

Πανεπιστήμιο Κρήτης  
Σχολή Θετικών και Τεχνολογικών Επιστημών  
Τμήμα Επιστήμης Υπολογιστών

Διαμοιρασμός και Διαχείριση Σχεσιακών Δεδομένων  
σε Ομότιμα Δίκτυα

Νικόλαος Δημόκας

Μεταπτυχιακή Εργασία

Ηράκλειο Κρήτης, Απρίλιος 2004



Πανεπιστήμιο Κρήτης  
Σχολή Θετικών και Τεχνολογικών Επιστημών  
Τμήμα Επιστήμης Υπολογιστών

## Διαμοιρασμός και Διαχείριση Σχεσιακών Δεδομένων σε Ομότιμα Δίκτυα

Εργασία που υποβλήθηκε από τον  
Νικόλαο Δημόκα  
ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση  
Μεταπτυχιακού Διπλώματος Ειδίκευσης στην Επιστήμη  
Υπολογιστών

Συγγραφέας: \_\_\_\_\_

Νικόλαος Δημόκας  
Τμήμα Επιστήμης Υπολογιστών

Εισηγητική Επιτροπή: \_\_\_\_\_

Πλεξουσάκης Δημήτρης, Αναπληρωτής Καθηγητής, Επόπτης

\_\_\_\_\_

Αντωνίου Γρηγόρης, Καθηγητής, Μέλος

\_\_\_\_\_

Μπίλας Άγγελος, Αναπληρωτής Καθηγητής, Μέλος

Δεκτή: \_\_\_\_\_

\_\_\_\_\_

Πλεξουσάκης Δημήτρης, Αναπληρωτής Καθηγητής,  
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ηράκλειο, Απρίλιος 2004



# Διαμοιρασμός και Διαχείριση Σχεσιακών Δεδομένων σε Ομότιμα Δίκτυα

Νικόλαος Δημόκας  
Μεταπτυχιακή Εργασία

Πανεπιστήμιο Κρήτης  
Τμήμα Επιστήμης Υπολογιστών

## Περίληψη

Τα ομότιμα (peer-to-peer) συστήματα έχουν γίνει πρόσφατα ένα δημοφιλές μέσο, διαμέσου του οποίου είναι δυνατόν να διαμοιραστούν μεγάλα ποσά δεδομένων. Τα peer-to-peer συστήματα είναι καταναμημένα συστήματα στα οποία οι κόμβοι μπορούν να διαμοιράσουν πόρους, πληροφορίες και υπηρεσίες, ενώ εξακολουθούν να παραμένουν πλήρως αυτόνομοι. Παρά το γεγονός ότι τα peer-to-peer συστήματα έχουν μια μεγάλη ποικιλία από εφαρμογές, υπάρχουν ακόμη πεδία εφαρμογών τα οποία δεν έχουν αναπτυχθεί πολύ. Ένα από αυτά έχει να κάνει με τα peer-to-peer δίκτυα και τις τεχνολογίες βάσεων δεδομένων. Η κύρια ιδέα είναι η υποστήριξη και ο διαμοιρασμός των σχεσιακών δεδομένων που βρίσκονται στις βάσεις δεδομένων μέσα σε ένα peer-to-peer περιβάλλον.

Εξαιτίας της δυναμικής φύσης κάθε κόμβου η θεώρηση ενός συνολικού σχήματος δεν είναι εφικτή, όπως και ο έλεγχος στην τοποθέτηση των σχεσιακών δεδομένων σε κάθε κόμβο του δικτύου. Συνακόλουθα, η αποστολή μιας SQL επερώτησης σε όλους τους κόμβους του δικτύου είναι μια μη πρακτική και κλιμακώσιμη ενέργεια, καθώς οι κόμβοι δε θα μπορούν να ερμηνεύσουν την επερώτηση εξαιτίας της ετερογένειας στην πληροφορία, ενώ συνακόλουθα θα υπάρχει μεγάλη επιβάρυνση στην κίνηση του δικτύου σε πλήθος μηνυμάτων. Για την εξυπηρέτηση και την σωστή δρομολόγηση μιας επερώτησης σε peer-to-peer περιβάλλον έχουν προταθεί τεχνικές για την δημιουργία αντιστοιχίσεων μεταξύ των σχημάτων.

Στην παρούσα εργασία παρουσιάζουμε την σχεδίαση και υλοποίηση της εφαρμογής RDPeer για τη διαχείριση και διαμοιρασμό των σχεσιακών δεδομένων, χωρίς την ύπαρξη ενός συνολικού σχήματος ή τη δημιουργία κοινοτήτων από peers βασισμένη στο σημασιολογικό περιεχόμενο των δεδομένων. Το RDPeer είναι ένα υβριδικό peer-to-peer σύστημα, που επιτρέπει σε κάθε peer να συμμετέχει ταυτόχρονα σε περισσότερες από μία κοινότητες, να συνδέεται σε μια βάση δεδομένων, να θέτει SQL επερωτήσεις και να λάβει απαντήσεις από τους peers που έχουν σχετικά με την επερώτηση δεδομένα. Το RDPeer προσπαθεί να επιλύσει το πρόβλημα της

δρομολόγησης των επερωτήσεων με την δυναμική δημιουργία των super-peers μέσω «εκλογών». Οι super-peers αναλαμβάνουν να συλλέξουν τα σχήματα των σχεσιακών βάσεων δεδομένων και να εξυπηρετήσουν τις επερωτήσεις. Η εξυπηρέτηση των επερωτήσεων πραγματοποιείται σε δύο φάσεις. Η πρώτη περιλαμβάνει τη δημιουργία αντιστοιχίσεων μεταξύ των SQL επερωτήσεων και των σχημάτων των peers κατά τη διάρκεια που τίθεται η επερώτηση, ενώ η δεύτερη λαμβάνοντας τις αντιστοιχίσεις διενεργεί μετασχηματισμό των επερωτήσεων. Η διαδικασία του μετασχηματισμού έχει σαν αποτέλεσμα την παραγωγή SQL επερωτήσεων για τους peers που διαθέτουν δεδομένα και βοηθάει στην ισότιμη εξαγωγή τους από κάθε peer.

Επόπτης: Πλεξουσάκης Δημήτρης, Αναπληρωτής Καθηγητής

# Sharing and Management of Relational Data in Peer-to-Peer Networks

Nikolaos Dimokas

Master of Science Thesis

University of Crete

Department of Computer Science

## Abstract

Peer-to-peer systems have recently become a popular means for transferring large amounts of data from one peer to another. Peer-to-peer systems are distributed systems in which nodes can share resources, information and services, while they continue to remain completely independent. Despite the fact that the peer-to-peer systems have a great variety of applications, there are still application areas which have not been developed. One of these has to do with peer-to-peer networks and databases technologies. The main idea is the support and sharing of relational data that can be found in the databases in a peer-to-peer environment.

Due to the dynamic nature of each node, neither the consideration of a global schema, nor the control of data placement is feasible in every node of the network. Besides, the sending of an SQL query to all nodes of the network is a non-practical and scalable action as the nodes will not be able to interpret the query due to the heterogeneity of the information. Moreover, there will be an overhead in network traffic as a consequence of the number of messages sent. Several techniques have been proposed for the creation of mappings among the schemas, in order to serve and route a query properly.

In the present study, we present the design and implementation of the RDPeer application for the management and sharing of relational data, without the existence of a global schema or the creation of communities from peers based on the semantic content of the data. RDPeer is a hybrid peer-to-peer system, which allows every peer to participate in more than one communities simultaneously, to communicate with a database, to pose SQL queries and receive answers from the peers that have data related to the queries. RDPeer tries to resolve the problem of query routing by the dynamic creation of super-peers through a system of “elections”. The super-peers undertake to collect the schemas of relational databases and to process the queries. Processing of queries is accomplished in two phases. The first phase includes the creation of mappings among

the SQL queries and the schemas of peers during querying. The second one transforms the queries based on the mappings. The transformation procedure results in the creation of SQL queries for the peers who contain relevant data and enables each peer to extract the data in a peer-to-peer way.

Supervisor: Plexousakis Dimitris, Associate Professor



## Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επόπτη καθηγητή μου, κ. Πλεξουσάκη Δημήτρη, για την δυνατότητα που μου έδωσε να ασχοληθώ με τον συγκεκριμένο τομέα, και για τις εύστοχες πάντα παρατηρήσεις και υποδείξεις του.

Επίσης, ευχαριστώ θερμά τους γονείς μου Χρήστο και Αποστολίνα και τον αδερφό μου Γιώργο που με στήριξαν σε όλη την διάρκεια των σπουδών μου.

Τέλος, θα ήθελα να ευχαριστήσω το Τμήμα Επιστήμης Υπολογιστών του Πανεπιστημίου Κρήτης καθώς και το Ινστιτούτο Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας για την υλικοτεχνική και οικονομική υποστήριξη που μου παρείχαν.



## Περιεχόμενα

Κατάλογος Πινάκων .....	XIII
Κατάλογος Εικόνων .....	XV
Κατάλογος Σχημάτων .....	XVII
1 Εισαγωγή.....	1
1.1 Ομότιμα Συστήματα .....	1
1.2 Αντικείμενο της εργασίας.....	2
1.3 Περιεχόμενο της εργασίας.....	4
2 Σχετικές εργασίες.....	5
2.1 Peer-to-Peer συστήματα.....	5
2.1.1 Αρχιτεκτονική υβριδικών συστημάτων .....	6
2.1.2 Αρχιτεκτονική καθαρών συστημάτων .....	8
2.1.3 Αρχιτεκτονική super-peer συστημάτων.....	11
2.2 Διαχείριση και διαμοιρασμός δεδομένων σε P2P συστήματα.....	13
2.2.1 Σχεδιαστικά θέματα για διαχείριση σχεσιακών δεδομένων σε peer-to-peer συστήματα.....	14
2.2.2 Το σύστημα PeerDB .....	15
2.2.3 Το σύστημα Piazza .....	16
2.2.4 Το σύστημα Hyperion.....	17
2.3 Σχεδιαστικές αποφάσεις για το RDPeer .....	18
3 Τεχνολογία JXTA .....	21
3.1 Αρχιτεκτονική της τεχνολογίας JXTA.....	22
3.1.1 Επίπεδο πλατφόρμας (JXTA Core) .....	22
3.1.2 Επίπεδο υπηρεσιών (JXTA Services).....	23
3.1.3 Επίπεδο εφαρμογών (JXTA Applications).....	23
3.1.4 Θέματα κλειδιά στην αρχιτεκτονική JXTA .....	23
3.2 Αρχές της τεχνολογίας JXTA .....	24
3.2.1 Peers.....	24
3.2.2 Κοινότητες από Peers .....	24
3.2.3 Σωληνώσεις.....	26
3.2.4 Μηνύματα .....	26
3.2.5 Δημοσιοποιήσεις.....	27
3.2.6 Ασφάλεια .....	27
3.2.7 Αναγνωριστικά.....	28
3.3 Αρχιτεκτονική του Δικτύου .....	28
3.3.1 Οργάνωση Δικτύου.....	29
3.3.2 Διαμοιραζόμενου Πόρου Κατανεμημένος Δείκτης .....	30
3.3.3 Firewall και NAT.....	30
3.4 Πρωτόκολλα τεχνολογίας JXTA .....	31
3.4.1 Πρωτόκολλο Ανακάλυψης Peer .....	31
3.4.2 Πρωτόκολλο Πληροφορίας Peer .....	32
3.4.3 Πρωτόκολλο Δέσμευσης Σωλήνων .....	32
3.4.4 Πρωτόκολλο Resolver Peer .....	33
3.4.5 Πρωτόκολλο Ραντεβού .....	33

3.4.6	Πρωτόκολλο Δρομολόγησης Κατάληξης .....	33
3.5	Πλεονεκτήματα και μειονεκτήματα της τεχνολογίας .....	34
3.6	Διαφορές της τεχνολογίας JXTA από την Jini και .NET .....	36
4	Περιγραφή του RDPeer .....	37
4.1	Peer Σχεσιακών δεδομένων (RDPeer) .....	37
4.1.1	Γενική περιγραφή λειτουργίας RDPeer .....	40
4.1.2	Αρχικοποίηση του RDPeer .....	41
4.1.3	Προεργασία για το διαμοιρασμό των δεδομένων .....	44
4.1.4	Φάση διαμοιρασμού των δεδομένων .....	45
4.1.4.1	Παροχή λέξεων κλειδιών .....	46
4.1.4.2	Έκδοση σχήματος βάσης δεδομένων .....	47
4.1.4.2.1	Δημιουργία θησαυρού όρων .....	48
4.1.4.2.2	Δημιουργία XML εγγράφου .....	50
4.1.4.3	Υποβολή SQL επερώτησης .....	52
4.2	Αρχιτεκτονική του Συστήματος .....	56
4.3	Αρχιτεκτονική του Peer .....	59
4.3.1	DatabaseHandler .....	61
4.3.2	DataDictionary .....	61
4.3.3	XmlHandler .....	62
4.3.4	QueryProcessor .....	66
4.3.5	JXTA module .....	66
4.3.6	WebService .....	67
4.3.7	RequestServer .....	67
4.3.8	ElectionServer .....	68
4.3.9	Τρόποι επικοινωνίας μεταξύ peers .....	68
4.3.9.1	Δομή επικοινωνίας με RequestServer .....	69
4.3.9.2	Δομή επικοινωνίας με ElectionServer .....	70
4.4	Αντιστοίχιση σχημάτων και μετασχηματισμός SQL επερώτησης .....	73
4.4.1	Δημιουργία αντιστοιχίσεων .....	74
4.4.2	Μετασχηματισμός SQL επερώτησης .....	79
4.5	Εκλογές .....	86
4.6	Τεχνολογίες του συστήματος RDPeer .....	90
5	Συμπεράσματα της εργασίας και μελλοντικές επεκτάσεις .....	93
6	Αναφορές και Βιβλιογραφία .....	96

## Κατάλογος Πινάκων

Πίνακας 1 Θησαυρός όρων για ονόματα πεδίων και σχέσεων.....	79
Πίνακας 2 Παράδειγμα παραγωγής αντιστοιχίσεων για δύο peers. ....	79
Πίνακας 3 Παράδειγμα μετασχηματισμένων ερωτήσεων.....	81
Πίνακας 4 Δημιουργία αντιστοιχίσεων και μετασχηματισμός SQL ερώτησης.....	82
Πίνακας 5 Παράδειγμα μετασχηματισμένων ερωτήσεων για την κεντροποιημένη σύνθεση της απάντησης.....	84
Πίνακας 6 Παράδειγμα μετασχηματισμένων ερωτήσεων για την κατακευματημένη σύνθεση της απάντησης.....	85



## Κατάλογος Εικόνων

Εικόνα 1 Αριστερά : απεικονίζεται ένα μέρος της αρχιτεκτονικής του Napster. Δεξιά : μέρος του δικτύου της Gnutella.....	7
Εικόνα 2 Παράδειγμα πίνακα δρομολόγησης που διατηρεί κάθε κόμβος του Freenet. ...	10
Εικόνα 3 Αρχιτεκτονική super-peer δικτύου.....	12
Εικόνα 4 Τα τρία επίπεδα αρχιτεκτονικής της τεχνολογίας JXTA.....	22
Εικόνα 5 Παράδειγμα δημοσιοποίησης σωλήνα. ....	27
Εικόνα 6 Οι επιλογές αναζήτησης των peers, και των κοινοτήτων.....	43
Εικόνα 7 Οι επιλογές δημιουργίας, συμμετοχής και εγκατάλειψης .....	43
Εικόνα 8 Οι επιλογές για παροχή λέξεων κλειδιών, κοινοποίησης του σχήματος της Βάσης Δεδομένων και δημιουργίας SQL επερώτησης.....	45
Εικόνα 9 Η φόρμα στην οποία μπορεί να δώσει ο χρήστης λέξεις κλειδιά για κάθε όνομα .....	47
Εικόνα 10 Το ηλεκτρονικό λεξικό που χρησιμοποιεί το RDPeer για την εύρεση συνωνύμων.....	49
Εικόνα 11 Δημιουργία SQL επερώτησης, .....	52
Εικόνα 12 Τα αποτελέσματα που εμφανίζονται σε έναν peer που έχει θέσει μια SQL επερώτηση.....	55





## Κατάλογος Σχημάτων

Σχήμα 1 Αρχιτεκτονικές peer-to-peer συστημάτων και η αρχιτεκτονική Client / Server.	57
Σχήμα 2 Η αρχιτεκτονική του συστήματος .....	58
Σχήμα 3 Η αρχιτεκτονική του multithreaded server που διαθέτει κάθε peer. ....	59
Σχήμα 4 Η αρχιτεκτονική του RDPeer .....	60
Σχήμα 5 Η δομή επικοινωνίας ενός peer με το RequestServer.....	70
Σχήμα 6 Η δομή της επικοινωνίας μεταξύ του peer, που ανακαλύπτει την απουσία του super-peer,.....	71
Σχήμα 7 Η δομή της επικοινωνίας του διαχειριστή των εκλογών με τους υπόλοιπους ...	72
Σχήμα 8 Κεντριοποιημένο σύστημα για σύνθεση της απάντησης σε μία SQL επερώτηση .....	83
Σχήμα 9 Κατανεμημένο σύστημα για σύνθεση της απάντησης σε μία SQL επερώτηση.	83



# 1 Εισαγωγή

## 1.1 Ομότιμα Συστήματα

Τα ομότιμα (peer-to-peer) συστήματα λαμβάνουν τελευταία μεγάλης προσοχής, καθώς προσφέρουν δυνατότητες για το διαμοιρασμό μεγάλου όγκου δεδομένων. Τα peer-to-peer συστήματα είναι κατακευμαμένα συστήματα στα οποία οι κόμβοι μπορούν να συμμετέχουν για να διαμοιράσουν τους πόρους, τις πληροφορίες και τις υπηρεσίες, ενώ εξακολουθούν να παραμένουν πλήρως αυτόνομοι. Επιπρόσθετα τα συστήματα αυτά συμπεριλαμβάνουν την εξισορρόπηση του φόρτου εργασίας, την προσαρμοστικότητα, και την ανοχή σε αποτυχίες.

Υπάρχουν διάφορες κατηγορίες από peer-to-peer συστήματα ανάλογα με την αρχιτεκτονική που χρησιμοποιείται για να οργανωθούν οι κόμβοι του δικτύου και να πραγματοποιηθεί η αναζήτηση της πληροφορίας. Στα *καθαρά* (pure) συστήματα, όπως είναι το Gnutella [15] και το Freenet [11] όλοι οι peers έχουν ισοδύναμους ρόλους και ευθύνες. Στα *υβριδικά* συστήματα παρουσιάζεται η ύπαρξη ορισμένων εξυπηρετητών, στους οποίους απευθύνονται όλοι οι κόμβοι του δικτύου και οι οποίοι αναλαμβάνουν την αναζήτηση της πληροφορίας. Ενώ η αναζήτηση πραγματοποιείται κεντροποιημένα, η εξαγωγή των δεδομένων εξακολουθεί να συμβαίνει με ισότιμο τρόπο. Στα *super-peer* δίκτυα, όπως είναι το KaZaA [23] εμφανίζεται μια ενδιάμεση κατάσταση μεταξύ των υβριδικών και pure συστημάτων. Οι πελάτες αποστέλλουν ερωτήσεις στο super-peer και λαμβάνουν απαντήσεις από αυτόν, όπως και σε ένα υβριδικό σύστημα. Όμως οι super-peers είναι επίσης συνδεδεμένοι μεταξύ τους, όπως είναι οι peers σε ένα pure σύστημα.

Τα συστήματα όμως που έχουν δημιουργηθεί [11,15,23,28,32,38] προσφέρουν πολύ περιορισμένες δυνατότητες για τη διαχείριση των δεδομένων. Στις περισσότερες των περιπτώσεων η αναζήτηση της πληροφορίας βασίζεται σε μια απλή επιλογή που διεξάγεται σε ένα προκαθορισμένο σύνολο από χαρακτηριστικά γνωρίσματα ή στην εύρεση ομοιοτήτων μεταξύ αλφαριθμητικών. Χαρακτηριστικό παράδειγμα αποτελεί το peer-to-peer σύστημα Gnutella, στο οποίο η αναζήτηση περιορίζεται σε αλφαριθμητικά που μπορεί να περιέχονται σε ένα αρχείο. Με τον τρόπο αυτό είναι δυνατόν να επιτευχθεί ο διαμοιρασμός αρχείων, αλλά δεν ενδείκνυται για την περίπτωση που πρέπει να διαμοιραστούν ετερογενή, εννοιολογικά πλούσια δεδομένα.

Όλα και περισσότεροι χρήστες στη σημερινή εποχή προτιμούν να αποθηκεύουν τα δεδομένα τους σε συστήματα βάσεων δεδομένων, και να τα διαμοιράζονται. Τα σχεσιακά δεδομένα όμως έχουν πλούσια δομή, σημασιολογία, και οι επερωτήσεις που τίθενται είναι πολύπλοκες. Παρά το γεγονός ότι τα peer-to-peer συστήματα έχουν μια μεγάλη

ποικιλία από εφαρμογές, υπάρχουν ακόμη πεδία εφαρμογών τα οποία δεν έχουν αναπτυχθεί πολύ. Ένα από αυτά έχει να κάνει με τα peer-to-peer δίκτυα και τις τεχνολογίες βάσεων δεδομένων. Η κύρια ιδέα είναι η υποστήριξη και ο διαμοιρασμός των σχεσιακών δεδομένων που βρίσκονται στις βάσεις δεδομένων μέσα σε ένα peer-to-peer περιβάλλον.

Η κλιμακοσιμότητα όμως του συστήματος και η δυναμική φύση των κόμβων καθώς είτε γίνονται μέλη του δικτύου, ή το εγκαταλείπουν δημιουργεί ποικίλες προκλήσεις. Εξαιτίας της δυναμικής φύσης κάθε κόμβου η θεώρηση ενός συνολικού σχήματος δεν ανταποκρίνεται στην πραγματική πληροφορία που είναι διαθέσιμη μια συγκεκριμένη χρονική στιγμή. Επιπλέον, η αποστολή μιας SQL επερώτησης σε όλους τους κόμβους του δικτύου είναι μια μη πρακτική ενέργεια, καθώς οι κόμβοι δε θα μπορούν να ερμηνεύσουν την επερώτηση εξαιτίας της ετερογένειας στην πληροφορία, ενώ συνακόλουθα θα υπήρχε μεγάλη επιβάρυνση του δικτύου σε πλήθος μηνυμάτων. Για το λόγο αυτό μπορούν να δημιουργηθούν αντιστοιχίσεις μεταξύ των διαφορετικών σχημάτων των peers, είτε κατά τη διάρκεια που τίθεται μια SQL επερώτηση [30], είτε εκ των προτέρων [3,5,13,16,24,27].

Η δημιουργία των αντιστοιχίσεων εμπεριέχει το ζήτημα σχετικά με τον τρόπο που μπορούν να δημιουργηθούν και να διατηρηθούν. Για τον σκοπό αυτό έχουν προταθεί διάφορες τεχνικές [17,24,26,27,34], είτε για το χειρονακτικό τρόπο δημιουργίας ή για αυτόματα. Συνακόλουθα, σημαντικό στοιχείο που πρέπει να οριστεί είναι ο τρόπος με τον οποίο γίνεται η επεξεργασία και βελτιστοποίηση μιας επερώτησης, γεγονός που μπορεί να συμβεί κεντροκοιμημένα ή κατανεμημένα από κάθε κόμβο του δικτύου με βάση της διαθέσιμη γνώση που κατέχει.

Σημαντική πρόκληση ακόμα αποτελεί ο τρόπος με τον οποίο μπορεί να περιοριστεί ο πλεονασμός της πληροφορίας και των υπολογισμών που διεξάγονται. Ο πλεονασμός της πληροφορίας δεν μπορεί να αποφευχθεί εκτός και αν υπάρξει κάποιος έλεγχος στην τοποθέτηση των δεδομένων όπως γίνεται στα συστήματα κατανεμημένου πίνακα κατακερματισμού (hash table) [11,35,36,40,48]. Ο πλεονασμός των υπολογισμών μπορεί να εμφανιστεί αν έχουμε κατανεμημένη επεξεργασία των ερωτήσεων, καθώς στην περίπτωση αυτή πολλοί κόμβοι συμμετέχουν στην επεξεργασία μιας επερώτησης.

## 1.2 Αντικείμενο της εργασίας

Κάθε peer που φέρει μια βάση δεδομένων, πρέπει να είναι ικανός να διαμοιράζεται ένα μέρος της βάσης δεδομένων, να μπορεί να στέλνει επερωτήσεις στο δίκτυο, να απαντάει και να δρομολογεί αποτελεσματικά τις επερωτήσεις που φθάνουν από το δίκτυο. Στην παρούσα εργασία θα παρουσιάσουμε το RDPeer (Relational Data Peer –

Peer Σχεσιακών Δεδομένων), ένα σύστημα βασισμένο στην τεχνολογία peer-to-peer για το διαμοιρασμό σχεσιακών δεδομένων.

Ο διαμοιρασμός των δεδομένων, που βρίσκονται στη σχεσιακή βάση δεδομένων κάθε peer, γίνεται χωρίς τη γνώση ενός συνολικού σχήματος. Το RDPeer επιτρέπει στο χρήστη να συνδέεται σε μια βάση δεδομένων, να συμμετέχει σε κοινότητες από peers μέσα στις οποίες μπορεί να θέσει SQL ερωτήσεις και να λάβει απαντήσεις από τους peers που έχουν σχετικά με την ερώτηση δεδομένα. Το RDPeer προσπαθεί να επιλύσει κάποιες σημαντικές διαστάσεις που επηρεάζουν το πρόβλημα του διαμοιρασμού των δεδομένων με την δυναμική δημιουργία των super-peers μέσω εκλογών. Οι super-peers συμβάλλουν στην επεκτασιμότητα του διαμοιρασμού γνώσης, στην αξιοπιστία του συστήματος, και στην καλύτερη απόδοση της αναζήτησης δεδομένων.

