ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΙΤΛΟΣ: «*Υλοποίηση μηχανισμού αποθήκευσης σε δίκτυα βασισμένα στο περιεχόμενο*»

ΕΥΘΥΜΙΑΔΗΣ ΑΡΧΑΓΓΕΛΟΣ

ΕΠΙΒΛΕΠΩΝ:

Τασιούλας Λέανδρος , Καθηγητής

Βόλος 2012

Περιεχόμενα

Ευχαριστίες

Ευχαριστώ πάνω απ’ όλα την οικογένειά μου για την οικονομική στήριξη και την αμέριστη συμπαράσταση που μου παρείχαν όλα αυτά τα χρόνια.

Θερμές ευχαριστίες στους καθηγητές μου κ. Λέανδρο Τασιούλα και κ.Δημήτριο Κατσαρό που δέχτηκαν να επιβλέψουν τη διπλωματική μου εργασία.

Ιδιαίτερα θέλω να ευχαριστήσω τον Πάρη Φλέγγα για την όλη καθοδήγηση του, για τις συμβουλές και τη βοήθεια που μου παρείχε καθ'όλη τη διάρκεια της εργασίας αυτής.

Επίσης , ευχαριστώ τον Βασίλη Σούρλα για τις σημαντικές παρεμβάσεις που έκανε.

Τέλος, η διπλωματική μου εργασία είναι αφιερωμένη στους φίλους μου, που με αγαπούν και με ανέχονται τόσα χρόνια.

«Στους γονείς μου οφείλω το ζην στους δασκάλους το ευ ζην»
Μέγας Αλέξανδρος

ΚΕΦΑΛΑΙΟ 1-ΕΙΣΑΓΩΓΗ

1.1.Εισαγωγη

Η βασισμένη στο περιεχόμενο δικτύωση δημοσιεύσεων/συνδρομών (publish/subscribe) είναι ένα ευέλικτο επικοινωνιακό μοντέλο που ικανοποιεί τις απαιτήσεις της διανομής περιεχομένου στο Internet, όπου οι πληροφορίες πρέπει να αντιμετωπιστούν με σημασιολογικές ιδιότητες αντί ταυτότητες προέλευσης και προορισμού. Σε τρέχουσες εφαρμογές publish/ subscribe δικτύων, τα μηνύματα δεν αποθηκεύονται και μόνο ενεργοί subscribers λαμβάνουν μηνύματα δημοσίευσης. Ωστόσο, σε ένα δυναμικό σενάριο, όπου χρήστες εισέρχονται στο δίκτυο σε διάφορες περιπτώσεις, ένας χρήστης μπορεί να ενδιαφέρεται για περιεχόμενο που δημοσιεύθηκε πριν από το χρόνο εγγραφής του. Σε αυτή την εργασία, προτείνουμε ένα νέο αλγόριθμο τοποθέτησης αποθηκών και ανάθεσης αντιγράφων που διαφοροποιεί τις κατηγορίες του περιεχομένου με βάση τη δημοτικότητά τους και ελαχιστοποιεί την καθυστέρηση ανταπόκρισης των πελατών και της συνολικής κίνησης του δικτύου. Η απόδοση του προτεινόμενου αλγορίθμου τοποθέτησης και ανάθεσης αντιγράφων και ο προτεινόμενος μηχανισμός αποθήκευσης αξιολογείται μέσω προσομοιώσεων και δίνονται ιδέες για μελλοντικές εργασίες.

1.2. Κίνητρο

Το σημαντικότερο κίνητρο που μας ώθησε να ασχοληθούμε με το συγκεκριμένο θέμα είναι ότι με την αποθήκευση διαδικτυακού περιεχομένου μέσα στο ίδιο δίκτυο θα πετύχουμε τη μείωση του χρόνου παράδοσης των δεδομένων που ζητούν οι χρήστες.
Επίσης, πετυχαίνουμε μείωση στο συνολικό φόρτο στο δίκτυο. Συνεπώς, ο χρήστης έχει τη δυνατότητα να βρει τα δεδομένα σε ενδιάμεσους κόμβους-αποθήκες οι οποίοι είναι πιο κοντά σε αυτόν. Με τη χρήση των αποθηκών μειώνεται παράλληλα και ο φόρτος του server ,ο οποίος είναι η αρχική πηγή των δεδομένων, αφού πλέον δεν δέχεται όλες τις αιτήσεις παρά μόνο τις κοντινότερες.

Εξαιτίας του γεγονότος ότι τα δίκτυα βασισμένα στο περιεχόμενο αποτελούν μια σχετικά νέα αρχιτεκτονική δικτύου, ο αλγόριθμος ο οποίος θα παρουσιάσουμε στην παρούσα εργασία είναι μια πρόταση που μπορεί να επιφέρει αρκετά οφέλη στο χώρο της δικτύωσης και χρήζει περαιτέρω έρευνας .

1.3 Ορισμός του Προβλήματος

Στην εργασία μας, υποθέτουμε ένα content-centric pub/sub δίκτυο με αυθαίρετη τοπολογία αναπαριστόμενη από ένα γράφημα G = (N, E).
Τ διαφορετικά θέματα θα πρέπει να αποθηκευθούν σε Μ αποθήκες, όπου κάθε αποθήκη έχει την δυνατότητα να αποθηκεύσει L θέματα. Κάθε θέμα t ∈ Τ θα πρέπει να αντιγραφεί kt φορές. Οι αιτήσεις για τα θέματα παράγονται σε διάφορους κόμβους και προκαλούν τη μεταφορά του αιτούμενου στοιχείου από μια αποθήκη, στον κόμβο που έγινε η αίτηση. Ο προτεινόμενος μηχανισμός αποτελείται από δύο φάσεις, τυπικές σε οποιαδήποτε εργασία διαχείρισης δικτύου, ήτοι οι φάσεις Σχεδιασμού και Ανάθεσης. Στη φάση του Σχεδιασμού, ο προτεινόμενος μηχανισμός επιλέγει Μ σημεία από τους Ν κόμβους του δικτύου για να τοποθετήσει αποθήκες, ενώ στη φάση της Ανάθεσης, κάθε θέμα t ανατίθεται σε ακριβώς kt διαφορετικές αποθήκες με στόχο να ελαχιστοποιηθεί ο συνολικός φόρτος στο δίκτυο.

Σε γενικές γραμμές, σε ένα πραγματικό pub/sub δίκτυο διανομής περιεχομένου ο σχεδιασμός των αποθηκών (Planning) αλλάζει σπάνια, δεδομένου ότι απαιτεί την ανακατανομή των αποθηκών μεταξύ των κόμβων του δικτύου. Από την άλλη πλευρά, η ανάθεση των θεμάτων(Replica Assignment) είναι πιο ευέλικτη και ο πάροχος CDN είναι σε θέση να αναθέσει εκ νέου τα θέματα μεταξύ των αποθηκών, όταν τα μοντέλα τοπικότητας και δημοτικότητας μεταβάλλονται με τέτοιο τρόπο έτσι ώστε η απόδοση του δικτύου να υποβαθμίζεται με την υπάρχουσα διαμόρφωση. Φυσικά, μια νέα ανάθεση απαιτεί τον υπολογισμό των νέων θέσεων για κάθε θέμα και τη μεταφορά των θεμάτων σε αυτές τις θέσεις, αλλά όπως φαίνεται αργότερα στην ανάλυση των επιδόσεων, είναι ένας αποτελεσματικός τρόπος για να διατηρήσει τις επιδόσεις του συστήματος σε υψηλά επίπεδα χωρίς τον εκ νέου σχεδιασμό όλου του CDN δικτύου.

Η χωρητικότητα του κάθε server αναπαραγωγής συνήθως αναφέρεται σε Tbytes αλλά για λόγους απλότητας υποθέτουμε εδώ ότι ο αριθμός των μηνυμάτων είναι ίδιος για κάθε θέμα και τα μηνύματα είναι του ίδιου μεγέθους, αφήνοντας για τις μελλοντικές εργασίες την περίπτωση των διαφόρων μεγεθών. Εναλλακτικά, σε κάθε στιγμιότυπο του συστήματος στο δίκτυο, υπάρχει ο ίδιος αριθμός των μηνυμάτων για κάθε θέμα. Η παράμετρος SC ή L είναι ένας περιορισμός που εισήχθη από τους παρόχους αποθήκευσης και αναφέρεται στη μέγιστη αποθηκευτική δυνατότητα της κάθε αποθήκης στο δίκτυο. Από την άλλη πλευρά, η παράμετρος kt (βαθμός αναπαραγωγής του κάθε θέματος) είναι ένας περιορισμός που εισήχθη από τον πάροχο αποθήκευσης και αναφέρεται στον αριθμό των αντιγράφων που ο πάροχος περιεχομένου είναι πρόθυμος να πληρώσει. Τέλος, η παράμετρος M αναφέρεται στον αριθμό των αποθηκών που θα πρέπει να εγκαταστήσει στο δίκτυο ένας πάροχος αποθήκευσης για να εξυπηρετήσει τις ανάγκες αποθήκευσης των θεμάτων.

1.4 Δομή της Αναφοράς

Στην εργασία αυτή, υποθέτουμε ότι τα μηνύματα είναι ταξινομημένα ανάλογα με την τάξη τους(θέμα) και εμείς:

- Προτείνουμε ένα νέο αλγόριθμο για την επιλογή M σημείων αποθήκευσης μεταξύ των N brokers του δικτύου (M <N) με βάση: α) την τοπικότητα και τη δημοτικότητα των ενδιαφερόντων για κάθε θέμα, β) το στοχευόμενο βαθμό αναπαραγωγής του κάθε θέματος (ως βαθμό αναπαραγωγής ονομάζουμε τον αριθμό των αντιγράφων kt ($1 \leq kt \leq M$) του θέματος $t \in T$ μεταξύ των αποθηκών, η οποία είναι ανάλογη με τη δημοτικότητά του) και γ) τη χωρητικότητα αποθήκης (περιορισμός) L κάθε αποθήκης.

- Προτείνουμε έναν μηχανισμό για την ανάθεση των αντιγράφων κάθε θέματος t E T ανάμεσα στις επιλεγμένες αποθήκες

- Αξιολογούμε μέσω προσομοιώσεων τον σχεδιασμό μας για τον μηχανισμό αποθήκευσης και τον νέο αλγόριθμο τοποθέτησης και ανάθεσης αντιγράφων

Ο στόχος του προγράμματος μας είναι η ελαχιστοποίηση του συνολικού φόρτου όλων των τάξεων περιεχομένου του Δικτύου, υπό την εγκατάσταση του ελάχιστου αριθμού αποθηκών και δεδομένου ότι οι διακομιστές-αποθήκες έχουν περιορισμούς αποθήκευσης.Το υπόλοιπο της εργασίας έχει οργανωθεί ως εξής.

Στο Κεφάλαιο 2 δίνεται μια σύντομη περιγραφή σχετικών εργασιών για την αποθήκευση σε pub/sub αρχιτεκτονικές, καθώς και μια περιγραφή των CDNs και του Web Caching. **Στο Κεφάλαιο 3**, περιγράφουμε σύντομα την pub / sub αρχιτεκτονική ενώ στο **Κεφάλαιο 4** παρουσιάζουμε το νέο αλγόριθμο για την επιλογή της τοποθεσίας αποθηκών και την ανάθεση αντιγράφων του περιεχομένου. **Στο Κεφάλαιο 5** παρουσιάζουμε την υλοποίηση του αλγορίθμου καθώς επίσης και τα εργαλεία που χρησιμοποιήσαμε για αυτήν.
Το Κεφάλαιο 6 είναι αφιερωμένο στην αξιολόγηση των επιδόσεων μέσω προσομοιώσεων. Τέλος, τερματίζουμε την εργασία δίνουμε ιδέες για τις μελλοντικές εργασίες στο **Κεφάλαιο 7.**

ΚΕΦΑΛΑΙΟ 2 - Σχετικές Εργασίες

Η χρήση του Διαδικτύου έχει αλλάξει σημαντικά τα τελευταία χρόνια από έναν μηχανισμό επιμερισμού πόρων μεταξύ δύο hosts σε μηχανισμό διανομής και ανάκτησης περιεχομένου. Σε αυτό το μεταβαλλόμενο περιβάλλον, το pub/sub παράδειγμα γίνεται ολοένα και πιο δημοφιλές για την πρόσβαση και τη διάδοση περιεχομένου. Συγκεκριμένα, υπάρχουν αρκετές ερευνητικές προσπάθειες που αναπτύσσουν μια επικαλυπτόμενη υπηρεσία ειδοποίησης με γεγονότα όπως το Gryphon της IBM [1] ,το Siena [2] ,το Elvin [3] και το REDS [4], τα οποία εκτελούν την pub/sub αρχιτεκτονική. Επιπλέον, υπάρχουν επίσης πολλές ερευνητικές προσπάθειες με στόχο τη μετάβαση από τη προσανατολισμένη σε hosts στην προσανατολισμένη στο περιεχόμενο δικτύωση, όπως το CCN [5] , το DONA [6] ,το PSIRP [7],το PURSUIT [8] και το 4WARD [9], οι οποίες επιχειρούν να κατονομάσουν δεδομένα/περιεχόμενο, αντί να κατονομάζουν hosts για να επιτύχουν επεκτασιμότητα, ασφάλεια και απόδοση.

Τα διάφορα pub/sub μοντέλα κατατάσσονται ανάλογα με τη σημασιολογία της γλώσσας συνδρομής. Μεταξύ των pub/sub μοντέλων, τα πιο γνωστά είναι αυτά που βασίζονται στο θέμα (topic-based pub/sub), που επιτρέπουν στους καταναλωτές πληροφορίας να εγγράφονται σε ένα σύνολο προκαθορισμένων θεμάτων οργανωμένα σε μια ιεραρχία, και αυτά που βασίζονται στο περιεχόμενο (content-based pub/sub),που υποστηρίζουν συνδρομές που ακολουθούν ένα σχήμα χαρακτηριστικού/τιμής(attribute/value).

Στον τομέα της προσωρινής αποθήκευσης(caching/storing) και αναπαραγωγής στα content-centric pub/sub δίκτυα [10], ένα pub/sub σύστημα ανάκτησης ιστορικών δεδομένων προτείνεται, όπου βάσεις δεδομένων συνδέονται με διάφορους brokers, ο καθένας των οποίων σχετίζεται με την αποθήκευση ένος θέματος. Στο [10], κάθε μήνυμα αποθηκεύεται μόνο μία φορά χωρίς να έχουν εξεταστεί στρατηγικές τοποθέτησης, ενώ δεν υπάρχει περιγραφή του μηχανισμού για την ανάκτηση των αποθηκευμένων δεδομένων. Επιπλέον, στο [11], έχει εισαχθεί ένα ευκαιριακό σχήμα caching για pub/sub δίκτυα όπου κάθε broker του δικτύου είναι ένα δυνητικό σημείο caching, ενώ μια πρώτη προσπάθεια με έναν off-line αλγόριθμο αντιγραφής σε topic-based pub/sub δίκτυα παρουσιάζεται στο [12].

Από την άλλη πλευρά το πρόβλημα τοποθέτησης, στο πλαίσιο της CDNs, είναι ένα διερευνημένο διεξοδικά πρόβλημα. Ιδιαίτερα στο [13] και [14] οι συγγραφείς προσεγγίζουν το πρόβλημα τοποθέτησης με την παραδοχή ότι οι υποκείμενες τοπολογίες δικτύων είναι δέντρα. Αυτή η απλή προσέγγιση επιτρέπει τους συγγραφείς να αναπτύξουν βέλτιστους αλγορίθμους, αλλά θεωρούν το πρόβλημα της τοποθέτησης αντιγράφων για μόνο ένα διακομιστή προέλευσης. Το πρόβλημα τοποθέτησης είναι στην πραγματικότητα ένα NP-hard πρόβλημα [15], αλλά υπάρχει μια σειρά από μελέτες [16]-[21] , όπου επιδιώκεται μια κατά προσέγγιση λύση . Το έργο τους είναι επίσης γνωστό ως εντοπισμός δικτύου ή διαχωρισμός και περιλαμβάνει την βέλτιστη τοποθέτηση k εγκαταστάσεων υπηρεσιών σε ένα δίκτυο N κόμβων με στόχο την ελαχιστοποίηση ενός συγκεκριμένου στόχου. Σε ορισμένες περιπτώσεις, μπορεί να αποδειχθεί ότι το πρόβλημα αυτό μειώνει στο γνωστό πρόβλημα k-median. Περισσότεροι αλγόριθμοι τοποθέτησης έχουν προταθεί στο [15]. Ειδικότερα , οι συγγραφείς διατυπώσουν το πρόβλημα σαν ένα συνδυαστικό πρόβλημα βελτιστοποίησης, αποδεικνύουν ότι αυτό είναι NP-hard και αναπτύσσουν και συγκρίνουν τέσσερις φυσικά ευρετικούς αλγορίθμους . Βρήκαν ότι τα καλύτερα αποτελέσματα επιτυγχάνονται με ευρετικές που έχουν όλες τις αποθήκες να συνεργάζονται στη λήψη των αποφάσεων αντιγραφής. Τέλος, στο [22] οι συγγραφείς συστήνουν ένα πλαίσιο για την αξιολόγηση των αλγορίθμων τοποθέτησης.

2.1 Δίκτυα Δημοσιεύσεων/Συνδρομών

Τα συστήματα publish / subscribe (pub/sub) (βασισμένα στο θέμα ή βασισμένα στο περιεχόμενο) είναι οργανωμένα σαν μια συλλογή από αυτόνομα συστατικά, δηλαδή τους πελάτες και τους αποστολείς γεγονότων.

Οι πελάτες συμπεριφέρονται είτε ως εκδότες, δημοσιεύοντας νέα γεγονότα στο δίκτυο, είτε ως συνδρομητές, με την εγγραφή στις τάξεις των γεγονότων που τους ενδιαφέρουν. Οι αποστολείς γεγονότων (ή event brokers ή απλά brokers) από την άλλη πλευρά, είναι υπεύθυνοι να συλλέγουν συνδρομές και να προωθούν δημοσιεύσεις σε ενδιαφερόμενους συνδρομητές.

Στα pub/sub δίκτυα, η επιλογή ενός μηνύματος καθορίζεται αποκλειστικά από τον πελάτη, ο οποίος χρησιμοποιεί εκφράσεις (φίλτρα) που επιτρέπουν εξελιγμένο ταίριασμα με το περιεχόμενο γεγονότος. Στις τρέχουσες pub/sub υλοποιήσεις, κάθε γεγονός είναι εγγυημένο να φθάσει σε όλους τους ενδιαφερόμενους συνδρομητές, εφόσον οι συνδρομές τους είναι γνωστές στο δίκτυο τη στιγμή της δημοσίευσης, υποθέτοντας σταθερή τοπολογία και όχι υπερχειλίσεις αναμονής.

Ωστόσο, σε ένα κατανεμημένο δυναμικό περιβάλλον, οι πελάτες εισέρχονται στο δίκτυο και εξέρχονται από αυτό με την πάροδο του χρόνου, και είναι πιθανό ένας συνδρομητής να εισέρθει στο δίκτυο μετά τη δημοσίευση ενός ενδιαφέροντος μήνυματος. Στα τρέχοντα pub/sub συστήματα, δεν είναι δυνατόν για ένα νέο συνδρομητή να ανακτήσει ήδη δημοσιευμένα μηνύματα που ταιριάζουν στη συνδρομή του. Ως εκ τούτου, επιτρέποντας την ανάκτηση του παλιού δημοσιευμένου περιεχομένου μέσω της αποθήκευσης, είναι ένα από τα πιο προκλητικά προβλήματα στην pub/sub δίκτυα. Οι Servers διανομής περιεχομένου («υποκατάστατοι servers" στο CDNs ή απλά "αποθήκες" σε αυτήν την εργασία), αναπαράγουν ολόκληρο το περιεχόμενο ενός συγκεκριμένου server και στοχεύουν να επιταχύνουν την παράδοση του περιεχομένου μειώνοντας το φορτίο στους διακομιστές(servers) προέλευσης και το ίδιο το δίκτυο. Όταν ένας πελάτης ενδιαφέρεται για κάποιες πληροφορίες από ένα δεδομένο διακομιστή, η αίτηση του ανακατευθύνεται σε μια από τις υπάρχουσες αποθήκες (π.χ., την πιο κοντινή ή αυτό που ικανοποιεί άλλα κριτήρια, όπως το φόρτο της υποψήφιας αποθήκης).

