ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μία Υλοποίηση της Αρχιτεκτονικής LibraRing Χρησιμοποιώντας το Σύστημα FreePastry

Χαράλαμπος Σ. Νικολάου

Επιβλέποντες: Εμμανουήλ Κουμπαράκης, Αναπληρωτής Καθηγητής Ε.Κ.Π.Α

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2007

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μία Υλοποίηση της Αρχιτεκτονικής LibraRing Χρησιμοποιώντας το Σύστημα FreePastry

Χαράλαμπος Σ. Νικολάου

Επιβλέποντες: Εμμανουήλ Κουμπαράκης, Αναπληρωτής Καθηγητής Ε.Κ.Π.Α

ΑΘΗΝΑ

ΟΚΤΩΒΡΙΟΣ 2007

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μία Υλοποίηση της Αρχιτεκτονικής LibraRing Χρησιμοποιώντας το Σύστημα FreePastry

Χαράλαμπος Σ. Νικολάου

Α.Μ.: 1115200300081

ΕΠΙΒΛΕΠΩΝ :

Εμμανουήλ Κουμπαράκης, Αναπληρωτής Καθηγητής Ε.Κ.Π.Α

Περίληψη

Παρουσιάζουμε την υλοποίηση μίας αρχιτεκτονικής για ψηφιακές βιβλιοθήκες βασισμένη στις ιδέες της παραδοσιακής κατανεμημένης Ανάκτησης Πληροφορίας και στις πρόσφατες εργασίες στο πεδίο των Δικτύων Ομότιμων Κόμβων. Η αρχιτεκτονική αυτή, η LibraRing (από τις λέξεις library και ring), είναι ιεραρχική και χρησιμοποιεί την τεχνολογία των Κατανεμημένων Πινάκων Κατακερματισμού για την επίτευξη ευρωστίας, ανοχής σε σφάλματα και κλιμάκωσης σε καθένα από τα επίπεδα δρομολόγησης και διαχείρισης των μεταδεδομένων της. Η αρχιτεκτονική προσφέρει δύο βασικές λειτουργίες: ανάκτηση πληροφορίας και δημοσίευση / συνδρομή. Τα βασικά δομικά στοιχεία αυτής της αρχιτεκτονικής είναι οι υπερ-κόμβοι, οι πελάτες και οι πάροχοι. Οι υπερ-κόμβοι αποτελούν τους κόμβους του δικτύου και παρέχουν τις δύο υπηρεσίες που αναφέραμε. Οι πελάτες και οι πάροχοι αλληλεπιδρούν με το σύστημα LibraRing μέσω των υπερ-κόμβων και μεταβάλλουν την κατάστασή του. Οι πελάτες μπορούν να υποβάλουν επερωτήσεις, χρησιμοποιώντας τη λειτουργία της ανάκτησης πληροφορίας, και να γίνονται συνδρομητές σε διάφορους πόρους που τους ενδιαφέρουν, χρησιμοποιώντας τη λειτουργία της συνδρομής. Οι πάροχοι μπορούν να δημοσιεύουν πόρους στο δίκτυο, επικοινωνώντας με τους υπερ-κόμβους και έτσι χρησιμοποιούν την παρεχόμενη λειτουργία της δημοσίευσης.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Κατανεμημένα Συστήματα

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: ψηφιακές βιβλιοθήκες, δίκτυα ομότιμων κόμβων, κατανεμημένοι πίνακες κατακερματισμού, ανάκτηση πληροφορίας, δημοσίευση / συνδρομή

Abstract

We present the implementation of an architecture for digital libraries based on ideas from traditional distributed Information Retrieval and recent work on peer-to-peer networks. This architecture, called LibraRing (from the words library and ring) is hierarchical and makes use of the technology of Distributed Hash Tables to achieve robustness, fault tolerance and scalability in its routing and meta-data management layer. The architecture provides two fundamental functionalities: informational retrieval and publish / subscribe. The main structural components of this architecture are super-peers, clients and providers. Super-peers are the nodes of the network and provide the two services we just mentioned. Both clients and providers interact with LibraRing through super-peers and alter its state. Clients are able to submit queries using information retrieval functionality and subscribe to various resources of their interest, using publication functionality. Providers are able to publish resources to the network communicating with super-peers and thus they make use of the provided functionality of publication.

SUBJECT AREA: Distributed Systems

KEYWORDS: digital libraries, peer-to-peer networks, distributed hash tables, information retrieval, publish / subscribe

Αφιερωμένο στη μουσική

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα Καθηγητή κ. Εμμανουήλ Κουμπαράκη που μου εμπιστεύτηκε τη διεκπεραίωση αυτής της πτυχιακής εργασίας και δημιούργησε ένα άνετο και φιλικό κλίμα εργασίας, δίνοντας βαρύτητα στην ουσία της γνώσης και όχι στο έργο της πτυχιακής καθ' αυτής. Έπειτα, θα ήθελα να ευχαριστήσω την Ίριδα Μηλιαράκη, η οποία με βοήθησε σε κάθε εμπόδιο, από την αρχή ως το τέλος, όντας πρόθυμη να μου λύσει κάθε απορία. Σημαντική, επίσης, ήταν και η συμβολή του Χρήστου Τρυφωνόπουλου (Max-Planck Institute of Informatics), ο οποίος μου έδωσε την άδεια χρήσης του κώδικα υπολογισμού της ομοιότητας δύο διανυσμάτων και συνετέλεσε στην πλήρη κατανόηση, από μέρους μου, των μικρολεπτομερειών της αρχιτεκτονικής LibraRing.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Πρόλογος

Αυτή η πτυχιακή εργασία εκπονήθηκε στην Αθήνα κατά την περίοδο 2006 - 2007 υπό την επίβλεψη του Αναπληρωτή Καθηγητή του τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστημίου Αθηνών κ. Εμμανουήλ Κουμπαράκη. Διενεργήθηκε σε δύο περιόδους· η πρώτη αφιερώθηκε στην υλοποίηση της αρχιτεκτονικής LibraRing, ενώ η δεύτερη στη σύνταξη του παρόντος πονήματος. Κλείνοντας, θα ήθελα να προσθέσω ότι το υλικό που αφορά στην υλοποίηση αυτής της αρχιτεκτονικής (πηγαίος κώδικας, εκτελέσιμα προγράμματα, αρχεία ρυθμίσεων, αρχεία εισόδου) συμπεριλαμβάνεται στο συνοδευτικό μέσο αποθήκευσης — CD.

Κεφάλαιο 1

Εισαγωγή

Σε αυτή την πτυχιακή εργασία παρουσιάζονται τεχνολογίες σχετικές με τα *δίκτυα ομότιμων κόμβων*, μία περιοχή της επιστήμης της πληροφορικής, η οποία αναπτύχθηκε, κυρίως, λόγω του μη αποδοτικού μοντέλου συστημάτων *πελάτη-εξυπηρετητή* (client-server). Η αδυναμία του μοντέλου αυτού έγινε γρήγορα εμφανής με την παγκόσμια εξάπλωση και διάδοση του *Παγκόσμιου Ιστού* (World Wide Web, WWW). Η αιτία είναι απλή: το μοντέλο πελάτη-εξυπηρετητή καθιστά αδύνατη την ταυτόχρονη παροχή πόρων και υπηρεσιών σε πολλούς χρήστες, ο αριθμός των οποίων ανέρχεται στην τάξη των εκατομμυριών, ίσως και δισεκατομμυρίων ανά τον κόσμο. Τέτοια συστήματα μπορεί να παρουσιάσουν δυσλειτουργίες, όταν τους αιτηθεί μεγάλος αριθμός εξυπηρετήσεων, όπως καθυστέρηση στην εξυπηρέτηση ή ακόμα χειρότερα ανεπανόρθωτες βλάβες από τις οποίες δεν υπάρχει δυνατότητα αυτόματης επανάκαμψης, δηλαδή, χωρίς την επέμβαση του ανθρώπινου παράγοντα. Εκτός αυτού, είναι συχνό φαινόμενο διάσημες υπηρεσίες υλοποιημένες πάνω σε αυτό το μοντέλο συστήματος να γίνονται θύματα επιθέσεων κακόβουλων στοιχείων, όπως hackers.

Αυτά τα προβλήματα, λοιπόν, ενεργοποίησαν και κατεύθυναν τους ερευνητές της επιστήμης της πληροφορικής προς ένα άλλο μοντέλο συστήματος, το οποίο έχει ανοχή σε τέτοιες δυσλειτουργίες, με ενσωματωμένους τρόπους επανάκαμψης και παρουσιάζει λογικούς ρυθμούς απόκρισης σε αιτήσεις πελατών. Αυτό το μοντέλο είναι γνωστό και ως *Δίκτυο Ομότιμων Κόμβων* (peer-to-peer network, p2p network). Σε ένα τέτοιο μοντέλο δεν υπάρχει η διάκριση μεταξύ πελάτη και εξυπηρετητή. Αντίθετα, κάθε κόμβος είναι στο ίδιο ιεραρχικό επίπεδο, δρα δηλαδή ταυτόχρονα και ως πελάτης και ως εξυπηρετητής. Σε ένα τέτοιο μοντέλο, οι πόροι διατείθενται και αποθηκεύονται σε πολλούς κόμβους από ότι σε έναν. Με αυτόν τον τρόπο η πρόσβαση σε ένα συγκεκριμένο πόρο είναι ευκολότερη, ταχύτερη, και αποδοτικότε-

ρη, διότι τώρα η πιθανότητα όλοι οι κόμβοι που παρέχουν έναν συγκεκριμένο πόρο να είναι εκτός δικτύου είναι ελάχιστη. Βέβαια, την αυξημένη απόδοση εξυπηρέτησης αιτήσεων έρχεται να αντισταθμίσει η πολυπλοκότητα αλγορίθμων διαχείρισης ενός τέτοιου συστήματος καθώς και επιπρόσθετα προβλήματα ασφάλειας. Μία τεχνολογία που είναι ευρέως διαδεδομένη και έρχεται να αντιμετωπίσει αυτό το πρόβλημα της πολυπλοκότητας εντοπισμού και διαμοιρασμού των πόρων μεταξύ των κόμβων του συστήματος είναι οι *Κατανεμημένοι Πίνακες Κατακερματισμού* (Distributed Hash Tables, DHTs), οι οποίοι αποτελούν και το κύριο κομμάτι ενασχόλησης και αφορμή αυτής της πτυχιακής εργασίας. Γενικά, αυτοί οι πίνακες είναι χρήσιμοι για την αποθήκευση πόρων και την εύκολη και γρήγορη ανάκτησή τους. Παράλληλα, συνεισφέρουν σε σημαντικό βαθμό σε αυτό που λέγεται *ανοχή σε σφάλματα* (fault tolerance). Για παράδειγμα σε μία περίπτωση, όπου ένας κόμβος τίθεται εκτός δικτύου, με τη χρήση του κατανεμημένου πίνακα κατακερματισμού οι πόροι αυτού του κόμβου μετατείθονται στην δικαιοδοσία άλλων κόμβων του δικτύου σύμφωνα, βέβαια, με κάποιους αλγόριθμους δρομολόγησης.

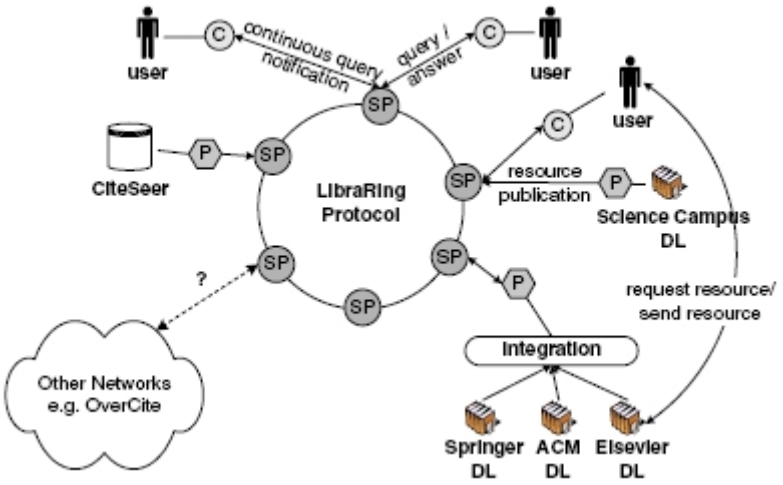
Στη συνέχεια, όπως έχει γίνει ήδη αντιληπτό, θα μελετηθούν σε βάθος αυτές οι τεχνολογίες που παρέχονται δια μέσου των δικτύων ομότιμων κόμβων.

1.1 Αντικείμενο της Πτυχιακής Εργασίας

Αυτή η πτυχιακή ασχολείται με την αρχιτεκτονική LibraRing που αρχικά παρουσιάστηκε στο [8], μια αρχιτεκτονική για *κατανεμημένες ψηφιακές βιβλιοθήκες* (distributed digital libraries) βασισμένη στους κατανεμημένους πίνακες κατακερματισμού. Η αρχιτεκτονική LibraRing προσφέρει δύο βασικές λειτουργίες: *ανάκτηση πληροφορίας* (information retrieval, IR) καθώς επίσης και *δημοσίευση / συνδρομή* (publish / subscribe, pub / sub). Σε ένα σενάριο που αφορά στην πρώτη λειτουργία ένας χρήστης έχει τη δυνατότητα να υποβάλει μία επερώτηση (π.χ. "Θα ήθελα δημοσιεύσεις από το χώρο της βιοπληροφορικής") και το σύστημα να του επιστρέψει πληροφορίες για τους πόρους που ταιριάζουν με αυτή. Σε ένα σενάριο που αφορά στην δεύτερη λειτουργία ένας χρήστης μπορεί να αποστείλει στο σύστημα μία συνδρομή, η οποία περιγράφει τα ενδιαφέροντά του (π.χ. "Ενδιαφέρομαι για δημοσιεύσεις βιοπληροφορικής"), έτσι ώστε να λάβει ειδοποιήσεις σχετικά με συγκεκριμένα γεγονότα που τον ενδιαφέρουν (π.χ. όταν μία δημοσίευση που αφορά στην βιοπληροφορική γίνει διαθέσιμη στο σύστημα). Παρουσιάζουμε, επίσης, τα βασικά δομικά στοιχεία της αρχιτεκτονικής LibraRing. Αυτά είναι τρία: *υπερ-κόμβοι* (super-peers), *πελάτες* (clients) και *πάροχοι* (providers).

Η πρώτη κατηγορία, υπερ-κόμβοι, αποτελούν τους κόμβους του δικτύου και παρέχουν τις δύο υπηρεσίες που προλογήθηκαν. Οι πελάτες και οι πάροχοι αποτελούν τους εξωτερικούς παράγοντες του συστήματός LibraRing, οι οποίοι αλληλεπιδρούν με αυτό, μέσω των υπερ-κόμβων, και μεταβάλλουν την κατάστασή του. Οι πελάτες μπορούν να υποβάλουν *επερωτήσεις* (queries), χρησιμοποιώντας τη λειτουργία της ανάκτησης πληροφορίας, και να γίνονται συνδρομητές σε διάφορους πόρους που τους ενδιαφέρουν, χρησιμοποιώντας τη λειτουργία της συνδρομής. Οι πάροχοι μπορούν να δημοσιοποιούν πόρους στο δίκτυο, επικοινωνώντας με τους υπερ-κόμβους και έτσι χρησιμοποιούν την παρεχόμενη λειτουργία της δημοσίευσης.

Η λειτουργικότητα που προσφέρει η αρχιτεκτονική LibraRing παρουσιάζεται στο Σχήμα 1.1.



Σχήμα 1.1: Η αρχιτεκτονική LibraRing.

1.2 Στόχοι της Πτυχιακής Εργασίας

Ο στόχος αυτής της πτυχιακής εργασίας είναι η υλοποίηση της αρχιτεκτονικής LibraRing, δηλαδή, η υλοποίηση των λειτουργιών της ανάκτησης πληροφορίας και της δημοσίευσης / συνδρομής καθώς επίσης και των πρωτοκόλλων επικοινωνίας των δομικών στοιχείων της (υπερ-κόμβοι, πελάτες, πάροχοι). Η αρχιτεκτονική αυτή υλοποιήθηκε χρησιμοποιώντας το σύστημα FreePastry, το οποίο αποτελεί μία υλοποίηση του συστήματος Pastry στη γλώσσα προγραμματισμού Java. Το σύστημα Pastry είναι αυτό που προσφέρει και υλοποιεί τη διεπαφή του Κατανεμημένου Πίνακα Κατακερματισμού. Εκτός από θέματα υλοποίησης, η πτυχιακή εργασία στοχεύει στην παρουσίαση και κριτική των τεχνολογιών των δικτύων ο-

μότιμων κόμβων, μία σταθερά ανερχόμενη περιοχή που έχει προσελκύσει τα βλέμματα των ερευνητών, αλλά και των επιχειρήσεων.

1.3 Διάρθρωση της Πτυχιακής Εργασίας

Η παρούσα πτυχιακή εργασία αποτελείται από πέντε κεφάλαια.

- Στο Κεφάλαιο 2 παρουσιάζεται η βιβλιογραφία που είναι σχετική με το αντικείμενο και τους στόχους αυτής της εργασίας — άλλοτε κατέχοντας υποστηρικτό ρόλο απέναντι στο θεωρητικό υπόβαθρο της εργασίας και άλλοτε ως εργαλεία που χρησιμοποιήθηκαν στην υλοποίηση της.
- Στο Κεφάλαιο 3 παρουσιάζεται η αρχιτεκτονική LibraRing.
- Στο Κεφάλαιο 4 περιγράφεται η υλοποίηση της αρχιτεκτονικής LibraRing· παρουσιάζονται θέματα υλοποίησης και τεχνικές.
- Στο Κεφάλαιο 5 γίνεται μια σύνοψη των όσων παρουσιάσθηκαν και αναφέρονται οι δυνατότητες μελλοντικής επέκτασης αυτής της εργασίας.

Τέλος, στο Παράρτημα, ακολουθεί η ορολογία που χρησιμοποιήθηκε, ο πίνακας συντμήσεων-αρκτικόλεξων και η σχετική βιβλιογραφία.

Κεφάλαιο 2

Σχετική Βιβλιογραφία

Σε αυτό το κεφάλαιο θα αναφερθούμε στις τεχνολογίες που κατέστησαν δυνατή αυτή την εργασία. Συγκεκριμένα, θα γίνει λόγος για τα *Δίκτυα Ομότιμων Κόμβων* (Peer-to-Peer Networks) και για την υποπεριοχή των *Κατανεμημένων Πινάκων Κατακερματισμού* (Distributed Hash Tables), που στο εξής θα αναφέρονται ως ΚΠΚ. Επίσης θα παρουσιάσουμε μία εισαγωγή στις περιοχές της *Ανάκτησης Πληροφορίας* (Information Retrieval, IR) και της *Δημοσίευσης / Συνδρομής* (Publish / Subscribe, Pub / Sub). Παράλληλα, θα παρουσιαστούν συστήματα που υλοποιούν την τεχνολογία των ΚΠΚ, καθώς επίσης και ένα μοντέλο αναπαράστασης πληροφορίας, το *AWPS* (Attribute Word-Pattern with Similarity), και μία γλώσσα αναπαράστασης δεδομένων, η *XML* (EXtensible Mark-up Language).

Πιο συγκεκριμένα, στην Ενότητα 2.1 γίνεται μία εισαγωγή στη τεχνολογία των Δικτύων Ομότιμων Κόμβων· δίνεται ένας περιγραφικός ορισμός, παρουσιάζονται οι αρχές που πρέπει να διέπουν τέτοια δίκτυα και αποπειράται μία γενική αποτίμηση τους.

Στην Ενότητα 2.2 εισάγεται η τεχνολογία των κατανεμημένων πινάκων κατακερματισμού· παρουσιάζονται τα προβλήματα που προυπήρχαν και τα οποία προτίθενται να λύσουν, ενώ παράλληλα περιγράφονται τα συστήματα Chord (εν συντομία) και Pastry (σε μεγαλύτερη έκταση) που υλοποιούν αυτή την τεχνολογία.

Στην Ενότητα 2.3 εισάγεται η έννοια της Ανάκτησης Πληροφορίας, μία έννοια που έχει προσελκύσει το ενδιαφέρον πολλών ερευνητών και επιχειρήσεων τα τελευταία χρόνια· αντιπαραβάλλεται με την έννοια της Ανάκτησης Δεδομένων και παρουσιάζονται οι βασικότερες εισαγωγικές της έννοιες.

Στην Ενότητα 2.4 παρουσιάζεται η έννοια της Δημοσίευσης / Συνδρομής, μία έννοια που ταιριάζει πλήρως με το μοντέλο και τις προδιαγραφές των δικτύων ομότιμων κόμβων.

Εισάγονται οι έννοιες που είναι απαραίτητες για την κατανόηση της λειτουργικότητας που προσφερει η αρχιτεκτονική LibraRing.

Στην Ενότητα 2.5 παρουσιάζεται ένα μοντέλο αναπαράστασης πληροφορίας, το AWPS. Το μοντέλο αυτό χρησιμοποιείται στην αρχιτεκτονική LibraRing για την αναπαράσταση των πόρων που διαχειρίζεται (επερωτήσεις, δημοσιεύσεις και ειδοποιήσεις).

Στην Ενότητα 2.6 παρουσιάζεται μία γλώσσα αναπαράστασης δεδομένων, η XML. Αυτή χρησιμοποιείται από το σύστημα LibraRing για την υλοποίηση του μοντέλου αναπαράστασης πληροφορίας της, το AWPS. Εισάγονται οι βασικές έννοιες που είναι απαραίτητες για την κατανόηση της χρήσης της XML σε θέματα υλοποίησης του συστήματος LibraRing και παρουσιάζεται η αναπαράσταση των δεδομένων του.

2.1 Δίκτυα Ομότιμων Κόμβων (P2P Networks)

Τα δίκτυα ομότιμων κόμβων δεν διαφέρουν τοπολογικά από τα δίκτυα που χρησιμοποιούνται ευρέως στην καθημερινότητα, όπως τα τοπικά δίκτυα ή το δίκτυο του Παγκόσμιου Ιστού. Διαφέρουν, όμως, ριζικά, ως προς τις σχέσεις και την ιεραρχία των κόμβων που αποτελούν αυτό το δίκτυο. Οι *κόμβοι* (peers) ενός τέτοιου δικτύου βρίσκονται συνήθως στο ίδιο επίπεδο ιεραρχίας, όσον αφορά στις αρμοδιότητες τους και είναι αυτόνομοι. Αυτό σημαίνει, ότι η λειτουργία ενός κόμβου δεν επηρεάζεται από την κατάσταση στην οποία βρίσκεται ένας άλλος κόμβος, όπως αντίθετα συμβαίνει στα δίκτυα της μορφής πελάτη-εξυπηρετητή. Αυτό το χαρακτηριστικό είναι που τα κάνει σήμερα τα δημοφιλέστερα δίκτυα παροχής υπηρεσιών διαμοιρασμού δεδομένων (π.χ. αρχείων κειμένου, τραγουδιών, ταινιών, εκτελέσιμων προγραμμάτων). Γνωστά συστήματα[1], όπως το Napster, το Gnutella, το FreeNet, το KaZaA, το Morpheus και πιο πρόσφατα τα δίκτυα Bit Torrent προσφέρουν τέτοιες υπηρεσίες διαμοιρασμού δεδομένων κάνοντας αυτό το μοντέλο δικτύου δημοφιλές. Παρ' όλα αυτά, τέτοια δίκτυα δεν προορίζονται μονάχα για υπηρεσίες διαμοιρασμού δεδομένων, αλλά και για άλλες

[1]Στη συνέχεια, όπου υπάρχει αναφορά σε ομότιμους κόμβους με τη χρήση του όρου συστήματος θα είναι όμοιας σημασίας με εκείνου του δικτύου. Ένα δίκτυο ομότιμων κόμβων αποτελεί ένα σύστημα ομότιμων κόμβων με την δυνατότητα επικοινωνίας και ανταλλαγής δεδομένων. Όμοια, ένα σύστημα ομότιμων κόμβων δεν νοείται χωρίς την δυνατότητα επικοινωνίας και γι' αυτό είναι επίσης και ένα δίκτυο. Για την αποφυγή σύγχυσης θα πρέπει να αναφέρουμε ότι το δίκτυο αναφέρεται καθαρά στην έννοια της επικοινωνίας δια μέσου ενός φυσικού μέσου (π.χ. καλώδιο ή αέρας στην περίπτωση της ασύρματης επικοινωνίας), ενώ το σύστημα αναφέρεται στις παρεχόμενες υπηρεσίες, στην εφαρμογή ως ολότητα και εμπεριέχει την έννοια του δικτύου.

κατανεμημένες εφαρμογές, όπως *υπολογισμοί σε Πλέγμα* (Grid Computation) (π.χ. οι εφαρμογές SETIHome και DataSynapse), τα *συνεργαζόμενα δίκτυα* (collaboration networks), (π.χ. Groove) ακόμη και για νέους τρόπους σχεδιασμού της υποδομής του διαδικτύου ώστε να υποστηρίζει εξελιγμένα πρότυπα επικοινωνίας [35].

Αν θέλαμε να αποδώσουμε έναν ορισμό στην έννοια των δικτύων ομότιμων κόμβων, σίγουρα θα καταφεύγαμε στον χονδρικό, αλλά περιεκτικό και διορατικό, ορισμό του Clay Shirkey (The Accelerator Group) [2]:

> Οι ομότιμοι κόμβοι είναι μία κλάση εφαρμογών που εκμεταλλεύονται στο έπακρο την αποθήκευση πόρων, τους κύκλους, το περιεχόμενο και την ανθρώπινη παρουσία στις ακμές του Παγκόσμιου Ιστού. Επειδή η πρόσβαση σε αυτούς τους κατανεμημένους πόρους σημαίνει την λειτουργία σε ένα περιβάλλον ασταθούς συνδεσιμότητας και αβέβαιων διευθύνσεων *IP* (Internet Protocol), οι ομότιμοι κόμβοι πρέπει να λειτουργούν έξω από τα πλαίσια ενός *Συστήματος Πεδίου Ονομάτων* (Domain Name System, DNS) και να έχουν σημαντικό ή ολοκληρωτικό βαθμό αυτονομίας, όμοια αυτής των κεντρικών εξυπηρετητών.

Σύμφωνα με αυτόν τον ορισμό, θα λέγαμε ότι οι ομότιμοι κόμβοι αποτελούν ένα επίπεδο εφαρμογών που λειτουργούν πάνω στο επίπεδο του Διαδικτύου, ή αλλιώς, ένα *δίκτυο επικάλυψης* (overlay network) κόμβων. Ένα δίκτυο επικάλυψης είναι ένα εικονικό δίκτυο από κόμβους και λογικές συνδέσεις, το οποίο είναι κτισμένο στην κορυφή ενός υπάρχοντος δικτύου με στόχο την υλοποίηση μίας δικτυακής υπηρεσίας η οποία δεν είναι διαθέσιμη στο υπάρχον δίκτυο. Επιπρόσθετα, τέτοιοι κόμβοι θα πρέπει να ικανοποιούν τις παρακάτω αρχές [16]:

1. *Η αρχή του διαμοιρασμού πόρων*: όλα τα συστήματα ομότιμων κόμβων περικλείουν μία άποψη διαμοιρασμού πόρων, όπου αυτοί μπορεί να είναι φυσικοί πόροι, όπως αποθηκευτικός χώρος σε κάποιον σκληρό δίσκο, καθώς επίσης και λογικοί πόροι, όπως παροχή υπηρεσιών ή διάφορες άλλες μορφές γνώσης και πληροφορίας. Με αυτή την έννοια του διαμοιρασμού πόρων γίνεται εύκολα κατανοητό ότι η χρήση της τεχνολογίας πελάτη-εξυπηρετητή είναι απαγορευτική για αυτόν τον σκοπό. Αυτό υπήρξε το ερέθισμα προς την επινόηση και κατασκευή των συστημάτων και δικτύων ομότιμων κόμβων όπως το Napster.

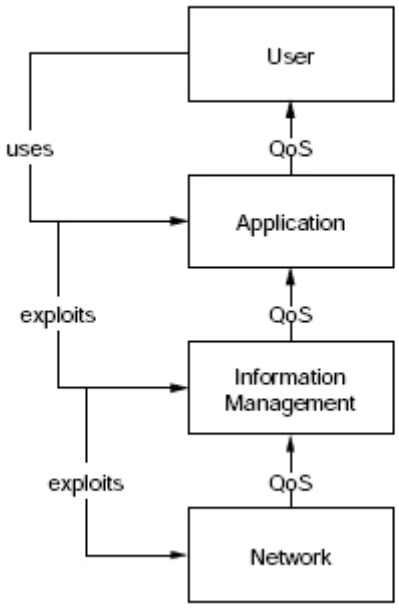
2. *Η αρχή της αποκέντρωσης*: αυτή η αρχή είναι μία άμεση συνέπεια του διαμοιρασμού

πόρων. Μέρη του συστήματος ή ακόμα και ολόκληρο το σύστημα δεν λειτουργούν κεντρικά. Η αποκέντρωση είναι ενδιαφέρουσα και ζωτικής σημασίας για την αποφυγή μονόπλευρων αποτυχιών ή άλλων ανασχετικών παραγόντων της απόδοσης του συστήματος. Απλούστερα, η αποκέντρωση έχει ως στόχο την κατανομή ίσων αρμοδιοτήτων και πόρων σε πολλούς κόμβους, οι οποίοι αποτελούν το σύστημα. Με αυτόν τον τρόπο ενισχύεται η απόδοση του συστήματος και η αδιάκοπη λειτουργία του. Ζωντανά παραδείγματα τέτοιων αποκεντρωμένων συστημάτων είναι το Gnutella και το FreeNet.

3. *Η αρχή της αυτο-οργάνωσης*: όταν ένα σύστημα ομότιμων κόμβων γίνεται ολοκληρωτικά αποκεντρωμένο παύει να υπάρχει κάποιος κόμβος που να μπορεί να οργανώνει και να ρυθμίζει τις δραστηριότητές του ή μία βάση δεδομένων κεντρικά, χρησιμοποιώντας καθολικές πληροφορίες του συστήματος. Ως εκ τούτου οι κόμβοι πρέπει να αυτο-οργανώνονται βασιζόμενοι σε οποιαδήποτε πληροφορία είναι διαθέσιμη τοπικά και να αλληλεπιδρούν με τοπικά γειτονικούς κόμβους (*γείτονες*, neighbours). Έτσι, η καθολική συμπεριφορά του συστήματος αναδύεται από τη συνισταμένη των επιμέρους τοπικών συμπεριφορών των κόμβων.

2.1.1 Τύποι Δικτύων Ομότιμων Κόμβων

Το μοντέλο των Ομότιμων Κόμβων υπήρχε πριν από την ανάπτυξη συστημάτων διαμοιρασμού αρχείων όπως το Napster, το Gnutella ή το FreeNet, τα οποία αύξησαν την δημοτικότητα του. Στην πραγματικότητα, πολύ γνωστά συστήματα βασίζονται σε παρόμοιες θεμέλιες βάσεις. Σε επίπεδο Διαδικτύου, το Arpanet ήταν ένα από τα πρώτα συστήματα ομότιμων κόμβων: Κάθε *ξένιος κόμβος* (host node) που επιτελούσε χρέη φιλοξενίας του πρωτοκόλλου του Διαδικτύου είχε τη δυνατότητα να χρησιμοποιήσει telnet και υπηρεσίες ftp προς άλλους τέτοιους, όμοιους, κόμβους (έτσι επιτυγχάνεται το ένα και μοναδικό επίπεδο ιεραρχίας που προαναφέρθηκε). Πιο πρόσφατα, κινητά ad-hoc δίκτυα αρχίζουν να ακολουθούν την προσέγγιση των ομότιμων κόμβων. Σε κατανεμημένα συστήματα βάσεων δεδομένων, όπως το σύστημα Mariposa [22], επιδεικνύονται ιδιότητες των ομότιμων κόμβων, όπως επίσης και συστήματα *ηλεκτρονικού εμπορίου* (e-commerce), όπως για παράδειγμα, το eBay, τα B2B (Business-to-Business) μέρη αγορών ή οι εξυπηρετητές με ενσωματωμένο το σύστημα B2B [16]. Ακολουθεί λογικά, λοιπόν, το συμπέρασμα ότι η προσέγγιση συστημάτων με ομότιμους κόμβους μπορεί να εφαρμοστεί σε πολλά και διάφορα επίπεδα, όπως διαφαίνεται και στο Σχήμα 2.1.



Σχήμα 2.1: Επίπεδα συστημάτων που θα μπορούσαν να εκμεταλλευτούν την προσέγγιση των Δικτύων Ομότιμων Κόμβων.

Όπως αναφέρθηκε προηγουμένως, το *επίπεδο δικτύου* (network layer), δηλαδή, το Διαδίκτυο, αποτελείται από τους Ομότιμους Κόμβους. Στο επίπεδο της *διαχείρισης της πληροφορίας* (information management) υπηρεσίες καταλόγου και βάσεων δεδομένων θα μπορούσαν να είναι είτε κεντρικοποιημένες, είτε κατανεμημένες (δηλαδή, να ακολουθούν την προσέγγιση των ομότιμων κόμβων). Τα συστήματα ηλεκτρονικού εμπορίου, για παράδειγμα, τα οποία θα βρίσκονταν στο *επίπεδο εφαρμογών* (application layer) θα μπορούσαν και αυτά με τη σειρά τους να είναι είτε κεντρικοποιημένα, είτε κατανεμημένα. Το *επίπεδο του χρήστη* (user layer), δηλαδή, εμπόριο, κοινωνία κ.λπ., ακολουθεί την προσέγγιση των ομότιμων κόμβων, όπως πολύ εύκολα μπορεί να διαπιστώσει κανείς. Ο Πίνακας 2.1 υποδεικνύει την παραπάνω ανάλυση σε επίπεδα για τέσσερα καλά ιδρυμένα συστήματα.

