησιμοποιείται ο πίνακας δρομολόγησης και το μήνυμα προωθείται στον κόμβο που

έχει πλησιέστερο αναγνωριστικό και όπως χρησιμοποιήθηκε για το σχεδιασμό προηγουμένως είναι και σε κοντινή απόσταση με βάση τη μέτρηση του RTT.

Η σύγκλιση είναι ένα χαρακτηριστικό που υπάρχει σε αρκετά υπερκείμενα δίκτυα και υπάρχει και στην μορφή του Pastry που προτείνεται. Σε κάθε βήμα δρομολόγησης, ένα μήνυμα προωθείται σε έναν κόμβο που έχει μεγαλύτερο κοινό πρόθεμα με το κλειδί, ή αλλιώς σε έναν κόμβο που είναι αριθμητικά πλησιέστερος από τον προηγούμενο κόμβο.

Τα κλειδιά και τα αναγνωριστικά θεωρούνται ως αλληλουχία ψηφίων με βάση το 2^b . Σύμφωνα με τη διαδικασία δρομολόγησης, και την ιδιότητα ότι σε κάθε βήμα ένας κόμβος προωθεί ένα μήνυμα σε έναν κόμβο του οποίου το αναγνωριστικό έχει μεγαλύτερο κοινό πρόθεμα κατά 1 ψηφίο ή b bit, ο αριθμός των βημάτων δρομολόγησης είναι της τάξης του $O(\log_{2^b} N)$. Όταν ένα μήνυμα προωθείται με τη χρήση του πίνακα δρομολόγησης ο αριθμός των κόμβων που έχουν αναγνωριστικό με μεγαλύτερο πρόθεμα το οποίο να ταιριάζει με το κλειδί μειώνεται κατά ένα παράγοντα 2^b σε κάθε βήμα και έτσι προκύπτει ότι η δρομολόγηση προς τον τελικό προορισμό γίνεται με αριθμό βημάτων της τάξης του $O(\log_{2^b} N)$.

8.2 Δημόσιες και ιδιωτικές σκανδάλεις

Οι σκανδάλεις, στο επίπεδο εφαρμογής, διακρίνονται σε δημόσιες και ιδιωτικές. Οι δημόσιες σκανδάλεις μπορούν να υπολογιστούν από όλους τους τερματικούς κόμβους και χρησιμοποιούνται για την επίτευξη αρχικής σύνδεσης με τον επιθυμητό κόμβο. Οι ιδιωτικές σκανδάλεις επιλέγονται μυστικά από τους τερματικούς κόμβους και έχουν μικρή διάρκεια κατά τη χρησιμοποίησή τους.

Έστω ο εξυπηρετητής B που διατηρεί μια δημόσια σκανδάλη με αναγνωριστικό id_p και ο τερματικός κόμβος A που θέλει πρόσβαση στον εξυπηρετητή. Ο τερματικός κόμβος A εισάγει μια ιδιωτική σκανδάλη με αναγνωριστικό id_a στο υπερκείμενο δίκτυο και στέλνει το id_a στον εξυπηρετητή B μέσω της δημόσιας σκανδάλης με αναγνωριστικό id_p . Ο εξυπηρετητής B λαμβάνει το αναγνωριστικό id_a και εισάγει στο δίκτυο μια ιδιωτική σκανδάλη με αναγνωριστικό id_b . Στη συνέχεια στέλνει το id_b στον κόμβο A μέσω της ιδιωτικής σκανδάλης με αναγνωριστικό id_a και ο κόμβος A το λαμβάνει από το

υπερκείμενο δίκτυο. Τελικά τα πακέτα δεδομένων από τον κόμβο A στον εξυπηρετητή B διακινούνται μέσω της ιδιωτικής σκανδάλης id_b , και μέσω του id_a στην αντίθετη κατεύθυνση. Η σημαντική ιδιότητα που εμφανίζεται σε αυτό το μέρος είναι η δυνατότητα των τερματικών κόμβων να έχουν πλήρη έλεγχο στην επιλογή των ιδιωτικών τους σκανδαλών.

8.3 Στοιβά αναγνωριστικών

Ένα επιπλέον χαρακτηριστικό που εκμεταλλεύεται το προτεινόμενο υπερκείμενο δίκτυο είναι η αντικατάσταση των αναγνωριστικών με στοιβες αναγνωριστικών. Μια στοιβά αναγνωριστικών είναι μια λίστα αναγνωριστικών που παίρνει τη μορφή $(id_1, id_2, id_3, \dots, id_k)$, όπου id_i είναι ένα αναγνωριστικό ή μια διεύθυνση. Τα πακέτα p και οι σκανδάεις t , είναι της μορφής: $p = (id_s, data)$, $t = (id, id_s)$.

Η γενικευμένη μορφή πακέτων επιτρέπει σε έναν αποστολέα να στέλνει ένα πακέτο σε μια σειρά αναγνωριστικών. Η γενικευμένη μορφή σκανδαλών επιτρέπει σε μια σκανδάλη να προωθεί ένα πακέτο σε ένα άλλο αναγνωριστικό από ότι σε μια διεύθυνση. Αυτό επιτρέπει μεγαλύτερη ευελιξία.

Ένα πακέτο p προωθείται με βάση το πρώτο αναγνωριστικό id της στοιβας αναγνωριστικών έως ότου βρεθεί ο εξυπηρετητής που είναι υπεύθυνος για την αποθήκευση της κατάλληλης σκανδάλης για το p . Θεωρώντας ένα πακέτο με στοιβά αναγνωριστικών (id_1, id_2, id_3) , αν δεν υπάρχει σκανδάλη της οποίας το αναγνωριστικό να ταιριάζει με το id_1 , τότε το id_1 απορρίπτεται από τη στοιβά. Η διαδικασία επαναλαμβάνεται έως ότου ένα αναγνωριστικό στη στοιβά αναγνωριστικών του p βρει την κατάλληλη σκανδάλη t . Αν δε βρεθεί ένα τέτοια σκανδάλη τότε το πακέτο p απορρίπτεται. Σε αντίθετη περίπτωση, όταν βρεθεί μια σκανδάλη t το αναγνωριστικό της οποίας ταιριάζει με το id_1 , τότε το id_1 αντικαθίσταται από τη στοιβά αναγνωριστικών της t . Πιο συγκεκριμένα, αν η στοιβά αναγνωριστικών του t είναι (x, y) , τότε η στοιβά αναγνωριστικών του πακέτου γίνεται (x, y, id_2, id_3) . Αν το id_1 είναι διεύθυνση IP, τότε το πακέτο προωθείται σε αυτή τη διεύθυνση και το υπόλοιπο μέρος της στοιβας αναγνωριστικών του p προωθείται στην εφαρμογή.

Με την παραλαβή ενός πακέτου p , ο εξυπηρετητής ελέγχει αν είναι υπεύθυνος για την σκανδάλη του πακέτου. Αν δεν είναι, τότε προωθεί το μήνυμα στο υπερκείμενο δίκτυο. Αν είναι ο υπεύθυνος κόμβος, τότε επιστρέφει το σύνολο των αναγνωριστικών που ταιριάζουν με το p . Το πακέτο p στη συνέχεια προωθείται με βάση το πρώτο αναγνωριστικό στη στοίβα αναγνωριστικών του.

8.4 Πλεονεκτήματα προτεινόμενου υπερκείμενου δικτύου.

Με τη χρήση των παραπάνω χαρακτηριστικών μπορούν να επιτευχθούν σε αρκετές εφαρμογές. Η λειτουργία του συγκεκριμένου προτεινόμενου υπερκείμενου δικτύου έχει πλεονεκτήματα στη διαχείριση της κινητικότητας και της πολυεκπομπής, μεταξύ άλλων εφαρμογών, για την επίτευξη της επικοινωνίας μεταξύ των κόμβων σε σχέση με τα άλλα υπερκείμενα δίκτυα που έχουν παρουσιαστεί.

8.4.1 Κινητικότητα

Η μορφή της κινητικότητας που παρέχεται από το προτεινόμενο υπερκείμενο δίκτυο, είναι όταν σε έναν τερματικό κόμβο ανατίθεται μια νέα διεύθυνση μετακινούμενος από τοποθεσία σε τοποθεσία. Ένας κινητός τερματικός κόμβος που αλλάζει τη διεύθυνσή του από R σε R' που είναι το αποτέλεσμα της μετακίνησής του από ένα υποδίκτυο σε ένα άλλο, μπορεί να διατηρήσει τη συνδεσιμότητά του ανανεώνοντας τα ζεύγη ενεργοποίησής του από (id, R) σε (id, R') . Ο κόμβος αποστολέας δεν χρειάζεται να γνωρίζει την τρέχουσα θέση ή τη διεύθυνση του κινητού τερματικού κόμβου. Επιπλέον, με τη δρομολόγηση κάθε πακέτου με βάση το αναγνωριστικό του στον εξυπηρετητή που έχει αποθηκεύσει την σκανδάλη, δεν χρειάζεται κάποια άλλη διεργασία κατά την κίνηση του κόμβου αποστολέα. Έτσι μπορεί να διατηρείται η συνδεσιμότητα κατά την κίνηση τόσο του αποστολέα όσο και του παραλήπτη.

Η αποτελεσματικότητα της δρομολόγησης είναι σημαντικός παράγοντας στη διαχείριση της κινητικότητας. Η διεύθυνση του κόμβου εξυπηρετητή που είναι υπεύθυνος για την σκανδάλη, αποθηκεύεται στον αποστολέα και τα πακέτα προωθούνται απ'ευθείας στον εξυπηρετητή μέσω IP. Έτσι η δρομολόγηση μπορεί να γίνει μέσω ενός εξυπηρετητή στο υπερκείμενο δίκτυο. Επιπλέον, οι τερματικοί κόμβοι

μπορούν να επιλέξουν σκανδάλεις που βρίσκονται σε κοντινούς εξυπηρετητές με τη μέτρηση της απόστασης με βάση τη μέτρηση του χρόνου καθυστέρησης RTT.

8.4.2 Πολυεκπομπή(Multicast)

Η δημιουργία ομάδας πολυεκπομπής ισοδυναμεί με την καταχώρηση σκανδαλών με το ίδιο αναγνωριστικό από όλα τα μέλη της ομάδας. Αυτό έχει σαν αποτέλεσμα κάθε πακέτο με το συγκεκριμένο αναγνωριστικό να προωθείται σε όλα τα μέλη της ομάδας. Σε αντίθεση με την IP πολυεκπομπή, στο προτεινόμενο υπερκείμενο δίκτυο δεν υπάρχει διαφορά μεταξύ των πακέτων απλής εκπομπής ή πολυεκπομπής κατά την αποστολή ή τη λήψη. Μια εφαρμογή μπορεί να μεταβάλλεται από απλή εκπομπή σε πολυεκπομπή άμεσα, ανάλογα με τον αριθμό των κόμβων που διατηρούν σκανδάλεις με το ίδιο αναγνωριστικό. Με αυτό τον τρόπο μπορεί να υπάρξει συμμετοχή από άλλους κόμβους σε μια σύνδεση μεταξύ δύο αρχικών χρηστών. Με τη χρήση του IP, μια εφαρμογή πρέπει να αλλάζει τη διεύθυνση προορισμού για να γίνει μετάβαση από απλή εκπομπή σε πολυεκπομπή.

8.5 Αποτελεσματικότητα δρομολόγησης

Η αποτελεσματικότητα της δρομολόγησης είναι ένα από τα βασικά στοιχεία ενός δικτύου. Το προτεινόμενο υπερκείμενο δίκτυο δρομολογεί κάθε πακέτο στον κόμβο-εξυπηρετητή που διατηρεί την κατάλληλη σκανδάλη. Για τη βελτιστοποίηση της δρομολόγησης ο αποστολέας αποθηκεύει στη μνήμη τη διεύθυνση IP του κόμβου-εξυπηρετητή που έχει χρησιμοποιήσει για τη δρομολόγηση πακέτων. Έτσι όταν χρειαστεί η δρομολόγηση πακέτων με το ίδιο αναγνωριστικό ο αποστολέας γνωρίζει τον κατάλληλο εξυπηρετητή για την προώθηση του πακέτου στον τελικό προορισμό.

Αν η σκανδάλη μετακινηθεί από τον προηγούμενο κόμβο-εξυπηρετητή, τότε τα πακέτα θα προωθηθούν στον νέο κόμβο-εξυπηρετητή. Τότε στη μνήμη του αποστολέα θα αποθηκευτεί η νέα διεύθυνση του κόμβου-εξυπηρετητή.

Στο υπερκείμενο δίκτυο που προτείνεται, ένας τερματικός κόμβος γνωρίζει τα αναγνωριστικά των κοντινών του κόμβων-εξυπηρετητών με τη χρήση της μέτρησης του RTT που χρησιμοποιείται και για τη συμπλήρωση του πίνακα δρομολόγησης στο Pastry. Έτσι μπορεί να επιλέξει τα αναγνωριστικά των σκανδαλών να αντιστοιχίζονται σε

αυτούς τους κόμβους-εξυπηρετητές. Αυτό έχει το πλεονέκτημα πως όταν ο κόμβος αποστολέας και ο τερματικός κόμβος βρίσκονται σε κοντινή μεταξύ τους απόσταση, ο εξυπηρετητής που έχει αποθηκευμένη την σκανδάλη να μην είναι απομακρυσμένος και απαιτείται περισσότερος χρόνος για τη μεταφορά των πακέτων.

Κατά τη μετάβαση από ένα υποδίκτυο σε ένα άλλο από έναν κινητό τερματικό κόμβο, μπορεί ένας μεγάλος αριθμός πακέτων που αποστέλλονται να χάνονται. Αυτό συμβαίνει αν ο τερματικός κόμβος σταματά να δέχεται πακέτα στην παλιά διεύθυνση IP πριν ξεκινήσει να δέχεται πακέτα από τη νέα του διεύθυνση, τότε τα πακέτα αυτά θα χάνονται.

Αυτό το πρόβλημα μπορεί να επιλυθεί στο προτεινόμενο υπερκείμενο δίκτυο με τη χρήση των σκανδαλών που αντιστοιχίζονται σε κοντινούς εξυπηρετητές με βάση τη μέτρηση της καθυστέρησης RTT. Όσο μεγαλύτερη είναι η καθυστέρηση μεταξύ κόμβου και σκανδάλης τόσο μεγαλύτερος είναι και ο αριθμός των πακέτων που χάνονται. Έτσι, με την επιλογή των κοντινών εξυπηρετητών μειώνεται ο αριθμός των χαμένων πακέτων και βελτιώνεται η σύνδεση του τερματικού κόμβου με το δίκτυο.

8.5.1 Αποφυγή σημείων συμφόρησης

Όταν ένας μεγάλος αριθμός τερματικών κόμβων επιχειρεί να επικοινωνήσει με μια δημοφιλή σκανδάλη για την επίτευξη σύνδεσης, μπορεί να δημιουργηθεί πρόβλημα στον εξυπηρετητή που αποθηκεύεται η σκανδάλη λόγω υπερφόρτωσης. Μια λύση για ένα τέτοιο πρόβλημα είναι η προώθηση αντιγράφων των σκανδαλών από τον εξυπηρετητή σε κόμβους γειτονικούς. Με αυτό τον τρόπο διαμοιράζεται το φορτίο και αποφεύγεται η συμφόρηση σε έναν μόνο εξυπηρετητή.

8.5.2 Ανεκτικότητα σε σφάλματα

Κατά την αποχώρηση ή κατάρρευση κόμβων-εξυπηρετητών στο υπερκείμενο δίκτυο, η διαδικασία αποκατάστασης είναι όμοια με αυτή του υπερκείμενου δικτύου Pastry. Η κατάρρευση ενός κόμβου που υπάρχει καταχωρημένος στον πίνακα δρομολόγησης ανιχνεύεται όταν ο κόμβος προσπαθεί να επικοινωνήσει με τον κόμβο που έχει καταρρεύσει και δεν λαμβάνει απάντηση. Αυτό το γεγονός δεν καθυστερεί τη δρομολόγηση ενός μηνύματος, αφού το μήνυμα μπορεί να δρομολογηθεί από τους

υπόλοιπους κόμβους του πίνακα δρομολόγησης. Παρ'όλα αυτά το κενό που δημιουργείται στον πίνακα δρομολόγησης πρέπει να καλυφθεί από έναν άλλο κόμβο εξυπηρετητή που να καλύπτει τις προϋποθέσεις που έχουν οριστεί. Δηλαδή να είναι ένας κόμβος με κοινό πρόθεμα ο οποίος θα είναι και σε κοντινή απόσταση με βάση το συγκεκριμένο τρόπο δρομολόγησης.

Οι τερματικοί κόμβοι εκτελούν περιοδική ανανέωση των σκανδαλών, για να διατηρούνται ενεργοί στο υπερκείμενο δίκτυο. Έτσι ο τερματικός κόμβος διατηρεί τη σύνδεσιμότητα για όσο χρονικό διάστημα αυτός επιθυμεί, ενώ κατά την κατάρρευση ενός κόμβου-εξυπηρετητή η σκανδάλη δε χάνεται, αλλά επανατοποθετείται σε άλλον κόμβο με την περιοδική ανανέωση.

Ένα πιθανό πρόβλημα που μπορεί να προκύψει είναι ότι το χρονικό διάστημα από την κατάρρευση ενός κόμβου έως την επανατοποθέτηση της σκανδάλης σε έναν άλλο κόμβο-εξυπηρετητή, μπορεί να είναι αρκετά μεγάλο. Μία λύση είναι η εισαγωγή αντιγράφων της σκανδάλης στους κόμβους που υπάρχουν στον πίνακα δρομολόγησης, ώστε να εξασφαλίζεται ότι θα βρίσκονται σε κόμβους που θα διατηρούν το κόστος δρομολόγησης στον ίδιο βαθμό. Ένας άλλος τρόπος αντιμετώπισης του προβλήματος αυτού είναι η εισαγωγή εναλλακτικής σκανδάλης της μορφής (id_b, R) διαφορετικού από του αρχικού (id, R) , από τον τερματικό κόμβο R. Ο αποστολέας θα στέλνει τα πακέτα με τη στοίβα αναγνωριστικών (id, id_b) . Έτσι, αν ο κόμβος-εξυπηρετητής που διατηρεί την αρχική σκανδάλη καταρρεύσει, τότε το πακέτο θα προωθηθεί μέσω της εναλλακτικής σκανδάλης από άλλο κόμβο στην διεύθυνση του τερματικού κόμβου R.

8.6 Ασφάλεια

Σε αυτό το μέρος γίνονται προτάσεις για τα θέματα ασφαλείας που προκύπτουν με τη χρήση και εφαρμογή του προτεινόμενου υπερκειμένου δικτύου.

Με τη χρήση των σκανδαλών δίνεται η δυνατότητα σε έναν κακοπροαίρετο χρήστη που γνωρίζει την σκανδάλη ενός τερματικού κόμβου, να εισάγει μια σκανδάλη με το ίδιο αναγνωριστικό και τη δική του διεύθυνση να αποκτήσει όλες τις πληροφορίες που λαμβάνει και ο τερματικός κόμβος. Μια λύση είναι η χρήση των ιδιωτικών και των δημόσιων σκανδαλών.

Οι ιδιωτικές σκανδάλεις επιλέγονται κρυφά από τους τερματικούς κόμβους και δεν είναι γνωστά σε κανέναν άλλο χρήστη. Το μήκος του αναγνωριστικού της σκανδάλης δυσχεραίνει την επίθεση από τρίτους. Παρά το γεγονός ότι μπορεί σε κάποιο γειτονικό εξυπηρετητή να υπάρχει αποθηκευμένη μια σκανδάλη τέτοιο που να περιορίζει το εύρος των διαθέσιμων τιμών, το μήκος του αναγνωριστικού είναι αρκετά μεγάλο ώστε μια εφαρμογή μπορεί να διαθέτει αρκετά bit για τυχαία ανάθεση. Έτσι αν ένα αναγνωριστικό έχει μέγεθος 256 bit και μια εφαρμογή επιλέγει 128 bit για τυχαία ανάθεση, οι πιθανότητες για την σωστή εύρεση του αναγνωριστικού θα είναι $1/2^{127}$.

Ακόμα, οι τερματικοί κόμβοι έχουν τη δυνατότητα περιοδικής αλλαγής των ιδιωτικών σκανδαλών. Ένας άλλος τρόπος προστασίας είναι η ανάθεση από τον τερματικό κόμβο πολλαπλών σκανδαλών και ο κόμβος αποστολέας να στέλνει τα πακέτα τυχαία σε κάποιον από αυτή την σκανδάλη και στη συνέχεια αυτή προωθείται στον τερματικό κόμβο. Ένας κακοπροαίρετος χρήστης θα πρέπει να βρει όλες τις πιθανές σκανδάλεις.

Όταν χρησιμοποιούνται δημόσιες σκανδάλεις που είναι γνωστά σε όλους όσους συμμετέχουν, τότε μπορεί ο οποιοσδήποτε να παρακολουθήσει την κίνηση που εκτελείται προς κάποιον προορισμό. Μια λύση είναι οι τερματικοί κόμβοι να χρησιμοποιούν τις δημόσιες σκανδάλεις για να επιλέξουν δύο ιδιωτικές σκανδάλεις και να χρησιμοποιήσουν αυτά για τη μεταφορά των δεδομένων. Για την επίτευξη της επικοινωνίας, ο κόμβος A κρυπτογραφεί την ιδιωτική σκανδάλη id_a κάτω από το δημόσιο κλειδί ενός παραλήπτη B και την αποστέλει σε αυτόν. Ο κόμβος B αποκρυπτογραφεί την ιδιωτική σκανδάλη id_a , επιλέγει μια δική του ιδιωτική σκανδάλη id_b και το αποστέλει στον κόμβο A με το id_a . Έτσι ένας κακοπροαίρετος χρήστης δεν μπορεί να έχει πρόσβαση στις κρυπτογραφημένες σκανδάλεις.

Μια άλλη ενέργεια που μπορεί να κάνει ένας κακοπροαίρετος χρήστης είναι να απομονώσει έναν κόμβο αφαιρώντας το δημόσιο ζεύγος ενεργοποίησής του. Με τον ίδιο τρόπο σε καθεστώς πολυεκπομπής μπορεί να αφαιρεθούν οι σκανδάλεις και να απομονωθούν οι χρήστες. Η αφαίρεση μιας σκανδάλης προϋποθέτει και τον προσδιορισμό της διεύθυνσης IP.

Αυτό το πρόβλημα μπορεί να αντιμετωπιστεί με τον εξυπηρετητή S να χρησιμοποιεί δύο σκανδάλεις, (id_p, x) και (x, S) , όπου x είναι ένα αναγνωριστικό γνωστό μόνο στον εξυπηρετητή S και id_p το αναγνωριστικό της δημόσιας σκανδάλης. Έτσι ένας κακοπροαίρετος χρήστης θα πρέπει να γνωρίζει το x για την απομάκρυνση των σκανδαλών. Ο τερματικός κόμβος μπορεί να επιλέξει ένα αναγνωριστικό x τέτοιο ώστε τα (id_p, x) και (x, S) να αποθηκεύονται στον ίδιο εξυπηρετητή και να μην υπάρχει μεγάλη επιβάρυνση στη δρομολόγηση.

8.6.1 Ανωνυμία

Δίκτυα επικοινωνιών που προσφέρουν σύνδεση από σημείο σε σημείο δεν έχουν μεγάλο περιθώριο για ανωνυμία. Τα πακέτα περιέχουν τις διευθύνσεις της πηγής και του προορισμού, κάτι που δίνει τη δυνατότητα σε έναν κακοπροαίρετο χρήστη να μάθει περισσότερες πληροφορίες. Με το προτεινόμενο δίκτυο, η παρακολούθηση της κίνησης από τον αποστολέα δεν αποκαλύπτει την ταυτότητα του παραλήπτη και η παρακολούθηση της κίνησης στον παραλήπτη δεν αποκαλύπτει την ταυτότητα του αποστολέα λόγω του διαχωρισμού της πράξης της αποστολής και της λήψης. Η ανωνυμία μπορεί να βελτιωθεί με τη χρήση αλληλουχίας σκανδαλών ή στοιβών αναγνωριστικών για τη δρομολόγηση των πακέτων.