Αφού οι αποθήκες εξυπηρετούν μόνο ένα μέρος του συνόλου των αιτήσεων και τοποθετούνται πιο κοντά στον πελάτη, οι πελάτες εξυπηρετούνται γρηγορότερα. Ένα αίτημα του πελάτη ανακατευθύνεται σε μια αποθήκη μόνο αν αυτή η αποθήκη είναι ένα αντίγραφο του στοχευμένου διακομιστή, διαφορετικά η αίτηση κατευθύνεται και εξυπηρετείται από τον ίδιο το διακομιστή.

2.1.1 Τύποι Δικτύων Δημοσιεύσεων/Συνδρομών

Όπως προαναφέραμε, υπάρχουν δύο βασικές κατηγορίες συστημάτων-δικτύων δημοσιεύσεων/συνδρομών [28] :
(i) αυτά που είναι βασισμένα σε ομάδες (group-based) ή σε θεματικές ενότητες (topic-based)
(ii) αυτά που είναι βασισμένα στο ίδιο το περιεχόμενο (content-based).

Η πρώτη γενιά συστημάτων δημοσιεύσεων/συνδρομών αποτελούνταν από group-based συστήματα [29][30][31]. Σε αυτά υπήρχε η έννοια των groups που ήταν ορισμένα κατάλληλα από το σύστημα. Οι συνδρομητές, δήλωναν τα ενδιαφέροντα τους με το να εγγράφονται σε ένα ή περισσότερα groups. Στη συνέχεια οι εκδότες διοχέτευαν τα μηνύματα τους στο κατάλληλο με τη πληροφορία που περιείχαν group, με αποτέλεσμα αυτά τελικά να φτάνουν στους τελικούς συνδρομητές. Μία ελαφρώς διαφορετική προσέγγιση, αποτέλεσαν αργότερα τα συστήματα δημοσιεύσεων/συνδρομών βασισμένα σε θεματικές ενότητες (topics). Τα μηνύματα πλέον χαρακτηρίζονταν από μία συμβολοσειρά που περιέγραφε τη θεματική περιοχή του μηνύματος. Οι συνδρομητές μπορούσαν πλέον να εκφραστούν καλύτερα, με το να δηλώνουν για παράδειγμα τα ενδιαφέροντα τους όχι μόνο με απόλυτη ταύτιση της θεματικής περιοχής αλλά χρησιμοποιώντας το κοινό πρόθεμα όλων των θεματικών ενοτήτων που τους ενδιέφεραν.
Στη κατηγορία των topic ή group-based συστημάτων δημοσιεύσεων/ συνδρομών ανήκουν το Scribe [32] , το Bayuex [33], το daMulticast το TIB/RV .Αργότερα, στις αρχές του αιώνα εμφανίστηκαν και επικράτησαν τα συστήματα δημοσιεύσεων/συνδρομών βασισμένα στο περιεχόμενο (content-based). Τα content-based συστήματα επικράτησαν επειδή ακριβώς προσφέρουν μεγαλύτερη ευελιξία κατά την διατύπωση των ενδιαφερόντων των χρηστών, με το να εφαρμόζουν συγκεκριμένους τελεστές πάνω στο ίδιο το περιεχόμενο.

Δίνεται έτσι μεγαλύτερη ευελιξία και έλεγχος στον συνδρομητή που πλέον μπορεί να ορίσει πιο σύνθετες και εκφραστικότερες συνδρομές, όπως για παράδειγμα «ενημέρωσε με για την τιμή της μετοχής της Google όταν αυτή υπερβεί τα 500 $».Συγκεκριμένα, υπάρχει η έννοια των ιδιοχαρακτηριστικών (attributes) τα οποία συνθέτουν τόσο τις συνδρομές όσο και τις δημοσιεύσεις. Κάθε ιδιοχαρακτηριστικό έχει συγκεκριμένο τύπο (κυρίως συμβολοσειρά ή αριθμητική τιμή) και μία συγκεκριμένη τιμή.

Τα υπό δημοσίευση μηνύματα περιέχουν ένα υποσύνολο από τα διαθέσιμα από το σύστημα ιδιοχαρακτηριστικά με συγκεκριμένες τιμές.
Από την άλλη, οι συνδρομές περιέχουν και αυτές ένα υποσύνολο των ιδιοχαρακτηριστικών, με συγκεκριμένους τελεστές πάνω στις τιμές τους. Πιθανοί τελεστές είναι για παράδειγμα το ισότητα, πρόθεμα (prefix), επίθεμα (suffix), υποσυμβολοσειρά (substring) για ιδιοχαρακτηριστικά τύπου συμβολοσειράς και ισότητα, ανισότητα, διαστήματα τιμών για αριθμητικούς τύπους. Κάτω από αυτό το μοντέλο, ένα μήνυμα συζευγνύεται με μία συνδρομή, εάν όλοι οι τελεστές των ιδιοχαρακτηριστικών της συνδρομής ικανοποιούνται από τις τιμές των ιδιοχαρακτηριστικών του μηνύματος.
Η μεγαλύτερη πρόκληση σε ένα πλήρως κατανεμημένο σύστημα δημοσιεύσεων/συνδρομών βασισμένο στο περιεχόμενο, είναι η ανάπτυξη αποδοτικών κατανεμημένων αλγορίθμων επεξεργασίας και σύζευξης των συνδρομών με τις δημοσιεύσεις, αλλά και παράδοσης των τελευταίων στους τελικούς χρήστες. Τα τελευταία χρόνια έχουν παρουσιαστεί αρκετές λύσεις όπως το SIENA [2], το REBECA [34], το Gryphon [1], το Hermes [35], το Elvin [3] και διάφορες άλλες.
Ας θεωρηθεί ότι για το υπόλοιπο κείμενο, οποτεδήποτε αναφερόμαστε σε συστήματα δημοσιεύσεων/συνδρομών, εννοούμε τα συστήματα δημοσιεύσεων/συνδρομών βασισμένα στο περιεχόμενο μιας και είναι πιο γενικός και επικρατέστερος τύπος. Άλλωστε, τα συστήματα δημοσιεύσεων/συνδρομών βασισμένα σε θεματικές ενότητες μπορούν να θεωρηθούν σαν υποσύνολο των συστημάτων βασισμένα στο περιεχόμενο, αν θεωρήσει κανείς ότι κάθε θεματική περιοχή μπορεί να θεωρηθεί ότι είναι η τιμή ενός μόνο ιδιοχαρακτηριστικού.

2.2 Ανασκόπηση Γνωστών Συστημάτων Δημοσιεύσεων/Συνδρομών Βασισμένα στο Περιεχόμενο

2.2.1 Τα Συστήματα Scribe και Bayuex

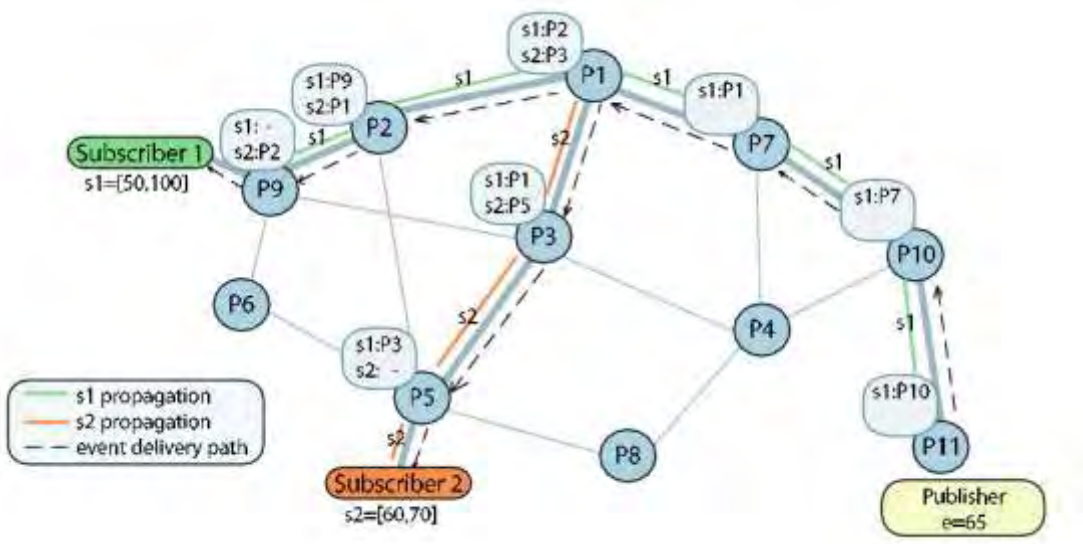
Το Scribe [31] είναι ένα κλιμακώσιμο σύστημα δημοσιεύσεων/ συνδρομών βασισμένο σε θεματικές ενότητες (topic based).Λειτουργεί σε ένα πλήρως κατανεμημένο περιβάλλον πάνω στο δίκτυο Pastry [36]. Ως εκ τούτου, εκμεταλλεύεται την αξιοπιστία, την ανοχή σε σφάλματα και την αυτόνομη οργάνωση του δικτύων Pastry. ΄Ολοι οι κόμβοι που συμμετέχουν στο Scribe είναι πλήρως αυτόνομοι και επεξεργάζονται τις δημοσιεύσεις και τις συνδρομές βασιζόμενοι αποκλειστικά σε τοπική πληροφορία. Κάθε κόμβος, μπορεί να είναι συνδρομητής, πάροχος πληροφορίας, ή να συμμετέχει σε multicast δέντρα που σκοπό έχουν την παράδοση των δημοσιεύσεων στους ενδιαφερόμενους συνδρομητές.
Στο Scribe, κάθε θεματική ενότητα (Subject ή Topic) συνοδεύεται από ένα τυχαίο αναγνωριστικό. Ακολουθώντας την προσέγγιση του κόμβου «ραντεβού», ο κόμβος με αναγνωριστικό που είναι αριθμητικά κοντά στο αναγνωριστικό της θεματικής ενότητας, είναι υπεύθυνος για την διαχείριση των δημοσιεύσεων και των συνδρομών που σχετίζονται με αυτή. Έτσι όλες η δημοσιεύσεις (αλλά και οι συνδρομές ή αλλιώς οι αιτήσεις συμμετοχής στη θεματική ενότητα) καταλήγουν στον κόμβο «ραντεβού» της θεματικής ενότητας, που στην συνέχεια είναι υπεύθυνος για την παράδοση τους στου συνδρομητές.
Κάθε συνδρομητής που επιθυμεί να εγγραφεί σε μία θεματική ενότητα, στέλνει ένα ειδικού τύπου μήνυμα (join subject) προς τον κόμβο «ραντεβού» της θεματικής ενότητας. Το μονοπάτι πάνω στο δίκτυο Pastry που σχηματίζεται κατά την δρομολόγηση αυτού του μηνύματος, αποτελεί τελικά ένα μονοπάτι στο multicast δέντρο που έχει ρίζα τον κόμβο «ραντεβού» και αξιοποιείται για την μετάδοση των δημοσιεύσεων στους συνδρομητές. Το Scribe είναι επίσης εξοπλισμένο με μηχανισμούς αυτόνομης οργάνωσης των multicast δέντρων σε περίπτωση που η συνεκτικότητα τους καταστραφεί εξαιτίας των συχνών αναχωρήσεων κόμβων ή πιθανών σφαλμάτων που μπορεί να παρουσιαστούν. Συγκεκριμένα κάθε κόμβος που ανήκει σε ένα multicast δέντρο στέλνει περιοδικά μηνύματα στον πατέρα του ώστε να διαπιστώσει εάν αυτός είναι «ζωντανός».

Αν κάτι τέτοιο δεν ισχύει, στέλνει ένα μήνυμα «συνδρομής» στη θεματική ενότητα (με το αναγνωριστικό της) κάτι που θα αναγκάσει τελικά τον κόμβο να συνδεθεί ξανά στο δέντρο.
Το σύστημα δημοσιεύσεων/συνδρομών Bayuex [33] ανήκει στην ίδια κατηγορία με το Scribe και παρουσιάζει αρκετές ομοιότητες με αυτό. Χρησιμοποιεί το δίκτυο ομότιμων εταίρων Tapestry [37] ώστε να διαχειριστεί και να παραδώσει τις δημοσιεύσεις στους ενδιαφερόμενους συνδρομητές. Και εδώ κτίζεται ένα multicast δέντρο για κάθε θεματική ενότητα ακολουθώντας την προσέγγιση του κόμβου «ραντεβού» που αποτελεί το σημείο συνάντησης των συνδρομών με τις δημοσιεύσεις.

2.2.2. Το Σύστημα SIENA

Στο χώρο των συστημάτων δημοσιεύσεων/συνδρομών με βάση το περιεχόμενο το SIENA [2] αποτέλεσε μια από τις πρώτες προσπάθειες. Λειτουργεί πάνω σε ένα δίκτυο από brokers και είναι ικανό να προσφέρει αποδοτική δρομολόγηση των δημοσιεύσεων στους συνδρομητές ανεξάρτητα από το μέγεθος τους δικτύου.
Η δρομολόγηση των δημοσιεύσεων επιτυγχάνεται με βάση το περιεχόμενο των συνδρομών. Ανήκει στην κατηγορία του filtering-based routing [38] και σκοπό έχει να σταματήσει όσο το δυνατόν συντομότερα τη μετάδοση μίας δημοσίευσης που δεν ενδιαφέρει κανένα συνδρομητή. Τα μονοπάτια παράδοσης των δημοσιεύσεων στους συνδρομητές, σχηματίζονται χρησιμοποιώντας πληροφορία που είναι τοπικά αποθηκευμένη σε κάθε κόμβο (φίλτρα) που σχηματίζεται από την κατάλληλη επεξεργασία των συνδρομών. Η πληροφορία δρομολόγησης σε ένα κόμβο σχετίζεται με τους γειτονικούς του κόμβους και τους συνδρομητές που μπορούν να προσπελαστούν μέσω αυτών. Αυτή η πληροφορία επιτρέπει τη δημιουργία αντίστροφων μονοπατιών (προς τους συνδρομητές) που μπορούν να ακολουθήσουν στη συνέχεια οι δημοσιεύσεις. Επιπλέον στο SIENA παρουσιάζεται μία τεχνική περιορισμού της μετάδοσης των συνδρομών με σκοπό να σχηματιστούν τα αντίστροφα μονοπάτια, με το να εκμεταλλεύεται την σχέση μεταξύ των συνδρομών.

Δηλαδή, μία συνδρομή σ1 περικλείει την συνδρομή σ2 εάν όλες οι δημοσιεύσεις που συζευγνύουν με την συνδρομή σ2, συζευγνύουν και με την σ1 (όπως μπορείτε να δείτε στην εικόνα 1 όπου η συνδρομή s2 με περιορισμό διαστήματος [60, 70] καλύπτεται από τη συνδρομή s1 με περιορισμό διαστήματος [50, 100]).



Εικόνα 1: Δρομολόγηση δημοσίευσης n1 (με τιμή 65) στους συνδρομητές με συνδρομές s1 και s2 στο σύστημα SIENA.

Εκμεταλλευόμενοι αυτή τη παρατήρηση οι συνδρομές ταξιδεύουν μέχρι ένα ορισμένο σημείο στο δίκτυο όπου συναντούν ένα κόμβο που κατέχει ένα φίλτρο κατασκευασμένο από συνδρομές που περικλείουν τη συνδρομή. Μια ακόμα τεχνική, που αποτελεί χαρακτηριστικό γνώρισμα του SIENA είναι οι διαφημίσεις. Μια διαφήμιση ορίζει το είδος των δημοσιεύσεων που ένα πάροχος πληροφορίας πρόκειται να προσφέρει στο σύστημα, και χρησιμοποιείται για την δημιουργία των μονοπατιών που τελικά θα ακολουθήσουν οι δημοσιεύσεις μέχρι τους τελικούς συνδρομητές.

Το σύστημα SIENA αποτελεί σημείο αναφοράς στην δρομολόγηση με βάση το περιεχόμενο και έχουν υιοθετηθεί οι λύσεις που παρουσιάζονται εκεί από άλλες ερευνητικές εργασίας. Υποστηρίζει αποδοτικά δημοσιεύσεις και συνδρομές με αριθμητικές τιμές και συμβολοσειρές, ενώ μπορεί να θεωρηθεί σαν μία stateful προσέγγιση εξαιτίας της πληροφορίας σχετικά με τη δρομολόγηση που πρέπει να διατηρείται σε κάθε κόμβο.

2.2.3 Το Σύστημα REBECA

Το σύστημα δημοσιεύσεων/συνδρομών REBECA [34], ενσωματώνει μία αντικειμενοστρεφή υπηρεσία παράδοσης δημοσιεύσεων, υλοποιημένη σε Java. Οι κόμβοι που συμμετέχουν στο σύστημα σχηματίζουν ένα ακυκλικό γράφο. Η δρομολόγηση των δημοσιεύσεων στους συνδρομητές ακολουθεί το μοντέλο filtering-based όπως και στο SIENA και έχει επεκταθεί ώστε να περιλαμβάνει

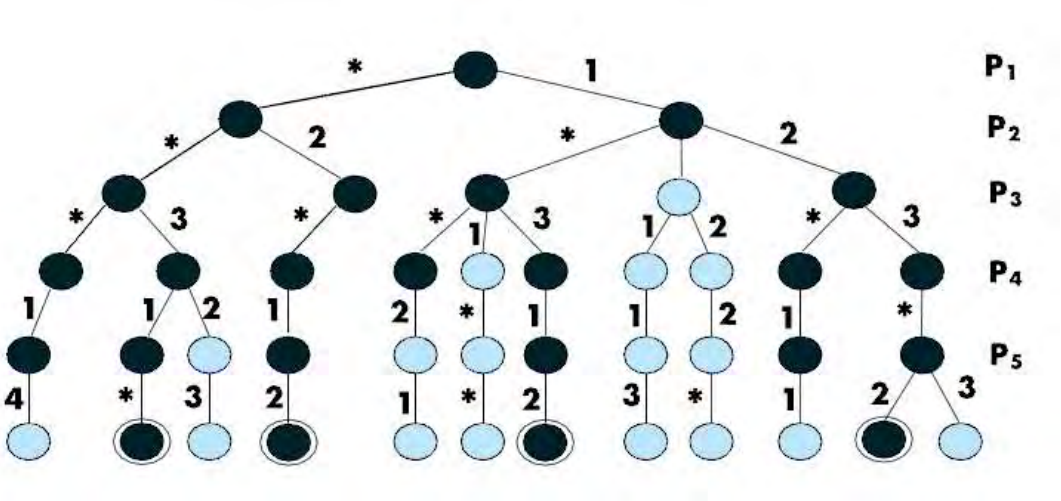
(i) απλή δρομολόγηση που αντιστοιχεί σε flooding των συνδρομών σε ολόκληρο το δίκτυο ώστε οι δημοσιεύσεις να φιλτράρονται στους κόμβους όπου γεννιούνται,
(ii) identity routing που αποφεύγει την προώθηση μιας συνδρομής σε επόμενο κόμβο όταν αυτό ήδη έχει γίνει για άλλη πανομοιότυπη συνδρομής,
(iii) covering routing, που μιμείται την δρομολόγηση στο SIENA όπου συνδρομές που περικλείονται από άλλες δεν προωθούνται παραπέρα, και
(iv) merging routing όπου οι συνδρομές μπορούν να ενσωματωθούν και να δημιουργήσουν μία νέα συνδρομή που στη συνέχεια προωθείται σε επόμενους κόμβους.

Στο σύστημα REBECA χρησιμοποιείται και η τεχνική των διαφημίσεων (όπου οι παραγωγοί πληροφορίας «διαφημίζουν» τα δεδομένα που πρόκειται να δημοσιεύσουν) ώστε να βελτιωθεί περαιτέρω η δρομολόγηση των δημοσιεύσεων. Αργότερα προτάθηκαν επεκτάσεις τους συστήματος REBECA , ώστε αυτό να ενσωματωθεί στο δίκτυο ομοτίμων εταίρων Chord.