2.1.2 Ιεραρχικά Δίκτυα Ομότιμων Κόμβων

Μία προσέγγιση που μπορεί να ακολουθηθεί για να χειριστούμε την αδυναμία *κλιμάκωσης*[2] (scalability), όπως το Gnutella, είναι να εισάγουμε την έννοια της ιεράρχησης. Σύμφωνα

[2]Με τον όρο κλιμάκωση εννοείται η ικανότητα και ευελιξία μίας εφαρμογής (στην περίπτωσή μας ολόκληρου συστήματος και των επιμέρους κόμβων που το αποτελούν) στην προσαρμογή της λειτουργικότητάς του σε

	P2P user interaction	**P2P application**	**P2P information management**
eBay	yes	no	no
Napster	yes	yes	no
Napster	yes	yes	yes
Gnutella	yes	yes	yes
FreeNet	yes	yes	yes

Πίνακας 2.1: Πόσο πολύ εμπλέκονται τα Δικτύα Ομότιμων Κόμβων;

με αυτή, οι ομότιμοι κόμβοι χωρίζονται σε δύο κατηγορίες, στους *υπερ-κόμβους* (super-peers) και στους πελάτες (clients). Αυτή η προσέγγιση δεν αντιβαίνει με τον ορισμό και τις ιδιότητες που χαρακτηρίζουν ένα σύστημα ή δίκτυο ομότιμων κόμβων, που παρουσιάστηκαν πρωτύτερα. Τα ιεραρχικά δίκτυα ομότιμων κόμβων είναι μία υποπερίπτωση των ομότιμων δικτύων υπό την έννοια ότι αποτελούνται από δύο σύνολα ομότιμων κόμβων όπου οι κόμβοι του κάθε συνόλου είναι ομότιμοι σε λειτουργίες και αρμοδιότητες. Αυτή η προσέγγιση ακολουθήθηκε από το σύστημα Gnutella καθώς και άλλα νέα συστήματα, όπως το KaZaA και το Morpheus. Επιπρόσθετα, η αρχιτεκτονική του συστήματος LibraRing, η υλοποίηση της οποίας έγινε η αφορμή για αυτή την πτυχιακή εργασία, ακολουθεί αυτήν την ιεραρχική προσέγγιση. Η φιλοσοφία ενός τέτοιου ιεραρχικού δικτύου είναι η εξής: κάθε υπερ-κόμβος εξυπηρετεί ένα μέρος των πελατών αποθηκεύοντας, επίσης, μεταδεδομένα για τον καθένα από αυτούς. Οι υπερ-κόμβοι ακολουθούν ένα πρωτόκολλο της επιλογής τους (π.χ. ένα συμμετρικό πρωτόκολλο, όπως αυτό του Gnutella, ένα δομημένο, όπως αυτό του Napster ή ένα πρωτόκολλο κατανεμημένων πινάκων κατακερματισμού, όπως αυτό του Pastry ή του LibraRing). Οι πελάτες μπορεί να εκτελούνται σε υπολογιστές χρηστών και οι πόροι να αποθηκεύονται στους κόμβους αυτών των πελατών. Οι πελάτες είναι και αυτοί ισότιμοι, δεδομένου ότι χρησιμοποιούν το λογισμικό (ίδιο είτε διαφορετικό) που παρέχει την ίδια λειτουργικότητα. Οι πελάτες μαθαίνουν για πόρους θέτοντας επερωτήσεις σε υπερ-κόμβους, ενώ ανακτούν τους πόρους αυτούς άμεσα από άλλους κόμβους πελάτες. Ένα σημαντικό πλεονέκτημα των ομότιμων δικτύων είναι ότι ένας συγκεκριμένος πόρος μπορεί να είναι αποθηκευμένος σε περισσότερους από έναν υπερ-κόμβους και έτσι ο πελάτης να συνδεθεί σε περισσότερους από έναν και να

αυξομειούμενο αριθμό χρηστών και χειρισμού δεδομένων. Ουσιαστικά, ένα σύστημα ικανοποιεί αυτή την ιδιότητα αν και μόνο αν παρουσιάζει καλή ή παρόμοια συμπεριφορά σε πιθανών μεγάλες αλλαγές στον αριθμό των χρηστών του και των δεδομένων που χειρίζεται.

λαμβάνει διαφορετικά κομμάτια του πόρου αυτού από αυτούς. Αυτή η δυνατότητα παρέχεται, κατά βάση, από συστήματα διαμοιρασμού δεδομένων, όπως το KaZaA και τα Bit Torrent. Τα πλεονεκτήματα αυτής της δυνατότητας αφορούν τόσο την πλευρά των υπερ-κόμβων, όπως επίσης και αυτή των πελατών. Από την μεριά των πρώτων, ένα πλεονέκτημα είναι η εξυπηρέτηση περισσότερων πελατών, ενώ από την πλευρά των δεύτερων είναι η εξοικονόμηση χρόνου [35].

Τα θέματα που περιλαμβάνονται στον σχεδιασμό και την υλοποίηση ιεραρχικών δικτύων ομότιμων κόμβων έχουν προσφάτως μελετηθεί από διάφορους ερευνητές. Αναφορικά, μπορείτε να ανατρέξετε στο άρθρο [6] και στα συστήματα Edutella [34] και P2P-DIET [24, 30].

2.1.3 Γενική Αποτίμηση των Δικτύων Ομότιμων Κόμβων

Έχοντας παρουσιάσει τη δομή και τα χαρακτηριστικά των δικτύων ομότιμων κόμβων, αυτή η υποενότητα θα εστιάσει στους λόγους που κάνουν αυτά τα συστήματα ελκυστικά και όλο και πιο δημοφιλή [12]:

- Τα φράγματα εκκίνησης και αύξησης των απαιτήσεων τέτοιων συστημάτων είναι χαμηλά (κλιμάκωση), αφού συνήθως δεν χρειάζονται ειδικές διαχειριστικές ή οικονομικές μεταχειρίσεις, σε αντίθεση με τα κεντρικοποιημένα συστήματα.
- Τα δίκτυα ομότιμων κόμβων προσφέρουν έναν τρόπο συγκέντρωσης και χρήσης μίας τρομακτικά μεγάλης υπολογιστικής και αποθηκευτικής ποσότητας σε υπολογιστές του Διαδικτύου.
- Η αποκεντρωμένη και κατανεμημένη φύση τους τούς δίνει τη δυνατότητα να είναι *εύρωστα σε εσωτερικά σφάλματα* (robust to faults) ή ηθελημένες επιθέσεις, κάνοντάς τα ιδανικά για μακράς διάρκειας αποθηκευτικά μέσα όπως επίσης και για πάρα πολύ μεγάλους υπολογισμούς.

Γι' αυτούς τους λόγους, η κύρια πρόκληση στην περιοχή των Ομότιμων Κόμβων είναι η σχεδίαση και υλοποίηση ενός εύρωστου και κλιμακούμενου κατανεμημένου συστήματος που να αποτελείται από φθηνούς, ανεξάρτητους και αναξιόπιστους[3] υπολογιστές που βρίσκονται

[3]Η χρήση του επιθέτου αναξιόπιστους δεν πρέπει να μας εκπλήσσει. Εξ' ορισμού τα συστήματα ομότιμων κόμβων θα πρέπει να είναι ικανά να επανακάμπτουν κάτω από οποιεσδήποτε περιστάσεις, οπότε ο αποκλεισμός τέτοιων συστημάτων δεν έχει λόγο έκφρασης.

κάτω από ασυσχέτιστους μεταξύ τους διαχειριστικούς τομείς. Οι συμμετέχοντες σε ένα τέτοιο τυπικό σύστημα δύναται να συγκροτούνται από υπολογιστές σε σπίτια, σχολεία, επιχειρήσεις, και να αυξάνουν σε πολλά εκατομμύρια ταυτόχρονους συμμετέχοντες.

2.2 Κατανεμημένοι Πίνακες Κατακερματισμού

Η χρήση της τεχνολογίας των Ομότιμων Κόμβων εγείρει πολλά και ενδιαφέροντα ερευνητικά προβλήματα στην περιοχή των κατανεμημένων συστημάτων. Σε αυτή την υποενότητα θα εστιάσουμε σε ένα από αυτά, το οποίο τυγχάνει να είναι και ένα από τα σπουδαιότερα ζητήματα που απασχόλησαν και συνεχίζουν να απασχολούν την κοινότητα της Επιστήμης της Πληροφορικής από τα πρώτα χρόνια της γέννησής της, το *πρόβλημα της αναζήτησης* (lookup problem). Πώς είναι δυνατόν να εντοπίσεις ένα συγκεκριμένο αντικείμενο δεδομένων σε ένα μεγάλο δίκτυο ομότιμων κόμβων με ένα κλιμακωτό τρόπο, χωρίς την ύπαρξη κάποιου κεντρικού εξυπηρετητή ή την ύπαρξη ιεραρχίας; Οι πρόσφατοι αλγόριθμοι που έχουν αναπτυχθεί από πολλές ερευνητικές ομάδες για αυτό το πρόβλημα της αναζήτησης παρουσιάζουν μία απλή και γενική διεπαφή, τον *Κατανεμημένο Πίνακα Κατακερματισμού* (ΚΠΚ, Distributed Hash Table, DHT). Τα αντικείμενα δεδομένων εισάγονται σε έναν ΚΠΚ και ανακτώνται καθορίζοντας ένα *μοναδικό κλειδί* (unique key) για αυτό το αντικείμενο. Για να υλοποιηθεί η τεχνολογία του ΚΠΚ, ο υποκείμενος αλγόριθμος πρέπει να είναι ικανός να αποφασίζει ποιος κόμβος είναι υπεύθυνος για την αποθήκευση ενός αντικειμένου δεδομένων που σχετίζεται με το δοθέν κλειδί. Για την επίλυση αυτού του προβλήματος, κάθε κόμβος διατηρεί πληροφορίες — όπως την διεύθυνση IP — για έναν μικρό αριθμό κόμβων του δικτύου, σχηματίζοντας, έτσι, ένα δίκτυο επικάλυψης μέσω του οποίου μπορεί να αποστείλει και να παραλάβει δρομολογημένα μηνύματα που αφορούν στην αποθήκευση και ανάκτηση των κλειδιών που είναι συσχετισμένα με κάποια αντικείμενα δεδομένων.

Στις επόμενες υποπαραγράφους θα παρουσιαστούν οι μέθοδοι που χρησιμοποιήθηκαν σταδιακά για την επίλυση του προβλήματος της αναζήτησης των οποίων τα μειονεκτήματα και πλεονεκτήματα έγιναν η αφορμή για την επινόηση του ΚΠΚ. Παράλληλα, θα παρουσιαστεί η ίδια η τεχνολογία των ΚΠΚ.

2.2.1 Το Πρόβλημα της Αναζήτησης — Διάφορες Προσεγγίσεις

Το πρόβλημα της αναζήτησης είναι απλό στην έκφρασή του: Δοθέντος ενός αντικειμένου δεδομένων X, που είναι αποθηκευμένο σε ένα δυναμικό σύνολο κόμβων του συστήματος-δικτύου, βρες το [12].

2.2.1.1 Δομημένη Αναζήτηση

Η πιο απλή προσέγγιση που επιλύει αυτό το πρόβλημα είναι η διατήρηση μίας κεντρικής βάσης δεδομένων, η οποία αντιστοιχίζει ένα όνομα αρχείου με την διεύθυνση των εξυπηρετητών στους οποίους είναι αποθηκευμένο. Το Napster[4] υιοθέτησε αυτήν την προσέγγιση για τους τίτλους των τραγουδιών, αλλά αποδείχτηκε πως είχε έμφυτα προβλήματα *κλιμάκωσης* (scalability) και *προσαρμοστικότητας* (resilience): η βάση μπορεί να γίνει τεράστια σε μέγεθος σε βαθμό που να μην είναι πλέον λειτουργική και το σημαντικότερο απ' όλα είναι ότι αποτελεί ένα κεντρικό σημείο αποτυχίας — για παράδειγμα ο εξυπηρετητής που φιλοξενεί τη βάση μπορεί να μην μπορεί να αντεπεξέλθει σε εκατομμύρια ταυτόχρονες αιτήσεις.

Η παραδοσιακή προσέγγιση για την επίτευξη κλιμάκωσης είναι η χρήση ιεραρχίας, μία προσέγγιση που ακολουθείται από το Σύστημα Πεδίου Ονομάτων του Διαδικτύου, για την αντιστοίχιση των IP διευθύνσεων με τα ονόματα των υπολογιστών. Η αναζήτηση ξεκινά από την ανώτατη βαθμίδα και ακολουθώντας αναφορές από κόμβο σε κόμβο, διανύει ένα μονοπάτι προς χαμηλότερες βαθμίδες φθάνοντας, τελικά, στον κόμβο που περιέχει το επιθυμητό αντικείμενο δεδομένων. Το μειονέκτημα αυτής της προσέγγισης είναι ότι η απομάκρυνση ενός κόμβου από το δίκτυο, λόγω εσωτερικών προβλημάτων, μπορεί να αποβεί καταστροφική για τη διεκπεραίωση της διαδικασίας της αναζήτησης, ειδικά όταν ένας τέτοιος κόμβος είναι η ρίζα, δηλαδή ο ιεραρχικά υψηλότερος και το επικαλυπτόμενο δίκτυο σχηματίζει μία δενδρική δομή. Επιπρόσθετα, είναι εμφανές ότι οι κόμβοι που βρίσκονται υψηλότερα σε αυτή τη δομή εμπλέκονται σε πολύ μεγαλύτερο ποσοστό της αναζήτησης από ότι οι κόμβοι φύλλα.

Αυτές οι δύο προσεγγίσεις αποτελούν παραδείγματα της *δομημένης αναζήτησης* (structured lookup), κατά την οποία κάθε κόμβος διατηρεί ένα καλά-ορισμένο σύνολο πληροφοριών σχετικά με άλλους κόμβους του δικτύου. Το πλεονέκτημα αυτής της μεθόδου είναι ότι εγγυάται την αξιόπιστη ανάκτηση των δεδομένων που έχουν, προηγουμένως, αποθηκευτεί σε κάποιους κόμβους, οι οποίοι παραμένουν συνδεδεμένοι στο δίκτυο και ενεργοί.

[4] `www.napster.com`

2.2.1.2 Συμμετρική Αναζήτηση

Η δομημένη αναζήτηση, όπως είδαμε, επιλύει το πρόβλημα της κλιμάκωσης, αλλά όχι και το πρόβλημα της προσαρμοστικότητας. Προς την επίλυση αυτού του προβλήματος τα συστήματα ομότιμων κόμβων χρησιμοποιούν την έννοια των *συμμετρικών αλγορίθμων αναζήτησης* (symmetric lookup algorithms). Αντίθετα από την ιεραρχική προσέγγιση, κανένας κόμβος δεν είναι σημαντικότερος από κάποιον άλλο όσο αφορά στην διαδικασία της αναζήτησης και κάθε ένας παίρνει μέρος σε ένα μικρό μόνο μέρος των μονοπατιών αναζήτησης του συστήματος. Αυτές οι ιδιότητες επιτρέπουν στους κόμβους να αυτο-οργανώνονται σε μία αποδοτική επικαλύπτουσα δομή.

Η φιλοσοφία αυτής της προσέγγισης είναι η εξής: όταν ένας κόμβος χρειάζεται ένα συγκεκριμένο αντικείμενο δεδομένων X εκπέμπει ένα μήνυμα σε όλους τους γείτονές του, αιτούμενος το αντικείμενο αυτό. Οι κόμβοι που λαμβάνουν μία τέτοια αίτηση συμβουλεύονται την τοπική τους βάση δεδομένων για την ύπαρξη του αντικειμένου X σε αυτούς. Εάν αυτό περιέχεται στη βάση, τότε απαντούν στο μήνυμα που δέχτηκαν στέλνοντας πίσω το αντικείμενο X. Σε διαφορετική περίπτωση αναλαμβάνουν να προωθήσουν την αίτηση στους γείτονές τους, οι οποίοι εκτελούν το ίδιο πρωτόκολλο. Παρ' όλα αυτά, η προσέγγιση της εκπομπής μίας αίτησης στους γείτονες ενός κόμβους και από εκεί στους γείτονες των γειτόνων δεν επιτυγχάνει σε μεγάλο βαθμό την έννοια της κλιμάκωσης και επιπρόσθετα επιβαρύνει τους γείτονες υπολογιστικά υπό την έννοια ότι αυτοί θα πρέπει να εμπλακούν ενεργά στην αναζήτηση κατασκευάζοντας ένα παρόμοιο μήνυμα αίτησης του αντικειμένου X και προωθώντας το στους γείτονές τους. Μία προσέγγιση που υιοθετήθηκε για τον χειρισμό τέτοιων προβλημάτων κλιμάκωσης είναι η προσθήκη υπερ-κόμβων σχηματίζοντας μία ιεραρχική δομή. Κάθε κόμβος C με χαμηλή ιεραρχική θέση συνδέεται σε έναν υπερ-κόμβο SP, ο οποίος διαδραματίζει το ρόλο του αντιπροσώπου και αναλαμβάνει να διεκπεραιώσει τις αιτήσεις αναζήτησης του κόμβου C. Τώρα, η μετάδοση των μηνυμάτων γίνεται σε ένα σαφώς πολύ μικρότερο υποσύνολο από το αρχικό, αλλά αυτό το κέρδος αντισταθμίζεται από την πολυέξοδη απαίτηση οι υπερ-κόμβοι να είναι προσαρμόσιμοι σε σφάλματα. Επιπλέον, αυτές οι δυσάρεστες περιπτώσεις δεν εξασφαλίζουν το γεγονός της εύρεσης ενός αντικειμένου δεδομένων ακόμα και αν είναι αποθηκευμένο σε κάποιον κόμβο. Σε αυτή την προσέγγιση στράφηκε η Πλατφόρμα Ομότιμων Κόμβων FastTrack και έγινε πιο δημοφιλής από εφαρμογές, όπως το KaZaA[5].

[5] www.kazaa.com

2.2.1.3 Συμμετρική Αναζήτηση Χρησιμοποιώντας Αδόμητους Πίνακες Δρομολόγησης

Το FreeNet [13] χρησιμοποιεί μία καινοτόμα συμμετρική τακτική αναζήτησης. Σύμφωνα με αυτή, οι επερωτήσεις για κάποιο αντικείμενο δεδομένων προωθούνται από κόμβο σε κόμβο μέχρι αυτό να βρεθεί. Η στρατηγική αυτή βασίζεται σε *αδόμητους πίνακες δρομολόγησης* (unstructured routing tables), οι οποίοι δημιουργούνται και ενημερώνονται δυναμικά χρησιμοποιώντας *κρυφή μνήμη* (caching). Έτσι, ο απώτερος σκοπός του FreeNet είναι η επίτευξη αποδοτικής αναζήτησης μαζί με τη διατήρηση ανωνυμίας. Για να το κάνει αυτό, αποφεύγει να συσχετίζει ένα έγγραφο με έναν εξυπηρετητή ή ακόμα με οποιαδήποτε άλλη τοπολογικής φύσεως πληροφορία. Αυτό έχει ως αποτέλεσμα μη δημοφιλή έγγραφα να εξαφανίζονται σταδιακά λόγω του ότι κανένας εξυπηρετητής δεν είναι σε θέση να κρατήσει αντίγραφό τους. Συχνά, επίσης, μία ενέργεια αναζήτησης χρειάζεται να επισκεφτεί μεγάλο ποσοστό των κόμβων του συστήματος και επιπλέον δεν υπάρχει καμία εγγύηση για την εύρεση του εγγράφου.

2.2.1.4 Συμμετρική και Δομημένη Αναζήτηση

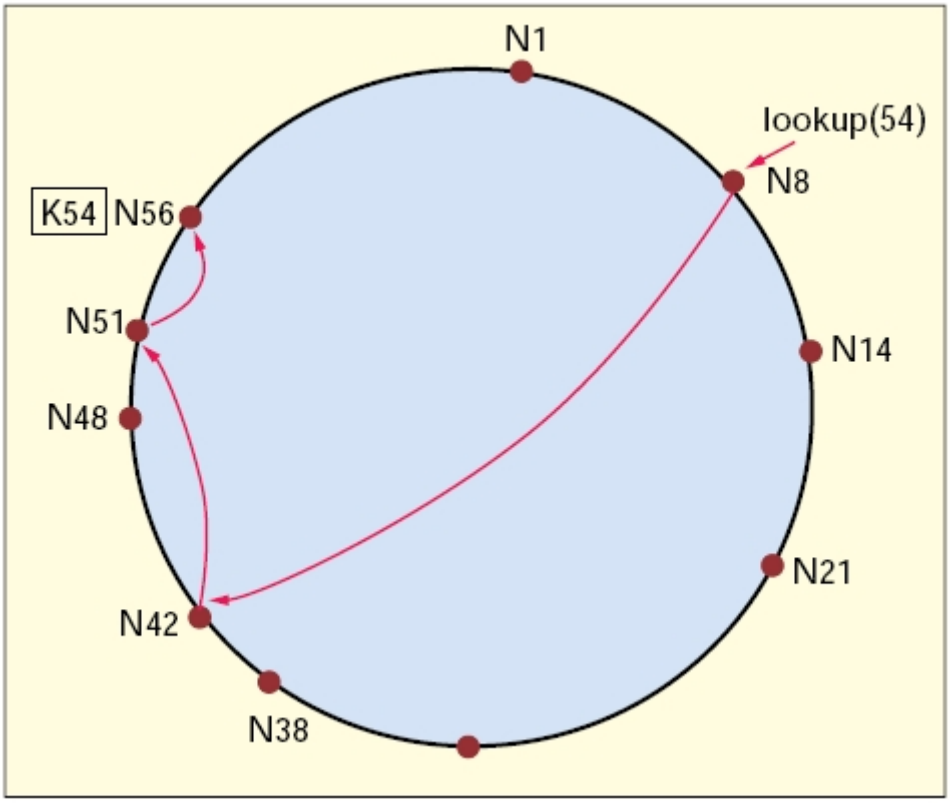
Οι πιο πρόσφατοι αλγόριθμοι αναζήτησης σε Δίκτυα Ομότιμων Κόμβων συνδυάζουν τη φιλοσοφία της δομημένης και συμμετρικής αναζήτησης και έχουν υλοποιηθεί από συστήματα, όπως το CAN [31], Chord [14], Kademlia [26], Pastry [3], Tapestry [17], Viceroy [9] κ.α. Όλα υλοποιούν την φιλοσοφία των ΚΠΚ και παρέχουν την ίδια διεπαφή για τη χρήση του. Η χρήση αλγορίθμων που χρησιμοποιούν την τεχνολογία των ΚΠΚ προσδίδει στο δίκτυο κλιμάκωση και αποδοτικότητα όσο αφορά στον αριθμό των κόμβων, των αφίξεων και αναχωρήσεων, στην εύρεση ενός συγκεκριμένου αντικειμένου δεδομένων με χαμηλή καθυστέρηση, στην ευκολία της διατήρησης και ενημέρωσης των πινάκων δρομολόγησης κάθε κόμβου καθώς επίσης και στην ισορροπημένη κατανομή των κλειδιών μεταξύ των κόμβων.

2.2.2 Κατηγοριοποίηση Κατανεμημένων Πινάκων Κατακερματισμού

Κάθε ΚΠΚ υλοποιεί μία και μοναδική λειτουργία, την $lookup(key)$. Αυτή παράγει την τοποθεσία στο δίκτυο του κόμβου που είναι υπεύθυνος για το δεδομένο κλειδί. Για παράδειγμα, μία απλή κατανεμημένη εφαρμογή αποθήκευσης μπορεί να χρησιμοποιήσει αυτή τη διεπαφή ως εξής: Για να δημοσιοποιήσει ένα αρχείο υπό ένα μοναδικό όνομα (κλειδί), ο πάροχος μετατρέπει το όνομα του σε ένα αριθμητικό κλειδί χρησιμοποιώντας μία συνάρτηση κατα-

κερματισμού, όπως η *SHA-1* (Secure Hash Algorithm-1) και μετά καλεί την $lookup(key)$, η οποία θα του επιστρέψει τον κόμβο που είναι υπεύθυνος για αυτό το κλειδί. Στη συνέχεια στέλνει σε αυτόν το αρχείο, το οποίο και αποθηκεύεται. Ένας καταναλωτής που επιθυμεί να διαβάσει αυτό το αρχείο, γνωρίζοντας, φυσικά το όνομά του, το μετατρέπει με την παραπάνω συνάρτηση στο αριθμητικό του κλειδί και καλεί την $lookup(key)$. Ο υπεύθυνος κόμβος θα απαντήσει στέλνοντας ένα αντίγραφο σε αυτόν.

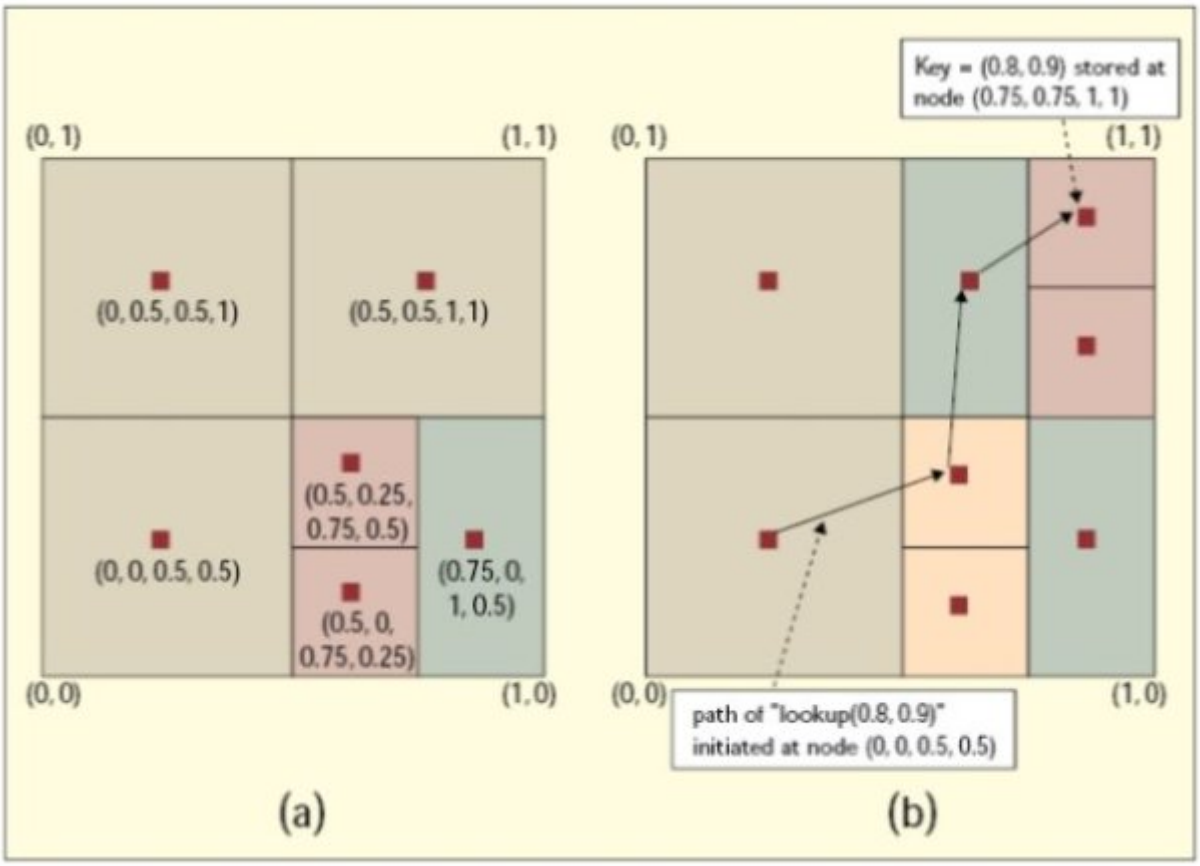
Αυτή είναι η διεπαφή που πρέπει να προσφέρει και να υλοποιεί κάθε τέτοιος ΚΠΚ. Παρ' όλα αυτά, οι ΚΠΚ διακρίνονται σε διάφορες κατηγορίες ανάλογα με τον τρόπο που υλοποιούν τις επόμενες διαδικασίες:



Σχήμα 2.2: Μονοδιάστατη δρομολόγηση στο σύστημα Chord.

- ***Αντιστοίχηση κλειδιών σε κόμβους***: τα κλειδιά και οι κόμβοι αναγνωρίζονται με έναν δυαδικό αριθμό που στη βιβλιογραφία αναφέρεται ως *αναγνωριστικό* (identifier, ID). Τα κλειδιά αποθηκεύονται σε έναν ή περισσότερους κόμβους με αναγνωριστικά "κοντά" στο κλειδί. Η έννοια του "κοντά" είναι σχετική και ποικίλει από σύστημα σε σύστημα. Καθορίζεται από τη *συνάρτηση απόστασης* (distance function). Περισσότερα θα αναφερθούν παρακάτω.

- ***Δρομολόγηση μηνυμάτων για επερωτήσεις κλειδιών***: κάθε κόμβος που λαμβάνει ένα τέτοιο μήνυμα, αν έχει το κλειδί k, τότε απαντάει με αυτόν στον αποστολέα του μηνύματος, διαφορετικά συμβουλευόμενος τις τοπικές του πληροφορίες προωθεί το μήνυμα στους γείτονές του. Αυτές οι τοπικές πληροφορίες συνήθως υλοποιούνται με τη δομή του *πίνακα δρομολόγησης* (routing table). Και σε αυτή τη διαδικασία υπάρχει μία διαφοροποίηση μεταξύ των συστημάτων που προσφέρουν την διεπαφή του ΚΠΚ. Η δρομολόγηση και η συντήρηση των πινάκων δρομολόγησης μπορεί να γίνει είτε σε μία διάσταση (βλέπε Σχήμα 2.2), είτε σε πολλαπλές (βλέπε Σχήμα 2.3).



Σχήμα 2.3: Δισδιάστατη δρομολόγηση του συστήματος CAN και προσομοίωση εκτέλεσης της διαδικασίας αναζήτησης (lookup()) .

- ***Διαχείριση δυναμικότητας δικτύου***: οι ΚΠΚ μπορούν να προσαρμόζονται σε εισαγωγές, αναχωρήσεις και αποτυχίες κόμβων και να ανανεώνουν τους πίνακες δρομολόγησης με μικρή προσπάθεια. Σε αυτή την ενότητα εμπίπτει και η ανάπτυξη αλγορίθμων που προσφέρουν τη δυνατότητα *ανοχής σε σφάλματα* (fault tolerance) καθώς επίσης και ανάκαμψης από αυτά.

Στις επόμενες υποενότητες θα αναφερθούμε σε δύο σημαντικά Συστήματα Ομότιμων Κόμβων που υλοποιούν και προσφέρουν τη διεπαφή του Κατανεμημένου Πίνακα Κατακερματισμού με διαφορετικό τρόπο όσον αφορά στα προαναφερθέντα θέματα. Αυτά είναι τα Chord [14] και το Pastry [3].

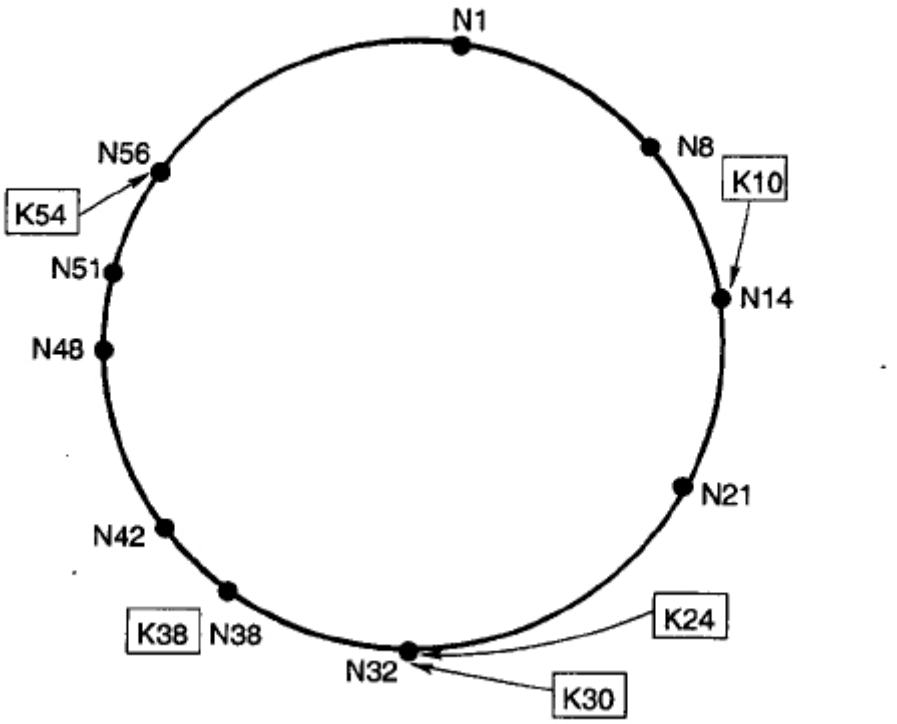
2.2.3 Chord

Το Chord [14] είναι ένα δίκτυο ομότιμων κόμβων που προσφέρει ένα κλιμακούμενο πρωτόκολλο αναζήτησης υλοποιώντας τη διεπαφή του ΚΠΚ. Αυτό το πρωτόκολλο υποστηρίζει μία και μόνο λειτουργία: δοθέντος ενός κλειδιού, το αντιστοιχίζει σε έναν κόμβο του δικτύου. Ο τρόπος αυτής της αντιστοίχισης γίνεται ακολουθώντας μία παραλλαγή του *συνεπή κατακερματισμού* (consistent hashing). Η μέθοδος αυτή τείνει να εξισορροπεί το φορτίο κάθε κόμβου κατανέμωντας περίπου το ίδιο πλήθος κλειδιών σε αυτούς και επιπλέον περιλαμβάνει σχετικά μικρή μετακίνηση κλειδιών μεταξύ των κόμβων όταν κόμβοι προσχωρούν ή εγκαταλείπουν το σύστημα. Η συνεπής συνάρτηση κατακερματισμού που χρησιμοποιεί το Chord λειτουργεί ως εξής: αναθέτει σε κάθε κόμβο και κάθε κλειδί (που αντιπροσωπεύει ένα αντικείμενο δεδομένων) ένα αναγνωριστικό m-διφίων[6] χρησιμοποιώντας μία βασική συνάρτηση κατακερματισμού, όπως η SHA-1 [10]. Επιπλέον, το m διαλέγεται σχετικά μεγάλο έτσι ώστε να αποφευχθεί η περίπτωση της ανάθεσης του ίδιου αναγνωριστικού σε δύο διαφορετικούς κόμβους. Για την περίπτωση των κόμβων, το αναγνωριστικό υπολογίζεται από τον κατακερματισμό της IP διεύθυνσής του, ενώ για την περίπτωση των κλειδιών πρώτα επιλέγεται ένα κατάλληλο κλειδί, το οποίο θα αντιστοιχεί σε ένα συγκεκριμένο αντικείμενο δεδομένων, και στη συνέχεια κατακερματίζεται. Για παράδειγμα, αν το Chord χρησιμοποιηθεί για την φιλοξενία μίας εφαρμογής διαμοιρασμού αρχείων, τότε ένα κατάλληλο κλειδί που θα ταυτοποιεί ένα πόρο (ένα αρχείο) μπορεί κάλλιστα να είναι το όνομα του. Αυτό θα αποτελέσει το κλειδί. Το αναγνωριστικό κλειδί θα υπολογιστεί από τον κατακερματισμό του κλειδιού αυτού.