Για να γίνει εφικτός ο διαμοιρασμός των δεδομένων, κάθε peer του συστήματος κοινοποιεί προς το super-peer το σχήμα της βάσης δεδομένων και μεταδεδομένα που περιγράφουν τα ονόματα των σχέσεων και πεδίων του σχήματος με τη μορφή XML εγγράφων. Τα μεταδεδομένα αποτελούν συνώνυμα των ονομάτων των σχέσεων και των πεδίων και παράγονται από κάθε peer είτε δυναμικά με τη χρήση ηλεκτρονικών λεξικών, είτε με τη βοήθεια του χρήστη. Κάθε φορά που ένας peer δημιουργεί μια SQL ερώτηση την απευθύνει στο super-peer. Ο super-peer αναλαμβάνει τη δημιουργία αντιστοιχίσεων μεταξύ των σχημάτων των peers και της SQL ερώτησης. Οι αντιστοιχίσεις δημιουργούνται, ώστε ο super-peer να διενεργεί μετασχηματισμό των SQL ερωτήσεων, με σκοπό να μη γίνεται η εξαγωγή των δεδομένων κεντροποιημένα, αλλά με peer-to-peer τρόπο.

Το RDPeer είναι ένα υβριδικό peer-to-peer σύστημα αποτελούμενο από συμπλέγματα από peers πάνω από την πλατφόρμα JXTA [21], όπου κάθε peer είναι υλοποιημένος με βάση το μοντέλο πελάτη/εξυπηρετητή. Κάθε σύμπλεγμα αποτελεί μια κοινότητα από peers, στην οποία υπάρχει ένας super-peer για να διεξάγει την αναζήτηση της πληροφορίας και το μετασχηματισμό των ερωτήσεων. Ένας peer μπορεί να ανήκει ταυτόχρονα σε περισσότερες της μίας κοινότητες. Οι κοινότητες είναι αυθαίρετα δίκτυα από peers και δεν οργανώνονται με βάση το σημασιολογικό περιεχόμενο των δεδομένων [9,35] που φέρουν τα μέλη τους.

Βασικός σκοπός της τεχνολογίας JXTA είναι η παροχή μιας πλατφόρμας με τις βασικές λειτουργίες που είναι απαραίτητες για ένα δίκτυο peer-to-peer, ενώ παράλληλα προσφέρει τη δυνατότητα σε μια ευρύτερη ποικιλία από υπηρεσίες, συσκευές, και δίκτυα μεταφοράς να συμμετέχουν σε δίκτυα peer-to-peer. Η επιλογή της υβριδικής αρχιτεκτονικής προσφέρει τη δυνατότητα της αποδοτικότερης αναζήτησης και δημιουργίας των αντιστοιχίσεων, καθώς και του μετασχηματισμού της SQL ερώτησης με τη μικρότερη επιβάρυνση της κίνησης του δικτύου.

Για να μην έχουμε την εμφάνιση αναπάντητων SQL ερωτήσεων, εξαιτίας της αποτυχίας του super-peer, υλοποιήσαμε έναν αλγόριθμο εκλογής ο οποίος συμβάλει στην

αξιοπιστία, συνέπεια και διατήρηση της αρχιτεκτονικής του peer-to-peer συστήματος. Επιπλέον οι εκλογές σχεδιάστηκαν και υλοποιήθηκαν, έτσι ώστε να επιτευχθεί η καλύτερη μελλοντική απόδοση του συστήματος.

### 1.3 Περιεχόμενο της εργασίας

Η αναφορά οργανώνεται ως εξής. Στο δεύτερο κεφάλαιο αναφερόμαστε στις σχετικές εργασίες που έχουν αναπτυχθεί στον τομέα της διαχείρισης και διαμοιρασμού των δεδομένων σε peer-to-peer συστήματα. Στο κεφάλαιο αυτό αναπτύσσονται οι διαφορετικές αρχιτεκτονικές και οι τρόποι αναζήτησης πληροφορίας που υπάρχουν στα peer-to-peer συστήματα μέσω της εξέτασης των πιο δημοφιλών συστημάτων που έχουν αναπτυχθεί με βάση κάθε μια αρχιτεκτονική. Επιπλέον γίνεται αναφορά σε ορισμένα συστήματα που στοχεύουν στο διαμοιρασμό σχεσιακών δεδομένων και αναλύεται πως επιτυγχάνεται αυτός ο στόχος.

Στο τρίτο κεφάλαιο αναλύεται η πλατφόρμα JXTA, που αποτελεί τη βάση πάνω στην οποία αναπτύσσεται η εφαρμογή RDPeer. Στο σημείο αυτό γίνεται λεπτομερής περιγραφή του τρόπου λειτουργίας και αιτιολογείται, γιατί επιλέχτηκε και ποια πλεονεκτήματα προσφέρει έναντι άλλων τεχνολογιών.

Στο τέταρτο κεφάλαιο γίνεται λεπτομερής αναφορά στην εφαρμογή RDPeer που υλοποιήθηκε. Αναλύεται ο τρόπος λειτουργίας της εφαρμογής, η αρχιτεκτονική του συστήματος, η αρχιτεκτονική κάθε peer, η τεχνική με την οποία πραγματοποιείται η αναζήτηση και εύρεση αντιστοιχίσεων μεταξύ της SQL επερώτησης και των σχημάτων των peers. Επίσης, αναλύεται η τεχνική με την οποία γίνεται ο μετασχηματισμός της επερώτησης, η διενέργεια εκλογών μεταξύ των peers για να αναδειχτεί ο νέος super-peer και οι τεχνολογίες που χρησιμοποιήθηκαν στην υλοποίηση της εφαρμογής.

Τέλος στο πέμπτο κεφάλαιο αναφέρονται τα συμπεράσματα που προκύπτουν από την παρούσα εργασία και γίνεται λόγος για τις μελλοντικές επεκτάσεις που θα μπορούσαν να πραγματοποιηθούν.

## 2 Σχετικές εργασίες

Η τεχνολογία peer-to-peer, η οποία ονομάζεται και υπολογισμός βάση ισοτιμίας, έχει λάβει πρόσφατα μεγάλη προσοχή τόσο από τον ερευνητικό χώρο όσο και από τη βιομηχανία εξαιτίας των πολλών πλεονεκτημάτων που προσφέρουν. Σε ένα καταναμημένο σύστημα peer-to-peer, ένας μεγάλος αριθμός από κόμβους μπορούν να συνυπάρξουν για να διαμοιράσουν τους πόρους, τις πληροφορίες και τις υπηρεσίες που έχουν. Οι κόμβοι που μπορούν να καταναλώνουν, αλλά και να παρέχουν δεδομένα και υπηρεσίες, είναι ικανοί να εισέρχονται και να εγκαταλείπουν το peer-to-peer δίκτυο οποιαδήποτε χρονική στιγμή, με αποτέλεσμα να δημιουργείται ένα δυναμικό και αυθαίρετο περιβάλλον. Πολλά peer-to-peer συστήματα έχουν ήδη αναπτυχθεί [32], όπως είναι για παράδειγμα τα Freenet[11], Gnutella [15], KaZaA [23], Napster [28] και Seti@home [38]. Όμως τα περισσότερα συστήματα παρέχουν δυνατότητες διαμοιρασμού αρχείων αγνοώντας τη σημασία των δεδομένων.

Παρά το γεγονός ότι τα peer-to-peer δίκτυα έχουν μια μεγάλη ποικιλία από εφαρμογές, υπάρχουν ακόμη πεδία εφαρμογών τα οποία δεν έχουν αναπτυχθεί πολύ, όπως είναι ο διαμοιρασμός σχεσιακών δεδομένων. Η προσπάθεια που πραγματοποιείται στοχεύει στην υποστήριξη των βάσεων δεδομένων και το διαμοιρασμό των σχεσιακών δεδομένων σε ένα peer-to-peer περιβάλλον. Για να επιτευχθεί ο στόχος, κάθε peer που φέρει μια βάση δεδομένων πρέπει να είναι ικανός να διαμοιράζεται ένα μέρος της βάσης δεδομένων, να μπορεί να στέλνει επερωτήσεις στο δίκτυο, να απαντάει και να δρομολογεί αποτελεσματικά τις επερωτήσεις που φθάνουν από το δίκτυο. Προς αυτή την κατεύθυνση έχουν οδηγηθεί αρκετές μελέτες [1,3,5,7,13,16,30].

### 2.1 Peer-to-Peer συστήματα

Τα peer-to-peer συστήματα διαμοιρασμού αρχείων είναι ιδιαίτερα δημοφιλή και συγκεντρώνουν μεγάλη ερευνητική δραστηριότητα τα τελευταία χρόνια. Τα συστήματα καταναίμουν το κύριο κόστος του διαμοιρασμού των αρχείων, δηλαδή το χώρο στο δίσκο που απαιτείται για την αποθήκευση των αρχείων και το εύρος ζώνης (bandwidth) που είναι αναγκαίο για τη μεταφορά τους, ανάμεσα στους peers του δικτύου και επομένως τα συστήματα μπορούν να κλιμακώνονται χωρίς την ανάγκη για ισχυρούς και ακριβούς εξυπηρετητές (servers). Χαρακτηριστικά παραδείγματα τέτοιων συστημάτων είναι τα πολύ γνωστά σε όλους Napster, Gnutella, και Freenet. Τα συστήματα είναι καταναμημένες εφαρμογές που δε χαρακτηρίζονται από το κλασικό μοντέλο πελάτη / εξυπηρετητή, αλλά χρησιμοποιούν μια διαφορετική προσέγγιση στην υποδομή τους,

κατά την οποία όλοι οι μέτοχοι σε αυτήν είναι ισότιμα μέλη. Δηλαδή έχουν τη δυνατότητα να ενεργούν και ως πελάτες, αλλά και ως εξυπηρετητές αιτήσεων.

Η συμμετοχή σε ένα peer-to-peer σύστημα είναι εθελοντική και δυναμική. Αποσκοπεί στη συνεργασία των peers για την επίτευξη της εκάστοτε ζητούμενης υπηρεσίας και στοχεύει στην ισότιμη συνεισφορά διαθέσιμων πόρων από όλους. Οι προκλήσεις που προκύπτουν κατά το σχεδιασμό τέτοιων συστημάτων, σχετίζονται με την ανάπτυξη των κατάλληλων μηχανισμών, ώστε να επιτυγχάνεται η καλύτερη δυνατή οργάνωση των peers με στόχο τη βελτίωση της ποιότητας της υπηρεσίας. Επιπλέον το κλειδί της χρηστικότητας ενός peer-to-peer συστήματος διαμοιρασμού δεδομένων είναι οι αποδοτικές τεχνικές για αναζήτηση και απόκτηση των δεδομένων.

Η καλύτερη τεχνική αναζήτησης για ένα σύστημα εξαρτάται από τις ανάγκες της εφαρμογής. Για συστήματα αποθήκευσης ή αρχειακά, τα οποία στοχεύουν στην διαθεσιμότητα, τεχνικές αναζήτησης, όπως περιγράφονται στα [11,35,36,40,48], θεωρούνται πολύ καλές, επειδή εγγυώνται την εύρεση του περιεχομένου μέσα σε ένα ορισμένο αριθμό από μεταβάσεις μεταξύ κόμβων (hops). Οι τεχνικές αυτές ελέγχουν πολύ στενά την τοποθέτηση των δεδομένων και την τοπολογία του δικτύου και προς το παρόν υποστηρίζουν μόνο την αναζήτηση βασισμένη σε κάποιο αναγνωριστικό.

Υπάρχουν διάφοροι τύποι από peer-to-peer συστήματα, τα οποία απεικονίζουν ποικίλους βαθμούς κεντρικοποίησης. Στα «καθαρά» (pure) συστήματα, όπως είναι το Gnutella και το Freenet όλοι οι peers έχουν ισοδύναμους ρόλους και ευθύνες σε όλα τα θέματα. Σε ένα υβριδικό σύστημα, όπως είναι ο Napster, η αναζήτηση πραγματοποιείται σε ένα κεντρικοποιημένο φάκελο, αλλά η μεταφορά αρχείων εξακολουθεί να συμβαίνει με ισότιμο τρόπο. Τα super-peer δίκτυα όπως είναι το KaZaA αποτελούν μια ενδιάμεση κατάσταση μεταξύ των υβριδικών και καθαρών συστημάτων. Οι πελάτες αποστέλλουν ερωτήσεις στο super-peer και λαμβάνουν απαντήσεις από αυτόν, όπως και σε ένα υβριδικό σύστημα. Όμως οι super-peers είναι επίσης συνδεδεμένοι μεταξύ τους όπως είναι οι peers σε ένα pure σύστημα. Επομένως οι super-peers αναλαμβάνουν να δρομολογήσουν τα μηνύματα, να υποβάλλουν και να απαντήσουν τις ερωτήσεις για λογαριασμό των πελατών τους.

Παρακάτω θα αναλύσουμε τα πιο δημοφιλή συστήματα με βάση την κατηγοριοποίηση που κάναμε για τα peer-to-peer συστήματα. Θα αναλύσουμε σε ότι αφορά τα υβριδικά συστήματα την αρχιτεκτονική του Napster, για τα pure συστήματα θα αναφερθούμε στο Gnutella και το Freenet, και τέλος για το πεδίο των super-peer δικτύων θα αναλύσουμε το KaZaA.

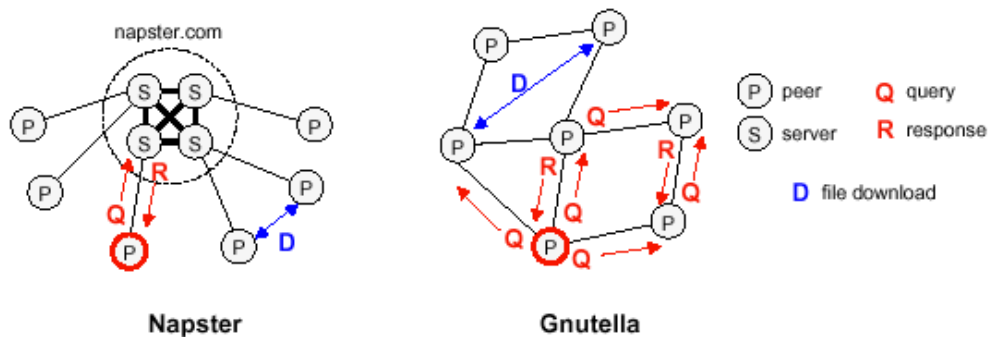
### 2.1.1 Αρχιτεκτονική υβριδικών συστημάτων



Το πιο δημοφιλές peer-to-peer σύστημα που ακολουθεί την υβριδική αρχιτεκτονική είναι το Napster. Το Napster έχει ως σκοπό να παρέχει στα μέλη του υπηρεσίες ανεύρεσης και ανταλλαγής αρχείων. Τα αρχεία αυτά βρίσκονται στους τοπικούς δίσκους του κάθε μέλους σε κατάλογο, του οποίου τα περιεχόμενα είναι διαθέσιμα στα υπόλοιπα μέλη. Η ανταλλαγή γίνεται με απευθείας σύνδεση δύο μελών βασισμένη σε ένα HTTP πρωτόκολλο ανταλλαγής αρχείων. Οι peers είναι συμμετρικοί, με την έννοια ότι έχουν τη δυνατότητα και να εξυπηρετήσουν εισερχόμενες αιτήσεις, λειτουργώντας ως εξυπηρετητές, αλλά και να δημιουργήσουν αιτήσεις για να αποκτήσουν κάποιο αρχείο, λειτουργώντας ως πελάτες.

Στο Napster, υπάρχει μια συνομοταξία από περίπου 160 κεντρικούς εξυπηρετητές. Κάθε μέλος - peer όπως φαίνεται στην εικόνα 1 συνδέεται με έναν από τους κεντρικούς εξυπηρετητές και διατηρεί αυτή τη σύνδεση καθ' όλη τη διάρκεια της συμμετοχής του στο σύστημα. Μέσω αυτού προωθούνται οι εκάστοτε αιτήσεις αναζήτησης, ενώ εκεί διατηρούνται και πληροφορίες σχετικά με τα αρχεία που διατίθενται από αυτόν. Εκτός τούτου ο εκάστοτε κεντρικός εξυπηρετητής καταγράφει μεταδεδομένα σχετικά με τον κάθε peer, με τον οποίο είναι συνδεδεμένος. Αυτά είναι πληροφορίες που αναφέρονται από τον peer κατά την εισαγωγή του στο σύστημα (αν το επιθυμεί) και περιέχουν το μέγεθος του διαθέσιμου bandwidth του peer, τη συνολική διάρκεια την οποία είναι συνδεδεμένος και φυσικά πληροφορίες για τα διαθέσιμα αρχεία του.

Το Napster είναι ένα υβριδικό σύστημα, το οποίο παρουσιάζει κάποια προβλήματα. Ενώ η κεντροκοποιημένη αναζήτηση είναι γενικά πιο αποδοτική από την κατακευματισμένη αναζήτηση σε ότι έχει να κάνει με το συνολικό κόστος, το κόστος που προσμετράται στο κόμβο που φιλοξενεί τον κεντροκοποιημένο δείκτη είναι πολύ μεγάλο. Αυτό έχει σαν αποτέλεσμα να υπάρχει ένα τελικό σημείο σε ότι αφορά την απόδοση και κλιμακοσιμότητα του συστήματος. Επιπλέον, όπως ισχύει για τα υβριδικά συστήματα έτσι και ο Napster είναι ευάλωτος σε επιθέσεις, καθώς υπάρχουν μόνο λίγοι ορατοί στόχοι που θα επιφέρουν την αποτυχία του συνολικού συστήματος.



Εικόνα 1 Αριστερά : απεικονίζεται ένα μέρος της αρχιτεκτονικής του Napster. Δεξιά : μέρος του δικτύου της Gnutella

## 2.1.2 Αρχιτεκτονική καθαρών συστημάτων

Στο πλαίσιο των καθαρών peer-to-peer συστημάτων θα εξετάσουμε τα συστήματα Gnutella και Freenet, τα οποία αποτελούν αντιπροσωπευτικά αυτής της κατηγορίας.

Το Gnutella (εικόνα 1) σε αντίθεση με το Napster έχει ακολουθήσει μια διαφορετική προσέγγιση στον τρόπο προώθησης των αιτήσεων. Εδώ ουσιαστικά σχηματίζεται ένα είδος «εικονικού» δικτύου, που αποτελείται από τους συνδεδεμένους μεταξύ τους peers, χωρίς να υπάρχει κεντρικός εξυπηρετητής, δημιουργώντας έτσι μια πιο «καθαρή» peer-to-peer υποδομή. Δύο κόμβοι οι οποίοι διατηρούν μια ανοιχτή σύνδεση μεταξύ τους ονομάζονται γείτονες (neighbors). Ο αριθμός των γειτόνων που έχει ένας κόμβος ονομάζεται *outdegree*. Τα μηνύματα δρομολογούνται μόνο διαμέσου των ανοιχτών συνδέσεων. Εάν ένα μήνυμα χρειάζεται να ταξιδέψει μεταξύ δύο κόμβων που δεν είναι γείτονες, τότε θα πρέπει να ταξιδέψει πάνω από πολλαπλές συνδέσεις. Το μήκος του μονοπατιού που διασχίζει ένα μήνυμα είναι γνωστό ως αριθμός από *hops*.

Ο κάθε peer είναι ελεύθερος να μπει ή να βγει από το δίκτυο των peers οποιαδήποτε στιγμή το επιθυμεί. Η είσοδος ενός peer στο δίκτυο προϋποθέτει τη γνώση ενός άλλου κόμβου που είναι ήδη στο δίκτυο. Για να διατηρείται όμως η ύπαρξη του δικτύου, το πρωτόκολλο που υποστηρίζει το Gnutella περιλαμβάνει και μηνύματα ring και pong. Τα πρώτα χρησιμοποιούνται για την ανεύρεση νέων peers και η εκτέλεση τους από έναν peer γίνεται σε τακτά χρονικά διαστήματα ή κατά την πρώτη εισαγωγή του στο δίκτυο. Τα pong μηνύματα αντίστοιχα είναι οι απαντήσεις στα μηνύματα ring που λαμβάνονται. Ενίοτε οι peers, που συμμετέχουν στο δίκτυο, ανοίγουν και νέες συνδέσεις με peers τους οποίους έχουν προηγουμένως εντοπίσει από την ανταλλαγή ring/pong μηνυμάτων. Στο Gnutella είναι δυνατόν να υπάρχουν διαφορετικά, μη συνδεδεμένα υποδίκτυα από peers. Δεν υπάρχει δηλαδή κάποιος μηχανισμός που να εξασφαλίζει ότι τελικά όλα τα μέλη μέσω κάποιου μονοπατιού θα είναι συνδεδεμένα.

Αναλυτικότερα κάθε μέλος δημιουργεί μία από σημείο σε σημείο σύνδεση με ένα υποσύνολο από κάποια άλλα ήδη ενεργά μέλη του συστήματος (γείτονες peers). Κάθε αίτηση αναζήτησης που λαμβάνει ένας peer την προωθεί με τη σειρά του στους γείτονες του κ.ο.κ. Έτσι δημιουργείται μια «πλημμύρα» από προωθούμενες αιτήσεις, η οποία ελέγχεται και τερματίζει, χάρη σε ένα πεδίο που περιέχεται σε κάθε αίτηση και είναι αντίστοιχο με το time-to-live (TTL) πεδίο που περιέχεται στα IP πακέτα. Αν στο μεταξύ υπάρξει και επιτυχία στην αναζήτηση αυτή επιστρέφει στον αρχικό αιτούντα μέσω του μονοπατιού, που έφτασε στον παραλήπτη, ενώ ταυτόχρονα συνεχίζεται και η προώθησή του.

Επομένως η τεχνική αναζήτησης η οποία χρησιμοποιείται στο Gnutella είναι η διάσχιση κατά πλάτος πρώτα του δικτύου (breadth-first traversal-BFS) με όριο βάθους  $D$

όπου το  $D$  είναι το μέγιστο time-to-live του μηνύματος και το οποίο μετριέται σε hops. Άλλα πρωτόκολλα δρομολόγησης όπως είναι αυτά που περιγράφονται στα [10,47] μπορούν να στείλουν την ερώτηση σε ένα επιλεγμένο υποσύνολο από γείτονες για λόγους καλύτερης απόδοσης.

Για να αποφευχθεί η επανάληψη αποστολής ενός προηγούμενου μηνύματος από ένα κόμβο του Gnutella, ανατίθενται στα μηνύματα μοναδικά αναγνωριστικά. Κάθε φορά που ένα μήνυμα διασχίζει το δίκτυο, το αναγνωριστικό του μηνύματος αποθηκεύεται από τους κόμβους από τους οποίους περνάει. Εάν υπάρχουν βρόγχοι στο δίκτυο τότε είναι πιθανόν ένας κόμβος να λάβει το ίδιο μήνυμα δύο φορές. Τότε το μήνυμα δε μεταδίδεται ξανά, καθώς το αναγνωριστικό του ήδη προϋπάρχει στον κόμβο.

Στο Gnutella η τοποθεσία της πηγής στην οποία βρίσκονται τα δεδομένα, δεν είναι γνωστή στον αποκρινόμενο κόμβο. Στην περίπτωση αυτή, το μήνυμα απάντησης θα δρομολογηθεί πίσω κατά μήκος του αντίστροφου μονοπατιού που διέσχισε το μήνυμα ερώτησης. Εάν όμως η τοποθεσία της πηγής είναι γνωστή, τότε μπορεί να δημιουργηθεί μια προσωρινή σύνδεση και να μεταφερθούν τα δεδομένα απευθείας. Ενώ η πρώτη μέθοδος χρησιμοποιεί περισσότερο bandwidth από τη δεύτερη, αποφεύγει το «βομβαρδισμό» της πηγής με αιτήσεις σύνδεσης όπως κάνει η δεύτερη μέθοδος και παρέχει επίσης επιπρόσθετα οφέλη όπως είναι η ανωνυμία.

Το Gnutella εμφανίζει πολύ καλή προσαρμοστικότητα στην αλλαγή του πληθυσμού των κόμβων. Επιπλέον, υπάρχει χαμηλό κόστος στη δρομολόγηση της πληροφορίας. Ένα άλλο θετικό στοιχείο που παρουσιάζεται είναι η μη ύπαρξη περιορισμού στην αποθήκευση, καθώς κάθε peer αποθηκεύει κλειδιά των αρχείων του. Το πρόβλημα όμως που εμφανίζεται είναι απόδοση της αναζήτησης αφού ο αριθμός των μηνυμάτων που πρέπει να ανταλλαχθούν είναι πολύ μεγάλος. Επιπρόσθετα σημαντικό είναι και το πρόβλημα του *free riding*, καθώς πολύ peers είτε καταναλώνουν πόρους χωρίς να παρέχουν τίποτα, είτε προσφέρουν δεδομένα που δεν έχουν κανένα ενδιαφέρον. Μια άλλη πηγή αναποτελεσματικότητας όπως έδειξε η μελέτη [42] οφείλεται σε peers που είναι περιορισμένων δυνατοτήτων, καθώς μπορεί να συνδέονται με συνδέσμους χαμηλής χωρητικότητας στο δίκτυο, και επιφέρουν την τμηματοποίηση του δικτύου κατά την αποχώρησή τους. Οι peers περιορισμένων δυνατοτήτων «πεθαίνουν» εξαιτίας των ίσων ρόλων και ευθυνών που τους ανατίθενται ανεξαρτήτως των δυνατοτήτων τους. Όμως μελέτες όπως οι [2,37] έχουν δείξει σημαντική ετερογένεια ανάμεσα στις δυνατότητες των συμμετεχόντων peers.

Το Freenet όπως και το Gnutella είναι κατακεκολλημένο. Το Freenet μπορεί να περιγραφεί σαν ένα δίκτυο διαμοιρασμού bandwidth και χώρου στους δίσκους, ενώ το Gnutella είναι ένα δίκτυο αναζήτησης και ανακάλυψης το οποίο προάγει την ελεύθερη μετάφραση και απάντηση στις ερωτήσεις. Το Freenet είναι ένα προσαρμόσιμο peer-to-peer σύστημα το οποίο υποστηρίζει την έκδοση, αντιγραφή, και ανάκτηση των δεδομένων. Επιπλέον διαφοροποιείται από το Gnutella στα παρακάτω θέματα:

1. Έχει έναν αποδοτικό μηχανισμό αναζήτησης, που ελέγχει τον αριθμό των μηνυμάτων που δημιουργούνται.
2. Έχει ένα μηχανισμό αντιγραφής για τη διασπορά των δεδομένων στο δίκτυο και με τον τρόπο αυτό αυξάνεται η απόδοση του συστήματος.
3. Παρέχει ανωνυμία. Οι peers δε γνωρίζουν τι αποθηκεύουν καθώς τα δεδομένα είναι κρυπτογραφημένα.