Η ιδιότητα του προτεινόμενου υπερκείμενου δικτύου για την επιλογή εξυπηρετητή σε κοντινή απόσταση με βάση τη μέτρηση της καθυστέρησης RTT για τη σύνδεση, βελτιώνει τη δρομολόγηση αλλά δίνει τη δυνατότητα να αποκαλυφθεί η τοποθεσία του τερματικού κόμβου. Για την αποφυγή αυτού του προβλήματος ο τερματικός κόμβος μπορεί να επιλέξει ένα αναγνωριστικό id τέτοιο ώστε η σκανδάλη να αποθηκεύεται σε έναν εξυπηρετητή κοντά στον κόμβο αποστολέα. Αυτό μπορεί να καθυστερήσει εν μέρει τη δρομολόγηση αλλά διασφαλίζει την απόκρυψη της τοποθεσίας του τερματικού κόμβου δέκτη. Έστω (id_1, A) η σκανδάλη που εισάγει ο κόμβος αποστολέας. Για την επίτευξη της γρήγορης μετάβασης, ο κόμβος δέκτης μπορεί να χρησιμοποιήσει δύο σκανδάλεις. Ένα της μορφής (id, id') κοντά στον κόμβο αποστολέα και ένα (id', B) κοντά στον κόμβο δέκτη. Παρ'όλο που εισάγονται δύο σκανδάλεις ο αποστολέας στέλνει πακέτα στη μορφή $(id, data)$ προς τον δέκτη. Επειδή ο κόμβος αποστολέας δεν χρειάζεται

να γνωρίζει το `id`, διασφαλίζεται η προστασία της τοποθεσίας του κόμβου δέκτη. Αν επιθυμείται η προστασία της τοποθεσίας και των δύο κόμβων, τότε θα επιλεγούν εξυπηρετητές τέτοιοι ώστε να διατηρείται όσο το δυνατόν γίνεται η ιδιότητα της κοντινής απόστασης με βάση τη μέτρηση της καθυστέρησης RTT, αλλά και να διασφαλίζεται όσο γίνεται η προστασία της τοποθεσίας των τερματικών κόμβων.

8.7 Εφαρμογές προτεινόμενου υπερκείμενου δικτύου

Σε αυτό το κομμάτι παρουσιάζονται μερικές εφαρμογές, όπως σύνθεση υπηρεσιών, ετερογενής πολυεκπομπή και πολυεκπομπή μεγάλης κλίμακας.

8.7.1 Σύνθεση υπηρεσιών

Μερικές εφαρμογές απαιτούν επεξεργασία των δεδομένων από τρίτους πριν την άφιξη στον προορισμό τους. Ένα παράδειγμα είναι μια πύλη πρωτοκόλου ασύρματων εφαρμογών (WAP) που μεταγλωττίζει σελίδες HTML σε μορφή κατάλληλη (WML) για ασύρματες συσκευές. Το WML είναι μια ελαφριά εφαρμογή του HTML σχεδιασμένο να λειτουργεί σε ασύρματες συσκευές με μικρές οθόνες και δυνατότητα εφαρμογών περιορισμένου εύρους. Σε μια τέτοια περίπτωση, ο εξυπηρετητής προωθεί τη σελίδα σε έναν τρίτο εξυπηρετητή T, ο οποίος εφαρμόζει τη μετατροπή HTML-WML και στη συνέχεια τα δεδομένα αποστέλονται στον προορισμό μέσω του WAP.

Σε αρκετές περιπτώσεις, τα δεδομένα χρειάζεται να υποστούν αρκετές μετατροπές από τρίτους εξυπηρετητές πριν φτάσουν στον τελικό προορισμό τους. Αυτή η διαδικασία μπορεί να εφαρμοστεί στο προτεινόμενο υπερκείμενο δίκτυο με τη χρήση της στοίβας των αναγνωριστικών. Για τη μετατροπή HTML-WML κάθε πακέτο δεδομένων που ξεκινά από τον αποστολέα έχει τη στοίβα αναγνωριστικών ($id_{HTML-WML}, id$). Έτσι, κάθε πακέτο δρομολογείται πρώτα στον εξυπηρετητή που είναι υπεύθυνος για τη μετατροπή HTML-WML. Στη συνέχεια το πακέτο δεδομένων ($id, data$) προωθείται στον τερματικό παραλήπτη.

8.7.2 Ετερογενής πολυεκπομπή

Η ετερογενής πολυεκπομπή δίνει τη δυνατότητα στον δέκτη να επιλέγει τη μορφή που θα παραλάβει τα δεδομένα ανάλογα με τις ανάγκες και τις δυνατότητες που έχει. Για παράδειγμα, για την αναπαραγωγή ενός τύπου MPEG βίντεο, υπάρχει ένας δέκτης της ίδιας μορφής MPEG και ένας δέκτης κωδικοποίησης άλλου τύπου έστω A . Το MPEG είναι ένα είδος συμπίεσης και αποσυμπίεσης των δεδομένων που χρησιμοποιείται για την αναπαραγωγή εικόνας βίντεο. Ο κάθε δέκτης έχει τη δυνατότητα να επιλέξει τη μετατροπή των πακέτων δεδομένων έτσι ώστε να μπορεί να τα αναπαράγει. Συγκεκριμένα ο δέκτης A , που έστω ότι ονομάζεται $R1$, εισάγει μια σκανδάλη $(id, (id_{MPEG-A}, R1))$, ενώ ο αποστολέας στέλνει πακέτα $(id, data)$. Για κάθε πακέτο που ταιριάζει με την σκανδάλη του $R1$, το αναγνωριστικό του αντικαθίσταται από τη στοιβία αναγνωριστικών της σκανδάλης (id_{MPEG-A}, T) . Στη συνέχεια το πακέτο προωθείται στον εξυπηρετητή που κάνει τη μετατροπή από MPEG σε μορφή A και τελικά προωθείται στον δέκτη $R1$. Αντίθετα ο δέκτης $R2$ που είναι δέκτης MPEG, εισάγει μόνο την σκανδάλη $(id, R1)$ στο υπερκείμενο δίκτυο. Με αυτόν τον τρόπο, δέκτες με διαφορετικούς τρόπους αναπαραγωγής δεδομένων μπορούν να χρησιμοποιήσουν την ίδια ομάδα πολυεκπομπής.

8.7.3 Πολυεκπομπή μεγάλης κλίμακας

Η πολυεκπομπή όπως παρουσιάστηκε προηγουμένως, λειτουργεί με όλα τα μέλη της ομάδας πολυεκπομπής να εισάγουν σκανδάλεις με το ίδιο αναγνωριστικό. Οι σκανδάλεις με το ίδιο αναγνωριστικό αποθηκεύονται στον ίδιο εξυπηρετητή και αυτός ο εξυπηρετητής είναι υπεύθυνος για την προώθηση των πακέτων σε κάθε μέλος της ομάδας πολυεκπομπής. Αυτό μπορεί να είναι μειονέκτημα για την επέκταση σε μεγάλες ομάδες πολυεκπομπής.

Ένας τρόπος για την αντιμετώπιση του προβλήματος αυτού είναι η δημιουργία μιας αλληλουχίας σκανδαλών, όπου κάθε μέλος R_i μιας ομάδας πολυεκπομπής με αναγνωριστικό id_g , αντικαθιστά την σκανδάλη με την αλυσίδα σκανδαλών με τα αναγνωριστικά $(id_g, x_1), (x_1, x_2), \dots, (x_i, R_i)$. Η αλλαγή αυτή είναι εμφανής στον

αποστολέα. Το αποτέλεσμα είναι το ίδιο, ένα πακέτο $(id_g, data)$ θα φτάσει στον κόμβο R_i μέσω της αλυσίδας των σκανδαλών. Η αλυσίδα σκανδαλών μπορεί να κατασκευαστεί και να διατηρηθεί από τα μέλη της ομάδας πολυεκπομπής, ή από κάποιον τρίτο πάροχο.

9. Σύγκριση προτεινόμενου υπερκείμενου δικτύου

Το δίκτυο που προτείνεται είναι ένα υπερκείμενο δίκτυο το οποίο χρησιμοποιεί τη λογική του προκαθορισμένου σημείου συνάντησης για την επικοινωνία μεταξύ των κόμβων μελών του δικτύου. Κάθε τερματικός κόμβος που επιθυμεί την σύνδεσή του στο δίκτυο για την επικοινωνία του με κάποιον κόμβο, εισάγει ένα ζεύγος αναγνωριστικού – διεύθυνσης σε έναν εξυπηρετητή. Έτσι τα πακέτα δεδομένων που φτάνουν στον εξυπηρετητή και έχουν το συγκεκριμένο αναγνωριστικό προωθούνται στον τερματικό κόμβο που έχει εισάγει το ζεύγος. Όπως φαίνεται, για την επίτευξη της σύνδεσης την αποκλειστική πρωτοβουλία την έχει ο τερματικός κόμβος που είναι αυτός που ξεκινά τη διαδικασία επικοινωνίας. Για την δρομολόγηση των πακέτων χρησιμοποιείται μία τροποποιημένη εκδοχή του υπερκείμενου δικτύου Pastry.

9.1 Ομοιότητες με Pastry

Το υπερκείμενο δίκτυο Pastry είναι ένα υπερκείμενο δίκτυο το οποίο χρησιμοποιεί τη λογική του κυκλικού χώρου αναγνωριστικών. Δεδομένου ενός κλειδιού, το Pastry κάνει τη δρομολόγηση ενός μηνύματος στον κόμβο του οποίου το αναγνωριστικό είναι αριθμητικά πλησιέστερο σε αυτό. Ακόμα κάθε κόμβος του υπερκείμενου δικτύου γνωρίζει τους γειτονικούς του κόμβους και ειδοποιείται για τυχόν αλλαγές όπως είσοδο ή αποχώρηση ενός κόμβου από το δίκτυο. Στο προτεινόμενο δίκτυο προτείνεται μια αλλαγή στο υπερκείμενο δίκτυο Pastry, τέτοια ώστε στον πίνακα δρομολόγησης κάθε κόμβου να βρίσκονται οι κόμβοι με την μικρότερη απόσταση με βάση τη μέτρηση της καθυστέρησης RTT από αυτούς που έχουν κοινό πρόθεμα. Έτσι μειώνεται ο χρόνος δρομολόγησης αφού επιλέγεται ο πλησιέστερος πάντα διαδοχικός κόμβος δρομολόγησης. Με τον ίδιο τρόπο επιλέγεται και η χρήση των αντιγράφων, έτσι ώστε σε περίπτωση κατάρρευσης ενός κόμβου να μην μεταβάλλεται σε μεγάλο βαθμό το κόστος δρομολόγησης.

Η λογική της δρομολόγησης με τη χρήση του σημείου συνάντησης έχει τα χαρακτηριστικά της αποδέσμευσης της διαδικασίας αποστολής από τη διαδικασία λήψης. Ο κόμβος αποστολέας δεν γνωρίζει τον κόμβο δέκτη και τερματικός κόμβος δέκτης δεν είναι απαραίτητο να γνωρίζει την ταυτότητα του αποστολέα.

Η διαχείριση της κινητικότητας είναι ένα χαρακτηριστικό πλεονέκτημα του προτεινόμενου υπερκείμενου δικτύου. Με το προτεινόμενο υπερκείμενο δίκτυο κατά την μετακίνηση του κόμβου δέκτη είναι δυνατή η διατήρηση της σύνδεσης που εξαρτάται μόνο από αυτόν. Ανανεώνοντας το ζεύγος ενεργοποίησης χρησιμοποιώντας τη νέα διεύθυνσή του μπορεί να παραμείνει συνδεδεμένος ακόμα και κατά την κίνησή του.

10.Περιβάλλον προσομοίωσης

Σε αυτό το κεφάλαιο παρουσιάζεται το περιβάλλον προσομοίωσης το οποίο θα χρησιμοποιηθεί για την εκτίμηση των υπερκείμενων δικτύων το οποίο είναι το OMNeT++ με τη χρήση του OverSim. Είναι λογισμικό ανοιχτού κώδικα ενώ είναι απλό και ευέλικτο στην εφαρμογή.

10.1 OMNeT++

Το Omnet++ [13],[15] είναι ένα περιβάλλον προσομοίωσης ανοιχτού κώδικα. Η κύρια λειτουργία του είναι η προσομοίωση δικτύων επικοινωνιών αλλά λόγω της ευέλικτης αρχιτεκτονικής του έχει χρησιμοποιηθεί με επιτυχία σε άλλες περιοχές όπως συστήματα αναμονής, αρχιτεκτονικές υλικού και επιχειρησιακές εφαρμογές.

Το OMNeT++ χρησιμοποιεί λογισμικές μονάδες που ορίζονται με μια γλώσσα ορισμού με την ονομασία NED(Network Description). Η γλώσσα NED διευκολύνει την περιγραφή ενός δικτύου με την περιγραφή των συνιστωσών του. Στην εφαρμογή τους είναι είτε σύνθεση λογισμικών μονάδων που προκύπτουν από το συνδυασμό πολλών λογισμικών μονάδων, είτε απλές λογισμικές μονάδες που εφαρμόζονται απ'ευθείας στην γλώσσα προγραμματισμού C++ .Αυτές οι λογισμικές μονάδες επικοινωνούν μεταξύ τους ανταλλάσσοντας μηνύματα μέσω συνδέσεων πυλών. Για τις συνδέσεις μεταξύ των λογισμικών μονάδων μπορούν να εισαχθούν παράμετροι, ανάλογα με τα δεδομένα του χρήστη.

Για την επίτευξη της εφαρμογής νέων πρωτοκόλλων το OMNeT++, περιλαμβάνει μια γεννήτρια μηνυμάτων που παράγει κώδικα σε C++, από τους ορισμούς των μηνυμάτων. Το OMNeT++ εφαρμόζει πολλά χαρακτηριστικά που απαιτούνται για την απεικόνιση. Με το ενσωματωμένο γραφικό περιβάλλον χρήστη μπορούν να απεικονιστούν η τοπολογία του δικτύου, οι κόμβοι και τα μηνύματα που ανταλλάσσονται κατά την προσομοίωση. Ακόμα είναι δυνατή η βαθύτερη μελέτη των περιεχομένων των μηνυμάτων και των μεταβλητών των κόμβων.

10.2 Πρόγραμμα προσομοίωσης OverSim

Το OverSim [14],[16] είναι ένα πρόγραμμα προσομοίωσης υπερκείμενων δικτύων το οποίο βασίζεται στο OMNeT++. Με το OverSim είναι δυνατή η προσομοίωση τόσο δομημένων υπερκείμενων δικτύων όπως τα Chord, Pastry όσο και μη δομημένων υπερκείμενων δικτύων όπως το GIA. Κατά την περίοδο συγγραφής της διπλωματικής το OverSim υλοποιείται μόνο σε περιβάλλον Linux. Επιλέχθηκε η έκδοση SuSE 11.0. Με το OverSim είναι δυνατόν να καλυφθούν οι παρακάτω ιδιότητες που απαιτούνται για ένα πρόγραμμα προσομοίωσης:

- **Προσομοίωση μεγάλων δικτύων:** Ένας προσομοιωτής πρέπει να έχει τη δυνατότητα να εκτελεί προσομοιώσεις με μεγάλο αριθμό κόμβων σε ένα λογικό χρονικό πλαίσιο.
- **Ευελιξία:** Ένας προσομοιωτής πρέπει να παρέχει τη δυνατότητα τόσο δομημένων, όσο και μη δομημένων υπερκείμενων δικτύων. Ο χρήστης θα έπρεπε να έχει τη δυνατότητα να προσδιορίζει όλες τις σχετικές παραμέτρους προσομοίωσης σε ένα ξεχωριστό αρχείο.
- **Μοντελοποίηση του υποκείμενου δικτύου:** Το μοντέλο του υποκείμενου δικτύου πρέπει να είναι μεταβλητό. Αυτό συμβαίνει γιατί πρέπει μια τοπολογία να προσαρμόζεται από τον χρήστη ανάλογα με την προσομοίωση που θέλει να εκτελέσει. Ακόμα πρέπει να παρέχονται απλά και γρήγορα εναλλακτικά μοντέλα για προσομοίωση με μεγάλο αριθμό κόμβων.
- **Επαναχρησιμοποίηση του κώδικα προσομοίωσης:** Οι εφαρμογές των υπερκείμενων δικτύων πρέπει να είναι δυνατόν να επαναχρησιμοποιηθούν για εφαρμογές πραγματικών δικτύων δίνοντας τη δυνατότητα επιβεβαίωσης των αποτελεσμάτων της προσομοίωσης. Έτσι το πρόγραμμα προσομοίωσης πρέπει να έχει τη δυνατότητα να χειρίζεται πραγματικά πακέτα δικτύου και να επικοινωνεί με άλλες εφαρμογές του ίδιου υπερκείμενου πρωτοκόλου.
- **Στατιστική:** Ο προσομοιωτής πρέπει να έχει τη δυνατότητα να συλλέξει στατιστικά δεδομένα όπως η κίνηση του δικτύου από και προς έναν κόμβο και η επιτυχημένη ή αποτυχημένη παράδοση πακέτων.
- **Καταγραφή αρχείων:** Για την χρήση και την επέκταση του προσομοιωτή με νέα πρωτόκολλα υπερκείμενων δικτύων πρέπει να υπάρχει ένα εκτεταμένο εγχειρίδιο

χρήστη. Ακόμα ο πηγαίος κώδικας αλλά και το περιβάλλον εργασίας πρέπει να είναι διαθέσιμα και καταγεγραμμένα.

- **Διαδραστική απεικόνιση:** Για την επιβεβαίωση και βελτίωση νέων ή ήδη υπάρχοντων υπερκείμενων δικτύων πρέπει να υπάρχει ένα γραφικό περιβάλλον που να απεικονίζει την τοπολογία του δικτύου και να μπορεί να προσαρμόζεται ξεχωριστά σε κάθε περίπτωση.

Σε έναν προσομοιωτή υπερκείμενων δικτύων πρέπει να υποστηρίζονται αρκετά μοντέλα υποκείμενων δικτύων. Στο OverSim παρέχονται τρία υποκείμενα δίκτυα, τα οποία είναι τα Simple, SingleHost και INET.

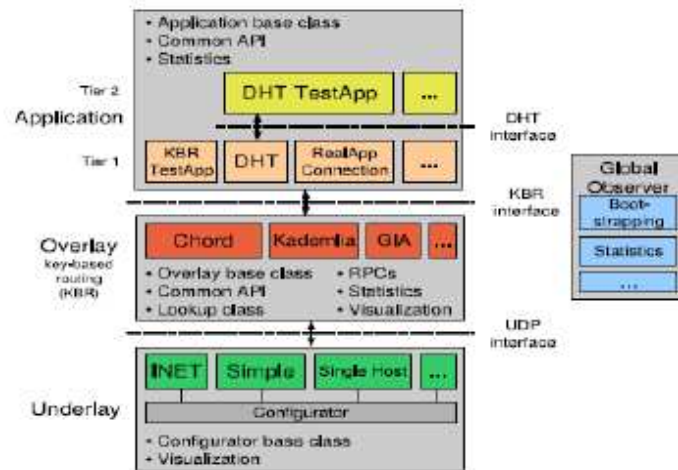
Το πιο κλιμακούμενο μοντέλο είναι η υποκείμενη τοπολογία Simple. Σε αυτό το μοντέλο τα πακέτα δεδομένων στέλνονται απ'ευθείας από έναν κόμβο του υπερκείμενου δικτύου σε έναν άλλο με τη χρήση ενός γενικού πίνακα δρομολόγησης. Τα πακέτα μεταξύ των κόμβων του υπερκείμενου δικτύου καθυστερούνται είτε από μια σταθερή χρονική περίοδο, είτε από μια καθυστέρηση που υπολογίζεται από την απόσταση μεταξύ των κόμβων. Κάθε κόμβος τοποθετείται στον Ευκλείδιο χώρο και σε κάθε κόμβο ανατίθενται χαρακτηριστικά όπως εύρος ζώνης και καθυστέρηση πρόσβασης. Λόγω της μικρής επιβάρυνσης της προσομοίωσης με αυτό το μοντέλο επιτυγχάνεται μεγάλη ακρίβεια και η ικανότητα να προσομοιώνονται δίκτυα με μεγάλο αριθμό κόμβων.

Με το μοντέλο SingleHost κάθε χρονική στιγμή προσομοιώνεται ένα μοναδικό δίκτυο το οποίο μπορεί να συνδεθεί με άλλα δίκτυα όπως το Internet. Με την εκτέλεση της σύνδεσης μειώνεται η ταχύτητα προσομοίωσης και απαιτείται περισσότερος χρόνος για την εκτέλεσή της.

Το μοντέλο υποκείμενου δικτύου INET προέρχεται από το πλαίσιο INET του OMNeT++, το οποίο περιλαμβάνει μοντέλα προσομοίωσης όλων των στρωμάτων δικτύου από το στρώμα MAC και έπειτα. Είναι χρήσιμο για προσομοιώσεις ολοκληρωμένων δικτύων αν αυτό χρειάζεται. Το πλαίσιο INET δεν είναι το βέλτιστο για δίκτυα μεγάλης κλίμακας. Έτσι υπάρχουν προσθήκες της δρομολόγησης με βάση πίνακες ανάμειξης για γρηγορότερη προώθηση των πακέτων μηνυμάτων.

Με αυτό το περίπλοκο μοντέλο μπορούν να αναλυθούν τα πλεονεκτήματα της τοποθέτησης υπερκείμενων συσκευών στο δίκτυο πρόσβασης που μπορεί να οδηγήσει

στην βελτιστοποίηση της υπερκείμενης τοπολογίας και στην μείωση της καθυστέρησης των πακέτων. Ακόμα το μοντέλο αυτό έχει επεκταθεί με παρόμοιες τεχνικές με το μοντέλο SingleHost. Με την προσθήκη μιας λογισμικής μονάδας δρομολόγησης στην προσομοίωση, αυτή μπορεί να χρησιμοποιηθεί σαν πύλη εξόδου για ένα πραγματικό δίκτυο.



Εικόνα 4 : Αρχιτεκτονική του OverSim

Το πρόγραμμα OverSim υποστηρίζει τόσο δομημένα όσο και μη δομημένα υπερκείμενα δίκτυα. Το σημαντικότερο πλεονέκτημα όμως είναι ότι επιτρέπει την εφαρμογή νέων υπερκείμενων πρωτοκόλων χρησιμοποιώντας μερικές κοινές λειτουργίες των υπερκείμενων δικτύων που έχουν ήδη προταθεί. Αυτές οι κοινές λειτουργίες είναι οι παρακάτω:

- Διαχείριση μηνυμάτων υπερκείμενου δικτύου (RPC και στατιστικά δεδομένα)
- Γενική συνάρτηση αναζήτησης
- Υποστήριξη για την απεικόνιση της υπερκείμενης τοπολογίας
- Υποστήριξη διαδικασίας εκκίνησης

Η διαχείριση των μηνυμάτων υπερκείμενου δικτύου παρέχει τη δυνατότητα διαχείρισης του χρόνου λήξης και επανεκπομπής των πακέτων, λόγω της απώλειας των πακέτων. Ακόμα συλλέγει στατιστικά δεδομένα που σχετίζονται με τα πακέτα όπως τον

αριθμό των πακέτων που στέλνονται, λαμβάνονται, προωθούνται, ή απορρίπτονται σε κάθε κόμβο.

Η συνάρτηση αναζήτησης διαχωρίζει την κοινή λειτουργικότητα των μηχανισμών αναζήτησης σε δομημένα υπερκείμενα δίκτυα και παρέχει έναν αναδρομικό και επαναληπτικό τρόπο αναζήτησης. Έτσι για κάθε πρωτόκολο υπερκείμενου δικτύου, εφαρμόζεται μόνο μια μέθοδος αναζήτησης του τοπικού πίνακα δρομολόγησης, η οποία επιστρέφει τον πλησιέστερο κόμβο στην υπερκείμενη τοπολογία. Επίσης η συνάρτηση αναζήτησης περιέχει βασική υποστήριξη για την προσομοίωση συμπεριφοράς ενός εχθρικού κόμβου.