2.2.4. Gryphon

Στο σύστημα Gryphon [2] (που βασίζεται στις εργασίες που παρουσιάστηκαν το 1999) η σύζευξη των συνδρομών με τις δημοσιεύσεις επιτυγχάνεται με την βοήθεια μίας δενδρικής δομής που σχηματίζεται με κατάλληλη επεξεργασία των συνδρομών. Συγκεκριμένα, σε αυτό το δέντρο τα φύλλα του αποτελούνται από τις συνδρομές. Κάθε ενδιάμεσος κόμβος αποτελεί μία έκφραση πάνω στα ιδιοχαρακτηριστικά των συνδρομών, ενώ οι ακμές ορίζονται από το αποτέλεσμα των εκφράσεων των κόμβων.

Στην εικόνα 2 μπορείτε να δείτε ένα παράδειγμα της δενδρικής δομής σύζευξης δημοσιεύσεων και συνδρομών που έχει σχηματιστεί με την επεξεργασία 5 ιδιοχαρακτηριστικών (ρ1 - ρ5). Ο κόμβος σε βάθος 0 (η ρίζα του δέντρου) αντιστοιχεί στον έλεγχο της τιμής του ιδιοχαρακτηριστικού ρ1 που ορίζεται στις συνδρομές. Οι κόμβοι σε μεγαλύτερα βάθη αντιστοιχούν σε ελέγχους των τιμών των υπολοίπων ιδιοχαρακτηριστικών, όπως μπορείτε να δείτε στην εικόνα. Οι ακμές του δέντρου συνοδεύονται από ετικέτες και αντιστοιχούν στην τιμή για την οποία ο έλεγχος είναι επιτυχής. Παραδείγματος χάριν, στην εικόνα 2 η δεξιά ακμή της ρίζας αντιπροσωπεύει τον έλεγχο του ρ1 = 1. Η αριστερή ακμή της ρίζας, με ετικέτα το*, δείχνει ότι οι συνδρομές διαμέσου της συγκεκριμένης διακλάδωσης δεν ενδιαφέρονται για την τιμή του ιδιοχαρακτηριστικού ρ1.



Εικόνα 2 : Παράδειγμα της δενδρικής δομής σύζευξης δημοσιεύσεων με συνδρομές στο σύστημα Gryphon.

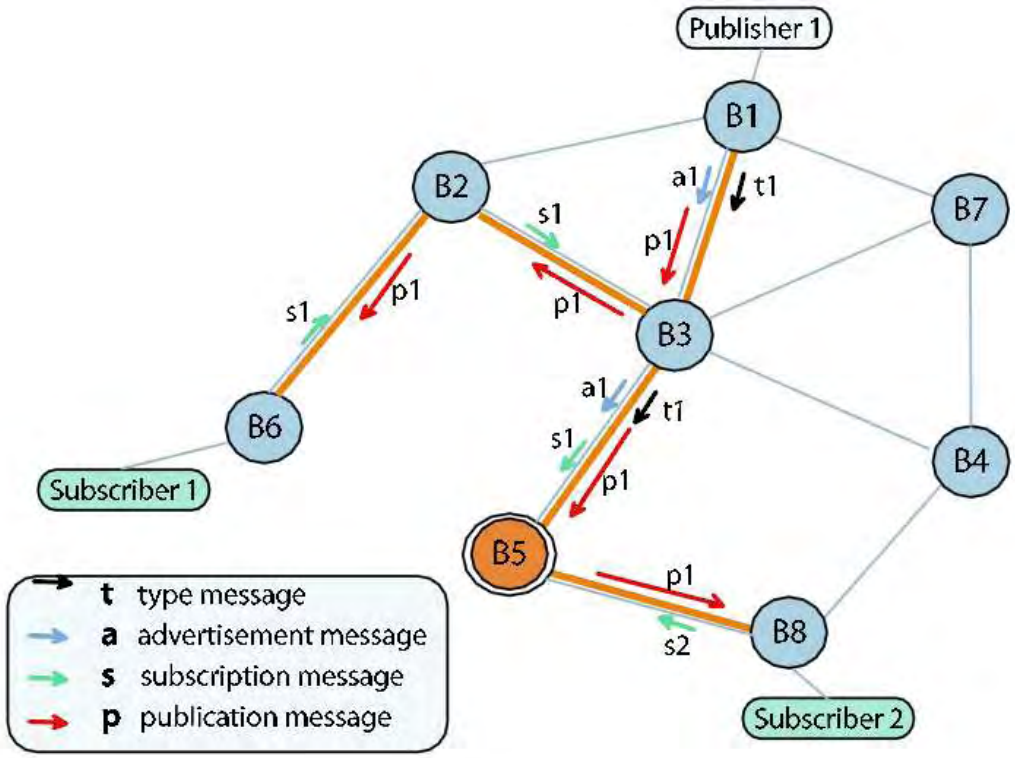
Εάν ακολουθήσουμε τη δεξιότερη διαδρομή, θα καταλήξουμε σε μία συνδρομή που έχει ως περιορισμούς τους ρ1=1Λ ρ2=2Λ ρ3=3Λ ρ 5=3.
Η σύζευξη μίας δημοσίευσης επιτυγχάνεται ξεκινώντας από την ρίζα του δέντρου και με βάση τις τιμές των ιδιοχαρακτηριστικών του, καταλήγει τελικά στα φύλλα που αντιστοιχούν σε συνδρομές που συζευγνύουν με τη δημοσίευση. Ξεκινώντας από τη ρίζα και ελέγχοντας το ιδιοχαρακτηριστικό ρ1 προχωράμε αριστερά ή δεξιά από τη ρίζα στον επόμενο κόμβο.

Σε κάθε ενδιάμεσο κόμβο του δέντρου, ακολουθούμε τη ακμή που η ετικέτα της ικανοποιεί την τιμή του ιδιοχαρακτηριστικού της δημοσίευσης που αντιστοιχεί στο συγκεκριμένο βάθος, ή ακολουθούμε την ακμή με ετικέτα *. Με αυτό τον κανόνα, ξεκινάει μια παράλληλη αναζήτηση στο εσωτερικό του δέντρου. Για παράδειγμα, αν εκτελέσουμε τον αλγόριθμο ταύτισης στο δέντρο της εικόνας 2 για την δημοσίευση με ιδιοχαρακτηριστικά ρ1=1, ρ2=2, ρ3=3, ρ4=1, ρ5=2, τότε θα επισκεφθούμε όλους τους κόμβους με μαύρο χρώμα και θα βρούμε τέσσερις συνδρομές (τα φύλλα του δέντρου με μαύρο χρώμα).

2.2.5. Το Σύστημα Hermes

Το σύστημα Hermes αποτελεί μια ολοκληρωμένη λύση επεξεργασίας δημοσιεύσεων και συνδρομών συνδυάζοντας το topicbased μοντέλο με το content-based. Είναι λοιπόν ένας συνδυασμός των δύο βασικών μοντέλων των συστημάτων δημοσιεύσεων συνδρομών. Το Hermes βασίζεται στη προσέγγιση του κόμβου «ραντεβού» με το μοντέλο δημοσιεύσεων/συνδρομών με βάση τη θεματική ενότητα.
Το δίκτυο πάνω στο οποίο λειτουργεί το Hermes αποτελείται από brokers ωστόσο έχουν προταθεί και προσεγγίσεις που εκμεταλλεύονται άλλου είδους αρχιτεκτονικές ώστε να αντιμετωπίσουν προβλήματα σχετικά με την αναχώρηση κόμβων από το δίκτυο ή την ύπαρξη σφαλμάτων σε αυτό. Κάθε συνδρομή ανήκει σε συγκεκριμένη θεματική ενότητα (ή αλλιώς χαρακτηρίζεται από συγκεκριμένο τύπο) και αντιστοιχίζεται με τον κόμβο του δικτύου του οποίου το αναγνωριστικό είναι αυτό που προκύπτει (ή τουλάχιστον είναι αριθμητικά κοντά) από την τιμή που επιστρέφει μια συνάρτηση κατακερματισμού με το όνομα της θεματικής ενότητας ως όρισμα. Όλες λοιπόν οι δημοσιεύσεις και οι συνδρομές που ανήκουν στην ίδια θεματική ενότητα καταλήγουν σε αυτό τον κόμβο «ραντεβού». Στην συνέχεια και με σκοπό να παραδοθεί η δημοσίευση στου τελικούς συνδρομητές, ακολουθείται μία προσέγγιση όμοια με αυτή που παρουσιάστηκε στα πλαίσια του SIENA. Δημιουργούνται έτσι μονοπάτια που αποτελούν μέρη του multicast δέντρου με ρίζα τον κόμβο «ραντεβού» και φύλλα τους συνδρομητές.
Στην εικόνα 3 μπορείτε να δείτε σχηματικά την ανάπτυξη του multicast δέντρου. Σε αυτό το παράδειγμα μπορείτε να δείτε 4 brokers, δύο παραγωγούς πληροφορίας (P1 και P2), δύο συνδρομητές (S1 και S2), και τον κόμβο «ραντεβού» R. Οι P1 και P2 στέλνουν διαφημίσεις (ak) στον κόμβο R σχετικά με την πληροφορία που πρόκειται να δημοσιεύσουν.

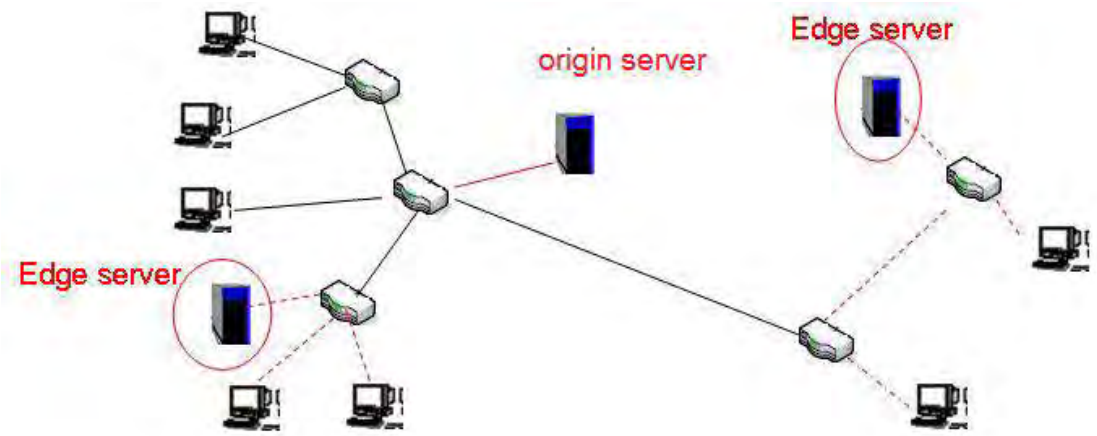
Οι δύο συνδρομητές S1 και S2 στέλνουν τις δημοσιεύσεις τους (sk) προς τον κόμβο R ενώ παράλληλα σχηματίζονται τα κατάλληλα φίλτρα στους ενδιάμεσους κόμβους στα πρότυπα του SIENA. Όταν οι δημοσιεύσεις (pk) καταφθάσουν στο σύστημα προωθούνται προς τον κόμβο R, από όπου στη συνέχεια ακολουθούν τα αντίστροφα μονοπάτια που σχημάτισαν οι συνδρομές, ώστε να καταλήξουν στους συνδρομητές.



Εικόνα 3 : Παράδειγμα ανάπτυξης του multicast δέντρου στο σύστημα Hermes.

2.3. Δίκτυα Παράδοσης Περιεχομένου (CDNs)

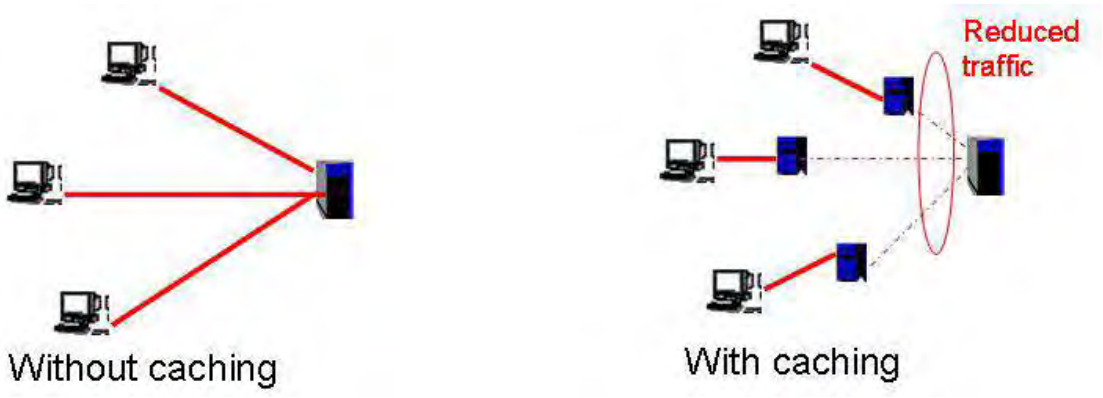
Ένα CDN είναι μια συλλογή από διακομιστές περιεχομένου διάσπαρτους σε όλο το δίκτυο, που προσπαθούν να απορροφήσουν ένα κομμάτι του φόρτου εργασίας των διακομιστών προέλευσης παραδίδοντας συνεργατικά περιεχόμενο για λογαριασμό τους.
Οι πελάτες παίρνουν το περιεχόμενο που ζήτησαν από το πλησιέστερο διακομιστή περιεχομένου που βρίσκεται από τους DNS servers. Επιτυγχάνει καλύτερη απόδοση και αξιοπιστία από την κρυφή μνήμη. Υπεύθυνος της λειτουργίας του CDN είναι ο Διαχειριστής του Περιεχομένου. Οι πελάτες του CDN δεν είναι οι ISPs όπως ήταν στην προσωρινή αποθήκευση(caching), αλλά οι πάροχοι περιεχομένου. Σε αντίθεση με την προσωρινή αποθήκευση στο διαδίκτυο(web caching), όπου η αναπαραγωγή του περιεχομένου αποφασίζεται ανάλογα με τις αιτήσεις που φτάνουν στις κρυφές μνήμες την τελευταία χρονική περίοδο, στα CDNS η αντιγραφή του περιεχομένου γίνεται με ντετερμινιστικό τρόπο που αποφασίζεται από έναν διευθυντή περιεχομένου. Η επόμενη εικόνα δείχνει ένα παράδειγμα του CDN.



Εικόνα 4 : Παράδειγμα CDN

2.4. Web Caching

Στο πλαίσιο της προσέγγισης αυτής αντικείμενα από περισσότερους του ενός διακομιστές προέλευσης αντιγράφονται στις κρυφές μνήμες κοντά στους χρήστες. Οι κρυφές μνήμες αποθηκεύουν πιο συχνά ή πιο πρόσφατα ζητούμενο περιεχόμενο κατά έναν παθητικό ανεξέλεγκτο τρόπο. Εάν το ζητούμενο αντικείμενο δεν περιέχεται στην κρυφή μνήμη, τότε η αίτηση ανακατευθύνεται στο διακομιστή προέλευσης.Η προσωρινή αποθήκευση στο Web(Web Caching) χρησιμοποιείται συνήθως από έναν διαδικτυακό Πάροχο Υπηρεσιών Διαδικτύου (ISP) προς όφελος των χρηστών έχουν πρόσβαση στο Διαδίκτυο.
Το όφελος για τον ISP είναι ότι ένα μέρος των αιτημάτων περιεχόμενο απορροφάται από τις κρυφές μνήμες, μειώνοντας το επίπεδο συμφόρησης του δικτύου και την υψηλή επεξεργαστική ισχύ που απαιτείται από τον κεντρικό server ενώ το όφελος για τους χρήστες του διαδικτύου είναι η εμπειρία μικρότερων καθυστερήσεων. Η εικόνα 5 απεικονίζει τα παραπάνω.

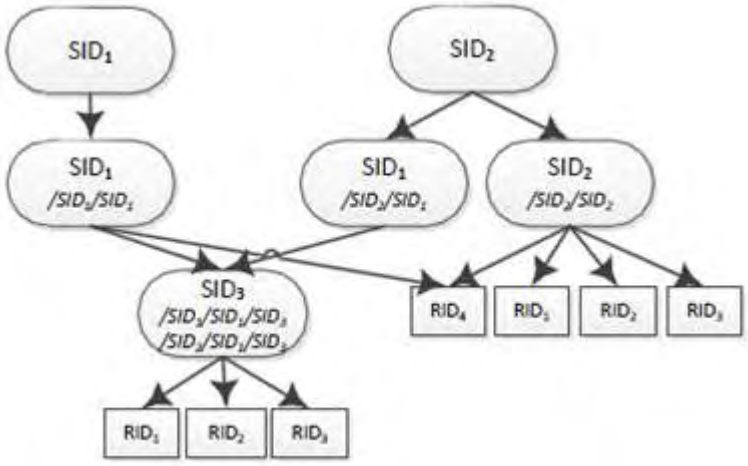


Εικόνα 5: Παράδειγμα Web Caching

ΚΕΦΑΛΑΙΟ 3- Αρχιτεκτονική Συστήματος [25]

3.1. Οι Αμετάβλητες της Αρχιτεκτονικής

Η πρώτη αμετάβλητη(invariant) είναι η ***επισήμανση(labelling) και συσχέτιση(referencing) πληροφοριακών στοιχείων***. Αντί της ανάθεσης ταυτοτήτων σε end-hosts, στην προσέγγισή μας, προσδιορίζουμε τα πληροφοριακά στοιχεία με τη χρήση ενός (στατιστικά μοναδικού) επίπεδου αναγνωριστικού ετικέτας(flat label identifier).
Αυτά τα αναγνωριστικά είναι η αυτο-παραγόμενα και η συνδεδεμένη σημασιολογία των πληροφοριών είναι γνωστή μόνο στο αναγνωριστικό που δημιουργεί όπως είπαμε την εκτέλεση.
Η δεύτερη invariant είναι αυτή της ***οριοθέτησης του πεδίου πληροφορίας(information scoping).*** Οι εμβέλειες καθορίζουν το σύνολο των πληροφοριών που είναι διαδίδονται σε ένα συγκεκριμένο πλαίσιο. Με αυτή, επιτυγχάνουμε οριοθέτηση του πεδίου εφαρμογής σε λειτουργικό και πληροφοριακό επίπεδο. Κάθε αντικείμενο πληροφορίας τοποθετείται σε τουλάχιστον ένα πεδίο-εμβέλεια. Όντας ένα σύνολο πληροφοριών, οι εμβέλειες θεωρούνται ως αντικείμενα πληροφορίας οι ίδιες. Ως εκ τούτου, αναγνωρίζονται με τον ίδιο τρόπο όπως και τα αντικείμενα πληροφορίας και μπορούν να τοποθετηθούν σε άλλα πεδία-εμβέλειες. Με αυτόν τον τρόπο ενδεχομένως πολύπλοκες γραφικές παραστάσεις πληροφοριών μπορούν να δημιουργηθούν.
Οι δύο πρώτες σταθερές καθορίζουν τον τρόπο με τον οποίο είναι δομημένες οι πληροφορίες με την προτεινόμενη προσέγγιση.