Η τεχνική αναζήτησης που χρησιμοποιείται είναι η αναζήτηση σε βάθος πρώτα (depth-first traversal-DFS) με όριο βάθους  $D$ . Κάθε κόμβος προωθεί την ερώτηση σε ένα μόνο γείτονα και περιμένει για μια οριστική απάντηση από αυτόν πριν προωθήσει την ερώτηση σε άλλο γείτονα, ή προωθεί τα αποτελέσματα πίσω στην πηγή της ερώτησης. Με την DFS τεχνική, επειδή κάθε κόμβος επεξεργάζεται την ερώτηση σειριακά, η αναζήτηση μπορεί να τερματιστεί όταν η ερώτηση θα έχει ικανοποιηθεί ελαχιστοποιώντας έτσι το κόστος, σε ότι αφορά το πλήθος των μηνυμάτων.

Οι peers στο Freenet διατηρούν επίσης ένα πίνακα δρομολόγησης για τους γειτονικούς peers. Αυτοί όμως αποθηκεύουν επιπρόσθετα το κλειδί των δεδομένων καθώς και τα αντίστοιχα δεδομένα. Έτσι όταν φθάνει μια αίτηση αναζήτησης σε ένα peer, είτε θα μπορεί να απαντηθεί απευθείας από τα δεδομένα που βρίσκονται στον πίνακα ή θα προωθήσει την αίτηση σε κάποιον άλλο peer. Αυτό γίνεται επιλέγοντας εκείνον τον peer που έχει το πιο παρόμοιο κλειδί. Όταν φθάσει μια απάντηση, ο peer την αποθηκεύει στην δική του πηγή δεδομένων. Εάν η πηγή είναι ήδη γεμάτη τότε θα πρέπει να διαγράψει κάποια άλλα αποθηκευμένα δεδομένα.

Η στρατηγική που ακολουθείται στο σημείο αυτό είναι η διαγραφή των λιγότερα πρόσφατα χρησιμοποιούμενων δεδομένων. Επομένως ύστερα από ένα σημείο ο κόμβος δεν αποθηκεύει τα δεδομένα που σχετίζονται με ένα κλειδί, αλλά μόνο τη διεύθυνση. Τα έγγραφα που έχουν μεγαλύτερη ζήτηση μετακινούνται προς τα πάνω αντικαθιστώντας τα λιγότερα ζητούμενα. Η απόσταση προς τα πάνω που μετακινούνται τα έγγραφα συνδέεται με το μέγεθός τους, έτσι το μεγάλο μέγεθος ενός εγγράφου αποτελεί ένα μειονέκτημα. Συνεπώς, το Freenet προτρέπει τους χρήστες να μη σπαταλούν χώρο στο δίσκο και για το λόγο αυτό να συμπιέζουν τα έγγραφα πριν τα εισάγουν. Στην εικόνα 2 φαίνεται ο πίνακας δρομολόγησης που χρησιμοποιεί κάθε κόμβος του συστήματος.

Key	Data	Address
8e47683isdd0932uje89	ZT38hwe01h02hdhgdzu	tcp/125.45.12.56:6474
456r5wero04d903iksd0	Rhwewui12340jhd091230	tcp/67.12.4.65:4711
f3682jkjdn9ndaqmmxia	eqwe1089341ih0zuhge3	tcp/127.156.78.20:8811
wen09hjfdh03uhn4218	erwq038382hjh3728ee7	tcp/78.6.6.7:2544
712345jb89b8nbopledh		tcp/40.56.123.234:1111
d0ui43203803ujoejqhh		tcp/128.121.89.12:9991

Εικόνα 2 Παράδειγμα πίνακα δρομολόγησης που διατηρεί κάθε κόμβος του Freenet.

Η εισαγωγή των δεδομένων στο Freenet είναι παρόμοια με την αναζήτηση. Για να εισαχθεί ένα καινούργιο έγγραφο στο σύστημα θα πρέπει πρώτα να υπολογιστεί ένα κλειδί για το αρχείο που να μην υπάρχει ήδη. Στη συνέχεια αποστέλλεται ένα μήνυμα εισαγωγής που περιέχει το κλειδί με μία hops-to-live τιμή στους γειτονικούς κόμβους. Κάθε peer που λαμβάνει το μήνυμα ελέγχει εάν έχει ήδη το κλειδί στην τοπική cache που διατηρεί. Εάν το κλειδί βρεθεί, τότε επιστρέφεται το αποθηκευμένο αρχείο διαφορετικά η δρομολόγηση συνεχίζεται μέχρι η τιμή hops-to-live να γίνει μηδέν. Εάν έχει γίνει η τιμή μηδέν και δεν έχει ανακαλυφθεί το ίδιο κλειδί, το έγγραφο αποστέλλεται κατά μήκος του ίδιου μονοπατιού και κάθε κόμβος που λαμβάνει το μήνυμα το αποθηκεύει.

Όπως συμβαίνει στο Gnutella έτσι και στο Freenet οι αιτήσεις αναζήτησης έχουν ένα χρονικό όριο ζωής (TTL). Τα μηνύματα επίσης χαρακτηρίζονται από μοναδικά αναγνωριστικά ώστε να αποφεύγονται οι επαναμεταδόσεις μηνυμάτων. Επιπλέον, ταύτιση των δύο συστημάτων υπάρχει και στο γεγονός ότι για να εισέλθει ένας κόμβος στο δίκτυο θα πρέπει να γνωρίζει κάποιον αρχικό κόμβο. Το κύριο πλεονέκτημα της προσέγγισης του Freenet προέρχεται από την στρατηγική δρομολόγησης που είναι κατά βάθος πρώτα και η ποιότητα της δρομολόγησης θα πρέπει συνεχώς να βελτιώνεται καθώς η στρατηγική caching σχεδιάστηκε με τρόπο που να τείνει να ομαδοποιήσει παρόμοια δεδομένα σε κόμβους του δικτύου και επομένως οι κόμβοι γίνονται συνεχώς περισσότερο ειδικευμένοι.

Το μειονέκτημα του Freenet έναντι του Gnutella, είναι ότι επιτρέπει μόνο την ακριβή ταύτιση των κλειδιών στην αναζήτηση ενός εγγράφου και συνεπώς δε γίνεται καθόλου αναζήτηση βάση περιεχομένου. Αυτό πρακτικά σημαίνει ότι για να βρει ένας χρήστης το έγγραφο που ζητάει θα πρέπει να ζητήσει είτε αρχεία τα οποία έχει ο ίδιος αποθηκευμένα, είτε άλλα αρχεία για τα οποία έχει μάθει το όνομά τους.

### 2.1.3 Αρχιτεκτονική super-peer συστημάτων

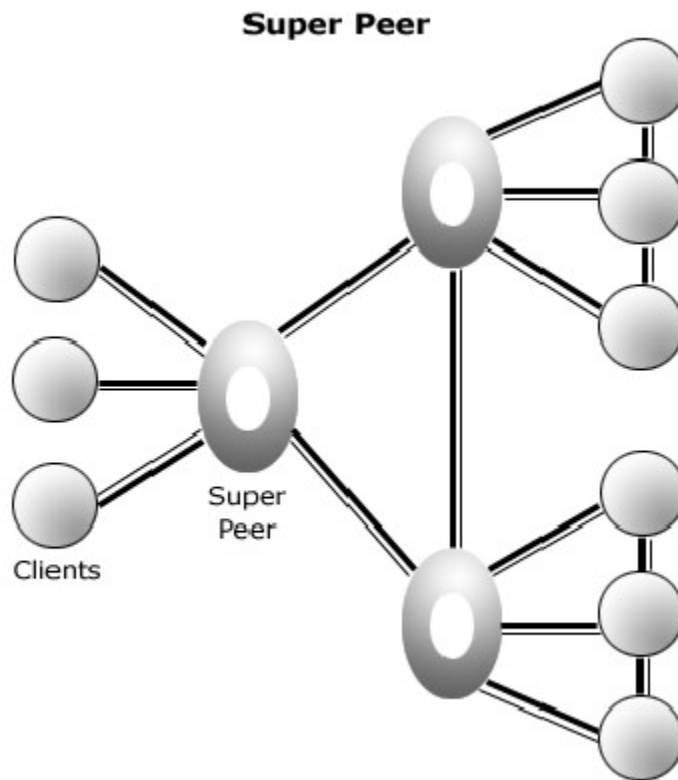
Τα super-peer συστήματα ακολουθούν μια αρχιτεκτονική που συνδυάζει στοιχεία που εμφανίζονται τόσο στα υβριδικά, όσο και στα καθαρά peer-to-peer συστήματα. Αντιπροσωπευτικό super-peer σύστημα αποτελεί το KaZaA, το οποίο και θα αναλύσουμε.

Το KaZaA είναι μια peer-to-peer εφαρμογή για το διαμοιρασμό των αρχείων βασισμένη στο πρωτόκολλο FastTrack. Το πρωτόκολλο FastTrack είναι βασισμένο στο Gnutella και το επεκτείνει με την προσθήκη των υπερκόμβων (supernodes), έτσι ώστε να βελτιώσει την κλιμακοσιμότητά του. Η λειτουργικότητα του supernode είναι

ενσωματωμένη σε κάθε κόμβο του δικτύου και επομένως κάθε κόμβος του δικτύου μπορεί να γίνει supernode κάποια στιγμή.

Για να μπορεί να συνδεθεί ένας κόμβος στο δίκτυο θα πρέπει αρχικά να έχει γνώση σχετικά με μια λίστα από γνωστούς supernodes. Αφού καταφέρει να συνδεθεί σε έναν από αυτούς μπορεί στη συνέχεια να μάθει και για άλλους σε περίπτωση μελλοντικών συνδέσεων. Συνδεόμενος ένας κόμβος στέλνει στο supernode μια λίστα με τα αρχεία που θέλει να μοιραστεί.

Όταν ένας κόμβος θέλει να στείλει αίτηση αναζήτησης, τότε αντίθετα με το Gnutella δε διαχειρίζεται ο ίδιος τη διαδικασία εύρεσης, αλλά αποστέλλει την αίτηση στο supernode. Ο supernode επικοινωνεί με τους υπόλοιπους supernodes, έτσι ώστε να ικανοποιήσει την αίτηση αναζήτησης. Στη συνέχεια, ο κόμβος που αρχικοποίησε την αίτηση αναζήτησης συνδέεται απευθείας σε έναν peer για να λάβει τα δεδομένα. Η μεταφορά γίνεται με τη χρήση του πρωτοκόλλου HTTP.



Εικόνα 3 Αρχιτεκτονική super-peer δικτύου.

Ο super-peer όπως φαίνεται και στην εικόνα 3 είναι ένας peer που λειτουργεί σαν ένας κεντρικοποιημένος εξυπηρετητής για ένα σύνολο από πελάτες. Οι super-peers συνδέονται μεταξύ τους σχηματίζοντας ένα «καθαρό» peer-to-peer σύστημα και

αναλαμβάνουν να εξυπηρετήσουν τις αιτήσεις που στέλνουν οι πελάτες. Κάθε super-peer διατηρεί δύο πρωτόκολλα ένα για να επικοινωνεί με τους υπόλοιπους super-peers και ένα για να επικοινωνεί με τους πελάτες.

Ένα σύνολο από πελάτες μαζί με το super-peer σχηματίζουν ένα σύμπλεγμα (cluster). Η επικοινωνία μέσα σε ένα σύμπλεγμα πραγματοποιείται με απευθείας συνδέσεις ανάμεσα στους πελάτες. Η επικοινωνία όμως ενός πελάτη του συμπλέγματος με έναν άλλο πελάτη ενός διαφορετικού συμπλέγματος είναι εφικτή διαμέσου του super-peer. Συγκεκριμένα, ο έλεγχος ροής για τους κανονικούς peers συμβαίνει διαμέσου του super-peer.

Επειδή το super-peer δίκτυο συνδυάζει στοιχεία του υβριδικού συστήματος και του pure συστήματος παρουσιάζει περισσότερα πλεονεκτήματα. Συνδυάζει την απόδοση της κεντροποιημένης αναζήτησης με την αυτονομία, την ισορροπία φόρτου και την αντοχή σε επιθέσεις της καταναμημένης αναζήτησης. Για παράδειγμα, επειδή οι super-peers λειτουργούν σαν κεντροποιημένοι εξυπηρετητές για τους πελάτες τους, μπορούν να χειριστούν τις ερωτήσεις πιο αποδοτικά από ότι θα έκανε ένας πελάτης. Επιπλέον όμως, επειδή υπάρχουν αρκετοί super-peers στο δίκτυο κανένας super-peer δε χρειάζεται να αντιμετωπίσει ένα μεγάλο φόρτο εργασίας και επομένως να αποτελέσει σημείο αποτυχίας του συστήματος.

## 2.2 Διαχείριση και διαμοιρασμός δεδομένων σε P2P συστήματα

Ένα μείζον πρόβλημα στη σημερινή κοινωνία της πληροφορίας είναι ότι ο διαμοιρασμός ετερογενών, σημασιολογικά πλούσιων δεδομένων είναι αρκετά δύσκολος. Η δυσκολία έγκειται στο γεγονός ότι τα δεδομένα σε διαφορετικά σχήματα πρέπει με κάποιο τρόπο να αντιστοιχηθούν ή να συσχετιστούν, ενώ οι επερωτήσεις είναι πολύ πλουσιότερες.

Στα περισσότερα peer-to-peer συστήματα που έχουν σχεδιαστεί για διαμοιρασμό δεδομένων, όπως αυτά που παρουσιάστηκαν στην προηγούμενη ενότητα, οι δυνατότητες που προσφέρονται περιορίζονται μόνο στο διαμοιρασμό αρχείων, ενώ οι έννοιες των δεδομένων αγνοούνται. Για παράδειγμα, στο Gnutella οι ερωτήσεις που τίθενται περιορίζονται μόνο σε ένα αλφαριθμητικό το οποίο μπορεί να περιέχεται σε ένα αρχείο και σε ένα μονοπάτι φακέλου, δηλαδή υποστηρίζονται μόνο αναζητήσεις σε απλές τιμές αρχείων. Παρόμοια επεξεργασία με το Gnutella περιλαμβάνει και το Napster, του οποίου η αναζήτηση προσπαθεί να βρει ομοιότητες μεταξύ ενός αλφαριθμητικού και των ονομάτων των αρχείων. Στο Freenet ακολουθείται μία άλλη τεχνική η οποία περιορίζει την αναζήτηση βάση περιεχομένου ακόμα περισσότερο σε σχέση με το Gnutella. Αυτό συμβαίνει γιατί σε κάθε αρχείο ανατίθενται ένα κλειδί και επιτρέπεται μόνο η ακριβής ταύτιση των κλειδιών στην αναζήτηση ενός εγγράφου και συνεπώς δε γίνεται καθόλου

αναζήτηση βάση περιεχομένου. Στα καταναμημένα συστήματα τα οποία βασίζονται στο καταναμημένο *hash table* [35,36,40,48] δε συμβαίνει κάτι καλύτερο, καθώς μόνο ακριβής ταύτιση κλειδιού υποστηρίζεται. Από τη σκοπιά της διαχείρισης των δεδομένων, ο διαμοιρασμός των δεδομένων είναι πολύ περιορισμένος.

Στην σημερινή εποχή πιστεύουμε ότι όλο και περισσότεροι χρήστες των υπολογιστών αποθηκεύουν τα δεδομένα τους σε συστήματα διαχείρισης βάσεων δεδομένων. Το RDPeer σχεδιάστηκε και υλοποιήθηκε για να ικανοποιήσει τις ανάγκες των χρηστών που θέλουν να διαμοιράσουν τα δεδομένα που έχουν αποθηκευμένα στις βάσεις δεδομένων τους. Τα σχεσιακά δεδομένα έχουν πιο πλούσια δομή και επομένως η διαχείριση και ο διαμοιρασμός τους είναι μια σύνθετη διαδικασία.

Στη συνέχεια θα παρουσιαστούν κάποια σχεδιαστικά θέματα που πρέπει να λάβει υπόψη κανείς, καθώς και σχετικές έρευνες που έχουν διεξαχθεί.

### **2.2.1 Σχεδιαστικά θέματα για διαχείριση σχεσιακών δεδομένων σε peer-to-peer συστήματα**

Η διαχείριση και ο διαμοιρασμός σχεσιακών δεδομένων σε peer-to-peer σύστημα μπορεί να ειπωθεί σαν ένα καταναμημένο σύστημα ετερογενών βάσεων δεδομένων. Η κλιμακοσιμότητα όμως του συστήματος και η δυναμική φύση των κόμβων καθώς είτε γίνονται μέλη του δικτύου, ή το εγκαταλείπουν δημιουργεί ποικίλες προκλήσεις οι οποίες είναι:

- Στο σύστημα δεν υπάρχει ένα προκαθορισμένο συνολικό σχήμα. Εξαιτίας της δυναμικής φύσης κάθε κόμβου, η θεώρηση ενός συνολικού σχήματος είναι κάτι μη πρακτικό, και περιορίζει την κλιμακοσιμότητα και επεκτασιμότητα. Μία πιθανή προσέγγιση είναι να δημιουργηθούν αντιστοιχίσεις μεταξύ των σχημάτων των βάσεων δεδομένων που διαθέτουν οι κόμβοι του δικτύου κατά τη διάρκεια που τίθεται μια επερώτηση [30]. Μια εναλλακτική προσέγγιση είναι να επιτραπεί σε κάθε κόμβο να δημιουργήσει αντιστοιχίσεις μεταξύ του σχήματός του και αυτών που έχουν οι γειτονικοί κόμβοι πριν τεθούν οι επερωτήσεις [3,12,13,16,24].
- Η δημιουργία αντιστοιχίσεων εμπεριέχει το ζήτημα σχετικά με τον τρόπο που μπορούν να περιγραφούν, να δημιουργηθούν και να διατηρηθούν. Στον τομέα αυτό έχουν προταθεί διάφορες τεχνικές [34], είτε για χειρονακτικό τρόπο δημιουργίας, ή για αυτόματο.
- Στο σύστημα που θα δημιουργηθεί πρέπει να οριστεί ο τρόπος με τον οποίο πρέπει να γίνει η επεξεργασία του ερωτήματος. Ο χρόνος απόκρισης σε ένα ερώτημα αναμένεται να είναι μεγάλος καθώς πρώτα χρειάζεται να



αναγνωριστούν τα σχετικά δεδομένα προτού πραγματοποιηθεί η επεξεργασία και βελτιστοποίηση του ερωτήματος. Η επεξεργασία και η βελτιστοποίηση του ερωτήματος μπορεί να γίνει είτε κεντρικοποιημένα είτε κατανεμημένα από κάθε κόμβο του δικτύου με βάση της διαθέσιμη γνώση που κατέχει.

- Ένα σημαντικό στοιχείο που πρέπει να περιοριστεί είναι ο πλεονασμός της πληροφορίας και των υπολογισμών που διεξάγονται. Ο πλεονασμός της πληροφορίας δεν μπορεί να αποφευχθεί εκτός και αν υπάρξει κάποιος έλεγχος στην τοποθέτηση των δεδομένων όπως γίνεται στα συστήματα κατανεμημένου hash table. Ο πλεονασμός των υπολογισμών μπορεί να εμφανιστεί αν έχουμε κατανεμημένη επεξεργασία των ερωτήσεων, καθώς στην περίπτωση αυτή μεγάλος αριθμός κόμβων συμμετέχουν στην επεξεργασία ενός ερωτήματος.

### 2.2.2 Το σύστημα PeerDB

Το PeerDB [30] είναι ένα σύστημα για τον κατανεμημένο διαμοιρασμό δεδομένων, χωρίς την ύπαρξη ενός συνολικού σχήματος. Το σύστημα αποτελείται από διάφορα διακριτά στοιχεία. Πρώτον, κάθε κόμβος είναι ένα πλήρες σύστημα διαχείρισης, το οποίο υποστηρίζει αναζήτηση βάση περιεχομένου. Δεύτερον, στο PeerDB, οι χρήστες μπορούν να διαμοιραστούν δεδομένα χωρίς την ύπαρξη ενός συνολικού σχήματος. Τρίτον, το PeerDB υιοθετεί τη χρήση κινητών πρακτόρων για να βοηθήσει την επεξεργασία των δεδομένων και να χρησιμοποιήσει καλύτερα το bandwidth του δικτύου.

Κάθε peer που συμμετέχει στο σύστημα διαθέτει μια βάση δεδομένων και για κάθε ένα όνομα σχέσης και πεδίου που έχει δημιουργηθεί, ο χρήστης καλείται να δώσει κάποιες λέξεις κλειδιά που θα περιγράψουν καλύτερα το σχήμα της βάσης δεδομένων. Τα μεταδεδομένα αποτελούν ουσιαστικά συνώνυμα των ονομάτων των πεδίων και των σχέσεων. Επομένως, τα μεταδεδομένα που δημιουργούνται σε κάθε κόμβο περιλαμβάνουν μόνο τις λέξεις κλειδιά και αποθηκεύονται σε μια τοπική cache που διαθέτει κάθε peer.

Κάθε peer μπορεί να δημιουργήσει μια SQL επερώτηση. Για να μπορέσει να ικανοποιηθεί η επερώτηση χρησιμοποιούνται οι πράκτορες. Αρχικά, ο peer που δημιουργεί την επερώτηση ψάχνει στην τοπική cache που διαθέτει για να διαπιστώσει ποιες σχέσεις μπορούν να την ικανοποιήσουν. Για να βρει τις ομοιότητες μεταξύ των σχέσεων και του ερωτήματος χρησιμοποιείται μια τεχνική από την περιοχή ανάκτησης της ανάκτησης πληροφορίας (Information Retrieval) [4]. Με βάση την προσέγγιση αυτή διενεργείται έρευνα για να βρεθούν ομοιότητες μεταξύ των ονομάτων και των λέξεων κλειδιών που αναφέρονται στις σχέσεις και τα πεδία της βάσης δεδομένων με τα αντίστοιχα ονόματα των σχέσεων και πεδίων που αναφέρονται στην SQL επερώτηση.

Με τον τρόπο αυτό ένας peer είναι σε θέση να ανακαλύψει τις δικές του σχέσεις που μπορούν να ικανοποιήσουν την επερώτηση που τέθηκε.

Για να καταφέρει ο peer που θέτει την επερώτηση να λάβει μια συνολική απάντηση, θα πρέπει να εξετάσει και ποιοι άλλοι peers είναι σε θέση να την ικανοποιήσουν. Αυτό είναι εφικτό με τη χρήση των πρακτόρων. Με την τεχνική αυτή ο peer πετυχαίνει να ανακαλύψει ομοιότητες μεταξύ των δικών του ονομάτων των σχέσεων, πεδίων και των λέξεων κλειδιών που έχει ανακαλύψει ότι ικανοποιούν την επερώτηση και των αντίστοιχων ονομάτων ή λέξεων κλειδιών που φέρουν οι άλλοι peers. Για να γραφτεί ξανά η επερώτηση για κάθε έναν από τους υπόλοιπους peers για τους οποίους υπάρχουν ομοιότητες χρησιμοποιούνται οι λέξεις κλειδιά, καθώς αυτά είναι τα μόνα διαθέσιμα μεταδεδομένα μεταξύ διαφορετικών σχημάτων.

Αφού βρεθούν οι ομοιότητες επιστρέφονται στο χρήστη για να επιλέξει τις πιο κατάλληλες. Ύστερα από την επιλογή, η επερώτηση αποστέλλεται με τη χρήση των πρακτόρων μόνο στους peers που συμμετέχουν στην επιλογή. Οι peers εκτελούν την SQL επερώτηση και τα αποτελέσματα επιστρέφονται στον peer που αρχικοποίησε την επερώτηση, ώστε να εμφανιστούν στο χρήστη.

### 2.2.3 Το σύστημα Piazza

Το έργο Piazza [16] προσπαθεί να επιλύσει το πρόβλημα του διαμοιρασμού εννοιολογικά ετερογενών δεδομένων με ένα κατανεμημένο και κλιμακώσιμο τρόπο. Οι συμμετέχοντες στο Piazza είναι πηγές δεδομένων που ενδιαφέρονται να διαμοιράσουν δεδομένα. Κάθε συμμετέχων επιλέγει ανεξάρτητα, αλλά σχετιζόμενα σχήματα για τα δεδομένα του, και οι επερωτήσεις που θέτει βασίζονται στο σχήμα που έχει. Όπως και στο PeerDB δεν υπάρχει ένα συνολικό σχήμα, αλλά το σύστημα προσφέρει δυνατότητες απάντησης επερωτήσεων σε ένα αυθαίρετο δίκτυο από τοπικά σχήματα και ζευγάρια αντιστοιχίσεων μεταξύ αυτών. Για να απαντηθεί μία επερώτηση που τίθεται σε ένα σχήμα λαμβάνονται υπόψη οι αντιστοιχίσεις που έχουν δημιουργηθεί μεταξύ αυτού και των υπολοίπων, ώστε να ανακαλυφθούν όλα τα σχετικά δεδομένα.

Σε αντίθεση με ένα περιβάλλον ολοκλήρωσης δεδομένων, το οποίο έχει μια δένδροειδή ιεραρχία με τις πηγές δεδομένων στους κόμβους φύλλα και ένα ή περισσότερα ενδιάμεσα σχήματα σε ενδιάμεσους κόμβους, στο Piazza υποστηρίζεται ένας αυθαίρετος γράφος από διασυνδεδεμένα σχήματα. Στα [14,17] περιγράφεται ο τρόπος με τον οποίο αντικαθίσταται το μοναδικό λογικό σχήμα που υπάρχει στην ολοκλήρωση δεδομένων με μια συλλογή από σημασιολογικές αντιστοιχίσεις μεταξύ των σχημάτων των peers. Μερικά από αυτά τα σχήματα ορίζονται ουσιαστικά για το σκοπό των επερωτήσεων και των αντιστοιχίσεων.