Στη συνέχεια περιγράφεται ο τρόπος, σύμφωνα με τον οποίο υλοποιείται η αντιστοίχιση κλειδιών σε κόμβους. Τα αναγνωριστικά ταξινομούνται σε έναν ιδεατό δακτύλιο αναγνωριστικών modulo 2^m, δηλαδή από 0 έως $2^m - 1$. Έστω ότι χρησιμοποιείται η συνάρτηση H ως συνάρτηση κατακερματισμού. Το κλειδί k ανατίθεται στον πρώτο κόμβο του οποίου το αναγνωριστικό είναι ίσο ή ακολουθεί το αναγνωριστικό κλειδί του κλειδιού k, $H(k)$. Αυτός ο κόμβος καλείται *διάδοχος κόμβος* (successor node) του κλειδιού k και συμβολίζεται ως

[6] Το διφίο αποτελεί την ελληνική μετάφραση του ξενόγλωσσου όρου bit ή αλλιώς διαδικό ψηφίο.

$successor(H(k))$. Πιο περιγραφικά, αν τα αναγνωριστικά αναπαρίστανται με έναν κύκλο αριθμών από το 0 έως το $2^m - 1$, τότε ο διάδοχος του αναγνωριστικού κλειδιού $H(k)$ είναι ο πρώτος κόμβος μετά το $H(k)$ κατά τη φορά των δεικτών του ρολογιού. Συχνά θα λέμε πως ο κόμβος με αναγνωριστικό αριθμό $successor(H(k))$ είναι υπεύθυνος για το αναγνωριστικό κλειδί $H(k)$ ή για το κλειδί k. Επομένως, εύκολα συμπεραίνεται πως ένας κόμβος $successor(H(k))$ είναι υπεύθυνος για ένα σύνολο αναγνωριστικών κλειδιών.



Σχήμα 2.4: Ο δακτύλιος αναγνωριστικών του Chord με m = 6.

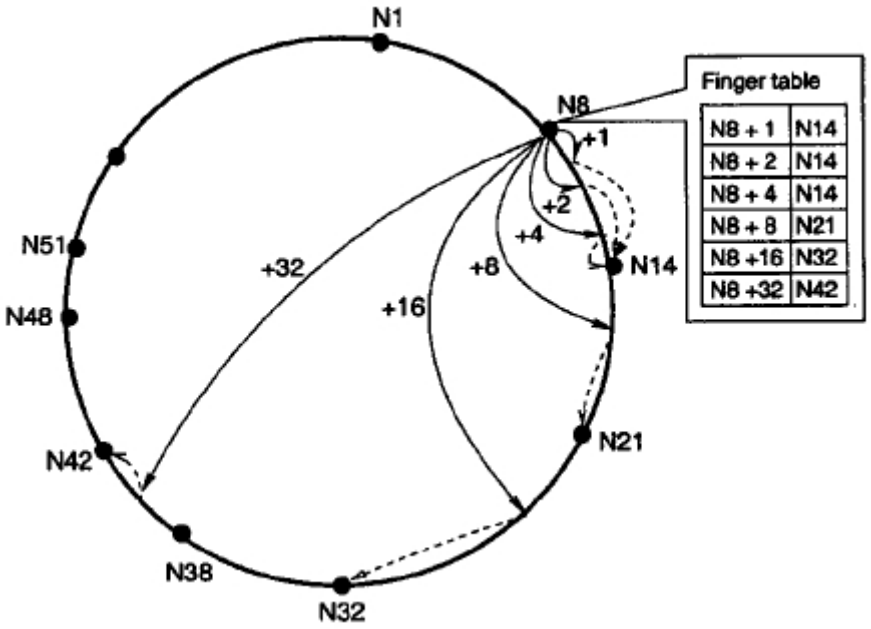
Στο Σχήμα 2.4, φαίνεται ένας δακτύλιος αναγνωριστικών με 64 αναγνωριστικά $(m = 6)$. Ο δακτύλιος αποτελείται από 10 μόλις κόμβους στους οποίους έχουν αποθηκευτεί, συνολικά, 5 κλειδιά. Ο διάδοχος κόμβος του αναγνωριστικού κλειδιού 10 είναι ο $N14$, επομένως το κλειδί 10 θα αποθηκευτεί σε αυτόν τον κόμβο. Όμοια, τα αναγνωριστικά κλειδιά 24 και 30 θα αποθηκευτούν στον κόμβο $N32$, το 38 στον κόμβο $N38$ και το 54 στον $N56$. Συνδυάζοντας τα παραπάνω με το παράδειγμα θα λέγαμε ότι ο κόμβος $N14$ θα είναι υπεύθυνος για όλα τα αναγνωριστικά κλειδιά του διαστήματος $(8, 14]$, ο κόμβος $N32$ για όλα τα κλειδιά του διαστήματος $(21, 32]$, ενώ ο $N56$ για αυτά του $(51, 56]$.

Ο συνεπής κατακερματισμός είναι σχεδιασμένος έτσι ώστε να προκαλεί την ελάχιστη διαταραχή των κλειδιών κατά την προσχώρηση και αποχώρηση κόμβων από και προς το δίκτυο.

Για την διατήρηση του συνεπή κατακερματισμού και της αντιστοίχισης των κλειδιών στις περιπτώσεις που ένας κόμβος n εισέρχεται στο δίκτυο, συγκεκριμένα κλειδιά που προηγουμένως είχαν ανατεθεί στον διάδοχο κόμβο του n τώρα ανατίθενται σε αυτόν. Αντίστοιχα, όταν ο κόμβος n εγκαταλείπει το δίκτυο, όλα τα κλειδιά που του είχαν ανατεθεί επαναανατίθενται στον διάδοχο του n. Καμία περαιτέρω αλλαγή δεν χρειάζεται να συμβεί όσο αφορά στην ανάθεση των κλειδιών στους κόμβους. Στο παραπάνω παράδειγμα, εάν ένας κόμβος προοριζόταν να προσχωρήσει στο δίκτυο με αναγνωριστικό 26, τότε θα συνελάμβανε το κλειδί με αναγνωριστικό 24 από τον κόμβο $N32$.

Παρουσιάζοντας και καλύπτοντας το θέμα του καταμερισμού των αναγνωριστικών κλειδιών στους κόμβους του δικτύου υπεισέρχεται αυτομάτως το θέμα του εντοπισμού ενός συγκεκριμένου αναγνωριστικού κλειδιού και εύλογα εγείρεται το ερώτημα του "πόσο αποτελεσματικά μπορεί να εντοπιστεί ένα συγκεκριμένο αναγνωριστικό κλειδί", όπου με τον όρο "αποτελεσματικά" εννοείται η ταχύτητα εντοπισμού όσον αφορά στην πολυπλοκότητα χρόνου. Έχοντας ως γνώμονα το γεγονός του ότι σε ένα δίκτυο Chord κάθε κόμβος γνωρίζει τον διάδοχό του, μια ερώτηση για την εύρεση του κόμβου που είναι υπεύθυνος για το κλειδί k μπορεί να απαντηθεί σε $O(N)$ βήματα, όπου N είναι το πλήθος των κόμβων του δικτύου. Για να βελτιωθεί αυτό το όριο, το σύστημα Chord διατηρεί σε κάθε κόμβο ένα ποσό πληροφορίας σχετικά με έναν συγκεκριμένο αριθμό κόμβων του δικτύου. Αυτή η πληροφορία είναι οργανωμένη στην δομή που αναφέρθηκε προηγουμένως ως *πίνακας δρομολόγησης* και στο Chord ονομάζεται *πίνακας δεικτών* (finger table). Κάθε τέτοιος πίνακας περιέχει το πολύ m εγγραφές. Κάθε εγγραφή i στον πίνακα δεικτών του κόμβου n περιέχει πληροφορίες, όπως την IP διεύθυνση και το αναγνωριστικό, του πρώτου κόμβου s που διαδέχεται τον n κατά τουλάχιστον 2^{i-1} στο δακτύλιο των αναγνωριστικών κλειδιών. Δηλαδή, ο s είναι ο διάδοχος του κόμβου $H(n)+2^{i-1}$, για $1 \leq i \leq m$, ενώ όλες οι αριθμητικές πράξεις υπολογίζονται βάσει modulo 2^m. Ένας τέτοιος κόμβος, s, αναφέρεται ως ο i-οστός *δείκτης* (finger) του κόμβου n. Σημειώνεται, ότι ο πρώτος δείκτης ενός κόμβου n είναι ο διάδοχός του. Σύμφωνα με την αριθμητική σχέση του κόμβου n και των δεικτών του, κάθε δείκτης i βρίσκεται σε διπλάσια απόσταση από την απόσταση του $i-1$ δείκτη από τον κόμβο n. Έτσι είναι εμφανές ότι με αυτή την τεχνική η αναζήτηση ενός κόμβου που είναι υπεύθυνος για ένα συγκεκριμένο κλειδί k μπορεί να επιταχυνθεί σημαντικά. Για να ποσοτικοποιήσουμε αυτόν τον ισχυρισμό μπορούμε να αναφέρουμε ότι εάν οι πίνακες δεικτών έχουν μέγεθος $m = O(logN)$, τότε η εύρεση του διαδόχου του κόμβου n μπορεί να γίνει με μεγάλη πιθανότητα σε $O(logN)$ βήματα. Η φράση "με μεγάλη πιθανότητα" επιδέχεται αρκετή συζήτηση η οποία μπορεί να

βρεθεί στο [14]. Ένα στιγμιότυπο του πίνακα δεικτών ενός κόμβου του συστήματος Chord υπάρχει στο Σχήμα 2.5.



Σχήμα 2.5: Παράδειγμα του πίνακα δεικτών του κόμβου N8 του συστήματος Chord με m = 6.

Μέχρι αυτού του σημείου έχουν καλυφθεί οι δύο από τις τρεις διαδικασίες που μία υλοποίηση ενός ΚΠΚ πρέπει να υλοποιεί και να προσφέρει, όπως αυτές αναφέρθηκαν στην προηγούμενη υποενότητα — Κατηγοριοποίηση Κατανεμημένων Πινάκων Κατακερματισμού. Στη συνέχεια ακολουθεί η τεχνική που ακολουθεί το Chord σχετικά με τον τρόπο διαχείρισης και αντιμετώπισης της δυναμικότητας του δικτύου.

Για την απλοποίηση των αφίξεων και των αναχωρήσεων, κάθε κόμβος n διατηρεί έναν δείκτη στον *προκάτοχό* του (predecessor), δηλαδή στον πρώτο κόμβο με αντίθετη φορά από αυτή του ρολογιού στο δακτύλιο αναγνωριστικών ξεκινώντας από τον n. Και σε αυτή την περίπτωση οι πληροφορίες που αποθηκεύονται σχετικά με αυτόν τον κόμβο είναι η IP διεύθυνσή του και το αναγνωριστικό του. Όταν ένας κόμβος n θελήσει να συνδεθεί σε ένα δίκτυο Chord βρίσκει έναν κόμβο n', που είναι ήδη συνδεδεμένος σε αυτό, χρησιμοποιώντας κάποιο εξωτερικό μέσο (από ένα δικτυακό τόπο με ασφαλή και περιορισμένη πρόσβαση που διατηρεί τέτοιου είδους πληροφορία) και κατόπιν ζητάει από αυτόν να τον βοηθήσει στην εύρεση της θέσης του μέσα στο δίκτυο εντοπίζοντας τον διάδοχο του [15]. Για την συντήρηση του πίνακα δεικτών, κάθε κόμβος εκτελεί περιοδικά έναν *αλγόριθμο σταθεροποίησης* (stabilization algorithm) για να

ενημερωθεί για τους κόμβους που έχουν πρόσφατα συνδεθεί στο δίκτυο. Σύμφωνα με αυτό το πρωτόκολλο, ένας κόμβος n ζητάει από τον διάδοχό του να του πει τον προκάτοχό του, p. Αν ο p είναι διαφορετικός από τον n, τότε σημαίνει ότι ο p είναι ένας νεο-εισερχόμενος κόμβος και πρέπει να γίνει ο διάδοχος του n. Επιπλέον, κάθε κόμβος εκτελεί περιοδικά δύο επιπλέον αλγόριθμους για να ελέγξει ότι ο πίνακας δεικτών και ο δείκτης στον προκάτοχό του είναι έγκυροι [15]. Αποδεικνύεται ότι αν οι δείκτες στους διαδόχους των κόμβων είναι σωστοί και ο χρόνος που χρειάζεται για την διόρθωση των πινάκων δεικτών είναι λιγότερος από το χρόνο που χρειάζεται για να διπλασιαστεί το δίκτυο σε μέγεθος, οι επερωτήσεις εντοπισμού ενός κλειδιού k μπορούν να απαντηθούν σωστά με υψηλή πιθανότητα σε $O(logN)$ βήματα [15].

Τέλος, όσον αφορά στο θέμα των αποτυχιών των κόμβων, όταν ένα κόμβος n αποτύχει, οι κόμβοι που τον συμπεριλάμβαναν στον πίνακα δεικτών τους θα πρέπει να τον αντικαταστήσουν με τον διάδοχο του n. Επιπλέον, ένα τέτοιο γεγονός δεν θα πρέπει να εμποδίζει την εξέλιξη μίας επερώτησης για ένα κλειδί k εν όσω το δίκτυο επανασταθεροποιείται. Για την επίτευξη αυτών των προδιαγραφών κάθε κόμβος του Chord διατηρεί μία *λίστα διαδόχων* (successor list) μεγέθους r που περιέχει τους r πρώτους (κοντινότερους) διαδόχους του. Αυτή η λίστα χρησιμοποιείται όταν ο κόμβος n έχει αποτύχει. Σε αυτή την περίπτωση όταν ένας κόμβος n' αντιληφθεί ότι ο διάδοχός του έχει αποτύχει αντικαθιστά την πρώτη εγγραφή του πίνακα δεικτών του με την πρώτη εγγραφή του πίνακα των διαδόχων του, που στην ουσία ήταν ο διάδοχος του n πριν αυτός αποτύχει. Εάν ένας κόμβος επιλέξει να αποχωρήσει οικειοθελώς από το δίκτυο, τότε μπορεί να ενημερώσει τον διάδοχό και τον προκάτοχό του έτσι ώστε να μπορούν να αλλάξουν τους δείκτες τους και επιπλέον να μπορεί να μεταφέρει τα κλειδιά στον διάδοχό του. Πρακτικά, σύμφωνα με στατιστικά πειράματα, ακόμα και μικρές τιμές του r είναι αρκετές για να επιτευχθεί η ευρωστία του δικτύου [15]. Αποδεικνύεται με μεγάλη πιθανότητα ότι όποιος κόμβος συνδέεται ή αποσυνδέεται από ένα δίκτυο Chord μπορεί να χρησιμοποιήσει $O(log^2N)$ μηνύματα έτσι ώστε να ενημερωθούν σωστά όλοι οι δείκτες στους διαδόχους, στους προκατόχους και οι πίνακες δεικτών όλων των κόμβων [14].

2.2.4 Pastry

Σε αυτή την ενότητα, εκτός της αναφοράς που θα γίνει στις τρεις συνιστώσες των ΚΠΚ που υλοποιούνται από το σύστημα Pastry, θα παρουσιαστεί επίσης και μία εφαρμογή για τη διαχείριση της αποθήκευσης των αντικειμένων δεδομένων, το σύστημα PAST [5]. Το PAST είναι υλοποιημένο πάνω από το επίπεδο του Pastry, χρησιμοποιώντας το για τη δρομολόγηση

των μηνυμάτων και την εύρεση αντικειμένων. Η παρουσίαση θα είναι λεπτομερέστερη καθώς το σύστημα Pastry είναι αυτό που χρησιμοποιήθηκε στην υλοποίηση της αρχιτεκτονικής LibraRing.

2.2.4.1 Γενική Περιγραφή

Το σύστημα Pastry [3] είναι ένα επικαλυπτόμενο δίκτυο ομότιμων κόμβων που προσφέρει δυνατότητα δρομολόγησης μηνυμάτων σε επίπεδο εφαρμογής και ένα κλιμακούμενο πρωτόκολλο αναζήτησης υλοποιώντας τη διεπαφή του ΚΠΚ. Το Pastry έχει σχεδιασθεί και υλοποιηθεί ώστε να αποτελεί ένα πολύ μεγάλο επικαλυπτόμενο δίκτυο ομότιμων κόμβων δια-συνδεδεμένων μέσω του Διαδικτύου. Έχει τη δυνατότητα υποστήριξης μίας μεγάλης ποικιλίας εφαρμογών ομότιμων κόμβων, συμπεριλαμβανομένων καθολικής-παγκόσμιας αποθήκευσης δεδομένων (PAST), διαμοιρασμού αρχείων (Scrivener[7]), επικοινωνίας και ονοματολογίας ομάδων (Scribe[8], POST[9]).

Κάθε κόμβος του δικτύου Pastry έχει ένα μοναδικό *αναγνωριστικό* ή *αναγνωριστικό κόμβου* (nodeId). Δοθέντος ενός μηνύματος και ενός κλειδιού, ένας κόμβος του Pastry δρομολογεί αποδοτικά αυτό το μήνυμα στον κόμβο με αναγνωριστικό που είναι αριθμητικά πλησιέστερα στο κλειδί, ανάμεσα σε όλους τους εν ζωή κόμβους του Pastry. Κάθε κόμβος διατηρεί πληροφορίες για τους άμεσους γείτονές του σύμφωνα με τα αναγνωριστικά τους και ειδοποιεί τις εφαρμογές που έχουν υλοποιηθεί πάνω σε αυτό το επίπεδο για την *άφιξη νέων κόμβων* (node arrival), τις *αποτυχίες κόμβων* (node failure) και για την *επανάκαμψη κόμβων* (node recovery). Το σύστημα Pastry λαμβάνει υπ' όψη του την τοπικότητα του δικτύου· ψάχνει να ελαχιστοποιήσει την απόσταση που θα διανύσει ένα μήνυμα, σύμφωνα με μία βαθμωτή *μετρική εγγύτητας* (proximity metric), όπως ο αριθμός των *αναπηδήσεων* (hops) κατά τη δρομολόγηση.

Το Pastry είναι απόλυτα κατανεμημένο, κλιμακούμενο, και αυτο-οργανώμενο· προσαρμόζεται αυτόματα σε αφίξεις, αναχωρήσεις και αποτυχίες κόμβων.

[7] http://www.cs.rice.edu/CS/Systems/Scrivener/default.htm
[8] http://research.microsoft.com/~antr/Scribe/default.htm
[9] http://www.cs.rice.edu/CS/Systems/POST/default.htm

2.2.4.2 Αντιστοίχηση Κλειδιών σε Κόμβους

Σε κάθε κόμβο του δικτύου Pastry ανατίθεται ένα *αναγνωριστικό κόμβου* (nodeId) μήκους 128 bit. Το αναγνωριστικό κόμβου ή απλά αναγνωριστικό χρησιμεύει στο να υποδεικνύει την θέση ενός κόμβου σε ένα κυκλικό χώρο που ορίζουν όλα τα δυνατά αναγνωριστικά και κυμαίνεται από 0 έως $2^{128} - 1$. Το αναγνωριστικό ανατίθεται τυχαία όταν κάποιος κόμβος συμμετέχει στο σύστημα. Τα αναγνωριστικά παράγονται κατά τέτοιο τρόπο έτσι ώστε το σύνολο των αναγνωριστικών που απαρτίζουν το δίκτυο να είναι κατανεμημένο ομοιόμορφα στον χώρο των αναγνωριστικών. Για παράδειγμα, τα αναγνωριστικά μπορούν να παραχθούν από τον υπολογισμό του κρυπτογραφικού κατακερματισμού του δημόσιου κλειδιού του κόμβου ή της διεύθυνσης IP του. Το αποτέλεσμα αυτής της τυχαίας ανάθεσης αναγνωριστικών είναι ότι με μεγάλη πιθανότητα οι κόμβοι με γειτονικά αναγνωριστικά δεν είναι γειτονικοί όσον αφορά στην γεωγραφική τους θέση, ιδιοκτησία, δικαιοδοσία κ.λπ.

Υποθέτοντας ότι ένα δίκτυο αποτελείται από N κόμβους, το Pastry μπορεί να δρομολογήσει ένα δεδομένο κλειδί στον αριθμητικά κοντινότερο κόμβο σε λιγότερα από $\lceil log_{2^b} N \rceil$ βήματα (το b είναι μία παράμετρος με τυπική τιμή 4). Παρά τις ταυτόχρονες αποτυχίες κόμβων, η παράδοση ενός μηνύματος είναι εγγυημένη εκτός και αν $\lfloor |L|/2 \rfloor$ κόμβοι με γειτονικά αναγνωριστικά αποτύχουν ταυτόχρονα (το L είναι μία παράμετρος με τυπική τιμή 16 ή 32). Για μία οπτικοποιημένη άποψη αυτού του σχεδιασμού μπορείτε να ανατρέξετε στο Σχήμα 2.4, του δακτυλίου του Chord με το οποίο δεν διαφέρει σε τίποτα.

2.2.4.3 Δρομολόγηση Μηνυμάτων

Για το σκοπό της δρομολόγησης μηνυμάτων, τα αναγνωριστικά κόμβων θεωρούνται ότι είναι μία ακολουθία ψηφίων στη βάση 2^b. Το Pastry δρομολογεί τα μηνύματα στον κόμβο του οποίου το αναγνωριστικό είναι αριθμητικά κοντινότερο στο δεδομένο κλειδί των μηνυμάτων. Αυτό επιτυγχάνεται ως εξής: Σε κάθε βήμα δρομολόγησης, ένας κόμβος, υπό κανονικές συνθήκες, προωθεί το μήνυμα σε έναν κόμβο του οποίου το αναγνωριστικό μοιράζεται με το κλειδί ένα πρόθεμα με μήκος τουλάχιστον ένα ψηφίο (ή b bits) περισσότερο από ότι το μήκος του προθέματος που μοιράζεται το κλειδί με το αναγνωριστικό του κόμβου που αποστέλλει το μήνυμα. Εάν δεν είναι γνωστός ή δεν υπάρχει ένας τέτοιος κόμβος, τότε το μήνυμα προωθείται στον κόμβο εκείνο που το αναγνωριστικό του μοιράζεται με το κλειδί ένα πρόθεμα μήκους όσου και αυτού του προθέματος μεταξύ του κλειδιού και του κόμβου που αποστέλλει το μήνυμα και που είναι αριθμητικά κοντινότερο στο κλειδί από ότι είναι ο κόμβος

που αποστέλλει το μήνυμα. Για την υποστήριξη αυτής της διαδικασίας δρομολόγησης κάθε κόμβος διατηρεί κάποια *κατάσταση δρομολόγησης* (routing state), η οποία περιγράφεται στη συνέχεια.

Κάθε κόμβος του Pastry διατηρεί έναν *πίνακα δρομολόγησης* (routing table), ένα *σύνολο γειτόνων* (neighborhood set) και ένα *σύνολο φύλλων* (leaf set). Θα ξεκινήσουμε με την περιγραφή του πίνακα δρομολόγησης. Ο πίνακας δρομολόγησης ενός κόμβου, R, είναι οργανωμένος σε $\lceil log_{2^b} N \rceil$ γραμμές με $2^b - 1$ εγγραφές η κάθε μία. Κάθε μία από αυτές τις εγγραφές της γραμμής n αναφέρεται στον κόμβο εκείνο του οποίου το αναγνωριστικό μοιράζεται με το αναγνωριστικό του τρέχοντος κόμβου ένα πρόθεμα μήκους n ψηφίων, αλλά το ψηφίο $n + 1$ έχει κάποια από τις δυνατές τιμές στο διάστημα $2^b - 1$ και διαφέρει από το αντίστοιχο ψηφίο του αναγνωριστικού του τρέχοντος κόμβου.

Κάθε εγγραφή στον πίνακα δρομολόγησης περιέχει τη διεύθυνση IP ενός από τους εν δυνάμει πολλούς κόμβους των οποίων τα αναγνωριστικά έχουν το κατάλληλο πρόθεμα· στην πράξη αυτός ο κόμβος δεν επιλέγεται τυχαία, αλλά έχει την ιδιότητα να είναι ο πλησιέστερος κόμβος σύμφωνα με την μετρική εγγύτητας. Αποδεικνύεται ότι μία τέτοια επιλογή παρέχει καλές ιδιότητες τοπικότητας. Εάν δεν υπάρχει κάποιος κόμβος με κατάλληλο αναγνωριστικό, τότε η αντίστοιχη εγγραφή στον πίνακα δρομολόγησης αφήνεται κενή. Η ομοιόμορφη κατανομή των αναγνωριστικών των κόμβων εγγυάται ότι θα υπάρχουν κατά μέσο όρο μόνο $\lceil log_{2^b} N \rceil$ γραμμές στον πίνακα δρομολόγησης.

Η επιλογή της παραμέτρου b εμπεριέχει έναν συνδυασμό ανάμεσα στο μέγεθος[10] του πίνακα δρομολόγησης (περίπου $\lceil log_{2^b} N \rceil * (2^b - 1)$) και του μέγιστου αριθμού απαιτούμενων αναπηδήσεων για τη δρομολόγηση μεταξύ οποιουδήποτε ζεύγους κόμβων ($\lceil log_{2^b} N \rceil$). Έτσι, με τιμή $b = 4$ και 10^6 κόμβους, ένας πίνακας δρομολόγησης περιέχει κατά μέσον όρο 75 εγγραφές και ο αναμενόμενος αριθμός αναπηδήσεων δρομολόγησης είναι 5, ενώ και με 10^9 κόμβους, ο πίνακας δρομολόγησης περιέχει κατά μέσο όρο μόλις 105 εγγραφές και ο αναμενόμενος αριθμός αναπηδήσεων δρομολόγησης είναι μόλις 7. Γι' αυτό το λόγο προαναφέρθηκε ότι η τυπική τιμή του b είναι 4, η οποία είναι μία "καλή" τιμή.

Στο Σχήμα 2.6, παρουσιάζεται μία υποθετική κατάσταση δρομολόγησης του κόμβου με αναγνωριστικό 10233102, $b = 2$ και $l = 8$. Όλοι οι αριθμοί είναι με βάση το 4. Ο πίνακας

[10] Όπου αναφερόμαστε στο μέγεθος ενός πίνακα δρομολόγησης ή διαφορετικά στο πλήθος των γραμμών του, λαμβάνονται υπ' όψη μόνο εκείνες οι γραμμές οι οποίες έχουν μία τουλάχιστον μη κενή εγγραφή. Επίσης, ένας εναλλακτικός τρόπος μέτρησης του μεγέθους είναι το πλήθος των μη κενών εγγραφών.

δρομολόγησης έχει $\lceil log_{2^b} N \rceil$ γραμμές και $2^b - 1$ στήλες. Η κορυφαία γραμμή του πίνακα δρομολόγησης είναι η μηδενική. Τα γραμμοσκιασμένα κελιά της γραμμής i περιέχουν την τιμή του i-οστού $+ 1$ ψηφίου του αναγνωριστικού του τρέχοντος κόμβου. Η πρώτη γραμμή περιέχει τους κόμβους που έχουν τα μηδέν πρώτα ψηφία του αναγνωριστικού τους ίδια με τα μηδέν πρώτα ψηφία του αναγνωριστικού του τρέχοντος κόμβου και το πρώτο ψηφίο διαφορετικό. Η δεύτερη γραμμή περιέχει τους κόμβους που έχουν το πρώτο ψηφίο του αναγνωριστικού τους ίδιο με το πρώτο ψηφίο του αναγνωριστικού του τρέχοντος κόμβου και το δεύτερο ψηφίο διαφορετικό. Η τρίτη γραμμή περιέχει τους κόμβους που έχουν τα δύο πρώτα ψηφία του αναγνωριστικού τους ίδια με τα δύο πρώτα ψηφία του αναγνωριστικού του τρέχοντος κόμβου και το τρίτο ψηφίο διαφορετικό κ.ο.κ. Τα αναγνωριστικά σε κάθε εγγραφή έχουν χωριστεί με μία παύλα για να δειχθεί το κοινό τους πρόθεμα με το 10233102 – επόμενο ψηφίο – υπόλοιπα ψηφία αναγνωριστικού. Οι διευθύνσεις IP των κόμβων έχουν παραλειφθεί. Στο ίδιο σχήμα επίσης διακρίνεται το σύνολο φύλλων και το σύνολο γειτόνων.

NodeId 10233102

Leaf set

SMALLER		LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Routing table

-0-2212102	1	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	2	10-3-23302
102-0-0230	102-1-1302	102-2-2302	3
1023-0-322	1023-1-000	1023-2-121	3
10233-0-01	1	10233-2-32	
0		102331-2-0	
		2	

Neighborhood set

13021022	10200230	11301233	31301233
02212102	22301203	31203203	33213321

Σχήμα 2.6: Η Κατάσταση Δρομολόγησης ενός υποθετικού κόμβου του Pastry με αναγνωριστικό κόμβου 10233102, $b = 2$, και $l = 8$.

Το σύνολο γειτόνων, M, περιέχει τα ζεύγη αναγνωριστικών-διευθύνσεων IP των $|M|$ πλησιέστερων (σύμφωνα με τη μετρική εγγύτητας) κόμβων στον τρέχον κόμβο. Το σύνολο γειτόνων δεν χρησιμοποιείται υπό κανονικές συνθήκες στη δρομολόγηση μηνυμάτων· είναι, όμως,

χρήσιμο στη διατήρηση ιδιοτήτων τοπικότητας. Η τυπική τιμή του $|M|$ είναι ίση με 2×2^b.

Το σύνολο φύλλων, L, είναι ένα σύνολο από κόμβους που οι $|L|/2$ από αυτούς έχουν μεγαλύτερα αναγνωριστικά από τον τρέχον κόμβο, αλλά που είναι τα κοντινότερα σε αυτόν ως προς την αριθμητική τιμή και οι υπόλοιποι $|L|/2$ έχουν μικρότερα αναγνωριστικά από τον τρέχον κόμβο, αλλά που είναι τα κοντινότερα σε αυτόν ως προς την αριθμητική τιμή. Το σύνολο φύλλων χρησιμοποιείται κατά τη διάρκεια της δρομολόγησης μηνυμάτων. Η τυπική τιμή του $|L|$ είναι ίση με 2^b. Στη συνέχεια και για να γίνει κατανοητός ο τρόπος με τον οποίο χρησιμεύουν οι δομές που ορίστηκαν παραπάνω δίνεται ο αλγόριθμος δρομολόγησης που ακολουθείται από το σύστημα Pastry. Η διαδικασία δρομολόγησης φαίνεται στο Σχήμα 2.7, υπό τη μορφή αλγορίθμου. Ο αλγόριθμος εκτελείται όποτε καταφθάνει σε έναν κόμβο με αναγνωριστικό A ένα μήνυμα με κλειδί D. Θα ξεκινήσουμε την περιγραφή ορίζοντας μερικούς συμβολισμούς.

R_l^i: η εγγραφή του πίνακα δρομολόγησης R στην στήλη $i, 0 \leq i < 2^b$ και γραμμή $l, 0 \leq l < \lfloor 128/b \rfloor$.

L_i: ο i-οστός αριθμητικά κοντινότερος κόμβος στο σύνολο φύλλων L, $-\lfloor |L|/2 \rfloor \leq i \leq \lfloor |L|/2 \rfloor$, όπου οι αρνητικοί / θετικοί δείκτες υποδεικνύουν κόμβους με μικρότερα / μεγαλύτερα αναγνωριστικά από τον τρέχον κόμβο αντίστοιχα.

D_l: η τιμή του l ψηφίου του κλειδιού D.