Το γραφικό περιβάλλον χρήστη του OMNeT++ υποστηρίζει τον έλεγχο των νέων πρωτοκόλων υπερκείμενων δικτύων με την απεικόνιση των μεταφερόμενων μηνυμάτων με λεπτομέρεια. Δίνεται ακόμα η δυνατότητα παρακολούθησης συγκεκριμένων πινάκων δρομολόγησης και άλλων διαδικασιών κατά τη διάρκεια της προσομοίωσης. Στο OverSim δίνεται η δυνατότητα σχεδιασμού με βέλη μεταξύ των κόμβων για την απεικόνιση της υπερκείμενης τοπολογίας σε σχέση με την υποκείμενη τοπολογία που σχεδιάζεται από το OMNeT++.

Μια γενική λογισμική μονάδα που ονομάζεται Global Observer έχει μια γενική εικόνα του υπερκείμενου δικτύου σε κάθε δρομολόγηση. Μπορεί να υλοποιήσει αρκετές λειτουργίες που καθορίζονται από τον χρήστη, αλλά προς το παρόν χρησιμοποιείται για την παροχή της διεύθυνσης ενός τυχαίου κόμβου που υπάρχει ήδη στο δίκτυο σε κόμβους που χρειάζονται αυτή τη διεύθυνση για την είσοδό τους στο υπερκείμενο δίκτυο.

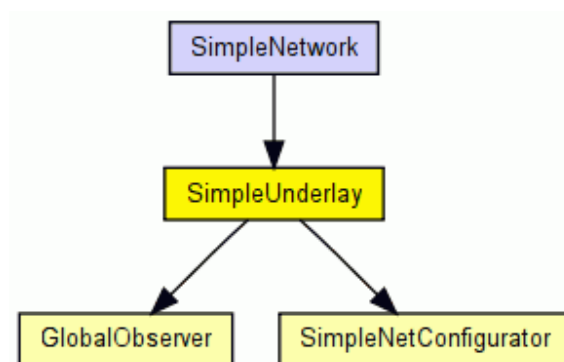
Το OverSim χρησιμοποιείται για την προσομοίωση του προτεινόμενου υπερκείμενου δικτύου της εργασίας γιατί έχει τη δυνατότητα να το υλοποιήσει αποτελεσματικά με τα χαρακτηριστικά του και να εξαχθούν συμπεράσματα για τις ιδιότητές του κατά την εφαρμογή.

11. Προσομοίωση προτεινόμενου υπερκείμενου δικτύου με το OverSim

Σε αυτό το μέρος της διπλωματικής θα γίνει περιγραφή της προσομοίωσης του προτεινόμενου υπερκείμενου δικτύου και παρουσίαση των αποτελεσμάτων που προκύπτουν. Ακόμα θα γίνει προσομοίωση της λειτουργίας με τη χρήση του Chord, ώστε να εξαχθούν συμπεράσματα για τα πλεονεκτήματα του προτεινόμενου υπερκείμενου δικτύου.

11.1 Υποκείμενη τοπολογία SimpleNetwork

Η εκτέλεση της προσομοίωσης γίνεται με βάση την υποκείμενη τοπολογία SimpleNetwork. Το κύριο πλεονέκτημά της είναι η μικρή επιβάρυνση κατά την εκτέλεση της προσομοίωσης που δίνει τη δυνατότητα χρησιμοποίησης μεγάλου αριθμού κόμβων με μεγάλη ακρίβεια στα αποτελέσματα. Η σχέση που έχει η τοπολογία SimpleNetwork στο πρόγραμμα OverSim, με τις υπόλοιπες λογισμικές μονάδες απεικονίζεται στο επόμενο διάγραμμα.



Εικόνα 5: Σχέση SimpleNetwork-λογισμικών μονάδων

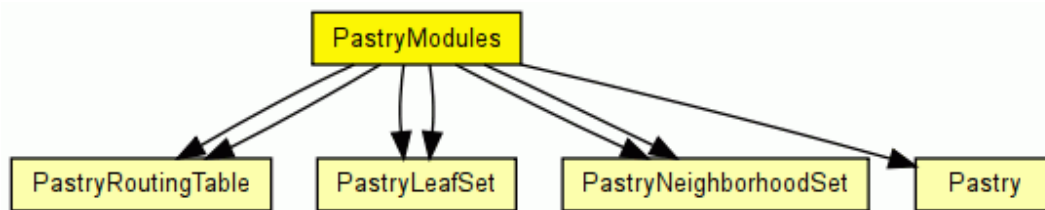
Η υποκείμενη τοπολογία SimpleNetwork υλοποιείται με την χρήση των λογισμικών μονάδων GlobalObserver και SimpleNetConfigurator. Η λογισμική μονάδα GlobalObserver χρησιμοποιεί τις BootstrapOracle, GlobalFunctions, GlobalParameters, GlobalStatistics, GlobalTraceManager. Η BootstrapOracle είναι υπεύθυνη για την διαδικασία εκκίνησης και την διανομή των κλειδιών στους κόμβους, η GlobalParameters είναι υπεύθυνη για την αποθήκευση παραμέτρων των γενικών προσομοιώσεων και η

GlobalStatistics είναι υπεύθυνη για την καταγραφή των γενικών στατιστικών των προσομοιώσεων.

Με την SimpleNetConfigurator διαμορφώνεται η τοπολογία SimpleNetwork, προστίθενται περιοδικά υπερκείμενοι κόμβοι κατά την φάση εκκίνησης. Στη συνέχεια προστίθενται, είτε διαγράφονται, είτε μετακινούνται υπερκείμενοι κόμβοι σύμφωνα με πιθανότητες που καθορίζονται από τις παραμέτρους creationProbability, removalProbability, migrationProbability.

11.2 Λογισμικές Μονάδες

Για την εφαρμογή του προτεινόμενου υπερκείμενου δικτύου θα χρησιμοποιηθεί το λογισμικό Pastry όπως αυτό εμφανίζεται στο OverSim. Η σχέση μεταξύ των λογισμικών μονάδων που χρησιμοποιούνται για την υλοποίηση του πρωτοκόλου Pastry απεικονίζεται στο επόμενο διάγραμμα.



Εικόνα 6: Σχέση λογισμικών μονάδων για την υλοποίηση του Pastry

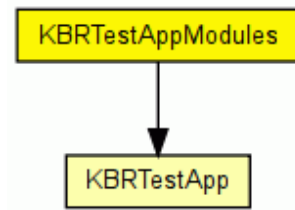
Όπως φαίνεται, περιέχει τα απαραίτητα στοιχεία για την υλοποίηση του Pastry όπως έχει περιγραφεί στην εργασία αυτή. Το PastryRoutingTable υλοποιεί τον πίνακα δρομολόγησης για το πρωτόκολο Pastry. Το PastryLeafSet υλοποιεί την ομάδα φύλλων του πρωτοκόλου Pastry, ενώ το PastryNeighborhoodSet υλοποιεί την ομάδα γειτνίασης. Το Pastry είναι το βασικό λογισμικό για την εφαρμογή του υπερκείμενου πρωτοκόλου Pastry.

Ακόμα, το λογισμικό που υλοποιεί τη λειτουργία του προτεινόμενου υπερκείμενου δικτύου με τη χρήση των σκανδαλών, δηλαδή την εισαγωγή τους και την ανανέωσή τους είναι το I3Hostmobility του OverSim.

Για την μελέτη και επεξεργασία των αποτελεσμάτων των προσομοιώσεων έχει χρησιμοποιηθεί το λογισμικό KBRTestApp. Το KBR είναι συντομογραφία του όρου

Key Based Routing. Με το KBRTestApp μπορούν να γίνουν μετρήσεις της προσομοίωσης ενός υπερκείμενου πρωτοκόλου ως προς την καθυστέρηση(Latency), τον αριθμό βημάτων που απαιτούνται για την αποστολή ενός πακέτου(Hop Count) και του λόγου της επιτυχημένης αποστολής μηνυμάτων(Delivery Ratio).

Η σχέση του λογισμικού που απαιτείται για την υλοποίηση του KBRTestApp απεικονίζεται στο επόμενο διάγραμμα:



Εικόνα 7: Σχέση λογισμικών μονάδων για υλοποίηση KBRTestApp

Για την λειτουργία της πολυεκπομπής(Multicast) στο προτεινόμενο υπερκείμενο δίκτυο με τη χρήση των σκανδαλών, χρησιμοποιείται το λογισμικό του OverSim I3Multicast.

12. Προσομοίωση

Με βάση τις παραπάνω λογισμικές μονάδες, δημιουργήθηκαν οι κώδικες εντολών για την προσομοίωση του προτεινόμενου υπερκείμενου δικτύου σε διάφορες περιπτώσεις. Στην αρχή για μεγάλο αριθμό τερματικών κόμβων, στη συνέχεια για περιπτώσεις τυχαίας αποχώρησης και εισαγωγής νέων κόμβων (random churn rate), αποχώρησης και εισαγωγής νέων κόμβων με σταθερό ρυθμό (lifetimechurn), λειτουργίας σε καθεστώς πολυεκπομπής καθώς και για μεταβολή του αριθμού των δρομολογητών και του ρυθμού δεδομένων. Για όλες τις προσομοιώσεις ο κοινός χρόνος προσομοίωσης είναι διάρκειας 2 ωρών και συνθήκες σταθερές.

12.1 Εντολές προσομοίωσης

. Οι εντολές που χρησιμοποιήθηκαν για την προσομοίωση της λειτουργίας του προτεινόμενου υπερκείμενου δικτύου σε μεγάλη κλίμακα τερματικών κόμβων είναι οι εξής:

```
description = "I3 over Pastry (SimpleNetwork, no churn)"
network = SimpleNetwork
**.churnGeneratorTypes = "NoChurn NoChurn"
**[0].overlayType = "PastryModules"
**[0].tier1Type = "KBRTestAppModules"
**[0].targetOverlayTerminalNum = 100
**[1].overlayType = "OverlayDummyModules"
**[1].tier1Type = "I3HostMobility"
**[1].targetOverlayTerminalNum = 1000
**.neighborCache.enableNeighborCache = true
```

Στην επιλογή για την παραγωγή των churnGeneratorTypes χρησιμοποιείται η επιλογή NoChurn, γιατί σε αυτή την προσομοίωση δεν ενδιαφέρει η αποχώρηση και εισαγωγή νέων κόμβων αλλά η ιδιότητα του προτεινόμενου υπερκείμενου δικτύου να υποστηρίζει μεγάλο αριθμό κόμβων. Χρησιμοποιείται το πρωτόκολλο Pastry και επιλέγεται ο αριθμός των κόμβων δρομολογητών να είναι 100, ενώ ο αριθμός των τερματικών κόμβων να φτάνει από τους 200 έως τους 1000 κόμβους, για 5 διαφορετικές προσομοιώσεις. Η εντολή `**[0].tier1Type = "KBRTestAppModules"` χρησιμοποιείται για τη διεξαγωγή των μετρήσεων για την καθυστέρηση, τον αριθμό των βημάτων για την αποστολή ενός μηνύματος και του λόγου επιτυχημένης αποστολής μηνυμάτων. Στον παραπάνω κώδικα υλοποιείται η περίπτωση των 1000 τερματικών κόμβων.

Οι εντολές που χρησιμοποιήθηκαν για την προσομοίωση της λειτουργίας του προτεινόμενου υπερκείμενου δικτύου με τυχαία αναχώρηση και εισαγωγή νέων κόμβων είναι οι εξής:

```
description = "I3 over Pastry (SimpleNetwork, RandomChurn)"
network = SimpleNetwork
**.churnGeneratorTypes = "NoChurn RandomChurn"
**[0].overlayType = "PastryModules"
**[0].tier1Type = "KBRTTestAppModules"
**[0].targetOverlayTerminalNum = 50
**[1].overlayType = "OverlayDummyModules"
**[1].tier1Type = "I3HostMobility"
**[1].targetOverlayTerminalNum = 400
**.neighborCache.enableNeighborCache = true
```

Στην επιλογή για την παραγωγή των `churnGeneratorTypes` χρησιμοποιείται η επιλογή `RandomChurn`. Κατά τη λειτουργία του `RandomChurn`, μόλις περάσει ένα καθορισμένο χρονικό διάστημα, επιλέγεται ένας τυχαίος αριθμός. Με βάση αυτό τον αριθμό ένας κόμβος είτε προστίθεται, είτε αφαιρείται, είτε μετακινείται. Χρησιμοποιείται η εντολή `**[0].tier1Type = "KBRTTestAppModules"` για τη διεξαγωγή των μετρήσεων για την καθυστέρηση, τον αριθμό των βημάτων για την αποστολή ενός μηνύματος και του λόγου επιτυχημένης αποστολής μηνυμάτων. Επιλέγεται ο αριθμός των κόμβων δρομολογητών να είναι 50, ενώ ο αριθμός των τερματικών κόμβων να φτάνει από τους 100 έως τους 400 κόμβους, για 3 διαφορετικές προσομοιώσεις. Στον παραπάνω κώδικα υλοποιείται η περίπτωση των 400 τερματικών κόμβων.

Οι εντολές που χρησιμοποιήθηκαν για την προσομοίωση της λειτουργίας του προτεινόμενου δικτύου για το καθεστώς πολυεκπομπής είναι οι εξής:

```
description = "I3 Multicast over Pastry (SimpleNetwork, no churn)"
network = SimpleNetwork
**.churnGeneratorTypes = "NoChurn NoChurn"
**[0].overlayType = "PastryModules"
**[0].tier1Type = "KBRTTestAppModules"
**[0].targetOverlayTerminalNum = 50
**[1].overlayType = "OverlayDummyModules"
**[1].tier1Type = "I3Multicast"
**[1].targetOverlayTerminalNum = 400
**.neighborCache.enableNeighborCache = true
```

Στην επιλογή για την παραγωγή των churnGeneratorTypes χρησιμοποιείται η επιλογή NoChurn, γιατί για αυτή την προσομοίωση δεν ενδιαφέρει η αποχώρηση και εισαγωγή νέων κόμβων αλλά η λειτουργία του προτεινόμενου υπερκείμενου δικτύου κατά την πολυεκπομπή. Χρησιμοποιείται και πάλι η εντολή `**[0].tier1Type = "KBRTestAppModules"` για τη διεξαγωγή των μετρήσεων για την καθυστέρηση, τον αριθμό των βημάτων για την αποστολή ενός μηνύματος και του λόγου επιτυχημένης αποστολής μηνυμάτων. Επιλέγεται ο αριθμός των κόμβων δρομολογητών να είναι 50, ενώ ο αριθμός των τερματικών κόμβων να φτάνει από τους 200 έως τους 400 κόμβους, για 3 διαφορετικές προσομοιώσεις. Στον παραπάνω κώδικα υλοποιείται η περίπτωση των 400 τερματικών κόμβων.

Για την προσομοίωση του υπερκείμενου δικτύου με μεταβολή των δρομολογητών, ο κώδικας προσομοίωσης είναι ο ίδιος με αυτόν για τη λειτουργία με μεγάλο αριθμό τερματικών κόμβων. Η διαφορά είναι ότι επιλέχθηκε σταθερός αριθμός 500 τερματικών κόμβων με την εντολή `**[1].targetOverlayTerminalNum = 500`, και μεταβολή των δρομολογητών με 100, 300 και 500 από την εντολή `**[0].targetOverlayTerminalNum = 100` και αντίστοιχα για τους 300 και 500 κόμβους.

Για όλες τις προσομοιώσεις που εκτελούνται, επιλέγεται το SimpleNetwork γιατί επιφέρει μικρή επιβάρυνση κατά την εκτέλεση της προσομοίωσης και δίνει τη δυνατότητα χρησιμοποίησης μεγάλου αριθμού κόμβων με μεγάλη ακρίβεια στα αποτελέσματα.

12.2 Αποτελέσματα προσομοίωσης

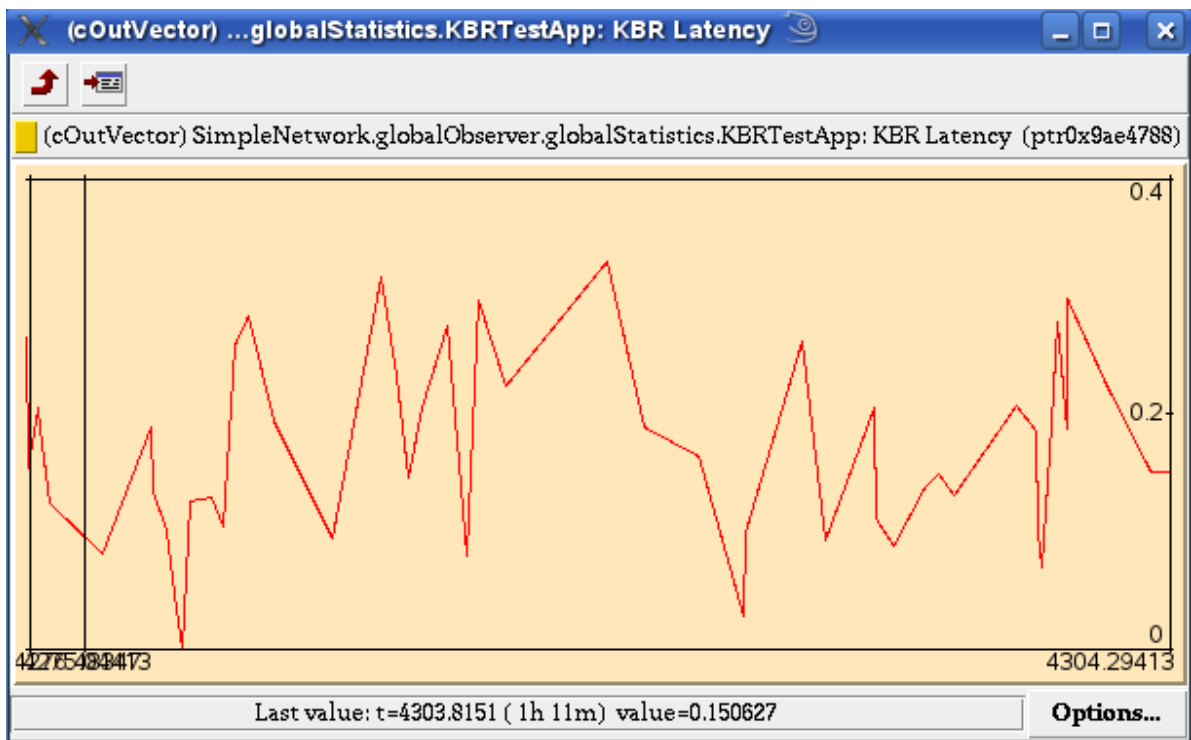
Σε αυτό το μέρος θα παρουσιαστούν τα αποτελέσματα των προσομοιώσεων με το πρόγραμμα OverSim.

12.2.1 Προσομοίωση για μεγάλο αριθμό κόμβων

Για την προσομοίωση της λειτουργίας με μεγάλο αριθμό κόμβων έγιναν 5 προσομοιώσεις με σταθερό αριθμό 100 δρομολογητών και αριθμό τερματικών κόμβων 200, 400, 600, 800, 1000. Για την προσομοίωση της λειτουργίας του δικτύου με 100

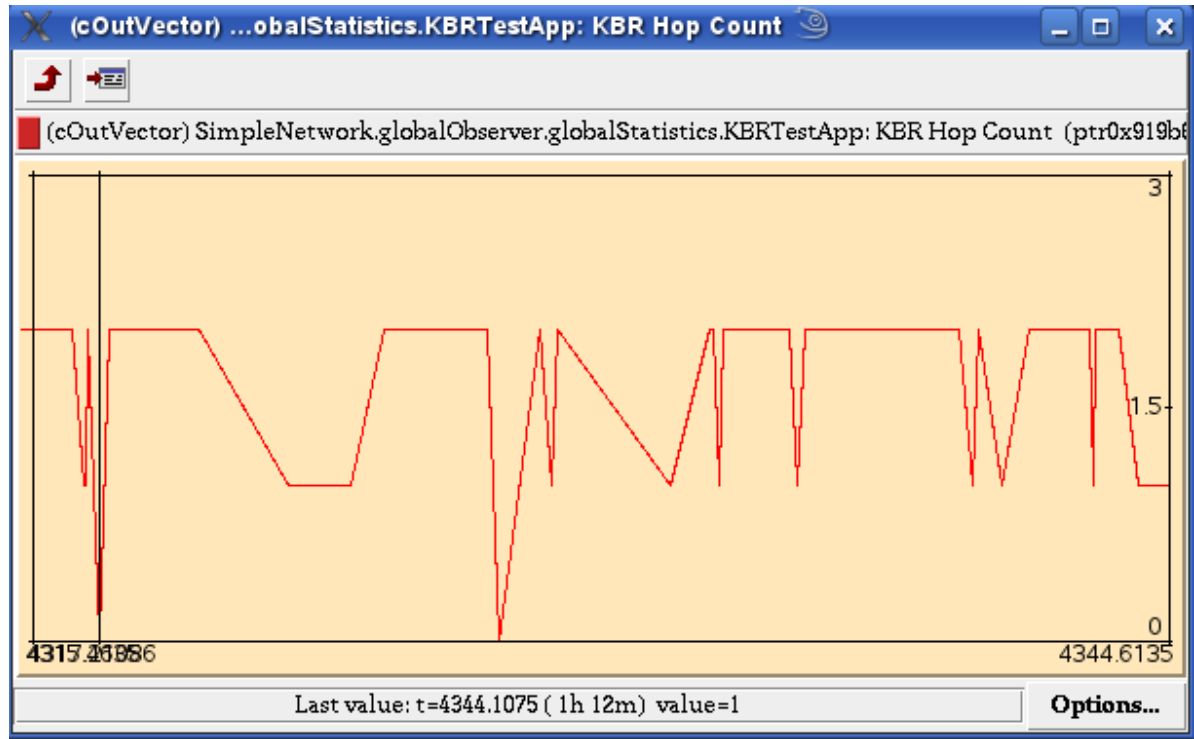
δρομολογητές και 200 τερματικούς κόμβους ένα ενδεικτικό γράφημα για την καθυστέρηση σε ένα χρονικό διάστημα της προσομοίωσης όπως αυτό παρουσιάζεται με το πρόγραμμα OverSim είναι το παρακάτω.

Κατά την προσομοίωση αυτή, κάθε κόμβος που εισέρχεται στο υπερκείμενο δίκτυο εισάγει μια σκανδάλη στον κόμβο δρομολογητή ο οποίος αντιστοιχεί στο κατάλληλο αναγνωριστικό με αυτό του τερματικού κόμβου. Στη συνέχεια, το πακέτο που αποστέλεται, δρομολογείται με βάση την υπερκείμενη λογική του Pastry μέχρι τον κόμβο δρομολογητή που αποθηκεύει τη σκανδάλη με το κατάλληλο αναγνωριστικό. Επόμενο βήμα είναι η αποστολή του πακέτου στον κόμβο η διεύθυνση του οποίου βρίσκεται στη σκανδάλη.



Εικόνα 8 : Καθυστέρηση ως προς το χρόνο προσομοίωσης για μεγάλο αριθμό κόμβων

Στη συνέχεια εμφανίζεται ένα ενδεικτικό γράφημα για τον αριθμό βημάτων που απαιτούνται για την αποστολή ενός μηνύματος όπως παρουσιάζεται στο OverSim.



Εικόνα 9: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης για μεγάλο αριθμό κόμβων

Τα δεδομένα για την προσομοίωση του υπερκείμενου δικτύου με τη μεταβολή των τερματικών κόμβων από 200 έως 1000, ως προς την καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
200	0.199107
400	0.199416
600	0.199022
800	0.199257
1000	0.199175

Τα δεδομένα για την προσομοίωση του υπερκείμενου δικτύου με τη μεταβολή των τερματικών κόμβων από 200 έως 1000, ως προς τον αριθμό βημάτων είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
200	1.688499
400	1.689208
600	1.689864
800	1.692646
1000	1.687216

Ακόμα, τα δεδομένα για το Delivery Ratio είναι τα παρακάτω.