Στο έργο Piazza υπάρχουν δύο τύποι από αντιστοιχίσεις. Υπάρχουν οι αντιστοιχίσεις που σχετίζουν δύο ή περισσότερα σχήματα διαφορετικών peers και ονομάζονται *peer description* και οι αντιστοιχίσεις που σχετίζουν ένα σχήμα που διαθέτει ένας peer με το αποθηκευμένο σχήμα και ονομάζονται *storage description*. Για να δημιουργηθεί η μεσολάβηση μεταξύ σχημάτων, το Piazza συνδυάζει δύο φορμαλισμούς της ολοκλήρωσης δεδομένων. Σύμφωνα με τον πρώτο, που ονομάζεται *global-as-view* (GAV) [12], οι σχέσεις στο ενδιάμεσο σχήμα ορίζονται σαν όψεις των σχέσεων των πηγών. Σύμφωνα με το δεύτερο, που ονομάζεται *local-as-view* (LAV) [25], οι σχέσεις στις πηγές ορίζονται σαν όψεις του ενδιάμεσου σχήματος. Για να γίνει η επεξεργασία των επερωτήσεων λαμβάνονται υπόψη οι αντιστοιχίσεις και η μεσολάβηση των σχημάτων [17].

Το Piazza σε αντίθεση με το PeerDB προσπαθεί να προκαθορίσει τις αντιστοιχίσεις μεταξύ των σχημάτων των γειτονικών peers, προτού τεθεί μια επερώτηση. Επιπλέον, στο PeerDB δε γίνεται προσπάθεια να δημιουργηθούν ενδιάμεσα σχήματα με τη μορφή ιδεατών όψεων όπως γίνεται στο έργο Piazza. Με τον τρόπο αυτό το PeerDB είναι πιο ευέλικτο, καθώς δε χρειάζεται να ανακατασκευάζει ενδιάμεσα σχήματα όπως συμβαίνει στο Piazza κάθε φορά που ένας peer είτε εγκαταλείπει το δίκτυο ή εισέρχεται σε αυτό. Στο PeerDB επίσης δε χρησιμοποιούνται οι αντιστοιχίσεις μεταξύ σχημάτων για τον επανασηματισμό των επερωτήσεων. Το Piazza θεωρεί πιο στατικό το δίκτυο και προσπαθεί ουσιαστικά μέσω των όψεων να επιλύσει το πρόβλημα της τοποθέτησης των δεδομένων.

#### 2.2.4 Το σύστημα Hyperion

Το έργο Hyperion [3] περιλαμβάνει τη δημιουργία ενός συστήματος διαχείρισης βάσης δεδομένων για peers. Η εργασία που έχει γίνει μέχρι στιγμής στοχεύει κυρίως στον καθορισμό και διαχείριση των λογικών μεταδεδομένων, που επιτρέπουν το διαμοιρασμό των δεδομένων και το συντονισμό ανάμεσα σε ανεξάρτητους και αυτόνομους peers.

Στο Hyperion προτείνεται η χρήση εκφράσεων αντιστοίχισης και πινάκων αντιστοίχισης [24] για να υποστηριχτεί η ανταλλαγή δεδομένων μεταξύ των peers. Οι εκφράσεις αντιστοίχισης σχημάτων είναι γενικεύσεις των GLAV εκφράσεων που μπορούν να δημιουργηθούν είτε χειρονακτικά είτε αυτόματα [33]. Με τις εκφράσεις αντιστοίχισης δηλώνονται οι συσχετίσεις που υπάρχουν μεταξύ σχέσεων που ανήκουν σε διαφορετικούς peers (για παράδειγμα μια σχέση  $x$  ενός peer  $i$  είναι υποσύνολο μιας σχέσης  $y$  ενός peer  $j$ ). Οι πίνακες αντιστοίχισεων δεν περιορίζουν το περιεχόμενο των βάσεων δεδομένων που διαθέτουν οι peers. Η χρήση των εκφράσεων γίνεται για να υπάρξουν περιορισμοί στην ανταλλαγή των δεδομένων. Αν και οι πίνακες

αντιστοιχίσεων παρέχουν στοιχειώδεις σχέσεις σε επίπεδο σχημάτων, η κύρια χρήση τους είναι για να δημιουργηθούν αντιστοιχίσεις ανάμεσα σε τιμές διαφορετικών peers. Αυτό αποτελεί ιδιαίτερης σημασίας θέμα στα peer-to-peer συστήματα όπου δεν υπάρχουν πρότυπα ονοματολογίας. Μειονέκτημα όμως αποτελεί για το έργο Hyreption το γεγονός ότι οι πίνακες αντιστοίχισης δημιουργούνται χειρωνακτικά.

Για να δοθεί η απάντηση σε μία επερώτηση που θέτει ένας peer θα πρέπει να συμβουλευτεί τους πίνακες αντιστοιχίσεων και τις εκφράσεις αντιστοιχίσεων που διαθέτει. Με τον τρόπο αυτό, ένας peer μπορεί να κάνει μετασχηματισμό της επερώτησης και να την κατευθύνει κατάλληλα μέσω των πινάκων και εκφράσεων αντιστοίχισης στους peers που διαθέτουν δεδομένα.

Εκτός από τις επερωτήσεις που μπορεί να θέτει κάθε peer που συμμετέχει στο δίκτυο, είναι δυνατόν οι peers να συντονίζουν τα δεδομένα τους με αυτά των γειτόνων τους. Γείτονες θεωρούνται δύο peers, όταν έχουν δημιουργηθεί μεταξύ τους πίνακες αντιστοιχίσεων και εκφράσεων. Για να πραγματοποιηθεί ο συντονισμός χρησιμοποιούνται κανόνες γεγονότος-συνθήκης-ενέργειας (event-condition-action – ECA) [13,22]. Με βάση τους κανόνες, όταν συμβαίνει ένα γεγονός σε μια βάση δεδομένων ενός peer, όπως για παράδειγμα η εισαγωγή ορισμένων πλειάδων σε μία σχέση, και με τη βοήθεια των πινάκων εκφράσεων είναι δυνατόν να πραγματοποιηθούν ορισμένες ενέργειες και στις σχέσεις των γειτονικών peers.

Γενικά υπάρχουν δύο σύνολα από κανόνες. Το πρώτο σύνολο περιλαμβάνει τους κανόνες που διαχειρίζονται τη συνέπεια μεταξύ των δεδομένων δύο peers. Αυτοί οι κανόνες δημιουργούνται αυτόματα κατά τη διάρκεια της φάσης εύρεσης γειτόνων και ενισχύουν τη συνέπεια που εκφράζεται από τις εκφράσεις αντιστοιχίσεων, οι οποίες δημιουργούνται στην ίδια φάση. Αυτοί οι κανόνες περιλαμβάνουν ένα τμήμα απλού γεγονότος και κανένα τμήμα συνθήκης. Το τμήμα ενέργειας δημοσιοποιεί ενημερώσεις προς τις κατάλληλες βάσεις δεδομένων.

Το δεύτερο σύνολο κανόνων δημιουργείται κατά τη διάρκεια των επερωτήσεων και αφού έχουν εγκαθιδρυθεί οι γειτνιάσεις. Οι κανόνες αυτοί υποστηρίζουν ένα μεγάλο εύρος από αυτόματη ανταλλαγή δεδομένων. Γενικά οι κανόνες αυτού του είδους περιλαμβάνουν ένα σύνθετο γεγονός, συνθήκη και ενέργεια, τα όποια περικλείουν διάφορους peers.

### 2.3 Σχεδιαστικές αποφάσεις για το RDPeer

Στα συστήματα ολοκλήρωσης δεδομένων γίνεται προσπάθεια να κατασκευαστεί μια ενοποιημένη όψη που να περιγράφει τα σχήματα των peers, και στην οποία όψη θα τίθενται οι επερωτήσεις. Όμως, η δυναμική φύση των peers σε συνδιασμό με την ετερογένεια της πληροφορίας που φέρουν καθιστούν την τεχνική προβληματική. Η

τεχνική απορρίφθηκε, επειδή δεν μπορεί να εφαρμοστεί σε ένα peer-to-peer περιβάλλον, παρά μόνο σε ένα περιβάλλον όπου οι κόμβοι και τα σχήματα που φέρουν είναι καλά ορισμένα, υπάρχει η εποπτεία ενός διαχειριστή, και δεν εμφανίζεται η περίπτωση αποτυχίας ενός κόμβου.

Το RDPeer είναι ένα υβριδικό peer-to-peer σύστημα αποτελούμενο από συμπλέγματα από peers, όπου κάθε peer είναι υλοποιημένος με βάση το μοντέλο πελάτη/εξυπηρετητή. Οι κοινότητες είναι αυθαίρετα δίκτυα από peers και δεν οργανώνονται με βάση το σημασιολογικό περιεχόμενο [9] των δεδομένων που φέρουν τα μέλη τους. Σε κάθε μια κοινότητα εκλέγεται ένας super-peer, ο οποίος αναλαμβάνει να διεξάγει την αναζήτηση της πληροφορίας και το μετασχηματισμό των επερωτήσεων. Η επιλογή της υβριδικής αρχιτεκτονικής υλοποιήθηκε για λόγους αποδοτικότερης αναζήτησης και δημιουργίας των αντιστοιχίσεων, καθώς και του μετασχηματισμού της SQL επερώτησης και της μικρότερης επιβάρυνσης της κίνησης του δικτύου.

Με τον τρόπο αυτό δεν αντιμετωπίζουμε το πρόβλημα που εμφανίζεται στα pure peer-to-peer συστήματα όπου υπάρχει μεγάλος αριθμός μηνυμάτων να διαδίδονται στο δίκτυο. Συνακόλουθα, το σύστημα έχει καλύτερο χρόνο απόκρισης στην επερώτηση που τίθεται από έναν peer. Η αρχιτεκτονική super-peer δεν επιλέχτηκε, γιατί κάθε super-peer θα έπρεπε να δέχεται επιπλέον φόρτο εργασίας, αφού θα έπρεπε να δρομολογεί τις αιτήσεις επερωτήσεων των πελατών του στους υπόλοιπους super-peer και να συλλέγει από αυτούς τις απαντήσεις. Αναμφίβολα όμως, μειονέκτημα αποτελεί το γεγονός ότι στην υβριδική αρχιτεκτονική που επιλέξαμε ο φόρτος εργασίας του συντονιστή της κοινότητας είναι πολύ μεγάλος και συνακόλουθα εμφανίζεται πρόβλημα σε ότι αφορά την επεκτασιμότητα, κλιμακοσιμότητα του συστήματος.

Ο σχηματισμός των κοινοτήτων δεν πραγματοποιείται με βάση το σημασιολογικό περιεχόμενο των δεδομένων που φέρουν οι peers. Κάθε peer είναι ελεύθερος να επιλέξει την κοινότητα στην οποία θέλει να συμμετέχει ή να εγκαταλείψει, όπως συμβαίνει και στον πραγματικό κόσμο. Αντίθετα με τον Napster, που είναι ένα υβριδικό peer-to-peer σύστημα και στο οποίο οι εξυπηρετητές είναι προκαθορισμένοι στο RDPeer υλοποιήσαμε τη δυναμική εκλογή των εξυπηρετητών για κάθε μια κοινότητα, ώστε σε περιπτώσεις αποτυχίας των εξυπηρετητών να μην επηρεάζεται η λειτουργία του συστήματος.

Η πραγματοποίηση του διαμοιρασμού των σχεσιακών δεδομένων στο RDPeer δε γίνεται με τη δημιουργία όψεων, και σχημάτων μεσολάβησης όπως συμβαίνει στο έργο Piazza. Επιπλέον, δε δημιουργούνται σε ενδιάμεσο επίπεδο συλλογές από σημασιολογικές αντιστοιχίσεις μεταξύ των σχημάτων των peers. Στο RDPeer ο super-peer κάθε κοινότητας είναι ο φορέας των σχημάτων όλων των μελών της κοινότητας. Σε αντίθεση με το έργο Piazza, δεν προσπαθούμε να κατασκευάσουμε αντιστοιχίσεις μεταξύ των σχημάτων ή ιδεατές όψεις εξαιτίας της δυναμικής φύσης των peers και της συχνότητας με την οποία μπορεί να μεταβάλλεται η σύνθεση κάθε κοινότητας. Επίσης, οι

αντιστοιχίσεις μεταξύ των σχημάτων δε δημιουργούνται γιατί οι κοινότητες δεν αποτελούνται από μέλη που φέρουν σημασιολογικά όμοιο περιεχόμενο δεδομένων και επομένως είναι πολύ πιθανόν ανάμεσα σε σχήματα να μην υπάρχει τρόπος για αυτόματη (χωρίς την παρέμβαση κάποιου διαχειριστή) δημιουργία αντιστοιχίσεων. Για το λόγο αυτό επιλέξαμε να δημιουργούμε αντιστοιχίσεις μεταξύ του σχήματος που περιγράφεται στην SQL επερώτηση και των σχημάτων των peers. Αυτό συμβαίνει για να μπορέσουμε στη συνέχεια να πραγματοποιήσουμε το μετασχηματισμό της επερώτησης.

Η τεχνική που ακολουθείται στο έργο Hyperion έχει το βασικό μειονέκτημα ότι οι πίνακες και οι εκφράσεις αντιστοίχισης δημιουργούνται χειρονακτικά και όχι με αυτόματο τρόπο με αποτέλεσμα το σύστημα να μην είναι ευέλικτο και εύκολα επεκτάσιμο. Επιπλέον, οι αντιστοιχίσεις έχουν το μειονέκτημα όπως και στο έργο Piazza ότι είναι δύσκολο να δημιουργηθούν με αυτόματο τρόπο και υπάρχει ο κίνδυνος της απομόνωσης μιας πηγής δεδομένων που έχει ετερογένεια πληροφορίας σε σχέση με τις υπόλοιπες.

Το έργο PeerDB είναι αυτό που εμφανίζει τις περισσότερες ομοιότητες με το RDPeer. Αυτό συμβαίνει, γιατί και στις δύο εφαρμογές έχουμε αντιστοίχιση της SQL επερώτησης με τα σχήματα των βάσεων δεδομένων. Η διαφορά όμως έγκειται στο γεγονός ότι στο PeerDB έχουμε την εύρεση ομοιοτήτων μεταξύ της SQL επερώτησης και του σχήματος, μεταδεδομένων του peer στον οποίο τίθεται η επερώτηση και στη συνέχεια το σύστημα ψάχνει για ομοιότητες ανάμεσα στο σχήμα του αρχικού peer με αυτά των υπολοίπων. Αντίθετα, στο RDPeer δημιουργούνται αντιστοιχίσεις μεταξύ της επερώτησης και κάθε σχήματος χωριστά. Επιπλέον, στο PeerDB οι ομοιότητες ανακαλύπτονται με καταναμημένο τρόπο, ενώ στο RDPeer κεντροποιημένα. Αυτό έχει σαν αποτέλεσμα να επιτυγχάνεται καλύτερη απόδοση στο RDPeer. Διαφορά εμφανίζεται επίσης στο γεγονός ότι τα μεταδεδομένα που περιγράφουν το σχήμα της βάσης δεδομένων και τα οποία δημιουργούνται από κάθε peer στο μεν PeerDB κατασκευάζονται μόνο με την παροχή λέξεων κλειδιών από το χρήστη, ενώ στο RDPeer εκτός από τις λέξεις κλειδιά που παρέχει ο χρήστης κάθε peer αναλαμβάνει για κάθε όνομα σχέσης και πεδίου να συνδεθεί σε δύο ηλεκτρονικά λεξικά και να συλλέξει συνώνυμα και υπερώνυμα. Τέλος διαφορά εμφανίζεται και στον τρόπο με τον οποίο εξάγονται τα δεδομένα. Στο RDPeer η εξαγωγή των δεδομένων γίνεται με επίκληση μιας υπηρεσίας που φέρει κάθε peer, ενώ στο PeerDB η διαδικασία πραγματοποιείται με τη χρήση πρακτόρων.

Στο επόμενο κεφάλαιο περιγράφεται η τεχνολογία JXTA[21], αναλύονται τα πλεονεκτήματα, μειονεκτήματα που εμπεριέχει και συγκρίνεται με άλλες αντίστοιχες τεχνολογίες που έχουν αναπτυχθεί. Η πλατφόρμα JXTA μας παρέχει όλα τα δομικά στοιχεία που απαιτούνται για να υλοποιήσουμε την peer-to-peer εφαρμογή RDPeer.

### 3 Τεχνολογία JXTA

Στο κεφάλαιο αυτό θα αναφερθούμε στην τεχνολογία JXTA [21], η οποία αποτέλεσε την πλατφόρμα πάνω στην οποία αναπτύχθηκε η εφαρμογή που υλοποιήσαμε. Η ραγδαία και διαρκής ανάπτυξη του παγκόσμιου ιστού τόσο από την πλευρά του περιεχομένου όσο και στο πλήθος των συνδεδεμένων συσκευών, ώθησε σε μεγάλη ανάπτυξη τους υπολογισμούς βάσει ισοτιμίας (peer-to-peer). Δημοφιλές λογισμικό αναπτύχθηκε βασισμένο σε τεχνολογίες peer-to-peer για τους σκοπούς του διαμοιρασμού αρχείων, καταναμημένων υπολογισμών, και υπηρεσιών *instant messenger*. Ενώ κάθε μια από αυτές τις εφαρμογές εκτελεί διαφορετική αποστολή, όλες μοιράζονται πολλές κοινές ιδιότητες, όπως είναι η ανακάλυψη των peers, η αναζήτηση στο δίκτυο, και η μεταφορά των αρχείων και δεδομένων. Βασικός σκοπός της τεχνολογίας JXTA είναι η παροχή μιας πλατφόρμας με τις βασικές λειτουργίες που είναι απαραίτητες για ένα δίκτυο peer-to-peer. Η τεχνολογία JXTA προσπαθεί να ξεπεράσει κάποιες αδυναμίες, που παρουσιάζονται στα περισσότερα υπάρχοντα peer-to-peer συστήματα όπως είναι:

- Διαλειτουργικότητα. Η τεχνολογία JXTA είναι σχεδιασμένη να επιτρέπει στους peers την παροχή μιας ποικιλίας υπηρεσιών για peer-to-peer δίκτυα, για ανεύρεση του ενός από τον άλλο και επικοινωνία μεταξύ τους.
- Ανεξαρτησία από πλατφόρμα. Είναι σχεδιασμένη να είναι ανεξάρτητη από γλώσσες προγραμματισμού, πρωτόκολλα μεταφοράς, και πλατφόρμες ανάπτυξης.
- Προσβασιμότητα. Η τεχνολογία JXTA έχει σχεδιαστεί έτσι ώστε να είναι προσβάσιμη από κάθε ψηφιακή συσκευή και όχι μόνο από ηλεκτρονικούς υπολογιστές ή από καθορισμένες πλατφόρμες ανάπτυξης.

Το JXTA εξυπηρετεί τους peers που βρίσκονται στην άκρη του δικτύου με την παροχή ενός συστήματος για μοναδική διευθυνσιοδότησή τους, που είναι ανεξάρτητη από τις παραδοσιακές υπηρεσίες ονομάτων. Με τη χρήση των αναγνωριστικών JXTA, ένας peer μπορεί να περιδιαβαίνει το δίκτυο, αλλάζοντας μεταφορικά μέσα και διευθύνσεις δικτύου, ακόμα και να είναι προσωρινά αποσυνδεδεμένος, και παρόλα αυτά να είναι προσβάσιμος από τους άλλους peers.

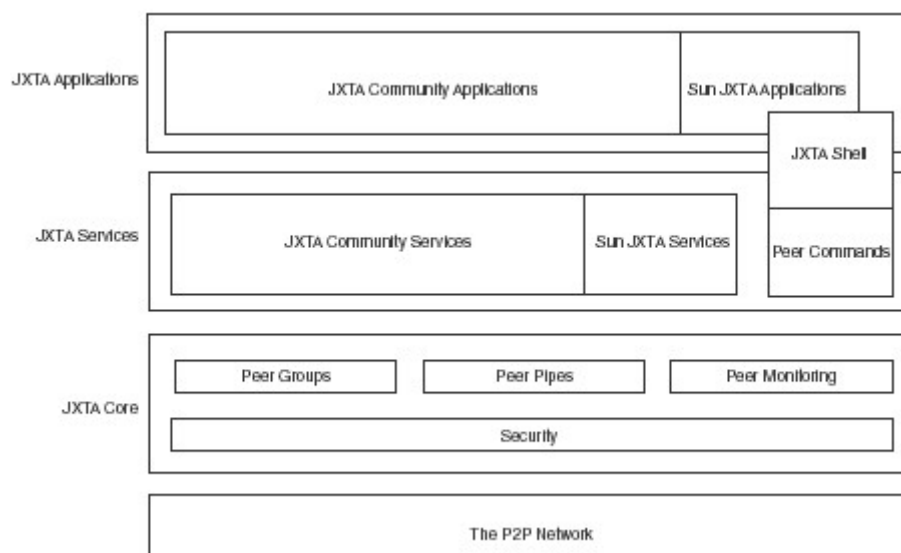
Το JXTA είναι μία ανοιχτού δικτύου υπολογιστική πλατφόρμα, σχεδιασμένη για υπολογισμούς peer-to-peer. Σκοπός της τεχνολογίας είναι η ανάπτυξη υπηρεσιών που θα επιτρέπουν την ανάπτυξη εφαρμογών για ομάδες από peers. Το JXTA παρέχει ένα σύνολο από ανοιχτά πρωτόκολλα και μια ανοιχτού λογισμικού υλοποίηση για την ανάπτυξη peer-to-peer εφαρμογών. Τα πρωτόκολλα του JXTA τυποποιούν τον τρόπο με τον οποίο οι peers:

- Ανακαλύπτουν ο ένας τον άλλο.
- Οργανώνονται σε κοινότητες από peers.

- Δημοσιοποιούν και ανακαλύπτουν δικτυακές υπηρεσίες
- Επικοινωνούν μεταξύ τους
- Παρακολουθούν και ελέγχουν ο ένας τον άλλο.

### 3.1 Αρχιτεκτονική της τεχνολογίας JXTA

Η αρχιτεκτονική διαιρείται σε τρία επίπεδα όπως φαίνεται στην παρακάτω εικόνα. Στη συνέχεια αναλύεται κάθε επίπεδο χωριστά.



Εικόνα 4 Τα τρία επίπεδα αρχιτεκτονικής της τεχνολογίας JXTA.

#### 3.1.1 Επίπεδο πλατφόρμας (JXTA Core)

Το επίπεδο πλατφόρμας, που είναι γνωστό και ως πυρήνας της τεχνολογίας JXTA, περικλύει ελάχιστα στοιχεία τα οποία είναι κοινά σε peer-to-peer δίκτυα. Συμπεριλαμβάνει στοιχεία που επιτρέπουν μηχανισμούς κλειδιά για peer-to-peer εφαρμογές, όπως είναι η ανακάλυψη, η μεταφορά (περιλαμβάνει και τον χειρισμό των firewalls), η δημιουργία των peers και των κοινοτήτων από peers και θέματα που σχετίζονται με την προστασία. Συνεπώς, το επίπεδο αυτό παρέχει τα στοιχεία που είναι απολύτως απαραίτητα σε κάθε σύστημα peer-to-peer και τα οποία είναι:

- Peers
- Κοινότητες από Peers



- Δίκτυα μεταφοράς (*pipes, endpoints, messages*)
- Δημοσιοποιήσεις (*Advertisements*)
- Ονομασία οντοτήτων
- Πρωτόκολλα
- Ασφάλεια και πιστοποίηση

Το επίπεδο αυτό επιπλέον περιλαμβάνει τα έξι κύρια πρωτόκολλα που παρέχονται από το JXTA. Αν και αυτά τα πρωτόκολλα είναι υλοποιημένα σαν υπηρεσίες βρίσκονται στο επίπεδο πλατφόρμας και προσδιορίζονται ως υπηρεσίες πυρήνα.

### 3.1.2 Επίπεδο υπηρεσιών (JXTA Services)

Το επίπεδο υπηρεσιών περιλαμβάνει υπηρεσίες δικτύου που μπορεί να μην είναι απαραίτητες για να λειτουργήσουν τα peer-to-peer δίκτυα, αλλά είναι επιθυμητές σε περιβάλλον peer-to-peer. Παραδείγματα υπηρεσιών δικτύου περιλαμβάνουν την τοποθέτηση δεικτών (*indexing*), την αναζήτηση, συστήματα αποθήκευσης, διαμοιρασμό των αρχείων, καταναμημένα συστήματα αρχείων, συνάθροιση πηγών πληροφοριών, μετάφραση πρωτοκόλλων, πιστοποίηση, και υπηρεσίες PKI (*Public Key Infrastructure*).

### 3.1.3 Επίπεδο εφαρμογών (JXTA Applications)

Το επίπεδο εφαρμογών χτίζεται πάνω στις δυνατότητες του επιπέδου υπηρεσιών και παρέχει εφαρμογές peer-to-peer. Το επίπεδο αυτό περιλαμβάνει υλοποιήσεις όπως είναι ο διαμοιρασμός εγγράφων και πηγών πληροφοριών, συστήματα ηλεκτρονικού ταχυδρομείου peer-to-peer, καταναμημένα συστήματα δημοπρασιών, και πολλά άλλα. Στο επίπεδο αυτό έχει υλοποιηθεί και η εφαρμογή RDPeer που δημιουργήσαμε.

### 3.1.4 Θέματα κλειδιά στην αρχιτεκτονική JXTA

Τρία στοιχεία της αρχιτεκτονικής τη διαχωρίζουν από τα άλλα μοντέλα καταναμημένων δικτύων. Αυτά είναι:

- Η χρήση των XML εγγράφων για την περιγραφή των δικτυακών πηγών.
- Η αφαίρεση των σωληνώσεων στους peers, και των peers στα σημεία κατάληξης (*endpoints*), χωρίς την εξάρτηση πάνω σε μια κεντρική εξουσία ονοματοδότησης και διευθυνσιοδότησης όπως είναι το DNS.

- Η χρήση ενός ομοιόμορφου σχήματος διευθυνσιοδότησης των peers, που ανάγεται στη χρήση ενός μοναδικού αναγνωριστικού για κάθε έναν από αυτούς.

## 3.2 Αρχές της τεχνολογίας JXTA

Οι αρχές της τεχνολογίας JXTA χωρίζονται σε επτά βασικές κατηγορίες. Τα κυριότερα σημεία αναλύονται παρακάτω.