$shl(A, B)$: το μήκος του κοινού προθέματος μεταξύ των αναγνωριστικών / κλειδιών A και B, σε ψηφία.

Δεδομένου ενός μηνύματος, ο κόμβος ελέγχει αρχικά αν το κλειδί του βρίσκεται μεταξύ του διαστήματος που καλύπτουν τα αναγνωριστικά των κόμβων του συνόλου φύλλων (γραμμή 1). Σε θετική απάντηση, το μήνυμα προωθείται κατ' ευθείαν (χωρίς την αποστολή του σε ενδιάμεσους κόμβους) στον κόμβο προορισμού, δηλαδή σε εκείνο τον κόμβο του συνόλου φύλλων που το αναγνωριστικό είναι αριθμητικά κοντινότερο στο κλειδί (πιθανότατα και στον τρέχον κόμβο) (γραμμή 3).

Σε αρνητική απάντηση, δηλαδή, εάν το κλειδί δεν εμπίπτει στο διάστημα αναγνωριστικών που καλύπτεται από το σύνολο φύλλων, χρησιμοποιείται ο πίνακας δρομολόγησης και το μήνυμα προωθείται στον κόμβο του οποίου το αναγνωριστικό μοιράζεται με το κλειδί ένα πρόθεμα μήκους τουλάχιστον ενός ψηφίου μεγαλύτερου από αυτό που μοιράζεται το αναγνωριστικό του τρέχοντος κόμβου με το κλειδί (γραμμές 6–8). Σε συγκεκριμένες περιπτώσεις,

```
(1)  if (L_{-⌊|L|/2⌋} ≤ D ≤ L_{⌊|L|/2⌋}) {
(2)      // D is within range of our leaf set
(3)      forward to L_i, s.th. |D − L_i| is minimal;
(4)  } else {
(5)      // use the routing table
(6)      Let l = shl(D, A);
(7)      if (R_l^{D_l} ≠ null) {
(8)          forward to R_l^{D_l};
(9)      }
(10)     else {
(11)         // rare case
(12)         forward to T ∈ L ∪ R ∪ M, s.th.
(13)             shl(T, D) ≥ l,
(14)             |T − D| < |A − D|
(15)     }
(16) }
```

Σχήμα 2.7: Ο βασικός αλγόριθμος δρομολόγησης που χρησιμοποιεί το σύστημα Pastry, υπό τη μορφή ψευδοκώδικα.

είναι πιθανό η αντίστοιχη εγγραφή στον πίνακα δρομολόγησης να είναι κενή ή ο αντίστοιχος κόμβος να μην είναι προσιτός (γραμμές 11–14), όπου το μήνυμα προωθείται στον κόμβο που το αναγνωριστικό του μοιράζεται με το κλειδί ένα πρόθεμα μήκους όσου και αυτού του προθέματος μεταξύ του κλειδιού και του τρέχοντος κόμβου και που είναι αριθμητικά κοντινότερο στο κλειδί από ότι είναι ο τρέχον κόμβος. Ένας τέτοιος κόμβος πρέπει να περιέχεται στο σύνολο φύλλων (αυτό το εξασφαλίζει το κοινό πρόθεμα) υπό την προϋπόθεση ότι δεν έχουν αποτύχει ταυτόχρονα οι $\lfloor |L|/2 \rfloor$ κόμβοι του συνόλου αυτού.

Αυτή η απλή διαδικασία δρομολόγησης πάντα συγκλίνει, επειδή σε κάθε βήμα το μήνυμα αποστέλλεται σε έναν κόμβο, ο οποίος είτε

1. μοιράζεται ένα μεγαλύτερου μήκους πρόθεμα με το κλειδί από ότι ο τρέχον κόμβος, είτε
2. μοιράζεται το ίδιου μήκους πρόθεμα, αλλά το αναγνωριστικό του είναι αριθμητικά κοντινότερο στο κλειδί από ότι αυτό του τρέχοντος κόμβου.

Μπορεί να αποδειχθεί ότι ο αναμενόμενος αριθμός βημάτων δρομολόγησης είναι ίσος

με $\lceil log_{2^b} N \rceil$, λαμβάνοντας ως δεδομένο ότι οι πίνακες δρομολόγησης είναι ακριβείς και ότι δεν έλαβε χώρα καμία αποτυχία κόμβου κατά την αποστολή του μηνύματος. Εν συντομία, αυτό μπορεί να δειχθεί από τις τρεις περιπτώσεις του παραπάνω αλγορίθμου. Εάν ένα μήνυμα προωθηθεί με την πρώτη περίπτωση (γραμμές 6–8), τότε το σύνολο των κόμβων, των οποίων τα αναγνωριστικά έχουν ένα κοινό και μεγαλύτερου μήκους πρόθεμα με το κλειδί του μηνύματος από ότι αυτού με το αναγνωριστικό του τρέχον κόμβου, μειώνεται σε κάθε βήμα κατά ένα παράγοντα ίσο με 2^b, το οποίο σημαίνει ότι το μήνυμα θα φθάσει στον προορισμό του σε $\lceil log_{2^b} N \rceil$ βήματα. Εάν το κλειδί είναι μεταξύ του διαστήματος του συνόλου φύλλων (γραμμές 2–3), τότε το μήνυμα θα φθάσει στον προορισμό του σε ένα βήμα με μία αναπήδηση.

Η τρίτη περίπτωση που δεν αναφέρθηκε προηγουμένως προκύπτει ότι το κλειδί δεν είναι μεταξύ του διαστήματος του συνόλου φύλλων και δεν υπάρχει κατάλληλα εγγραφή στον πίνακα δρομολόγησης (γραμμές 11–14). Η πιθανότητα αυτής της περίπτωσης, δεδομένου της ομοιόμορφης κατανομής των αναγνωριστικών εξαρτάται από το $|L|$. Αναλύσεις έχουν δείξει ότι με $|L| = 2^b$ ή $|L| = 2 \times 2^b$, η πιθανότητα εμφάνισης αυτής της περίπτωσης κατά τη διάρκεια αποστολής ενός μηνύματος είναι μικρότερη από 0.02 και 0.006 αντίστοιχα. Όταν αυτό συμβεί ο αριθμός των βημάτων με μεγάλη πιθανότητα είναι ίσος με $\lceil log_{2^b} N \rceil + 1$.

Σε περίπτωση πολλών ταυτόχρονων αποτυχιών κόμβων, ο αριθμός των απαιτούμενων βημάτων δρομολόγησης μπορεί να φτάσει στη χειρότερη περίπτωση να είναι γραμμικός ως προς το N, ενόσω οι κόμβοι ενημερώνουν την κατάστασή τους, δηλαδή τον πίνακα δρομολόγησης και τα σύνολα γειτόνων και φύλλων.

2.2.4.4 Διαχείριση Δυναμικότητας Δικτύου

Ένα σημαντικό θέμα στο σχεδιασμό του Pastry είναι ο τρόπος με τον οποίο οι κόμβοι του μπορούν να διατηρούν την κατάσταση δρομολόγησής τους αποδοτικά και δυναμικά κάτω από συνθήκες αποτυχίας κόμβων, επανάκαμψης κόμβων και νέων αφίξεων.

Κατά την άφιξη ενός νέου κόμβου του ανατίθεται ένα αναγνωριστικό κόμβου, έστω X. Στη συνέχεια επικοινωνώντας με έναν γειτονικό του κόμβο A (σύμφωνα με την μετρική εγγύτητας) του αναθέτει την αποστολή ενός μηνύματος χρησιμοποιώντας ως κλειδί το X. Αυτό το μήνυμα δρομολογείται και στο τέλος παραλαμβάνεται από έναν κόμβο Z με αναγνωριστικό το πιο κοντινό στο X[11] όσον αφορά στην αριθμητική του τιμή. Ο X στη συνέχεια λαμβάνει το

[11] Στην ελάχιστη πιθανότητα του γεγονότος κατά το οποίο το X είναι ίσο με το Z, ο νέος κόμβος θα πρέπει να λάβει ένα νέο αναγνωριστικό κόμβου.

σύνολο φύλλων από τον Z και την i-οστή γραμμή του πίνακα δρομολόγησης από τον i-οστό κόμβο που ενεπλάκη στη δρομολόγηση του μηνύματος από το A στον Z. Στο [4] μπορεί να δειχθεί ότι χρησιμοποιώντας αυτές τις πληροφορίες, ο X μπορεί να αρχικοποιήσει σωστά την κατάσταση δρομολόγησής του και να ειδοποιήσει τους κόμβους που χρειάζεται να γνωρίζουν την άφιξή του.

Για τον χειρισμό των αποτυχιών των κόμβων, οι γειτονικοί κόμβοι ως προς το χώρο των αναγνωριστικών (οι οποίοι είναι ενήμεροι για την μεταξύ τους ύπαρξη από το γεγονός του ότι κάθε ένας βρίσκεται στο σύνολο φύλλων του άλλου) ανταλλάσσουν περιοδικά μηνύματα κατάστασης *keep-alive*. Εάν κάποιος κόμβος δεν ανταποκρίνεται για κάποια δεδομένη διάρκεια T, θεωρείται ότι έχει αποτύχει. Όλοι οι κόμβοι που βρίσκονται στο σύνολο φύλλων του κόμβου που απέτυχε ειδοποιούνται και ενημερώνουν κατάλληλα τα σύνολα φύλλων τους. Δεδομένου ότι τα σύνολα φύλλων γειτονικών κόμβων επικαλύπτονται, αυτή η ενημέρωση είναι στοιχειώδης. Η διαδικασία αυτή αναλύεται λεπτομερέστερα στο [4].

Τέλος, όσον αφορά στις επανακάμψεις κόμβων, κατά τη διάρκεια επανάκαμψης ενός τέτοιου γίνεται επικοινωνία με του κόμβους που υπήρχαν στην τελευταία γνωστή κατάσταση στο σύνολο φύλλων του. Από αυτούς λαμβάνονται όλοι οι κόμβοι των συνόλων φύλλων τους, για το λόγο του ότι είναι πολύ πιθανό να είναι γείτονες του ίδιου του κόμβου, βάσει των οποίων ενημερώνει το σύνολο φύλλων του και τέλος ειδοποιεί για την παρουσία του όλα τα μέλη του συνόλου αυτού. Και αυτή η διαδικασία αναλύεται λεπτομερέστερα στο [4].

Η διαδικασία ενημέρωσης των πινάκων δρομολόγησης των οποίων κάποιες εγγραφές αναφέρονται σε αποτυχίες ή επανακάμψεις κόμβων περιγράφονται με λεπτομέρεια στο [4, 18].

2.2.4.5 PAST

Το σύστημα PAST [5] αποτελείται από κόμβους που είναι συνδεδεμένοι στο Διαδίκτυο, όπου κάθε τέτοιος έχει τη δυνατότητα της αρχικοποίησης και της δρομολόγησης αιτήσεων για την εισαγωγή ή ανάκτηση αρχείων. Προαιρετικά, αυτοί οι κόμβοι μπορούν να συνεισφέρουν στην αποθήκευση του συστήματος στο οποίο φιλοξενούνται. Οι κόμβοι του συστήματος PAST διαμορφώνουν ένα αυτο-οργανώμενο επικαλυπτόμενο δίκτυο. Κατά την εισαγωγή ενός αρχείου κατασκευάζεται ένα πανομοιότυπο, το οποίο στη συνέχεια αποστέλλεται προς εισαγωγή σε συγκεκριμένους κόμβους του δικτύου για την εξασφάλιση της διαθεσιμότητάς του. Με μεγάλη πιθανότητα, οι κόμβοι που κατέχουν αντίγραφο του ίδιου αρχείου διαφέρουν ως προς την γεωγραφική τους θέση, ιδιοκτησία, διαχείριση κ.λπ. Το σύστημα PAST είναι μία εφαρμο-

γή υλοποιημένη πάνω από το επίπεδο που ορίζει το σύστημα Pastry. Το PAST χρησιμοποιεί το Pastry ως ένα αποδοτικό σχήμα δρομολόγησης που του εξασφαλίζει την αξιόπιστη επικοινωνία μεταξύ των κόμβων του είτε αυτή πραγματοποιείται σε επίπεδο εισαγωγής-ανάκτησης αρχείων, είτε σε αποστολή-λήψη μηνυμάτων.

Το σύστημα PAST προσφέρει σε κάθε κόμβο δύο βασικές λειτουργίες, την εισαγωγή και την αναζήτηση, οι οποίες υλοποιούνται με τις παρακάτω διαδικασίες:

fileId = Insert(name, owner-credentials, k, file): αποθηκεύει ένα αρχείο σε ένα καθορισμένο από το χρήστη αριθμό k κόμβων του δικτύου PAST. Η λειτουργία παράγει ένα *αναγνωριστικό* (fileId) μήκους 160 bit το οποίο μπορεί να χρησιμοποιηθεί στη συνέχεια για την ταυτοποίηση του αρχείου. Το αναγνωριστικό αυτό μπορεί να υπολογισθεί από τον κατακερματισμό του ονόματος του αρχείου, το δημόσιο κλειδί του ιδιοκτήτη και κάποιες άλλες τυχαίες παραμέτρους. Αυτή η επιλογή εξασφαλίζει με μεγάλη πιθανότητα την παραγωγή μοναδικών αναγνωριστικών. Στη σπάνια περίπτωση του εντοπισμού μίας σύγκρουσης μεταξύ δύο αναγνωριστικών η εισαγωγή του νεότερου απορρίπτεται.

file = Lookup(fileId): ανακτά αξιόπιστα ένα αντίγραφο του αρχείου που το αναγνωριστικό του ταυτοποιείται με το fileId εάν υπάρχει και εάν κάποιος από τους k κόμβους, που έχουν αποθηκεύσει το αρχείο, είναι προσβάσιμος μέσω του Διαδικτύου. Υπό κανονικές συνθήκες, το αρχείο ανακτάται από τον πλησιέστερο κόμβο του δικτύου PAST (σύμφωνα με την μετρική εγγύτητας) στον κόμβο που πραγματοποίησε την αναζήτηση και που κατέχει ένα αντίγραφο του αιτούμενου αρχείου.

Σε αυτό το σημείο τελειώνει η παρουσίαση μας για τα δίκτυα ομότιμων κόμβων. Το επόμενο κεφάλαιο δίνει μία εισαγωγή στην περιοχή της Ανάκτησης Πληροφορίας.

2.3 Ανάκτηση Πληροφορίας (Information Retrieval)

Η Ανάκτηση Πληροφορίας (ΑΠ) [29] πραγματεύεται την αναπαράσταση, αποθήκευση, οργάνωση και πρόσβαση των αντικειμένων πληροφορίας. Ως επί το πλείστον στην ΑΠ τα αντικείμενα πληροφορίας είναι τα έγγραφα. Η αναπαράσταση και οργάνωση των αντικειμένων πληροφορίας θα πρέπει να παρέχουν στο χρήστη εύκολη πρόσβαση σε πληροφορίες για τις οποίες αυτός ενδιαφέρεται. Δυστυχώς, η απόδοση *των πληροφοριών που χρειάζεται ο χρήστης*

(user information need) δεν είναι απλό πρόβλημα. Έστω για παράδειγμα, η παρακάτω υποθετική πληροφορία που χρειάζεται ο χρήστης στα πλαίσια του Παγκόσμιου Ιστού:

> Βρες όλες τις σελίδες (έγγραφα) που περιέχουν πληροφορίες για ομάδες αντισφαίρισης κολεγίων που (1) συντηρούνται από ένα πανεπιστήμιο των Ηνωμένων Πολιτειών και (2) συμμετέχουν στο τουρνουά αντισφαίρισης NCAA. Η κάθε σελίδα θα πρέπει να περιλαμβάνει πληροφορίες για την εθνική κατάταξη της ομάδας τα τρία τελευταία χρόνια και τη διεύθυνση ηλεκτρονικού ταχυδρομείου ή τον αριθμό τηλεφώνου του προπονητή της ομάδας.

Είναι σαφές, ότι αυτή η εκτενής περιγραφή των πληροφοριών που χρειάζεται ο χρήστης δεν μπορεί να αιτηθεί άμεσα χρησιμοποιώντας την τρέχουσα διεπαφή που προσφέρουν οι μηχανές αναζήτησης του Ιστού. Έτσι, ο χρήστης θα πρέπει πρώτα να μεταφράσει τις πληροφορίες που χρειάζεται σε μία *επερώτηση* (query), η οποία να μπορεί να επεξεργασθεί από μία μηχανή αναζήτησης (ή σύστημα ΑΠ). Στην πιο απλή μορφή της, αυτή η μετάφραση παράγει ένα σύνολο *λέξεων-κλειδιών* (keywords) ή *όρους ευρετηρίου* (index terms) που αποτελούν μία περιληπτική περιγραφή της πληροφορίας που χρειάζεται ο χρήστης. Δοθείσης μίας επερώτησης ενός χρήστη, ο αντικειμενικός σκοπός ενός συστήματος ΑΠ είναι να ανακτήσει τις πληροφορίες που *δύναται* να είναι χρήσιμες ή σχετικές στον χρήστη.

2.3.1 Ανάκτηση Δεδομένων και Ανάκτηση Πληροφορίας

Πολλές φορές η έννοια της ΑΠ συγχέεται με αυτή της Ανάκτησης Δεδομένων (ΑΔ, Data Retrieval). Η ΑΔ, στα πλαίσια ενός συστήματος ΑΠ, έχει να κάνει κατά βάση με τον προσδιορισμό των εγγράφων μίας συλλογής που περιέχουν τις λέξεις-κλειδιά που περιέχονται στην επερώτηση του χρήστη, κάτι το οποίο τις περισσότερες φορές δεν είναι επαρκές για να καλύψει τις πληροφορίες που χρειάζεται ο χρήστης. Στη πράξη, ένας χρήστης ενός συστήματος ΑΠ ενδιαφέρεται περισσότερο για την ανάκτηση πληροφοριών σχετικά με ένα θέμα παρά για την ανάκτηση δεδομένων που ικανοποιούν τη δεδομένη επερώτηση. Μία γλώσσα ανάκτησης δεδομένων έχει ως σκοπό την ανάκτηση όλων των αντικειμένων που ικανοποιούν κάποιες συνθήκες, όπως αυτές των κανονικών εκφράσεων ή των σχεσιακών αλγεβρικών εκφράσεων. Έτσι, για ένα σύστημα ΑΔ, ένα και μόνο εσφαλμένο (που δεν ικανοποιεί, δηλαδή, τις συνθήκες της επερώτησης) αντικείμενο ανάμεσα σε χιλιάδες άλλα ανακτηθέντα σημαίνει ολική αποτυχία. Από την άλλη, για ένα σύστημα ΑΠ, τα ανακτηθέντα αντικείμενα μπορεί να είναι

ανακριβή και μικρά σφάλματα είναι πιθανόν να μην γίνουν αντιληπτά. Η κύρια αιτία για αυτή τη διαφορά των δύο τύπων συστημάτων είναι ότι αυτό του δεύτερου τύπου συχνά έχει να κάνει με κείμενα γραμμένα σε φυσική γλώσσα, τα οποία μπορεί να μην είναι πάντοτε καλά δομημένα και μπορεί να είναι διφορούμενα ως προς τη σημασιολογία. Από την άλλη πλευρά, ένα σύστημα ΑΔ (όπως μία σχεσιακή βάση δεδομένων) έχει να κάνει με δεδομένα που έχουν μία καλά ορισμένη δομή και σημασιολογία.

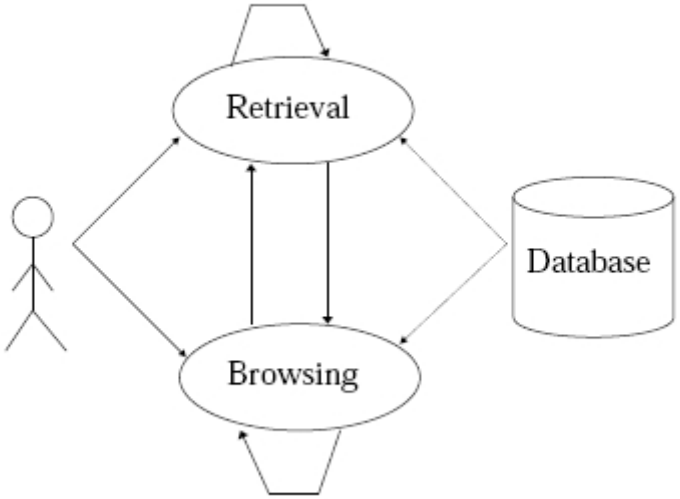
Η ΑΔ, ενώ παρέχει μία λύση στον χρήστη ενός συστήματος βάσης δεδομένων, δεν λύνει το πρόβλημα της ανάκτησης πληροφορίας σχετικά με κάποιο θέμα ή ζήτημα. Για να είναι αποτελεσματική η προσπάθεια ενός συστήματος ΑΠ να ικανοποιήσει το αίτημα του χρήστη για πληροφορίες, θα πρέπει αυτό να ερμηνεύσει με κάποιο τρόπο τα περιεχόμενα των αντικειμένων πληροφορίας (των εγγράφων) μίας συλλογής και να τα κατατάξει σύμφωνα με ένα βαθμό συνάφειας ως προς την επερώτηση του χρήστη. Αυτή η ερμηνεία των περιεχομένων των εγγράφων εμπεριέχει την απόσπαση συντακτικών και σημασιολογικών πληροφοριών από το κείμενο και σε δεύτερο στάδιο το ταίριασμά τους με τις πληροφορίες που χρειάζεται ο χρήστης. Η δυσκολία δεν εμπεριέχεται μόνο στον τρόπο εξαγωγής αυτών των πληροφοριών, αλλά και στον τρόπο χρήσης τους έτσι ώστε να αποφασιστεί η συνάφειά τους ως προς την επερώτηση του χρήστη. Έτσι, η έννοια της συνάφειας βρίσκεται στο επίκεντρο της ΑΠ. Στην πράξη, ο πρωτεύον σκοπός ενός συστήματος ΑΠ είναι να ανακτήσει όλα τα έγγραφα που είναι συναφή ως προς την επερώτηση του χρήστη, ενώ ταυτόχρονα να ανακτήσει όσο το δυνατόν λιγότερα άσχετα έγγραφα.

2.3.2 Βασικές Έννοιες

Η αποτελεσματική ανάκτηση συναφούς πληροφορίας επηρεάζεται άμεσα τόσο από την *εργασία του χρήστη* (user task) όσο και από τη *λογική άποψη των εγγράφων* (logical view of the documents) που υιοθετούν τα συστήματα ανάκτησης. Το Σχήμα 2.8 απεικονίζει την αλληλεπίδραση του χρήστη με το σύστημα ανάκτησης.

Με τον όρο εργασία του χρήστη εννοείται η προσπάθεια που καταλαμβάνει ο χρήστης έτσι ώστε να μεταφράσει τις πληροφορίες που χρειάζεται να βρει σε μία επερώτηση χρησιμοποιώντας ένα *μοντέλο αναπαράστασης πληροφορίας*[12] που παρέχεται από το σύστημα ΑΠ. Με ένα τέτοιο σύστημα ΑΠ, αυτό συνήθως απαιτεί τον προσδιορισμό ενός συνόλου από λέξεις σε

[12] Ένα τέτοιο μοντέλο είναι και αυτό που χρησιμοποιείται στην αρχιτεκτονική του συστήματος LibraRing, το AWPS, το οποίο θα περιγραφεί στην επόμενη ενότητα.



Σχήμα 2.8: Αλληλεπίδραση του χρήστη με το σύστημα ανάκτησης διά μέσου ξεχωριστών εργασιών.

συνδυασμό με την εφαρμογή τελεστών πάνω σε αυτές, που αποδίδουν τη σημασιολογία της πληροφορίας που χρειάζεται. Ανάλογα με τον τύπο των τελεστών λέμε ότι χρησιμοποιούμε και ένα διαφορετικό μοντέλο ΑΠ (π.χ. Boolean, VSM, Probabilistic). Με ένα σύστημα ΑΔ, χρησιμοποιείται μία έκφραση επερώτησης (όπως, για παράδειγμα, μία κανονική έκφραση) για να αποδώσει τους περιορισμούς που πρέπει να ικανοποιούν τα αντικείμενα του συνόλου της απάντησης. Και στις δύο περιπτώσεις, λέμε ότι ο χρήστης αναζητά χρήσιμες πληροφορίες εκτελώντας μία *εργασία ανάκτησης* (retrieval task).

Τα έγγραφα μίας συλλογής συχνά αναπαριστώνται μέσω ενός συνόλου όρων ευρετηρίου ή λέξεων-κλειδιών. Τέτοιες λέξεις-κλειδιά μπορούν να εξαχθούν από το κείμενο των εγγράφων ή μπορούν να προσδιοριστούν από ένα ανθρώπινο υποκείμενο (όπως γίνεται συνήθως στις επιστήμες πληροφορίας). Ανεξάρτητα του αν οι λέξεις κλειδιά έχουν προέλθει αυτόματα ή έχουν παραχθεί από κάποιον ειδικό, παρέχουν μία *λογική άποψη του εγγράφου*.

2.3.3 Μοντέλα Ανάκτησης Πληροφορίας

Τα μοντέλα ΑΠ [23] εστιάζουν στη διαδικασία της σύγκρισης των λέξεων-κλειδιών που προσδιορίζει ο χρήστης με τα έγγραφα της συλλογής για την παραγωγή των αποτελεσμάτων που ταιριάζουν με τις επιθυμίες του χρήστη. Τα τρία βασικότερα μοντέλα ΑΠ είναι το Boo-

lean, το Διανυσματικού Χώρου (Vector Space) και το Πιθανοτικό (Probabilistic). Το πρώτο βασίζεται στην αρχή του επακριβούς ταιριάσματος, ενώ τα άλλα δύο στην έννοια του καλύτερου ταιριάσματος.

2.3.3.1 Boolean

Η ανάκτηση πληροφορίας χρησιμοποιώντας το Boolean μοντέλο βασίζεται στο επακριβές ταίρισμα μίας επερώτησης με μία συγκεκριμένη πληροφορία ενός εγγράφου (π.χ. όνομα συγγραφέα, τίτλος). Ο όρος "Boolean" χρησιμοποιείται λόγω του ότι μία επερώτηση εκφράζεται με λέξεις ή φράσεις, συνδυαζόμενες με τους τελεστές της λογικής Boolean. Σύμφωνα με αυτό το μοντέλο ανάκτησης, όλα τα έγγραφα που περιέχουν έναν συνδυασμό από λέξεις ή φράσεις της εκφρασμένης επερώτησης ανακτώνται, και δεν υπάρχει καμία διάκριση μεταξύ των ανακτηθέντων εγγράφων. Έτσι, το αποτέλεσμα αυτής της σύγκρισης συνίσταται στο διαχωρισμό της συλλογής εγγράφων σε δύο σύνολα· στο σύνολο των ανακτηθέντων εγγράφων και στο σύνολο των μη ανακτηθέντων.

Ένα μεγάλο πρόβλημα με αυτό το μοντέλο ανάκτησης είναι ότι δεν επιτρέπει κανενός είδους κατάταξης των στοιχείων του συνόλου των ανακτηθέντων εγγράφων. Παρ' όλα αυτά η παρουσίαση των εγγράφων στο χρήστη σύμφωνα με ένα ποσοστό σχετικότητας καταλήγει σε περισσότερο αποδοτικά και εύχρηστα συστήματα. Όμοια, αποκλείοντας έγγραφα που δεν ταιριάζουν επακριβώς με μία επερώτηση καταλήγουμε σε υποβαθμισμένη αποδοτικότητα.

2.3.3.2 Vector Space

Το πιο διαδεδομένο μοντέλο ΑΠ καλύτερου ταιριάσματος είναι το Μοντέλο Διανυσματικού Χώρου (Vector Space Model). Αυτό το μοντέλο θεωρεί τόσο το κείμενο όσο και τις επερωτήσεις ως διανύσματα ενός πολυδιάστατου χώρου, οι διαστάσεις των οποίων είναι οι λέξεις που χρησιμοποιούντα για την αναπαράσταση του κειμένου. Οι επερωτήσεις και τα κείμενα συγκρίνονται μέσω της σύγκρισης των αντίστοιχων διανυσμάτων, χρησιμοποιώντας για παράδειγμα τη μετρική του συνημίτονου της γωνίας που αυτά σχηματίζουν. Κάτω από αυτό το πρίσμα σύγκρισης, η υπόθεση είναι ότι όσο πιο όμοιο είναι ένα διάνυσμα που αναπαριστά ένα κείμενο, με ένα διάνυσμα που αναπαριστά μία επερώτηση, τόσο πιο πιθανό είναι το κείμενο να σχετίζεται με την επερώτηση. Μία παραλλαγή αυτού του μοντέλου είναι η χρήση βαρών στους όρους (ή διαστάσεις) των επερωτήσεων, ή της αναπαράστασης του κειμένου, έτσι ώστε να ληφθεί υπ ' όψη ο βαθμός σημαντικότητάς τους.

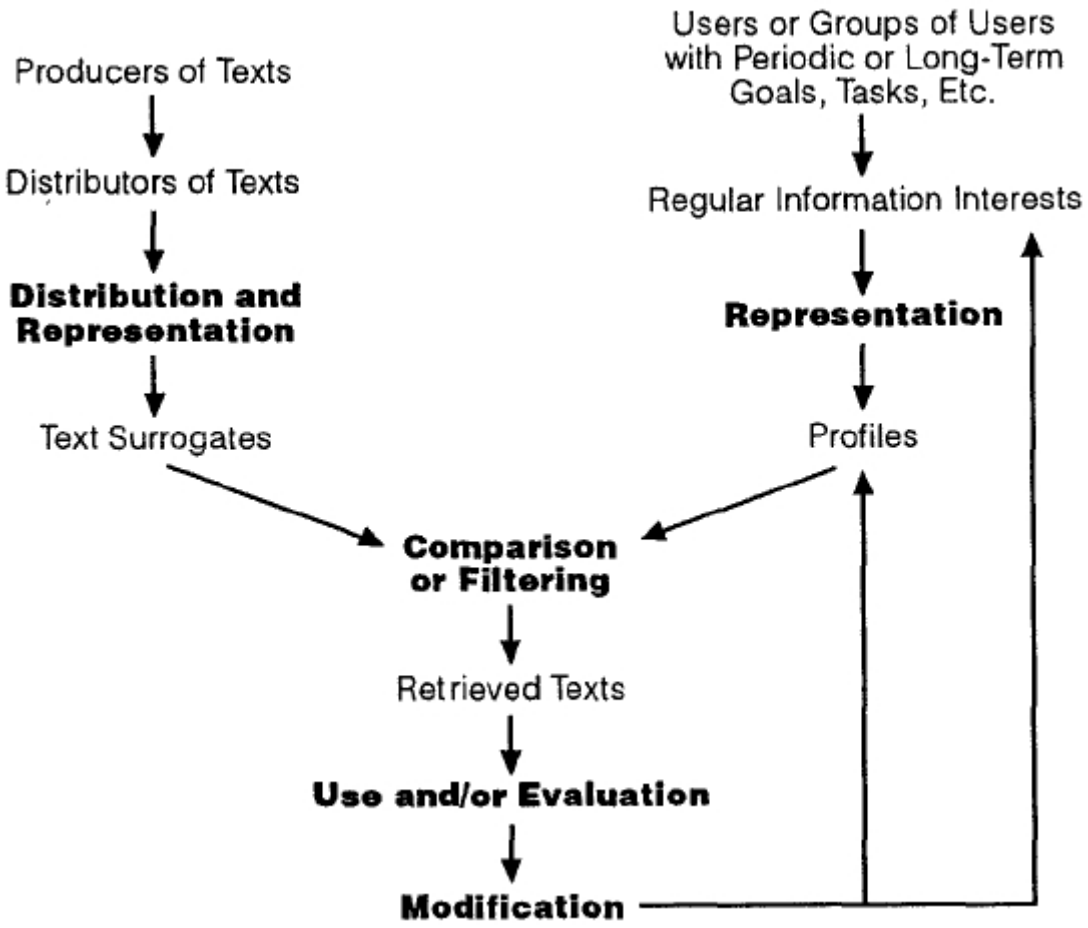
2.3.3.3 Probabilistic

Τα Πιθανοτικά μοντέλα ΑΠ βασίζονται στην Αρχή της Πιθανοτικής Κατάταξης (Probabilistic Ranking Principle) [32]. Σύμφωνα με αυτή ο στόχος ενός συστήματος ανάκτησης είναι η κατάταξη των εγγράφων μιας συλλογής σύμφωνα με την πιθανότητα ομοιότητας τους ως προς την επερώτηση, δεδομένων όλων των διαθέσιμων στοιχείων / ενδείξεων. Αυτή η αρχή λαμβάνει υπ' όψη της ότι τόσο η αναπαράσταση του κειμένου, όσο και της επερώτησης είναι αβέβαιη και ότι η σχέση ομοιότητας μεταξύ αυτών είναι επίσης αβέβαιη. Το πιθανοτικό μοντέλο θεωρεί ότι υπάρχουν ποικίλες πηγές ενδείξεων που θα μπορούσαν να χρησιμοποιηθούν για τον υπολογισμό της πιθανότητας της ομοιότητας μεταξύ ενός κειμένου και μιας επερώτησης. Η πιο συνηθισμένη πηγή είναι η στατιστική κατανομή των όρων των εγγράφων μιας συλλογής.