Εικόνα 6 : Δομή Πληροφορίας

Η τρίτη invariant είναι αυτή του ***διαχωρισμού των λειτουργιών (separating the functions)*** ,για τη διάδοση πληροφοριών στο εσωτερικό ενός δεδομένου πεδίου εφαρμογής, σε τρεις διακριτές οντότητες:

(i) **ραντεβού(rendezvous)** : αφορά το ταίριασμα της διαθεσιμότητας των πληροφοριών των εκδοτών και του ενδιαφέροντος των συνδρομητών σε αυτές. Αυτό οδηγεί σε κάποια μορφή πληροφορίας τοποθεσίας για το σύνολο των εκδοτών και των συνδρομητών που έχουν ταιριάξει

(ii) **διαχείριση και διαμόρφωση τοπολογίας(topology management and formation)** : καθορίζει μία κατάλληλη τοπολογία για τη μεταφορά των αντίστοιχων πληροφοριών

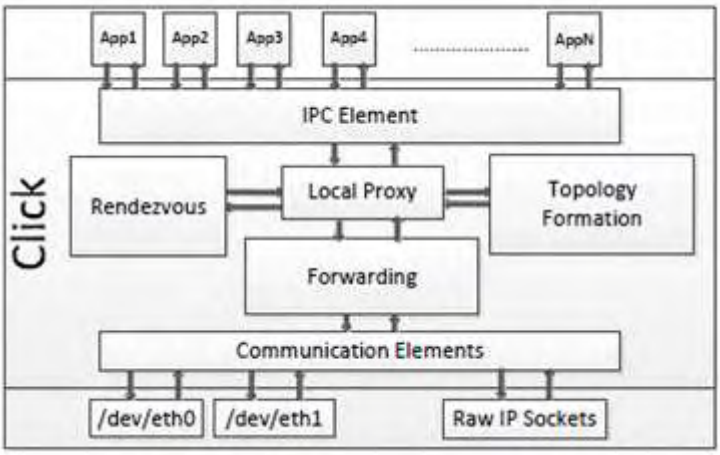
(iii) **προώθηση**: εκτελεί αυτή τη μεταφορά των πληροφοριών.

Η τελευταία invariant είναι το ***μοντέλο των υπηρεσιών (service model)*** που υποστηρίζεται από τη νέα πληροφοριο-κεντρική «μέση». Χωρίς να μπούμε σε λεπτομέρειες, σημειώνουμε ότι το προτεινόμενο μοντέλο είναι ένα καθαρό pub/sub μοντέλο επικοινωνίας, όπου κάθε κόμβος του δικτύου είναι σε θέση να δημοσιεύσει τα πεδία εφαρμογής και τα αντικείμενα πληροφοριών σε αυτά τα πεδία, καθώς και να εγγραφεί για συγκεκριμένα αντικείμενα πληροφορίας αντικείμενα ή πεδία. Αντίστοιχα, οι hosts του δικτύου μπορούν να σταματήσουν να δημοσιεύουν καθώς και να διαγραφούν από τα αντικείμενα πληροφορίας και τα πεδία εφαρμογής. Το service model περιλαμβάνει επίσης μια συνάρτηση για τη δημοσίευση των πραγματικών δεδομένων για ένα συγκεκριμένο αντικείμενο πληροφορίας. Η συμπεριφορά της αυτής της συνάρτησης εξαρτάται από την στρατηγική διάδοσης που ανατίθεται στο πεδίο εφαρμογής στο οποίο υπόκειται αυτό το αντικείμενο. Όπως περιγράφουμε παρακάτω, παραδείγματα στρατηγικών διάδοσης θα ήταν μια τοπικού-κόμβου στρατηγική όπου οι πληροφορίες διαδίδονται στα πλαίσια ενός μόνου κόμβου, μια τοπικής-σύνδεσης στρατηγική όπου οι πληροφορίες διαδίδονται σε δύο φυσικά συνδεδεμένους κόμβους ή μια τοπική-domain, όπου οι πληροφορίες διαδίδονται σε μια domain διαχείρισης.

Όπως εξετάζουμε στο επόμενο τμήμα, κάθε κόμβος του δικτύου μπορεί να εκτελέσει όλες τις τρεις λειτουργίες του δικτύου που περιγράφονται στην τρίτη μας invariant. Για παράδειγμα, ένας ενιαίος host δικτύου μπορεί να λειτουργήσει ως το σημείο συνάντησης (RP) για ένα αυθαίρετα μεγάλο διάγραμμα πληροφοριών που είναι προσβάσιμο μόνο σε τοπικό επίπεδο. Δηλαδή, μόνο εφαρμογές που τρέχουν σε αυτόν τον κόμβο μπορούν να έχουν πρόσβαση στη δομή της πληροφορίας χρησιμοποιώντας το εξαγόμενο μοντέλο παροχής υπηρεσιών. Σε αυτό το παράδειγμα, η συνάρτηση σχηματισμού μιας τοπολογίας είναι απούσα, ενώ η συνάρτηση προώθησης αναλαμβάνει την δημιουργία της Inter-Process Επικοινωνίας.
Ένα πιο σύνθετο παράδειγμα επικοινωνίας θα ήταν αυτό που περιλαμβάνει δύο κόμβους που γειτονεύουν φυσικά .
Στην περίπτωση αυτή, η διαδικασία του ραντεβού μπορεί να τρέξει και στους δύο κόμβους. Με αυτόν τον τρόπο, κάθε κόμβος μπορεί να έχει πρόσβαση στο γράφο πληροφορίας για τον οποίο το RP είναι ο γειτονικός κόμβος. Ως εκ τούτου, η πληροφορία μπορεί να μοιραστεί μεταξύ των δύο κόμβων. Σε αυτό το παράδειγμα, η συνάρτηση σχηματισμού μιας τοπολογίας είναι ελάχιστη, δεδομένου ότι η τοπολογία δύο-κόμβων είναι εκ των προτέρων γνωστή και από τους δύο κόμβους. Η συνάρτηση προώθησης είναι επίσης πολύ απλή, δεδομένου ότι οι πληροφορία διαδίδεται πάντα στις εφαρμογές που εκτελούνται στο γειτονικό δίκτυο κόμβων. Τέλος, σε ένα παράδειγμα domain-wide επικοινωνίας, αφιερωμένοι κόμβοι θα αναλάβουν τη συνάρτηση του ραντεβού και απομακρυσμένοι εκδότες και συνδρομητές θα έχουν πρόσβαση στη δομή πληροφορίες που διατηρείται στα σημεία των ραντεβού(RP) χρησιμοποιώντας το μοντέλο των υπηρεσιών. Η συνάρτηση σχηματισμού τοπολογίας σε αυτό το παράδειγμα είναι ζωτικής σημασίας, δεδομένου ότι έχει να δημιουργήσει μονοπάτια προώθησης μεταξύ των ενδιαφερόμενων φορέων. Η συνάρτηση προώθησης, η οποία τρέχει σε όλους τους ενδιάμεσους κόμβους του δικτύου, εδώ, αναλαμβάνει την υποχρέωση να προωθήσει πληροφορίες από τους εκδότες προς τους συνδρομητές που χρησιμοποιούν τις διαδρομές προώθησης που δημιουργούνται από τη συνάρτηση σχηματισμού τοπολογίας.

3.2. Η Αρχιτεκτονική Κόμβου

Μετά την εξέταση των αρχιτεκτονικών σταθερών του προτεινόμενου ICN, θα περιγράψουμε εν συντομία την υλοποίηση ενός κόμβου του δικτύου, η οποία βασίζεται στο αρθρωτό click router. Στην εικόνα 8 παρουσιάζονται τα βασικά στοιχεία που απαρτίζουν ένα κόμβο του δικτύου στο ICN μας.



Εικόνα 8 : Αρχιτεκτονική Κόμβου Δικτύου

Το **στοιχείο IPC** εξάγει το μοντέλο παροχής υπηρεσιών που περιγράφεται στην τελευταία αμετάβλητη στις εφαρμογές που εκτελούνται τοπικά, οι οποίες στέλνουν τα πληροφοριο-κεντρικά αιτήματά τα τους σε αυτό το στοιχείο μέσω ενός μηχανισμού IPC.
Σημειώστε ότι η τρέχουσα υλοποίηση κόμβου τρέχει ως μια διαδικασία στο χώρο του χρήστη. Επί του παρόντος, το στοιχείο IPC υλοποιεί μια ασύγχρονη loopback socket TCP στην οποία οι εφαρμογές στέλνουν pub / sub αιτήματα και λαμβάνουν γεγονότα(ειδοποιήσεις) ή δεδομένα από το υποκείμενο λογισμικό δικτύωσης. Για παράδειγμα, μια εφαρμογή μπορεί να στείλει ένα αίτημα πεδίου-δημοσίευσης που θα ωθηθεί περαιτέρω προς το ***στοιχείο Local Proxy***. Επιπλέον, μέσω του ίδιου μηχανισμού, οι συνδρομητές λαμβάνουν ειδοποιήσεις για δημοσιοποιημένα ή μη υπο-πεδία ή τα πραγματικά δεδομένα για τα στοιχεία πληροφορίας που υπάρχουν κάτω από τα πεδία στα οποία έχουν προηγουμένως εγγραφεί. Οι εκδότες λαμβάνουν επίσης ειδοποιήσεις να ξεκινήσουν (όταν τουλάχιστον ένας συνδρομητής έχει εκδηλώσει ενδιαφέρον για ένα θέμα που διαφημίζεται από τον εκδότη) ή να σταματήσουν (όταν δεν υπάρχουν συνδρομητές) τη δημοσίευση δεδομένων για ένα συγκεκριμένο στοιχείο πληροφορίας.

Όπως απεικονίζεται στην εικόνα 8, ***το στοιχείο Local Proxy*** έχει κεντρικό ρόλο στην αρχιτεκτονική του κόμβου.Είναι υπεύθυνο για την παρακολούθηση όλων των εκκρεμών πληροφοριών (συμπεριλαμβανομένων των πεδίων) εκδόσεων και συνδρομών. Λειτουργεί επίσης ως πληρεξούσιο για όλες τις εφαρμογές που εκτελούνται τοπικά. Δηλαδή, οι απομακρυσμένοι κόμβοι ποτέ δεν γνωρίζουν για συγκεκριμένες εφαρμογές που τρέχουν σε έναν host δικτύου, αλλά μόνο γι 'αυτόν τον host. Για το λόγο αυτό, στατιστικά μοναδικές ετικέτες αυτο-ανατίθενται από κάθε στοιχείο Local Proxy. Όταν τα δεδομένα για ένα αντικείμενο πληροφορίας ωθούνται σε αυτό το στοιχείο, είτε από μια εφαρμογή που εκτελείται τοπικά ή από το δίκτυο (δηλαδή το στοιχείο της προώθησης), αυτό το στοιχείο αναζητά για όλες τις εφαρμογές που έχουν ήδη εκφράσει το ενδιαφέρον τους για αυτό το αντικείμενο και ωθεί τα αντίστοιχα πακέτα δεδομένων σε αυτές. Εάν τα στοιχεία ωθήθηκαν από έναν εκδότη που τρέχει τοπικά και υπήρχαν οι πληροφορίες για την προώθηση δεδομένων στο δίκτυο (π.χ. ένα αναγνωριστικό LIPSIN σε μια domain-local στρατηγική διάδοσης), θα ωθούσε τα δεδομένα για το ***στοιχείο Προώθησης (Forwarding Element).*** Αντίστοιχα, όταν μια ειδοποίηση για ένα πρόσφατα δημοσιευμένο πεδίο φτάνει από το δίκτυο (δηλαδή μια ειδοποίηση που απέστειλε η συνάρτηση των ραντεβού που εκτελείται στο RP για τη συγκεκριμένο πεδίο στο δίκτυο) ***το Local Proxy στοιχείο*** πρέπει να ειδοποιήσει τις ενδιαφερόμενες εφαρμογές.
Τέλος, ειδοποιεί μια εφαρμογή εκδότη όταν φθάνει από την RP ενός συγκεκριμένου αντικειμένου πληροφορίας, μια ειδοποίηση για να αρχίσει ή να σταματήσει τη δημοσίευση δεδομένων.

Το **στοιχείο Rendezvous** υλοποιεί το pub / sub μοντέλο παροχής υπηρεσιών, διατηρώντας το γράφο πληροφοριών για τις οποίες ο κόμβος του δικτύου είναι το RP. Για παράδειγμα, το στοιχείο ραντεβού που τρέχει σε κάθε κόμβο του δικτύου διατηρεί και ενημερώνει τη δομή πληροφοριών για τις οποίες η στρατηγική διάδοσης είναι κόμβο-τοπική(node-local). Σε κάθε κόμβο του δικτύου, μια εφαρμογή μπορεί να διαφημίσει ένα κομμάτι πληροφορίας στο οποίο μια άλλη εφαρμογή μπορεί να γίνει συνδρομητής. Σε αυτή την περίπτωση το στοιχείο ραντεβού θα ταιριάξει τα συμφέροντα αυτών των εφαρμογών και θα ωθήσει μια ειδοποίηση για να ξεκινήσει η δημοσίευση δεδομένων στο Local Proxy στοιχείο, το οποίο με τη σειρά του, θα την ωθήσει προς τον εκδότη.

Στη domain-wide στρατηγική διάδοσης, ένας ή περισσότεροι ειδικοί και γνωστοί κόμβοι του δικτύου θα είναι αρμόδιοι για την παραλαβή αιτήσεων από άλλους κόμβους που υπάρχουν στο ίδιο domain.

Το **στοιχείο Σχηματισμού Τοπολογίας(Topology Formation Element)** τρέχει σε κάθε κόμβο του δικτύου και είναι υπεύθυνο για τον εμπλουτισμό των δομών δεδομένων πεδίων με τις κατάλληλες πληροφορίες προώθησης μόλις βρεθεί ένα ταίριασμα μεταξύ εκδότη και συνδρομητή . Ένας ή περισσότεροι κόμβοι που λειτουργούν ως διαχειριστές της τοπολογία του domain χρησιμοποιούν αυτό το στοιχείο για να εγγραφούν σε γνωστά πεδία, προκειμένου να λάβουν τοπολογικές πληροφορίες από άλλους κόμβους προώθησης, καθώς και αιτήσεις τοπολογίας και κατασκευάζουν αναγνωριστικά προώθησης τα οποία στη συνέχεια τα δημοσιεύουν σε όλους τους ενδιαφερόμενους κόμβους στο δίκτυο. Για παράδειγμα, όταν ένας ειδικός κόμβος domain ραντεβού ταιριάζει έναν εκδότη με ένα σύνολο συνδρομητών, δημοσιεύει ένα αίτημα ανάλυσης τοπολογίας χρησιμοποιώντας ένα αναγνωριστικό πληροφορίας στο οποίο το στοιχείο σχηματισμού τοπολογίας είναι ήδη εγγεγραμένο (ή σε ένα πεδίο γονέα αυτού του αντικειμένου). Στη συνέχεια, το στοιχείο σχηματισμού τοπολογίας υπολογίζει ένα LIPSIN αναγνωριστικό από τον εκδότη στους συνδρομητές και το δημοσιεύει χρησιμοποιώντας ένα άλλο αναγνωριστικό πληροφορίας στο οποίο η συνάρτηση ραντεβού είναι εγγεγραμμένη. Με αυτόν τον τρόπο η συνάρτηση RV ειδοποιεί τον εκδότη να δημοσιεύσει δεδομένα για την αρχική πληροφορία που χρησιμοποιώντας χρησιμοποιώντας το ήδη υπολογισμένο αναγνωριστικό LIPSIN. Η συνάρτηση ραντεβού υποστηρίζει μια M*N σχέση μεταξύ του αριθμού των εκδοτών και των συνδρομητών για ένα συγκεκριμένο αντικείμενο πληροφορίας. Είναι ευθύνη του διαχειριστή τοπολογίας να δημιουργήσει βέλτιστα μονοπάτια (π.χ. συντομότερα μονοπάτια) από έναν ή περισσότερους εκδότες στους συνδρομητές.

Το **στοιχείο Προώθησης(Forwarding Element)** λαμβάνει τα πακέτα από τα υποκείμενα στοιχεία επικοινωνίας, καθώς και από το Local Proxy στοιχείο και τις προωθεί με βάση το αναγνωριστικό LIPSIN που μεταφέρεται από το πακέτο. Ως εκ τούτου, ένα πακέτο που λαμβάνεται από το δίκτυο μπορεί να ωθηθεί σε κάποιο άλλο στοιχείο επικοινωνίας από το οποίο θα σταλεί πίσω στο δίκτυο και / ή να ωθηθεί στο Local Proxy στοιχείο. Η ίδια διαδικασία ακολουθείται και όταν ένα πακέτο ωθείται από το Local Proxy στοιχείο.

Τέλος, τα **στοιχεία επικοινωνίας(Communication Elements)** είναι στοιχεία click μέσω των οποίων τα pub/sub αιτήματα ωθούνται στο δίκτυο από το στοιχείο Προώθησης. Η τρέχουσα εφαρμογή υποστηρίζει καθαρή επικοινωνία Ethernet, καθώς και επικοινωνία μέσω “πρώτων” IP sockets όταν τα pub / sub δικτυακά "νησιά" πρέπει να διασυνδέονται μέσω της τακτικής σύνδεσης μέσω IP.

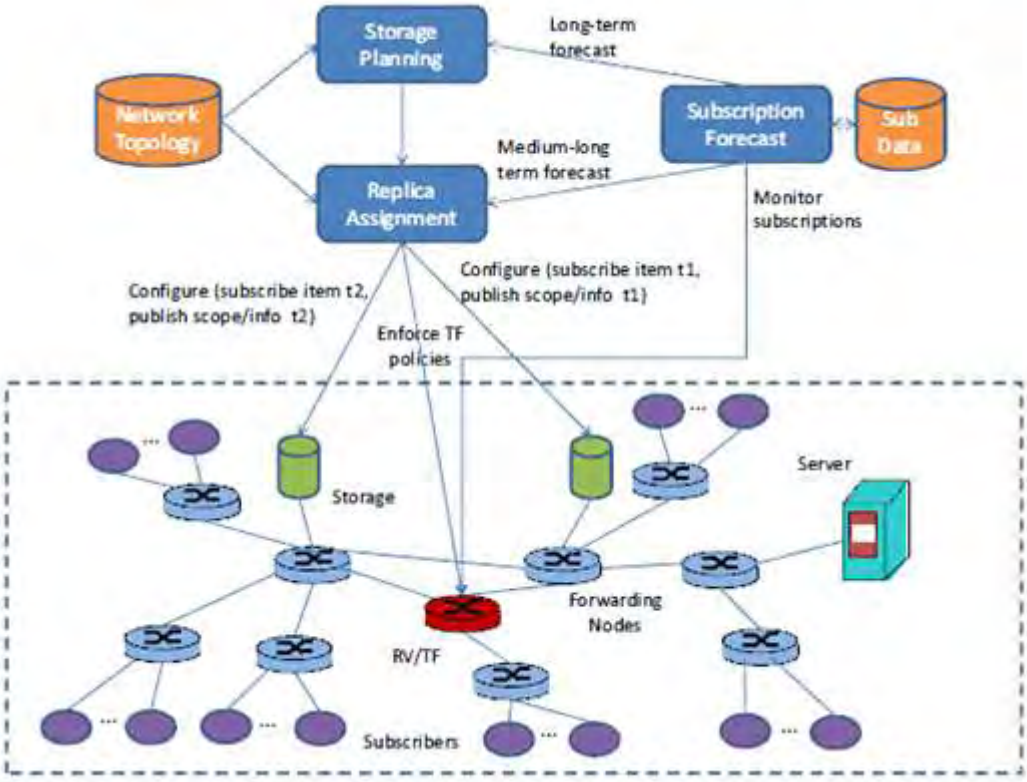
3.3. Αποθήκευση στην Πληροφοριο-Κεντρική Αρχιτεκτονική

Προκειμένου να υποστηριχτεί αποτελεσματική αποθήκευση με ειδικές συσκευές αποθήκευσης που συνδέονται με τους κόμβους σε ένα ICN, οι δύο μηχανισμοί / φάσεις που περιγράφονται ανωτέρω, και συγκεκριμένα ο σχεδιασμός αποθήκευσης και η ανάθεση αντιγράφων (storage planning – replica assignment), θα πρέπει να παρέχονται από σχετικά λειτουργικά στοιχεία-εξαρτήματα. Αυτά τα στοιχεία-εξαρτήματα συνήθως διαμένουν εκτός του δικτύου και τρέχουν off-line αλγόριθμους σε δύο διαφορετικές αλλά μακρο-μεσο χρονικές κλίμακες. Τα δομικά στοιχεία και οι αλληλεπιδράσεις τους απεικονίζονται στην εικόνα 9.