Τερματικοί κόμβοι	Delivery Ratio
200	1
400	0.999914
600	1
800	0.999812
1000	0.999904

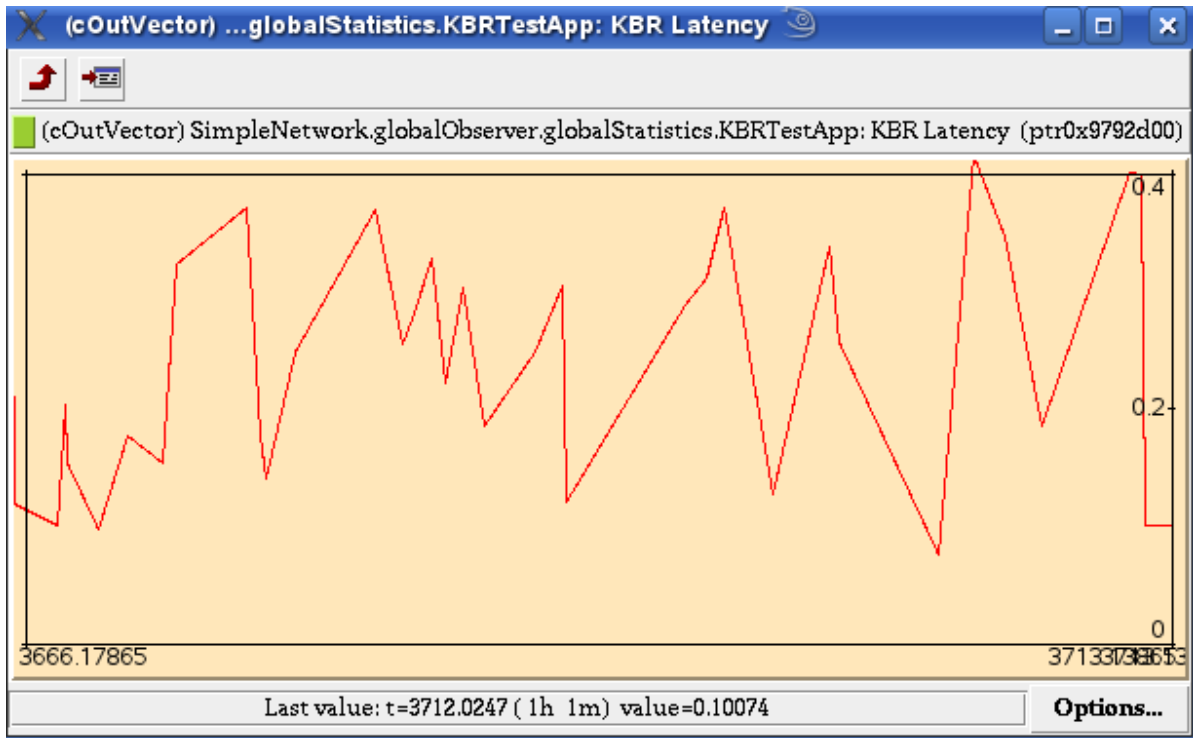
Από τα αποτελέσματα για το Delivery Ratio, παρατηρείται ότι το υπερκείμενο δίκτυο είναι πολύ αποτελεσματικό στην παράδοση των μηνυμάτων.

12.2.2 Προσομοίωση με τη χρήση RandomChurn

Για τη λειτουργία του υπερκείμενου δικτύου με τη χρήση RandomChurn έγιναν 3 προσομοιώσεις με σταθερό αριθμό 50 δρομολογητών και αριθμό τερματικών κόμβων 100, 200, 400. Για την προσομοίωση της λειτουργίας του δικτύου με 50 δρομολογητές και 100 τερματικούς κόμβους ένα ενδεικτικό γράφημα για την καθυστέρηση σε ένα χρονικό διάστημα της προσομοίωσης όπως αυτό παρουσιάζεται με το πρόγραμμα OverSim είναι το παρακάτω.

Η λειτουργία του υπερκείμενου δικτύου είναι όμοια με την προηγούμενη περίπτωση. Κάθε κόμβος που εισέρχεται στο υπερκείμενο δίκτυο εισάγει μια σκανδάλη στον κόμβο δρομολογητή ο οποίος αντιστοιχεί στο κατάλληλο αναγνωριστικό με αυτό του τερματικού κόμβου. Στη συνέχεια, το πακέτο που αποστέλεται δρομολογείται μέχρι τον κόμβο δρομολογητή που αποθηκεύει τη σκανδάλη με το κατάλληλο αναγνωριστικό. Επόμενο βήμα είναι η αποστολή του πακέτου στον κόμβο η διεύθυνση του οποίου βρίσκεται στην σκανδάλη.

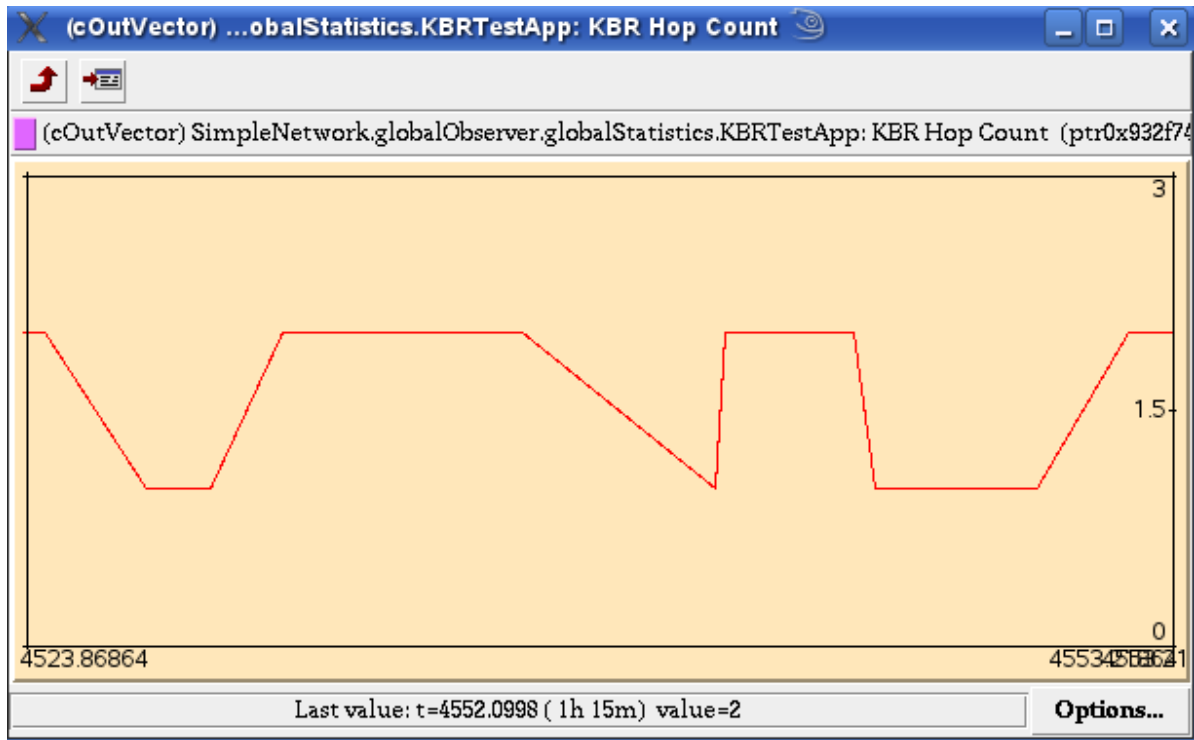
Μετά από αυτή τη διαδικασία, η χρησιμοποίηση του RandomChurn προκαλεί την αποχώρηση κόμβων και την εισαγωγή νέων στο υπερκείμενο δίκτυο με τυχαίο τρόπο. Τα αποτελέσματα της προσομοίωσης για αυτή τη λειτουργία του δικτύου εμφανίζονται στη συνέχεια.



Εικόνα 10: Καθυστέρηση ως προς το χρόνο προσομοίωσης στο RandomChurn

Στο γράφημα αυτό φαίνεται ότι σε σχέση με την προσομοίωση με τον μεγάλο αριθμό κόμβων, εμφανίζονται υψηλότερες τιμές στα μέγιστα, η μέση καθυστέρηση όμως είναι σε κοντινή τιμή.

Στη συνέχεια εμφανίζεται ένα ενδεικτικό γράφημα για τον αριθμό βημάτων που απαιτούνται για την αποστολή ενός μηνύματος όπως παρουσιάζεται στο OverSim.



Εικόνα 11: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης στο RandomChurn

Τα δεδομένα για την προσομοίωση του υπερκείμενου δικτύου με την μεταβολή των τερματικών κόμβων από 100 έως 400, ως προς την καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
100	0.208109
200	0.208865
400	0.205071

Τα δεδομένα για την προσομοίωση του υπερκείμενου δικτύου με την μεταβολή των τερματικών κόμβων από 100 έως 400, ως προς τον αριθμό βημάτων είναι τα παρακάτω.

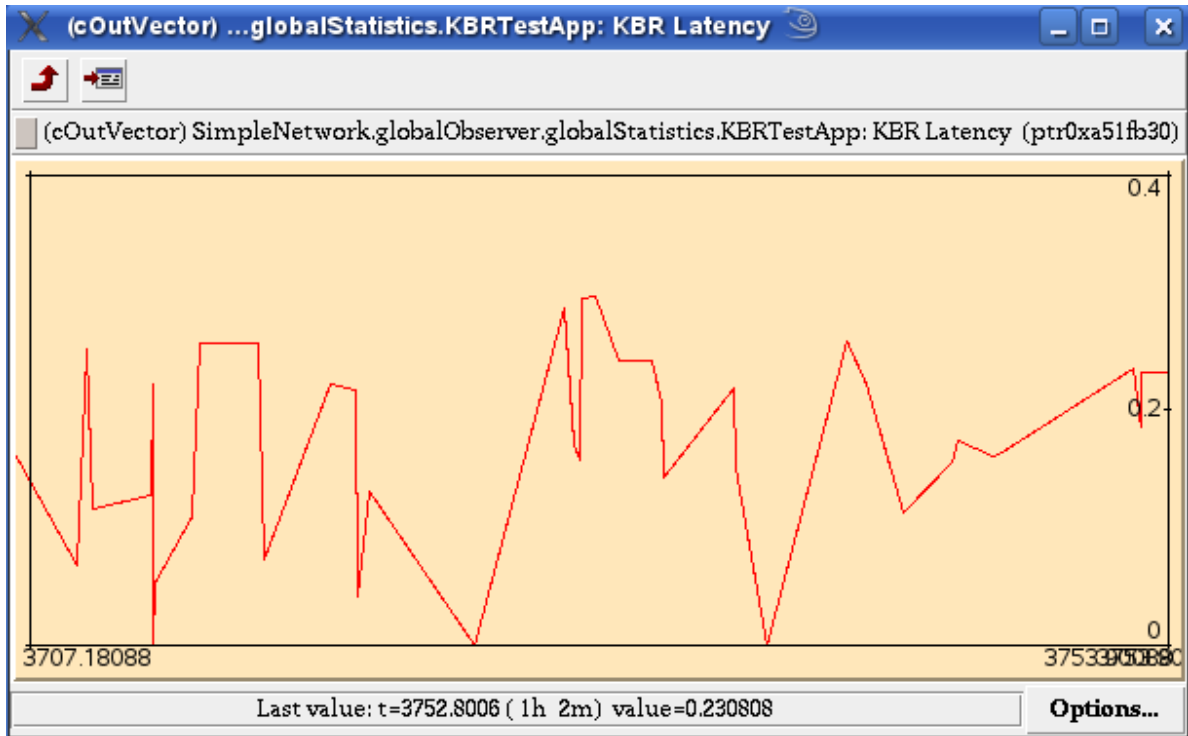
Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
100	1.426601
200	1.439611
400	1.429632

Τα δεδομένα για το Delivery Ratio είναι τα παρακάτω.

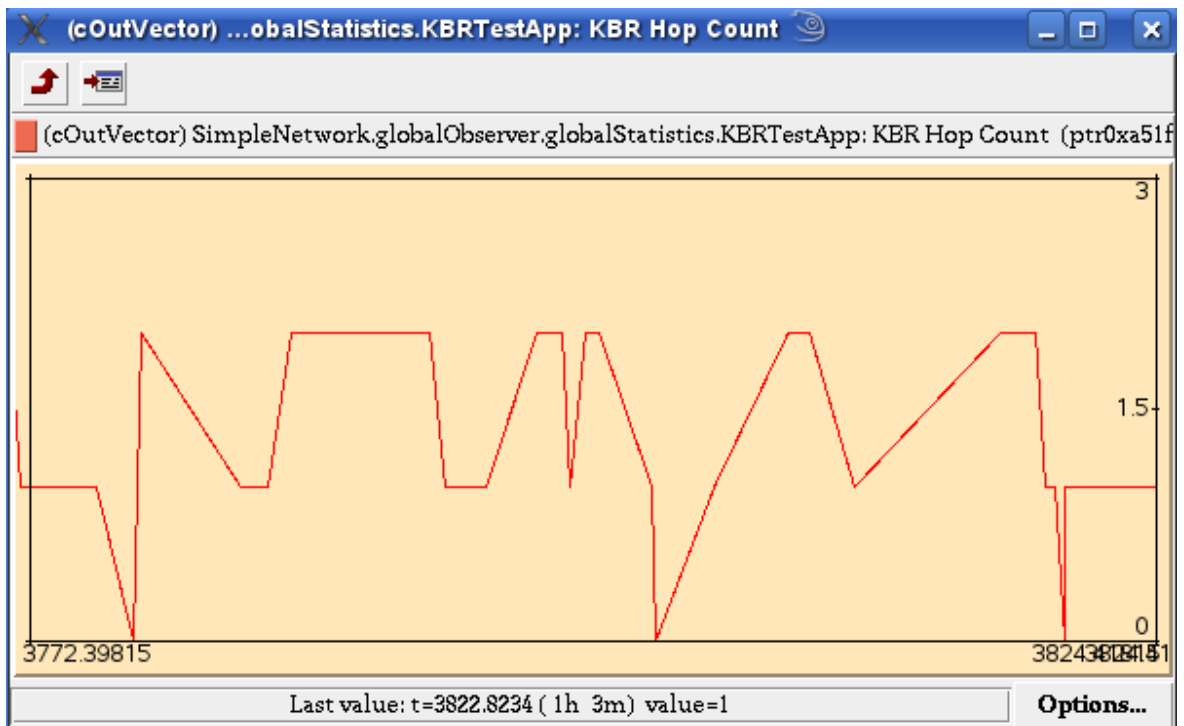
Τερματικοί κόμβοι	Delivery Ratio
100	1
200	1
400	1

12.2.3 Προσομοίωση με τη χρήση LifetimeChurn

Για τη λειτουργία του υπερκείμενου δικτύου με τη χρήση RandomChurn έγιναν 3 προσομοιώσεις με σταθερό αριθμό 50 δρομολογητών και αριθμό τερματικών κόμβων 100, 200, 400. Η διαφορά με την προηγούμενη προσομοίωση είναι ότι στην περίπτωση του LifetimeChurn, ο ρυθμός εισαγωγής των νέων κόμβων είναι σταθερός. Αυτό επιτυγχάνεται με τη χρήση της εντολής `**.churnGeneratorTypes = "NoChurn LifetimeChurn"`. Ενδεικτικά γραφήματα για την καθυστέρηση και τον αριθμό βημάτων σε μια τυχαία χρονική στιγμή με το πρόγραμμα OverSim είναι τα παρακάτω.



Εικόνα 12: Καθυστέρηση ως προς το χρόνο προσομοίωσης στο LifetimeChurn



Εικόνα 13: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης στο LifetimeChurn

Τα δεδομένα για την προσομοίωση του υπερκείμενου δικτύου με την μεταβολή των τερματικών κόμβων από 100 έως 400, ως προς την καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
100	0.168290
200	0.194434
400	0.205102

Τα δεδομένα για την προσομοίωση του υπερκείμενου δικτύου με την μεταβολή των τερματικών κόμβων από 100 έως 400, ως προς τον αριθμό βημάτων είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
100	1.416653
200	1.430471
400	1.429656

Τα δεδομένα για το Delivery Ratio είναι τα παρακάτω.

Τερματικοί κόμβοι	Delivery Ratio
100	1
200	1
400	1

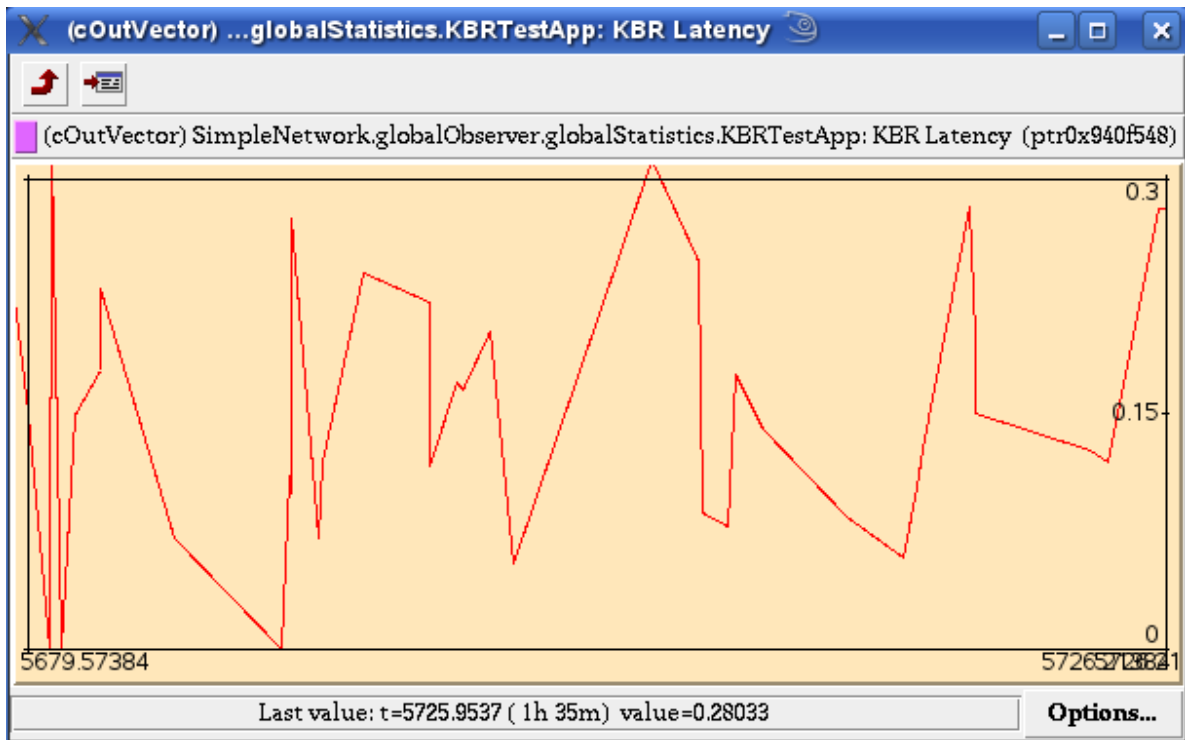
Παρατηρείται ότι με τη χρήση του LifetimeChurn, η μέση καθυστέρηση παρουσιάζει μικρότερες τιμές σε σχέση με τη χρήση του RandomChurn.

12.2.4 Προσομοίωση με πολυεκπομπή

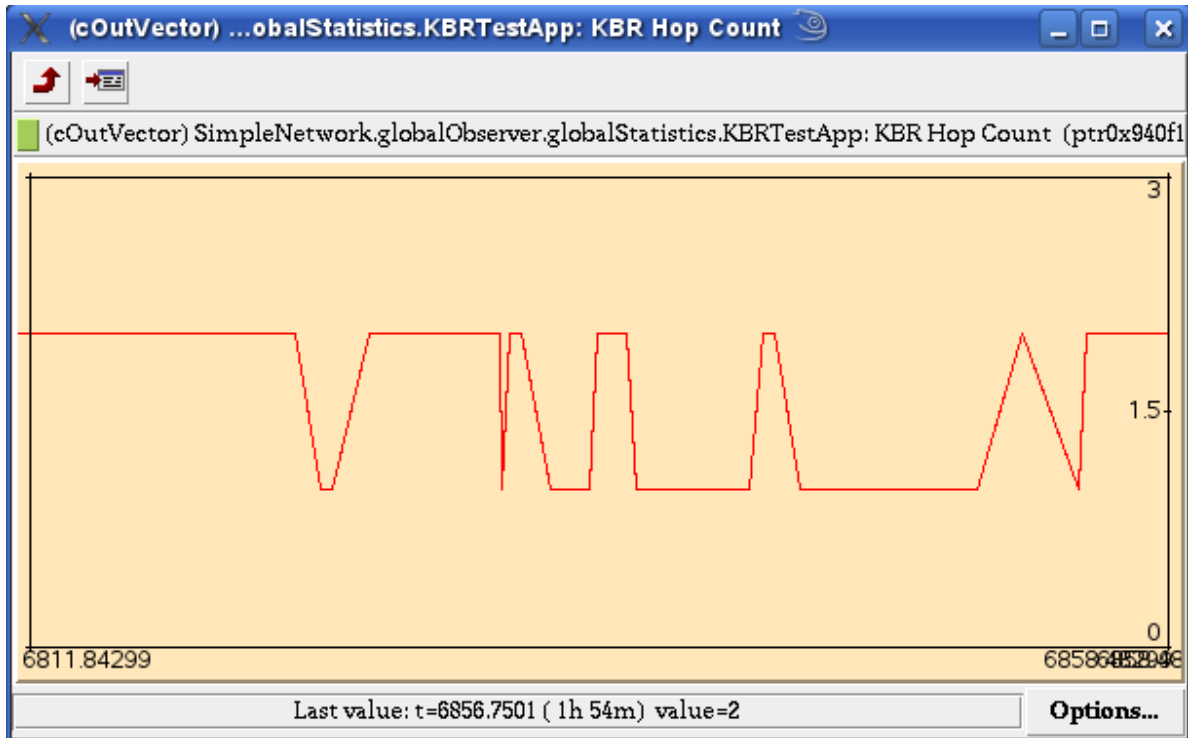
Για τη λειτουργία του υπερκείμενου δικτύου κατά την πολυεκπομπή έγιναν 3 προσομοιώσεις με σταθερό αριθμό 50 δρομολογητών και αριθμό τερματικών κόμβων 100, 200, 400. Για την προσομοίωση της λειτουργίας του δικτύου με 50 δρομολογητές και 100 τερματικούς κόμβους ένα ενδεικτικό γράφημα για την καθυστέρηση σε ένα χρονικό διάστημα της προσομοίωσης όπως αυτό παρουσιάζεται με το πρόγραμμα OverSim είναι το παρακάτω.

Η λειτουργία του υπερκείμενου δικτύου κατά την πολυεκπομπή είναι ως εξής. Κάθε κόμβος που εισέρχεται στο υπερκείμενο δίκτυο εισάγει μια σκανδάλη στον κόμβο δρομολογητή ο οποίος αντιστοιχεί στο κατάλληλο αναγνωριστικό με αυτό του τερματικού κόμβου. Στη συνέχεια, το πακέτο που αποστέλεται από τον κόμβο αποστολέα δρομολογείται μέχρι τον κόμβο δρομολογητή που αποθηκεύει τις σκανδάλεις με το κατάλληλο αναγνωριστικό και τις αντίστοιχες διευθύνσεις των τερματικών κόμβων δεκτών στις οποίες αποστέλεται το πακέτο της πολυεκπομπής.

Παρακάτω εμφανίζονται ενδεικτικά γραφήματα για την καθυστέρηση και τον αριθμό βημάτων που απαιτούνται για την αποστολή ενός μηνύματος όπως παρουσιάζεται στο OverSim.



Εικόνα 14: Καθυστέρηση ως προς το χρόνο προσομοίωσης στην πολυεκπομπή



Εικόνα 15: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης στην πολυεκπομπή

Τα δεδομένα για την προσομοίωση του υπερκείμενου δικτύου με την μεταβολή των τερματικών κόμβων από 100 έως 400, ως προς την καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
100	0.166191
200	0.167119
400	0.166963

Τα δεδομένα για την προσομοίωση του υπερκείμενου δικτύου με την μεταβολή των τερματικών κόμβων από 100 έως 400, ως προς τον αριθμό βημάτων είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
100	1.423077
200	1.439150
400	1.431714

Τα δεδομένα για το Delivery Ratio είναι τα παρακάτω.