2.3.4 Διήθηση/Φιλτράρισμα Πληροφορίας (Information Filtering)

Η Διήθηση Πληροφορίας (Information Filtering) ή ΔΠ [23] είναι ένα όνομα, το οποίο χρησιμοποιείται για να περιγράψει μία ποικιλία διεργασιών σχετικά με την παράδοση πληροφοριών σε ανθρώπους που τη χρειάζονται. Τα συστήματα ΔΠ είναι σχεδιασμένα ώστε να εργάζονται με αδόμητα ή ημι-αδόμητα δεδομένα. Για να καταλάβουμε αυτούς τους δύο όρους θα τους αντιπαραθέσουμε με τα πλήρως δομημένα δεδομένα μίας βάσης δεδομένων, όπου αυτά είναι οργανωμένα ανά εγγραφές και για κάθε πεδίο εγγραφής καθορίζεται ένα πεδίο ορισμού ή ακόμα και ένα πεδίο τιμών. Εκτός όμως από τις δυνατές και έγγυρες τιμές τους, μία βάση δεδομένων ορίζει και τη σημασιολογία αυτών των δεδομένων. Κάτω από αυτό το πρίσμα (των τύπων δεδομένων), τα συστήματα ΔΠ ασχολούνται με δεδομένα που αναπαριστούν κείμενο, εικόνα, ήχο ή βίντεο και που δεν μπορούν να τα διαχειριστούν αποτελεσματικά οι βάσεις δεδομένων. Έτσι, τυπικές εφαρμογές ΔΠ αντλούν τα δεδομένα προς επεξεργασία από μία συνεχόμενη ροή δεδομένων, που είτε προέρχεται από κάποια απομακρυσμένη πηγή που μεταδίδει καθολικά πληροφορίες (π.χ. μία υπηρεσία μετάδοσης νέων), είτε από μία πηγή που αποστέλλει άμεσα πληροφορίες (π.χ. e-mail).

Παρ' όλα αυτά το σημαντικότερο χαρακτηριστικό των συστημάτων ΔΠ, που τα κάνει να ξεχωρίζουν από τα συστήματα ΑΠ, είναι το γεγονός ότι βασίζονται σε περιγραφές πληροφοριών για την ύπαρξη των οποίων θα ήθελαν ανεξάρτητα άτομα ή ομάδες να ενημερώνονται. Αυτός ο καθορισμός των πληροφοριών γίνεται ανά χρήστη ή ομάδα χρηστών και ονομάζεται *προφίλ*. Τα προφίλ χρηστών συνήθως αναπαριστούν μακροπρόθεσμα ενδιαφέροντα.



Σχήμα 2.9: Ένα γενικό μοντέλο Διήθησης Πληροφορίας.

Ένα άλλο χαρακτηριστικό της ΔΠ είναι ότι χρησιμοποιείται συχνά για την αφαίρεση δεδομένων από μία εισερχόμενη ροή παρά για την εύρεση τους σε αυτή. Στην πρώτη περίπτωση, οι χρήστες βλέπουν τί απέμεινε στη ροή δεδομένων μετά την αφαίρεση, ενώ στη δεύτερη περίπτωση, βλέπουν τα δεδομένα που τελικά εξάχθηκαν. Ένα τυπικό παράδειγμα της πρώτης περίπτωσης είναι ένα φίλτρο για τα μηνύματα ηλεκτρονικού ταχυδρομείου, που είναι σχεδιασμένο ώστε να αφαιρεί τα μηνύματα που θεωρούνται ως ανεπιθύμητη αλληλογραφία (spam). Σύμφωνα με αυτά, ένα προφίλ μπορεί να εκφράζει τόσο το τί είδους πληροφορία θα ήθελε ένας χρήστης, όσο και το τί πληροφορία δεν θα ήθελε να παραλαμβάνει. Το Σχήμα 2.9 απεικονίζει ένα γενικό μοντέλο Διήθησης Πληροφορίας ενσωματώνοντας όλα τα χαρακτηριστικά στα οποία αναφερθήκαμε.

Σύμφωνα με αυτά θα μπορούσαμε να απαριθμήσουμε τις κύριες διαφορές των συστημάτων ΔΠ και ΑΠ:

- Σε ένα σύστημα ΑΠ η μορφή αλληλεπίδρασής του με το χρήστη είναι στιγμιαία, δηλαδή

ένας χρήστης, έχοντας ένα παροδικό σκοπό, θέτει στο σύστημα μία επερώτηση και μόλις αυτή απαντηθεί η αλληλεπίδρασή τους σταματάει (π.χ. ένας χρήστης του διαδικτύου — χρήστης του συστήματος ΑΠ — που ψάχνει για μία φράση χρησιμοποιώντας μία μηχανή αναζήτησης — το σύστημα ΑΠ). Σε ένα σύστημα ΔΠ η αλληλεπίδρασή του με το χρήστη (που μπορεί να είναι και ομάδα χρηστών) είναι πολλαπλή, δηλαδή ο χρήστης δημιουργεί ένα προφίλ με τις πληροφορίες που τον ενδιαφέρουν και το σύστημα τον ειδοποιεί κάθε φορά που λαμβάνει μία πληροφορία που ικανοποιεί την περιγραφή του προφίλ του. Θα λέγαμε ότι η ΔΠ είναι το αντίστροφο της ΑΠ όσον αφορά στη σχέση συστήματος-χρήστη.

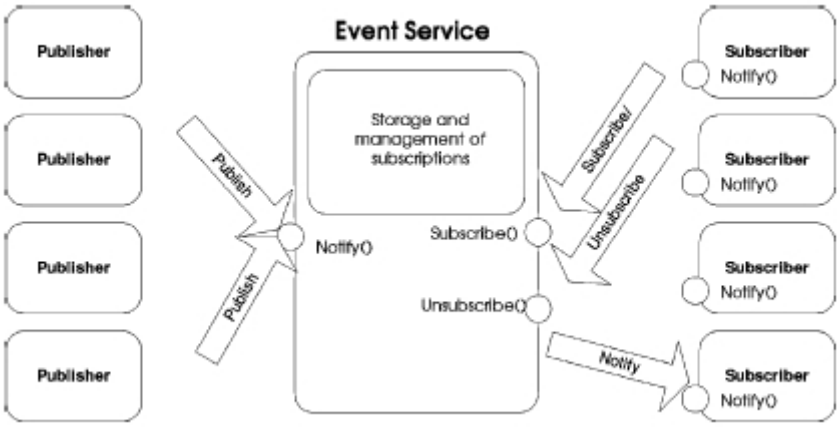
- Τα συστήματα ΑΠ ασχολούνται κυρίως με έγγραφα κειμένου που είναι οργανωμένα σε μία συλλογή. Από την άλλη, τα συστήματα ΔΠ δεν ενδιαφέρονται για τη δομή και οργάνωση των δεδομένων πληροφορίας, αλλά για την διανομή τους στους χρήστες τους.

Έχοντας περιγράψει τα διάφορα μοντέλα ανάκτησης πληροφορίας και τις παραλλαγές της (ανάκτηση δεδομένων, διήθηση πληροφορίας) θα περάσουμε στην περιγραφή της Δημοσίευσης / Συνδρομής. Αυτή η μορφή αλληλεπίδρασης συστήματος-χρήστη είναι όμοια με αυτή της διήθησης πληροφορίας. Η κύρια διαφορά τους έγγειται στο ότι η πρώτη *ειδοποιεί* τους χρήστες για *γεγονότα* που δύναται να τους ενδιαφέρουν, ενώ η δεύτερη *προωθεί* στους χρήστες της *πληροφορίες* που δύναται να τους ενδιαφέρουν.

2.4 Δημοσίευση / Συνδρομή (Publish / Subscribe)

Η Δημοσίευση / Συνδρομή αποτελεί ένα σχήμα αλληλεπίδρασης μεταξύ δύο οντοτήτων, τους *εκδότες* (publishers), που δημοσιοποιούν ένα αντικείμενο, και τους *συνδρομητές* (subscribers), που γίνονται συνδρομητές σε δημοσιοποιημένα αντικείμενα. Πιο συγκεκριμένα, το σχήμα της Δημοσίευσης / Συνδρομής παρέχει στους συνδρομητές την δυνατότητα να εκφράσουν το ενδιαφέρον τους για ένα γεγονός ή ένα πρότυπο γεγονότων έτσι ώστε να ειδοποιηθούν στη συνέχεια για κάποιο γεγονός, που έχει παραχθεί από έναν εκδότη, το οποίο ταιριάζει με το καταχωρημένο ενδιαφέρον τους. Με άλλα λόγια, οι παραγωγοί (δηλαδή οι εκδότες) δημοσιοποιούν πληροφορίες σε έναν *υπεύθυνο γεγονότων* (event manager) — ένα κομμάτι λογισμικού / εφαρμογής στο διαδίκτυο — και οι καταναλωτές (δηλαδή οι συνδρομητές) γίνονται συνδρομητές σε πληροφορίες που επιθυμούν να λάβουν από τον υπεύθυνο γεγονότων. Τυπικά, οι πληροφορίες δηλώνονται με τον όρο γεγονός και η πράξη της παράδοσής τους με τον όρο

ειδοποίηση [28].



Σχήμα 2.10: Υπηρεσία ειδοποίησης γεγονότων στο πλαίσιο ενός συστήματος Δημοσίευσης / Συνδρομής.

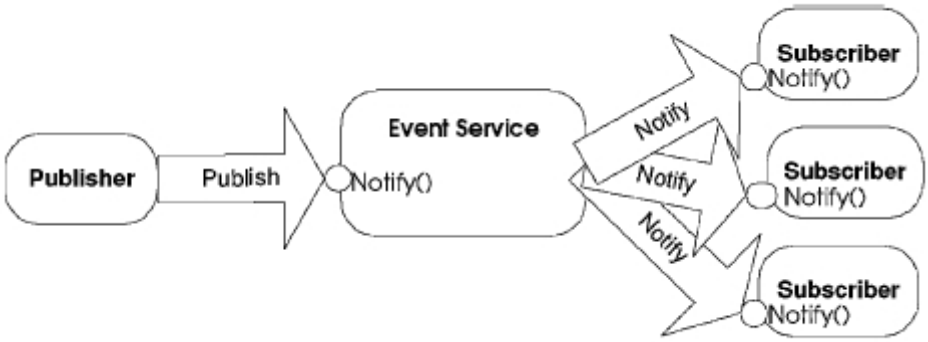
Το βασικό μοντέλο συστήματος για μία αλληλεπίδραση δημοσίευσης / συνδρομής (Σχήμα 2.10) βασίζεται σε μία *υπηρεσία ειδοποίησης γεγονότων* (event notification service) παρέχοντας αποθήκευση και διαχείριση των συνδρομών και της αποδοτικής παράδοσης των γεγονότων. Μία τέτοια υπηρεσία αναπαριστά ένα φυσικό διαμεσολαβητή μεταξύ των εκδοτών, δρώντας ως παραγωγός γεγονότων, και των συνδρομητών, δρώντας ως καταναλωτής γεγονότων. Οι συνδρομητές καταχωρούν το ενδιαφέρον τους σε γεγονότα με την κλήση της λειτουργίας $subscribe()$ που προσφέρει η υπηρεσία ειδοποίησης γεγονότων, χωρίς να γνωρίζουν τις πηγές των γεγονότων αυτών. Η συνδρομή ως πληροφορία αποθηκεύεται στην υπηρεσία, ενώ δεν προωθείται ούτε γίνεται γνωστή στους εκδότες. Η συμμετρική λειτουργία της συνδρομής, η $unsubscribe()$ τερματίζει μία συνδρομή.

Για την παραγωγή ενός γεγονότος, ένας εκδότης καλεί την λειτουργία $publish()$ που προσφέρει η υπηρεσία γεγονότων. Στη συνέχεια, η υπηρεσία διαδίδει το γεγονός στους σχετιζόμενους συνδρομητές· έτσι μπορεί να δρα ως πληρεξούσιος των συνδρομητών. Κάθε συνδρομητής θα ειδοποιηθεί για εκείνα τα γεγονότα που ταιριάζουν στις καταχωρημένες προτιμήσεις του. Επίσης, οι εκδότες έχουν τη δυνατότητα να διαφημίσουν ένα γεγονός πριν την δημοσιοποίησή του καλώντας την λειτουργία $advertise()$. Αυτή η λειτουργία είναι χρήσιμη, διότι δίνει τη δυνατότητα:

1. στην υπηρεσία διαχείρισης γεγονότων να προσαρμοσθεί στην αναμενόμενη ροή των γεγονότων.

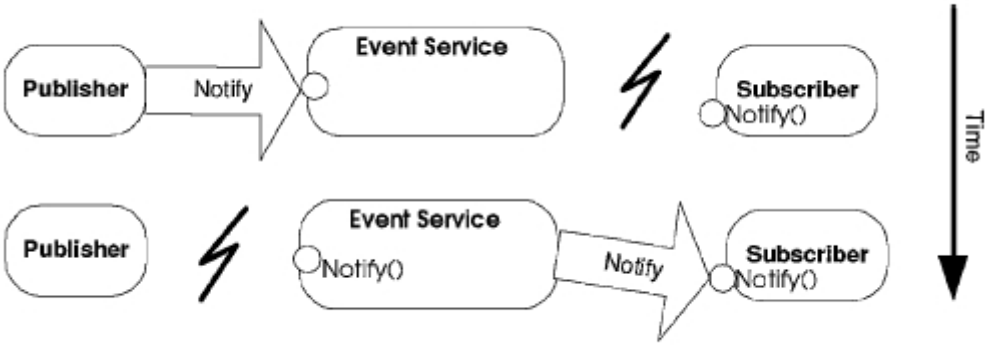
2. στους συνδρομητές να μάθουν πότε ένας συγκεκριμένος τύπος πληροφορίας θα είναι διαθέσιμος.

Η αρχιτεκτονική του συστήματος LibraRing προσφέρει μόνο τις δύο βασικές λειτουργίες, $publish()$, $subscribe()$, όμως είναι εύκολο να επεκταθεί και με τις λειτουργίες που παρουσιάστηκαν παραπάνω.



Σχήμα 2.11: Αποσύνδεση χώρου στα πλαίσια ενός συστήματος παροχής υπηρεσιών Δημοσίευσης / Συνδρομής.

Αυτή η υπηρεσία διαχείρισης γεγονότων απλοποιεί την διαδικασία επικοινωνίας μεταξύ των εκδοτών και των συνδρομητών ένα χαρακτηριστικό το οποίο είναι γνωστό με τον όρο *αποσύνδεση* (decoupling). Η αποσύνδεση που προσφέρει αυτή η υπηρεσία μπορεί να αποσυντεθεί σε τρεις διαστάσεις:

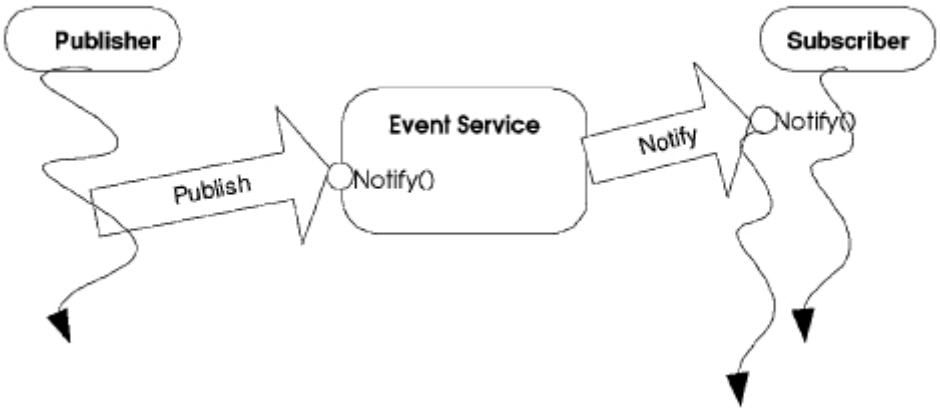


Σχήμα 2.12: Αποσύνδεση χρόνου στα πλαίσια ενός συστήματος παροχής υπηρεσιών Δημοσίευσης / Συνδρομής.

1. *Αποσύνδεση χώρου* (Σχήμα 2.11): οι οντότητες που αλληλεπιδρούν δεν χρειάζεται να γνωρίζουν η μία την άλλη. Οι εκδότες δημοσιοποιούν κάποιο γεγονός μέσω της παραπάνω υπηρεσίας και οι συνδρομητές ειδοποιούνται έμμεσα από αυτή. Ούτε οι εκδότες, ούτε οι συνδρομητές γνωρίζουν ποιός/ποιοί έχει/έχουν γίνει συνδρομητής(ές) σε/δημοσιοποιήσει κάποιο γεγονός.

2. *Αποσύνδεση χρόνου* (Σχήμα 2.12): οι οντότητες που αλληλεπιδρούν δεν χρειάζεται να συμμετέχουν ενεργά την ίδια χρονική στιγμή. Πιο συγκεκριμένα, ο εκδότης μπορεί να δημοσιοποιήσει κάποια γεγονότα, ενώ ο συνδρομητής να είναι αποσυνδεδεμένος από την υπηρεσία, και αντίστροφα, ο συνδρομητής μπορεί να ειδοποιηθεί από την υπηρεσία για κάποιο γεγονός, ενώ ο πραγματικός εκδότης αυτού γεγονότος να είναι αποσυνδεδεμένος.

3. *Αποσύνδεση συγχρονισμού* (Σχήμα 2.13): Οι εκδότες δεν αποτρέπονται από την πολλαπλή δημοσιοποίηση γεγονότων, ενώ οι συνδρομητές μπορούν να ειδοποιούνται για την ύπαρξη ενός γεγονότος που ικανοποιεί τα ενδιαφέροντά τους ασύγχρονα ενόσω πραγματοποιούν κάποιες άλλες ενέργειες ασυσχέτιστες με τη συνδρομή τους. Η παραγωγή και η κατανάλωση των γεγονότων δεν γίνεται στην κύρια ροή ελέγχου των εκδοτών και των συνδρομών και έτσι δεν πραγματοποιούνται σύγχρονα.



Σχήμα 2.13: Αποσύνδεση συγχρονισμού στα πλαίσια ενός συστήματος παροχής υπηρεσιών Δημοσίευσης / Συνδρομής.

Αποσυνδέοντας την παραγωγή και την κατανάλωση των πληροφοριών αυτό που πετυχαίνεται είναι η αύξηση της κλιμάκωσης του συστήματος λόγω της αφαίρεσης των ρητών εξαρτήσεων μεταξύ των συμμετεχόντων στην αλληλεπίδραση.

2.5 AWPS — Ένα Μοντέλο Αναπαράστασης Πληροφορίας

Στις Ενότητες 2.3 και 2.4 παρουσιάστηκαν δύο βασικές λειτουργίες που προσφέρει η αρχιτεκτονική του συστήματος LibraRing, η Ανάκτηση Πληροφορίας και η Δημοσίευση / Συνδρομή. Και οι δύο λειτουργίες μαζί ορίζουν τη διεπαφή του συστήματος LibraRing προς το χρήστη του και τα όρια στα οποία μπορεί να το εκμεταλλευτεί. Έτσι, ένας χρήστης μπορεί να χρησιμοποιήσει τη λειτουργία της Ανάκτησης Πληροφορίας κατασκευάζοντας μία *επερώτηση* (query) και θέτοντάς την στο σύστημα LibraRing προς απάντηση. Εναλλακτικά, χρησιμοποιώντας τη δεύτερη λειτουργία, τη Δημοσίευση / Συνδρομή, ένας χρήστης είτε μπορεί να εισάγει δεδομένα στο σύστημα κατασκευάζοντας μία *δημοσίευση* (publication) και αποστέλλοντάς την σε αυτό, είτε μπορεί να ενημερωθεί για δεδομένα που καλύπτουν διάφορες περιοχές των ενδιαφερόντων του, μέσω της λήψης μίας *ειδοποίησης* (notification), αφού πρώτα έχει καταχωρήσει τις προτιμήσεις του μέσω της κατασκευής μίας *συνδρομής* (subscription).

Συνολικά, για αυτές τις δύο λειτουργίες, το σύστημα LibraRing προσφέρει στον χρήστη του τη διεπαφή που συνίσταται από τις παρακάτω ενέργειες:

1. Επερώτηση,
2. Δημοσίευση,
3. Συνδρομή,
4. Ειδοποίηση.

Το θέμα αυτής της ενότητας είναι η παρουσίαση ενός τρόπου μοντελοποίησης των πληροφοριών που εμπεριέχονται σε κάθε ενέργεια έτσι ώστε αυτές να είναι καταληπτές τόσο από το σύστημα όσο και από το χρήστη, ενώ ταυτόχρονα να έχουν μία μορφή που να μπορεί να είναι επεξεργάσιμη σε ένα χαμηλότερο επίπεδο αφαίρεσης από το σύστημα. Για την επίτευξη αυτού του σκοπού, είναι απαραίτητη η χρήση ενός Μοντέλου Αναπαράστασης Πληροφορίας, όπως το AWPS.

Το AWPS (Attribute Word-Pattern with Similarity) είναι ένα μοντέλο αναπαράστασης πληροφορίας που βασίζεται σε πολύ γνωστές έννοιες από τα μοντέλα Boolean και VSM. Υπό αυτή του την ιδιότητα αποτελεί ένα άλλο μοντέλο που θα μπορούσε να χρησιμοποιηθεί από κάθε σύστημα ΑΠ ή ΔΠ. Το AWPS παρουσιάστηκε για πρώτη φορά στη μορφή που θα το χρησιμοποιήσουμε στις εργασίες [21, 27] από τους Μ. Κουμπαράκη, Τ. Κούτρης, Χ. Τρυφονόπουλος και Π. Ραυτοπούλου κατά την ανάπτυξη της αρχιτεκτονικής DIAS (Distributed

Information Alert System) στα πλαίσια του ευρωπαϊκού προγράμματος DIET [1, 11, 25] για τον προσδιορισμό και την αναπαράσταση των επερωτήσεων και ειδοποιήσεων. Το μοντέλο αυτό είναι επέκταση των μοντέλων WP (Word-Pattern) και AWP (Attribute Word-Pattern) και προσφέρει μία αποτελεσματικότερη και υψηλότερης πιστότητας αναπαράσταση πληροφοριών. Το AWPS βασίζεται σε *γνωρίσματα* (attributes) ή *πεδία* (fields) με λεκτικές τιμές πεπερασμένου μήκους. Επίσης, παρέχεται και ο τελεστής '$\sim$' που αναπαριστά την *ομοιότητα* (similarity) για να εκφράσει τη συνάφεια που υπεισέρχεται σε συστήματα ΑΠ, όπως παρουσιάστηκε και στην ενότητα 2.3. Ο ακριβής μαθηματικός ορισμός του WP δίνεται στα [19, 20], ενώ των AWP και AWPS δίνεται στο [21].

Στη συνέχεια θα περιγράψουμε το μοντέλο AWPS δίνοντας ένα παράδειγμα εφαρμογής του πάνω στην αναπαράσταση μίας δημοσίευσης και μίας επερώτησης. Όμοια μπορεί να εφαρμοστεί και για την αναπαράσταση της συνδρομής. Σημειώνεται, ότι για την ειδοποίηση δεν τίθεται θέμα αναπαράστασης, αφού δεν φέρει πολύπλοκες και δυναμικές πληροφορίες. Ειδικά στο σύστημα LibraRing μία ειδοποίηση συνίσταται σε ένα ζεύγος τιμών που αναπαριστούν την οντότητα του παρόχου που δημοσίευσε τον πόρο και ένα αναγνωριστικό για τον ίδιο τον πόρο.

Μία δημοσίευση είναι ένα σύνολο από ζεύγη γνώρισμα-τιμή (A, s), όπου A είναι το όνομα του γνωρίσματος, s η λεκτική τιμή του και όλα τα γνωρίσματα είναι *μοναδικά*. Το παρακάτω είναι ένα παράδειγμα μίας δημοσίευσης:

{ (AUTHOR, "John Smith"), (TITLE, "Information dissemination in P2P ..."), (ABSTRACT, "In this paper we show that ...") }

Για μία επερώτηση, το μοντέλο AWPS προσφέρει επιπλέον τελεστές ισότητας, περιεκτικότητας και ομοιότητας που εφαρμόζονται πάνω στις τιμές των γνωρισμάτων. Ο τελεστής περιεκτικότητας μεταφράζεται βάσει του μοντέλου Boolean[13] και επιτρέπει επερωτήσεις τύπου Boolean και *εγγύτητας λέξεων* (word proximity). Ο τελεστής ομοιότητας ορίζεται ως το συνημίτονο της γωνίας που σχηματίζουν δύο διανύσματα που αντιστοιχούν στις λεκτικές τιμές αντίστοιχων γνωρισμάτων μίας δημοσίευσης και της επερώτησης. Οι διανυσματικές αναπαραστάσεις των λεκτικών τιμών των γνωρισμάτων μπορούν να υπολογισθούν χρησιμοποιώντας τα μοντέλα VSM[14] (Vector Space Model) ή LSI (Latent Semantic Indexing). Στην υλοποίηση του συστήματος LibraRing έχει χρησιμοποιηθεί μόνο το μοντέλο VSM.

[13] Μία πλήρης περιγραφή του Boolean μοντέλου υπάρχει στο [29].

[14] Μία πλήρης περιγραφή του VSM μοντέλου υπάρχει στο [29].

Πιο συγκεκριμένα, μία επερώτηση είναι μία σύζευξη από ατομικές επερωτήσεις της μορφής

$$A = s, A \leq wp \text{ ή } A \sim_k s$$

, όπου το A είναι ένα γνώρισμα, το s είναι η λεκτική του τιμή, το wp είναι μία σύζευξη από λέξεις και *φόρμουλες εγγύτητας* (proximity formulas) με υποφόρμουλες που αποτελούνται μόνο από λέξεις, και το k είναι το *κατώφλι ομοιότητας* (similarity threshold), δηλαδή ένας πραγματικός αριθμός στο διάστημα $[0, 1]$. Έτσι, οι επερωτήσεις μπορούν να έχουν δύο τμήματα: ένα τμήμα που ερμηνεύεται σύμφωνα με το μοντέλο Boolean και ένα τμήμα που ερμηνεύεται σύμφωνα με το μοντέλο VSM ή LSI. Το παρακάτω είναι ένα παράδειγμα μίας επερώτησης:

> (AUTHOR = "John Smith") ∧ (TITLE ⊒ P2P ∧ (information $\prec_{[0,0]}$ alert)) ∧ (ABSTRACT $\sim_{0.7}$ "P2P architecture have been...")

Η παραπάνω επερώτηση αιτεί όλες τις δημοσιεύσεις που έχουν συνταχθεί από τον John Smith, και ο τίτλος τους περιέχει τη λέξη P2P και ένα *λεκτικό πρότυπο ή σχήμα* (word pattern), όπου η λέξη information ακολουθείται άμεσα από τη λέξη alert. Επιπροσθέτως, οι δημοσιεύσεις θα πρέπει να έχουν μία περίληψη που το περιεχόμενό της να είναι όμοιο με τη λεκτική τιμή "P2P architecture have been..." με ποσοστό ομοιότητας μεγαλύτερο του 0.7.

2.6 XML

Στην προηγούμενη ενότητα παρουσιάστηκε το AWPS, ένα μοντέλο αναπαράστασης πληροφορίας. Θα λέγαμε ότι χρησιμοποιήσαμε το AWPS για να μετατρέψουμε το αρχικό πρόβλημα, που ήταν εκφρασμένο σε μία κοινή γλώσσα με σημασιολογικές και συντακτικές ασάφειες, σε μία μαθηματική μορφή, πιο εύληπτη, πιο σαφή και πιο λακωνική. Σε αυτή την ενότητα θα εστιάσουμε στον τρόπο αναπαράστασης των δεδομένων και της σημασιολογίας που ορίζει το μοντέλο AWPS χρησιμοποιώντας μία γλώσσα.

Μία τέτοια γλώσσα θα μπορούσε να ήταν η XML (EXtensible Markup Language) ή η XPath (XML Path Language). Για την υλοποίηση της αρχιτεκτονικής του συστήματος LibraRing έχει χρησιμοποιηθεί η γλώσσα XML και γι' αυτό στην επόμενη υποενότητα δίνεται μία σύντομη περιγραφή της. Αυτές οι γλώσσες είναι οι κατάλληλες για την κωδικοποίηση

της μοντελοποίησης ενός προβλήματος, διότι παρέχουν δύο σημαντικά χαρακτηριστικά που χρησιμοποιεί και το μοντέλο AWPS. Αυτά είναι η ύπαρξη *νοητών αντικειμένων ή γνωρισμάτων* που δύναται να έχουν κάποια τιμή και η ύπαρξη σχέσεων μεταξύ τους. Προφανώς, το πρώτο χαρακτηριστικό αντιστοιχεί στα ζευγάρια γνώρισμα-τιμή (A, s) του μοντέλου AWPS, ενώ το δεύτερο αντιστοιχεί στους τελεστές μεταξύ των τιμών των γνωρισμάτων.

2.6.1 Απαιτούμενο Υπόβαθρο

Η XML εκτός από το να είναι μία γλώσσα σήμανσης και μορφοποίησης (markup language), όπως είναι η HTML, είναι και μία μεταγλώσσα βάσει της οποίας μπορούν να δημιουργηθούν νέες γλώσσες. Η XML δεν έχει προκαθορισμένες *ετικέτες* (tags) και εστιάζεται στην περιγραφή των δεδομένων και των μεταξύ τους σχέσεων, δηλαδή στη σημασιολογία τους, και όχι στον τρόπο εμφάνισής τους. Είναι μία γλώσσα επεκτάσιμη και ευέλικτη, καθώς επιτρέπει την προσθήκη νέων ετικετών και στοιχείων για την περιγραφή των δεδομένων της. Ένα επιπλέον χαρακτηριστικό της είναι ότι έχει τη δυνατότητα να ορίζει τα στοιχεία που χρησιμοποιούνται, αλλά και να επαληθεύει την εγκυρότητά τους. Αυτό προσφέρεται μέσω του Ορισμού Τύπου Εγγράφου (Document Type Definition, DTD). Το βασικό της πλεονέκτημα, όμως, είναι ότι μπορεί να χρησιμοποιηθεί σε κάθε πλατφόρμα, επειδή αποτελείται από κώδικα σε μορφή κειμένου ASCII. Έτσι χρησιμοποιώντας την μπορεί να επιτευχθεί η ανταλλαγή δεδομένων μεταξύ διαφορετικών εφαρμογών ή και συστημάτων διασφαλίζοντας με αυτό τον τρόπο τη διαλειτουργικότητα.

Σύμφωνα με τα παραπάνω, θα μπορούσαμε να αναπαραστήσουμε μία οντότητα, έναν τύπο δεδομένων, μία συγκεκριμένη πληροφορία κ.ο.κ μέσω μίας ετικέτας. Το παρακάτω παράδειγμα είναι ένα μήνυμα για τον John στη Mary χρησιμοποιώντας τη γλώσσα XML [36].

```
<message>
    <to>John</to>
    <from>Mary</from>
    <date>15-12-03</date>
    <subject>Meeting</subject>
    <content>Don't forget our meeting tomorrow at 11 am!</content>
</message>
```

Το περιεχόμενο του μηνύματος έχει συμπεριληφθεί σε μία ετικέτα <message>, ενώ η

πληροφορία που περιλαμβάνει το μήνυμα έχει περαιτέρω δομηθεί με τη χρήση επιπλέον ετικετών. Το μήνυμα έχει ημερομηνία date, ένα θέμα subject και περιεχόμενο content. Επίσης, έχει πληροφορία για τον αποστολέα from και τον παραλήπτη to. Τονίζεται, ότι οι ετικέτες που χρησιμοποιήθηκαν στο παράδειγμα δεν είναι καθορισμένες σε κάποιο πρότυπο XML, αλλά "επινοήθηκαν" από το συγγραφέα του εγγράφου XML. Αυτό το έγγραφο XML δεν περιέχει εκτελέσιμο κώδικα, αλλά απλώς παρουσιάζει πληροφορία με ένα δομημένο τρόπο.