### 3.2.1 Peers

Ένας peer είναι μια συσκευή του δικτύου που υλοποιεί ένα ή περισσότερα από τα πρωτόκολλα του JXTA. Οι peers μπορούν να είναι αισθητήρια όργανα, κινητά τηλέφωνα, PDA, ηλεκτρονικοί υπολογιστές, εξυπηρετητές, και υπερυπολογιστές. Κάθε peer λειτουργεί ανεξάρτητα και ασύγχρονα από όλους τους άλλους peers, και γίνεται γνωστός από ένα μοναδικό αναγνωριστικό. Οι peers δημοσιεύουν μία ή περισσότερες διεπιφάνειες δικτύου για χρήση μαζί με τα πρωτόκολλα του JXTA. Κάθε δημοσιευμένη διεπαφή γίνεται γνωστή σαν ένα σημείο κατάληξης του peer, το οποίο μοναδικά αναγνωρίζει τη διεπαφή του δικτύου. Τα σημεία κατάληξης του peer χρησιμοποιούνται από τους peers για να εγκαθιδρύσουν απευθείας σημείο προς σημείο συνδέσεις ανάμεσα σε δύο peers. Οι peers δεν απαιτείται να έχουν σημείο προς σημείο συνδέσεις μεταξύ τους. Ενδιάμεσοι peers μπορούν να χρησιμοποιηθούν για να δρομολογήσουν μηνύματα σε peers, που είναι χωριστά εξαιτίας των φυσικών συνδέσεων του δικτύου ή της διαμόρφωσης, διάταξης του δικτύου.

### 3.2.2 Κοινότητες από Peers

Μία κοινότητα από peers είναι μια συλλογή από peers, οι οποίοι έχουν συμφωνήσει πάνω σε ένα κοινό σύνολο από υπηρεσίες. Οι peers οργανώνονται μόνοι τους σε κοινότητες, όπου η κάθε κοινότητα γίνεται αντιληπτή από ένα μοναδικό αναγνωριστικό. Κάθε κοινότητα μπορεί να ορίσει τη δική της πολιτική στη διαδικασία εγγραφής μέλους και μπορεί να είναι από ανοιχτή σε όλους τους peers μέχρι πολύ ασφαλής και προστατευμένη. Οι peers μπορούν ταυτόχρονα να ανήκουν σε παραπάνω της μίας κοινότητας. Αρχικά η πρώτη κοινότητα που αρχικοποιείται είναι η *NetPeerGroup*. Όλοι

οι peers ανήκουν σε αυτή και μπορούν να συμμετέχουν και σε επιπρόσθετες κοινότητες. Υπάρχουν διάφορα κίνητρα για τη δημιουργία των κοινοτήτων όπως είναι:

- Η δημιουργία ασφαλούς περιβάλλοντος. Η κοινότητα δημιουργεί μια τοπική περιοχή ελέγχου στην οποία μπορεί να εφαρμοστεί μια καθορισμένη πολιτική ασφαλείας.
- Η δημιουργία περιβάλλοντος ειδικού ενδιαφέροντος. Οι κοινότητες επιτρέπουν, βοηθούν στη δημιουργία περιοχών εξειδίκευσης.
- Η δημιουργία περιβάλλοντος και παρατήρησή του. Οι κοινότητες επιτρέπουν στους peers την παρατήρηση ενός συνόλου από peers για κάποιους ειδικούς σκοπούς όπως είναι η ενδοσκόπηση της κίνησης, η προσβασιμότητα και η υπευθυνότητα.

Οι κοινότητες επιπλέον σχηματίζουν μια ιεραρχική σχέση πατέρα-παιδιού, μέσα στις οποίες κάθε κοινότητα έχει ένα μόνο πατέρα. Οι αιτήσεις αναζήτησης διαδίδονται μέσα στα όρια της κοινότητας. Η δημοσιοποίηση για την κοινότητα διαδίδεται μέσα στην πατρική κοινότητα.

Το JXTA ορίζει ένα σύνολο από υπηρεσίες κοινότητας. Προκειμένου δύο peers να αλληλεπιδράσουν μέσω μιας υπηρεσίας πρέπει να ανήκουν στην ίδια κοινότητα. Οι υπηρεσίες κοινότητας περιλαμβάνουν τα ακόλουθα:

- Υπηρεσία ανακάλυψης (*Discovery Service*). Η υπηρεσία αυτή χρησιμοποιείται από τα μέλη μιας κοινότητας για να ανακαλύψουν πόρους, όπως peers, κοινότητες από peers, σωληνώσεις και υπηρεσίες.
- Υπηρεσία εγγραφής μέλους (*Membership Service*). Η υπηρεσία αυτή χρησιμοποιείται από τα τρέχοντα μέλη μιας κοινότητας, έτσι ώστε να απορρίψουν ή να κάνουν δεκτή μια καινούργια αίτηση ενός peer που θέλει να εισαχθεί στην ομάδα. Οι peers που επιθυμούν να συμμετέχουν σε μια ομάδα πρέπει πρώτα να εντοπίσουν ένα τρέχον μέλος και μετά να αιτηθούν για συμμετοχή.
- Υπηρεσία πρόσβασης (*Access Service*). Η υπηρεσία πρόσβασης χρησιμοποιείται για να ελέγχουν την αξιοπιστία της αίτησης, που έγινε από έναν peer σε έναν άλλο. Ο peer που λαμβάνει την αίτηση παρέχει στον αιτούντα peer τα διαπιστευτήρια και πληροφορία σχετικά με την αίτηση που έγινε για να καθοριστεί αν η πρόσβαση γίνεται δεκτή.
- Υπηρεσία σωλήνα (*Pipe Service*). Η υπηρεσία αυτή χρησιμοποιείται για τη δημιουργία και διαχείριση συνδέσεων σωλήνων ανάμεσα σε μέλη της κοινότητας.
- Υπηρεσία Resolver. Η υπηρεσία χρησιμοποιείται για να στείλει γενικές ερωτήσεις προς άλλους peers.

- Υπηρεσία παρατήρησης (*monitoring service*). Η υπηρεσία παρατήρησης χρησιμοποιείται από ένα peer για να παρατηρήσει άλλα μέλη της ίδιας κοινότητας.

Οι παραπάνω υπηρεσίες δεν είναι απαραίτητο να υλοποιούνται όλες από κάθε κοινότητα. Μια κοινότητα είναι ελεύθερη να υλοποιήσει μόνο τις υπηρεσίες που βρίσκει χρήσιμες.

### 3.2.3 Σωληνώσεις

Οι peers στην τεχνολογία JXTA χρησιμοποιούν τις σωληνώσεις για να μπορέσουν να στείλουν μηνύματα ο ένας στον άλλο. Οι σωληνώσεις είναι ένας ασύγχρονος και μονής κατεύθυνσης μηχανισμός μεταφοράς μηνυμάτων, που χρησιμοποιείται για επικοινωνία. Υποστηρίζουν την μεταφορά κάθε αντικειμένου. Τα σημεία κατάληξης του σωλήνα αναφέρονται σαν σωλήνας εισόδου και σωλήνας εξόδου. Τα σημεία κατάληξης δεσμεύονται δυναμικά και ανταποκρίνονται στις διαθέσιμες διασυνδέσεις δικτύου (π.χ. μια TCP πύλη και η σχετιζόμενη IP διεύθυνση) που μπορούν να χρησιμοποιηθούν για την αποστολή και λήψη μηνύματος.

Οι σωληνώσεις είναι ιδεατά κανάλια επικοινωνίας και μπορούν να συνδέουν peers, οι οποίοι δεν έχουν απευθείας φυσικό σύνδεσμο. Σε αυτή την περίπτωση, ένα ή περισσότερα ενδιάμεσα σημεία κατάληξης peer χρησιμοποιούνται για να αναμεταδώσουν τα μηνύματα ανάμεσα στα δύο σημεία κατάληξης σωλήνα. Οι σωληνώσεις προσφέρουν δυο μεθόδους επικοινωνίας, σημείο προς σημείο και διασπορά.

### 3.2.4 Μηνύματα

Ένα μήνυμα είναι ένα αντικείμενο το οποίο στέλνεται μεταξύ των peers. Είναι η βασική μονάδα ανταλλαγής δεδομένων μεταξύ των peers. Τα μηνύματα στέλνονται και λαμβάνονται από την υπηρεσία σωλήνα και από την υπηρεσία σημείου κατάληξης. Μια εφαρμογή χρησιμοποιεί την υπηρεσία σωλήνωσης για να δημιουργήσει, να στείλει και να λάβει μηνύματα. Το μήνυμα είναι μια διατεταγμένη ακολουθία από ονόματα και περιεχόμενα που ονομάζονται στοιχεία του μηνύματος. Έτσι το μήνυμα δεν είναι τίποτα άλλο από ένα σύνολο από ζεύγη ονόματος / τιμής. Το περιεχόμενο μπορεί να είναι αυθαίρετου τύπου. Τα πρωτόκολλα της τεχνολογίας JXTA καθορίζονται σαν ένα σύνολο από μηνύματα που ανταλλάσσονται μεταξύ των peers. Υπάρχουν δύο μορφές αναπαράστασης για μηνύματα. Αυτές είναι η XML μορφή και η δυαδική μορφή.

### 3.2.5 Δημοσιοποιήσεις

Όλοι οι πόροι του δικτύου, όπως οι peers, οι κοινότητες από peers, οι σωληνώσεις, και οι υπηρεσίες αναπαρίστανται από μια δημοσιοποίηση. Οι δημοσιοποιήσεις είναι δομές μεταδεδομένων που αναπαρίστανται σαν XML έγγραφα. Τα πρωτόκολλα του JXTA χρησιμοποιούν τις δημοσιοποιήσεις για να περιγράψουν και να κοινοποιήσουν την ύπαρξη πόρων ενός peer. Οι peers ανακαλύπτουν πόρους ψάχνοντας για τις αντίστοιχες δημοσιοποιήσεις, τις οποίες μπορούν προσωρινά να αποθηκεύσουν τοπικά. Κάθε δημοσιοποίηση γίνεται γνωστή στο ευρύ κοινό μαζί με ένα χρονικό όριο ζωής, το οποίο καθορίζει τη διαθεσιμότητα του σχετιζόμενου πόρου. Μια δημοσιοποίηση πρέπει να κοινοποιηθεί ξανά προκειμένου να επεκταθεί το χρονικό όριο ζωής της.

Οι δημοσιοποιήσεις συντίθενται σαν μια σειρά από ιεραρχικά διευθετημένα στοιχεία. Κάθε στοιχείο μπορεί να περιέχει δεδομένα ή επιπλέον στοιχεία. Ένα στοιχείο μπορεί ακόμα να έχει ιδιότητες. Οι ιδιότητες είναι αλφαριθμητικά ζεύγη ονόματος τιμής. Μια ιδιότητα χρησιμοποιείται για να αποθηκεύσει μεταδεδομένα, τα οποία βοηθούν στην περιγραφή των δεδομένων που βρίσκονται μέσα στο στοιχείο. Ένα παράδειγμα δημοσιοποίησης σωληνώσεως περιγράφεται στην παρακάτω εικόνα.

```
<?xml version="1.0" ?>
<!DOCTYPE jxta:PipeAdvertisement>
<jxta:PipeAdvertisement xmlns:jxta="http://jxta.org">
  <Id>
    urn:jxta:uuid-
    59616261646162614E504720503250338E3E786229EA460DADC1A176B69B731504
  </Id>
  <Type>
    JxtaUnicast
  </Type>
  <Name>
    TestPipe.end1
  </Name>
</jxta:PipeAdvertisement>
```

Εικόνα 5 Παράδειγμα δημοσιοποίησης σωλήνα.

### 3.2.6 Ασφάλεια

Πέντε βασικές απαιτήσεις ασφάλειας παρέχονται από την τεχνολογία JXTA. Αυτές είναι:

- Εμπιστευτικότητα: Εγγυάται ότι το περιεχόμενο ενός μηνύματος δεν αποκαλύπτεται σε μη εξουσιοδοτημένα άτομα.
- Πιστοποίηση: εγγυάται ότι ο αποστολέας είναι αυτός που ισχυρίζεται ότι είναι.

- Εξουσιοδότηση: εγγυάται ότι ο αποστολέας είναι εξουσιοδοτημένος να στείλει ένα μήνυμα.
- Ακεραιότητα δεδομένων: εγγυάται ότι το μήνυμα δε μεταβλήθηκε κατά τη μεταφορά.
- Ανασκευασιμότητα: εγγυάται ότι το μήνυμα μεταδόθηκε από ένα κατάλληλα αναγνωρισμένο αποστολέα και δεν είναι επανάληψη ενός προηγούμενου μεταδομένου μηνύματος.

Τα XML μηνύματα παρέχουν την δυνατότητα της προσθήκης μεταδεδομένων όπως είναι τα πιστοποιητικά, οι συνόψεις, και τα δημόσια κλειδιά στα μηνύματα JXTA, τα οποία επιτρέπουν την εφαρμογή των βασικών απαιτήσεων ασφαλείας. Τα μηνύματα επιπλέον μπορεί να είναι κρυπτογραφημένα (χρησιμοποιώντας δημόσια κλειδιά) και υπογεγραμμένα (χρησιμοποιώντας πιστοποιητικά) για εμπιστευτικότητα και ανασκευασιμότητα.

### 3.2.7 Αναγνωριστικά

Οι peers, οι κοινότητες από peers, οι σωληνώσεις και όλοι οι άλλοι πόροι της τεχνολογίας JXTA πρέπει να είναι μοναδικά αναγνωρίσιμοι. Ένα αναγνωριστικό ορίζει μοναδικά μια οντότητα και υπηρετεί έναν κανονικό τρόπο αναφοράς σ' αυτή την οντότητα. Υπάρχουν έξι τύποι από JXTA οντότητες, που έχουν ορισμένους τύπους αναγνωριστικών και αυτές είναι: οι peers, οι κοινότητες από peers, οι σωληνώσεις, τα περιεχόμενα, τα *module classes*, και τα *module specifications*.

Τα URN χρησιμοποιούνται για να εκφράσουν τα αναγνωριστικά της τεχνολογίας JXTA. Τα URN είναι ένας τύπος URI. Όπως και άλλοι τύποι URI έτσι και τα αναγνωριστικά της τεχνολογίας JXTA αναπαρίστανται σαν κείμενο. Παράδειγμα ενός αναγνωριστικού peer είναι το ακόλουθο:

```
urn:jxta:uuid-  
1CCCDF5AD8154D3D87A391210404E59BE4B888209A2241A4A162A10916074A9504
```

Υπάρχουν δύο ειδικά δεσμευμένα αναγνωριστικά στην τεχνολογία JXTA: το αναγνωριστικό NULL, και το Net Peer Group αναγνωριστικό.

## 3.3 Αρχιτεκτονική του Δικτύου

Η αρχιτεκτονική του δικτύου μπορεί να χωριστεί με βάση τον τρόπο με τον οποίο οργανώνεται το δίκτυο, τον κατανομημένο δείκτη διαμοιραζόμενου πόρου και την τεχνική που ακολουθείται για να πραγματοποιηθεί η διάσχιση ενός firewall.

### 3.3.1 Οργάνωση Δικτύου

Το δίκτυο JXTA είναι ένα συγκεκριμένου σκοπού, πολλαπλών κόμβων, και προσαρμόσιμο δίκτυο, που συντίθεται από τους συνδεδεμένους peers. Οι συνδέσεις στο δίκτυο μπορεί να είναι προσωρινές και η δρομολόγηση των μηνυμάτων μεταξύ των peers δεν είναι πάντοτε συνεπής. Οι peers μπορούν να λαμβάνουν μέρος ή να φεύγουν από το δίκτυο οποιαδήποτε χρονική στιγμή και οι δρομολογήσεις μπορούν να μεταβάλλονται συχνά.

Η οργάνωση του δικτύου δεν προστάζεται από το πλαίσιο εργασίας του JXTA, αλλά στην πράξη τέσσερις τύποι από peers χρησιμοποιούνται.

1. Ελάχιστης λειτουργίας peer. Ένας ελάχιστης λειτουργίας peer μπορεί να στείλει και να λάβει μηνύματα, αλλά δεν αποθηκεύει δημοσιοποιήσεις ή δρομολογεί μηνύματα για άλλους peers. Οι peers σε συσκευές με περιορισμένους πόρους (π.χ. ένα PDA ή ένα ψηφιακό τηλέφωνο) θα προτιμούσαν να είναι peer ελάχιστης λειτουργίας.
2. Πλήρους λειτουργικότητας peer. Ο πλήρους λειτουργικότητας peer μπορεί να στείλει και να λάβει μηνύματα, έχει τη δυνατότητα να αποθηκεύει τις δημοσιοποιήσεις, αλλά δεν προωθεί καθόλου τις αιτήσεις αναζήτησης.
3. Ραντεβού peer. Ένας ραντεβού peer μπορεί να είναι κάθε άλλος peer και διατηρεί μια λανθάνουσα μνήμη με δημοσιοποιήσεις. Όμως οι ραντεβού peers προωθούν αιτήσεις αναζήτησης για να βοηθήσουν άλλους peers να ανακαλύψουν πόρους. Όταν ένας peer γίνεται μέλος μιας κοινότητας από peers, αυτόματα αναζητεί ένα ραντεβού peer. Εάν δε βρεθεί κάποιος ραντεβού peer, τότε δυναμικά γίνεται ραντεβού peer για αυτή την κοινότητα. Κάθε ραντεβού peer διατηρεί μια λίστα από όλους τους άλλους γνωστούς ραντεβού peers και επίσης τους peers που τον χρησιμοποιούν σαν ραντεβού peer. Κάθε κοινότητα από peers διατηρεί το δικό του σύνολο από ραντεβού peers και ο αριθμός τους δεν περιορίζεται. Μόνο οι ραντεβού peers που είναι μέλη μιας κοινότητας θα δουν τις αιτήσεις αναζήτησης που είναι ειδικές για την κοινότητα. Οι peers στέλνουν αιτήσεις αναζήτησης και ανακάλυψης στους ραντεβού peers, και αυτοί προωθούν τις αιτήσεις σε όλους τους άλλους γνωστούς ραντεβού peers στην περίπτωση που δεν μπορούν να τις απαντήσουν. Η διαδικασία ανακάλυψης συνεχίζεται μέχρι ένας peer να μπορεί να απαντήσει ή η αίτηση να πεθάνει. Τα μηνύματα έχουν ένα προκαθορισμένο χρονικό διάστημα ζωής και είναι επτά hops. Οι δρομολογήσεις των μηνυμάτων

δεν πέφτουν στην λανθασμένη κατάσταση της ανακύκλωσης, καθώς αυτό εμποδίζεται με τη διατήρηση της λίστας των peers κατά μήκος του μονοπατιού που ακολουθείται από το μήνυμα.

4. Αναμεταδότης Peer. Ένας peer αναμεταδότης διατηρεί πληροφορία σχετικά με τα δρομολόγια, διαδρομές προς τους άλλους peers. Ένας peer κοιτάει πρώτα στην τοπική λανθάνουσα μνήμη για πληροφορίες δρομολόγησης. Εάν δε βρεθούν, τότε ο peer στέλνει επερωτήσεις στους αναμεταδότες ζητώντας πληροφορίες δρομολόγησης. Οι αναμεταδότες επίσης προωθούν μηνύματα εκ μέρους άλλων peers, οι οποίοι δεν μπορούν να επικοινωνήσουν απευθείας με κάποιον άλλο peer, γεφυρώνοντας έτσι διαφορετικά φυσικά ή λογικά δίκτυα.

### 3.3.2 Διαμοιραζόμενου Πόρου Κατανεμημένος Δείκτης

Η πλατφόρμα JXTA 2.0 J2SE υποστηρίζει ένα διαμοιραζόμενου πόρου κατανεμημένο δείκτη (*Shared Resource Distributed Index - SRDI*) υπηρεσία για να παρέχει ένα πιο κατάλληλο μηχανισμό για την διάδοση των αιτημάτων ερωτήσεων μέσα στο δίκτυο JXTA. Οι ραντεβού peers διατηρούν ένα δείκτη των δημοσιοποιήσεων που κοινοποίησαν οι peers. Όταν οι peers κάνουν μια καινούργια κοινοποίηση χρησιμοποιούν την υπηρεσία SRDI για να προωθήσουν τους δείκτες της δημοσίευσης στους ραντεβού peers με τους οποίους συνδέονται. Τηρώντας αυτή την ιεραρχία με τους ραντεβού peers και τους απλούς peers, πετυχαίνεται η διάδοση των ερωτημάτων μόνο μεταξύ των ραντεβού peers, γεγονός που μειώνει σημαντικά τον αριθμό των peers που εμπλέκονται στην αναζήτηση για κάποια κοινοποίηση.

Κάθε ραντεβού peer διατηρεί τη δική του λίστα με τους γνωστούς ραντεβού peers μέσα στην κοινότητα. Επιπλέον, μπορεί να ανακτεί πληροφορία από ένα προκαθορισμένο σύνολο από ραντεβού peers, ενώ περιοδικά γίνεται εκκαθάριση από τους ραντεβού peers που δεν ανταποκρίνονται. Έτσι καταφέρνουν να διατηρούν ένα συνεπές δίκτυο από γνωστούς ραντεβού peers. Όταν ένας peer κοινοποιεί μια καινούργια δημοσίευση, τότε αυτή δεικτοδοτείται από την SRDI υπηρεσία χρησιμοποιώντας κλειδιά όπως είναι το όνομα της δημοσίευσης ή το αναγνωριστικό της. Μόνο οι δείκτες των δημοσιεύσεων προωθούνται από το SRDI στους ραντεβού peers, ελαχιστοποιώντας με αυτό τον τρόπο το πλήθος των δεδομένων που χρειάζονται να αποθηκευτούν στους ραντεβού peers. Τέλος οι ραντεβού peers προωθούν τους δείκτες στους γνωστούς ομοίους τους.

### 3.3.3 Firewall και NAT



Ένας peer που βρίσκεται πίσω από ένα firewall μπορεί να στείλει απευθείας ένα μήνυμα σε ένα peer που βρίσκεται έξω από το firewall. Αλλά ένας peer εκτός του firewall δεν μπορεί να εγκαθιδρύσει απευθείας σύνδεση με το peer πίσω από το firewall. Προκειμένου οι peers να μπορέσουν να επικοινωνήσουν μεταξύ τους διαμέσου του firewall πρέπει να ισχύουν οι παρακάτω συνθήκες:

- Τουλάχιστον ένας peer της κοινότητας μέσα στο firewall πρέπει να αντιλαμβάνεται έναν τουλάχιστον peer έξω από το firewall.
- Ο peer που βρίσκεται εντός και αυτός που βρίσκεται εκτός του firewall πρέπει να γνωρίζονται και πρέπει να υποστηρίζουν το πρωτόκολλο HTTP.
- Το firewall πρέπει να επιτρέπει τις HTTP μεταφορές δεδομένων.

### 3.4 Πρωτόκολλα τεχνολογίας JXTA

Το JXTA ορίζει μια σειρά από σχήματα μηνυμάτων XML, ή πρωτόκολλα για επικοινωνία μεταξύ των peers. Οι peers χρησιμοποιούν αυτά τα πρωτόκολλα για να ανακαλύψουν ο ένας τον άλλο, να κοινοποιήσουν και να αναζητήσουν πόρους στο δίκτυο, να επικοινωνήσουν και να ανακαλύψουν μηνύματα δρομολόγησης. Υπάρχουν έξι πρωτόκολλα συνολικά. Όλα τα πρωτόκολλα είναι ασύγχρονα και είναι βασισμένα στο μοντέλο ερώτηση / απάντηση. Ένας JXTA peer χρησιμοποιεί ένα από τα πρωτόκολλα για να στείλει μια ερώτηση σε ένα ή περισσότερους peers μέσα στην ίδια κοινότητα. Μπορεί να λάβει μηδέν, μία ή περισσότερες απαντήσεις στην ερώτηση που έθεσε. Οι peers δεν είναι απαραίτητο να υλοποιήσουν και τα έξι πρωτόκολλα αλλά μόνο αυτά που θα χρησιμοποιήσουν. Η χρησιμότητα των πρωτοκόλλων αναλύεται παρακάτω.

#### 3.4.1 Πρωτόκολλο Ανακάλυψης Peer

Το πρωτόκολλο ανακάλυψης peer (*Peer Discovery Protocol - PDP*) χρησιμοποιείται για την ανακάλυψη των κοινοποιημένων πόρων peer. Το PDP επιτρέπει στους peers να βρουν δημοσιοποιήσεις πάνω σε άλλους peers. Είναι το προκαθορισμένο, προεγκατεστημένο πρωτόκολλο ανακάλυψης για όλες τις ορισμένες από το χρήστη κοινότητες από peers καθώς και για το Net Peer Group.

Υπάρχουν πολλαπλοί τρόποι για να ανακαλύψει κανείς κατανεμημένη πληροφορία. Στην συγκεκριμένη τεχνολογία χρησιμοποιείται ένας συνδιασμός από IP multicast για το τοπικό υποδίκτυο και τη χρήση των ραντεβού peers. Οι ραντεβού peers παρέχουν το μηχανισμό της αποστολής των αιτήσεων από ένα γνωστό peer στον επόμενο για να

επιτύχουν τη δυναμική ανακάλυψη της πληροφορίας. Ένας peer μπορεί να είναι προεγκατεστημένος με ένα σύνολο από προκαθορισμένους ραντεβού peers. Οι peers δημιουργούν μηνύματα με αιτήσεις ανακάλυψης. Το μήνυμα αυτό περιέχει το πιστοποιητικό της κοινότητας στην οποία ανήκει ο αποστολέας peer και γνωστοποιεί τον peer που κάνει την έρευνα στον παραλήπτη peer. Τα μηνύματα μπορούν να σταλούν μέσα σε μια περιοχή ή σε ένα ραντεβού peer. Το μήνυμα της απόκρισης μπορεί να περιέχει καμία, μία ή περισσότερες δημοσιοποιήσεις.