Τερματικοί κόμβοι	Delivery Ratio
100	1
200	1
400	1

12.3 Σύγκριση αποτελεσμάτων

Η λειτουργία του προτεινόμενου υπερκείμενου δικτύου υλοποιείται με τη χρήση του τροποποιημένου πρωτοκόλου Pastry. Όπως παρατηρείται από τα αποτελέσματα της προσομοίωσης το πλεονέκτημα είναι η μικρή μέση καθυστέρηση και ο μικρός αριθμός βημάτων που αποτελείται για την αποστολή ενός μηνύματος. Ένας άλλος τρόπος λειτουργίας για ένα υπερκείμενο δίκτυο που χρησιμοποιεί τη λογική των σκανδαλών θα ήταν η χρησιμοποίηση του Chord σε αντίθεση με τη χρησιμοποίηση του Pastry, ως εναλλακτικός τρόπος εφαρμογής και σύγκρισης ενός γνωστού τρόπου υπερκείμενης δρομολόγησης.

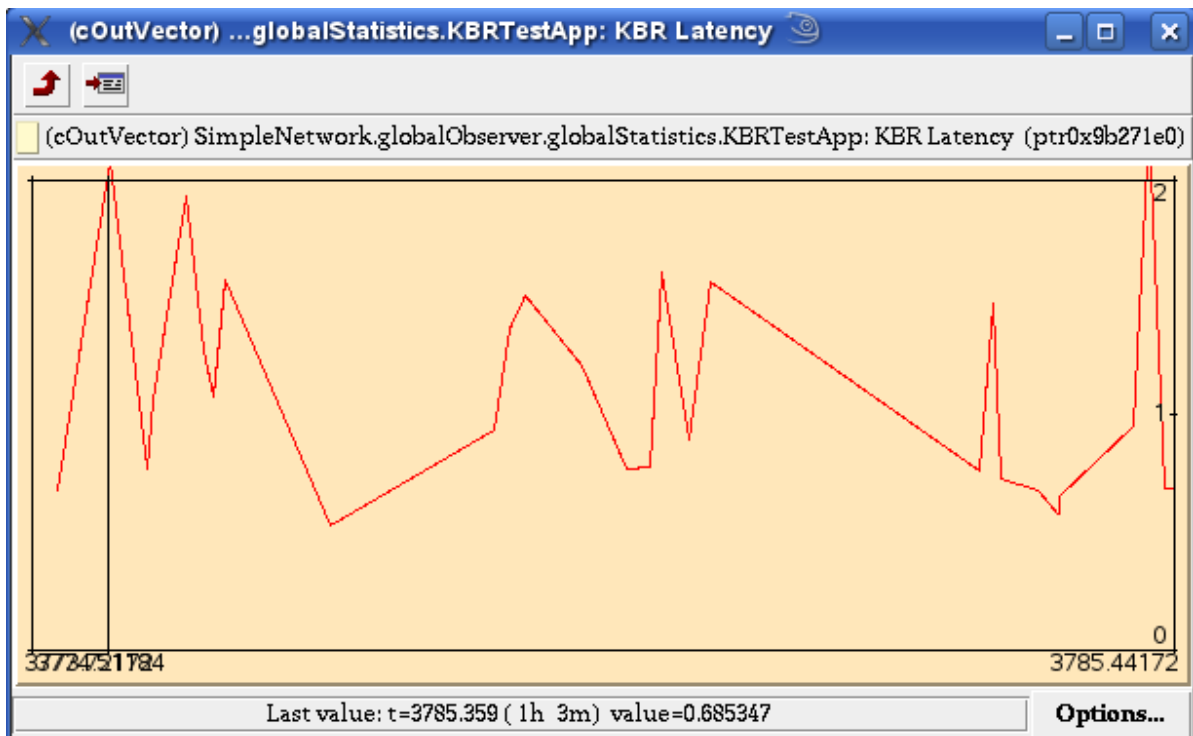
Για μια τέτοια υλοποίηση τα αποτελέσματα της προσομοίωσης όπως προκύπτουν από το OverSim για σενάρια προσομοίωσης ίδια με τα προηγούμενα, εμφανίζονται στη συνέχεια.

12.3.1 Προσομοίωση για μεγάλο αριθμό κόμβων

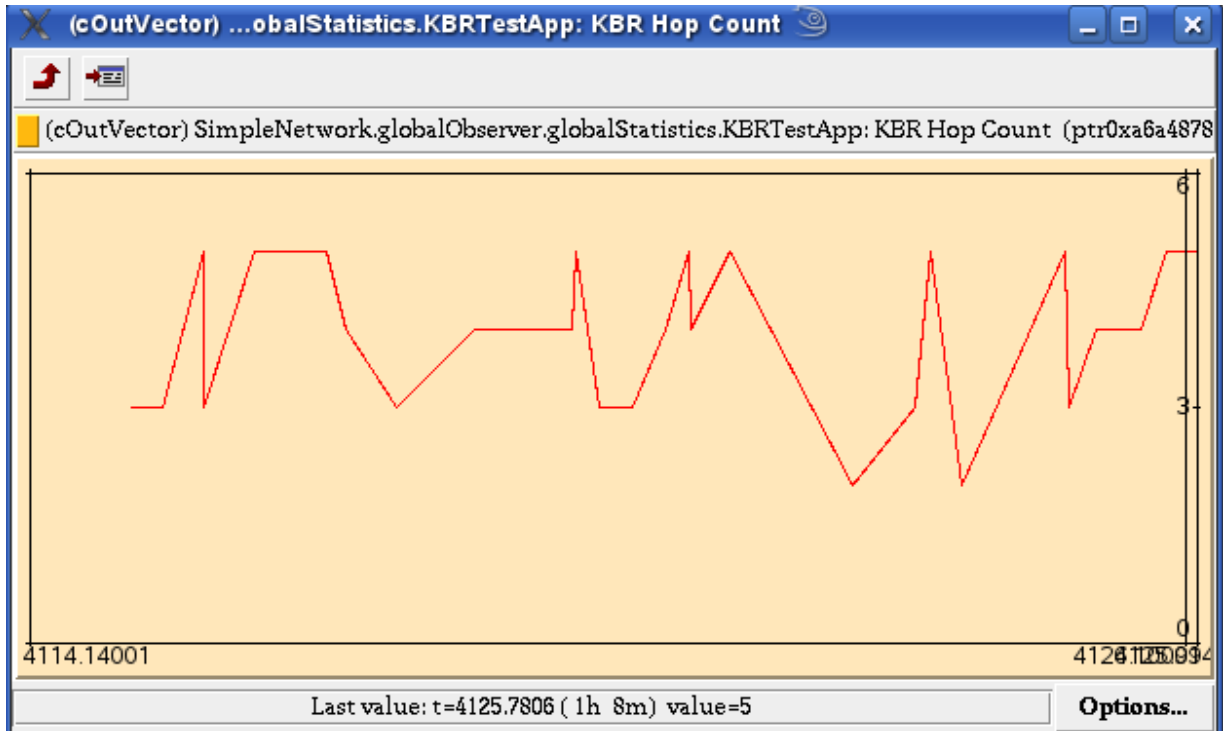
Για την πρώτη περίπτωση, δηλαδή την προσομοίωση με μεγάλο αριθμό κόμβων, ο κώδικας προσομοίωσης θα είναι:

```
description = "I3 over Chord (SimpleNetwork, no churn)"
network = SimpleNetwork
**.churnGeneratorTypes = "NoChurn NoChurn"
**[0].overlayType = "ChordModules"
**[0].tier1Type = "KBRTTestAppModules"
**[0].targetOverlayTerminalNum = 100
**[1].overlayType = "OverlayDummyModules"
**[1].tier1Type = "I3HostMobility"
**[1].targetOverlayTerminalNum = 1000
**.neighborCache.enableNeighborCache = true
```

Για αυτή την υλοποίηση χρησιμοποιούνται τα ChordModules του προγράμματος OverSim, αντί για τα PastryModules που χρησιμοποιήθηκαν στην προηγούμενη προσομοίωση. Ενδεικτικά γραφήματα σε μια τυχαία χρονική στιγμή για την καθυστέρηση και τον αριθμό βημάτων σε μια τυχαία χρονική στιγμή είναι τα παρακάτω.



Εικόνα 16 : Καθυστέρηση ως προς το χρόνο προσομοίωσης για μεγάλο αριθμό κόμβων



Εικόνα 17: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης για μεγάλο αριθμό κόμβων

Τα αποτελέσματα σε συνθήκες ίδιες με αυτές της προσομοίωσης του προτεινόμενου υπερκείμενου δικτύου, δηλαδή με μεταβολή των τερματικών κόμβων από 200 έως 1000 είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
200	1.069028
400	1.074136
600	1.069371
800	1.064881
1000	1.060946

Ομοίως, τα αποτελέσματα για την προσομοίωση του υπερκείμενου δικτύου με τη χρήση του πρωτοκόλλου Chord, με την μεταβολή των τερματικών κόμβων από 200 έως 1000, ως προς τον αριθμό βημάτων είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
200	4.052455
400	4.062059
600	4.051300
800	4.035717
1000	4.030183

Τα δεδομένα για το Delivery Ratio είναι τα παρακάτω.

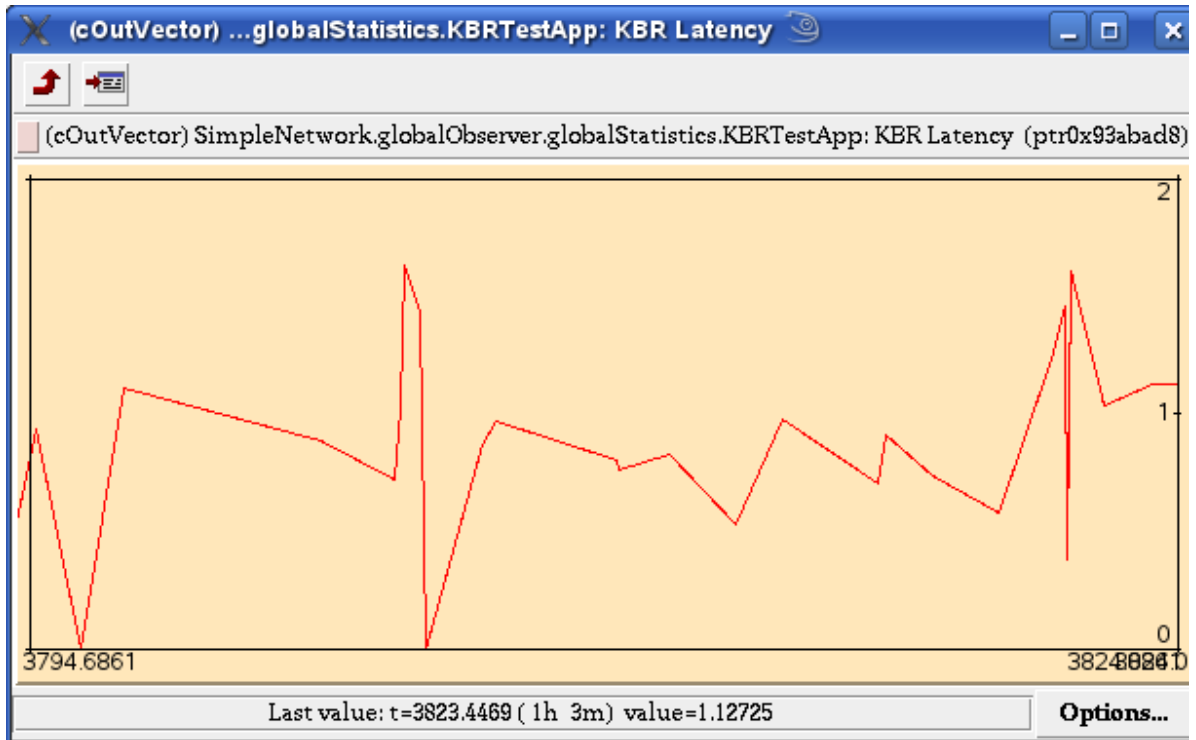
Τερματικοί κόμβοι	Delivery Ratio
200	0.999914
400	0.999912
600	0.999727
800	0.999726
1000	1

12.3.2 Προσομοίωση με τη χρήση RandomChurn

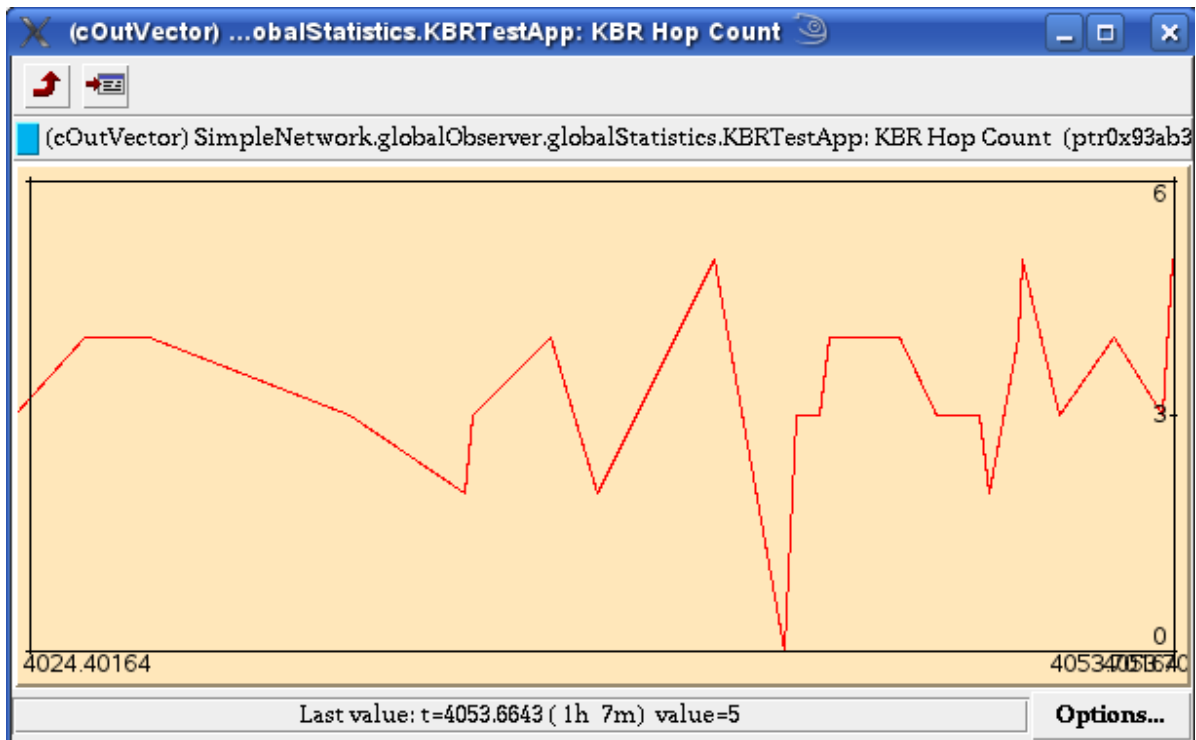
Για τη δεύτερη περίπτωση, ο κώδικας προσομοίωσης είναι ο εξής:

```
description = "I3 over Chord (SimpleNetwork, RandomChurn)"
network = SimpleNetwork
**.churnGeneratorTypes = "NoChurn RandomChurn"
**[0].overlayType = "ChordModules"
**[0].tier1Type = "KBRTTestAppModules"
**[0].targetOverlayTerminalNum = 50
**[1].overlayType = "OverlayDummyModules"
**[1].tier1Type = "I3HostMobility"
**[1].targetOverlayTerminalNum = 400
**.neighborCache.enableNeighborCache = true
```

Έχει γίνει η αλλαγή με τη χρήση της εντολής `**[0].overlayType = "ChordModules"`, αφού η δρομολόγηση γίνεται με το πρωτόκολο Chord. Τα γραφήματα για την καθυστέρηση και το αριθμό βημάτων σε μια τυχαία χρονική στιγμή είναι τα παρακάτω.



Εικόνα 18: Καθυστέρηση ως προς το χρόνο προσομοίωσης στο RandomChurn



Εικόνα 19: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης στο RandomChurn

Τα αποτελέσματα της προσομοίωσης με τη χρήση RandomChurn για τη μέση καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
100	0.874725
200	0.869839
400	0.869798

Τα αποτελέσματα για το μέσο αριθμό βημάτων με τη χρήση RandomChurn και την εφαρμογή του πρωτοκόλλου Chord, είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
100	3.560121
200	3.551565
400	3.527400

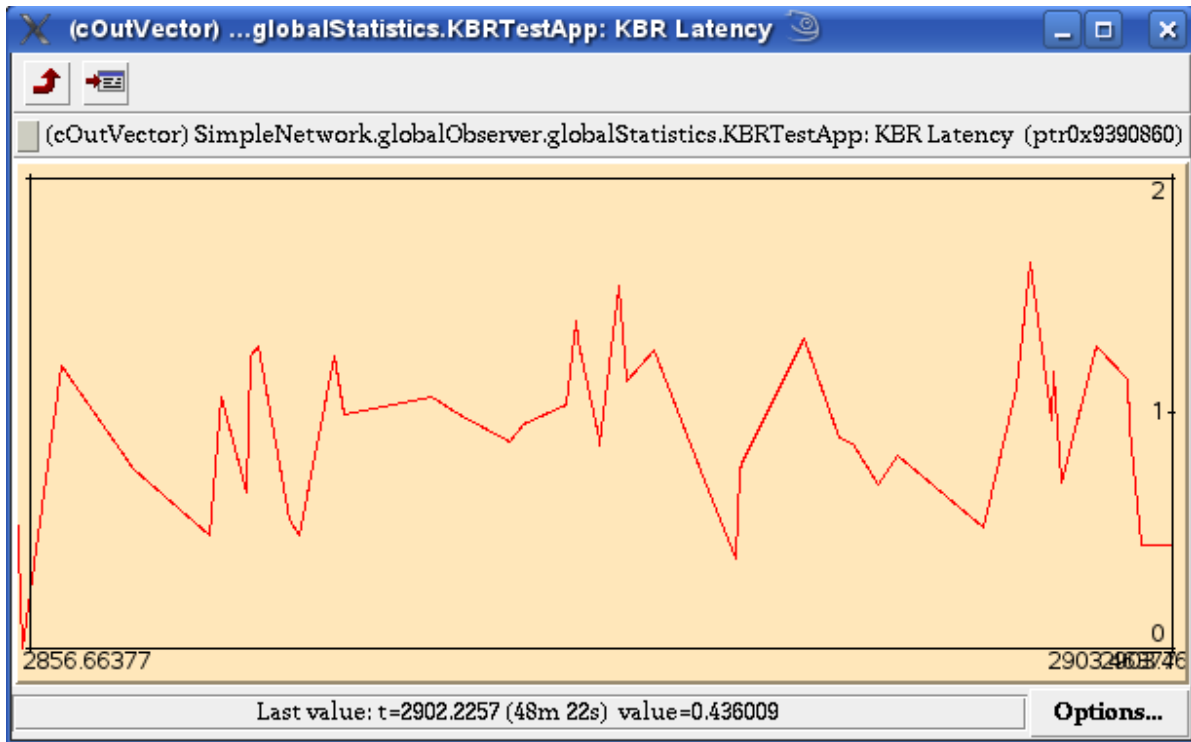
Τα αποτελέσματα για το Delivery Ratio με τη χρήση RandomChurn και την εφαρμογή του πρωτοκόλλου Chord, είναι τα παρακάτω.

Τερματικοί κόμβοι	Delivery Ratio
100	1
200	0.999830
400	0.999472

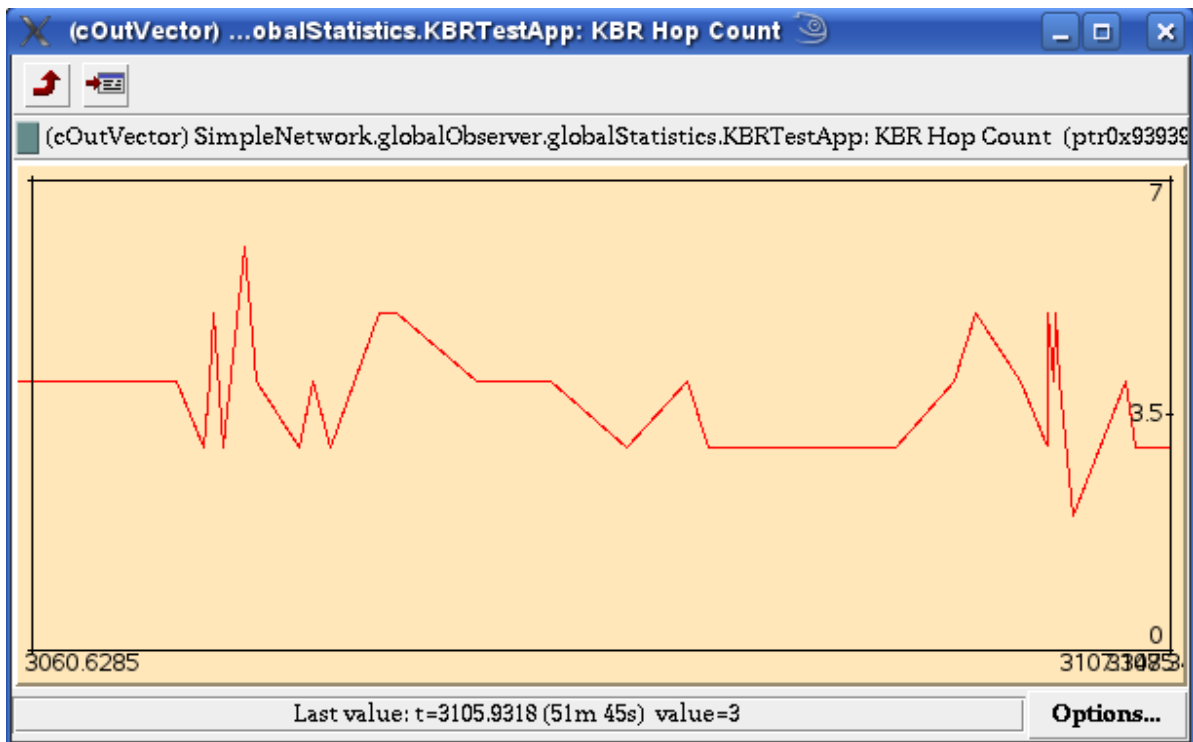
12.3.3 Προσομοίωση με τη χρήση LifetimeChurn

```
description = "I3 over Chord (SimpleNetwork, RandomChurn)"
network = SimpleNetwork
**.churnGeneratorTypes = "NoChurn LifetimeChurn"
**[0].overlayType = "ChordModules"
**[0].tier1Type = "KBRTestAppModules"
**[0].targetOverlayTerminalNum = 50
**[1].overlayType = "OverlayDummyModules"
**[1].tier1Type = "I3HostMobility"
**[1].targetOverlayTerminalNum = 400
**.neighborCache.enableNeighborCache = true
```


Όπως φαίνεται και από τον κώδικα, έχει αλλαχθεί το RandomChurn με την εντολή LifetimeChurn, ενώ η εντολή PastryModules με την εντολή ChordModules για τη λειτουργία με το πρωτόκολλο Chord για αυτή την προσομοίωση.



Εικόνα 20: Καθυστέρηση ως προς το χρόνο προσομοίωσης στο LifetimeChurn



Εικόνα 21: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης στο LifetimeChurn

Τα αποτελέσματα της προσομοίωσης με τη χρήση RandomChurn για τη μέση καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
100	0.963243
200	0.868929
400	0.924345

Τα αποτελέσματα για το μέσο αριθμό βημάτων με τη χρήση RandomChurn και την εφαρμογή του πρωτοκόλλου Chord, είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
100	3.573589
200	3.531639
400	3.520277

Τα αποτελέσματα για το Delivery Ratio με τη χρήση RandomChurn και την εφαρμογή του πρωτοκόλλου Chord, είναι τα παρακάτω.