Ένα άλλο χαρακτηριστικό της XML είναι ότι κάθε ετικέτα μπορεί να ενσωματώνει ένα πλήθος από γνωρίσματα (attributes), τα οποία της δίνουν επιπρόσθετη πληροφορία. Για παράδειγμα θα μπορούσαμε να έχουμε μία ετικέτα, product, που αναπαριστά τη νομισματική αξία ενός προϊόντος. Σε αυτή την περίπτωση η χρήση ενός γνωρίσματος, currency, θα βοηθούσε στον προσδιορισμό του νομίσματος. Κάτι τέτοιο, θα μπορούσαμε να το αναπαριστήσουμε ως ακολούθως:

```
<product>
    <price currency="Euro">25</price>
    <!- define optional characteristics here ->
</product>
```

Βέβαια, στο παραπάνω παράδειγμα θα μπορούσαμε να μην είχαμε χρησιμοποιήσει το γνώρισμα, αλλά διαφορετικά να είχαμε εμφωλιάσει άλλη μία ετικέτα στην ετικέτα του προϊόντος που θα είχε ως τιμή τη λεκτική τιμή του νομίσματος. Γενικά, εάν υπάρχει περίπτωση να έχουμε ταυτόχρονα δύο ή και περισσότερες τιμές για την αναπαράσταση μίας συγκεκριμένης πληροφορίας προτιμάται η χρήση των ετικετών. Αντίθετα, εάν γνωρίζουμε ότι η πληροφορία μπορεί να λάβει μόνο μία τιμή, είναι προτιμότερο να γίνεται χρήση των γνωρισμάτων. Με αυτόν τον τρόπο εξοικονομείται ο όγκος των δεδομένων και έτσι μία τέτοια απόφαση είναι αρκετά σημαντική για ανταλλαγή μεγάλων δεδομένων XML. Έτσι, για την αναπαράσταση ενός βιβλίου με πολλούς συγγραφείς θα διαλέγαμε την πρώτη περίπτωση.

```
<book>
    <author>First Author</author>
    <author>Second Author</author>
    <author>Third Author</author>
</book>
```

Έχοντας μιλήσει για τα βασικά δομικά συστατικά της γλώσσας XML ας μιλήσουμε για την εγκυρότητα των εγγράφων XML και την επαλήθευσή της. Ένα τέτοιο έγγραφο που είναι συντακτικά σωστό ονομάζεται καλά σχηματισμένο. Παρ' όλα αυτά, ένα καλά σχηματισμένο έγγραφο δεν είναι απαραίτητα και έγκυρο. Για να είναι ένα έγγραφο έγκυρο θα πρέπει εκτός από συντακτικά σωστό να βασίζεται και στους κανόνες ενός ορισμού τύπου εγγράφου. Ένα τέτοιο έγγραφο προσδιορίζει τη δομή ενός εγγράφου XML καθώς και τη λίστα με τις "νόμιμες" ετικέτες που το αποτελούν. Θα λέγαμε ότι παρέχει τους γραμματικούς κανόνες του εγγράφου και των στοιχείων του. Στην ουσία περιγράφει ένα μοντέλο για τη δομή και το περιεχόμενο ενός εγγράφου XML. Περιγράφει τις σχέσεις ανάμεσα στα στοιχεία του εγγράφου και προσδιορίζει ποια στοιχεία είναι απαραίτητο να υπάρχουν και ποια μπορεί να παραληφθούν. Ένα παράδειγμα ενός ορισμού τύπου εγγράφου είναι το παρακάτω:

```
<!ELEMENT product (name, category?, color?, description?, product-number,
availability?)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT product-number (#PCDATTA)>
<!ELEMENT availability (#PCDATA)>
<!ATTLIST product price CDATA #REQUIRED>
```

Οι αρχικές ετικέτες του παραδείγματος αφορούν στον προσδιορισμό στοιχείων, ενώ η τελευταία στον προσδιορισμό των χαρακτηριστικών ενός στοιχείου. Οι ετικέτες ενός DTD ξεκινούν με θαυμαστικό "!", το οποίο ακολουθείται από μία δεσμευμένη λέξη (ELEMENT, ENTITY, ATTLIST κ.α.) που προσδιορίζει το είδος της οντότητας που θα οριστεί. Ο παραπάνω ορισμός μπορεί να ειπωθεί ως εξής: ένα προϊόν που είναι μία ετικέτα μπορεί να έχει ως τιμή ένα μοναδικό όνομα, να ανήκει σε καμία ή μία κατηγορία (αυτό υποδηλώνεται από το χαρακτήρα "?"), να έχει ή να μην έχει χρώμα και περιγραφή, να έχει ένα μοναδικό αριθμό προϊόντος και τέλος να προσδιορίζεται ή όχι η διαθεσιμότητά του. Ακολούθως, οι ετικέτες του ονόματος, της περιγραφής, του αριθμού προϊόντος και της διαθεσιμότητας θα πρέπει να έχουν ως τιμή ένα αλφαριθμητικό — δηλώνεται με τη χρήση του "#PCDATA". Τέλος, η ετικέτα του προϊόντος θα πρέπει υποχρεωτικά — δηλώνεται με το προσδιορισμό "#REQUIRED" — να φέρει το γνώρισμα της νομισματικής του αξίας, η τιμή της οποίας μπορεί να είναι ένα αλφαριθμητικό.

Σύμφωνα με αυτά, το παρακάτω έγγραφο XML είναι καλά σχηματισμένο, αλλά όχι και

έγκυρο, διότι απουσιάζει το γνώρισμα της τιμής από την ετικέτα του προϊόντος και επίσης έχουν οριστεί δύο κατηγορίες για αυτό.

```
<product>
    <name>La Trappe Quadrupel</name>
    <category>Beers</category>
    <category>Wines</category>
    <product-number>39841748567300</product-number>
    <availability>true</availability>
    <description>La Trappe Quadrupel is La Trappe's strongest beer with a
    marvelous amber colour. Its warm taste is full-bodied and mild, slightly
    sweet and pleasantly bitter. A unique beer that is actually laid down for
    years in the cellars of the abbey for further condition. La Trappe
    Quadrupel has 10% ABV.</description>
</product>
```

Τονίζεται ότι η σειρά με την οποία εμφανίζονται οι εμφωλιασμένες ετικέτες δεν έχει σημασία.

2.6.2 Αναπαράσταση Στοιχείων του AWPS με Χρήση XML

Σε αυτή την υποενότητα και έχοντας παρουσιάσει το απαιτούμενο υπόβαθρο της γλώσσας XML θα αναφερθούμε στον τρόπο περιγραφής των δεδομένων που ανταλλάσσονται μεταξύ των πελατών, παρόχων και των υπερ-κόμβων του συστήματος LibraRing. Αυτά συνίστανται στα δεδομένα που ήδη έχουν αναπαρασταθεί με τη χρήση του AWPS στην ενότητα 2.5, δηλαδή στις δημοσιεύσεις, στις επερωτήσεις και στις συνδρομές[15].

Μία δημοσίευση μπορεί να αναπαρασταθεί με τη γλώσσα XML βάσει του μοντέλου AWPS χρησιμοποιώντας μία ετικέτα που καθορίζει την αρχή μίας δημοσίευσης και ενός αυθαίρετου αριθμού από εμφωλιασμένες ετικέτες που περιγράφουν την δημοσίευση και αντιστοιχούν στα ζεύγη γνωρίσματος-τιμής του μοντέλου AWPS. Επειδή, το σύστημα LibraRing έχει σχεδιαστεί

[15]Σημειώνεται ότι το σύστημα LibraRing χρησιμοποιεί τον όρο "επερώτηση διαρκείας" για να εκφράσει τα δεδομένα που ανταλλάσσονται μεταξύ ενός πελάτη και ενός υπερ-κόμβου, ώστε ο πρώτος να γίνει συνδρομητής σε συγκεκριμένες δημοσιεύσεις. Αυτός ο όρος εισάγεται στην Κεφάλαιο 3, όπου περιγράφεται η αρχιτεκτονική του συστήματος.

ώστε να μην περιορίζει τον τύπο της δημοσίευσης, τα ονόματα και ο αριθμός των ετικετών δεν μπορεί να είναι προκαθορισμένος. Έτσι, κάθε πάροχος έχει τη δυνατότητα να κατασκευάσει μία περιγραφή που να αρμόζει στις επιθυμίες και τις ανάγκες του ή ακόμα να δημοσιοποιήσει οποιουδήποτε τύπου πόρους, όπως ταινίες, μουσική, βιβλία. Σε κάθε περίπτωση το σύστημα LibraRing προσφέρει δυνατότητα εξακρίβωσης της εγκυρότητας ενός XML εγγράφου, οπότε κάθε τέτοια περιγραφή θα πρέπει να συνοδεύεται και από το αντίστοιχο έγγραφο DTD.

Σύμφωνα με τα παραπάνω, θα μπορούσε ένας πάροχος να δημοσιοποιεί πόρους που να αναπαριστούν βιβλία. Αυτά τα βιβλία μπορεί να περιγράφονται από συγγραφέα, τίτλο, ημερομηνία έκδοσης, τύπος έκδοσης, εκδοτικό οίκο, κριτική, θέμα και ένα αναγνωριστικό κωδικό. Έτσι, μία δημοσίευση βιβλίου θα μπορούσε να αναπαρασταθεί με XML ως ακολούθως:

```
<Book>
    <Author>Joseph Heller</Author>
    <Title>Catch</Title>
    <PublicationDate>09-2000</PublicationDate>
    <Edition>Trade</Edition>
    <Isbn>0684833395</Isbn>
    <Publisher>Simon and Schuster<Publisher>
    <Subject>Science Fiction</Subject>
    <Review>http://www.book-review.com/heller-catch22.html</Review>
</Book>
```

Πίνακας 2.7: Αναπαράσταση δημοσίευσης με XML

Αυτός ο τύπος περιγραφής ακολουθεί τους κανόνες που ορίζει το επόμενο έγγραφο DTD:

```
<!ELEMENT Book (Author?, Title, PublicationDate?, Edition?, Isbn?,
Publisher?, Subject?, Review?)* >
<!ELEMENT Author (#PCDATA)>
<!ELEMENT Title (#PCDATA)>
<!ELEMENT PublicationDate (#PCDATA)>
<!ELEMENT Edition (#PCDATA)>
<!ELEMENT Isbn (#PCDATA)>
<!ELEMENT Publisher (#PCDATA)>
<!ELEMENT Subject (#PCDATA)>
```

```
<!ELEMENT Review (#PCDATA)>
```

Πίνακας 2.8: Έγγραφο DTD για την αναπαράσταση μίας δημοσίευσης

Όμοια, ένας πελάτης έχει τη δυνατότητα να θέσει μία επερώτηση που να αφορά σε οποιοδήποτε τύπο δημοσίευσης. Παρ' όλα αυτά, οι επερωτήσεις έχουν μία πιο καθορισμένη δομή από τις δημοσιεύσεις, όσον αφορά στις ετικέτες. Όλες οι επερωτήσεις έχουν συγκεκριμένες ετικέτες, οι οποίες αντιστοιχούν στους τελεστές του μοντέλου AWPS και είναι οι επόμενες:

equals: αναπαριστά τον τελεστή ισότητας,

contains: αναπαριστά τον τελεστή περιεκτικότητας,

resembles: αναπαριστά τον τελεστή ομοιότητας.

Κάθε τέτοιος τελεστής είναι δυαδικός, δηλαδή εφαρμόζεται σε δύο τελεστέους και αναπαριστά τη σχέση τους. Κάτι τέτοιο μπορεί να εκφραστεί αν κάθε ετικέτα τελεστή φέρει ένα γνώρισμα που θα αντιστοιχεί στον πρώτο τελεστέο, ενώ ο άλλος τελεστέος θα αντιστοιχεί στη τιμή της ετικέτας. Αυτό το γνώρισμα είναι που κάνει μία επερώτηση να είναι γενική και να μπορεί να αναφερθεί σε οποιοδήποτε τύπο δημοσίευσης. Ειδικά για τον τελεστή ομοιότητας θα πρέπει να φέρει και ένα δεύτερο γνώρισμα μέσω του οποίου θα καθορίζεται το κατώφλι ομοιότητας. Επίσης, το μοντέλο AWPS παρέχει τη δυνατότητα επερώτησης της ομοιότητας και της περιεκτικότητας μίας λέξης ή ενός λεκτικού προτύπου με/σε ένα γνώρισμα. Κάτι τέτοιο, μπορεί να αναπαρασταθεί εύκολα, αν οι αντίστοιχες ετικέτες έχουν ως τιμή έναν αριθμό από ετικέτες (<word>) που έχουν ως τιμή τη λέξη ή το λεκτικό πρότυπο. Έτσι, για την επερώτηση ενός βιβλίου θα μπορούσαμε να χρησιμοποιήσουμε το επόμενο έγγραφο XML:

```
<query>
    <equals attrName="Edition">Trade</equals>
    <contains attrName="Subject">
        <word>Science</word>
    </contains>
    <contains attrName="Author">
        <word>Joseph</word>
    </contains>
```

```
    <resembles attrName="Publisher" threshold="0.6">
        <word>Simon Nikolaou Schuster</word>
    </resembles>
<query>
```

Πίνακας 2.9: Αναπαράσταση επερώτησης με XML

Το έγγραφο DTD που περιγράφει τις επερωτήσεις που ορίστηκαν είναι ο επόμενος:

```
<!ELEMENT query (equals*,contains*,resembles*)>
<!ELEMENT equals (#PCDATA)>
<!ATTLIST equals attrName CDATA #REQUIRED>
<!ELEMENT contains (word+)>
<!ATTLIST contains attrName CDATA #REQUIRED>
<!ELEMENT resembles (word+)>
<!ATTLIST resembles attrName CDATA #REQUIRED threshold CDATA #REQUIRED>
<!ELEMENT word (#PCDATA)>
```

Πίνακας 2.10: Έγγραφο DTD για την αναπαράσταση μίας επερώτησης

Ο τρόπος συνδρομής ενός πελάτη μπορεί να αναπαρασταθεί κατά τον ίδιο τρόπο με αυτό της επερώτησης, αφού μία συνδρομή είναι μία επερώτηση διαρκείας, δηλαδή εκτελείται κάθε φορά που μία νέα δημοσίευση καταχωρείται στο σύστημα LibraRing. Σε αυτή την περίπτωση η αναπαράσταση θα αποθηκευτεί στον υπερ-κόμβο του συστήματος LibraRing και ο πελάτης δεν θα αναμένει άμεση απάντηση.

2.7 Συμπεράσματα

Στο Κεφάλαιο αυτό έγινε μια παρουσίαση του θεωρητικού υπόβαθρου που είναι απαραίτητο για την κατανόηση αυτής της πτυχιακής εργασίας. Συγκεκριμένα, αναφερθήκαμε στην τεχνολογία των δικτύων ομότιμων κόμβων και ιδιαίτερα στους κατανεμημένους πίνακες κατακερματισμού. Επιπλέον, παρουσιάστηκαν δύο συστήματα που υλοποιούν τη διεπαφή των κατανεμημένων πινάκων κατακερματισμού, το Chord και Pastry. Ακόμα, παρουσιάστηκαν οι

έννοιες της ανάκτησης πληροφορίας, ανάκτησης δεδομένων και διήθησης πληροφορίας και της δημοσίευσης / συνδρομής καθώς επίσης και το μοντέλο αναπαράστασης πληροφορίας AWPS που χρησιμοποιήθηκε στην υλοποίηση της αρχιτεκτονικής LibraRing. Τέλος, έγινε μία μικρή εισαγωγή στη γλώσσα XML παρουσιάζοντας τα απαραίτητα συστατικά της που θα βοηθήσουν στην κατανόηση της χρήσης της στην υλοποίηση της αρχιτεκτονικής LibraRing.

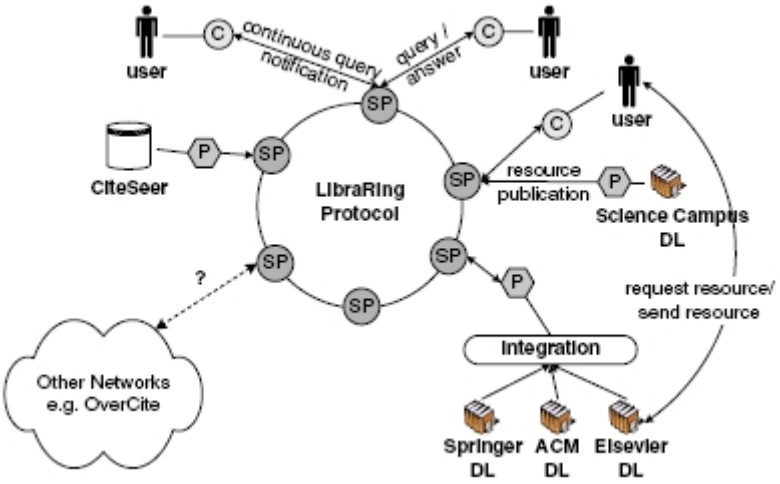
Κεφάλαιο 3

Η Αρχιτεκτονική LibraRing

Σε αυτό το κεφάλαιο θα μιλήσουμε για την αρχιτεκτονική LibraRing, μια αρχιτεκτονική για κατανεμημένες ψηφιακές βιβλιοθήκες βασισμένη στους κατανεμημένους πίνακες κατακερματισμού, όπως αυτή περιγράφεται στο [8]. Η αρχιτεκτονική αυτή προσφέρει δύο βασικές λειτουργίες: ανάκτηση πληροφορίας καθώς επίσης και δημοσίευση / συνδρομή. Σε ένα σενάριο που αφορά στην πρώτη λειτουργία ένας χρήστης έχει τη δυνατότητα να υποβάλει μία επερώτηση (π.χ. "Θα ήθελα δημοσιεύσεις από το χώρο της βιοπληροφορικής") και το σύστημα να του επιστρέψει πληροφορίες για τους πόρους που ταιριάζουν με αυτή. Σε ένα σενάριο που αφορά στην δεύτερη λειτουργία ένας χρήστης μπορεί να αποστείλει στο σύστημα μία συνδρομή, η οποία περιγράφει τα ενδιαφέροντά του (π.χ. "Ενδιαφέρομαι για δημοσιεύσεις βιοπληροφορικής"), έτσι ώστε να λάβει ειδοποιήσεις σχετικά με συγκεκριμένα γεγονότα που τον ενδιαφέρουν (π.χ. όταν μία δημοσίευση που αφορά στην βιοπληροφορική γίνει διαθέσιμη στο σύστημα).

Θα μιλήσουμε, επίσης, για τα βασικά δομικά στοιχεία της· τους υπερ-κόμβους, τους πελάτες και τους παρόχους. Οι υπερ-κόμβοι αποτελούν τους κόμβους του δικτύου και παρέχουν τις δύο υπηρεσίες που προλογήθηκαν. Οι πελάτες και οι πάροχοι αλληλεπιδρούν με το σύστημα LibraRing, μέσω των υπερ-κόμβων, και μεταβάλλουν την κατάστασή του. Οι πελάτες μπορούν να υποβάλουν επερωτήσεις, χρησιμοποιώντας τη λειτουργία της ανάκτησης πληροφορίας, και να γίνονται συνδρομητές σε διάφορους πόρους που τους ενδιαφέρουν, χρησιμοποιώντας τη λειτουργία της συνδρομής. Οι πάροχοι μπορούν να δημοσιοποιούν πόρους στο δίκτυο, επικοινωνώντας με τους υπερ-κόμβους και έτσι χρησιμοποιούν την παρεχόμενη λειτουργία της δημοσίευσης.

Μία άποψη υψηλού επιπέδου της αρχιτεκτονικής LibraRing φαίνεται στο Σχήμα 3.1. Οι



Σχήμα 3.1: Η αρχιτεκτονική LibraRing.

κόμβοι του συστήματος LibraRing υλοποιούν τους παρακάτω τύπους υπηρεσιών: *υπηρεσία υπερ-κόμβων, υπηρεσία παρόχων* και *υπηρεσία πελατών*.

Στην Ενότητα 3.1 περιγράφονται οι υπηρεσίες που προσφέρει η αρχιτεκτονική αυτή, αυτές των υπερ-κόμβων, των πελατών και των παρόχων, ενώ στην Ενότητα 3.2 περιγράφονται τα προσφερόμενα πρωτόκολλα επικοινωνίας μεταξύ των πελατών-παρόχων και των υπερ-κόμβων.

3.1 Περιγραφή Προσφερόμενων Υπηρεσιών

3.1.1 Υπηρεσία Υπερ-Κόμβων

Οι κόμβοι που υλοποιούν την υπηρεσία υπερ-κόμβων (υπερ-κόμβοι) διαμορφώνουν το επίπεδο του δικτύου που είναι υπεύθυνο για τη δρομολόγηση μηνυμάτων. Κάθε υπερ-κόμβος είναι υπεύθυνος για την εξυπηρέτηση ενός ποσοστού των πελατών αποθηκεύοντας *επερωτήσεις διαρκείας* (continuous queries) και *δημοσιεύσεις* (publications), απαντώντας σε *στιγμιαίες επερωτήσεις* (one-time queries), και δημιουργώντας *ειδοποιήσεις* (notifications).

Οι πελάτες των υπερ-κόμβων μπορεί να είναι είτε οι πελάτες, είτε οι πάροχοι. Οι υπερ-κόμβοι εκτελούν ένα πρωτόκολλο ΚΠΚ, το οποίο αποτελεί επέκταση του Pastry. Ο ρόλος του ΚΠΚ στο σύστημα LibraRing είναι πολύ σημαντικός. Πρώτα από όλα, δρα ως ένα σημείο επικοινωνίας μεταξύ των παραγωγών πληροφορίας (πάροχοι) και των καταναλωτών αυτής της

πληροφορίας (πελάτες). Με λίγα λόγια, διαδραματίζει το ρόλο της υπηρεσίας ειδοποίησης γεγονότος, όπως αυτός παρουσιάστηκε στην Ενότητα 2.4. Κατά δεύτερον, αποτελεί μία εύρωστη, ανεκτική σε αποτυχίες και κλιμακούμενη υποδομή δρομολόγησης. Όταν ο αριθμός των υπερ-κόμβων είναι μικρός, κάθε κόμβος μπορεί εύκολα να εντοπίσει τους υπόλοιπους μόνο με μία αναπήδηση διατηρώντας έναν πλήρη πίνακα δρομολόγησης. Όταν ο αριθμός των υπερ-κόμβων αυξάνεται, ο ΚΠΚ προσφέρει κλιμακούμενους τρόπους εντοπισμού άλλων κόμβων, όπως αυτοί συζητήθηκαν στην υποενότητα 2.2.4 για την αρχιτεκτονική του Pastry. Τέλος, όντας μία υπηρεσία ευρετηριασμού καθολικών μεταδεδομένων χωρισμένη μεταξύ των υπερ-κόμβων, ο ΚΠΚ διευκολύνει την ανάπτυξη μίας κατανεμημένης *αποθήκης* (repository) μεταδεδομένων που μπορούν να επερωτηθούν αποδοτικά.

Επερωτήσεις Διαρκείας (Continuous Queries): Είναι ο τρόπος μέσω του οποίου προσφέρεται η λειτουργικότητα της συνδρομής ενός πελάτη σε δημοσιεύσεις που είτε υπάρχουν ήδη αποθηκευμένες στο σύστημα LibraRing, είτε πρόκειται να υπάρξουν στο μέλλον. Ουσιαστικά εκφράζουν τις επιθυμίες και τα ενδιαφέροντα των πελατών ως προς τις δημοσιεύσεις. Ο χαρακτηρισμός της επερώτησης ως "διαρκείας" δικαιολογείται από το γεγονός ότι η επερώτηση αποθηκεύεται στο σύστημα και ενεργοποιείται, δηλαδή ελέγχεται η ικανοποίησή της, κάθε φορά που μία νέα δημοσίευση παρουσιάζεται στο σύστημα. Μία επερώτηση αυτού του τύπου και όταν αυτή ικανοποιείται απαντάται πάντοτε υπό τη μορφή ειδοποίησης.

Στιγμιαίες Επερωτήσεις (One-time Queries): Αυτού του τύπου οι επερωτήσεις είναι όμοιες, ως προς τη λειτουργικότητά τους με τις επερωτήσεις που γίνονται σε βάσεις δεδομένων. Ο πελάτης του συστήματος LibraRing θέτει μία επερώτηση στον υπερ-κόμβο που αποτελεί το σημείο πρόσβασής του, αυτή αποστέλλεται και επεξεργάζεται στον υπερ-κόμβο που είναι υπεύθυνος για αυτή την επερώτηση και τέλος του απαντά αποστέλλοντάς του όλες τις περιγραφές των δημοσιεύσεων που υπάρχουν αποθηκευμένες στο σύστημα LibraRing και ταιριάζουν με αυτή την επερώτηση. Ο πελάτης έχοντας την περιγραφή της δημοσίευσης και τον πάροχο που την δημοσιοποίησε έχει τη δυνατότητα να την αιτήσει από αυτόν. Σε περίπτωση που δεν υπάρχουν τέτοιες δημοσιεύσεις η απάντηση είναι κατάλληλη.

3.1.2 Υπηρεσία Πελατών

Οι κόμβοι που υλοποιούν την υπηρεσία πελατών ονομάζονται πελάτες. Ένας πελάτης μπορεί να συνδεθεί στο δίκτυο του LibraRing μέσω ενός υπερ-κόμβου, ο οποίος αποτελεί το *σημείο πρόσβασης* (access point) αυτού. Οι πελάτες μπορούν να συνδεθούν, αποσυνδεθούν ή

ακόμα και να αποχωρήσουν σιωπηλά από το δίκτυο. Οι πελάτες αποτελούν τους καταναλωτές πληροφοριών. Μπορούν να

- θέτουν στιγμιαίες επερωτήσεις και να λαμβάνουν απαντήσεις,
- γίνονται συνδρομητές σε δημοσιεύσεις και να λαμβάνουν ειδοποιήσεις σχετικά με δημοσιεύσεις που ταιριάζουν στις καταχωρημένες προτιμήσεις τους.

Εάν οι πελάτες δεν είναι συνδεδεμένοι στο δίκτυο LibraRing, οι ειδοποιήσεις που ταιριάζουν με τις προτιμήσεις τους αποθηκεύονται στον υπερ-κόμβο που αποτελεί το σημείο πρόσβασής τους και παραδίδονται όταν αυτοί επανασυνδεθούν.

3.1.3 Υπηρεσία Παρόχων

Αυτή η υπηρεσία υλοποιείται από τις πηγές πληροφοριών που θα ήθελαν να εκθέσουν το περιεχόμενό τους στους πελάτες του συστήματος LibraRing. Ένας κόμβος που υλοποιεί την υπηρεσία παρόχων (πάροχος) μπορεί να συνδεθεί στο δίκτυο μέσω ενός υπερ-κόμβου, ο οποίος αποτελεί το σημείο πρόσβασής του. Για την υλοποίηση αυτής της υπηρεσίας, κάθε πηγή πληροφοριών (δηλαδή κάθε πάροχος) θα πρέπει

- να δημιουργεί μεταδεδομένα για τους πόρους που αποθηκεύει χρησιμοποιώντας το μοντέλο αναπαράστασης πληροφορίας AWPS, και
- να τους δημοσιοποιεί σε όλο το δίκτυο του LibraRing δια μέσου του σημείου πρόσβασής του.

Ας πάρουμε ως ένα παράδειγμα εφαρμογής ένα πανεπιστήμιο με τρία γεωγραφικά κατανεμημένα πανεπιστημιακά παραρτήματα (Arts, Sciences, Medicine) και μία τοπική ψηφιακή βιβλιοθήκη σε κάθε τέτοιο παράρτημα. Κάθε παράρτημα διατηρεί τον δικό της υπερ-κόμβο, ο οποίος χρησιμεύει ως ένα σημείο πρόσβασης για τον πάροχο που αντιπροσωπεύει την ψηφιακή βιβλιοθήκη του παραρτήματος, και τους πελάτες που χρησιμοποιούνται από τους χρήστες. Το πανεπιστήμιο μπορεί να ενδιαφέρεται να διαθέσει στους φοιτητές και το προσωπικό του, έγκαιρα, το περιεχόμενο που παρέχεται από άλλους εκδότες (όπως, για παράδειγμα, CiteSeer, ACM, Springer, Elsevier). Στο Σχήμα 1.1 διακρίνεται ο τρόπος μέσω του οποίου η αρχιτεκτονική LibraRing μπορεί να εκπληρώσει μία τέτοια απαίτηση.

3.2 Περιγραφή Προσφερόμενων Πρωτοκόλλων

Σε αυτή την υποενότητα περιγράφεται ο τρόπος με τον οποίο οι πελάτες και οι πάροχοι εισέρχονται/εξέρχονται από το/στο δίκτυο του συστήματος LibraRing. Επίσης, περιγράφονται τα πρωτόκολλα κατάθεσης δημοσιεύσεων και επερωτήσεων. Χρησιμοποιούνται οι συναρτήσεις $H(w)$, $key(n)$, $ip(n)$ και $id(n)$ που δηλώνουν το αναγνωριστικό κόμβου που είναι υπεύθυνο για το αντικείμενο w, το κλειδί, τη διεύθυνση IP και το αναγνωριστικό του κόμβου n αντίστοιχα.

3.2.1 Συμμετοχή Πελάτη

Την πρώτη φορά που ένας πελάτης C επιθυμεί να συνδεθεί στο δίκτυο του LibraRing πρέπει να ακολουθήσει το πρωτόκολλο συμμετοχής. Ο C πρέπει να βρει την διεύθυνση IP ενός υπερ-κόμβου S χρησιμοποιώντας πληροφορίες που δεν έχουν σχέση με το δίκτυο (όπως, για παράδειγμα, μέσω ενός ασφαλούς δικτυακού τόπου που περιέχει διευθύνσεις IP για τους υπερ-κόμβους του δικτύου LibraRing που είναι εκείνη τη στιγμή συνδεδεμένοι). Ο C αποστέλλει στον S ένα μήνυμα $NewClient(key(C),\ ip(C))$ και ο S προσθέτει τον C στον *πίνακα πελατών* (client table - CT), ο οποίος είναι ένας πίνακας κατακερματισμού που χρησιμεύει για την καταχώρηση των ομότιμων κόμβων που χρησιμοποιούν τον S ως το σημείο πρόσβασής τους. Το $key(C)$ χρησιμοποιείται ως πεδίο ευρετηριασμού των πελατών στον πίνακα CT, ενώ κάθε κελί αυτού του πίνακα αποθηκεύει πληροφορίες επικοινωνίας για κάθε πελάτη, όπως την κατάστασή του (συνδεδεμένος / ασύνδετος) και τις αποθηκευμένες ειδοποιήσεις του (βλέπε 3.2.8). Επιπροσθέτως, ο S αποστέλλει στον C ένα μήνυμα βεβαίωσης $AckNewClient(id(S))$. Άπαξ και ο C έχει συμμετάσχει στο δίκτυο LibraRing μπορεί να χρησιμοποιήσει το πρωτόκολλο σύνδεσης / αποσύνδεσης για να συνδεθεί ή αποσυνδεθεί από το δίκτυο.

3.2.2 Συμμετοχή Παρόχου

Οι πάροχοι χρησιμοποιούν ένα παρόμοιο πρωτόκολλο με αυτό των πελατών για να συμμετάσχουν στο δίκτυο LibraRing.

3.2.3 Σύνδεση/Αποσύνδεση Πελάτη

Όταν ένας πελάτης C επιθυμεί να συνδεθεί στο δίκτυο LibraRing, αποστέλλει στο σημείο πρόσβασής του, S, ένα μήνυμα $ConnectClient(key(C),\ ip(C),\ id(S))$. Εάν το κλειδί $key(C)$ υπάρχει στον πίνακα CT του S, ο C μαρκάρεται ως συνδεδεμένος και του αποστέλλονται οι αποθηκευμένες ειδοποιήσεις. Εάν το κλειδί $key(C)$ δεν υπάρχει στον πίνακα CT, αυτό σημαίνει ότι ο S δεν ήταν το σημείο πρόσβασης του C την τελευταία φορά που αυτός συνδέθηκε στο δίκτυο, οπότε τον προσθέτει σύμφωνα με την περιγραφή του πρωτοκόλλου συμμετοχής.

3.2.4 Σύνδεση/Αποσύνδεση Παρόχου

Οι πάροχοι χρησιμοποιούν ένα παρόμοιο πρωτόκολλο με αυτό των πελατών για να συνδεθούν/αποσυνδεθούν στο/από το δίκτυο LibraRing.

3.2.5 Ευρετηριασμός Πόρων

Ένας πόρος, όπως, για παράδειγμα μία δημοσίευση, ευρετηριάζεται σε τρία βήματα.

Βήμα 1: Ο πάροχος P κατασκευάζει μία δημοσίευση $p = \{\ (A_1,\ s_1), (A_2,\ s_2), \ldots, (A_n,\ S_n)\ \}$ που αποτελεί την περιγραφή του πόρου και αποστέλλει το μήνυμα $PubResource(key(P),\ ip(P),\ key(p),\ p)$ στο σημείο πρόσβασής του S.

Βήμα 2: Ο S προωθεί το p στους κατάλληλους υπερ-κόμβους σύμφωνα με τα επόμενα. Έστω ότι $D_1,\ D_2,\ \ldots,\ D_n$ είναι τα σύνολα που αποτελούνται μόνο από τις διαφορετικές λέξεις στα $s_1,\ s_2,\ \ldots,\ s_n$ αντίστοιχα. Τότε το p αποστέλλεται σε όλους τους υπερ-κόμβους με αναγνωριστικά που βρίσκονται στη λίστα $L = \{\ H(w_j) :\ w_j \in D_1 \cup \cdots \cup D_n\ \}$[1]. Το πρωτόκολλο κατάθεσης εγγυάται ότι το σύνολο L θα είναι ένα υπερσύνολο των αναγνωριστικών των υπερ-κόμβων που είναι υπεύθυνοι για επερωτήσεις που ταιριάζουν με το p. Στη συνέχεια, ο S αφαιρεί τα διπλότυπα αναγνωριστικά και ταξινομεί την λίστα L κατά αύξουσα σειρά. Με αυτόν τον τρόπο το L περιέχει λιγότερα αναγνωριστικά από τις διπλότυπες λέξεις του συνόλου $D_1 \cup \cdots \cup D_n$, αφού κάθε υπερ-κόμβος δύναται να

[1] Στην πραγματικότητα, όπως έχει γίνει κατανοητό και σε προηγούμενες ενότητες, υπάρχει η περίπτωση πολλά από τα αναγνωριστικά της λίστας L να μην αντιστοιχούν σε αναγνωριστικά υπαρκτών συνδεδεμένων κόμβων. Σε αυτή την περίπτωση αυτό που γίνεται είναι να αποστέλλεται το μήνυμα στον πλησιέστερο κόμβο ως προς το αναγνωριστικό αυτό.

είναι υπεύθυνος για περισσότερες από μία λέξεις που περιέχονται στο έγγραφο προς δημοσίευση. Έχοντας υπολογίσει το L, ο S ευρετηριάζει το p δημιουργώντας το μήνυμα $msg = IndexResource(ip(P),\ key(P),\ ip(S),\ key(p),\ p)$ και αποστέλλοντάς το στους κατάλληλους κόμβους με τη συνάρτηση $send(msg,\ L)$.

Βήμα 3: Τέλος, κάθε υπερ-κόμβος S' που παραλαμβάνει το μήνυμα msg αποθηκεύει το p σε ένα ανεστραμμένο ευρετήριο[2] το οποίο θα διευκολύνει το ταίριασμά τους με τις στιγμιαίες επερωτήσεις που θα παραλάβει αργότερα ο S' από έναν πελάτη.