### 3.4.2 Πρωτόκολλο Πληροφορίας Peer

Το πρωτόκολλο πληροφορίας Peer (*Peer Information Protocol – PIP*) παρέχει ένα σύνολο από μηνύματα για να αποκομίσει κάποιος πληροφορία σχετικά με την κατάσταση ενός peer. Η πληροφορία μπορεί να χρησιμοποιηθεί για εμπορική ή εσωτερική ανάπτυξη εφαρμογών. Για παράδειγμα, σε εμπορικές εφαρμογές η πληροφορία μπορεί να χρησιμοποιηθεί για να καθορίσει τη χρήση μιας υπηρεσίας ενός peer και να χρεώσει τους πελάτες της υπηρεσίας για τη χρήση τους. Σε μια εσωτερική εφαρμογή, η πληροφορία μπορεί να χρησιμεύσει για να παρατηρηθεί η συμπεριφορά ενός κόμβου και να ξαναδρομολογηθεί η κίνηση του δικτύου, ώστε να βελτιώσουμε τη συνολική απόδοση. Το μήνυμα που επιστρέφεται όταν ζητείται πληροφορία για ένα peer περιέχει το πιστοποιητικό του αποστολέα, το αναγνωριστικό του peer αποστολέα και παραλήπτη, το χρόνο λειτουργίας, και τη δημοσιοποίηση του peer.

### 3.4.3 Πρωτόκολλο Δέσμευσης Σωλήνων

Το πρωτόκολλο δέσμευσης σωλήνων (*Pipe Binding Protocol – PBP*) χρησιμοποιείται από τα μέλη μιας κοινότητας για να δεσμεύσουν μια δημοσιοποίηση σωλήνα σε μια κατάληξη σωλήνα. Ο εικονικός σύνδεσμος σωλήνα μπορεί να τοποθετηθεί πάνω από οποιοδήποτε αριθμό από φυσικούς συνδέσμους μεταφοράς όπως είναι το TCP/IP. Μια σωλήνα μπορεί να ειπωθεί σαν μια ουρά μηνυμάτων, που υποστηρίζει τις λειτουργίες της δημιουργίας, του ανοίγματος, του κλεισίματος, της διαγραφής, της αποστολής και λήψης. Κατά τη λειτουργία της δημιουργίας, ένας τοπικός peer αναλαμβάνει να δεσμεύσει μια κατάληξη ενός σωλήνα με ένα μεταφορικό μέσο σωλήνα. Το μήνυμα ερώτησης PBP στέλνεται από την κατάληξη σωλήνα ενός peer για να βρεθεί μια κατάληξη σωλήνα δεσμευμένη στην ίδια δημοσιοποίηση σωλήνα.

### 3.4.4 Πρωτόκολλο Resolver Peer

Το Πρωτόκολλο Resolver Peer (Peer Resolver Protocol - PRP) επιτρέπει στους peers να στέλνουν γενικής χρήσης ερωτήσεις σε άλλους peers και να αναγνωρίζουν τις απαντήσεις που ταιριάζουν. Τα αιτήματα των ερωτήσεων μπορούν να σταλούν σε ένα συγκεκριμένο peer ή μπορούν να διαδοθούν διαμέσου της υπηρεσίας ραντεβού μέσα σε μια κοινότητα. Το PRP χρησιμοποιεί την υπηρεσία ραντεβού για να διασπείρει μια ερώτηση σε πολλούς peers και χρησιμοποιεί τη μετάδοση προς ένα παραλήπτη για να στείλει ερωτήσεις σε συγκεκριμένους peers.

Το PRP είναι ένα θεμελιώδες πρωτόκολλο που υποστηρίζει γενικής χρήσης αιτήματα ερωτήσεων. Το PIP και το PDP είναι δομημένα πάνω στο PRP και υποστηρίζουν συγκεκριμένα αιτήματα, ερωτήσεις. Όπως αναλύσαμε και παραπάνω το PIP χρησιμοποιείται για συγκεκριμένες ερωτήσεις σχετικά με την πληροφορία που κρατείται με την κατάσταση ενός peer, ενώ το PDP χρησιμεύει για να ανακαλύψει άλλους peers που βρίσκονται στην ίδια κοινότητα. Το PRP μπορεί να χρησιμοποιηθεί για γενικής χρήσης ερωτήσεις, οι οποίες μπορεί να είναι χρήσιμες για μια εφαρμογή. Για παράδειγμα το PRP επιτρέπει στους peers να ορίσουν και να ανταλλάξουν ερωτήσεις για να βρουν ή να αναζητήσουν πληροφορίες όπως είναι η κατάσταση μιας υπηρεσίας ή η κατάσταση του σημείου τέλους ενός σωλήνα.

### 3.4.5 Πρωτόκολλο Ραντεβού

Το πρωτόκολλο ραντεβού (*Rendezvous Protocol – RVP*) είναι υπεύθυνο για τη διάδοση μηνυμάτων μέσα σε μια κοινότητα. Ενώ διαφορετικές κοινότητες από peers μπορούν να έχουν διαφορετικούς τρόπους για να διαδώσουν μηνύματα, το RVP ορίζει ένα απλό πρωτόκολλο το οποίο επιτρέπει:

- Τους peers να συνδέονται στην υπηρεσία (να μπορούν να διαδώσουν μηνύματα και να λάβουν μηνύματα διάδοσης).
- Να ελέγχουν τη διάδοση του μηνύματος.

Το RVP χρησιμοποιείται από το PRP και το PBP για να μπορέσουν να διαδώσουν μηνύματα.

### 3.4.6 Πρωτόκολλο Δρομολόγησης Κατάληξης

Το πρωτόκολλο δρομολόγησης κατάληξης (*Endpoint Routing Protocol – ERP*) ορίζει ένα σύνολο από αιτήματα, ερωτήσεις τα οποία χρησιμοποιούνται για να βρεθεί

πληροφορία δρομολόγησης. Αυτή η πληροφορία δρομολόγησης είναι απαραίτητη για να σταλεί ένα μήνυμα από ένα peer σε ένα άλλο. Όταν ζητείται από ένα peer να στείλει ένα μήνυμα σε μια διεύθυνση κατάληξης ενός άλλου peer, πρέπει πρώτα να κοιτάξει στην τοπική λανθάνουσα μνήμη για να διαπιστώσει εάν έχει μια διαδρομή προς τον peer. Εάν δε βρει κάποια διαδρομή, τότε στέλνει ένα αίτημα ερώτησης διαδρομής σε όλους τους διαθέσιμους αναμεταδότες peers, ζητώντας πληροφορία δρομολόγησης. Όταν ένας αναμεταδότης peer λάβει μια ερώτηση διαδρομής, ελέγχει εάν γνωρίζει τη διαδρομή. Εάν τη γνωρίζει, θα επιστρέψει την πληροφορία για τη διαδρομή σαν μια απαρίθμηση των hops. Οι αναμεταδότες peers περιέχουν αποθηκευμένες πληροφορίες δρομολόγησης.

Η πληροφορία δρομολόγησης εμπεριέχει το αναγνωριστικό του peer αποστολέα, το αναγνωριστικό του peer παραλήπτη, ένα χρονικό διάστημα ζωής για τη διαδρομή, και μια ταξινομημένη ακολουθία από αναγνωριστικά των peers. Η ακολουθία από αναγνωριστικά των peers μπορεί να μην είναι ολοκληρωμένη, αλλά θα πρέπει να περιέχει τουλάχιστον τον πρώτο αναμεταδότη peer.

### 3.5 Πλεονεκτήματα και μειονεκτήματα της τεχνολογίας

Η πλατφόρμα JXTA παρέχει πολλά πλεονεκτήματα από τα οποία μπορούν ωφεληθούν τόσο οι προγραμματιστές εφαρμογών όσο και οι χρήστες. Το JXTA είναι μια peer-to-peer πλατφόρμα, γεγονός που σημαίνει ότι δεν υπάρχει η ανάγκη για ύπαρξη κεντρικού εξυπηρετητή. Το JXTA προμηθεύει την απαιτούμενη τεχνολογία για δημιουργία κοινοτήτων από peers, οι οποίοι μπορούν να επικοινωνούν και να ανταλλάσσουν αρχεία πάνω από μια τεράστια δικτυακή περιοχή.

Η τεχνολογία JXTA παρέχει μια περισσότερο αφαιρετική γλώσσα για επικοινωνία μεταξύ peers. Με τον τρόπο αυτό, επιτρέπεται σε μια ευρύτερη ποικιλία από υπηρεσίες, συσκευές, και δίκτυα μεταφοράς να συμμετέχουν σε δίκτυα peer-to-peer. Η χρησιμοποίηση της XML έχει σαν αποτέλεσμα ένα σχήμα για δομημένα δεδομένα το οποίο είναι εύκολα κατανοητό, καλά υποστηριζόμενο, και εύκολα προσαρμόζεται σε μεγάλη ποικιλία από μεταφορές. Η XML επιπλέον έχει το πλεονέκτημα ότι είναι εύκολα αναγνώσιμη από τον άνθρωπο, διευκολύνοντας έτσι τους μηχανικούς λογισμικού να κάνουν την εκσφαλμάτωση και την κατανόηση.

Ένα σημαντικό στοιχείο που δεν προσπαθεί το JXTA να επιλύσει είναι ο τρόπος επίκλησης των υπηρεσιών. Διάφορα πρότυπα υπάρχουν για τον ορισμό της επίκλησης των υπηρεσιών, όπως είναι η Web Services Description Language (WSDL) [43]. Το JXTA παρέχει ένα γενικό πλαίσιο εργασίας για ανταλλαγή πληροφορίας μεταξύ των peers, γεγονός που δίνει τη δυνατότητα σε κάθε μηχανισμό, όπως είναι το WSDL, να

μπορεί να χρησιμοποιηθεί από το JXTA για ανταλλαγή πληροφοριών που απαιτούνται και για επίκληση των υπηρεσιών.

Ένα όφελος από τη χρήση του πλαισίου εργασίας JXTA είναι η διαλειτουργικότητα. Με την παροχή ενός βασικού πλαισίου εργασίας πάνω στο οποίο μπορούν να αναπτυχθούν πολλοί διαφορετικοί τύποι από peer-to-peer εφαρμογές, πετυχαίνεται ο στόχος της συμβατότητας μεταξύ των εφαρμογών και των υπηρεσιών. Ένα επιπλέον θετικό στοιχείο που πηγάζει από τη διαλειτουργικότητα είναι ότι επιτρέπει στους μηχανικούς λογισμικού να αφιερώνουν περισσότερο χρόνο στην ανάπτυξη εφαρμογών και όχι στη δημιουργία λογισμικού και αρχών που ήδη έχουν αναπτυχθεί.

Πλεονέκτημα επίσης της πλατφόρμας JXTA είναι η ανεξαρτησία της από πλατφόρμες γεγονός που διευκολύνει τους μηχανικούς λογισμικού, που δε χρειάζεται να ανησυχούν για τη συγκεκριμένη πλατφόρμα στην οποία τρέχει ο χρήστης την εφαρμογή. Επομένως, από τη στιγμή που θα γραφτεί μια εφαρμογή ή μια υπηρεσία δεν θα πρέπει να ξαναγραφτεί ή να αλλαχθεί προκειμένου να τρέξει σε διαφορετικές πλατφόρμες.

Η πλατφόρμα JXTA πετυχαίνει αναμφίβολα να λύσει και το εύρος των συσκευών που μπορούν να είναι συνδεδεμένες σε peer-to-peer δίκτυο, αφού μια οποιαδήποτε συσκευή επιτρέπεται να λειτουργεί σαν κόμβος του δικτύου. Αυτό πρακτικά σημαίνει ότι ένας peer μπορεί να είναι ένας ηλεκτρονικός υπολογιστής, ένα PDA, ένα κινητό τηλέφωνο κ.α.. Τέλος ένα σημαντικό πρόβλημα των σημερινών peer-to-peer συστημάτων είναι ότι τα περισσότερα υποστηρίζουν μόνο το πρωτόκολλο TCP/IP, κάτι που δε συμβαίνει με το JXTA, το οποίο είναι ανεξάρτητο από το μέσο μεταφοράς.

Μειονέκτημα ασφαλώς της τεχνολογίας JXTA είναι ότι σε περιπτώσεις όπου έχουμε ευρεία μετάδοση ενός αιτήματος μπορεί να παρατηρηθεί κάμψη της απόδοσης λόγω συμφόρησης καθώς αδύναμοι κόμβοι όπως οι συνδέσεις διαποδιαμορφωτών θα καθυστερήσουν τη διάδοση του αιτήματος. Επιπλέον, δημοφιλείς ραντεβού κόμβοι οι οποίοι μπορούν να έχουν πολλές ταυτόχρονες συνδέσεις μπορούν να αυξήσουν το χρόνο που απαιτείται για να γίνει η σύνδεση μεταξύ δυο peer, λόγω φόρτου εργασίας. Αντιλαμβανόμαστε επομένως ότι κάποιοι κόμβοι μπορούν να υποφέρουν από συμφόρηση.

Ένα πρόβλημα που μπορεί να εμφανιστεί ακόμα στις εφαρμογές peer-to-peer που αναπτύσσονται με την πλατφόρμα JXTA είναι η ποιότητα των χρηστών που προσελκύουν. Το πρόβλημα αυτό εμφανίζεται στις περισσότερες peer-to-peer εφαρμογές και έχει να κάνει με τη διάθεση των χρηστών να χρησιμοποιούν το σύστημα μόνο για να λαμβάνουν αρχεία. Το πρόβλημα αυτό μειώνει τη συνολική απόδοση του συστήματος τόσο σε ότι αφορά την απόδοση μεταφοράς αρχείων όσο και στην ποιότητα των αρχείων που διαμοιράζονται. Έχει παρατηρηθεί ότι στο Gnutella το 70% των χρηστών δε διαμοιράζονται αρχεία [37].

Αν και η χρήση της XML καθορίζει όλα τα θέματα επικοινωνίας για κάθε εφαρμογή peer-to-peer, το JXTA μπορεί να μην είναι κατάλληλο για συγκεκριμένες αυτόνομες

peer-to-peer εφαρμογές. Σε μια μεμονωμένη εφαρμογή, το δικτυακό επιπλέον κόστος των XML μηνυμάτων μπορεί να αποτελεί πρόβλημα, ιδιαίτερα αν το άτομο που αναπτύσσει την εφαρμογή δεν έχει πρόθεση να επωφεληθεί από τις δυνατότητες του JXTA να ενσωματώσει άλλες peer-to-peer υπηρεσίες στην εφαρμογή. Τέλος, το JXTA παρέχει μια πλατφόρμα για παραγωγή peer-to-peer εφαρμογών που έχουν την απαιτούμενη ευκαμψία για να μεγαλώσουν στο μέλλον. Η ικανότητα να επιδέχεται άλλες peer-to-peer υπηρεσίες και το γεγονός ότι επιτρέπεται η εκτεταμένη ανάπτυξη των peer-to-peer κοινοτήτων είναι η κυριότερη ωφέλεια που προσφέρει η πλατφόρμα JXTA.

### 3.6 Διαφορές της τεχνολογίας JXTA από την Jini και .NET

Η διασύνδεση κάθε τύπου συσκευής πάνω από κάθε τύπο δικτύου είναι μια αρχή που εκτός από την τεχνολογία JXTA ακολουθεί και η τεχνολογία Jini [20] της εταιρίας Sun. Αν και οι στόχοι είναι όμοιοι η διαφορά έγκειται στο γεγονός ότι η τεχνολογία Jini βασίζεται αποκλειστικά στην τεχνολογία JAVA [19] για τη λειτουργικότητά της, ενώ η τεχνολογία JXTA δεν εξαρτάται από κάποια ιδιαίτερη γλώσσα προγραμματισμού. Αντίθετα με την τεχνολογία JXTA, η τεχνολογία Jini χρησιμοποιεί ένα κεντρικό εξυπηρετητή για να εντοπίζει υπηρεσίες στο δίκτυο και βασίζεται στο Remote Method Invocation (RMI) και στη σειριακή διάταξη των αντικειμένων για επικοινωνία με απομακρυσμένες συσκευές. Η τεχνολογία JXTA βασίζεται στην XML για την ανταλλαγή δομημένων δεδομένων και όχι στη σειριακή διάταξη των αντικειμένων, και ανακαλύπτει υπηρεσίες διαμέσου όλων των peers στο peer-to-peer δίκτυο.

Το θέμα των υπηρεσιών ιστού της πλατφόρμα .NET [29] της εταιρίας Microsoft είναι εμποτισμένο με τη χρήση XML, αλλά από μόνη της η χρήση της XML δεν κάνει τις δυο τεχνολογίες ισοδύναμες, και συγκρίσιμες. Οι τεχνολογίες JXTA και .NET έχουν εντελώς διαφορετικούς σκοπούς. Η τεχνολογία .NET εστιάζει περισσότερο στην παραδοσιακή αρχιτεκτονική πελάτη / εξυπηρετητή σε ότι αφορά την παράδοση των υπηρεσιών. Αν και η τεχνολογία .NET θα μπορούσε να αποτελέσει τη βάση για ανάπτυξη μιας peer-to-peer εφαρμογής, θα απαιτούσε αρκετή επιπλέον εργασία από τη μεριά του μηχανικού λογισμικού. Η ανάπτυξη μιας ολοκληρωμένης λύσης peer-to-peer με τη χρήση .NET απαιτεί από το μηχανικό λογισμικού να καθορίσει όλα τις θεμελιώδεις αλληλεπιδράσεις ενός peer-to-peer συστήματος, όπως για παράδειγμα είναι η ανακάλυψη peer. Επομένως, η επιλογή αυτής της λύσης περιλαμβάνει την επαναδημιουργία όλων των μηχανισμών που είναι ήδη ορισμένα από τα πρωτόκολλα JXTA.

## 4 Περιγραφή του RDPeer

Στα πλαίσια αυτής της εργασίας υλοποιήθηκε ένα peer-to-peer σύστημα που σκοπό έχει να βελτιώσει το διαμοιρασμό σχεσιακών δεδομένων σε ομότιμα δίκτυα, μέσω της αποδοτικότερης αναζήτησης και εύρεσης απαντήσεων σε SQL επερωτήσεις. Η εφαρμογή RDPeer που υλοποιήθηκε είναι ένα καταναμημένο σύστημα, το οποίο επιτρέπει στους χρήστες που έχουν αποθηκευμένα δεδομένα σε σχεσιακές βάσεις δεδομένων να αναζητήσουν, να ανακαλύψουν και να εξάγουν δεδομένα μέσα σε ένα peer-to-peer περιβάλλον.

### 4.1 Peer Σχεσιακών δεδομένων (RDPeer)

Η εφαρμογή που θα παρουσιαστεί είναι ένα peer-to-peer σύστημα για καταναμημένο διαμοιρασμό των δεδομένων. Ο διαμοιρασμός των δεδομένων, που βρίσκονται στη σχεσιακή βάση δεδομένων κάθε peer, γίνεται χωρίς τη γνώση ενός συνολικού σχήματος. Ουσιαστικά, πρόκειται για μια εφαρμογή βάσεων δεδομένων υλοποιημένη πάνω από την πλατφόρμα JXTA. Το RDPeer επιτρέπει στο χρήστη να συμμετέχει σε κοινότητες από peers, μέσα στις οποίες μπορεί να θέτει SQL επερωτήσεις και να παίρνει απαντήσεις από τους peers που έχουν δεδομένα σχετικά με τις επερωτήσεις. Σε κάθε κοινότητα ένας κόμβος μπορεί να συμμετάσχει είτε σαν απλός peer, είτε σαν super-peer.

Ο super-peer έχει επιπλέον λειτουργικότητα, η οποία είναι ενσωματωμένη σε οποιοδήποτε peer, και επιλέγεται δυναμικά στα πλαίσια μιας κοινότητας με τη διενέργεια εκλογών που μπορεί να διεξαχθούν μεταξύ των μελών μιας κοινότητας. Η τεχνική της εκλογής ενός νέου super-peer που σχεδιάστηκε και υλοποιήθηκε στοχεύει στην ελαχιστοποίηση των μηνυμάτων που διακινούνται μέσα στο δίκτυο αμέσως μετά τις εκλογές και στην αποφυγή λανθασμένων καταστάσεων που μπορούν να δημιουργηθούν κατά τη διεξαγωγή των εκλογών. Για να επιτύχουμε τις δύο αυτές θεωρήσεις παίρνουμε υπόψη για τη μεν πρώτη το πλήθος των επερωτήσεων που έχει θέσει ένας peer και για τη δεύτερη βασιζόμαστε στο μοναδικό αναγνωριστικό που φέρει κάθε peer.

Ο super-peer επιφορτίζεται με τη λειτουργία της συλλογής των σχημάτων των βάσεων δεδομένων από τους peers, της εύρεσης αντιστοιχίσεων μεταξύ αυτών και μίας SQL επερώτησης και το μετασχηματισμό της επερώτησης. Τα σχήματα που συλλέγονται από τον εκάστοτε super-peer αποθηκεύονται για να ελαχιστοποιήσουμε τα μηνύματα που ανταλλάσσουν οι peers του δικτύου μεταξύ τους και για να βελτιώσουμε την απόδοση του συστήματος. Η κοινοποίηση επομένως του σχήματος από ένα peer μέσα σε μια

κοινότητα αρκεί να γίνει μόνο μια φορά. Η επεξεργασία μιας επερώτησης υποβοηθείται από web service που διαθέτει κάθε peer του δικτύου.

Πρακτικά τα περισσότερα peer-to-peer συστήματα είναι σχεδιασμένα να υποστηρίζουν διαμοιρασμό των δεδομένων στο επίπεδο των αρχείων και εγγράφων [11,15,23,28]. Στην παρούσα εργασία αντίθετα με προηγούμενες που έγιναν στο τομέα της κατανεμημένης διαχείρισης δεδομένων, αναπτύσσουμε ένα σύστημα για το διαμοιρασμό των δεδομένων χωρίς ένα προκαθορισμένο σχήμα. Επιπλέον, το σύστημα δεν περιορίζεται σε ότι αφορά την επεκτασιμότητά του και την ευκαμψία του, και οι peers δεν ορίζονται στατικά. Συνακόλουθα, προσέχουμε να μην έχουμε μεγάλη επιβάρυνση της κίνησης στο δίκτυο, ώστε να έχουμε καλύτερη απόδοση στο σύστημά μας.

Στο χώρο της κατανεμημένης διαχείρισης δεδομένων σε peer-to-peer συστήματα υπάρχει ένα θεμελιώδες πρόβλημα, που έγκειται στην τοποθέτηση και ανάκτηση των δεδομένων και το οποίο επηρεάζεται από κάποιες σημαντικές παραμέτρους [14]. Το πρόβλημα έχει να κάνει με την κατανομή των δεδομένων και της εργασίας, έτσι ώστε μια επερώτηση να απαντηθεί με το μικρότερο κόστος από πλευράς πόρων και περιορισμών σε bandwidth. Το πρόβλημα της τοποθέτησης δεδομένων ενώ δείχνει να παρουσιάζει πολλές ομοιότητες με τη βελτιστοποίηση πολλαπλών επερωτήσεων σε κατανεμημένη βάση δεδομένων, στην πραγματικότητα υπάρχουν σημαντικές διαφορές. Ένα peer-to-peer σύστημα δεν έχει ένα κεντρικοποιημένο σχήμα ούτε κεντρική διαχείριση. Επιπλέον, υπάρχουν αρκετοί περιορισμοί και συμβιβασμοί που γίνονται εξαιτίας παραγόντων όπως περιορισμένο bandwidth και πόροι, καθυστερήσεις στη διάδοση των μηνυμάτων, προβλήματα συντονισμού και συνεργασίας, δυναμική φύση των peers, επεκτασιμότητα του συστήματος, κ.α.

Το RDPeer προσπαθεί να επιλύσει κάποιες σημαντικές διαστάσεις που επηρεάζουν το πρόβλημα της τοποθέτησης των δεδομένων με τη δυναμική δημιουργία των super-peers, δηλαδή τη χρήση ιεραρχικής οργάνωσης των peers. Οι super-peers συμβάλλουν στην επεκτασιμότητα του διαμοιρασμού γνώσης, στην αξιοπιστία του συστήματος, και στην καλύτερη απόδοση της αναζήτησης δεδομένων. Η εφαρμογή που υλοποιήσαμε δε θέτει περιορισμούς στην ετερογένεια των πηγών δεδομένων και στη δυναμική συμμετοχή των peers στις κοινότητες. Τέλος, τα αντίγραφα των δεδομένων που υπάρχουν μέσα σε ένα peer-to-peer σύστημα περιορίζονται μόνο στην πηγή των δεδομένων, γεγονός που επιτρέπει στο σύστημα να μην παρουσιάζει προβλήματα ενημέρωσης, και εγκυρότητας των δεδομένων. Η τακτική αυτή προσβλέπει στην ομοιομορφία, αποτελεσματικότητα, σταθερότητα του συστήματος, καθώς και στην καλύτερη απόδοσή του, ενώ με τον τρόπο αυτό δεν εμφανίζεται επιβάρυνση στην κίνηση του δικτύου. Ακολουθώντας την υβριδική (hybrid) αρχιτεκτονική στην υλοποίηση του peer-to-peer συστήματος επιτρέπουμε σε κάθε peer να συνδέεται σε οποιοδήποτε άλλο peer της κοινότητας για να λάβει δεδομένα,



και καθορίζουμε το ρόλο του super-peer ως του ενδιάμεσου φορέα αναζήτησης των peers που διαθέτουν απαντήσεις σε SQL ερωτήσεις.

Ένας από τους κύριους στόχους του RDPeer είναι να επιτρέψει τους χρήστες να διαχειριστούν τα δεδομένα τους χρησιμοποιώντας ένα σύστημα διαχείρισης βάσης δεδομένων (DBMS). Επειδή διαφορετικοί χρήστες ορίζουν διαφορετικά τα δεδομένα ακόμα και αν ενδιαφέρονται για δεδομένα από μια κοινή περιοχή, θα ήταν δύσκολο για την εφαρμογή να εντοπίσει δεδομένα αν χρησιμοποιούνταν η παραδοσιακή τεχνική της ακριβής ταύτισης των ονομάτων των πεδίων και των σχέσεων. Συνεπώς, η αντιστοίχιση σχημάτων είναι μια απαραίτητη διαδικασία που διενεργείται από το RDPeer και συγκεκριμένα στο επίπεδο του super-peer.