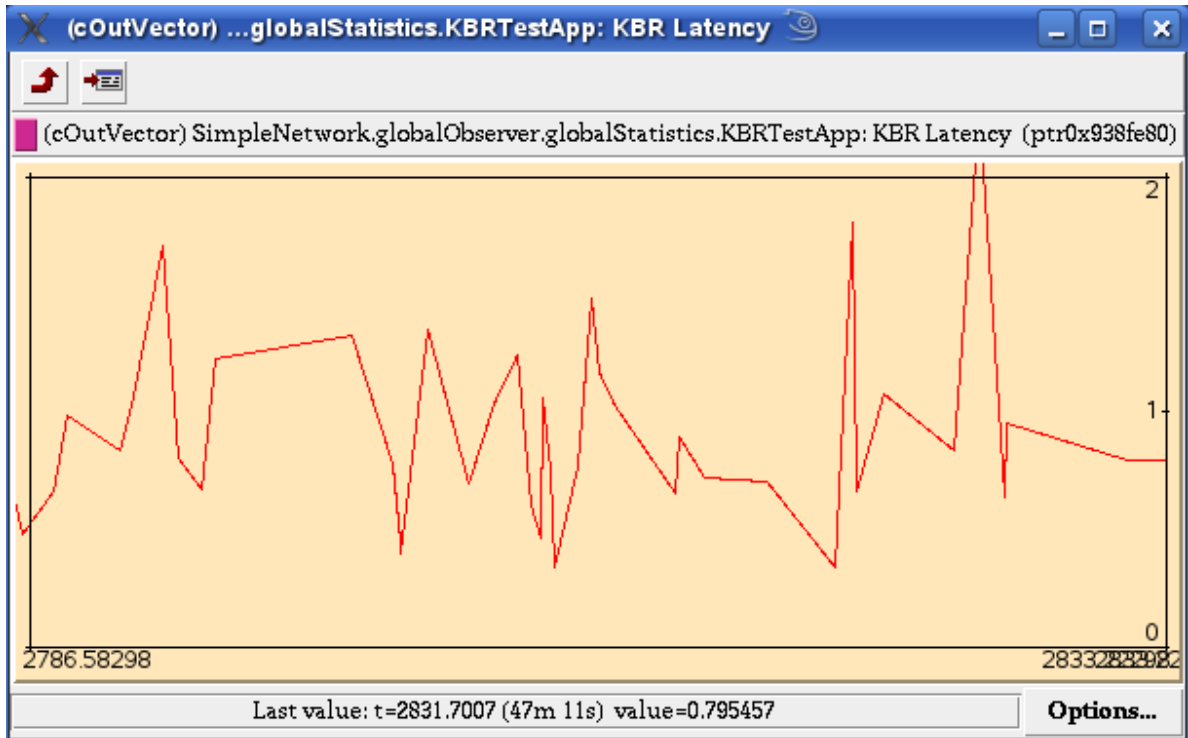
Τερματικοί κόμβοι	Delivery Ratio
100	0.999833
200	1
400	0.999829

12.3.4 Προσομοίωση με πολυεκπομπή

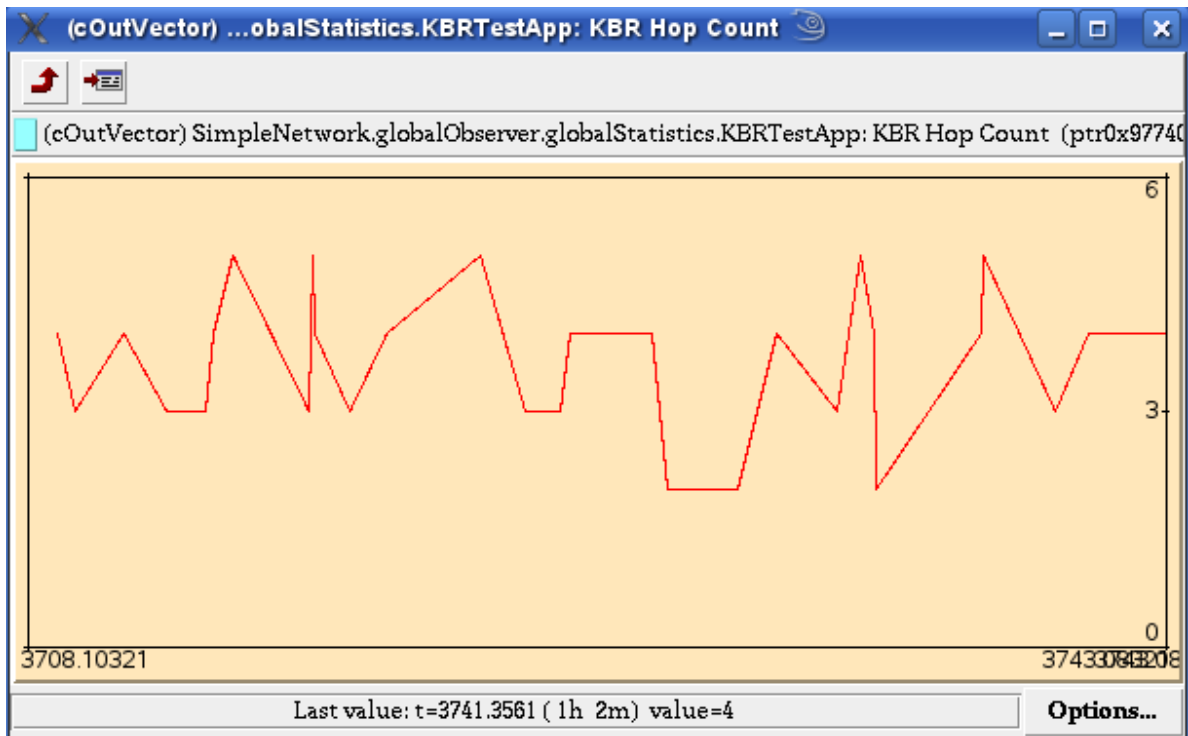
Για την τρίτη περίπτωση, της πολυεκπομπής, ο κώδικας για την προσομοίωση θα είναι:

```
description = "I3 Multicast over Chord (SimpleNetwork, no churn)"
network = SimpleNetwork
**.churnGeneratorTypes = "NoChurn NoChurn"
**[0].overlayType = "ChordModules"
**[0].tier1Type = "KBRTestAppModules"
**[0].targetOverlayTerminalNum = 50
**[1].overlayType = "OverlayDummyModules"
**[1].tier1Type = "I3Multicast"
**[1].targetOverlayTerminalNum = 400
**.neighborCache.enableNeighborCache = true
```

Τα γραφήματα για την καθυστέρηση και τον αριθμό βημάτων σε μια τυχαία χρονική στιγμή όπως απεικονίζονται με το OverSim είναι τα παρακάτω.



Εικόνα 22: Καθυστέρηση ως προς το χρόνο προσομοίωσης στην πολυεκπομπή



Εικόνα 23: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης στην πολυεκπομπή

Τα αποτελέσματα της προσομοίωσης κατά τη χρήση της πολυεκπομπής για τη μέση καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
100	0.968135
200	0.970224
400	0.967852

Τα αποτελέσματα για το μέσο αριθμό βημάτων είναι τα παρακάτω.

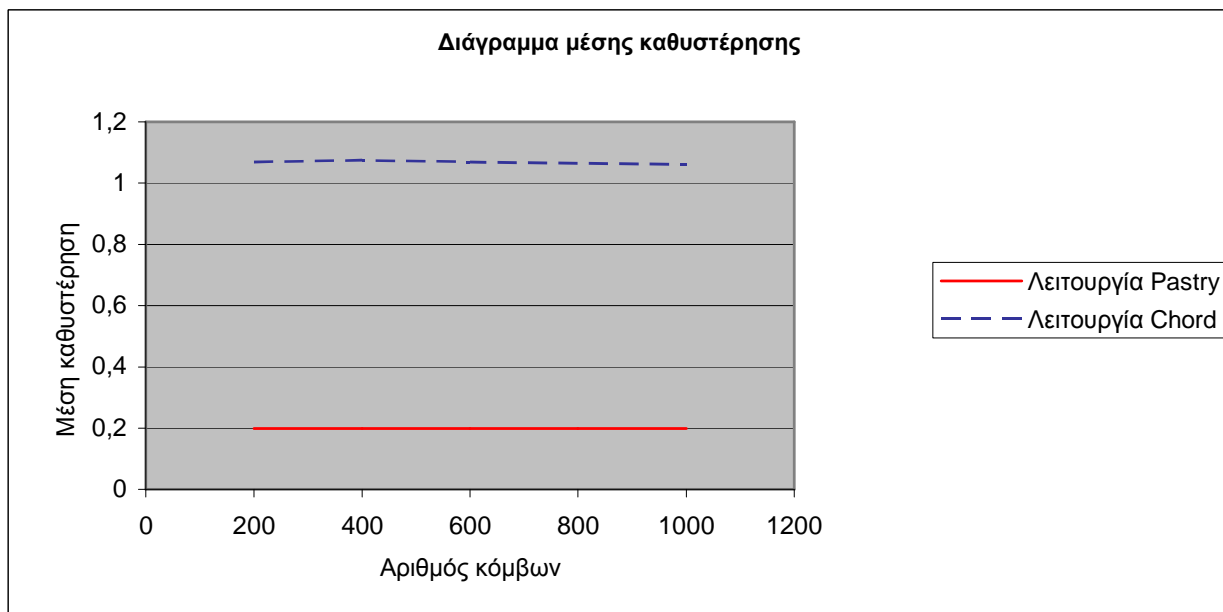
Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
100	3.540423
200	3.550008
400	3.549610

Τα αποτελέσματα για το Delivery Ratio κατά την πολυεκπομπή και την εφαρμογή του πρωτοκόλου Chord, είναι τα παρακάτω.

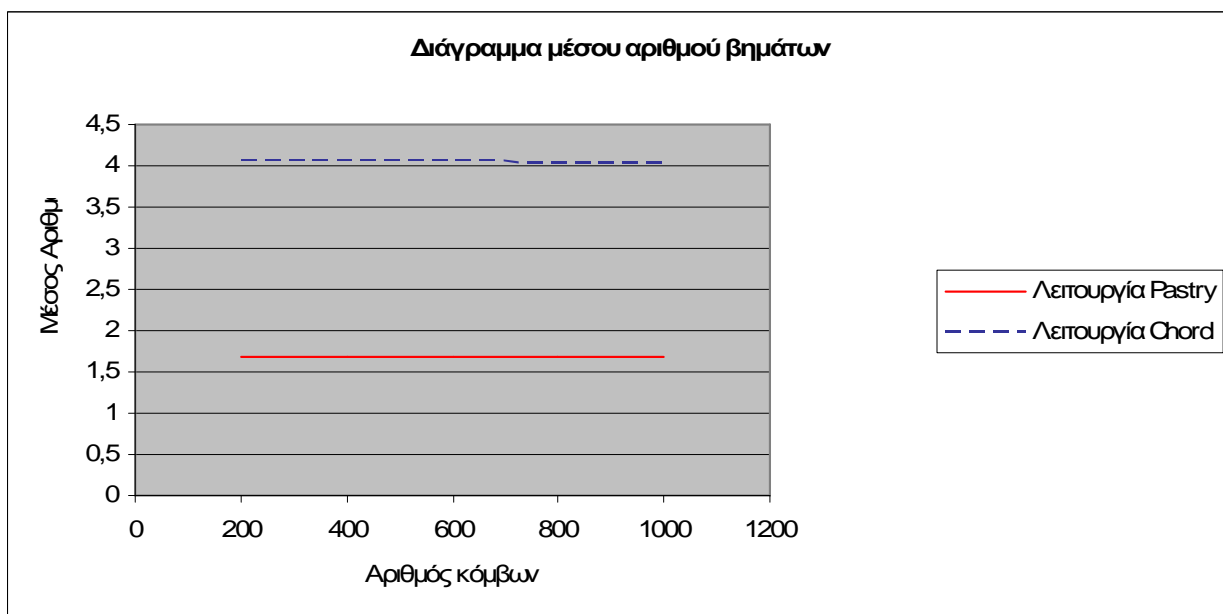
Τερματικοί κόμβοι	Delivery Ratio
100	1
200	0.999829
400	0.999826

Με τη χρήση των παρακάτω διαγραμμάτων απεικονίζεται η διαφορά μεταξύ των διαφορετικών προσομοιώσεων ως προς τη μέση καθυστέρηση και το μέσο αριθμό βημάτων. Για το Delivery Ratio, παρατηρείται ότι είναι σε κάθε προσομοίωση που εκτελέστηκε κοντά στη μονάδα. Δηλαδή σε σχέση με την ικανότητα της μετάδοσης των μηνυμάτων χωρίς να απορρίπτονται και να χάνονται μηνύματα, οι υπερκείμενες τεχνικές είναι πολύ αποτελεσματικές.

Για την υλοποίηση με μεγάλο αριθμο κόμβων τα διαγράμματα όπως προκύπτουν από τα αποτελέσματα των προσομοιώσεων είναι τα παρακάτω.



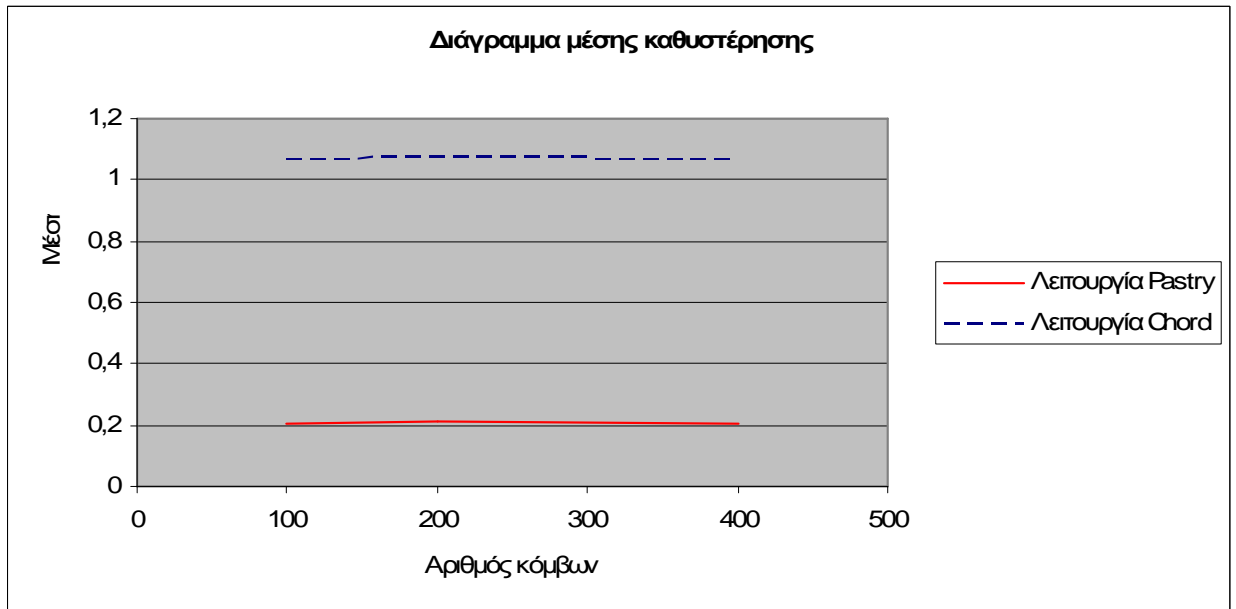
Εικόνα 24: Σύγκριση μέσης καθυστέρησης για μεγάλο αριθμό κόμβων



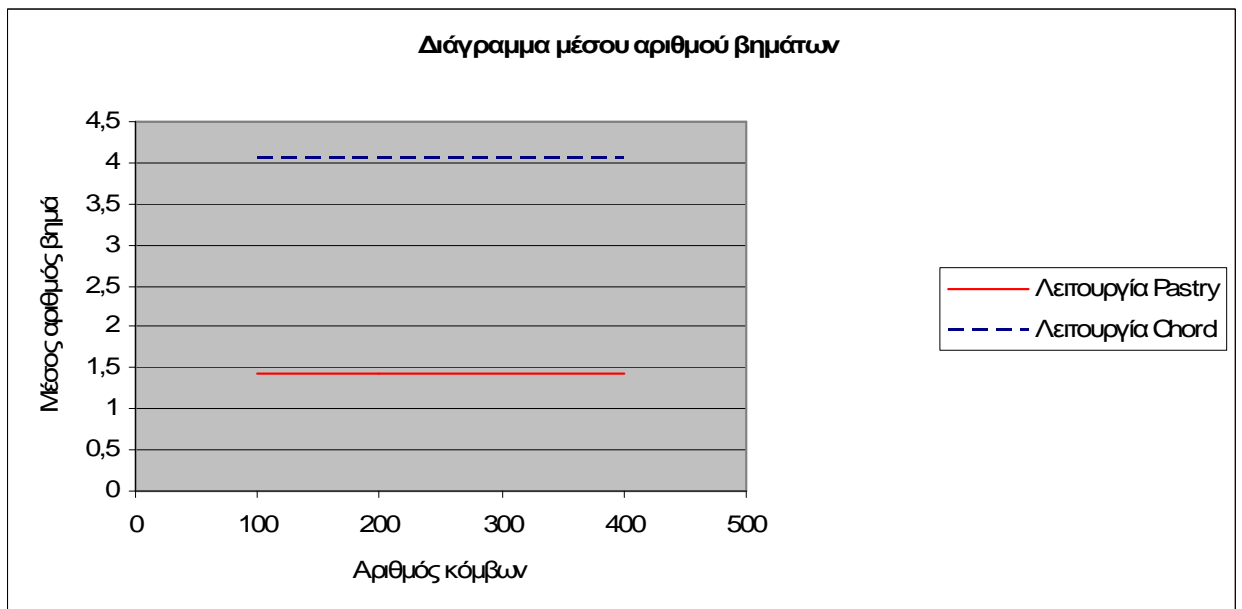
Εικόνα 25: Σύγκριση μέσου αριθμού βημάτων για μεγάλο αριθμό κόμβων

Παρατηρείται ότι τόσο η μέση καθυστέρηση όσο και ο μέσος αριθμός βημάτων είναι μικρότερα με τη χρήση του Pastry σε σχέση με τη χρήση του Chord για την προσομοίωση με μεγάλο αριθμό κόμβων.

Για την προσομοίωση με τη χρήση του RandomChurn, τα διαγράμματα για τη μέση καθυστέρηση και το μέσο αριθμό βημάτων είναι σύμφωνα με τα αποτελέσματα είναι τα παρακάτω.



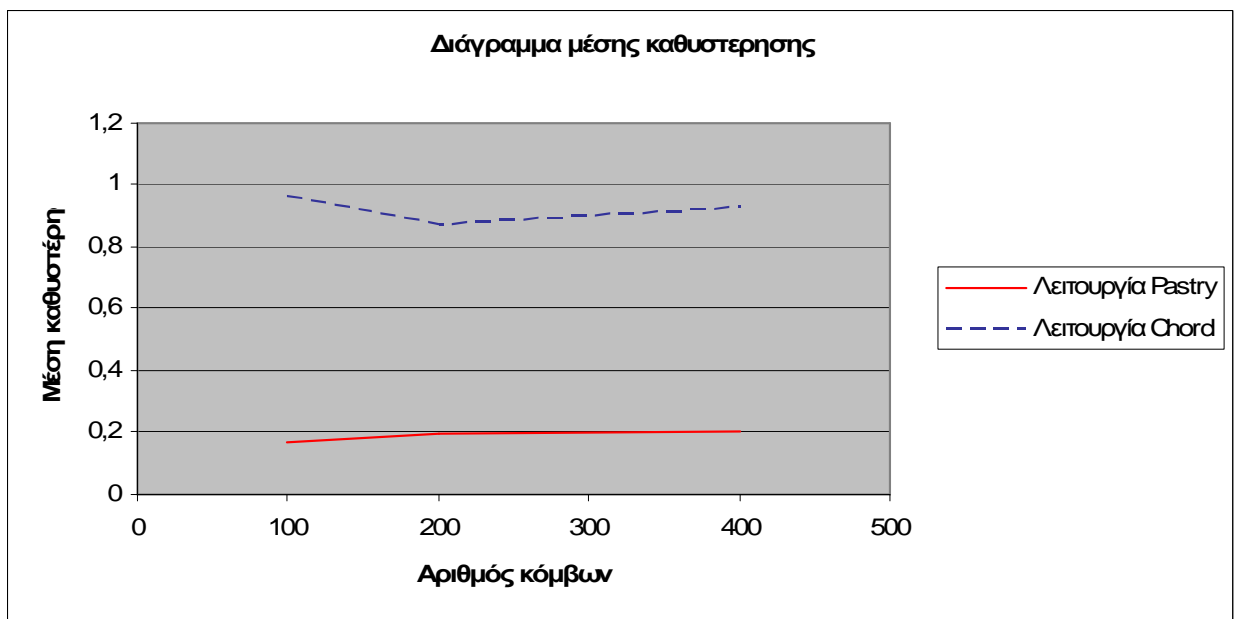
Εικόνα 26: Σύγκριση μέσης καθυστέρησης με τη χρήση RandomChurn



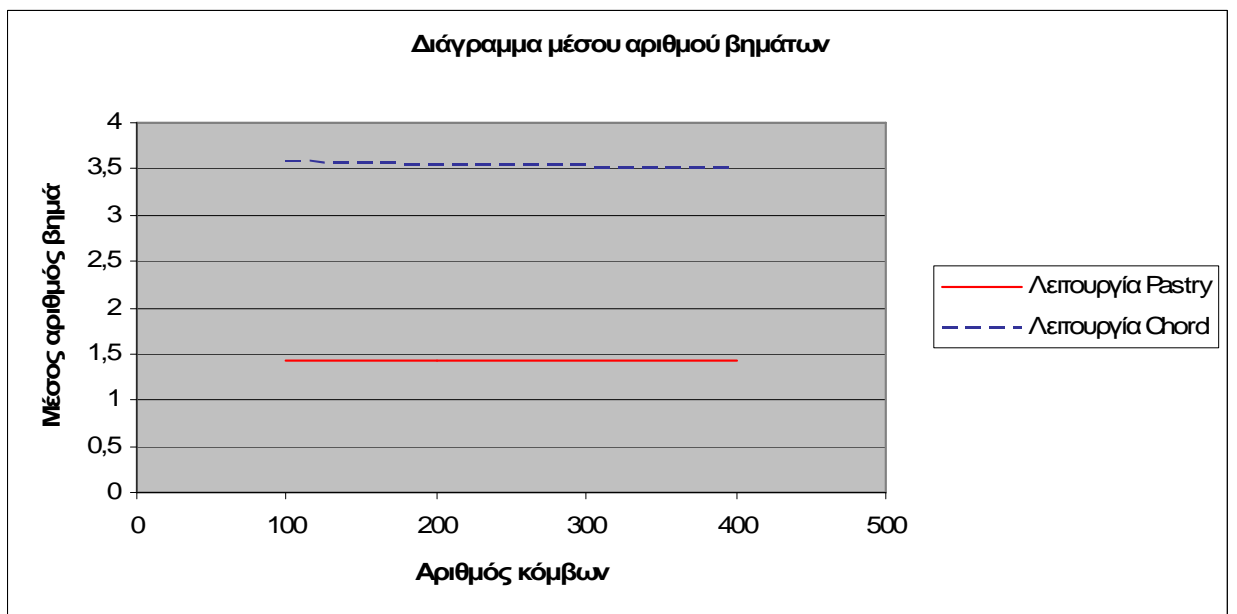
Εικόνα 27: Σύγκριση μέσου αριθμού βημάτων με τη χρήση RandomChurn

Παρατηρείται ότι και για τη χρήση του RandomChurn για την λειτουργία του υπερκείμενου δικτύου, με το Pastry επιτυγχάνονται μικρότερες τιμές για τη μέση καθυστέρηση και το μέσο αριθμό βημάτων απ'ότι με το Chord.

Για την προσομοίωση με τη χρήση του LifetimeChurn, τα διαγράμματα για τη μέση καθυστέρηση και το μέσο αριθμό βημάτων είναι σύμφωνα με τα αποτελέσματα είναι τα παρακάτω.