3.2.6 Κατάθεση Στιγμιαίων Επερωτήσεων

Σε αυτή την υποενότητα θα δείξουμε πώς μπορούν να απαντηθούν στιγμιαίες επερωτήσεις που περιέχουν Boolean και διανυσματικού χώρου τμήματα (δηλώνονται ως επερωτήσεις τύπου $T1$) ή μόνο διανυσματικού χώρου τμήματα (δηλώνονται ως επερωτήσεις τύπου $T2$). Ο πρώτος τύπος επερωτήσεων ($T1$) ευρετηριάζεται πάντα χρησιμοποιώντας το Boolean τμήμα του. Ας υποθέσουμε ότι ένας πελάτης C επιθυμεί να καταθέσει μία επερώτηση q (τύπου $T1$) που έχει τη μορφή

$$\bigwedge_{i=1}^{m} A_i = s_i \ \wedge \bigwedge_{i=m+1}^{n} A_i \sqsupseteq wp_i \ \wedge \bigwedge_{i=n+1}^{k} A_i \ \sim_{a_i} \ s_i$$

Τα επόμενα τρία βήματα παίρνουν μέρος:

Βήμα 1: Ο C αποστέλλει στο σημείο πρόσβασής του, S, το μήνυμα $SubmitQ(key(C),\ ip(C),\ key(q),\ q)$.

Βήμα 2: Ο S επιλέγει τυχαία μία και μόνο λέξη w που περιέχεται σε κάποιο από τις λεκτικές τιμές $s_1,\ s_2,\ \ldots,\ s_m$ ή λεκτικά σχήματα $wp_{m+1},\ \ldots,\ wp_n$ και υπολογίζει την τιμή $H(w)$ έτσι ώστε να αποκτήσει το αναγνωριστικό του υπερ-κόμβου που είναι υπεύθυνος για την αποθήκευση της δημοσίευσης που μπορεί να ταιριάζει με το q. Στη συνέχεια αποστέλλει το μήνυμα $msg = PoseQuery(ip(C),\ key(C),\ ip(S),\ key(q),\ q)$ σε αυτόν τον κόμβο καλώντας τη συνάρτηση $send(msg,\ H(w))$.

Εάν η επερώτηση q είναι της μορφής $A_{n+1} \ \sim_{a_1} \ s_1 \wedge \ldots \wedge A_n \ \sim_{a_n} \ s_n$ (επερώτηση τύπου $T2$) τότε το δεύτερο βήμα τροποποιείται ως εξής: Έστω $D_1,\ D_2,\ \ldots,\ D_n$ τα σύνολα που

[2]Αυτή η δομή αποθήκευσης παρέχεται από τον ΚΠΚ.

αποτελούνται μόνο από τις διαφορετικές λέξεις στα $s_1,\ s_2,\ \ldots,\ s_n$ αντίστοιχα. Τότε το q θα πρέπει να αποσταλεί σε όλους τους υπερ-κόμβους με αναγνωριστικά που ανήκουν στη λίστα $L = \{\ H(w_j) :\ \ w_j \in D_1 \cup \cdots \cup D_n\ \}$. Για να επιτευχθεί κάτι τέτοιο, ο S αφαιρεί τα διπλότυπα αναγνωριστικά από το L, ταξινομεί την L κατά αύξουσα σειρά και αποστέλλει το μήνυμα $msg = PoseQuery(ip(C),\ key(C),\ ip(S),\ key(q),\ q)$ στους κατάλληλους κόμβους με τη συνάρτηση $send(msg,\ L)$.

Βήμα 3: Κάθε υπερ-κόμβος που παραλαμβάνει μία στιγμιαία επερώτηση την ταιριάζει με τις τοπικές αποθηκευμένες δημοσιεύσεις του, έτσι ώστε να πληροφορηθεί για τους παρόχους που έχουν δημοσιοποιήσει έγγραφα που ταιριάζουν με την επερώτηση q και διανέμει αυτές τις απαντήσεις σύμφωνα με την περιγραφή της υποενότητας 3.2.8.

3.2.7 Λειτουργικότητα Δημοσίευσης / Συνδρομής

Αυτή η υποενότητα περιγράφει τον τρόπο με τον οποίο μπορούν να επεκταθούν οι υποενότητες 3.2.5 και 3.2.6 έτσι ώστε να προσφερθεί η λειτουργικότητα της Δημοσίευσης / Συνδρομής.

Για τον ευρετηριασμό μιας επερώτησης διαρκείας, cq, το πρωτόκολλο κατάθεσης μίας στιγμιαίας επερώτησης θα πρέπει να τροποποιηθεί. Τα δύο πρώτα βήματα είναι ακριβώς ίδια, ενώ το τρίτο έχει ως εξής. Κάθε υπερ-κόμβος που παραλαμβάνει ένα cq, το αποθηκεύει σε μία τοπική δομή δεδομένων για επερωτήσεις διαρκείας έτσι ώστε να τις ταιριάζει με τις αφίξεις νέων δημοσιεύσεων. Ένας υπερ-κόμβος S χρησιμοποιεί ένα πίνακα κατακερματισμού για να ευρετηριάσει όλες τις ατομικές επερωτήσεις cq, χρησιμοποιώντας ως κλειδιά τα γνωρίσματα $A_1,\ A_2,\ \ldots,\ A_k$. Για τον ευρετηριασμό κάθε ατομικής επερώτησης, χρησιμοποιούνται τρεις διαφορετικές δομές δεδομένων:

- ένας πίνακας κατακερματισμού για τις λεκτικές τιμές $s_1,\ s_2,\ \ldots,\ s_m$,
- ένα *προθεματικό δέντρο* (prefix tree - trie), το οποίο αποθηκεύει τις κοινές λέξεις στα λεκτικά σχήματα $wp_{m+1},\ \ldots,\ wp_n$, και
- έναν ανεστραμμένο πίνακα (ή πίνακα κατακερματισμού) για την αποθήκευση των πιο "σημαντικών" λέξεων από τις λεκτικές τιμές $s_{n+1},\ \ldots,\ s_k$.

Ο υπερ-κόμβος S' (αυτός του βήματος 3, δηλαδή, ο υπερ-κόμβος στον οποίο τελικά φθάνει η επερώτηση διαρκείας για να απαντηθεί) χρησιμοποιεί τις παραπάνω δομές δεδομένων για

να φιλτράρει τις επερωτήσεις διαρκείας και να βρει γρηγορότερα αυτές που ταιριάζουν με κάποια νέα δημοσίευση p. Αυτό γίνεται χρησιμοποιώντας έναν αλγόριθμο που συνδυάζει τους αλγορίθμους BestFitTrie [7] και SQI [33].

Για τον ευρετηριασμό ενός πόρου, δηλαδή μίας δημοσίευσης, το πρωτόκολλο της παραγράφου 3.2.5 χρειάζεται επέκταση. Τα δύο πρώτα βήματα είναι ακριβώς τα ίδια, ενώ στο τρίτο βήμα, κάθε υπερ-κόμβος που λαμβάνει το p, το ταιριάζει με τις τοπικές του επερωτήσεις διαρκείας χρησιμοποιώντας τους αλγορίθμους BestFitTrie και SQI. Θα πρέπει να σημειωθεί ότι στα πλαίσια της υλοποίησης της αρχιτεκτονικής του συστήματος LibraRing δεν προσφέρεται η λειτουργία του φιλτραρίσματος. Ως εκ τούτου, οι τρεις δομές που προαναφέρθηκαν, όπως και οι αλγόριθμοι BestFitTrie και SQI, τόσο για το φιλτράρισμα όσο και για το ταίριασμα που πραγματοποιείται στο βήμα 3, δεν χρησιμοποιούνται επίσης.

3.2.8 Παράδοση Ειδοποιήσεων

Ας υποθέσουμε ότι ένας υπερ-κόμβος, S, πρέπει να παραδώσει μία ειδοποίηση n για μία επερώτηση διαρκείας cq στον πελάτη C. Ο S δημιουργεί το μήνυμα $msg = Notification(ip($ $P)$, $key(P)$, $key(p)$, $key(cq))$, όπου ο P είναι ο πάροχος που δημοσίευσε τον πόρο p που ταιριάζει με το cq και το αποστέλλει στον C. Εάν ο C δεν είναι συνδεδεμένος στο δίκτυο LibraRing, τότε ο S αποστέλλει το msg στον S', όπου ο S' είναι το σημείο πρόσβασης του C, χρησιμοποιώντας την $ip(S')$ που είναι συσχετισμένη[3] με το cq. Ο S' αποθηκεύει το μήνυμα msg για να το παραδώσει στον C κατά την επανασύνδεσή του στο δίκτυο LibraRing. Οι απαντήσεις σε στιγμιαίες επερωτήσεις χειρίζονται με παρόμοιο τρόπο.

3.3 Συμπεράσματα

Σε αυτό το κεφάλαιο παρουσιάστηκε η αρχιτεκτονική LibraRing, όπως αυτή δίνεται στο [8]. Συγκεκριμένα, παρουσιάστηκαν τα δομικά στοιχεία του συστήματος LibraRing· οι υπερ-κόμβοι, οι πελάτες και οι πάροχοι. Παρουσιάστηκαν, επίσης, οι υπηρεσίες που αυτά υλοποιούν και τα πρωτόκολλα που ορίζουν τη μεταξύ τους επικοινωνία.

[3]Αυτή η συσχέτιση μπορεί να διαπιστωθεί από το μήνυμα που δημιουργείται και αποστέλλεται από τον υπερκόμβο S κατά το βήμα 2 του πρωτοκόλλου κατάθεσης στιγμιαίας επερώτησης ή επερώτησης διαρκείας. Για τη δημιουργία του, χρησιμοποιείται η συνάρτηση $PoseQuery(ip(C),\ key(C),\ ip(S),\ key(q),\ q)$, της οποίας το τρίτο όρισμα είναι πράγματι η διεύθυνση IP του σημείου πρόσβασης S του πελάτη C.

Κεφάλαιο 4

Περιγραφή της Υλοποίησης της Αρχιτεκτονικής LibraRing

Σε αυτή την ενότητα θα παρουσιαστούν διάφορα σημαντικά θέματα υλοποίησης της αρχιτεκτονικής του συστήματος LibraRing. Η σειρά με την οποία θα παρουσιαστούν είναι όμοια με τη σειρά παρουσίασης της προηγούμενης ενότητας, της περιγραφής της αρχιτεκτονικής LibraRing. Έτσι, στην Ενότητα 4.1 περιγράφεται η υλοποίηση των προσφερόμενων υπηρεσιών, ενω στην Ενότητα 4.2 περιγράφεται η υλοποίηση των προσφερόμενων πρωτοκόλλων και παράλληλα παρουσιάζονται μερικοί επιλεγμένοι αλγόριθμοι που βοηθούν στην κατανόησή τους.

Για την υλοποίηση του συστήματος LibraRing χρησιμοποιήθηκε η γλώσσα προγραμματισμού Java σε συνδυασμό με τη τεχνολογία του ΚΠΚ Pastry που προσφέρεται από το πακέτο FreePastry[1]. Το FreePastry είναι και αυτό γραμμένο στη γλώσσα προγραμματισμού Java. Παράλληλα, χρησιμοποιείται η τεχνολογία XML για την αναπαράσταση των δεδομένων, όπως επερωτήσεις, δημοσιεύσεις και ειδοποιήσεις, όπως περιγράφηκε και στην Ενότητα 2.6.

Εδώ θα πρέπει να σημειώσουμε ότι υπάρχουν δύο λειτουργίες της αρχιτεκτονικής LibraRing — η πολλαπλή αποστολή μηνυμάτων από έναν υπερ-κόμβο σε άλλους και η δυνατότητα αποχώρησης / συμμετοχής νέων υπερ-κόμβων — που δεν έχουν υλοποιηθεί. Και οι δύο αναφέρονται σε επικοινωνία μεταξύ των υπερ-κόμβων, δηλαδή των κόμβων που αναλαμβάνουν να εκτελέσουν και να προσφέρουν τη διεπαφή των κατανεμημένων πινάκων κατακερματισμού και είναι λογικό διαφορετικά συστήματα που υλοποιούν αυτή τη διεπαφή να διαφέρουν

[1] `http://freepastry.rice.edu/FreePastry/`

σε σχεδιαστικά θέματα. Δεδομένου, επίσης, ότι η αρχιτεκτονική LibraRing έχει περιγραφεί βάσει του ΚΠΚ που υλοποιείται από το σύστημα Chord, ενώ η υλοποίησή του έγινε χρησιμοποιώντας το σύστημα Pastry, κάτι τέτοιο είναι απολύτως λογικό.

Πιο συγκεκριμένα, το σχεδιαστικό θέμα του Chord, το οποίο δεν υιοθετείται και από το Pastry είναι η έννοια του successor, predecessor κόμβου στο δακτύλιο αναγνωριστικών του Chord σε συνδυασμό με τη δομή του πίνακα δεικτών[2]. Στην πρώτη λειτουργία, δηλαδή στην αποστολή του ίδιου μηνύματος από έναν υπερ-κόμβο σε πολλούς[3] ο αποστολέας υπερ-κόμβος θα κατασκεύαζε μία ταξινομημένη λίστα, σε αύξουσα σειρά ως προς τα αναγνωριστικά των παραληπτών υπερ-κόμβων, και θα έστελνε το μήνυμα στον πρώτο από αυτούς. Ο παραλήπτης θα έλεγχε αν είναι υπεύθυνος για αυτό το μήνυμα και σε αρνητική απάντηση θα προωθούσε το μήνυμα στον πρώτο κόμβο της λίστας συμβουλευόμενος τον πίνακα δεικτών του. Σε θετική απάντηση θα αφαιρούσε από τη λίστα τα αναγνωριστικά για τα οποία θα ήταν υπεύθυνος, κρατώντας επίσης και το ίδιο το μήνυμα και θα προωθούσε και πάλι το μήνυμα στον πρώτο κόμβο της λίστας, αν αυτός υπήρχε, συμβουλευόμενος πάντα τον πίνακα δεικτών του. Αυτή η διαδικασία αποτελεί μία βελτιστοποίηση για την πολλαπλή αποστολή ενός μηνύματος σε περισσότερους από έναν υπερ-κόμβους, αφού με τη χρήση της μειώνονται σε σημαντικό βαθμό ο αριθμός των hops που χρειάζεται για να φτάσει το μήνυμα σε όλους τους παραλήπτες.

Το ίδιο σχεδιαστικό θέμα του Chord εμποδίζει την ακριβή υλοποίηση της λειτουργίας σύνδεσης / αποσύνδεσης ενός υπερ-κόμβου[4]. Σύμφωνα με αυτή, κατά τη σύνδεση ενός υπερ-κόμβου, αυτός θα πρέπει να γίνει ο predecessor ενός κόμβου n και ο successor του πρώην predecessor του κόμβου n. Στη συνέχεια ο νεο-συνδεδεμένος κόμβος θα πρέπει να παραλάβει όλα τα δεδομένα για τα οποία είναι υπεύθυνος από τον κόμβο n και να ενημερώσει κατάλληλα το δίκτυο για την ύπαρξή του. Μία παρόμοια διαδικασία ενημέρωσης του successor και predecessor και αποστολής των δεδομένων γίνεται και όταν ένας υπερ-κόμβος χρειάζεται να αποχωρήσει από το δίκτυο.

[2]Ανατρέξτε στην παράγραφο 2.2.3 για την έννοιά τους.

[3]Περιγράφεται στην παράγραφο 6 του [8] ως mutlisend().

[4]Περιγράφεται στην παράγραφο 7.7 του [8].

4.1 Υλοποίηση Προσφερόμενων Υπηρεσιών

4.1.1 Υπερ-Κόμβοι (Super-Peers)

Η "ζωή" ενός υπερ-κόμβου συνίσταται στην εκτέλεση μίας όμοιας, αέναης διαδικασίας. Από τη στιγμή της δημιουργίας του κάθε τέτοιος κόμβος καλεί την συνάρτηση $run()$, η οποία αυτό που κάνει είναι να αναγκάζει έναν υπερ-κόμβο να δέχεται ανελλιπώς αιτήσεις από έναν πελάτη ή έναν πάροχο. Η υλοποίηση αυτής της διαδικασίας έχει γίνει με τη χρήση *νημάτων* (threads), τα οποία προσφέρονται από την Java μέσω της κλάσης Thread. Αυτό σημαίνει, ότι κάθε υπερ-κόμβος έχει τη δυνατότητα να επεξεργάζεται περισσότερες από μία αιτήσεις (είτε από πελάτη, είτε από πάροχο) ψευδοπαράλληλα ή καλύτερα σύγχρονα. Σε αυτή τη συνάρτηση εκτελείται ουσιαστικά η *χειραψία* (handshake) μεταξύ των δύο οντοτήτων για την εξυπηρέτηση της αίτησης του πελάτη[5] από τον υπερ-κόμβο. Αυτή η αίτηση μπορεί να αναφέρεται σε πέντε ενέργειες που μπορεί να χρειάζεται ο πελάτης-client και σε τέσσερις για τον πάροχο και η κάθε μία αντιστοιχεί σε μία συνάρτηση που θα κληθεί από τον υπερ-κόμβο για την εξυπηρέτησή τους. Αυτές μπορεί να είναι οι παρακάτω:

Πελάτης

- *Συμμετοχή (Join)*

  ο υπερ-κόμβος καλεί την συνάρτηση $clientJoin(CLIENT)$. Εκτελεί τις απαραίτητες ενέργειες, όπως περιγράφονται στο πρωτόκολλο συμμετοχής.

- *Σύνδεση (Connect)*

  ο υπερ-κόμβος καλεί την συνάρτηση $clientConnect(CLIENT)$. Εκτελεί τις απαραίτητες ενέργειες, όπως περιγράφονται στο πρωτόκολλο σύνδεσης / αποσύνδεσης (δηλαδή, ενημερώνει τον πίνακα CT, μαρκάρει τον πελάτη ως συνδεδεμένο και πιθανώς του προωθεί τις αποθηκευμένες ειδοποιήσεις).

- *Αποσύνδεση (Disconnect)*

  ο υπερ-κόμβος καλεί την συνάρτηση $clientDisconnect(CLIENT)$. Μαρκάρει τον πελάτη ως αποσυνδεδεμένο στον πίνακα CT.

[5] Από τώρα και στη συνέχεια, όπου αναφέρεται η λέξη πελάτης στα πλαίσια της λειτουργίας ενός υπερ-κόμβου, αν δεν αναφέρεται ρητά, θα εννοείται είτε η οντότητα του *πελάτη* (client), είτε η οντότητα του *παρόχου* (provider) του συστήματος LibraRing.

- *Στιγμιαία Επερώτηση (One-time Query)*

  ο υπερ-κόμβος καλεί την συνάρτηση $clientQuery()$. Αναλαμβάνει να αποστείλει προς επεξεργασία την επερώτησή του πελάτη στους κατάλληλους υπερ-κόμβους, σύμφωνα με τις μεθόδους ευρετηριασμού των επερωτήσεων (όπως ορίζει το αντίστοιχο πρωτόκολλο — 3.2.6 Κατάθεση Στιγμιαίων Επερωτήσεων), καλώντας τις συναρτήσεις $clientQueryT1()$ (για επερώτηση τύπου $T1$) και $clientQueryT2()$ (για επερώτηση τύπου $T2$). Ο παραλήπτης υπερ-κόμβος επεξεργάζεται την επερώτηση που του θέτει ο πελάτης και του αποστέλλει άμεσα τις απαντήσεις που ταιριάζουν με τις δημοσιεύσεις που υπάρχουν αποθηκευμένες στο σύστημα LibraRing. Για την επίτευξη αυτού του σκοπού χρησιμοποιείται η συνάρτηση $lookup()$, η οποία ανακτά μία δημοσίευση από τον ΚΠΚ του Pastry, και η συνάρτηση $matchQuery()$ που απαντά αν μία δημοσίευση ικανοποιεί μία επερώτηση.

- *Επερώτηση Διαρκείας (Continuous Query)*

  ο υπερ-κόμβος καλεί την συνάρτηση $clientCQuery()$. Αναλαμβάνει να αποστείλει προς αποθήκευση την επερώτηση του πελάτη στους κατάλληλους υπερ-κόμβους, σύμφωνα με τις μεθόδους ευρετηριασμού των επερωτήσεων διαρκείας (όπως ορίζει το αντίστοιχο πρωτόκολλο — 3.2.7 Λειτουργικότητα Δημοσίευσης / Συνδρομής). Ο παραλήπτης υπερ-κόμβος αποθηκεύει την επερώτηση σε έναν τοπικό πίνακα (continuousQuery), έτσι ώστε κατά την εισαγωγή μίας νέας δημοσίευσης, που θα εισαχθεί από αυτόν τον υπερ-κόμβο, στο PAST, να ελέγξει την ικανοποίησή της. Ο έλεγχος της ικανοποίησης υλοποιείται με την συνάρτηση $checkClientSubscription()$. Σε περίπτωση ικανοποίησης κάποιας αποθηκευμένης επερώτησης διαρκείας (αυτό ελέγχεται και πάλι με την συνάρτηση $matchQuery()$), μία ειδοποίηση δημιουργείται ($Notification()$) και αποστέλλεται στον πελάτη καλώντας τη συνάρτηση $notifyContinuousQuery()$. Αυτή η συνάρτηση ελέγχει, αρχικά, αν ο πελάτης είναι ακόμα συνδεδεμένος στο LibraRing και σε θετική απάντηση του αποστέλλει την ειδοποίηση. Σε διαφορετική περίπτωση, αναλαμβάνει να προωθήσει την ειδοποίηση στο σημείο πρόσβασης του πελάτη, ο οποίος τελικά την αποθηκεύει στον πίνακα CT. Σημειώνεται, ότι το σημείο πρόσβασης αναλαμβάνει να αφαιρέσει τα διπλότυπα των ειδοποιήσεων — αυτό δύναται να προκύψει από το γεγονός ότι η ίδια δημοσίευση και άρα η ίδια επερώτηση διαρκείας μπορεί να αποθηκευτεί σε περισσότερους από έναν υπερ-κόμβους — έτσι ώστε ο πελάτης να μην λάβει περισσότερες από μία ειδοποιήσεις για την ίδια δημοσίευση.

Πάροχος

- *Συμμετοχή (Join)*

  ο υπερ-κόμβος καλεί την συνάρτηση $clientJoin(PROVIDER)$. Πράττει ό,τι και στην περίπτωση του πελάτη.

- *Σύνδεση (Connect)*

  ο υπερ-κόμβος καλεί την συνάρτηση $clientConnect(PROVIDER)$. Πράττει ό,τι και στην περίπτωση του πελάτη.

- *Αποσύνδεση (Disconnect)*

  ο υπερ-κόμβος καλεί την συνάρτηση $clientDisconnect(PROVIDER)$. Πράττει ό,τι και στην περίπτωση του πελάτη.

- *Δημοσίευση (Publish)*

  ο υπερ-κόμβος καλεί την συνάρτηση $providerPublish()$. Αναλαμβάνει να αποστείλει προς αποθήκευση την περιγραφή της δημοσίευσης, που λαμβάνει από τον πάροχο, στους κατάλληλους υπερ-κόμβους, σύμφωνα με τις μεθόδους ευρετηριασμού των δημοσιεύσεων (όπως ορίζει το αντίστοιχο πρωτόκολλο — 3.2.5 Ευρετηριασμός Πόρων). Κάθε παραλήπτης/υπερ-κόμβος, δημιουργεί μία δημοσίευση ($Publication()$) βάσει της περιγραφής της και στη συνέχεια την εισάγει στον ΚΠΚ καλώντας τη συνάρτηση $insert()$. Τέλος, ελέγχει αν ικανοποιούνται οι τοπικές αποθηκευμένες επερωτήσεις διαρκείας και προωθεί τις κατάλληλες ειδοποιήσεις στους πελάτες σύμφωνα με τον τρόπο που περιγράφτηκε παραπάνω στην Επερώτηση Διαρκείας των πελατών.

4.1.2 Πελάτες (Clients)

Η λειτουργία ενός πελάτη δεν είναι προκαθορισμένη με την έννοια που είναι για έναν υπερ-κόμβο, που αναμένει να αποδεχθεί και να εξυπηρετήσει αιτήσεις από πελάτες. Παρ' όλα αυτά, η λειτουργία του είναι συγκεκριμένη και υποδεικνύεται ρητά από την αρχιτεκτονική LibraRing και τα πρωτόκολλά της. Έτσι, ένας πελάτης μπορεί να συμμετάσχει στο δίκτυο LibraRing, μέσω του πρωτοκόλλου συμμετοχής (καλώντας την συνάρτηση $newClientJoin()$), να συνδεθεί ή να αποσυνδεθεί από αυτό μέσω του πρωτοκόλλου σύνδεσης / αποσύνδεσης (καλώντας τη συνάρτηση $connectClient()$ / $disconnectClient()$ αντίστοιχα) και να δημιουργήσει μία επερώτηση στιγμιαία ή διαρκείας (καλώντας τη συνάρτηση $newQuery()$) καταθέτοντάς

την σε έναν υπερ-κόμβο (καλώντας τη συνάρτηση $submitQuery()$). Επιπροσθέτως, κατά τη δημιουργία ενός κόμβου-πελάτη αυτός αναμένει αιτήσεις από έναν υπερ-κόμβο. Αυτό γίνεται για την παραλαβή πιθανών ειδοποιήσεων από το σημείο πρόσβασής του, που είχαν δημιουργηθεί κατά το διάστημα που ο πελάτης ήταν αποσυνδεδεμένος.

Ως επέκταση της αρχιτεκτονικής LibraRing έχει προστεθεί η λειτουργικότητα της απόκτησης ενός πόρου — δημοσίευσης — από τον πάροχο που τον δημοσίευσε. Για να γίνει κάτι τέτοιο, τη στιγμή που ο πελάτης λαμβάνει μία ειδοποίηση ή μία απάντηση από μία στιγμιαία επερώτηση, αιτεί από τον πάροχο κάθε πόρου τη δημοσίευση. Αυτό γίνεται μέσω των συναρτήσεων $connectToProvider()$ και $providerRequest()$ και προϋποθέτει ότι ο πάροχος θα είναι συνδεδεμένος και θα αναμένει την εξυπηρέτηση τέτοιων αιτήσεων στην κατάλληλη διεύθυνση και πόρτα (ή θύρα) που είχε ορίσει στην περιγραφή αυτού του πόρου που δημοσίευσε στο δίκτυο LibraRing.

4.1.3 Πάροχοι (Providers)

Όμοια, η λειτουργία ενός παρόχου δεν είναι προκαθορισμένη με την έννοια που είναι για έναν υπερ-κόμβο, που αναμένει να αποδεχθεί και να εξυπηρετήσει αιτήσεις από πελάτες. Παρ' όλα αυτά, η λειτουργία του είναι συγκεκριμένη και υποδεικνύεται ρητά από την αρχιτεκτονική LibraRing και τα πρωτόκολλά της. Έτσι, ένας πάροχος μπορεί να συμμετάσχει στο δίκτυο LibraRing, μέσω του πρωτοκόλλου συμμετοχής (καλώντας την συνάρτηση $newProviderJoin()$), να συνδεθεί ή να αποσυνδεθεί από αυτό μέσω του πρωτοκόλλου σύνδεσης / αποσύνδεσης (καλώντας τη συνάρτηση $connectProvider()$ / $disconnectProvider()$ αντίστοιχα) και να δημιουργήσει έναν πόρο/μία δημοσίευση (με τη συνάρτηση $newPublication()$) καταθέτοντάς τον/την σε έναν υπερ-κόμβο (καλώντας τη συνάρτηση $pubResource()$).

Σύμφωνα με την επέκταση της αρχιτεκτονικής LibraRing όσον αφορά στην απόκτηση ενός πόρου από έναν πελάτη, κάθε πάροχος, κατά τη δημιουργία του, φροντίζει να δέχεται και να επεξεργάζεται αιτήσεις από πελάτες.

4.2 Υλοποίηση Προσφερόμενων Πρωτοκόλλων

Σε αυτή την ενότητα θα παρουσιαστούν:

1. ο τρόπος με τον οποίο πραγματοποιείται η χειραψία μεταξύ των πελατών και των υπερ-κόμβων σε κάθε πρωτόκολλο και

2. επιλεγμένοι αλγόριθμοι που επιτυγχάνουν την λειτουργικότητα του συστήματος LibraRing (από την πλευρά των υπερ-κόμβων, οι οποίοι αποτελούν την καρδιά του συστήματος LibraRing).

Σημειώνεται ότι η επικοινωνία μεταξύ των πελατών και των υπερ-κόμβων, δηλαδή η αμφίδρομη αποστολή δεδομένων για την εξυπηρέτηση των διαφόρων αιτήσεων και λειτουργιών, όπως αυτές ορίζονται στα πρωτόκολλα του Κεφαλαίου 3, πραγματοποιείται με τη χρήση των πρωτοκόλλων TCP/IP και με χρήση *υποδοχών* (sockets). Όσον αφορά στα δεδομένα που ανταλλάσσονται μεταξύ αυτών των δύο τύπων οντοτήτων, πρέπει να τονιστεί ότι αποστέλλονται ως *πρωταρχικές δομές δεδομένων* (primitive data types) των αντίστοιχων *αντικειμένων* (objects) της Java. Για να γίνει κάτι τέτοιο, η Java τα *σειρικοποιεί*[6] (serialise) στο ένα άκρο και τα *αποσειρικοποιεί*[7] (deserialise) στο άλλο. Αυτό έχει την επίπτωση της εξαρτημένης υλοποίησης των υπερ-κόμβων και των πελατών, η οποία στη συγκεκριμένη περίπτωση, δεν είναι δυνατή σε άλλη γλώσσα προγραμματισμού εκτός της Java.

Στη συνέχεια θα χρησιμοποιηθούν οι συναρτήσεις που εισήχθηκαν στην υποενότητα των πρωτοκόλλων. Για την αναφορά στους υπερ-κόμβους, πελάτες και παρόχους θα χρησιμοποιούνται τα γράμματα S, C και P αντίστοιχα, ενώ για την αναφορά σε επερωτήσεις στιγμιαίες, διαρκείας, δημοσιεύσεις και ειδοποιήσεις τα q, cq, p και n αντίστοιχα. Κάθε γραμμή αναπαριστά την σειρικοποίηση και αποστολή ενός αντικειμένου δεδομένων (π.χ. ενός αλφαριθμητικού όπως το πρώτο και τελευταίο μήνυμα που αποστέλλεται από τους εμπλεκόμενους στην επικοινωνία για τη σηματοδοσία της έναρξης και σωστής λήξης της, ενός ακεραίου όπως το κλειδί ενός πελάτη ή παρόχου, της διεύθυνσης IP, μίας δημοσίευσης, μίας επερώτησης, κ.τλ.). Η οντότητα που παραλαμβάνει τα δεδομένα τα διαβάζει με την ίδια σειρά και με τον αντίστοιχο τύπο τους και προβαίνει στις ανάλογες ενέργειες, όπως αυτές έχουν περιγραφεί στα πρωτόκολλα του Κεφαλαίου 3.

[6]Με τη λειτουργία της σειρικοποίησης εννοείται η κωδικοποίηση μίας οποιαδήποτε δομής δεδομένων σε μία πρωταρχική μορφή τύπων δεδομένων (π.χ. χαρακτήρες) έτσι ώστε να ενσωματώνει όλες τις πληροφορίες αυτής της δομής δεδομένων και να μπορεί να αποθηκευτεί σε ένα μέσο ή ακόμα και να αποσταλλεί μέσω ενός δικτύου.

[7]Με τη λειτουργία της αποσειρικοποίησης εννοείται η αντίστροφη διαδικασία της σειρικοποίησης, δηλαδή η ανάκτηση του πλήρους σειρικοποιημένου αντικειμένου.

4.2.1 Συμμετοχή Πελάτη

Σύμφωνα με την παράγραφο 3.2.1 Συμμετοχή Πελάτη, για να συμμετάσχει ένας πελάτης στο δίκτυο LibraRing θα πρέπει να βρει με κάποιο τρόπο έναν υπερ-κόμβο και να του στείλει ένα μήνυμα συμμετοχής, μαζί με το κλειδί του και τη διεύθυνσή IP του. Σε επίπεδο υλοποίησης αυτό το μήνυμα συνίσταται στην αποστολή των παρακάτω δεδομένων, ώστε να επιτευχθεί η χειραψία του πελάτη με τον υπερ-κόμβο.

1. $Client\ Join\backslash n$
2. $key(C)$
3. $ip(C)$

Στη συνέχεια αναμένει την επιβεβαίωση από τον υπερ-κόμβο, η οποία πρέπει να είναι η

$$SuperPeer\ Join\backslash n$$

4.2.2 Συμμετοχή Παρόχου

Σύμφωνα με την παράγραφο 3.2.2 Συμμετοχή Παρόχου, για να συμμετάσχει ένας πάροχος στο δίκτυο LibraRing θα πρέπει να βρει με κάποιο τρόπο έναν υπερ-κόμβο και να του στείλει ένα μήνυμα συμμετοχής, μαζί με το κλειδί του και τη διεύθυνσή IP του. Σε επίπεδο υλοποίησης αυτό το μήνυμα συνίσταται στην αποστολή των παρακάτω δεδομένων, ώστε να επιτευχθεί η χειραψία του παρόχου με τον υπερ-κόμβο.

1. $Provider\ Join\backslash n$
2. $key(P)$
3. $ip(P)$

Στη συνέχεια αναμένει την επιβεβαίωση από τον υπερ-κόμβο, η οποία πρέπει να είναι η

$$SuperPeer\ Join\backslash n$$

4.2.3 Σύνδεση Πελάτη

Σύμφωνα με την παράγραφο 3.2.3 Σύνδεση/Αποσύνδεση Πελάτη, για να συμμετάσχει ένας πελάτης στο δίκτυο LibraRing θα πρέπει να στείλει στον υπερ-κόμβο που αποτελεί το σημείο πρόσβασής του ένα μήνυμα σύνδεσης, μαζί με το κλειδί του και τη διεύθυνσή IP του. Σε επίπεδο υλοποίησης αυτό το μήνυμα συνίσταται στην αποστολή των παρακάτω δεδομένων, ώστε να επιτευχθεί η χειραψία του πελάτη με τον υπερ-κόμβο.