Η αντιστοίχιση σχημάτων τυπικά επιτυγχάνεται χειρωνακτικά ή με τη βοήθεια μιας γραφικής διεπιφάνειας χρήσης, γεγονός που την καθιστά μια χρονοβόρα και ακριβή διαδικασία. Πολλές τεχνικές έχουν προταθεί για να επιτευχθεί ένας μερικός αυτοματισμός στη δημιουργία αντιστοιχίσεων μεταξύ σχημάτων [34]. Η τεχνική που υιοθετείται στα πλαίσια της εργασίας είναι η γλωσσολογική προσέγγιση [26]. Για κάθε σχέση που έχει δημιουργηθεί από ένα χρήστη, διατηρούνται μεταδεδομένα τόσο για το όνομα της σχέσης όσο και για τα ονόματα πεδίων. Τα μεταδεδομένα είναι συνώνυμα, υπερώνυμα των ονομάτων των σχέσεων και πεδίων και λέξεις κλειδιά που δίνει ο χρήστης, έτσι ώστε να επιτευχθεί μια πλουσιότερη περιγραφή του σχήματος. Η γλωσσολογική προσέγγιση εφαρμόζεται στο RDPeer και χρησιμοποιεί τα ονόματα των σχέσεων και πεδίων και τα μεταδεδομένα για να μπορέσει να ανακαλύψει αντιστοιχίσεις μεταξύ των σχημάτων και των SQL ερωτήσεων. Συνεπώς, δε δημιουργούνται αντιστοιχίσεις μεταξύ των διαφορετικών σχημάτων, επειδή πρώτον υπάρχει η περίπτωση να απομονωθούν οι πηγές που έχουν ετερογένεια πληροφορίας σε σχέση με τις υπόλοιπες και δεύτερον γιατί πολλές φορές πεδία διαφορετικών σχημάτων δε θα ήταν εφικτό να αντιστοιχηθούν χωρίς την ανθρώπινη παρέμβαση, γεγονός που θεωρήσαμε σημαντικό μειονέκτημα για ένα δυναμικό, και αυθαίρετο περιβάλλον χωρίς την ύπαρξη διαχειριστή.

Οι αντιστοιχίσεις δημιουργούνται στη διάρκεια αναζήτησης των peers της κοινότητας που μπορούν να απαντήσουν σε μια ερώτηση, και για να διενεργηθεί ο μετασχηματισμός της εκάστοτε SQL ερώτησης. Η εφαρμογή παρέχει τη δυνατότητα του μετασχηματισμού ενός SQL αιτήματος (query transformation), ώστε να ταιριάζει στο σχήμα της εκάστοτε βάσης δεδομένων που θα απευθυνθεί. Ο μετασχηματισμός είναι απαραίτητος για να μπορούν να εξαχθούν δεδομένα από τη σχεσιακή βάση δεδομένων κάθε peer, που είναι σε θέση να απαντήσει στην ερώτηση. Η διαδικασία του μετασχηματισμού, όπως και η εύρεση αντιστοιχίσεων διενεργείται στο επίπεδο του super-peer. Αυτό έχει βέβαια το μειονέκτημα ότι ο φόρτος εργασίας του συγκεκριμένου peer αυξάνεται, αλλά πλεονεκτεί στο γεγονός της ταχύτητας επεξεργασίας, του συντονισμού, και της μικρότερης επιβάρυνσης της κίνησης του δικτύου.

Το JXTA όπως αναφέραμε εκτενώς στο προηγούμενο κεφάλαιο παρέχει ένα σύνολο από ανοιχτά πρωτόκολλα και μια ανοιχτού λογισμικού υλοποίηση για την ανάπτυξη peer-to-peer εφαρμογών. Τα πρωτόκολλα του JXTA τυποποιούν τον τρόπο με τον οποίο οι peers:

- Ανακαλύπτουν ο ένας τον άλλο.
- Οργανώνονται σε κοινότητες από peers.
- Δημοσιοποιούν και ανακαλύπτουν δικτυακές υπηρεσίες.
- Επικοινωνούν μεταξύ τους.
- Παρακολουθούν και ελέγχουν ο ένας τον άλλο.

Το RDPeer επεκτείνοντας τις λειτουργίες της πλατφόρμας JXTA επιτρέπει σε κάθε κόμβο του δικτύου να διαχειρίζεται μια βάση δεδομένων και να δημιουργεί ένα θησαυρό όρων βασισμένο στο σχήμα της βάσης δεδομένων που κατέχει, έτσι ώστε να επιτύχει την καλύτερη περιγραφή της. Επιπλέον, παρέχεται σε κάθε peer η δυνατότητα να κάνει τις δικές του δημοσιοποιήσεις. Κάθε peer συνεπώς, μπορεί να παράγει τρία ξεχωριστά XML έγγραφα, που το μεν πρώτο αναφέρεται στο σχήμα της σχεσιακής βάσης δεδομένων και τα μεταδεδομένα που διαθέτει, το δεύτερο περιγράφει το web service και πως μπορεί να γίνει η επίκλησή του, ενώ το τρίτο δημιουργείται και εκδίδεται μόνο στην περίπτωση που ο κόμβος είναι ένας super-peer και θα πρέπει να ενημερώσει όλα τα υπόλοιπα μέλη της κοινότητας για αυτή την ιδιότητά του. Συνεπώς, το RDPeer με την υλοποίηση του web service επιτρέπει την απομακρυσμένη επίκληση μεθόδων, γεγονός που συμβάλει στην καλύτερη ισορροπία του φόρτου εργασίας του συστήματος, καθώς η εξαγωγή των δεδομένων δε γίνεται κεντροποιημένα αλλά με ισότιμο τρόπο από όλα τα μέλη.

Το RDPeer είναι ένας πολυνηματικός εξυπηρετητής (multithreaded server) όπως είναι οι περισσότεροι κόμβοι που συμμετέχουν σε peer-to-peer συστήματα, και κάθε peer υλοποιεί το μοντέλο πελάτη / εξυπηρετητή. Στη συνέχεια θα παρουσιαστούν τα ιδιαίτερα χαρακτηριστικά του συστήματος. Αρχικά θα παρουσιάσουμε τον τρόπο λειτουργίας της εφαρμογής.

#### 4.1.1 Γενική περιγραφή λειτουργίας RDPeer

Το RDPeer είναι ένα peer-to-peer σύστημα που προσπαθεί να διευκολύνει τους χρήστες να θέτουν SQL ερωτήσεις σε βάσεις δεδομένων που δεν κατέχουν και να λαμβάνουν όσο πιο έγκυρα και έγκαιρα δεδομένα. Το πρόβλημα που είχαμε ως αρχικό κίνητρο για την υλοποίηση του RDPeer εμφανίζεται στη διευθέτηση του θέματος του διαμοιρασμού των κατανεμημένων δεδομένων.

Το RDPeer διενεργεί εννοιολογική σύγκριση των ονομάτων των πεδίων, σχέσεων μιας SQL ερώτησης με τα σχήματα των βάσεων δεδομένων που διαθέτουν οι peers,

ώστε να ανακαλύψει αυτούς που θα μπορούσαν να το απαντήσουν. Η διαδικασία της σύγκρισης γίνεται στο επίπεδο του ενδιαμέσου λογισμικού και υπεύθυνος για τη λειτουργία αυτή είναι ο *super-peer*. Από τη στιγμή που θα τελειώσει η διαδικασία εύρεσης των πιθανών *peers* που μπορεί να διαθέτουν δεδομένα που ικανοποιούν την επερώτηση, η επεξεργασία της επερώτησης ανατίθεται στο *peer* που την αρχικοποίησε. Αυτός αναλαμβάνει να επικοινωνήσει με τις πηγές των δεδομένων (*peers* που μπορούν να ανταποκριθούν στην επερώτηση), οι οποίες αναλαμβάνουν για λογαριασμό του να εξάγουν τα δεδομένα από τις βάσεις δεδομένων τους και να του τα επιστρέψουν.

Η διαδικασία που ακολουθείται από το RDPeer εμφανίζεται κεντρικοποιημένη μόνο στο επίπεδο της εύρεσης αντιστοιχίσεων και του μετασχηματισμού μιας SQL επερώτησης (*query transformation*), ώστε να ταιριάζει στο σχήμα της εκάστοτε βάσης δεδομένων κάθε *peer*. Η διαδικασία αυτή λαμβάνει χώρα στο *super-peer* και είναι απαραίτητη για να γίνει ολοκλήρωση των δεδομένων και επιλέξαμε να είναι κεντρικοποιημένη για να έχουμε μικρότερη επιβάρυνση στην κίνηση του δικτύου, και καλύτερη απόδοση του συστήματος. Επιπλέον, κάθε *peer* δε χρειάζεται να ξοδεύει υπολογιστικούς πόρους ή να κάνει *caching* δεδομένα άλλων *peer*. Η υπόλοιπη λειτουργία της εφαρμογής είναι πλήρως κατανεμημένη.

Θεωρούμε ότι κάθε ένας κόμβος του *peer-to-peer* δικτύου διαθέτει μια σχεσιακή βάση δεδομένων και επιθυμεί να κάνει μια επερώτηση για δεδομένα, τα οποία είτε βρίσκονται στη βάση δεδομένων που έχει είτε όχι και ότι χρήστης του κόμβου είναι γνώστης της SQL. Επίσης, θεωρούμε ότι κάθε *peer* του δικτύου JXTA είναι τουλάχιστον ένας πλήρους λειτουργικότητας *peer*, δηλαδή έχει τη δυνατότητα να στείλει και να λάβει μηνύματα καθώς και να αποθηκεύσει δημοσιοποιήσεις. Στη συνέχεια, θα αναλυθεί εκτενώς η λειτουργία της εφαρμογής.

#### 4.1.2 Αρχικοποίηση του RDPeer

Η εφαρμογή RDPeer ξεκινώντας αρχικοποιεί την πλατφόρμα JXTA. Ο χρήστης πιστοποιεί την αυθεντικότητά του μέσω του ονόματος χρήστη και του κωδικού που έχει επιλέξει. Με το τρόπο αυτό ο *peer* εισέρχεται στο JXTA δίκτυο έχοντας ένα μοναδικό αναγνωριστικό που τον διαχωρίζει από όλους τους υπόλοιπους. Εισερχόμενος ο *peer* εντάσσεται στο Net Peer Group που είναι η πρωταρχική και κοινή κοινότητα, στην οποία αρχικά ανήκουν όλοι οι *peers*. Οι πρώτες λειτουργίες που κάνει ένας *peer* υποχρεωτικά αμέσως μετά την ένταξη του είναι η έναρξη του Jakarta-Tomcat web server [18] στον οποίο τρέχει ουσιαστικά το web service, και η δημιουργία και έκδοση μιας δημοσιοποίησης στην οποία περιγράφεται ο τρόπος με τον οποίο θα έχουν πρόσβαση όλοι οι υπόλοιποι *peers* στο web service. Η δημοσιοποίηση είναι ένα XML έγγραφο στο οποίο αναφέρεται το όνομα του *peer*, το URL στο οποίο μπορεί να γίνει η επίκληση της

υπηρεσίας και το αναγνωριστικό της υπηρεσίας. Το αναγνωριστικό όπως φαίνεται και παρακάτω αποτελείται από το όνομα του peer και τη λέξη «υπηρεσίες» (services). Έχει προκαθοριστεί για λόγους ομοιομορφίας όλοι οι peers του δικτύου να ακολουθούν την ίδια ονοματολογία.

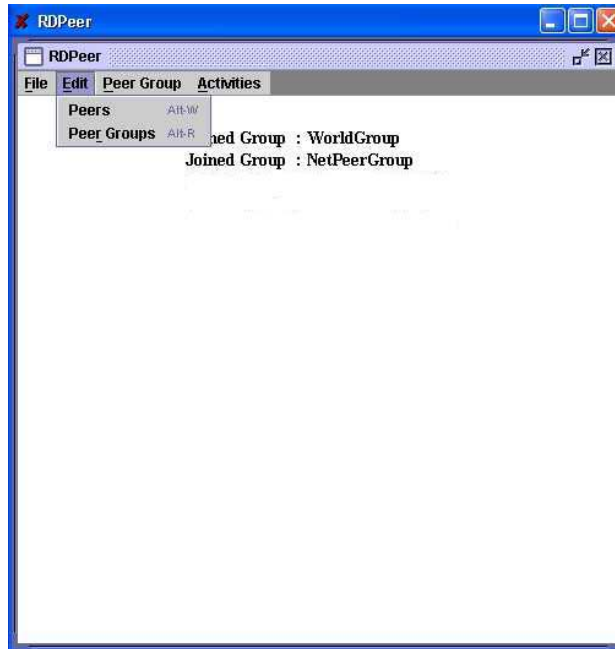
`http://calliope.ics.forth.gr:10001/soap/servlet/rpcrouter`  
`urn:peer2-rdv-services`

Το web service που φέρει κάθε peer έχει μια μόνο μέθοδο την οποία μπορούν να καλέσουν εξ αποστάσεως οι άλλοι peers της ίδιας κοινότητας. Η μέθοδος αυτή προσφέρει τη δυνατότητα σε κάποιο peer να δώσει σαν παράμετρο μια SQL επερώτηση και να λάβει ως απάντηση τα δεδομένα που βρίσκονται στη σχεσιακή βάση δεδομένων. Επιπλέον, ο εκάστοτε peer εκτός από τις δύο αυτές λειτουργίες κάνει δημοσιοποίηση της ύπαρξης του (έκδοση ενός peer advertisement) και ψάχνει να βρει ραντεβού peers για να συνδεθεί. Όπως είχαμε αναφέρει και στο προηγούμενο κεφάλαιο, κάθε peer διαθέτει μια τοπική cache, στην οποία αποθηκεύει δημοσιοποιήσεις. Η σύνδεση του peer με τους ραντεβού peers είναι αυτή που θα τον βοηθήσει να αποκτήσει δημοσιοποιήσεις, που έχουν γίνει μέχρι εκείνη τη χρονική στιγμή στο δίκτυο. Στην περίπτωση που δεν καταφέρει να βρει ραντεβού peers, τότε γίνεται αυτόματα ο ίδιος.

Ο peer εισερχόμενος στο δίκτυο JXTA έχει αρχικά τις ακόλουθες δυνατότητες, επιλογές στη διάθεση του όπως φαίνονται και στις δύο παρακάτω εικόνες:

1. Εύρεση όλων των peers που βρίσκονται στο δίκτυο JXTA
2. Εύρεση όλων των κοινοτήτων που έχουν δημιουργηθεί μέχρι εκείνη τη στιγμή
3. Δημιουργία μιας νέας κοινότητας και ταυτόχρονη δημοσιοποίηση της.
4. Συμμετοχή σε μια κοινότητα.
5. Εγκατάλειψη μιας κοινότητας στην οποία ήταν μέλος.

Οι επιλογές αυτές είναι πάντα στη διάθεση κάθε peer ανεξαρτήτως της κοινότητας, στην οποία μπορεί να ανήκει.



Εικόνα 6 Οι επιλογές αναζήτησης των peers, και των κοινοτήτων.



Εικόνα 7 Οι επιλογές δημιουργίας, συμμετοχής και εγκατάλειψης μιας κοινότητας από peers.

### 4.1.3 Προεργασία για το διαμοιρασμό των δεδομένων

Κάθε peer μπορεί να επιλέξει μια κοινότητα στην οποία επιθυμεί να γίνει μέλος ή να δημιουργήσει μία δική του. Όταν ένας peer δημιουργεί μία κοινότητα αυτόματα γίνεται και super-peer σε αυτή. Εισερχόμενος ένας super-peer στην κοινότητα που έχει δημιουργήσει εκτελεί τρεις βασικές λειτουργίες που είναι απαραίτητες για την ομαλή λειτουργία και τη σταθερότητα του συστήματος.

1. Εκδίδει μια δημοσιοποίηση η οποία τον χαρακτηρίζει ως super-peer για τη συγκεκριμένη κοινότητα και την οποία πρέπει υποχρεωτικά να κατέχει κάθε μέλος της κοινότητας.
2. Εκδίδει μια δημοσιοποίηση σωλήνα. Η κοινοποίηση αυτή περιγράφει ένα κανάλι επικοινωνίας και χρησιμοποιείται από την υπηρεσία σωλήνωσης για να δημιουργήσει τα σχετιζόμενα σημεία κατάληξης (endpoints) εισόδου, εξόδου της σωλήνας. Κάθε δημοσιοποίηση σωλήνα περιέχει ένα προαιρετικό συμβολικό αναγνωριστικό, ένα τύπο σωλήνα και ένα μοναδικό αναγνωριστικό σωλήνα. Με τη χρήση αυτή της δημοσιοποίησης θα μπορεί κάθε μέλος της κοινότητας να συνδέεται στον super-peer και να του θέτει επερωτήσεις. Η εύρεση της συγκεκριμένης δημοσιοποίησης είναι υποχρεωτική από κάθε μέλος.
3. Θέτει σε λειτουργία δύο πολυνηματικούς εξυπηρετητές (multithreaded servers). Ο πρώτος server έχει ως σκοπό να δέχεται και να εξυπηρετεί SQL επερωτήσεις από τους peers. Αφού δεχτεί μια επερώτηση, βρίσκει τους peers που μπορούν να το απαντήσουν, διενεργεί το μετασχηματισμό της SQL επερώτησης, ώστε να ταιριάζει στο σχήμα της εκάστοτε βάσης δεδομένων, και εν τέλει αποστέλλει τις μετασχηματισμένες επερωτήσεις που προέκυψαν από την αρχική που δέχτηκε στον peer που έθεσε την επερώτηση. Ο δεύτερος server συμβάλλει στο να γνωρίζει ο super-peer τα τρέχον μέλη της κοινότητας. Αυτό γίνεται με την αποστολή ping pong μηνυμάτων. Η εύρεση των ενεργών μελών μιας κοινότητας είναι μια διαδικασία μείζονος σημασίας τόσο για την εύρεση αντιστοιχίσεων, όσο και για το μετασχηματισμό των επερωτήσεων.

Ο peer που δημιουργεί και γίνεται μέλος μιας κοινότητας εκτός από τις προαναφερθείσες λειτουργίες εκδίδει και μια δημοσιοποίηση που χαρακτηρίζει την κοινότητα που δημιούργησε και η οποία γνωστοποιεί στους υπόλοιπους peers την ύπαρξη της κοινότητας.

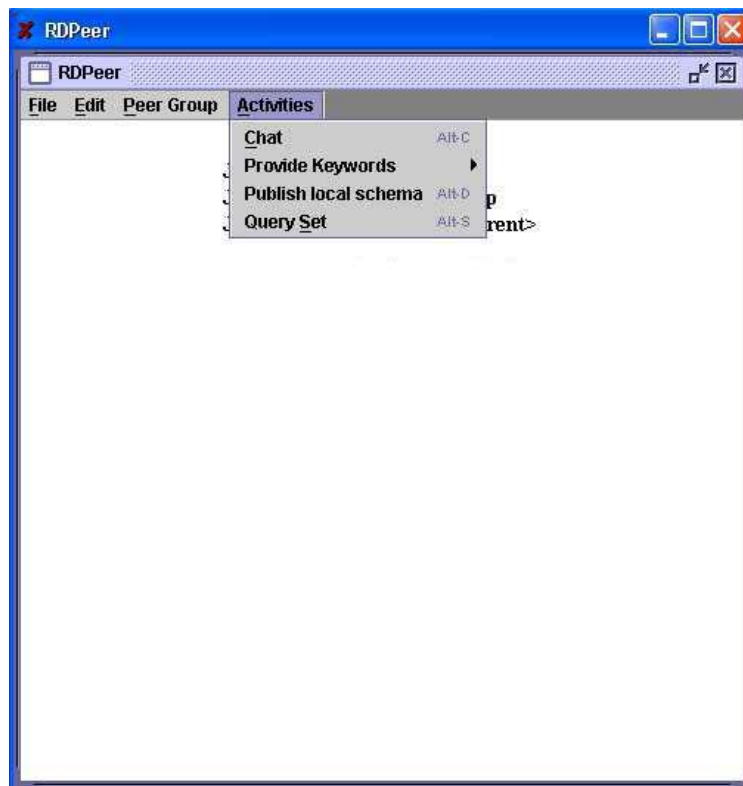
Για να γίνει ένας peer μέλος μιας κοινότητας πρέπει υποχρεωτικά να αναζητήσει και να βρει τις δυο δημοσιοποιήσεις που έχει εκδώσει ο super-peer. Ένας peer όταν γίνεται μέλος μιας κοινότητας εν' αντιθέσει με τον super-peer εκτελεί αυτόματα μόνο μια λειτουργία. Αυτή είναι να θέσει σε λειτουργία μόνο έναν πολυνηματικό εξυπηρετητή του οποίου η λειτουργικότητα αναφέρεται στην εύρεση των ενεργών μελών της κοινότητας

και εξυπηρετεί στη διενέργεια των εκλογών στην περίπτωση που πάψει να είναι ενεργός ο super-peer. Τέλος, κοινοποιεί την ύπαρξή του στα υπόλοιπα μέλη της κοινότητας.

#### 4.1.4 Φάση διαμοιρασμού των δεδομένων

Στη φάση του διαμοιρασμού των δεδομένων κάθε peer πρέπει να είναι μέλος μιας κοινότητας. Στη διάρκεια αυτής της φάσης ένας peer εμπεριέχει τη λειτουργικότητα που περιγράφηκε στις δύο προηγούμενες ενότητες. Ένας peer που είναι μέλος μιας κοινότητας ανεξαρτήτως αν είναι απλός ή super-peer έχει στη διάθεσή του, όπως φαίνεται και στην παρακάτω εικόνα, τις ακόλουθες επιλογές:

1. Την παροχή λέξεων κλειδιών για το σχήμα της βάσης δεδομένων που διαθέτει.
2. Την έκδοση μιας δημοσίευσης με το σχήμα της βάσης δεδομένων και του θησαυρού όρων που δημιουργεί για αυτό.
3. Τη δυνατότητα να θέτει SQL ερωτήσεις.



Εικόνα 8 Οι επιλογές για παροχή λέξεων κλειδιών, κοινοποίησης του σχήματος της Βάσης Δεδομένων και δημιουργίας SQL ερωτήσεων.

Η φάση του διαμοιρασμού των δεδομένων μπορεί να χωριστεί σε τρεις υποενότητες, οι οποίες είναι οι επιλογές που έχει ο χρήστης. Η πρώτη υποενότητα δεν είναι υποχρεωτική αλλά έχει βοηθητικό ρόλο για το σύστημα. Ο διαμοιρασμός των δεδομένων γίνεται πραγματικότητα μόνο όταν κάποιος peer εκδώσει μια SQL επερώτηση. Για να γίνει όμως αυτό θα πρέπει κάθε peer που θέλει να συμβάλει με τα δεδομένα που διαθέτει στη βάση δεδομένων οπωσδήποτε να εκδώσει μια δημοσιοποίηση που θα περιέχει τα στοιχεία του σχήματός του. Αν κάποιος peer δεν κοινοποιήσει το σχήμα του τότε σ' αυτή τη φάση θα συμπεριφέρεται μόνο ως πελάτης, αφού ποτέ δεν πρόκειται να του ζητηθεί να απαντήσει μια SQL επερώτηση, αλλά θα έχει μόνο τη δυνατότητα να θέσει επερωτήσεις. Κάθε υποενότητα είναι αυτόνομη λειτουργικά. Αυτό σημαίνει ότι ένας peer για παράδειγμα μπορεί να μην παρέχει λέξεις κλειδιά, να εκδώσει το σχήμα της βάσης και να μη θέσει ποτέ κάποια SQL επερώτηση. Στις δύο πρώτες υποενότητες η λειτουργικότητα που έχει τόσο ο απλός peer όσο και ο super-peer είναι η ίδια. Εκεί που έχουμε σαφή διαφοροποίηση είναι στη συμπεριφορά και λειτουργικότητα που εμφανίζει κάθε peer στην τρίτη υποενότητα όταν δηλαδή τεθεί μια SQL επερώτηση. Παρακάτω, αναφέρεται κάθε υποενότητα και η λειτουργικότητα που περικλείεται σε αυτή.

#### 4.1.4.1 Παροχή λέξεων κλειδιών

Το RDPeer συνδέεται στη βάση δεδομένων που διαθέτει ο peer ανεξαρτήτως αν είναι απλός peer ή super-peer και εξάγει το σχήμα της. Με τη λειτουργία αυτή πετυχαίνουμε να γνωρίζει η εφαρμογή και ο χρήστης τα στοιχεία από τα οποία αποτελείται το σχήμα της βάσης. Από τα στοιχεία που εξάγονται αυτά που διατηρούνται είναι:

- Τα ονόματα των σχέσεων και των πεδίων.
- Η περιοχή τιμών κάθε πεδίου.
- Τα πρωτεύον και εξωτερικά κλειδιά κάθε σχέσης.

Κάθε σχέση που δημιουργήθηκε από το χρήστη εμφανίζεται δυναμικά (ονόματα των σχέσεων και των πεδίων), ώστε αν το επιθυμεί να παρέχει λέξεις κλειδιά όπως φαίνεται στην παρακάτω εικόνα. Με τον τρόπο αυτό, ο χρήστης μπορεί να παρέχει μεταδεδομένα, τα οποία χαρακτηρίζουν τα στοιχεία κάθε σχέσης και τα οποία διατηρεί το σύστημα. Τα μεταδεδομένα αποτελούν ένα μικρό μέρος του θησαυρού όρων που διαθέτει κάθε peer για τον προσδιορισμό των δεδομένων που διαχειρίζεται. Η παροχή λέξεων κλειδιών είναι χρήσιμη για στοιχεία τα οποία δεν είναι αυτονόητη η σημασία τους. Για παράδειγμα σε μια σχέση της βάσης μπορεί ο δημιουργός να έχει επιλέξει ως όνομα πεδίου τη λέξη «TEL» εννοώντας το τηλέφωνο. Αν στη συγκεκριμένη αυτή περίπτωση δε δοθεί μια λέξη κλειδί τότε θα είναι πολύ δύσκολη η εύρεση εννοιολογικής ταύτισης με πεδία, που περιγράφονται στην SQL επερώτηση. Για κάθε ένα στοιχείο της βάσης δεδομένων



ζητείται η παροχή στην εφαρμογή το πολύ δύο λέξεων κλειδιών. Δε θεωρήσαμε χρήσιμο να προτρέπουμε το χρήστη για μεγαλύτερο αριθμό λέξεων κλειδιών, καθώς θα μπορούσαμε να τον οδηγήσουμε στην παροχή πολύ γενικών όρων. Με την τακτική που ακολουθείται οδηγείται ο χρήστης στην όσο το δυνατόν πιο εύστοχη περιγραφή κάθε όρου της σχέσης.