Η συνιστώσα ***Σχεδιασμού Αποθήκευσης*** παίρνει ως είσοδο το πλήθος των διαθέσιμων κόμβων αποθήκευσης που ο χειριστής επιθυμεί να εγκαταστήσει, την τοπολογία του δικτύου και μια μακροπρόθεσμη πρόβλεψη των συνδρομών στο δίκτυο. Μπορεί να τρέξει περιοδικά αποφασίζοντας για τη βέλτιστη τοποθέτηση των αποθηκών σε ένα μακροπρόθεσμο χρονικό διάστημα (π.χ. μία φορά το χρόνο) ή όταν η τρέχουσα θέση των κόμβων αποθήκευσης οδηγεί σε αναποτελεσματική ανάπτυξη λόγω σημαντικών αλλαγών συνδρομών που δεν έχουν προβλεφθεί με επιτυχία από την συνιστώσα Πρόγνωση Εγγραφέων.

Η εκτέλεση και η επιβολή της απόφασης του στοιχείου σχεδιασμού περιλαμβάνει συνήθως υψηλού επιπέδου επιχειρηματικές αποφάσεις, δεδομένου ότι υπάρχει ένα υψηλό κόστος που συνδέεται με τη μεταφορά μιας συσκευής αποθήκευσης σε μια διαφορετική φυσική τοποθεσία ή την επέκταση του αριθμού των συσκευών αποθήκευσης.

Η συνιστώσα ***Ανάθεσης Αντιγράφων*** τρέχει επίσης περιοδικά, αλλά σε μεσοπρόθεσμη / μακροπρόθεσμη κλίμακα και είναι σε θέση να επιβάλλει αυτόματα την απόφασή της στο δίκτυο. Παίρνει ως είσοδο το αποτέλεσμα του στοιχείου Σχεδιασμού Αποθήκευσης σχετικά με τις θέσεις των συσκευών αποθήκευσης που είναι εγκατεστημένες στο δίκτυο, τη φυσική τοπολογία του δικτύου και τη μεσοπρόθεσμη / μακροπρόθεσμη πρόβλεψη. Η μετεγκατάσταση των αντιγράφων μπορεί να εκτελεστεί καθοδηγώντας τους κόμβους αποθήκευσης να εγγράφονται σε ένα διαφορετικό σύνολο αντικείμενων πληροφορίας.
Η καθοδήγηση πραγματοποιείται σαν μια δημοσίευση ενός αντικείμενου πληροφορίας στο οποίο έχουν εγγραφεί οι κόμβοι αποθήκευσης.



Εικόνα 9 : Αρχιτεκτονική Αποθήκευσης

Γενικά, οι κόμβοι αποθήκευσης ενεργούν τόσο ως εκδότες όπως και σαν συνδρομητές για τα αντικείμενα πληροφορίας που έχουν καθοδηγηθεί να αποθηκεύσουν. Εγγράφονται για να λαμβάνουν νέες εκδοχές των αντικειμένων , ενώ δρουν ως εκδότες για τα ίδια αντικείμενα στους ενδιαφερόμενους συνδρομητές. Με αυτόν τον τρόπο, όταν ένας πελάτης εγγράφεται σε ένα συγκεκριμένο κομμάτι πληροφοριών, η λειτουργία ραντεβού, σε συνεργασία με το σχηματισμό τοπολογίας, καθοδηγούν έναν ή περισσότερους εκδότες με βάση την πολιτική του φορέα (π.χ. οι πλησιέστερες αποθήκες στους συνδρομητές) να δημοσιοποιούν τις πληροφορίες για αυτό το αντικείμενο. Δεδομένου ότι οι περίοδοι προβλέψεων μπορούν να είναι αρκετά μεγάλες (π.χ. μία φορά την ημέρα ή την εβδομάδα), οι συνδρομές μπορούν να διαφέρουν σημαντικά κατά τη διάρκεια αυτής της περιόδου. Για το λόγο αυτό, γίνεται επίκληση στη συνιστώσα μια φορά σε κάθε περίοδο προβλέψεων, αλλά υπολογίζει πολλαπλές ρυθμίσεις, οι οποίες μπορούν να επιβληθούν σε κάθε περίοδο πρόβλεψης. Παραδείγματα περιόδων πρόβλεψης θα μπορούσαν να είναι οι καθημερινές και τα σαββατοκύριακα, ή ακόμη και εντός της ημέρας μεταξύ των κορυφαίων και αντίθετα κορυφαίων ωρών , όταν οι συνδρομητές μπορούν να αλλάξουν σημαντικά. Η μετάβαση από τη μια ρύθμιση στην άλλη συνεπάγεται κάποιο κόστος που πρέπει να λαμβάνεται υπόψη όταν αποφασίζεται αν θα στραφεί σε διαφορετική ρύθμιση ή όχι.

Τέλος, η συνιστώσα ***Πρόγνωσης Εγγραφών*** παρακολουθεί τις συνδρομές που λαμβάνουν χώρα στο στοιχείο των ραντεβού και δημιουργεί τις προβλέψεις για τις διάφορες περιόδους για το στοιχείο ***Ανάθεσης Αντιγράφων*** καθώς και τις μακροπρόθεσμες προβλέψεις για τη συνιστώσα ***Σχεδιασμού***.

ΚΕΦΑΛΑΙΟ 4 - Αλγόριθμος Planning-Assignment

4.1.Αλγόριθμοι Τοποθέτησης

Στην ενότητα αυτή, παρουσιάζουμε μια σειρά αλγορίθμων για την επίλυση του προβλήματος των ελάχιστων k-median . Ο στόχος είναι να ελαχιστοποιηθεί το συνολικό κόστος όλων των αιτημάτων. Ορίζουμε το κόστος της αίτησης από τον κόμβο i στον κόμβο j ως την απόσταση μεταξύ των δύο κόμβων, όπου η απόσταση μπορεί να αναφέρεται σε κάθε μετρικό απόδοσης που θέλουμε να βελτιστοποιήσουμε , όπως η καθυστέρηση, μέτρηση βημάτων(hops), ή το οικονομικό κόστος της διαδρομής μεταξύ δύο κόμβων (υποθέτοντας ότι υπάρχει ένα κόστος σχετικό με τις συνδέσεις των μονοπατιών).
Οι αλγόριθμοι λειτουργούν το ίδιο ανεξάρτητα από το τι μετρική χρησιμοποιείται.

A. Tree-based Αλγόριθμος

Οι Li et al. πρότεινε έναν αλγόριθμο τοποθέτησης [13] με βάση την παραδοχή ότι η βασικές τοπολογίες είναι δέντρα, και το μοντελοποίησε ως ένα δυναμικό πρόβλημα προγραμματισμού. Ο αλγόριθμος αρχικά σχεδιάστηκε για την τοποθέτηση cache του Web Proxy, και εφαρμόζεται επίσης για την τοποθέτηση αντιγράφων Web. Σε πολύ υψηλό επίπεδο, διαιρούν ένα δέντρο T σε πολλά μικρά δέντρα Ti, και δείχνουν ότι ο καλύτερος τρόπος για την τοποθέτηση t> 1 πληρεξούσιων(proxies) στο δέντρο T ,είναι να τοποθετήσουν ti πληρεξούσια(proxies) με τον καλύτερο τρόπο σε κάθε μικρό δέντρο Ti, όπου Σti=t. Ο αλγόριθμος φαίνεται να βρίσκει μια βέλτιστη τοποθέτηση, όταν οι υποκείμενες τοπολογίες είναι δέντρα, και οι πελάτες ζητούν από τον πληρεξούσιο(proxy) στο μονοπάτι προς το διακομιστή Web, δηλαδή, οι πελάτες δεν μπορούν να ζητήσουν από έναν παρόμοιο πληρεξούσιο(proxy) Ωστόσο, αυτές οι δύο υποθέσεις μπορούν ενδεχομένως να φέρουν καλύτερες επιλογές τοποθέτησης. Οι βέλτιστες λύσεις στο πλαίσιο των υποθέσεων αυτών, δεν είναι συνήθως τόσο καλές όσο οι λύσεις που βρέθηκαν από τον greedy και hot ευρετικές σημείου (χωρίς τις υποθέσεις), οι οποίες περιγράφονται αργότερα σε αυτό το τμήμα.

Β. Άπληστος Αλγόριθμος

Η βασική ιδέα του άπληστου (greedy) αλγορίθμου είναι ως εξής. Υποθέτοντας πως πρέπει να επιλέξουμε Μ αντίγραφα ανάμεσα σε Ν πιθανές θέσεις . Επιλέγουμε ένα αντίγραφο τη φορά. Στην πρώτη επανάληψη, αξιολογούμε κάθε μία από τις πιθανές θέσεις Ν ξεχωριστά για να προσδιοριστεί η καταλληλότητα της για τη φιλοξενία ενός αντιγράφου. Υπολογίζουμε το κόστος που σχετίζεται με την κάθε θέση, με την παραδοχή ότι η πρόσβαση από όλους τους πελάτες συγκλίνει σε αυτή την θέση, και επιλέγουμε την θέση εκείνη που παράγει το χαμηλότερο κόστος. Στη δεύτερη επανάληψη, ψάχνουμε για μια δεύτερη θέση αντιγράφου η οποία, σε συνδυασμό με αυτήν που έχουμε ήδη διαλέξει, αποδίδει το χαμηλότερο κόστος. Σε γενικές γραμμές, κατά τον υπολογισμό του κόστους, υποθέτουμε ότι οι πελάτες τους κατευθύνουν την πρόσβασή τους προς το πλησιέστερο αντίγραφο (δηλαδή, κάποιο που μπορεί να προσεγγιστεί με το χαμηλότερο κόστος).Επαναλαμβάνουμε τη διαδικασία μέχρι να έχουμε επιλέξει Μ αντίγραφα .

Γ. Τυχαίος Αλγόριθμος

Ο τυχαίος αλγόριθμος αγνοεί το φόρτο εργασίας πελάτη, και επιλέγει τυχαία Μ αντίγραφα μεταξύ των πιθανών Ν θέσεων μέσα από μια ομοιόμορφη κατανομή. Για να βελτιώσουμε την απόδοση του, εκτελούμε τον αλγόριθμο αρκετές φορές - στις προσομοιώσεις μας, εκτελούμε πάνω από 10 φορές, και διαλέγουμε την τυχαία ανάθεση που παράγει το χαμηλότερο κόστος.

Δ. Hot Spot

Ο αλγόριθμος Hot Spot προσπαθεί να τοποθετήσει αντίγραφα κοντά στους πελάτες που παράγουν το μεγαλύτερο φορτίο. Ταξινομεί τις Ν πιθανές θέσεις ανάλογα με το ποσό της κίνησης που δημιουργείται στο περιβάλλον τους. Τοποθετεί τα αντίγραφα στις κορυφαίες Μ θέσεις που παράγουν το μεγαλύτερο ποσό της κίνησης. Ορίζουμε τη γειτνίαση του Α, ως τον κύκλο με κέντρο το Α και κάποια ακτίνα. Στις προσομοιώσεις μας, μεταβάλλουμε την ακτίνα από 0 έως τη μέγιστη απόσταση μεταξύ κάθε ζεύγους κόμβων στο γράφημα, και αναφέρουμε την καλύτερη απόδοση σε σχέση με όλες τις ακτίνες που έχουν δοκιμαστεί.

Tree-based [16]	Greedy	Random	Hot Spot
$O(N^3M^2)$	$O(N^2M)$	$O(NM)$	$N^2 + min(NlognN + NM)$

TABLE I

COMPARISON OF COMPUTATIONAL TIME OF VARIOUS ALGORITHMS

Πίνακας 1.

Ο Πίνακας 1 παραθέτει τον υπολογιστικό χρόνο διάφορων αλγορίθμων για την επιλογή Μ αντιγράφων μεταξύ των Ν πιθανών θέσεων. Αν μόνο μια χούφτα πιθανών θέσεων hosts είναι διαθέσιμες, το κόστος των υπολογιστικά πολύπλοκων αλγορίθμων μπορεί να μην είναι σημαντικό. Ωστόσο, στην ανάλυσή μας, θεωρούμε ομάδες που ορίζονται από προθέματα διευθύνσεων ως πιθανές θέσεις αντιγράφων. Στην περίπτωση αυτή, το Ν είναι της τάξης των 100.000 (ο αριθμός των προθεμάτων διευθύνσεων στο Internet), οπότε η υπολογιστική πολυπλοκότητα του αλγορίθμου τοποθέτησης αντιγράφων γίνεται σαφώς πολύ σημαντική. Για να μειωθεί το υπολογιστικό κόστος, θεωρούμε μόνο τις πρώτες, από την άποψη των αιτήσεων που παράγονται, εκατοντάδες ή χιλιάδες συστάδες. Δεδομένου ότι αυτές οι πρώτες συστάδες παράγουν το μεγαλύτερο μέρος της κίνησης(φόρτου), το να αγνοήσεις τα αιτήματα από μη δημοφιλείς συστάδες έχει μικρή επίδραση στα αποτελέσματα.

4.2. Στρατηγική Τοποθέτησης και Ανάθεσης Αντιγράφων

Χρησιμοποιούμε ως βάση του σχήματος τοποθέτησης και ανάθεσης αντιγράφων, αλγόριθμους που παρουσιάστηκαν στο πλαίσιο των CDN δικτύων. Ιδιαίτερα αναπτύχθηκαν αρκετοί αλγόριθμοι τοποθέτησης, [15] και [16], που χρησιμοποιούν πληροφορίες για το φόρτο εργασίας, όπως η καθυστέρηση (απόσταση από τα σημεία αποθήκευσης) και ποσοστά αιτήσεων ,για να πάρουν την απόφαση τοποθέτησης. Βασικό τους συμπέρασμα είναι ότι ο ονομαζόμενος "άπληστος" αλγόριθμος που τοποθετεί αποθήκες που βασίζονται τόσο σε ένα μετρικό απόστασης και στο φόρτο των αιτημάτων, παρουσιάζει την καλύτερη και πολύ κοντά στη βέλτιστη λύση.

4.2.1. Άπληστος αλγόριθμος

Εδώ, παρουσιάζουμε σύντομα τον άπληστο αλγόριθμο υποθέτοντας ότι υπάρχει μόνο μια κατηγορία περιεχομένου στο σύστημα μας (ένα θέμα), ή ισοδύναμα εκεί δεν υπάρχει διάκριση όσον αφορά το περιεχόμενο. θεωρούμε ως ri τη ζήτηση (σε αιτ. / sec)από τους πελάτες που συνδέονται με τον κόμβο i. Επίσης, θεωρούμε pij ως το ποσοστό της συνολικής ζήτησης αιτημάτων που προσεγγίζουν το διακομιστή (server) στόχο i (παραδοσιακοί αλγόριθμοι τοποθέτησης αντιγράφουν ένα συγκεκριμένο διακομιστή προέλευσης) και περνούν μέσα από τον κόμβο i. Επίσης θεωρούμε την καθυστέρηση διάδοσης (άλματα) από τον κόμβο i στον διακομιστή προορισμούj, ως dij. Εάν μια αποθήκη τοποθετείται στο κόμβο i ορίζουμε το κέρδος να είναι gij = pij · dij. Αυτό σημαίνει ότι pij ποσοστό της κυκλοφορίας δεν χρειάζεται να διασχίσει την απόσταση από τον κόμβο i στον διακομιστή j μειώνοντας τη συνολικής κίνηση του δικτύου κατά:

$$d_{ij} \cdot \sum_{l=1}^{N} R_l$$

$$R_l = \begin{cases} r_l & \text{if } i \text{ is on the path from } l \text{ to } j \\ 0 & \text{otherwise.} \end{cases}$$

Ο άπληστος αλγόριθμος επιλέγει μια αποθήκη κάθε στιγμή (χρειαζόμαστε k αποθήκες από τους N κόμβους του δικτύου). Στον πρώτο γύρο, αξιολογεί καθέναν από τους κόμβους N να προσδιορίσει την καταλληλότητά του για να γίνει αποθήκη (σημείο αντιγραφής του διακομιστή i). Υπολογίζει το Κέρδος που σχετίζεται με κάθε κόμβο και επιλέγει αυτό που μεγιστοποιεί το κέρδος. Στο δεύτερο γύρο, ψάχνει για μια δεύτερη αποθήκη που, σε συνδυασμό με την αποθήκη που έχει ήδη επιλεγεί, αποδίδει το υψηλότερο κέρδος. Ο άπληστος αλγόριθμος επαναλαμβάνεται μέχρι να έχουν επιλεγεί k αποθήκες να αντιγράψουν το διακομιστή i.

4.2.2. Τροποποιημένος άπληστος αλγόριθμος

Στην αρχιτεκτονική δικτύου pub/sub που υποθέτουμε σε αυτό το κείμενο, η έννοια του διακομιστή προέλευσης - το οποίο είναι ζωτικής σημασίας για τον άπληστο αλγόριθμο – δεν υπάρχει. Οι εκδότες συνδέονται στο δίκτυο, δημοσιεύουν το περιεχόμενό τους και εξαφανίζονται. Έτσι ,για να αποκτήσουμε τη θέση των αποθηκών τροποποιούμε τον άπληστο αλγόριθμο.

Συγκεκριμένα , επαναλαμβάνουμε την παραπάνω διαδικασία N φορές υποθέτοντας κάθε φορά ότι ο διακομιστής προορισμού είναι ένας διαφορετικός κόμβος (broker) του δικτύου. Με αυτόν τον τρόπο παίρνουμε N διανύσματα k πιθανών αποθηκών. Ακριβώς, κάθε διάνυσμα έχει N στοιχεία με k άσσους στους δείκτες των επιλεγμένων αποθηκών και N-k μηδενικά σε κάθε άλλη θέση. Για παράδειγμα, το διάνυσμα [0 0 0 1 0 1] σημαίνει ότι από 6 οι κόμβους του δικτύου οι επιλεγμένοι k = 2 πιθανές αποθήκες είναι οι κόμβοι 4 και 6.

Τέλος, επιλέγουμε ως αποθήκες μας τους κόμβους k που εμφανίστηκαν περισσότερες φορές στο ανά στοιχείο άθροισμα των N διανυσμάτων και εγκαθιστούμε σε κάθε έναν μια αποθήκη. Ο τροποποιημένος άπληστος αλγόριθμος που παρουσιάζεται εδώ προϋποθέτει ομοιόμορφη κατανομή της πιθανότητας μεταξύ των N κόμβων του δικτύου ότι μπορούν να συμβούν δημοσιεύσεις. Φυσικά άλλες μορφές κατανομών πιθανοτήτων θα μπορούσε να χρησιμοποιηθεί, και κάθε διάνυσμα θα πρέπει πρώτα να σταθμίζεται με την πιθανότητα του πριν από την ανά στοιχείο άθροιση των διανυσμάτων N.

4.2.3. Αλγόριθμος Τοποθέτησης και Ανάθεσης Αντιγράφων για Δίκτυα Δημοσιεύσεων/Συνδρομών

r_i^t	: request rate for topic $t \in T$ in broker i
N	: number of nodes (brokers) in the network
M	: $(M < N)$ number of storages in the network
k_t	: $(k_t \leq M)$ replication of each topic $t \in T$ in the network
L	: storage capacity of each storage point in the network
T	: number of classes of content (topics)
w_t	: weight of topic $t \in T$ in the network
$\mathcal{S}$	: storage brokers vector
s_t	: possible stores vector for topic $t \in T$
ξ_t	: relative weight of topic $t \in T$

Πίνακας 2: Παράμετροι που χρησιμοποιούνται από τον Αλγόριθμο

Εδώ, χρησιμοποιούμε τον τροποποιημένο άπληστο αλγόριθμο που περιγράφεται παραπάνω για την περίπτωση όπου στο δίκτυο μας υπάρχουν T διαφορετικές κατηγορίες περιεχομένου (θέματα). Στη συνέχεια, παρουσιάζουμε τα βήματα του προτεινόμενου αλγόριθμου ένα-ένα με το παράδειγμα της εικόνας 10 (ο Πίνακας 1 περιέχει όλες τις χρήσιμες παραμέτρους που απαιτούνται από την προτεινόμενη αλγόριθμο):

1) Για κάθε θέμα t ∈ T εκτελούμε τον τροποποιημένο άπληστο αλγόριθμο που παρουσιάστηκε στην προηγούμενη ενότητα και παίρνουμε T διανύσματα των πιθανών αποθηκών St. Όσον αφορά το παράδειγμα έχουμε:

   sa = [0 3 5 0 2 2]
   sb = [0 2 5 0 5 0]
   sc = [0 2 5 0 5 0]

   για τα τρία θέματα αναλόγως. Το [0 3 5 0 2 2] σημαίνει ότι από τις N = 6 εκτελέσεις του τροποποιημένου άπληστου αλγορίθμου, ο κόμβος 2 εμφανίστηκε 3 φορές, ο κόμβος 3 εμφανίστηκε 5 φορές και ούτω καθεξής.