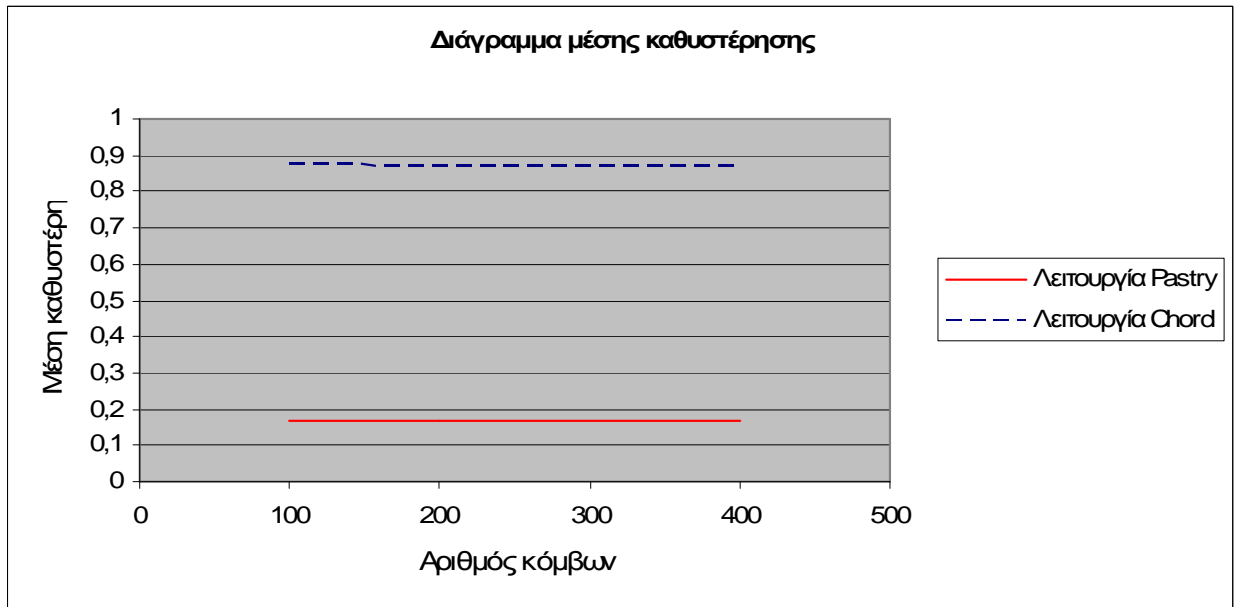


Εικόνα 28: Σύγκριση μέσης καθυστέρησης με τη χρήση LifetimeChurn

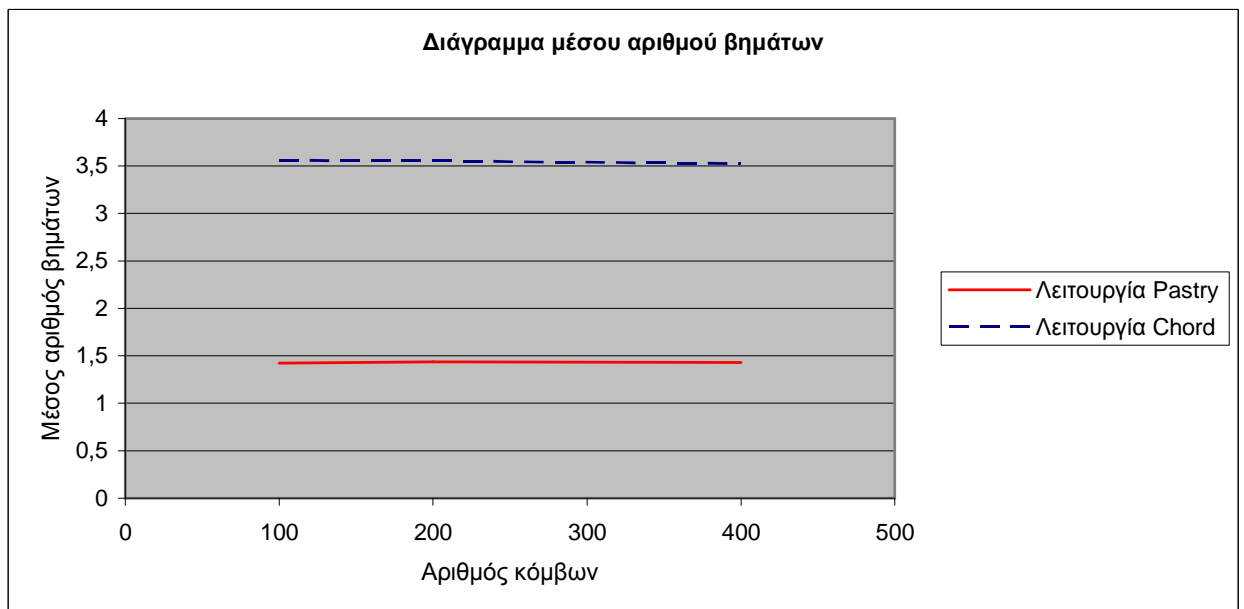


Εικόνα 29: Σύγκριση μέσου αριθμού βημάτων με τη χρήση LifetimeChurn

Σύμφωνα με τα αποτελέσματα κατά την προσομοίωση της πολυεκπομπής στο προτεινόμενο υπερκείμενο δίκτυο προκύπτουν τα παρακάτω διαγράμματα για τη μέση καθυστέρηση και το μέσο αριθμό βημάτων.



Εικόνα 30: Σύγκριση μέσης καθυστέρησης κατά την πολυεκπομπή



Εικόνα 31: Σύγκριση μέσου αριθμού βημάτων κατά την πολυεκπομπή

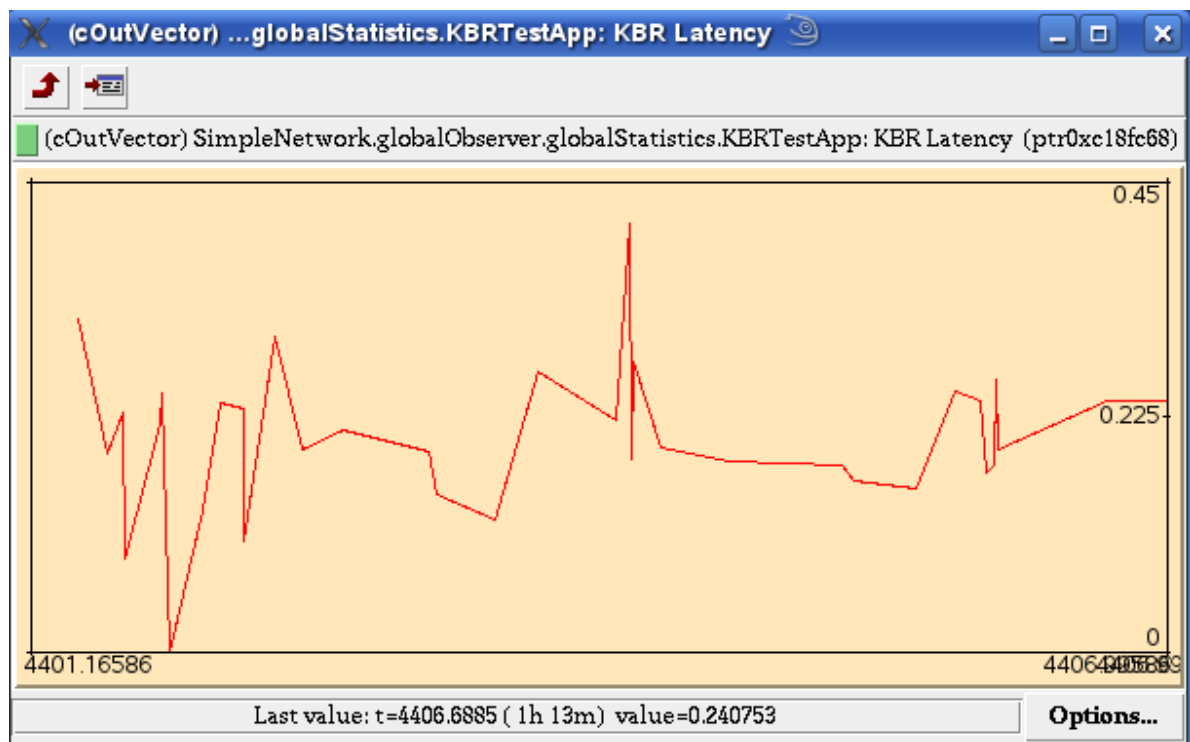
Στο προτεινόμενο υπερκείμενο δίκτυο, η προώθηση των μηνυμάτων γίνεται στους κόμβους δρομολογητές. Όταν ένας κόμβος επιθυμεί την παραλαβή δεδομένων εισάγει τη σκανδάλη σε ένα δρομολογητή. Στη συνέχεια τα πακέτα προωθούνται στους κόμβους δρομολογητές έως ότου βρουν την κατάλληλη σκανδάλη για να σταλούν στον τερματικό κόμβο που την έχει εισάγει. Η δρομολόγηση των μηνυμάτων και των πακέτων γίνεται στους κόμβους δρομολογητές. Όταν ο αριθμός των δρομολογητών παραμένει σταθερός, η καθυστέρηση στην προώθηση των μηνυμάτων και ο αριθμός των βημάτων παραμένουν και αυτά σταθερά.

Στις προσομοιώσεις όπου ο αριθμός των δρομολογητών παραμένει σταθερός και αυξάνεται ο αριθμός των τερματικών κόμβων, τα αποτελέσματα είναι σχεδόν σταθερά. Επιλέχθηκε αυτή η μεθοδολογία για να παρατηρηθεί η ικανότητα του προτεινόμενου υπερκείμενου δικτύου να εξυπηρετήσει μεγάλο αριθμό χρηστών που είναι οι τερματικοί κόμβοι, με σταθερό έναν ικανό αριθμό δρομολογητών για τη μεταφορά των μηνυμάτων. Έτσι όταν αυξάνεται ο αριθμός των τερματικών χρηστών δεν επηρεάζεται αρνητικά η καθυστέρηση και ο αριθμός βημάτων, αφού η προώθηση γίνεται στους κόμβους δρομολογητές που παραμένουν σταθεροί, κάτι που επιβεβαιώνει την ιδιότητα του προτεινόμενου δικτύου να είναι κλιμακούμενο. Δηλαδή, να μπορεί να εξυπηρετήσει αυξανόμενο αριθμό τερματικών χρηστών.

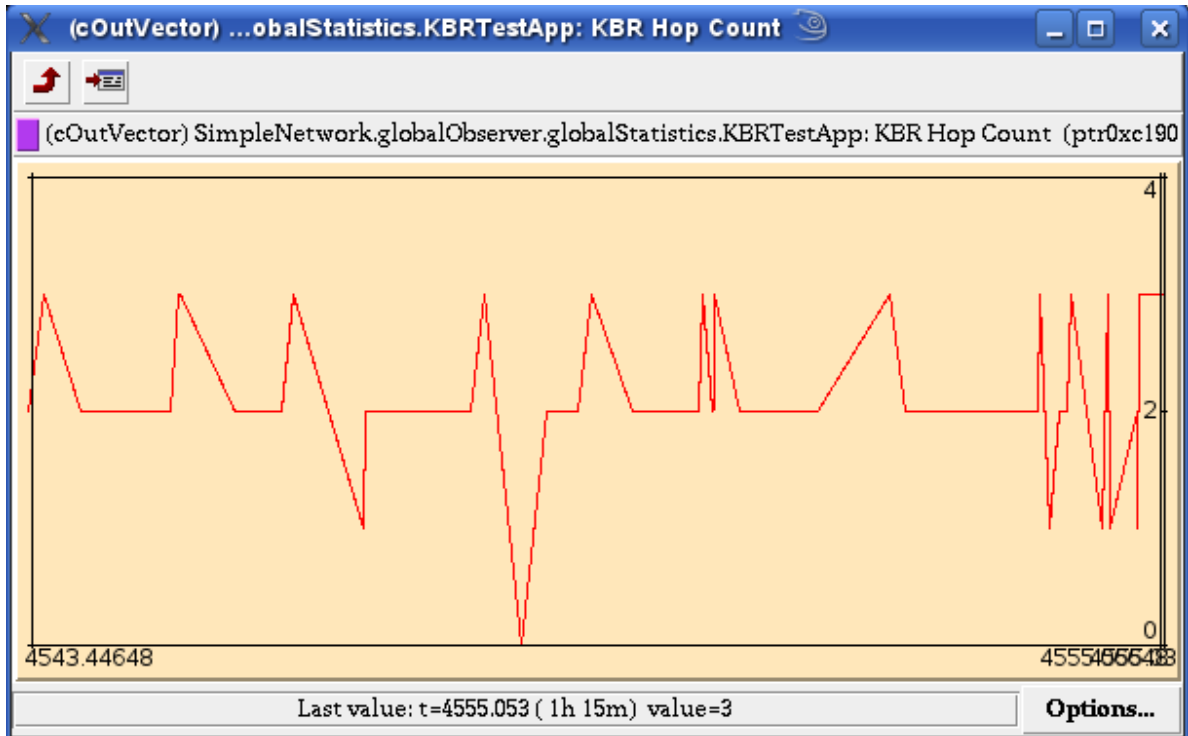
Από τα αποτελέσματα παρατηρείται το πλεονέκτημα της χρήσης του προτεινόμενου υπερκείμενου δικτύου από τους μικρούς χρόνους καθυστέρησης και αριθμού βημάτων για την αποστολή ενός μηνύματος, χαρακτηριστικά που δεν αυξάνονται σημαντικά κατά την προσθήκη νέων κόμβων. Με τη χρήση της υπερκείμενης λογικής του Pastry, μπορούν να εκμεταλλευτούν τα πλεονεκτήματα της χρήσης του πίνακα δρομολόγησης στους κόμβους που υλοποιείται σε αυτό. Με αυτό τον τρόπο μπορούν να επιλεγούν οι πλησιέστεροι κόμβοι από αυτούς που έχουν κοινό πρόθεμα και έτσι εμφανίζονται μικρότεροι χρόνοι καθυστέρησης και αριθμού βημάτων για την μετάδοση ενός μηνύματος. Με τη χρήση του Chord, για τη δρομολόγηση ενός μηνύματος επιλέγονται μόνο οι κόμβοι με συνεχόμενα αναγνωριστικά, σε λογική διαδοχικών κόμβων, έτσι ώστε ένα μήνυμα να προωθείται μόνο προς μια κατεύθυνση.

12.4 Μεταβολή των δρομολογητών

Για τη λειτουργία του υπερκείμενου δικτύου με μεταβολή των δρομολογητών, έγιναν 3 προσομοιώσεις με σταθερό αριθμό τερματικών κόμβων ίσο με 500 και τον αριθμό των δρομολογητών να μεταβάλλεται από 100 κόμβους σε 300 και 500 κόμβους. Ενδεικτικά γραφήματα της καθυστέρησης και του αριθμού βημάτων είναι τα παρακάτω.



Εικόνα 32: Καθυστέρηση ως προς το χρόνο προσομοίωσης στη μεταβολή δρομολογητών



Εικόνα 33: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης στη μεταβολή δρομολογητών

Τα δεδομένα που προκύπτουν από τις προσομοιώσεις για την καθυστέρηση.

Κόμβοι δρομολογητές	Μέση καθυστέρηση
100	0.199465
300	0.205040
500	0.218303

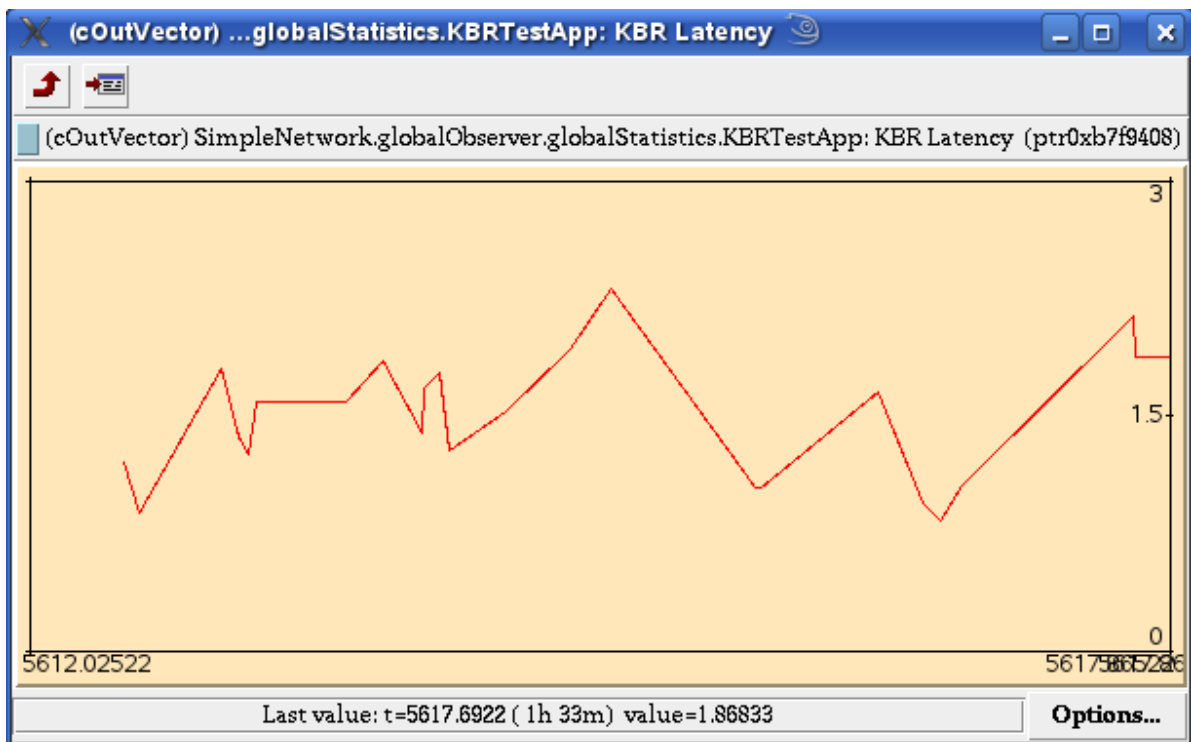
Για το μέσο αριθμό βημάτων προέκυψαν τα παρακάτω δεδομένα.

Κόμβοι δρομολογητές	Μέσος αριθμός βημάτων
100	1.682973
300	1.993646
500	2.197936

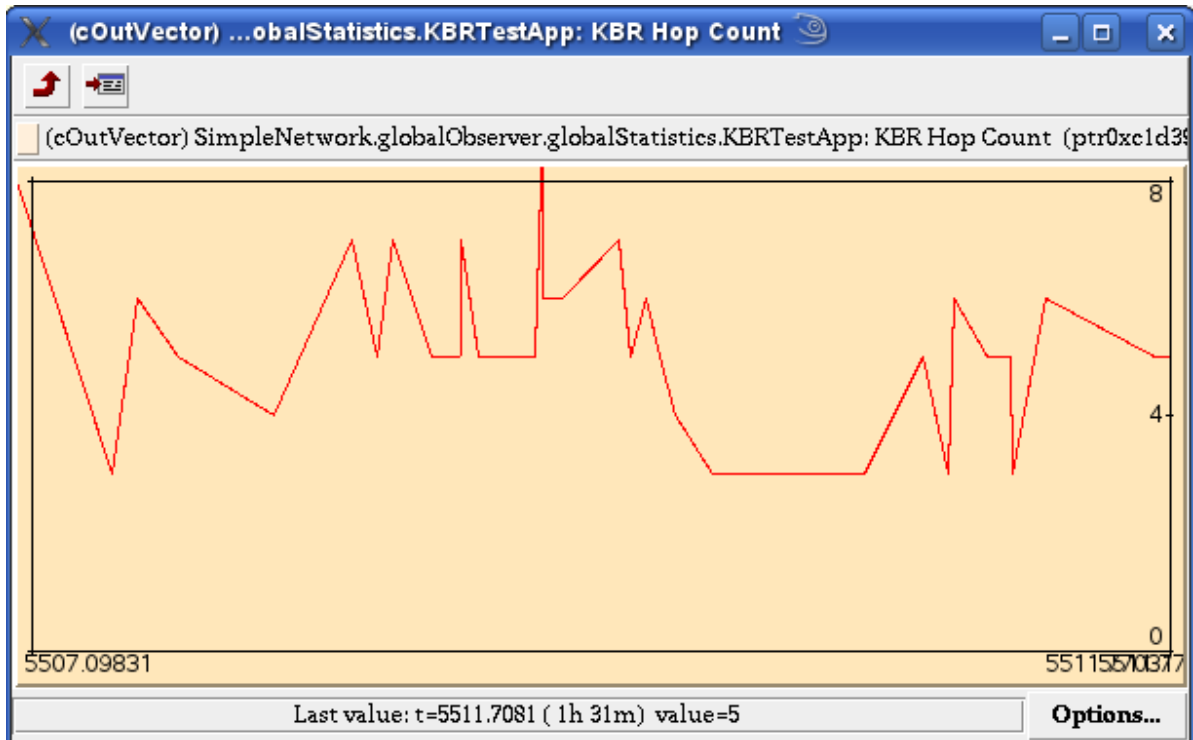
Για το Delivery Ratio τα αποτελέσματα είναι τα εξής.

Κόμβοι δρομολογητές	Delivery Ratio
100	1
300	0.999941
500	0.999947

Κατά τον ίδιο τρόπο με τη μεταβολή των PastryModules με τα ChordModules, και με σταθερό αριθμό τερματικών, προκύπτουν τα παρακάτω ενδεικτικά διαγράμματα.



Εικόνα 34: Καθυστέρηση ως προς το χρόνο προσομοίωσης στη μεταβολή δρομολογητών



Εικόνα 35: Αριθμός βημάτων ως προς το χρόνο προσομοίωσης στη μεταβολή δρομολογητών

Τα δεδομένα που προκύπτουν από τις προσομοιώσεις για την καθυστέρηση, το μέσο αριθμό βημάτων και το Delivery Ratio είναι τα παρακάτω.

Κόμβοι δρομολογητές	Μέση καθυστέρηση
100	1.070147
300	1.311109
500	1.446677

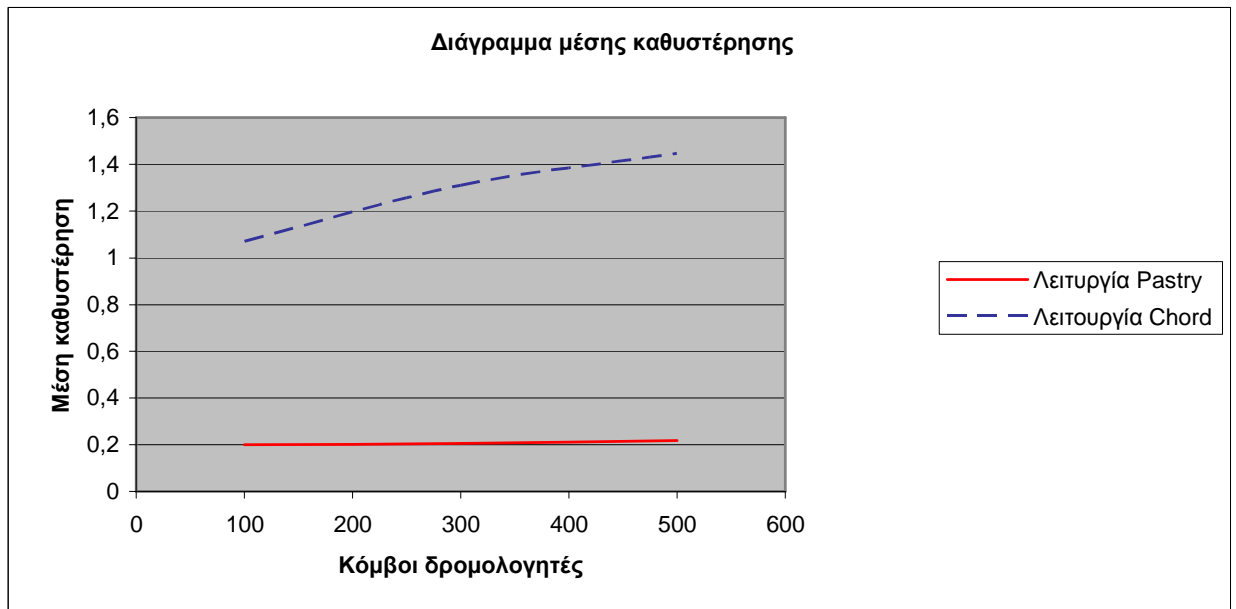
Για το μέσο αριθμό βημάτων προέκυψαν τα παρακάτω δεδομένα.