1. $Client\ Connect\backslash n$
2. $key(C)$
3. $ip(C)$

Στη συνέχεια αναμένει την επιβεβαίωση από τον υπερ-κόμβο, η οποία πρέπει να είναι η

$$SuperPeer\ Connected\backslash n$$

4.2.4 Σύνδεση Παρόχου

Σύμφωνα με την παράγραφο 3.2.4 Σύνδεση/Αποσύνδεση Πελάτη, για να συμμετάσχει ένας πάροχος στο δίκτυο LibraRing θα πρέπει να στείλει στον υπερ-κόμβο που αποτελεί το σημείο πρόσβασής του ένα μήνυμα σύνδεσης, μαζί με το κλειδί του και τη διεύθυνσή IP του. Σε επίπεδο υλοποίησης αυτό το μήνυμα συνίσταται στην αποστολή των παρακάτω δεδομένων, ώστε να επιτευχθεί η χειραψία του παρόχου με τον υπερ-κόμβο.

1. $Provider\ Connect\backslash n$
2. $key(P)$
3. $ip(P)$

Στη συνέχεια αναμένει την επιβεβαίωση από τον υπερ-κόμβο, η οποία πρέπει να είναι η

$$SuperPeer\ Connected\backslash n$$

4.2.5 Αποσύνδεση Πελάτη

Η διαδικασία χειραψίας του πελάτη με τον υπερ-κόμβο, που αποτελεί το σημείο πρόσβασής του, ωστε ο πρώτος να αποσυνδεθεί από το δίκτυο LibraRing είναι όμοια με αυτή της Σύνδεσης Πελάτη, αλλάζοντας μόνο τη λέξη Connect σε Disconnect και βέβαια αναμένοντας την αντίστοιχη επιβεβαίωση από τον υπερ-κόμβο.

1. $Client\ Disconnect\backslash n$
2. $key(C)$
3. $ip(C)$

Στη συνέχεια αναμένει την επιβεβαίωση από τον υπερ-κόμβο, η οποία πρέπει να είναι η

$$SuperPeer\ Disconnected\backslash n$$

4.2.6 Αποσύνδεση Παρόχου

Η διαδικασία χειραψίας του παρόχου με τον υπερ-κόμβο, που αποτελεί το σημείο πρόσβασής του, ωστε ο πρώτος να αποσυνδεθεί από το δίκτυο LibraRing είναι όμοια με αυτή της Σύνδεσης Παρόχου, αλλάζοντας μόνο τη λέξη Connect σε Disconnect και βέβαια αναμένοντας την αντίστοιχη επιβεβαίωση από τον υπερ-κόμβο.

1. $Provider\ Disconnect\backslash n$
2. $key(P)$
3. $ip(P)$

Στη συνέχεια αναμένει την επιβεβαίωση από τον υπερ-κόμβο, η οποία πρέπει να είναι η

$$SuperPeer\ Disconnected\backslash n$$

4.2.7 Ευρετηριασμός Πόρων (μόνο για Παρόχους)

Το πρωτόκολλο του ευρετηριασμού των πόρων από την μεριά ενός παρόχου σημαίνει τη δημιουργία και αποστολή μίας περιγραφής δημοσίευσης σε έναν υπερ-κόμβο. Σε αυτό το πλαίσιο, η χειραψία είναι η επόμενη:

1. $Provider\ Publish\backslash n$
2. $key(P)$
3. $ip(P)$
4. p

Στη συνέχεια αναμένει την επιβεβαίωση από τον υπερ-κόμβο, η οποία πρέπει να είναι η

$$SuperPeer\ Published\backslash n$$

Στη συνέχεια ακολουθεί ο αλγόριθμος ευρετηριασμού και αποθήκευσης μίας δημοσίευσης:

Αλγόριθμος ευρετηριασμού και αποθήκευσης δημοσιεύσεων

1. $String[]\ words\ =\ getDistinctWords(p)$
2. $Set\ ids$
3. **for every** $word$ **in** $words[]$ **do**
4. $\quad NodeId\ nodeId\ =$ **lookup(word)**[8]
5. $\quad put(nodeId,\ ids)$
6. **done**
7. **for every** $node$ **in** ids **do**

Συνεχίζεται στην επόμενη σελίδα

[8]Στο εξής κάθε διαδικασία που παρέχεται από το σύστημα FreePastry, όπως η $lookup()$, θα εμφανίζεται με γκρί και έντονη γραμματοσειρά.

Πίνακας 4.8 – συνέχεια από την προηγούμενη σελίδα

Αλγόριθμος ευρετηριασμού και αποθήκευσης δημοσιεύσεων

8. **if** $responsibleFor(id,\ myId)$ **then**
9. **insert(p)** // **procedure provided by DHT**
10. $add(key(p),\ myStoredPublications)$
11. **else**
12. **send(p, node)**
13. **done**
// Finally all super-peers that receive p store it in PAST calling
// insert(p) and add its key in their myStoredPublications table

Πίνακας 4.8: Αλγόριθμος ευρετηριασμού και αποθήκευσης δημοσιεύσεων.

4.2.8 Κατάθεση Στιγμιαίων Επερωτήσεων (μόνο για Πελάτες)

Η χειραψία που ακολουθείται από τους πελάτες προς τους υπερ-κόμβους κατά την κατάθεση στιγμιαίων επερωτήσεων είναι η ακόλουθη:

1. $Client\ Query\backslash n$
2. $key(C)$
3. $ip(C)$
4. q

Στη συνέχεια αναμένει την επιβεβαίωση από τον υπερ-κόμβο, η οποία πρέπει να είναι

$$SuperPeer\ QueryDelivered\backslash n$$

Σε αυτό το σημείο θα πρέπει να παρουσιαστεί και η χειραψία που πραγματοποιείται

μεταξύ του υπερ-κόμβου και του πελάτη για την αποστολή των απαντήσεων. Την επόμενη επικοινωνία την ξεκινάει ο υπερ-κόμβος.

1. $SuperPeer\ AnswerQuery\backslash n$
2. $key(q)$
3. $number$

Ως αυτού του σημείου, ο υπερ-κόμβος έχει ενημερώσει τον πελάτη ότι θα του στείλει $number$ στον αριθμό απαντήσεις, δηλαδή δημοσιεύσεις, που ικανοποιούν την στιγμιαία επερώτηση που του τέθηκε με κλειδί το $key(q)$. Στη συνέχεια, γίνεται μία επαναληπτική διαδικασία αποστολής των κλειδιών ($key(p)$) αυτών των δημοσιεύσεων για $number$ φορές. Τέλος, ο υπερ-κόμβος αναμένει την επιβεβαίωση από τον πελάτη για τη λήψη όλων των απαντήσεων, η οποία πρέπει να είναι η

$$Client\ NotificationDelivered\backslash n$$

Η χρήση της λέξης *Notification* στο μήνυμα της επιβεβαίωσης του πελάτη δεν θα πρέπει να συγχέεται με την έννοια της ειδοποίησης στις επερωτήσεις διαρκείας. Επιλέχθηκε, διότι αφενός αποτελεί μία ειδοποίηση (υπό την καθημερινή έννοια των πραγμάτων) και αφετέρου, διότι ο τρόπος χειραψίας είναι όμοιος με αυτόν που ακολουθείται στην παράδοση των ειδοποιήσεων για επερωτήσεις διαρκείας — 4.2.10 Παράδοση Ειδοποιήσεων (μόνο για Υπερ-κόμβους).

Στη συνέχεια ακολουθεί ο αλγόριθμος ευρετηριασμού και απάντησης μίας στιγμιαίας επερώτησης:

Αλγόριθμος ευρετηριασμού και απάντησης στιγμιαίων επερωτήσεων

1. **if** $q\ is\ of\ type\ T1$ **then**
2. $String\ word\ =\ getArbitraryWord(q)$
3. $NodeId\ id\ =$ **lookup(word)** // lookup() procedure of DHT
4. **if** $responsibleFor(id,\ myId)$ **then**

Συνεχίζεται στην επόμενη σελίδα

Πίνακας 4.11 – συνέχεια από την προηγούμενη σελίδα

Αλγόριθμος ευρετηριασμού και απάντησης στιγμιαίων επερωτήσεων

```
5.          Vector answers
6.          for every pubKey in myStoredPublications[] do
7.              Publication pub = lookup(pubKey)
8.              if match(q, pub) then
9.                  add(C, q, pubKey, answers)
10.         done
11.         sendAnswers(C, q, answers)
12.     else // this node is not responsible for answering q
13.         send(q, node)
14. else // q is of type T2
15.     String[] words = getDistinctWords(q)
16.     Set ids
17.     for every word in words[] do
18.         NodeId nodeId = lookup(word)
19.         put(nodeId, ids)
20.     done
21.     for every node in ids do
22.         if responsibleFor(id, myId) then
23.             do lines 5 − 11
24.         else
25.             send(q, node)
26.     done
    // Finally all super-peers that receive q answer it executing
    // the code of lines 5 - 11
```

Πίνακας 4.11: Αλγόριθμος ευρετηριασμού και απάντησης στιγμιαίων επερωτήσεων.

4.2.9 Κατάθεση Επερωτήσεων Διαρκείας (μόνο για Πελάτες)

Η χειραψία που ακολουθείται από τους πελάτες προς τους υπερ-κόμβους κατά την κατάθεση επερωτήσεων διαρκείας είναι η ακόλουθη:

1. $Client\ CQuery\backslash n$
2. $key(C)$
3. $ip(C)$
4. cq

Στη συνέχεια αναμένει την επιβεβαίωση από τον υπερ-κόμβο, η οποία πρέπει να είναι η

$$SuperPeer\ QueryDelivered\backslash n$$

Στη συνέχεια ακολουθεί ο αλγόριθμος ευρετηριασμού και αποθήκευσης μίας επερώτησης διαρκείας από έναν υπερ-κόμβο:

Αλγόριθμος ευρετηριασμού και αποθήκευσης επερωτήσεων διαρκείας

```
1.  if cq is of type T1 then
2.      String word = getArbitraryWord(cq)
3.      NodeId id = lookup(word) // lookup() procedure of DHT
4.      if responsibleFor(id, myId) then
5.          add(cq, myStoredContinuousQueries)
6.      else
7.          send(cq, node)
8.  else // cq is of type T2
9.      String[] words = getDistinctWords(cq)
10.     Set ids
```

Συνεχίζεται στην επόμενη σελίδα

Πίνακας 4.13 – συνέχεια από την προηγούμενη σελίδα

Αλγόριθμος ευρετηριασμού και αποθήκευσης επερωτήσεων διαρκείας

11. **for every** $word$ **in** $words[]$ **do**
12. $NodeId\ nodeId\ =$ **lookup(word)**
13. $put(nodeId,\ ids)$
14. **done**
15. **for every** $node$ **in** ids **do**
16. **if** $responsibleFor(id,\ myId)$ **then**
17. $add(cq,\ myStoredContinuousQueries)$
18. **else**
19. **send(cq, node)**
20. **done**
// Finally all super-peers that receive cq store it in
// their local myStoredContionuousQueries table.

Πίνακας 4.13: Αλγόριθμος ευρετηριασμού και αποθήκευσης επερωτήσεων διαρκείας.

4.2.10 Παράδοση Ειδοποιήσεων (μόνο για Υπερ-κόμβους)

Η ειδοποίηση για ικανοποιημένες επερωτήσεις διαρκείας αποστέλλονται στους πελάτες από τους υπερ-κόμβους. Η χειραψία που ακολουθείται είναι η επόμενη:

1. $SuperPeer\ AnswerCQuery\backslash n$
2. $key(cq)$
3. 1[9]
4. n

[9]Η αποστολή του αριθμού 1 σημαίνει ότι υπάρχει μία ειδοποίηση για μία δημοσίευση που ικανοποιεί μία επερώτηση διαρκείας. Προφανώς, αν υπάρχουν πολλές δημοσιεύσεις που ικανοποιούν την ίδια επερώτηση, θα γίνουν τόσες χειραψίες όσες και ο αριθμός τους.

Στη συνέχεια αναμένει την επιβεβαίωση από τον πελάτη, η οποία πρέπει να είναι η

$$Client\ NotificationDelivered\backslash n$$

Στη συνέχεια ακολουθεί ο αλγόριθμος δημιουργίας και αποστολής ειδοποίησης σε έναν πελάτη από έναν υπερ-κόμβο για κάποια ικανοποιημένη επερώτηση διαρκείας. Ο αλγόριθμος εκτελείται κατά την εισαγωγή μία νέας δημοσίευσης:

Αλγόριθμος δημιουργίας και αποστολής ειδοποιήσεων

```
1.  for every cq in myStoredContinuousQueries do
2.      if match(cq, p) then
3.          Notification n = createNotification(C, cq, key(p))
4.          if C isOnline() then
5.              notify(C, n)
6.          else
7.              send(cq, ap) // ap is the access point of C
8.  done
    // Finally, every access point that receives cq stores it
    // in its CT table and notifies C upon its reconnection
```

Πίνακας 4.15: Αλγόριθμος δημιουργίας και αποστολής ειδοποιήσεων.

4.3 Συμπεράσματα

Σε αυτό το κεφάλαιο παρουσιάστηκε η υλοποίηση της αρχιτεκτονικής LibraRing. Η υλοποίηση έγινε χρησιμοποιώντας τη διεπαφή του ΚΠΚ που προσφέρει το σύστημα Pastry μέσω του πακέτου υλοποίησής του, FreePastry. Τόσο η υλοποίηση του LibraRing όσο και

του FreePastry πραγματοποιήθηκε με την αντικειμενοστραφή γλώσσα προγραμματισμού Java. Παράλληλα, χρησιμοποιήθηκε η γλώσσα XML για την αναπαράσταση των δεδομένων που διαχειρίζεται το LibraRing, όπως δημοσιεύσεις, στιγμιαίες επερωτήσεις και επερωτήσεις διαρκείας. Επίσης, παρουσιάσθηκε ο τρόπος επικοινωνίας/χειραψίας των υπερ-κόμβων με τους πελάτες και τους παρόχους. Πιο συγκεκριμένα, δώθηκαν τα ακριβή μηνύματα και η σειρά αυτών που πρέπει ένας κόμβος να αποστείλει σε έναν άλλο ώστε να εξυπηρετηθεί η αίτησή του. Τέλος, παρουσιάσθηκαν επιλεγμένοι αλγόριθμοι που περιγράφουν τον τρόπο παράδοσης ειδοποιήσεων, ευρετηριασμού και αποθήκευσης δημοσιεύσεων/επερωτήσεων διαρκείας και ευρετηριασμού και απάντησης στιγμιαίων επερωτήσεων.

Κεφάλαιο 5

Επίλογος

5.1 Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Ξεκινήσαμε την παρουσίασή μας, περιγράφοντας το απαραίτητο θεωρητικό υπόβαθρο για την κατανόηση του αντικειμένου αυτής της πτυχιακής εργασίας. Συγκεκριμένα, αναφερθήκαμε στις τεχνολογίες των δικτύων ομότιμων κόμβων και στους κατανεμημένους πίνακες κατακερματισμού. Παρουσιάσαμε δύο πολύ γνωστά συστήματα που υλοποιούν την τεχνολογία του κατανεμημένου πίνακα κατακερματισμου, τα Chord και Pastry. Ειδικά για το δεύτερο, δώθηκε μία αναλυτική περιγραφή καθώς χρησιμοποιήθηκε για την υλοποίηση της αρχιτεκτονικής LibraRing. Στη συνέχεια, αναφερθήκαμε σε δύο μεγάλες κατηγορίες λειτουργιών που αρμόζουν στα δίκτυα ομότιμων κόμβων και που προσφέρει η αρχιτεκτονική LibraRing, την ανάκτηση πληροφορίας και τη δημοσίευση / συνδρομή. Παρουσιάσαμε, επίσης, τα εργαλεία εκείνα που είναι απαραίτητα για την αναπαράσταση των δεδομένων που διαχειρίζεται η αρχιτεκτονική LibraRing, που είναι το μοντέλο αναπαράστασης πληροφορίας, AWPS και η γλώσσα αναπαράστασης δεδομένων, XML. Τέλος, παρουσιάσαμε εκτενώς την αρχιτεκτονική LibraRing καθώς και την υλοποίησή της.

Αυτή η διπλωματική εργασία μπορεί να επεκταθεί με διάφορους τρόπους. Συγκεκριμένα,

- η υλοποίηση της αρχιτεκτονικής LibraRing θα μπορούσε να επεκταθεί προσθέτοντας την δυνατότητα της αποχώρησης / συμμετοχής νέων υπερ-κόμβων.

  Αυτό που γίνεται τώρα είναι κάπως πιο στατικό. Οι υπερ-κόμβοι είναι οι πρώτοι που δημιουργούνται και διαμορφώνουν το επικαλυπτόμενο δίκτυο του LibraRing. Αυτό, βέβαια, δεν αποκλείει την μετέπειτα συμμετοχή ή αποχώρησή τους. Κάτι τέτοιο, όμως,

επειδή δεν έχει υλοποιηθεί, έχει το μειονέκτημα ότι άπαξ και υπάρξει ένα τέτοιο ενδεχόμενο, τα δεδομένα που αποθηκεύει ο εκάστοτε υπερ-κόμβος χάνονται σε περίπτωση αποχώρησης του ή σε περίπτωση συμμετοχής του τα δεδομένα των γειτόνων του δεν μοιράζονται από κοινού με αυτόν. Τονίζεται, ότι η αρχιτεκτονική LibraRing έχει σχεδιαστεί έτσι ώστε να λαμβάνει υπ' όψη της αυτή τη λειτουργικότητα.

- Θα μπορούσε να αλλάξει ο τρόπος διεξαγωγής της επικοινωνίας μεταξύ των πελατών του συστήματος LibraRing και των υπερ-κόμβων.

  Όπως έχει αναφερθεί, η τωρινή επικοινωνία γίνεται σειρικοποιώντας και αποσειρικοποιώντας τα μεταφερόμενα δεδομένα. Το μειονέκτημα αυτής της μεθόδου είναι ότι δεν προσφέρεται για εφαρμογές πελατών υλοποιημένες σε διαφορετική γλώσσα προγραμματισμού. Μία πιο ευέλικτη μέθοδος, που θα προσέδιδε στην επικοινωνία τη διαλειτουργικότητα που χρειάζεται, θα ήταν η ανταλλαγή των δεδομένων υπό μορφή XML, κάτι για το οποίο η γλώσσα XML ενδείκνυται.

- Θα μπορούσαν να διεξαχθούν πειράματα για την αξιολόγηση της αποδοτικότητας τόσο της υλοποίησης του LibraRing, όσο και της ίδιας της αρχιτεκτονικής.

  Τα πειράματα αυτά θα μπορούσαν να διεξαχθούν στο PlanetLab. Μάλιστα, έχουν γίνει ήδη κάποια πειράματα που αφορούν στην αποτίμηση της αρχιτεκτονικής και τα οποία μπορούν να βρεθούν στο [8].

- Θα μπορούσε να υπάρξει μία διαφορετική υλοποίηση της αρχιτεκτονικής LibraRing χρησιμοποιώντας ένα διαφορετικό σύστημα που προσφέρει τη λειτουργικότητα του Κατανεμημένου Πίνακα Κατακερματισμού, όπως το Chord ή το Bamboo.

  Αυτό, σε συνδυασμό με την προηγούμενη επέκταση θα μπορούσαν να συνεισφέρουν στην αποτελεσματικότερη αξιολόγηση της υλοποίησης και της σχεδίασης της αρχιτεκτονικής του LibraRing.

Ορολογία

Πίνακας 1: Ορολογία.

Ξενόγλωσσος όρος	Ελληνικός όρος
access point	σημείο πρόσβασης
application layer	επίπεδο εφαρμογών
attribute	γνώρισμα
bit	διφίο ή διαδικό ψηφίο
cache	κρυφή μνήμη
client table	πίνακας πελατών
client	πελάτης
collaboration networks	συνεργαζόμενα δίκτυα
consistent hashing	συνεπής κατακερματισμός
continuous query	επερώτηση διαρκείας
data retrieval	ανάκτηση δεδομένων
decoupling	αποσύνδεση
deserialis(z)e	αποσειρικοποιώ
distance function	συνάρτηση απόστασης
distributed digital libraries	κατανεμημένες ψηφιακές βιβλιοθήκες
Distributed Hash Table	Κατανεμημένος Πίνακας Κατακερματισμού
Domain Name System	Σύστημα Πεδίου Ονομάτων
e-commerce	ηλεκτρονικό εμπόριο
event manager	υπεύθυνος γεγονότων
event notification service	υπηρεσία ειδοποίησης γεγονότων
fault tolerance	ανοχή σε σφάλματα
Συνεχίζεται στην επόμενη σελίδα	

Πίνακας 1 – συνέχεια από την προηγούμενη σελίδα

Ξενόγλωσσος όρος	Ελληνικός όρος
field	πεδίο
finger table	πίνακας δεικτών
grid	πλέγμα
handshake	χειραψία μεταξύ δύο οντοτήτων για την εδραίωση δικτυακής σύνδεσης
hop	αναπήδηση
host node	ξένιος κόμβος
identifier	αναγνωριστικό
index term	όρος ευρετηρίου
information management	διαχείριση πληροφορίας
information filtering	διήθηση / φιλτράρισμα πληροφορίας
information retrieval	ανάκτηση πληροφορίας
IP address	διεύθυνση IP
Java	Αντικειμενοστραφής γλώσσα προγραμματισμού
keyword	λέξη-κλειδί
key	κλειδί
leaf set	σύνολο φύλλων
logical view of document	λογική άποψη εγγράφου
lookup problem	πρόβλημα αναζήτησης
neighborhood set	σύνολο γειτόνων
neighbour nodes / peers	γειτονικοί κόμβοι
network layer	επίπεδο δικτύου
node arrival	άφιξη κόμβου
node failure	αποτυχία κόμβου
node recovery	επανάκαμψη κόμβου
notification	ειδοποίηση
object	αντικείμενο
one-time query	στιγμιαία επερώτηση
overlay network of nodes	δίκτυο επικάλυψης κόμβων

Συνεχίζεται στην επόμενη σελίδα

Πίνακας 1 – συνέχεια από την προηγούμενη σελίδα

Ξενόγλωσσος όρος	Ελληνικός όρος
peer-to-peer network	δίκτυο ομότιμων κόμβων
peer	κόμβος
predecessor	προκάτοχος
prefix tree	προθεματικό δέντρο
primitive data types	πρωταρχικές δομές δεδομένων
provider	πάροχος
proximity formula	φόρμουλα εγγύτητας
proximity metric	μετρική εγγύτητας
publication	δημοσίευση
publisher	εκδότης
publish / subscribe	δημοσίευση / συνδρομή
query	επερώτηση
repository	αποθήκη
resilience	προσαρμοστικότητα
retrieval task	εργασία ανάκτησης
robust	εύρωστος
routing state	κατάσταση δρομολόγησης
routing table	πίνακας δρομολόγησης
scalability	κλιμάκωση
serialis(z)e	σειρικοποιώ
similarity threshold	κατώφλι ομοιότητας
similarity	ομοιότητα
sockets	υποδοχές
stabilization algorithm	αλγόριθμος σταθεροποίησης
structured lookup	δομημένη αναζήτηση
subscriber	συνδρομητής
subscription	συνδρομή
successor list	λίστα διαδόχων
successor node	διάδοχος κόμβος

Συνεχίζεται στην επόμενη σελίδα

Πίνακας 1 – συνέχεια από την προηγούμενη σελίδα

Ξενόγλωσσος όρος	Ελληνικός όρος
super-peer	υπερ-κόμβος
symmetric lookup algorithms	συμμετρικοί αλγόριθμοι αναζήτησης
tag	ετικέτα
thread	νήμα
trie	προθεματικό δέντρο
unstructured routing table	αδόμητος πίνακας δρομολόγησης
user information need	πληροφορίες που χρειάζεται ο χρήστης
user layer	επίπεδο χρήστη
user task	εργασία χρήστη
word pattern	λεκτικό πρότυπο / σχήμα
word proximity	εγγύτητα λέξεων
World Wide Web	Παγκόσμιος Ιστός

Συντμήσεις - Αρκτικόλεξα

Πίνακας 2: Συντμήσεις - Αρκτικόλεξα.

Σύντμηση	**Πλήρης Ανάπτυξη**
AP	Access Point
ASCII	American Standard Code for Information Interchange
AWPS	Attribute Word-Pattern with Similarity
B2B	Business-to-Business
CT	Client Table
DHT	Distributed Hash Table
DNS	Domain Name System
DTD	Document Type Definition
FTP	File Transfer Protocol
HTML	HyperText Markup Language
IP	Internet Protocol
IF	Information Filtering
IR	Information Retrieval
LSI	Latent Semantic Indexing
p2p	peer-to-peer
pub / sub	publish / subscribe
SHA-1	Secure Hash Algorithm-1
SP	Super-Peer
TCP	Transfer Control Protocol
telnet	TELecommunication NETwork
VSM	Vector Space Model

Συνεχίζεται στην επόμενη σελίδα

Πίνακας 2 – συνέχεια από την προηγούμενη σελίδα

Σύντμηση	Πλήρης Ανάπτυξη
WWW	World Wide Web
XML	EXtensible Markup Language
ΑΔ	Ανάκτηση Δεδομένων
ΑΠ	Ανάκτηση Πληροφορίας
ΚΠΚ	Κατανεμημένος Πίνακας Κατακερματισμού

Βιβλιογραφία

[1] A. Galardo-Antolin, A. Navia-Vasquez, H.Y. Molina-Bulla, A.B. Rodriquez-Gonzalez, F.J. Valvarde-Albacete, A.R. Figueiras-Vidal, T. Koutris, A. Xiruhaki, and M. Koubarakis. *I-Gaia: an Information Processing Layer for the DIET Platform*. In Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002). September 15-19 2002.

[2] A. Oram, editor. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O' Reilly & Associates. March 2001.

[3] A. Rowstron and P. Druschel. *Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems*. In Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Paltforms (Middleware 2001). November 2001.

[4] A. Rowstron and P. Druschel. *Pastry: Scalable, Distributed Object Location and Routing for Large Scale Peer-to-Peer Storage Utility*. In Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Paltforms (Middleware 2001). November 2001.

[5] A. Rowstron and P. Druschel. *Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility*. 18th ACM SOSP '01, Lake Louise, Alberta, Canada. October 2001.

[6] B. Yang and H. GarciaMolina. *Designing a superpeer network*. In Proceedings of the 19th International Conference on Data Engineering (ICDE 2003). March 2003.

[7] C. Tryfonopoulos, M. Koubarakis, and Y. Drougas. *Filtering Algorithms for Information Retrieval Models with Named Attributes and Proximity Operators*. In Proc. of ACM SIGIR. 2004.

[8] C. Tryfonopoulos, S. Idreos, and M. Koubarakis. *LibraRing: An Architecture for Distributed Digital Libraries Based on DHTs*. In Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries (ECDL), pages 25-36, Vienna, Austria. September 2005.

[9] D. Malkhi, M. Naor, and D. Ratajczak. *Viceroy: A scalable and dynamic emulation of the butterfly*. In Proceedings of ACM Principles of Distributed Computing (PODC) Monterey. CA (July 2002).

[10] FIPS 180-1. *Secure Hash Standard*. U.S. Department of Commerce/NIST, National Technical Information Service, Springfield, VA. Apr. 1995.

[11] F. Wang. *Self-organising Communities Formed by Middle Agents*. In Proceedings of the 1st International Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS 2002). September 15-19 2002.

[12] H. Balakrishnan, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. *Looking Up Data in P2P Systems*. Communications of the ACM, Vol. 46, No. 2. February 2003.

[13] I. Clarke, O. Sandberg, B. Wiley, and T. Hong. *Freenet: A distributed anonymous information storage and retrieval system*. In Proceedings of ICSI Workshop on Design Issues in Anonymity and Unobservability. Berkeley. California (June 2000), freenet.sourceforge. net.

[14] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. *Chord: A Scalable peer-to-peer Lookup Service for Internet Applications*. In Proceedings of the ACM SIGCOMM '01 Conference, San Diego, California. August 2001.

[15] I. Stoica, R. Morris, D. LibenNowell, D. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. *Chord: a scalable peertopeer lookup protocol for internet applications*. IEEE/ACM Transactions on Networking. 11(1):17-32, 2003.

[16] K. Aberer, M. Hauswirth. *An Overview on Peer-to-Peer Information Systems*. W-DAS2002. 2002.

[17] K. Hildrum, J. Kubiatowicz, S. Rao, and B. Zhao. *Distributed Object Location in a Dynamic Network*. In Proceedings of 14th ACM Symp. on Parallel Algorithms and Architectures (SPAA). August 2002.

[18] M. Castro, P. Druschel, Y. C. Hu, and A. Rowstron. *Topology-aware routing in structured peer-to-peer overlay networks*. 2002.

[19] M. Koubarakis. *Boolean Queries with Proximity Operators for Information Dissemination*. Proceedings of the workshop on Foundations of Models and Languages for Information Integration (FMII-2001), Viterbo, Italy. 16-18 September 2001.

[20] M. Koubarakis. *Textual Information Dissemination in Distributed Event-Based Systems*. Proceedings of the International Workshop on Distributed Event-Based systems (DEBS'02). July 2-3, 2002, Vienna, Austria.

[21] M. Koubarakis, T. Koutris, C. Tryfonopoulos, and P. Raftopoulou. *Information Alert in Distributed Digital Libraries: The Models, Languages and Architecture of DIAS*. In Proc. of ECDL. 2002.

[22] M. Stonebraker, P. M. Aoki, W. Litwin, A. Pfeffer, A. Sah, J. Sidell, C. Staelin, and A. Yu. *Mariposa: A Wide-Area Distributed Database System*. VLDB Journal, 5(1):48-63. 1996.

[23] N. Belkin and B. Croft. *Information filtering and information retrieval: Two sides of a same coin?*. Communications of the ACM, 35(12):29-38. December 1992.

[24] P.A. Chirita, S. Idreos, M. Koubarakis, and W. Nejdl. *Publish/Subscribe for RDFbased P2P Networks*. In Proceedings of the 1st European Semantic Web Symposium (ESWS 2004), volume 3053 of Lecture Notes in Computer Science, pages 182-197, Heraklion, Crete, Greece. May 1012 2004.

[25] P. Marrow, M. Koubarakis, R.H. van Lengen, F. Valverde-Albacete, E. Bonsma, J. Cid-Suerio, A.R. Figueiras-Vidal, A. Gallardo-Antolin, C. Hoile, T. Koutris, H. Molina-Bulla, A. Navia-Vazquez, P. Raftopoulou, N. Skarmeas, C. Tryfonopoulos, F. Wang, and C. Xiruhaki. In M. Schroeder and K. Stathis, editors. *Agents in Decentralised Information Ecosystems: The DIET Approach*. Proceedings of the AISB'01 Symposium on Information Agents for Electronic Commerce, AISB'01 Convention, pages 109-117, University of York, United Kingdom. March 2001.

[26] P. Maymounkov, and D. Mazieres. *Kademlia: A peer-to-peer information system based on the XOR metric*. In Proceedings of the 1st International Workshop on

Peer-to-Peer Systems, Springer-Verlag version, Cambridge. MA (Mar. 2002), kademia.scs.cs.nyu.edu.

[27] P. Raftopoulou, M. Koubarakis, C. Tryfonopoulos and T. Koutris. *Data models and languages for agent-based textual information dissemination*. In Proceedings of the 6th International Workshop on Cooperative Information Agents(CIA2002), Madrid, Lecture Notes in Computer Science. Springer. 2002.

[28] P. Th. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec. *The Many Faces of Publish / Subscribe*. ACM Computing Surveys, Vol. 35, No. 2, pp. 114-131. June 2003.

[29] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrival*. Addison Wesley. 1999.

[30] S. Idreos, C. Tryfonopoulos, M. Koubarakis, and Y. Drougas. *Query Processing in SuperPeer Networks with Languages Based on Information Retrieval: the P2PDIET Approach*. In Proceedings of the 1st International Workshop on PeertoPeer Computing and DataBases (P2P&DB 2004), volume 3268 of Lecture Notes in Computer Science, pages 496-505, Heraklion, Crete, Greece. March 2004.

[31] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. *A scalable content-addressable network*. In Proceedings of ACM SIGCOMM, San Diego. CA (August 2001).

[32] S.E Robertson. *The probability ranking principle in IR*. J. Doc. 33, 4 (Dec.1977), 294-304. CA (August 2001).

[33] T.W. Yan and H. Garcia-Molina. *The SIFT information dissemination system*. ACM TODS, 24(4):529-565. 1999.

[34] W. Neidl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. *EDUTELLA: A P2P networking infrastructure based on RDF*. In Proceedings of the 11th International World Wide Web Conference. May 2002.

[35] Σ. Μηλιαράκη. *Ανάκτηση Σημασιολογικής Πληροφορίας για Υπηρεσίες του Παγκόσμιου Ιστού Χρησιμοποιώντας Κατανεμημένους Πίνακες Κατακερματισμού*. Διπλωματική Θέση. Τμήμα Πληροφορικής & Τηλεπικοινωνιών, Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών. 2005.

[36] Χ. Δουληγέρης, Ρ. Μαυροπόδη και Ε. Κοπανάκη. Τεχνολογίες Εφαρμογών Διαδικτύου. Εκδόσεις Νηρηίδες. 2004.