2) Κάθε διάνυσμα (st) σταθμίζεται από το:

$$w_t = \frac{\sum_{i=1}^{N} r_i^t}{\sum_{i=1}^{N}\sum_{t=1}^{T} r_i^t}.$$

Το wt. δείχνει τη σημασία (δημοτικότητα) σχετικά με τη ζήτηση της κυκλοφορίας του κάθε θέματος στο δίκτυο. Οι συντελεστές στάθμισης για το συγκεκριμένο παράδειγμα είναι οι εξής:

wa = 17/50 = 0.34, wb = 27/50 = 0.54, wc = 6/50 = 0.12

Λαμβάνουμε τα ακόλουθα σταθμισμένα διανύσματα :

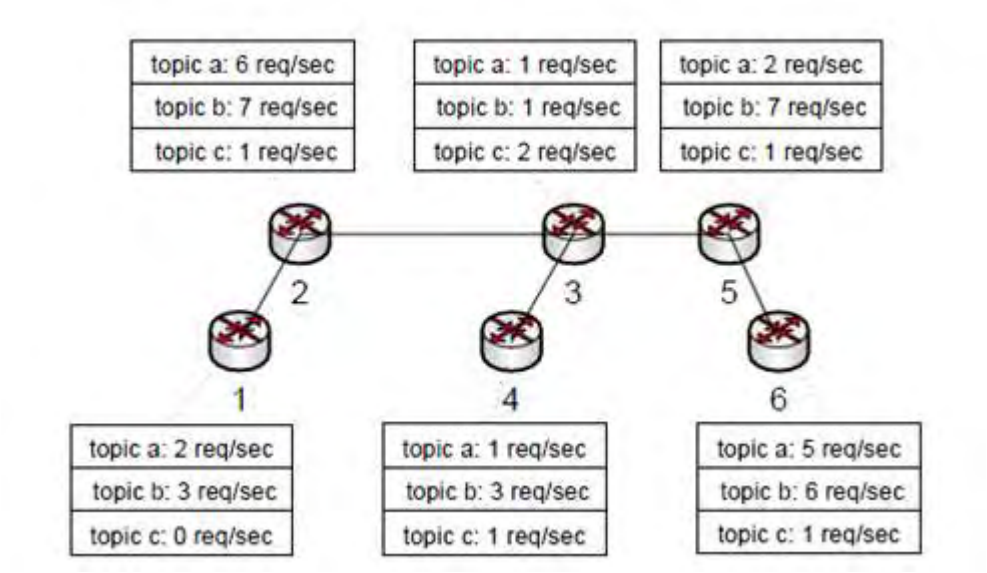
wa · sa = [0 1.02 1.7 0 0.68 0.68]

wb · sb = [0 1.08 2.7 0 2.7 0]

wc · sc = [0 0.24 0.6 0 0.6 0].

3) Επιλέγουμε ως αποθήκες μας τα κόμβους M που εμφανίστηκαν περισσότερες φορές στην ανά στοιχείο σταθμισμένη άθροιση των T διανυσμάτων. Καλούμε αυτό το διάνυσμα, ως storage brokers vector S. Η άθροιση ανά στοιχείο των πιο πάνω τριών διανυσμάτων σε ένα μοναδιαίο διάνυσμα δίνει
[0 2,34 5 0 3,98 0,68] πράγμα που σημαίνει ότι οι τελικές M = 3 αποθήκες στο S είναι οι κόμβοι 3, 5 και 2.

4) Για κάθε θέμα t, ξεκινώντας από το πιο σημαντικό (με βάση το βάρος), αναθέτουμε kt αποθήκες κυκλικά, ακολουθώντας την παρακάτω διαδικασία:
   - Για κάθε εγγραφή στο st του θέματος t που έχει υπολογιστεί στο βήμα 1 ανάθεσε μια αποθήκη τη φορά, εάν η καταχώρηση εμφανίζεται επίσης στο S, το οποίο υπολογίστηκε στο βήμα 3, και μόνο αν στην εν λόγω αποθήκη έχει ανατεθεί κάτω από L (χωρητικότητα αποθήκης) θέματα μέχρι να έχουμε kt αποθήκες (αναπαραγωγή του θέματος t).



Εικόνα 10 : Τοπολογία δικτύου και πληροφορία φόρτου εργασίας(αιτήσεις/δευτερόλεπτο) για κάθε κλάση περιεχομένων .Είσοδοι για τον αλγόριθμο Τοποθέτησης για το pub/sub δίκτυο (T = 3 , k = 2 , SC = 2 και M = 3)

Στο παράδειγμα ξεκινώντας από το θέμα b , έπειτα το θέμα a και τέλος το θέμα c(με βάση το βάρος τους) τα τοποθετούμε σε k = 2 αποθήκες. Το θέμα b ανατίθεται στους κόμβους 3 και 5, που ήταν οι κόμβοι στους οποίους το θέμα b εμφανίστηκε περισσότερες φορές στο Βήμα 1. Το θέμα a ανατίθεται επίσης στους κόμβους που παράχθηκαν από το βήμα 1, τους κόμβοι 2 και 3, ενώ το θέμα c ανατίθεται στους κόμβους 2 και 5. Ο κόμβος 5 ήταν από τις πιο δημοφιλείς επιλογές που παράχθηκε στο Βήμα 1, ενώ ο κόμβος 2 ήταν η μόνη αποθήκη στο S με λιγότερο από το L = 2 αναθέσεις.

Το Βήμα 4 του αλγορίθμου μας είναι επίσης γνωστό ως **Γενικευμένο Πρόβλημα Ανάθεσης** το οποίο ακόμα και στην απλούστερη μορφή του μειώνεται σε NP-complete πρόβλημα πολλαπλών σακιδίων.

4.2.4. Μοντέλο Κόστους

Τα βήματα 1-3 του προτεινόμενου αλγορίθμου που περιγράφεται παραπάνω περιλαμβάνουν τη φάση Σχεδιασμού του αλγορίθμου, ενώ το βήμα 4 είναι η φάση Ανάθεσης.
Στο κεφάλαιο αυτό, παρουσιάζουμε το μοντέλο κόστους [24] της φάσης Ανάθεσης, η οποία όπως προαναφέρθηκε νωρίτερα είναι NP-complete πρόβλημα. Η πρόσβαση ενός αντικειμένου πληροφορίας που είναι αποθηκευμένο στην αποθήκη x από τον κόμβο y δημιουργεί ένα φορτίο κυκλοφορίας ίσο με το μήκος (αριθμός hops) του μονοπατιού από το x στο y. Δεδομένου ότι θέλουμε να βελτιστοποιήσουμε το συνολικό φορτίο κυκλοφορίας, το σύστημα πρόσβασης είναι ότι έχουμε πρόσβαση πάντα στην πιο κοντινή αποθήκη (συντομότερο μονοπάτι) μεταξύ εκείνων που κατέχουν το συγκεκριμένο αντικείμενο.
Έτσι, δεδομένου του μηχανισμού πρόσβασης, θα επιδιώξουμε να αποφασίσουμε την ανάθεση αντίγραφων του κάθε θέματος.

Έστω C, το φορτίο κυκλοφορίας που αντιστοιχεί σε κάθε ρύθμιση αποθήκης.
Για αυτή τη ρύθμιση της αποθήκης μπορούμε να γράψουμε:

$$C = \sum_{t=1}^{T} C_t \qquad (1)$$

όπου Ct είναι το κυκλοφοριακός φορτίο που αντιστοιχεί στη διαμόρφωση του t θέματος μόνο. Τότε έχουμε,

$$C_t = \sum_{n=1}^{k_t} \sum_{l \in \mathcal{N}_n^t} r_l^t \cdot d_{ln} \qquad (2)$$

όπου N_n^t, είναι η συλλογή των κόμβων που έχουν πρόσβαση στο αντικείμενο t από το σημείο αναπαραγωγής του στον κόμβο n, r_l^t είναι ο ρυθμός αιτήσεων για το θέμα t από τον κόμβο l και d_{ln} είναι η απόσταση (σε hops) από τον κόμβο l στον κόμβο n.

Και για το σύνολο της κυκλοφορίας του δικτύου (overall network traffic) από τις Εξισώσεις 1 και 2, έχουμε:

$$C = \sum_{t=1}^{T} C_t = \sum_{t=1}^{T} \sum_{n=1}^{k_t} \sum_{l \in \mathcal{N}_n^t} r_l^t \cdot d_{ln} \tag{3}$$

Η ελαχιστοποίηση του συνολικού κόστους κυκλοφορίας δίνεται από την ελαχιστοποίηση της ακόλουθης περιορισμένησ μη γραμμικής συνάρτησης πολλών μεταβλητών.

$$\min\left(C\left(k_1, k_2, \ldots, k_t\right)\right) \text{ such that } \begin{cases} \sum_{t=1}^{T} k_t \leq L \cdot M \\ 1 \leq k_t \leq M, \forall t \in T \end{cases} \tag{4}$$

ΚΕΦΑΛΑΙΟ 5- Υλοποίηση Αλγορίθμου

5.1. Τεχνολογίες

Όσον αφορά την υλοποίηση του αλγορίθμου για Storage Planning και Replica Assignment , έγινε χρησιμοποιώντας το περιβάλλον προγραμματισμού NetBeans , σε γλώσσα προγραμματισμού Java και κάνοντας χρήση πακέτων της βιβλιοθήκης JUNG

5.1.1. NetBeans

Το NetBeans [26] είναι ένα open-source ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment IDE) για την Java, PHP, C + + και άλλες γλώσσες προγραμματισμού. Είναι επίσης γνωστό και ως μια πλατφόρμα αποτελούμενη από αρθρωτά εξαρτήματα που χρησιμοποιούνται για την ανάπτυξη εφαρμογών Java desktop. Επιπλέον είναι γραμμένο σε Java και τρέχει στα περισσότερα συστήματα που λειτουργούν με Java Virtual Machine (JVM), συμπεριλαμβανομένων των Solaris, Mac OS και Linux.
Διαχειρίζεται τα ακόλουθα χαρακτηριστικά και συστατικά πλατφόρμας:

- User settings
- User interface
- Windows (placement, appearance, etc.)
- NetBeans Visual Library
- Storage
- Integrated development tools
- Framework wizard

Επιπλέον η πλατφόρμα NetBeans επιτρέπει στις εφαρμογές να αναπτυχθούν από ένα σύνολο ανεξάρτητων κομματιών λογισμικού που ονομάζεται ενότητες. Οι εφαρμογές που βασίζονται στην πλατφόρμα NetBeans μπορεί να επεκταθούν από τρίτους προγραμματιστές .
Τέλος οι NetBeans IDE ενότητες περιλαμβάνουν NetBeans Profiler, μια γραφική διεπαφή χρήστη (GUI) εργαλείο σχεδιασμού και το NetBeans JavaScript Editor.

Σημαντικό εργαλείο που χρησιμοποιήσαμε για την υλοποίηση του αλγορίθμου είναι η βιβλιοθήκη JUNG , μέσω των συναρτήσεων που έχουν να κάνουν με τις τοπολογίες/γράφους των διαφόρων δικτύων μας και την εύρεση των συντομότερων μονοπατιών.

Χάρη στη βιβλιοθήκη της JUNG, καταφέραμε να μεταφράσουμε και να κωδικοποιήσουμε τα αρχεία τύπου τοπολογίας που περιείχαν τις πληροφορίες των συνδέσεων των κόμβων των δικτύων, σε μεταβλητές κλάσης, και στη συνέχεια να συλλέξουμε και να επεξεργαστούμε τα δεδομένα των δικτύων μέσω των μεταβλητών αυτών.

5.1.2. JUNG — the Java Universal Network/Graph Framework

Είναι μια βιβλιοθήκη λογισμικού [27] που παρέχει μια κοινή και έκτακτη γλώσσα για την μοντελοποίηση, ανάλυση και οπτικοποίηση δεδομένων που μπορούν να αναπαρασταθούν ως ένα γράφημα ή ένα δίκτυο. Είναι γραμμένη σε Java, το οποίο επιτρέπει JUNG-based εφαρμογές να κάνουν χρήση των εκτεταμένων ενσωματωμένων δυνατοτήτων του Java API, καθώς και εκείνων των άλλων υφιστάμενων τρίτων βιβλιοθηκών της Java.

Η αρχιτεκτονική JUNG έχει σχεδιαστεί για να υποστηρίζει μια ποικιλία από αναπαραστάσεις των οντοτήτων και των σχέσεών τους, όπως οι κατευθυνόμενοι και μη κατευθυνόμενοι γράφοι, πολυτροπικοί γράφοι, γράφοι με παράλληλες ακμές, και υπερ-γράφοι(hypergraphs). Παρέχει ένα μηχανισμό για το σχολιασμό γράφων, οντοτήτων και σχέσεων με μεταδεδομένα. Αυτό διευκολύνει τη δημιουργία αναλυτικών εργαλείων για πολύπλοκα σύνολα δεδομένων που μπορούν να εξετάσουν τις σχέσεις μεταξύ των οντοτήτων καθώς και τα μεταδεδομένα που συνδέονται με κάθε οντότητα και σχέση. Η πρόσφατη διανομή της JUNG περιλαμβάνει υλοποιήσεις ενός αριθμού αλγορίθμων από τη θεωρία γράφων, την εξόρυξη δεδομένων, καθώς και ανάλυση των κοινωνικών δικτύων, όπως ρουτίνες για την ομαδοποίηση, την αποσύνθεση, τη βελτιστοποίηση, την τυχαία παραγωγή γράφων, τη στατιστική ανάλυση, και τον υπολογισμό αποστάσεων , ροών, και μέτρων σπουδαιότητας ενός δικτύου (κεντρικότητα, PageRank, HITS, κλπ.).

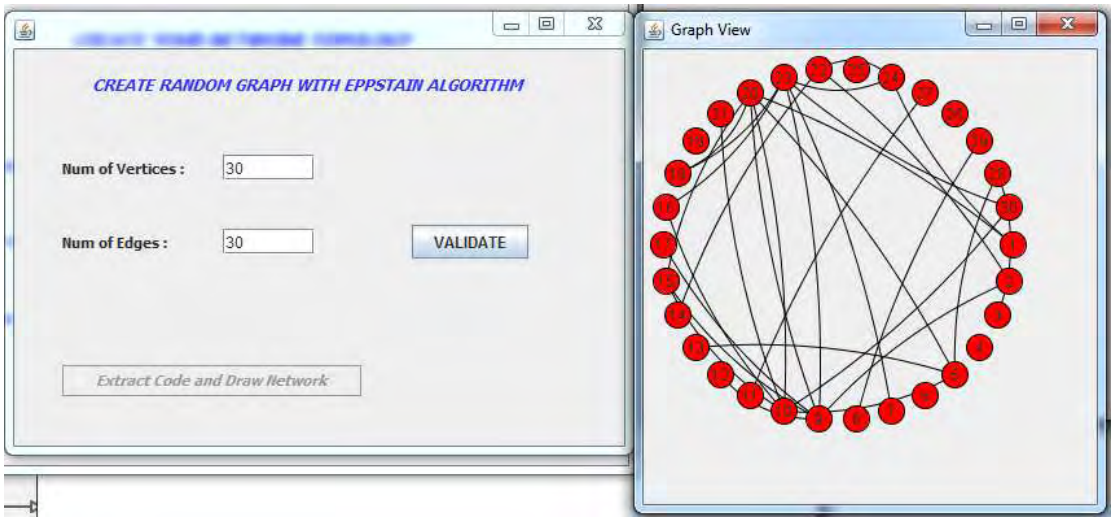
Η JUNG παρέχει επίσης ένα πλαίσιο οπτικοποίησης που καθιστά εύκολη την κατασκευή εργαλείων για την διαδραστική εξερεύνηση δεδομένων του δικτύου. Οι χρήστες μπορούν να χρησιμοποιήσουν έναν από τους αλγόριθμους εμφάνισης που παρέχονται , ή να χρησιμοποιήσουν το πλαίσιο για να δημιουργήσουν τις δικές τους προσαρμοσμένες εμφανίσεις. Επιπλέον, παρέχονται μηχανισμοί φιλτραρίσματος που επιτρέπουν στους χρήστες να εστιάσουν την προσοχή τους, ή τους αλγόριθμους τους, σε συγκεκριμένα τμήματα του γραφήματος. Ως μια open-source βιβλιοθήκη, η JUNG παρέχει ένα κοινό πλαίσιο για την ανάλυση και την απεικόνιση γράφων/δικτύων. Ελπίζουμε ότι η JUNG θα καταστήσει ευκολότερο για εκείνους που εργάζονται με σχεσιακά δεδομένα να κάνουν χρήση των προσπαθειών ανάπτυξης κάποιων άλλων, και έτσι να αποφευχθεί η συνεχής ξανα-ανακάλυψη του τροχού.

5.2. Δομή Κώδικα

Για την σχεδίαση του αλγορίθμου μας, πρέπει αρχικά να γνωρίζουμε την τοπολογία του δικτύου που μας ενδιαφέρει, δηλαδή τον αριθμό των κόμβων του δικτύου κατά κύριο λόγο. Επίσης πρέπει να γνωρίζουμε τις αιτήσεις που γίνονται από τους πελάτες του δικτύου σε κάθε κόμβο , για κάθε θέμα ξεχωριστά.

Χρησιμοποιούμε το πρόγραμμα Network_topology έτσι ώστε να παράγουμε το εκάστοτε δίκτυο που θέλουμε, με άλλα λόγια είναι μια γεννήτρια τυχαίων τοπολογιών με αριθμό κόμβων που ορίζουμε εμείς οι ίδιοι.

Εικόνα 11 : Στιγμιότυπο εκτέλεσης του προγράμματος για τον σχεδιασμό τυχαίου δικτύου



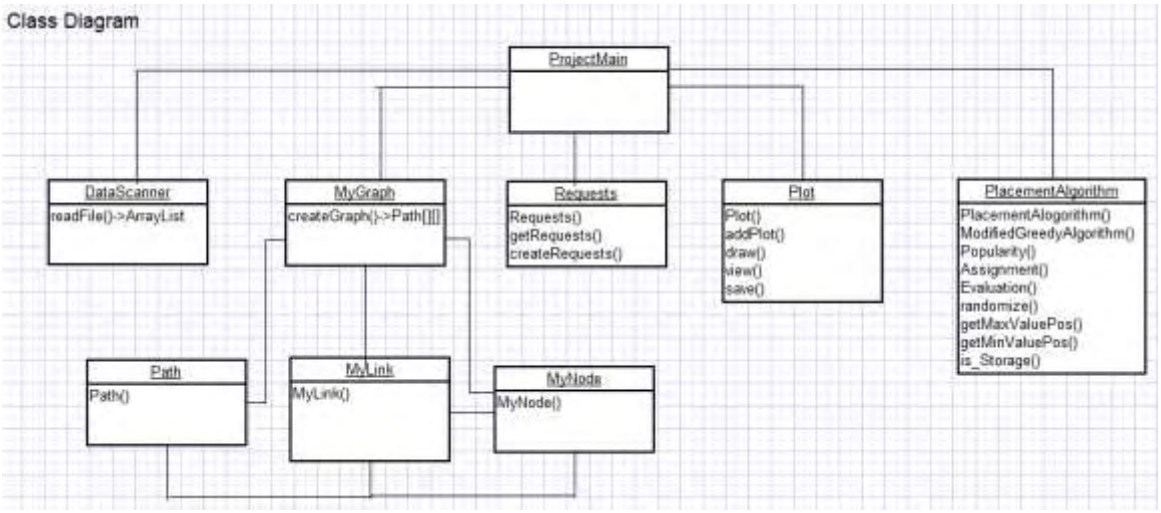
Εικόνα 12 : Στιγμιότυπο εκτέλεσης όπου επιλέγουμε τη δημιουργία δικτύου 30 κόμβων που συνδέονται με 30 ακμές μεταξύ τους

Όσον αφορά τις αιτήσεις των πελατών σε κάθε κόμβων , χρησιμοποιούμε μια συνάρτηση που παράγει τυχαίους αριθμούς αιτήσεων για κάθε κόμβο ανάλογα με τον αριθμό των κόμβων του κάθε δικτύου.

```
public void createRequests() throws Exception {
    File file;
    Writer writer;
    String outputf = "Requests/requests" + nodes + topics + ".txt";

    file = new File(outputf);
    writer = new BufferedWriter(new FileWriter(file));

    Random rand = new Random();

    for (i = 0; i < nodes; i++) {
        for (j = 0; j < topics; j++) {

            writer.write(rand.nextInt(nodes) + "\t");

        }
        writer.write("\n");
    }
    writer.close();
}
```

Εικόνα 13 : Κώδικας για την αρχική δημιουργία των αιτήσεων σε κάθε κόμβο του δικτύου

Για την συνέχεια της εργασίας μας, είναι χρήσιμο να παράγουμε τοπολογίες για διάφορους αριθμούς κόμβων , στην περίπτωσή μας από 10 έως 220 και για αιτήσεις πελατών που ανταποκρίνονται σε τόσους κόμβους και για αριθμό θεμάτων 3, 5 , 8 και 10. Αυτό γίνεται έτσι ώστε όταν θα τρέξουμε τα πειράματα σχετικά με την αποτελεσματικότητα του αλγορίθμου μας , να έχουμε κάποιο εύρος επιλογών στη αλλαγή διαφόρων μεταβλητών του δικτύου.