Κόμβοι δρομολογητές	Μέσος αριθμός βημάτων
100	4.058719
300	4.826714
500	5.226225

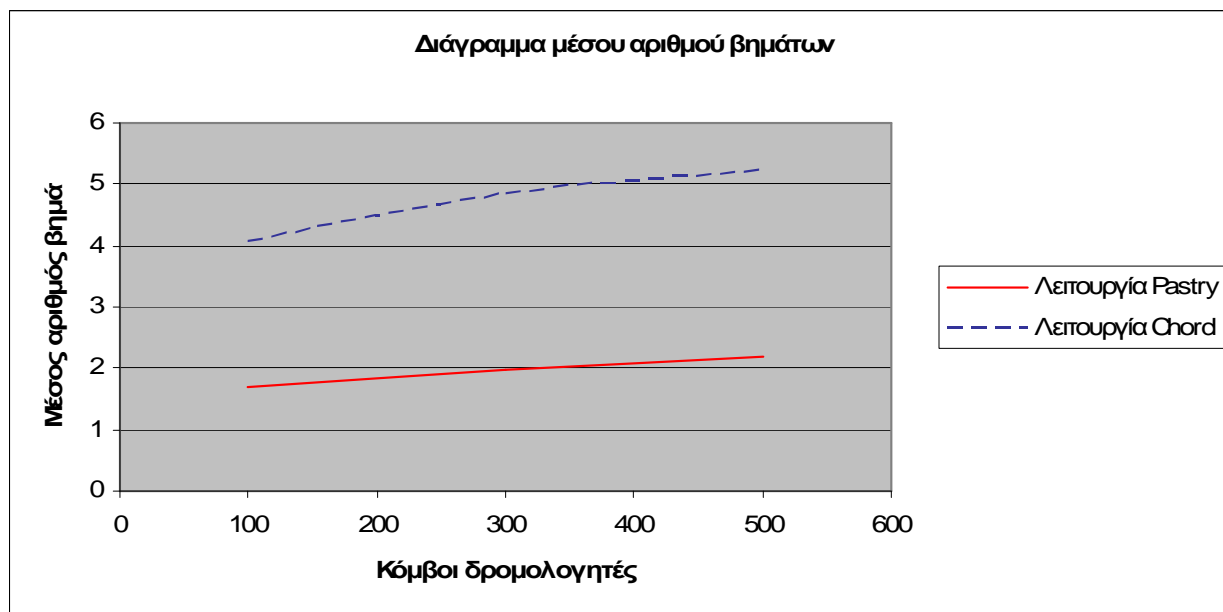
Για το Delivery Ratio τα αποτελέσματα είναι τα εξής.

Κόμβοι δρομολογητές	Delivery Ratio
100	0.999911
300	0.999613
500	0.999804

Σύμφωνα με τα αποτελέσματα κατά την προσομοίωση με μεταβαλλόμενο αριθμό κόμβων δρομολογητών στο προτεινόμενο υπερκείμενο δίκτυο προκύπτουν τα παρακάτω διαγράμματα για τη μέση καθυστέρηση και το μέσο αριθμό βημάτων.



Εικόνα 36: Σύγκριση μέσης καθυστέρησης στη μεταβολή των δρομολογητών



Εικόνα 37: Σύγκριση μέσου αριθμού βημάτων στη μεταβολή των δρομολογητών

Στο προτεινόμενο υπερκείμενο δίκτυο, στο οποίο η προώθηση των μηνυμάτων γίνεται στους κόμβους δρομολογητές, όταν αυξάνονται οι κόμβοι στους οποίους γίνεται η δρομολόγηση, απαιτείται μεγαλύτερος αριθμός βημάτων για την προώθηση των μηνυμάτων στον τελικό τους προορισμό και αυξάνεται η καθυστέρηση.

Η χρήση των πινάκων δρομολόγησης στο Pastry δίνει τη δυνατότητα να βρίσκει τις διαδρομές που οδηγούν σε μικρότερη καθυστέρηση σε σχέση με το πρωτόκολο Chord, για μεγαλύτερους αριθμούς των κόμβων δρομολογητών. Έτσι με την αύξηση των δρομολογητών, η δρομολόγηση του Pastry μπορεί να διατηρήσει με μεγαλύτερη ευκολία τις αρχικές τιμές στην καθυστέρηση.

Από τα διαγράμματα φαίνεται η ιδιότητα του προτεινόμενου υπερκείμενου δικτύου με το πρωτόκολο Pastry να επιτυγχάνει μικρότερες τιμές για τη μέση καθυστέρηση και το μέσο αριθμό βημάτων σε σχέση με το Chord, ακόμα και στην αύξηση των δρομολογητών. Με το Chord αυξάνεται περισσότερο η καθυστέρηση και ο αριθμός βημάτων με την αύξηση των δρομολογητών. Με τη χρήση του Pastry, και των πινάκων

δρομολόγησης στην προώθηση των μηνυμάτων επιλέγονται οι πλησιέστεροι κόμβοι από αυτούς που έχουν κοινό πρόθεμα και έτσι εμφανίζονται μικρότεροι χρόνοι καθυστέρησης και αριθμού βημάτων για την μετάδοση ενός μηνύματος. Με τη χρήση του Chord, για τη δρομολόγηση ενός μηνύματος επιλέγονται μόνο οι κόμβοι με συνεχόμενα αναγνωριστικά, σε λογική διαδοχικών κόμβων, έτσι ώστε ένα μήνυμα να προωθείται μόνο προς μια κατεύθυνση, ενώ μπορεί να χρειαστεί ένα μήνυμα να περάσει από όλους τους κόμβους μέχρι να φτάσει στον τελικό του προορισμό. Από τα αποτελέσματα επιβεβαιώνονται τα πλεονεκτήματα της χρήσης του πρωτοκόλου Pastry για την εφαρμογή ενός υπερκείμενου δικτύου με τη χρήση των σκανδαλών.

12.5 Μεταβολή ρυθμού δεδομένων

Στην προσομοίωση του προτεινόμενου υπερκείμενου δικτύου ο ρυθμός δεδομένων από τις ρυθμίσεις του OverSim είναι 10 Mbps. Στη συνέχεια θα μελετηθεί η λειτουργία του προτεινόμενου υπερκείμενου δικτύου για ρυθμό δεδομένων 512Kbps, όπως υπάρχει στις ρυθμίσεις του OverSim. Τα αποτελέσματα για προσομοιώσεις εμφανίζονται στη συνέχεια..

Προσομοίωση με μεγάλο αριθμό τερματικών κόμβων

Για τη λειτουργία με μεγάλο αριθμό τερματικών κόμβων τα αποτελέσματα για τη μέση καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
200	0.313637
400	0.311571
600	0.313641
800	0.314149
1000	0.313341

Τα αποτελέσματα για το μέσο αριθμό κόμβων είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
200	1.696373
400	1.685666
600	1.699302
800	1.696989
1000	1.695610

Για το Delivery Ratio τα αποτελέσματα είναι τα παρακάτω.

Τερματικοί κόμβοι	Delivery Ratio
200	1
400	0.999914
600	0.999909
800	0.999909
1000	1

Προσομοίωση με RandomChurn

Για τη λειτουργία του RandomChurn τα αποτελέσματα για τη μέση καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
100	0.274575
200	0.275587
400	0.275338

Τα αποτελέσματα για το μέσο αριθμό βημάτων είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
100	1.420531
200	1.426796
400	1.425386

Τα αποτελέσματα για το Delivery Ratio κατά τη λειτουργία RandomChurn και την εφαρμογή του πρωτοκόλλου Pastry, είναι τα παρακάτω.

Τερματικοί κόμβοι	Delivery Ratio
100	0.999833
200	0.999835
400	1

Προσομοίωση με LifetimeChurn

Για τη λειτουργία του LifetimeChurn τα αποτελέσματα για τη μέση καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
100	0.278390
200	0.284383
400	0.270287

Τα αποτελέσματα για το μέσο αριθμό βημάτων είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
100	1.429219
200	1.409994
400	1.429274

Τα αποτελέσματα για το Delivery Ratio κατά τη λειτουργία LifetimeChurn και την εφαρμογή του πρωτοκόλλου Pastry, είναι τα παρακάτω.

Τερματικοί κόμβοι	Delivery Ratio
100	1
200	1
400	1

Προσομοίωση σε λειτουργία πολυεκπομπής

Για τη λειτουργία της πολυεκπομπής τα αποτελέσματα για τη μέση καθυστέρηση είναι τα παρακάτω.

Τερματικοί κόμβοι	Μέση καθυστέρηση
100	0.284870
200	0.288361
400	0.284190

Τα αποτελέσματα για το μέσο αριθμό βημάτων είναι τα παρακάτω.

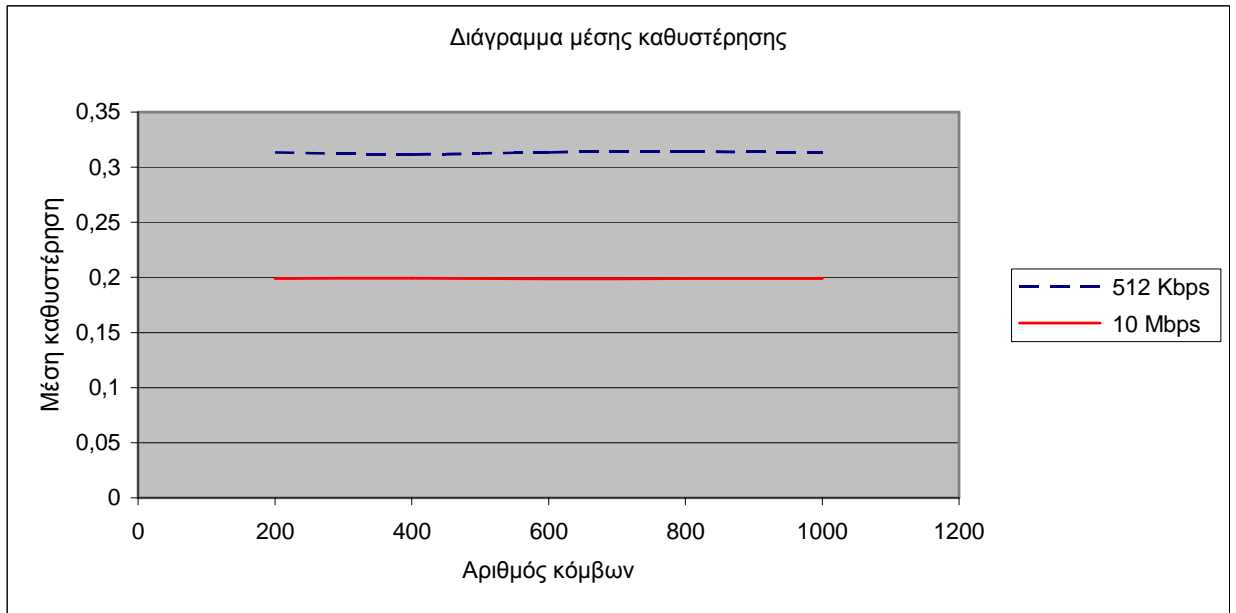
Τερματικοί κόμβοι	Μέσος αριθμός βημάτων
100	1.439359
200	1.450185
400	1.434707

Τα αποτελέσματα για το Delivery Ratio κατά την πολυεκπομπή και την εφαρμογή του πρωτοκόλου Pastry, είναι τα παρακάτω.

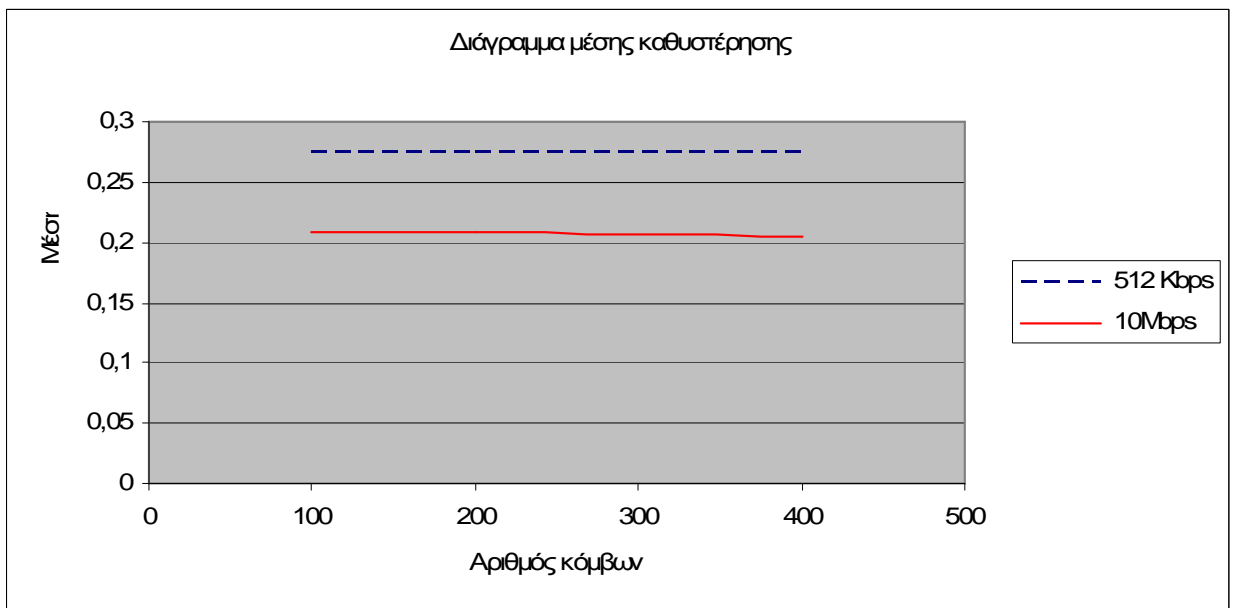
Τερματικοί κόμβοι	Delivery Ratio
100	0.999835
200	1
400	1

Από αυτά τα αποτελέσματα παρατηρείται ότι για την αλλαγή του ρυθμού μετάδοσης των δεδομένων ο μέσος αριθμός βημάτων παραμένει σχεδόν σταθερός, όπως και το Delivery Ratio. Παρατηρείται αλλαγή στη μέση καθυστέρηση και συγκεκριμένα αύξηση τόσο στην προσομοίωση με μεγάλο αριθμό κόμβων, όσο και στη λειτουργία με RandomChurn και με πολυεκπομπή. Αυτό μπορεί να οφείλεται στο γεγονός ότι για την ολοκληρωμένη δρομολόγηση των δεδομένων απαιτείται περισσότερος χρόνος, λόγω του μικρότερου ρυθμού μετάδοσης.

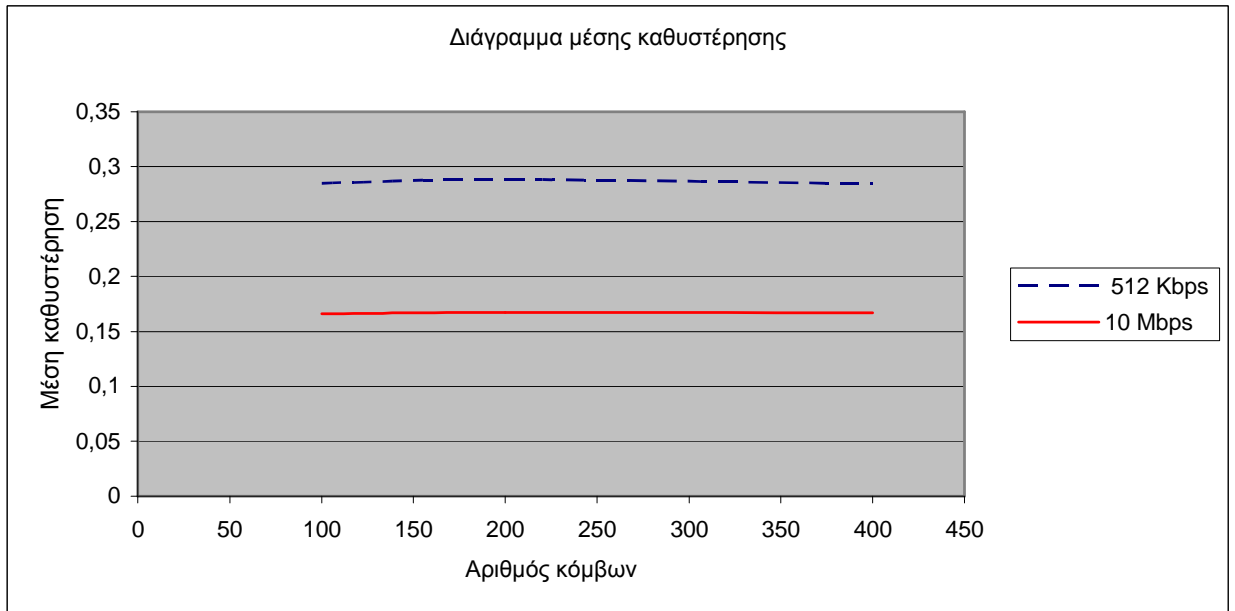
Στο παρακάτω διάγραμμα απεικονίζεται αυτή η διαφορά στη μέση καθυστέρηση για μεγάλο αριθμό τερματικών κόμβων, σε λειτουργία RandomChurn και πολυεκπομπής.



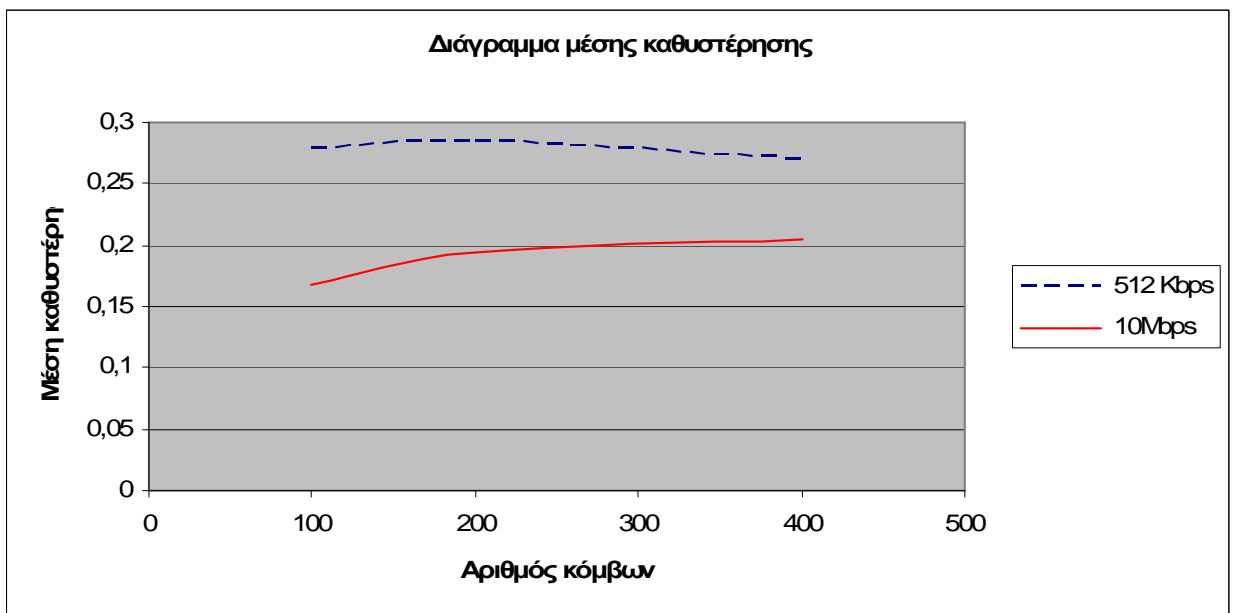
Εικόνα 38: Διάγραμμα μέσης καθυστέρησης για διαφορετικούς ρυθμούς δεδομένων με μεγάλο αριθμό τερματικών κόμβων



Εικόνα 39: Διάγραμμα μέσης καθυστέρησης για διαφορετικούς ρυθμούς δεδομένων σε λειτουργία RandomChurn



Εικόνα 40: Διάγραμμα μέσης καθυστέρησης για διαφορετικούς ρυθμούς δεδομένων σε λειτουργία LifetimeChurn



Εικόνα 41: Διάγραμμα μέσης καθυστέρησης για διαφορετικούς ρυθμούς δεδομένων σε λειτουργία πολεκπομπής

Σε σύγκριση με τη μεταβολή του αριθμού των δρομολογητών, σε αυτή την περίπτωση παρατηρείται ότι δεν αυξάνεται ο αριθμός των βημάτων που απαιτούνται

κατά τη δρομολόγηση, κάτι που αναμένεται αφού ο αριθμός των δρομολογητών παραμένει σταθερός. Ακόμα, η μέση καθυστέρηση στη μεταβολή του ρυθμού δεδομένων αυξάνεται σε σχεδόν σταθερή τιμή, ενώ στη μεταβολή του αριθμού των δρομολογητών παραμένει σχεδόν σταθερή με την αρχική τιμή.

13. Επίλογος

Σκοπός αυτής της διπλωματικής ήταν η μελέτη της λειτουργίας της δρομολόγησης σε υπερκείμενα δίκτυα. Περιγράφηκε η λογική της λειτουργίας των υπερκείμενων δικτύων και αναλύθηκαν συγκεκριμένα τα υπερκείμενα δίκτυα CAN, Chord, Pastry, Tapestry, Mithos. Τα παραπάνω υπερκείμενα δίκτυα χρησιμοποιούν τους κατανεμημένους πίνακες ανάμειξης για τη δρομολόγηση μηνυμάτων. Έγινε σύγκριση μεταξύ των παραπάνω υπερκείμενων δικτύων για την εξαγωγή συμπερασμάτων σε θεωρητικό επίπεδο. Στη συνέχεια προτάθηκε ένα υπερκείμενο δίκτυο το οποίο χρησιμοποιεί τις σκανδάλεις για την προώθηση των μηνυμάτων με μια τροποποιημένη εκδοχή του υπερκείμενου δικτύου Pastry. Παρατηρήθηκαν τυχόν ομοιότητες ή διαφορές με τα προηγούμενα υπερκείμενα δίκτυα. Τέλος έγινε προσομοίωση του προταθέντος υπερκείμενου δικτύου με το πρόγραμμα OverSim, και παρατηρήθηκαν μερικά πλεονεκτήματα της υλοποίησης του υπερκείμενου δικτύου αυτού στην εφαρμογή του, για τις περιπτώσεις μεγάλης κλίμακας δικτύου, εξόδου- τυχαίας επανεισαγωγής κόμβων, εξόδου- σταθερής επανεισαγωγής κόμβων, πολυεκπομπής, μεταβολής δρομολογητών και ρυθμού δεδομένων του δικτύου.

Βιβλιογραφία

1. «A Scalable Content-Addressable Network» : Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp
2. «Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications» : Ian Stoica, Robert Morris, David Karger, M.Frans Kaashoek, Hari Balakrishnan
3. «Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems» : Antony Rowstron and Peter Druschel
4. «Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing» : Ben Y.Zhao, John Kubiawicz, and Anthony D.Joseph
5. «Efficient Topology-Aware Overlay Network» : Marcel Waldvogel, Roberto Rinaldi
6. «Internet Indirection Infrastructure» : Ian Stoica, Daniel Adkins, Shelley Zhuang, Scott Shenker, Sonesh Surana
7. «Resilient Overlay Networks» : David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris
8. «Peer-to-peer Overlay Networks: A Survey» : Chonggang Wang, Bo Li
9. «A Routing Underlay for Overlay Networks» : Akihiro Nakao, Larry Peterson and Andy Bavier
10. «A Framework for Structured Peer-to-Peer Overlay Networks» : Luc Onana Alima, Ali Ghodsi, and Seif Haridi

11. «Exploiting network proximity in peer-to-peer overlay networks» : Miguel Castro, Peter Druschel, Y.Charlie Hu, Antony Rowstron
12. «The essence of P2P: A reference architecture for overlay networks» : Karl Aberer, Luc Onana Alima, Ali Ghodsi, Sarunas Girdzijauskas, Seif Haridi, Manfred Hauswirth
13. «Peer-to-Peer Simulators» : Mark Baker and Rahim Lakhoo
14. «OverSim: A Flexible Overlay Network Simulation Framework» : Ingmar Baumgart, Bernahard Heep, Stephan Krause

Ακόμα χρησιμοποιήθηκαν περιεχόμενα σελίδων του διαδικτύου για την συγγραφή της διπλωματικής εργασίας.

15. <http://www.omnetpp.org>

16. <http://www.oversim.org>