Εικόνα 14 : Κλάσεις που χρησιμοποιούνται στο πρόγραμμα

Όσον αφορά τη δομή του κώδικα, φαίνεται από την παραπάνω εικόνα.
Η διαδικασία εκτέλεσής του έχει ως εξής :

Αφού ορίσω τις μεταβλητές ανάλογα με το είδος του δικτύου και την περίπτωση που θέλω να εξετάσω, ξεκινώ διαβάζοντας από τα αρχεία με τις τοπολογίες τον αριθμό των κόμβων και τις συνδέσεις μεταξύ τους και επιστρέφω μία λίστα με όλους τους κόμβους.

```
DataScanner ds = new DataScanner();
nodes = ds.readFile("Topologies/topology" + maxSize + ".cfg");
```

Έχοντας του κόμβους και τις συνδέσεις τους, φτιάχνω το γράφο

```
MyGraph graph = new MyGraph();
paths = graph.createGraph(nodes);
```

και υπολογίζω τα καλύτερα μονοπάτια μέσω Dijkstra, τα οποία και αποθηκεύω σ΄έναν πίνακα (paths[][]).

```
List<MyLink> l = alg.getPath(nodes.get(i).id, nodes.get(j).id);
Path p = new Path(l);
paths[i][j] = p;
```

Στη συνέχεια παίρνω τα requests ανάλογα με τον αριθμό των κόμβων (τοπολογία) και τον αριθμό των θεμάτων (t).

Requests rq = new Requests(maxSize, topics);
requests = rq.getRequests();

και τρέχω το βασικό μέρος του αλγόριθμου τοποθέτησης , δηλαδή τη φάση υπολογισμού των αποθηκών και την φάση ανάθεσης των θεμάτων. Αναλυτικότερα, αφού αρχικοποιήσω τις μεταβλητές για έναν αλγόριθμο τοποθέτησης , τρέχω τον ModifiedGreedyAlgorithm και έχοντας υπολογίσει τις πιθανές αποθήκες(PS) ,απλές αρχικά και βεβαρημένες αργότερα, βρίσκω το SBV , οπότε και βρίσκω τις καλύτερες αποθήκες.

PlacementAlgorithm place = new PlacementAlgorithm(paths, requests, nodes, topics, k, m, SC);
place.ModifiedGreedyAlgorithm();
place.Popularity();

Αργότερα ,τρέχω τη διαδικασίας ανάθεσης των θεμάτων στις καλύτερες αποθήκες και υπολογίζω το ONT(overall network traffic) και το mean hop distance τα οποία και αναθέτω σε έναν πίνακα 2 θέσεων (temp) για να φτιάξω αργότερα τις γραφικές.

place.Assignment();
temp = place.Evaluation();

Τέλος κάνω τυχαία εύρεση καλύτερων αποθηκών και ανάθεση θεμάτων και ξαναυπολογίζω τα ONT και mean hop distance για τα νέα δεδομένα.

place.randomize();
temp = place.Evaluation();

Συνεπώς, οι γραφικές παραστάσεις γίνονται για την εφαρμογή του αλγορίθμου με ανάθεση Round Robin (RR), αρχίζοντας από το πιο δημοφιλές θέμα και για την περίπτωση όπου εύρεση και ανάθεση γίνονται τυχαία (Random).

Όσον αφορά τα inputs του κώδικα, δίνω τιμές τέτοιες ώστε:

k*topics < m*SC (***)

δηλαδή το σύνολα των αντιγράφων για όλα τα θέματα είναι μικρότερο ή το πολύ ίσον με τη συνολική αποθηκευτική δυνατότητα του δικτύου.

Σε περίπτωση που **k*topics >= m*SC (το ίσον είναι οριακό)** , υπάρχει περίπτωση να έχω πρόβλημα με το assignment (περίπτωση RR και Random), λόγω μη ύπαρξης θέσεως για την αποθήκευση κάποιου αντιγράφου.(είτε έχει μείνει μία θέση και ένα αντικείμενο, και ήδη υπάρχει σ' αυτή τη θέση αντικείμενο με το ίδιο θέμα , είτε υπάρχουν παραπάνω από μία θέσεις διαθέσιμες αλλά ένα αντικείμενο δεν μπορεί να ανατεθεί πουθενά καθώς σε όλες τις ελεύθερες θέσεις υπάρχει ήδη αντικείμενο του ίδιου θέματος αποθηκευμένο).

Στο πρόγραμμα, σε κάθε μία από τις περιπτώσεις(ProjectMain, ProjectMain2, ProjectMainMixed, ProjectMainMixed2) κάνω έλεγχο αν ισχύει η πρώτη σχέση (***) και αν δεν ισχύει προσπαθώ να την επιδιώξω.

Κάποιοι άλλοι περιορισμοί είναι ότι:

- **SC <= topics** : η χωρητικότητα μιας αποθήκης δεν πρέπει να είναι μεγαλύτερη από τα θέματα (όχι αναγκαία συνθήκη αλλά προτεινόμενη)
- **k <= m**: τα αντίγραφα δεν πρέπει να ξεπερνούν τον αριθμό των αποθηκών.

ΚΕΦΑΛΑΙΟ 6 - Αποτελέσματα

Σε αυτό το κεφάλαιο, εκτιμούμε το προτεινόμενο μηχανισμό αποθήκευσης χρησιμοποιώντας μια προσομοίωση διακριτών γεγονότων. Ν κόμβοι(brokers) είναι οργανωμένοι σε μια τοπολογία δικτύου (κοινή τοπολογία σε pub/sub δίκτυα) και οι πελάτες αιτούνται δυναμικά σε κάθε κόμβο i για αποθηκευμένο περιεχόμενο, με ρυθμό r_i^t, διαφορετικό για κάθε θέμα t.
Υποθέτουμε ότι στο δίκτυο μας υπάρχουνΤ θέματα και που μπορούν να αντιγραφούν τουλάχιστον k φορές, Μ αποθήκες που πρέπει να τοποθετηθούν σε αυτό.Κάθε αποθήκη έχει χωρητικότητα SC διαφορετικών θεμάτων.

Κάθε θέμα μπορεί να χαρακτηρισθεί από 2 παραμέτρους,

- τη **δημοτικότητα** και
- την **τοπικότητα**.

Η δημοτικότητα αναφέρεται στο ρυθμό ζήτησης ενός θέματος ενώ η τοπικότητα στην περιοχή της τοπολογίας του δικτύου ,από όπου είναι πιθανόν να προέλθουν οι αιτήσεις

Έχοντας επιλέξει τα σημεία τοποθέτησης των Μ αποθηκών και έχοντας αναθέσει σε αυτά τα Τ θέματα αφήνουμε το σύστημα να λειτουργήσει υπό την ενέργεια των πελατών και των αιτήσεων τους. Οι τιμές μετρήσεων που μας ενδιαφέρουν είναι :

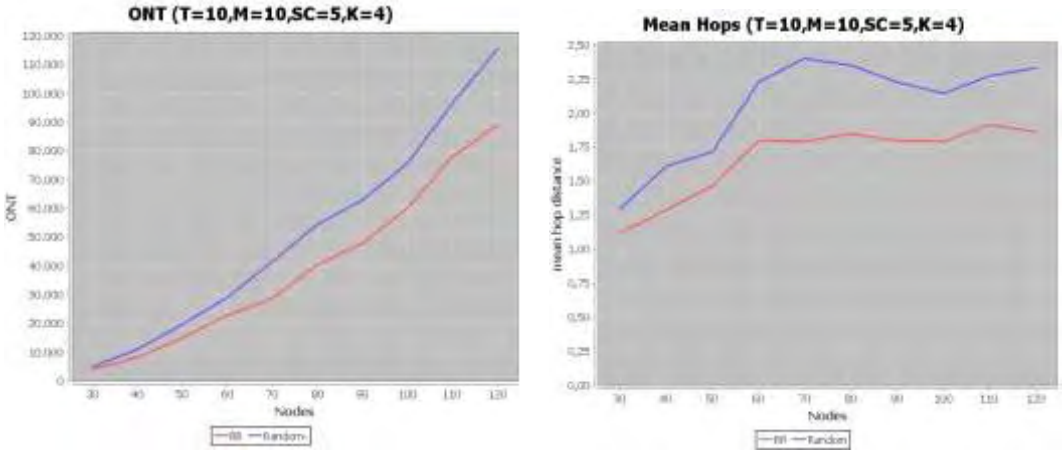
- **overall network traffic, ONT (req*hops/sec)** μετά την ολοκλήρωση του αλγόριθμου Τοποθέτησης/Αντιγραφής.

- **mean hop distance**, η οποία αντιστοιχεί στο μέσο αριθμό αλμάτων (hops) ανάμεσα σε μία αποθήκη που ανταποκρίνεται και τον πελάτη που κάνει την αίτηση.
  Αυτή η τιμή μέτρησης είναι ενδεικτική του response latency σαν μια συνάρτηση αλμάτων(hops) στο δίκτυο.

Υπάρχουν 2 είδη πειραμάτων τα οποία μπορώ να τρέξω και να βγάλω τις γραφικές. Κάθε είδος έχει 2 περιπτώσεις καθώς έχω δύο παραμέτρους που μπορώ να χρησιμοποιήσω σαν μεταβλητές (αριθμός αντιγράφων και αριθμός των κόμβων). Στο πρώτο είδος τρέχω τα πειράματα για **μία τιμή της μιας μεταβλητής** και **πολλές τιμές της άλλης** ενώ στο δεύτερο είδος τα τρέχω **για δύο τιμές μιας μεταβλητές** και **πολλές πάλι τιμές της άλλης** :

1ο Είδος – 1η περίπτωση(ProjectMain)

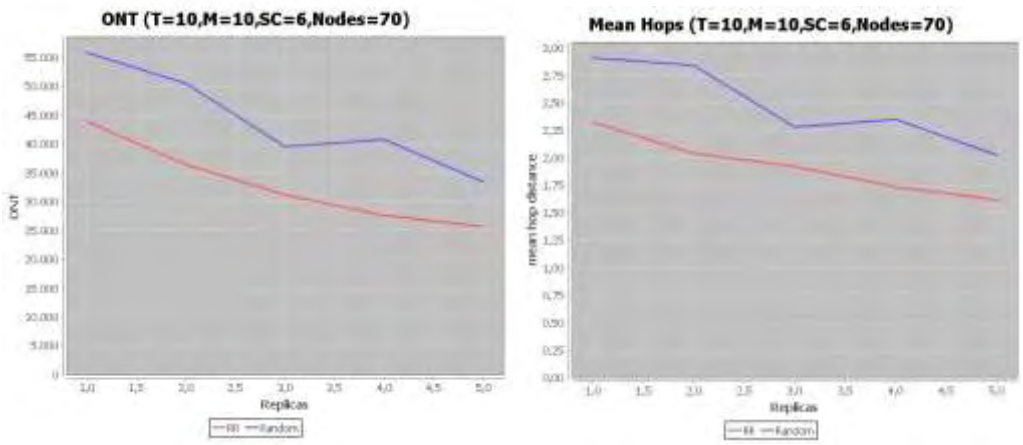
Στο **ProjectMain** τρέχω τον αλγόριθμο για ένα συγκεκριμένο αριθμό αντιγράφων το οποίο το δίνω σαν input (k), και τρέχει για τοπολογίες από 30 έως 120 κόμβους, το οποίο δίνεται στα αρχικά δεδομένα(topNum=10). Επίσης, θεωρώ την χωρητικότητα των αποθηκών σταθερή.(SC=5)
Όπως φαίνεται και στην εικόνα παρακάτω, όσο αυξάνεται ο αριθμός των κόμβων ενός δικτύου, αυξάνονται και οι τιμές του mean hop distance και του ONT καθώς η πληροφορία που αναζητά ο κάθε πελάτης πλέον είναι κατανεμημένη σε μεγαλύτερες αποστάσεις σε σχέση με πριν. Η αύξηση αυτή γίνεται **υπογραμμικά (sublinearly)** σε σχέση με το μέγεθος του δικτύου



Εικόνα 15 : Διαγράμματα απεικόνισης αποτελεσμάτων

1ο Είδος – 2η περίπτωση(ProjectMain2)

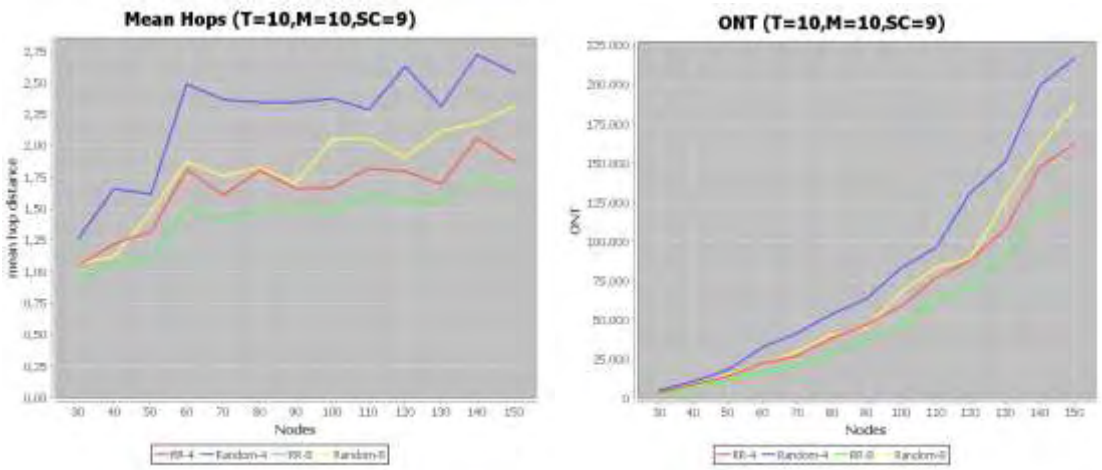
Στο **ProjectMain2** τρέχω τον αλγόριθμο για ένα συγκεκριμένο αριθμό κόμβων το οποίο το δίνω σαν input (maxSize), και τρέχει για αντίγραφα από 1 έως 5, το οποίο δίνεται στα αρχικά δεδομένα(replNum=5). Επίσης, θεωρώ την χωρητικότητα των αποθηκών σταθερή.(SC=6)
Όπως φαίνεται και στην εικόνα παρακάτω, αυξάνοντας τον αριθμό των αντιγράφων κάθε θέματος, μειώνονται οι τιμές του mean hop distance και του ONT καθώς πλέον κάθε θέμα αντιγράφεται σε περισσότερα μέρη του δικτύου και κάθε πελάτης έχει μεγάλη πιθανότητα να βρει την πληροφορία που αναζητά σε κοντινότερη απόσταση .Επίσης, παρατηρούμε ότι οι γραφικές παραστάσεις του ONT και του mean hop distance έχουν την ίδια μορφή . Αυτό συμβαίνει διότι ο βαθμός στον οποίο αντιγράφεται ένα θέμα δεν επηρεάζει το συνολικό ποσό κυκλοφορίας των αιτήσεων που παράγονται στο δίκτυο, και το ONT είναι το γινόμενο της απόστασης (πελάτη-κοντινής αποθήκης) και του αριθμού των αιτήσεων που παράγονται από τους πελάτες.
Βεβαίως, η προσθήκη νέων κόμβων (και νεων πελατών επομένως) αυξάνει το ποσό της κυκλοφορίας στο δίκτυο και τη μέση απόσταση μεταξύ πελατών και αποθηκών και γι’ αυτό η γραφική παράσταση ONT έχει διαφορετική μορφή από αυτή του mean hop distance στα προηγούμενα αποτελέσματα.



Εικόνα 16 : Διαγράμματα απεικόνισης αποτελεσμάτων

2^{ο} Είδος – 1^{η} περίπτωση(ProjectMainMixed)

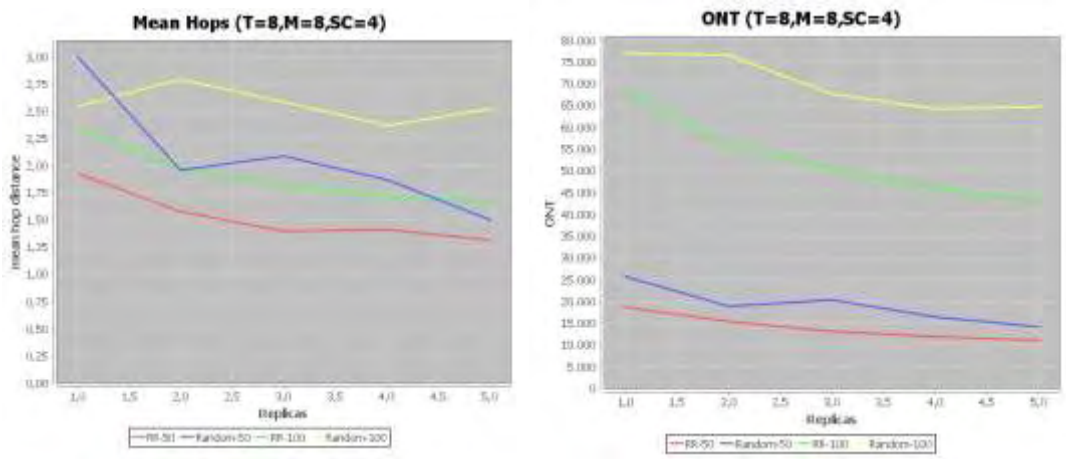
Στο **ProjectMainMixed** τρέχω τον αλγόριθμο για δύο συγκεκριμένους αριθμούς αντιγράφων τους οποίους δίνω σαν input (k[0],k[1]), και τρέχει για τοπολογίες από 30 έως 150 κόμβους, το οποίο δίνεται στα αρχικά δεδομένα(topNum=13). Επίσης, θεωρώ την χωρητικότητα των αποθηκών σταθερή.(SC=9)
Όπως φαίνεται και στην εικόνα παρακάτω, τα αποτελέσματα έχουν την ίδια μορφή που είχαν και στην 1^{η} περίπτωση του 1^{ου} είδους πειραμάτων. Το νέο στοιχείο που παρουσιάζει ενδιαφέρον είναι πως για μεγαλύτερα δίκτυα μπορώ να αυξήσω τον αριθμό των αντιγράφων και να πάρω αποτελέσματα σαφώς καλύτερα (περίπου 15%-20%). Αυτό ωστόσο επιφέρει κάποιο επιπλέον κόστος οπότε ο πάροχος περιεχομένου πρέπει κάθε φορά να λαμβάνει υπόψη τον τελικό σκοπό και να αποφασίζει ανάλογα.



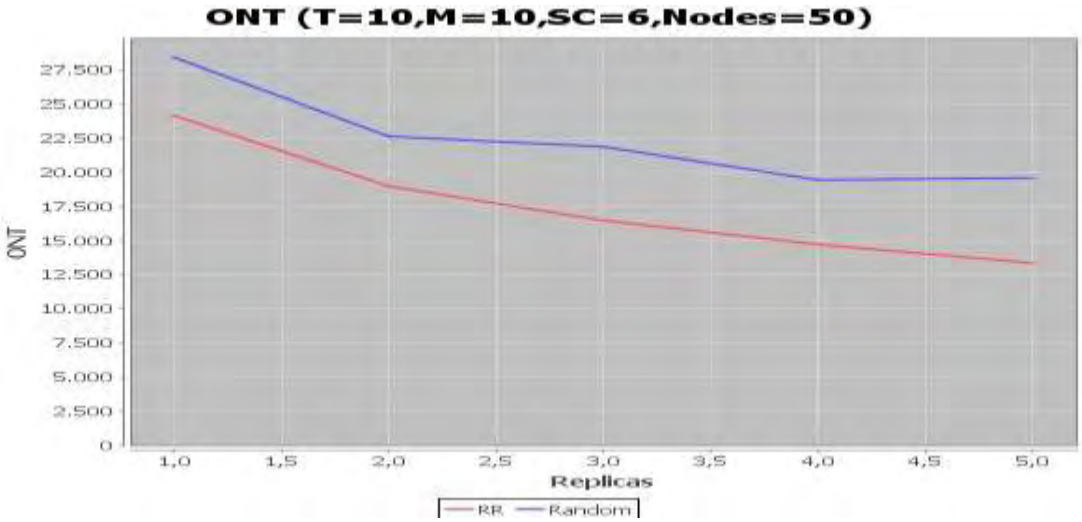
Εικόνα 17 : Διαγράμματα απεικόνισης αποτελεσμάτων

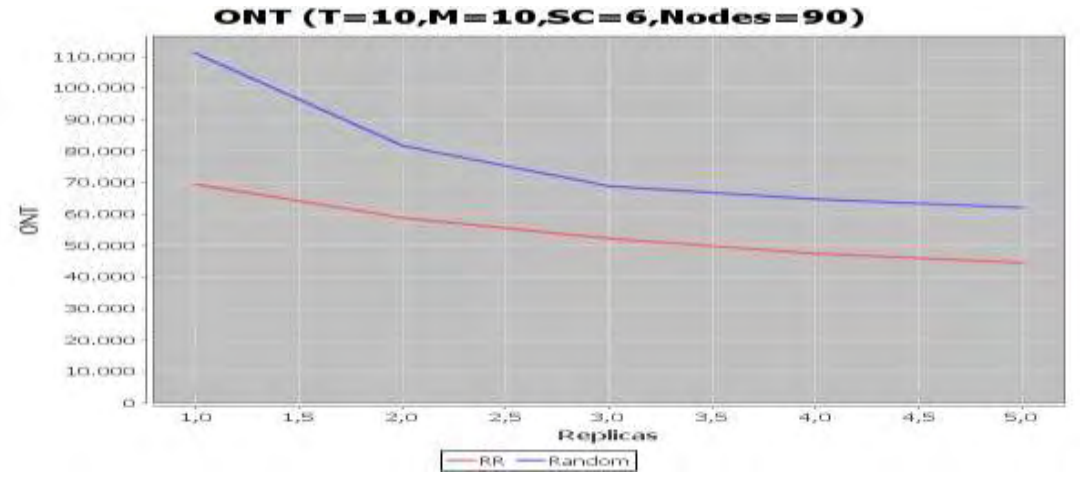
2ο Είδος – 2η περίπτωση(ProjectMainMixed2)

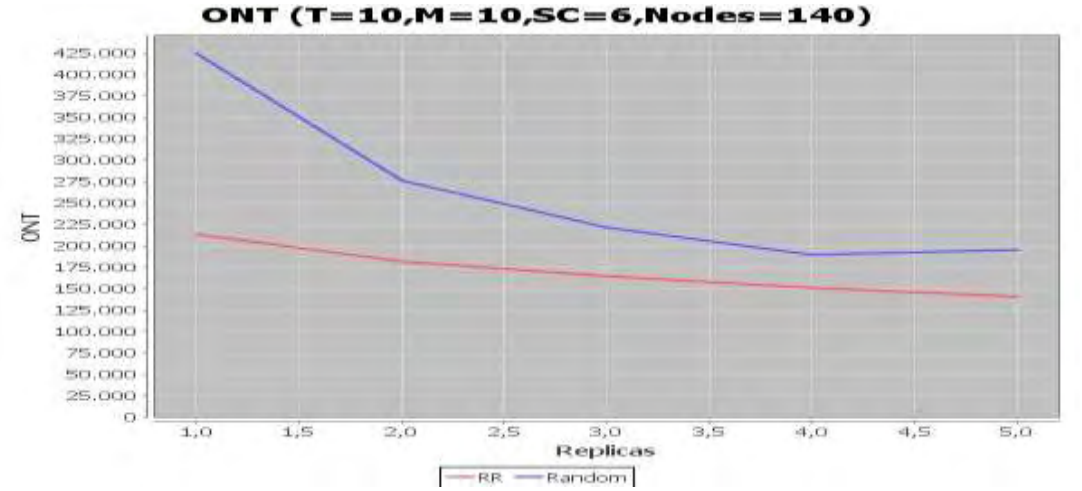
Στο **ProjectMainMixed2** τρέχω τον αλγόριθμο για δύο συγκεκριμένες τοπολογίες τις οποίες δίνω σαν input (maxsize [0], maxsize [1]), και τρέχει για αντίγραφα από 1 έως 5, το οποίο δίνεται στα αρχικά δεδομένα(replNum=5). Επίσης, θεωρώ την χωρητικότητα των αποθηκών σταθερή.(SC=4)
Όπως φαίνεται και στην εικόνα παρακάτω, τα αποτελέσματα έχουν την ίδια μορφή που είχαν και στην 2η περίπτωση του 1ου είδους πειραμάτων. Το νέο στοιχείο που παρουσιάζει ενδιαφέρον είναι οι μεγάλες διαφορές των γραφικων-αποτελεσμάτων ανάμεσα σε δύο διαφορετικά δίκτυα. Η συμπεριφορά αυτή οφείλεται στο γεγονός ότι προστίθονται νέοι κόμβοι , συνεπώς και νέες αιτήσεις, οι οποίες αυξάνουν σημαντικά τις τιμες του ONT και του mean hop distance.



Εικόνα 18 : Διαγράμματα απεικόνισης αποτελεσμάτων

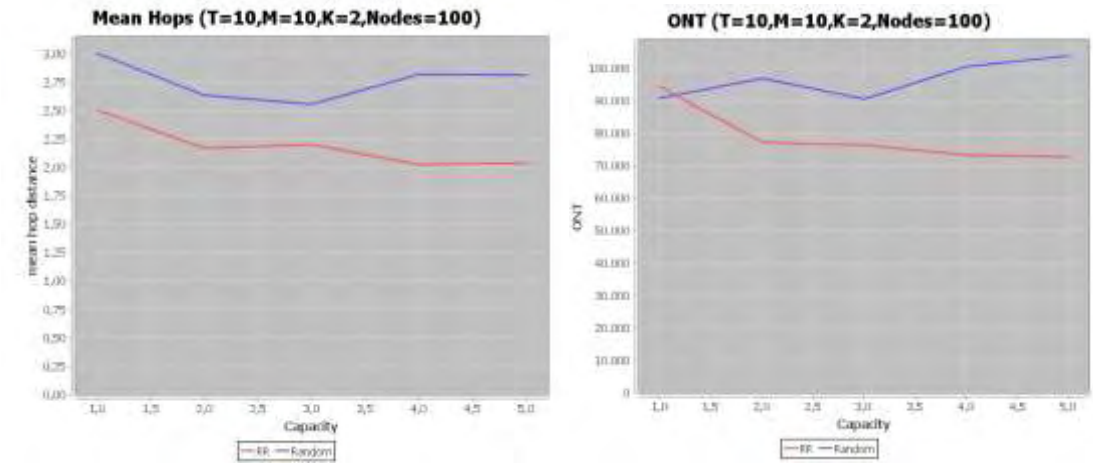






Εικόνα 19 : Διαγράμματα απεικόνισης αποτελεσμάτων όπου φαίνεται η μεγάλη διαφορά στην τιμή του ONT για δίκτυα 40, 90 και 140 κομβων

Επίσης έχω και την **ProjectMainAtomicTest** την οποία χρησιμοποιώ για να δοκιμάζω δίκτυα «πειράζοντας» τις μεταβλητές.
Τέλος, έχω δημιουργήσει και την **ProjectMain3**, μέσω της οποίας τρέχω τον αλγόριθμο για ένα συγκεκριμένο αριθμό κόμβων το οποίο δίνω σαν input (maxSize), και τρέχει για χωρητικότητες αποθηκών από 1 έως 5, το οποίο δίνεται στα αρχικά δεδομένα(SCNum=5). Επίσης, θεωρώ τα αντίγραφα των θεματων σταθερά.(k=2)
Παρατηρούμε πάλι ότι οι γραφικές παραστάσεις του ONT και του mean hop distance έχουν την ίδια μορφή και αυτό συμβαίνει για τους λόγους που προαναφέραμε.



Εικόνα 20 : Διαγράμματα απεικόνισης αποτελεσμάτων

Ενδεικτικά , έχω χρησιμοποιήσει τις παρακάτω τιμές :

ProjectMain → k=4

ProjectMain2 →maxSize = 70

ProjectMain3 →maxSize = 70

ProjectMainMixed → k=4, k=8

ProjectMainMixed → maxSize=50, maxSize =100

Γενικά έχω τοπολογίες για δίκτυα μέχρι 220 κόμβους και requests για δίκτυα μέχρι και 220 κόμβους και 5,8 και 10 θέματα . Οπότε και στα πειράματα που τρέχω φροντίζω έτσι ώστε t=5,t=8 ή t=10 και maxsize μέχρι 220. Εγώ τρέχω μέχρι και 150 κόμβους στα πειράματά μου.

ΚΕΦΑΛΑΙΟ 7 – Συμπεράσματα και Μελλοντικές Ιδέες

7.1. Συμπεράσματα

Στη διπλωματική εργασία αυτή , παρουσιάσαμε ένα νέο αλγόριθμο τοποθέτησης και ανάθεσης αντιγράφων με σκοπό την ελαχιστοποίηση του χρόνου παράδοσης των δεδομένων στους πελάτες του δικτύου καθώς επίσης και τη μείωση στο συνολικό φόρτο στο δίκτυο και ιδιαίτερα στους εξυπηρετητές.
Όπως φαίνεται από τα πειράματα που κάναμε, η ανάθεση των θεμάτων χρησιμοποιώντας τη μέθοδο *Round Robin* επιφέρει καλύτερα αποτελέσματα από την *τυχαία ανάθεση* , περίπου *25%-30%* κατά μέσο όρο. Αυτό συμβαίνει γιατί έχουμε μια πιο δίκαια κατανομή , οπότε ακόμα και λιγότερα δημοφιλή θέματα έχουν τη δυνατότητα να τοποθετηθούν σε αποθήκες που έχουν ήδη επιλέξει ως καλύτερες σε προηγούμενη φάση.
Ωστόσο , παρουσιάζει ενδιαφέρον να μελετήσουμε τη συμπεριφορά του αλγορίθμου μας σε περιπτώσεις όπου χρησιμοποιούνται και άλλοι μέθοδοι ανάθεσης θεμάτων στις αποθήκες , καθώς επίσης και στην περίπτωση που χρησιμοποιούμε άλλους μεθόδους τοποθέτησης. Αυτό το αφήνουμε για μελλοντικές εργασίες .

7.2 Μελλοντικές Ιδέες

Αυτή η εργασία όπως αναφέρθηκε και παραπάνω μπορεί να επεκταθεί με διάφορους τρόπος, όπως:

- Βελτίωση του Μηχανισμού παραγωγής των request.
  Στην εργασία μας χρησιμοποιήσαμε μια συνάρτηση που παράγει τυχαία requests για κάθε κόμβο σε κάθε συγκεκριμένη τοπολογία δικτύου. Μπορεί να δημιουργηθεί ένας μηχανισμός που θα βασίζεται στην παρατήρηση του ρυθμού με την οποία γίνονται τα requests σε μια τοπολογία δικτύου και μέσω ενός προβλεπτικού μηχανισμού να παράγει νέα requests που θα αφορούν τη λειτουργία του ίδιου δικτύου στο βραχυπρόθεσμο μέλλον. Καλύτερα δεδομένα οδηγούν σε καλύτερα αποτελέσματα του αλγορίθμου μας .

- Ενοποίηση του μηχανισμού μας με αυτούς που λειτουργούν σε πραγματικά δίκτυα.

- Παρατήρηση συμπεριφοράς δικτύου χρησιμοποιώντας άλλες μεθόδους τοποθέτησης αποθηκών και ανάθεσης θεμάτων σε αυτές.

- Χρησιμοποίηση ξεχωριστού αριθμού αντιγράφων k_t για κάθε θέμα t, ανάλογα με τη δημοτικότητά τους, και όχι ενός κοινού αριθμού αντιγράφων για όλα τα θέματα.
  Μπορούμε να χρησιμοποιήσουμε τον παρακάτω τύπο για την εύρεση του κάθε k_t :

$$k_t = \left\lceil \frac{w_t}{w_{t'}} \cdot min_k \right\rceil$$

  όπου t’ Ε Τ , είναι το θέμα με το μικρότερο βάρος.

- Ο αριθμός μηνυμάτων κάθε θέματος να μπορεί να είναι διαφορετικός καθώς και τα μεγέθη των μηνυμάτων. Στην εργασία μας θεωρήσαμε ίδιος αριθμό μηνυμάτων και ισομεγέθη μηνύματα για λόγους απλότητας.

Παραπομπές

[1] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley and T. D. Chandra, "Matching events in a content-based subscription system," In Proc. of 18th ACM PODC Atlanta, May, 1999.

[2] A. Carzaniga, D. Rosenblum and A. Wolf, "Design and evaluation of a wide-area event notification service," ACM Transactions On Computer Systems, vol. 19, pp. 332–383, 2001.

[3] B. Segall and D. Arnold, "Elvin has left the building: A publish/subscribe notification service with quenching," In Proc. of AUUG, Brisbane, Australia, Sept. 3-5, pp. 243–255, 1997.

[4] G. Cugola and G. Picco, "REDS, A Reconfigurable Dispatching System," In Proc. of 6th Inter. workshop on Software Engineering and Middleware, pp. 9–16, Oregon, 2006.

[5] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. Briggs, R. Braynard, "Networking named content," In. Proc. of the 5th ACM CoNEXT, Rome, Italy, Dec. 1-4, 2009.

[6] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A Data-Oriented (and Beyond) Network Architecture," In Proc. of SIGCOMM, 2007.

[7] M. M Sarela, T. Rinta-aho, and S. Tarkoma, "RTFM: Publish/Subscribe Internetworking Architecture," In ICT-MobileSummit,2008.

[8] http://www.fp7-pursuit.eu/PursuitWeb/

[9] http://www.4ward-project.eu/index.php?s=file download&id=39.

[10] Li G., Cheung A., Hou S., Hu S., Muthusamy V., Sherafat R., Wun A., Jacobsen H., and Manovski S., "Historic data access in publish/subscribe," In Proc. of DEBS, pp. 80–84, Toronto, Canada, 2007.

[11] Sourlas V., Paschos G. S., Flegkas P. and Tassiulas L., "Caching in content-based publish/subscribe systems," in Proc of IEEE Globecom, Honolulu, USA, Dec. 2009.

[12] V. Sourlas, P. Flegkas, G. S. Paschos, D. Katsaros and L. Tassiulas, "Storing and Replication in Topic-Based Publish/Subscribe Networks," In Proc. of IEEE Globecom, Miami, USA, Dec. 2010.

[13] B. Li, M. J. Golin, G. F. Ialiano and X. Deng, “On the Optimal Placement of Web Proxies in the Internet,” In Proc. of INFOCOM, March 1999.

[14] I. Cidon, S. Kutten, R. Soffer, “Optimal allocation of electronic content,” In Proc. of INFOCOM, Anchorage, April 2001.

[15] J. Kangasharju, J. Roberts, K. Ross, “Object replication strategies in content distribution networks,” Comput. Commun. Elsevier, vol. 25, pp. 376–383, March 2002.

[16] L. Qiu, V.N. Padmanabhan and G. Voelker, “On the placement of web server replicas,” In Proc. of IEEE INFOCOM, Anchorage, USA, Apr. 2001.

[17] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala and V.Pandit, “Local search heuristics for k-median and facility location problems,” In Proc. of 33rd ACM Symp. on Theory of Computing, 2001.

[18] M. Charikar and S. Guha, “Improved combinatorial algorithms for facility location and k-median problems,” In Proc. of the 40th Annual IEEE Symp. on Foundations of Computer Science, pp. 378-388, Oct. 1999.

[19] M. Charikar, S. Khuller, D. Mount, and G. Narasimhan, “Facility location with outliers,” In Proc. of the 12th Annual ACM-SIAM Symp.on Discrete Algorithms, Washington DC, Jan. 2001.

[20] D.B. Shmoys, E. Tardos and K.I. Aardal, “Approximation algorithms for facility location problems,” In Proc. of the 29th Annual ACM Symp.on Theory of Computing, pp. 265-274, 1997.

[21] E. Cronin, S. Jamin, C. Jin, T. Kurc, D. Raz and Y. Shavitt, “Constrained mirror placement on the Internet,” in IEEE JSAC, 36(2), Sept. 2002.

[22] M. Karlsson, Ch. Karamanolis and M. Mahalingam, “A Framework for Evaluating Replica Placement Algorithms”, http://www.hpl.hp.com/ techreports/2002/HPL-2002-21, 2002.

[23] G. Cornuelos, M. L. Fisher, and G. L. Nemhauser. Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms. Management Sciences, Vol. 23, 1977, pp. 789-810.

[24] V.Sourlas, P.Flegkasa, G. S. Paschos,D. Katsaros, L.Tassiulas “Storage Planning and Replica Assignment in Content-Centric Publish/Subscribe Networks”, COMNET 2010

[25] P. Flegkas, V. Sourlas, G. Parisis, D. Trossen, *Storage Replication in Information-Centric Networking*, 2011

[26] http://jung.sourceforge.net/index.html

[27] http://netbeans.org/

[28] P. T. Eugster, P. Felber, R. Guerraoui, and A. M. Kermarrec. The many faces of publish/subscribe.*ACM Computing Surveys*, 2003.

[29] *CORBAservices: Common Object Services Specification*, 1998.

[30] K. P. Birman. The process group approach to reliable distributed computing. *Communications ofthe ACM*, 36(12):36–53, 1993.

[31] D. R. Cheriton and S. E. Deering. Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computing Systems*, 8(2):85, 1990.

[32] M. Castro, P. Druschel, A. Kermarrec, and A. Rowstron. Scribe: A largescale and decentralized applicationlevel multicast infrastructure. *Journal on Selected Areas in Communication*, 2002.

[33] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. D. Kubiatowicz. Bayeux: An architecture for scalable and fault tloerant widearea data dissemination. In *ACM NOSSDAV*, 2001.

[34] L. Fiege and G. Muhl. Rebeca eventbased electronic commerce architecture. In *http://www.gkec.informatik.tudarmstadt.de/rebeca*, 2000

[35] P. R. Pietzuch and J. Bacon. Hermes: a distributed eventbased middleware architecture. In *DEBSb02*, 2002.

[36] A. Rowstron and P. Druschel. Pastry: Scalable and distributed object location and routing for largescale peertopeer systems. In *Proceedings of the International Middleware Conference, Middleware 2001*, 2001.

[37] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: A globalscale overlay for rapid service deployment. *IEEE Journal on Selected Areas in Communications*,2003.

[38] G. Muhl, L. Fiege, F. C. Gartner, and A. P. Buchmann. Evaluating advanced routing algorithms for contentbased publish/subscribe systems. In *Proceedings of the 10th International Workshop on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, MASCOTS*. IEEE Computer Society, 2002.