Το στάδιο αυτό είναι το μοναδικό στο οποίο έχουμε ανάμειξη του χρήστη στη διαδικασία διαμοιρασμού των δεδομένων σε peer-to-peer περιβάλλον. Η δημιουργία του πλήρη θησαυρού όρων είναι μέρος της διαδικασίας που λαμβάνει χώρα κατά τη δημοσιοποίηση του σχήματος.

The screenshot shows a window titled "RDPeer" with a menu bar containing "File", "Edit", "Peer Group", and "Activities". The main area contains the following text and form:

**Give two keywords  
for each name**

<table> ORGANIZATION	<input type="text"/>	<input type="text"/>
<attr> NAME	<input type="text"/>	<input type="text"/>
<attr> ADDRESS	<input type="text"/>	<input type="text"/>
<attr> CITY	<input type="text"/>	<input type="text"/>
<attr> DESCRIPTION	<input type="text"/>	<input type="text"/>
<attr> NUM_WORKERS	<input type="text"/>	<input type="text"/>
<input type="button" value="INSERT"/>		

Εικόνα 9 Η φόρμα στην οποία μπορεί να δώσει ο χρήστης λέξεις κλειδιά για κάθε όνομα σχέσης και πεδίου.

#### 4.1.4.2 Έκδοση σχήματος βάσης δεδομένων

Η έκδοση του σχήματος της βάσης δεδομένων που κατέχει ένας κόμβος του δικτύου είναι μια διαδικασία που είναι προαιρετική, αλλά πολύ σημαντική για να επιτευχθεί ο

διαμοιρασμός των δεδομένων μέσα στο peer-to-peer σύστημα. Η έκδοση του σχήματος της βάσης δεδομένων είναι η διαδικασία με την οποία ένας peer κοινοποιεί τις σχέσεις που περιέχει μαζί με ένα θησαυρό όρων που έχει δημιουργήσει. Η κοινοποίηση αυτή έχει ως τελικό παραλήπτη τον super-peer, ο οποίος συγκεντρώνοντας τα σχήματα από κάθε peer θα είναι σε θέση, όταν τεθεί μια SQL επερώτηση να ανακαλύψει τους peers που μπορούν να απαντήσουν. Η δημοσιοποίηση του σχήματος μπορεί να χωριστεί σε δύο μεγάλες ενότητες, λειτουργίες. Η μεν πρώτη είναι η δημιουργία του θησαυρού όρων ενώ η δεύτερη περιλαμβάνει τη δημιουργία του XML εγγράφου που περιγράφει το σχήμα της βάσης δεδομένων και την κοινοποίηση του στην κοινότητα.

#### 4.1.4.2.1 Δημιουργία θησαυρού όρων

Η δημιουργία του θησαυρού όρων είναι μια εσωτερική διαδικασία αδιαφανής στον χρήστη. Η φάση αυτή ξεκινάει με την εξαγωγή του σχήματος. Από την εξαγωγή του σχήματος τα στοιχεία που διατηρούνται είναι τα ονόματα των σχέσεων και τα πεδία που διαθέτει κάθε μία από αυτές, η περιοχή τιμών κάθε πεδίου και τέλος τα πρωτεύον και εξωτερικά κλειδιά κάθε σχέσης.

Αφού παρθούν τα στοιχεία του σχήματος ο RDPeer συνδέεται μέσω *urlconnection* σε δύο *online* ηλεκτρονικά λεξικά. Τα δύο ηλεκτρονικά λεξικά στα οποία συνδέεται είναι:

- <http://www.rhymezone.com/r/rhyme.cgi>
- <http://www.cogsci.princeton.edu/cgi-bin/webwn>

Η ενέργεια αυτή γίνεται για να παρθούν τα συνώνυμα, υπερώνυμα των ονομάτων των σχέσεων, πεδίων και των λέξεων κλειδιών. Ο RDPeer κάνει για κάθε όνομα σχέσης ή πεδίου δύο συνδέσεις μια σε κάθε λεξικό. Αφού καλέσει κάθε λεξικό με παράμετρο το όνομα που τον ενδιαφέρει, κάθε λεξικό θα επιστρέψει μια ιστοσελίδα στην εφαρμογή. Η ιστοσελίδα θα περιέχει από το μεν πρώτο λεξικό τα υπερώνυμα του ονόματος που ψάχναμε, ενώ η δεύτερη θα περιέχει τα συνώνυμα. Ο RDPeer είναι εφοδιασμένος με έναν αναλυτή της HTML γλώσσας (HTMLParser), ώστε να μπορούμε να γνωρίζουμε τι είδους τμήμα της ιστοσελίδας διαβάζουμε κάθε φορά και ανάλογα να επεμβαίνουμε όπου χρειάζεται. Με την τεχνική αυτή καταφέρνουμε να απομονώσουμε από τις ιστοσελίδες μόνο τις λέξεις που μας ενδιαφέρουν.

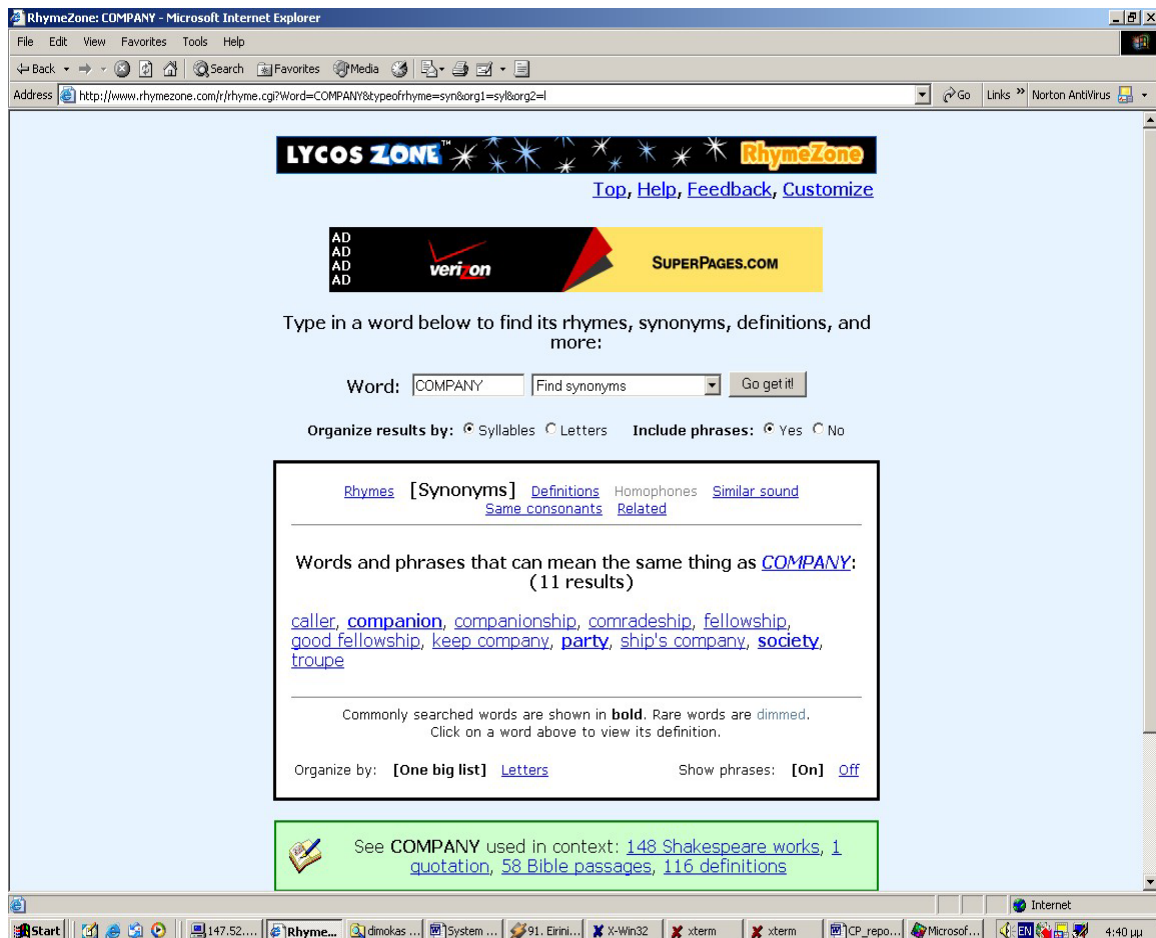
Ο RDPeer αποθηκεύει, τόσο τα συνώνυμα που βρίσκει όσο και τα υπερώνυμα, σε δύο αρχεία ένα για κάθε κατηγορία, μέσα σε φάκελο που διαθέτει για το σκοπό αυτό κάθε peer. Η σημασία της αποθήκευσης είναι σημαντική γιατί με τον τρόπο αυτό η εφαρμογή σε όλη τη διάρκεια της ζωής της μόνο μια φορά θα χρειαστεί να κάνει την αναζήτηση και εύρεση των συνωνύμων και υπερωνύμων. Από τη στιγμή που θα ολοκληρωθεί μια φορά η παραπάνω χρονοβόρος διαδικασία ο peer θα έχει τα δεδομένα

πάντα στη διάθεσή του. Συνεπώς, κάθε κόμβος του δικτύου κερδίζει σε απόδοση και ταχύτητα, ενώ παράλληλα ο αποθηκευτικός χώρος που απαιτείται είναι πολύ μικρός.

Αν για παράδειγμα μια σχέση περιέχει ένα πεδίο με όνομα COMPANY, τότε η ιστοσελίδα με τα συνώνυμα που θα λάβει η εφαρμογή μας φαίνεται στην παρακάτω εικόνα. Η εφαρμογή μας προμηθεύει το online λεξικό μέσω URL τη λέξη που αναζητούμε και αυτό διενεργεί την αναζήτηση για λογαριασμό μας.

Αφού ολοκληρωθεί η διαδικασία για τα ονόματα των σχέσεων, πεδίων στη συνέχεια ακολουθεί η εύρεση των συνωνύμων, υπερωνύμων για τις λέξεις κλειδιά που είχε προμηθεύσει την εφαρμογή μας ο χρήστης στο προηγούμενο στάδιο (Παροχή λέξεων κλειδιών).

Με τον τρόπο αυτό πετυχαίνουμε να δημιουργήσουμε έναν θησαυρό όρων που προσδιορίζει τα δεδομένα που είναι αποθηκευμένα στη βάση δεδομένων ενός κόμβου του δικτύου. Η διεργασία αυτή συντελείται σε κάθε peer.



Εικόνα 10 Το ηλεκτρονικό λεξικό που χρησιμοποιεί το RDPeer για την εύρεση συνωνύμων.

#### 4.1.4.2.2 Δημιουργία XML εγγράφου

Η δημιουργία του XML εγγράφου είναι η εσωτερική διαδικασία που συντελείται έπειτα από τη δημιουργία του θησαυρού όρων. Η δημιουργία του XML εγγράφου έχει ως σκοπό να περιλάβει το σχήμα της βάσης και το θησαυρό όρων, και στη συνέχεια να κοινοποιηθεί το έγγραφο στον super-peer. Με τη χρήση της XML καταφέρνουμε να περιγράψουμε επαρκώς τα δεδομένα που περιέχει το σχήμα της βάσης. Η δομή που επιλέχτηκε για το XML έγγραφο αναλύεται στην παράγραφο της αρχιτεκτονικής του peer. Η ίδια προκαθορισμένη δομή ακολουθείται από όλους τους peers για λόγους συμβατότητας και ευκολίας στην ανάλυση του XML εγγράφου από τον super-peer. Η χρήση της XML επιλέχτηκε, γιατί όπως είδαμε στο προηγούμενο κεφάλαιο όλα τα θέματα της πλατφόρμας JXTA βασίστηκαν στην XML για να δομηθούν τα δεδομένα σαν δημοσιοποιήσεις, μηνύματα και πρωτόκολλα.

Εάν θεωρήσουμε ότι έχουμε ένα peer που ονομάζεται peer2-rdv και έχει μία σχέση μόνο στη βάση δεδομένων, που είναι η ORGANIZATION (NAME,ADDRESS) με πρωτεύον κλειδί το πεδίο NAME, και ότι ο χρήστης δεν έχει δώσει καμία λέξη κλειδί τότε το XML έγγραφο που θα δημιουργηθεί και θα κοινοποιηθεί θα έχει την παρακάτω μορφή.

```
<?xml version="1.0"?>
<SCHEMA AUTHOR="peer2-rdv">
  <TABLE> ORGANIZATION
  <SYN> ARRANGEMENT </SYN>
  <SYN> BRASS </SYN>
  <SYN> CONSTITUTION </SYN>
  <SYN> ESTABLISHMENT </SYN>
  <SYN> GOVERNANCE </SYN>
  <HYPER> SOCIAL GROUP </HYPER>
  <HYPER> GROUP </HYPER>
  <HYPER> GROUPING </HYPER>
  <KEYWORD0> </KEYWORD0>
  <KEYWORD0_SYN> </KEYWORD0_SYN>
  <KEYWORD0_HYPER> </KEYWORD0_HYPER>
  <KEYWORD1> </KEYWORD1>
  <KEYWORD1_SYN> </KEYWORD1_SYN>
  <KEYWORD1_HYPER> </KEYWORD1_HYPER>
  <ITEM> NAME
  <ITEMTYPE> VARCHAR2(40) NOT NULL </ITEMTYPE>
  <SYN> ADVERT </SYN>
  <SYN> APPOINT </SYN>
```

```

<SYN> BRING UP </SYN>
<SYN> CALL </SYN>
<SYN> CITE </SYN>
<HYPER> LANGUAGE UNIT </HYPER>
<HYPER> LINGUISTIC UNIT </HYPER>
<HYPER> PART </HYPER>
<HYPER> PORTION </HYPER>
<HYPER> COMPONENT PART </HYPER>
<KEYWORD0> </KEYWORD0>
<KEYWORD0_SYN> </KEYWORD0_SYN>
<KEYWORD0_HYPER> </KEYWORD0_HYPER>
<KEYWORD1> </KEYWORD1>
<KEYWORD1_SYN> </KEYWORD1_SYN>
<KEYWORD1_HYPER> </KEYWORD1_HYPER>
</ITEM>
<ITEM> ADDRESS
<ITEMTYPE> VARCHAR2(50) </ITEMTYPE>
<SYN> ACCOST </SYN>
<SYN> COME UP TO </SYN>
<SYN> COMPUTER ADDRESS </SYN>
<SYN> COVER </SYN>
<SYN> DEAL </SYN>
<HYPER> CODE </HYPER>
<HYPER> COMPUTER CODE </HYPER>
<HYPER> CODING SYSTEM </HYPER>
<HYPER> WRITING </HYPER>
<HYPER> WRITTEN COMMUNICATION </HYPER>
<KEYWORD0> </KEYWORD0>
<KEYWORD0_SYN> </KEYWORD0_SYN>
<KEYWORD0_HYPER> </KEYWORD0_HYPER>
<KEYWORD1> </KEYWORD1>
<KEYWORD1_SYN> </KEYWORD1_SYN>
<KEYWORD1_HYPER> </KEYWORD1_HYPER>
</ITEM>
<PRKEY> NAME </PRKEY>
</TABLE>
</SCHEMA>

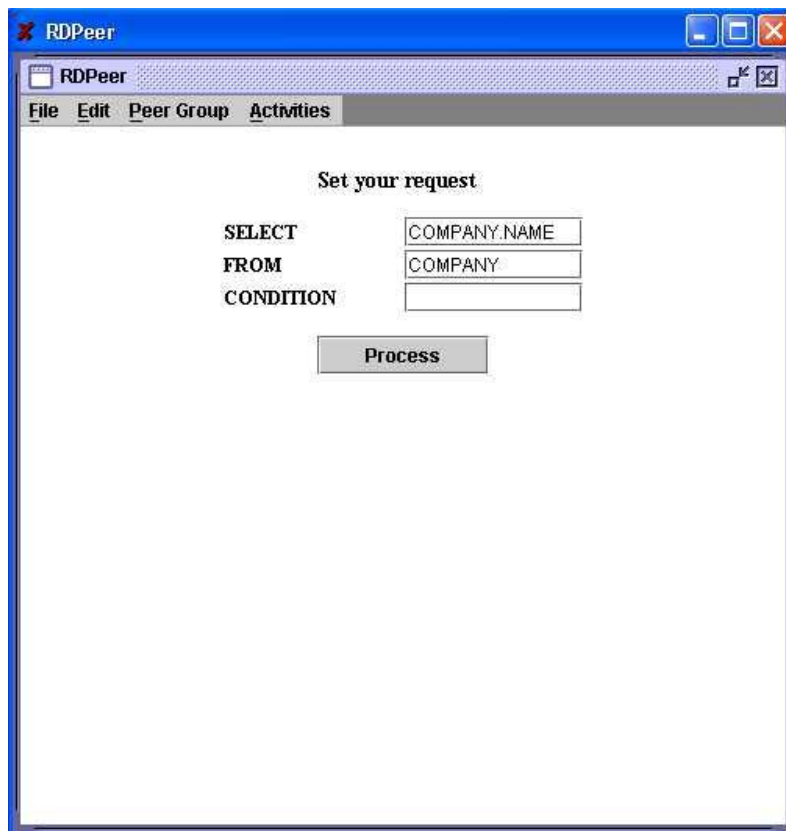
```

Όπως παρατηρούμε αφού δηλωθεί το όνομα του peer, στον οποίο ανήκει το σχήμα της βάσης, στη συνέχεια έχουμε τις δηλώσεις των σχέσεων. Για κάθε όνομα σχέσης ή πεδίου έχουμε κάποια χαρακτηριστικά γνωρίσματα του XML εγγράφου, τα οποία περιέχουν το θησαυρό όρων που αντιστοιχεί σε κάθε όνομα. Επιπλέον, στην περίπτωση του πεδίου δηλώνεται ο τύπος του, ενώ στο τέλος κάθε σχέσης δηλώνονται τα πρωτεύον και εξωτερικά κλειδιά. Επομένως, με τον τρόπο αυτό οι κόμβοι του δικτύου μπορούν να μεταφέρουν και να περιγράψουν την πληροφορία που βρίσκεται τοπικά αποθηκευμένη.

Η δημοσιοποίηση του σχήματος φέρει όπως και όλες οι δημοσιοποιήσεις του JXTA ένα χρονικό όριο ζωής. Όταν παρέλθει αυτό κάθε peer θα πρέπει να κοινοποιήσει ξανά το σχήμα.

#### 4.1.4.3 Υποβολή SQL επερώτησης

Κάθε peer που ανήκει σε μια κοινότητα είναι ικανός να θέσει μια SQL επερώτηση. Όπως φαίνεται στην παρακάτω εικόνα ζητείται από το χρήστη να δημιουργήσει μία έγκυρη SQL επερώτηση.



The image shows a screenshot of the RDPeer application window. The window title is "RDPeer". The menu bar includes "File", "Edit", "Peer Group", and "Activities". The main content area is titled "Set your request" and contains a form for entering an SQL query. The form has three rows: "SELECT" with a text box containing "COMPANY.NAME", "FROM" with a text box containing "COMPANY", and "CONDITION" with an empty text box. Below the form is a "Process" button.

Εικόνα 11 Δημιουργία SQL επερώτησης.

Στην περίπτωση που ο peer δεν είναι super-peer, τότε αφού λάβει από την επιφάνεια διεπαφής την SQL επερώτηση επικοινωνεί με τον super-peer και του το μεταφέρει. Όπως είχαμε πει σε προηγούμενη ενότητα ο super-peer έχει έναν πολυνηματικό εξυπηρετητή, που λαμβάνει SQL επερωτήσεις από τους peers και προσπαθεί να τις εξυπηρετήσει. Ο super-peer από τη στιγμή που θα λάβει την επερώτηση η πρώτη λειτουργία που θα

επιτελέσει είναι να συλλέξει τα σχήματα των βάσεων δεδομένων. Η συλλογή των σχημάτων δε πραγματοποιείται μόνο στην περίπτωση που ένας peer είναι ενεργός και το σχήμα του είναι ήδη αποθηκευμένο. Τα σχήματα που συλλέγονται, αποθηκεύονται σε μια τοπική cache που διατηρεί για το σκοπό αυτό ο super-peer και παραμένουν εκεί σύμφωνα με το χρονικό όριο ζωής που φέρει η δημοσίευση του σχήματος. Όταν παρέλθει το χρονικό όριο ζωής της δημοσίευσης παύει να είναι έγκυρη και διαγράφεται. Επομένως, αρχικά ο super-peer βρίσκει μέσω ανταλλαγής ring pong μηνυμάτων τους ενεργούς peers και εν συνεχεία παίρνει τα σχήματα που έχουν. Η διαδικασία αυτή μπορεί να επιβραδύνει λίγο την απόδοση αλλά προσφέρει πιο αξιόπιστα αποτελέσματα.

Η δεύτερη λειτουργία που θα επιτελέσει είναι να δημιουργήσει ένα θησαυρό όρων για την SQL επερώτηση. Ο RDPeer χρησιμοποιεί έναν αναλυτή της SQL γλώσσας (SQLParser), ώστε να μπορούμε να αναλύσουμε την επερώτηση και να βρούμε τα ονόματα των πεδίων και σχέσεων. Επιπλέον εξάγουμε τις εκφράσεις που υπάρχουν τόσο στον όρο SELECT, όσο και στον όρο WHERE. Ο super-peer εκτελεί τη διαδικασία που περιγράφηκε στην παράγραφο της δημιουργίας θησαυρού όρων. Αφού ανακαλύψει τα συνώνυμα, υπερώνυμα των ονομάτων των σχέσεων και πεδίων, τα αποθηκεύει ώστε σε επόμενη SQL επερώτηση που διαθέτει κάποια ίδια ονόματα να μη χρειαστεί να ξανακάνει ολόκληρη τη διαδικασία, αλλά να πάρει το θησαυρό όρων από την cache που διατηρεί.

Έπειτα, ο super-peer αναζητεί ομοιότητες μεταξύ των στοιχείων της επερώτησης και των σχημάτων των βάσεων δεδομένων που έχει συλλέξει. Η εύρεση ομοιοτήτων ξεκινάει από τα ονόματα σχέσεων. Στην περίπτωση που δε βρεθεί κάποια ομοιότητα μεταξύ του ονόματος της σχέσης της επερώτησης με κάποια σχέση που αναφέρει το σχήμα ενός peer, τότε απευθείας ο συγκεκριμένος peer κρίνεται ακατάλληλος να απαντήσει την επερώτηση και η διαδικασία εξετάζει το σχήμα του επόμενου peer. Γενικότερα, για να κριθεί ένας peer ικανός να απαντήσει μια SQL επερώτηση θα πρέπει να έχει οπωσδήποτε στο σχήμα του μια σχέση εννοιολογικά όμοια με μια από τις σχέσεις που περιγράφονται στο όρο FROM. Επιπλέον, πρέπει να υπάρχει οπωσδήποτε τουλάχιστον μια ομοιότητα μεταξύ ενός πεδίου της σχέσης, που βρέθηκε ότι γειτνιάζει εννοιολογικά με μια σχέση της SQL επερώτησης, με ένα πεδίο που περιγράφεται στο όρο SELECT.

Ο super-peer από τη στιγμή που θα βρει τους peers που μπορούν να απαντήσουν έστω και εν' μέρει την επερώτηση διενεργεί το μετασχηματισμό, ώστε να μπορεί να απαντηθεί από αυτούς. Αυτό σημαίνει ότι γράφει ξανά την SQL επερώτηση σε διαφορετικές εκδόσεις μια για κάθε peer. Η κάθε έκδοση περιλαμβάνει ονόματα σχέσεων και πεδίων, τα οποία αντιστοιχούν σε αυτά που περιγράφονται στο σχήμα κάθε peer.

Έστω για παράδειγμα ότι έχουμε τρεις peers, με ονόματα peer1, peer2, peer2-rdn, οι οποίοι συμμετέχουν σε μια κοινότητα της οποίας super-peer είναι ο peer2-rdn. Θεωρούμε ότι ο peer1 έχει εκδώσει το σχήμα της βάσης δεδομένων που έχει, χωρίς να έχει προσθέσει λέξεις κλειδιά. Το σχήμα είναι το εξής: COMPANY(COMPANYID, NAME,

ADDRESS, CITY). Το ίδιο θεωρούμε ότι έχει κάνει και ο peer2-rdn του οποίου το σχήμα είναι: ORGANIZATION(NAME, TOWN, ADDRESS, NUM\_WORKERS). Αντίθετα με τους δύο παραπάνω peers ο peer2 δεν έχει εκδώσει το σχήμα του αλλά θέτει την ακόλουθη SQL επερώτηση:

```
SELECT COMPANY.NAME, COMPANY.NUM_WORKERS
FROM COMPANY;
```

Η επερώτηση στέλνεται από τον peer2 στον peer2-rdn. Ο peer2-rdn αρχικά συλλέγει τα σχήματα των peers που έχουν κοινοποιηθεί, δηλαδή το δικό του και του peer1. Αφού συγκρίνει τους θησαυρούς όρων των σχημάτων με τον θησαυρό όρων που δημιουργεί για την SQL επερώτηση, διαπιστώνει ότι η σχέση COMPANY της επερώτησης συμπίπτει με τη σχέση που υπάρχει στο σχήμα του peer1 και τη σχέση που υπάρχει στο σχήμα του peer2-rdn. Η ομοιότητα με τη σχέση του peer2-rdn προκύπτει από το γεγονός ότι η λέξη COMPANY έχει κοινά συνώνυμα με τη λέξη ORGANIZATION. Στη συνέχεια, ο super-peer ψάχνει αν τα πεδία NAME και NUM\_WORKERS υπάρχουν στις δύο σχέσεις. Για τη σχέση ORGANIZATION του peer2-rdn αυτό ικανοποιείται, αλλά η σχέση του peer1 περιέχει μόνο το ένα από τα δύο πεδία. Αποτέλεσμα του μετασχηματισμού που συμβαίνει στον super-peer είναι να παραχθούν δύο SQL επερωτήσεις, οι οποίες είναι:

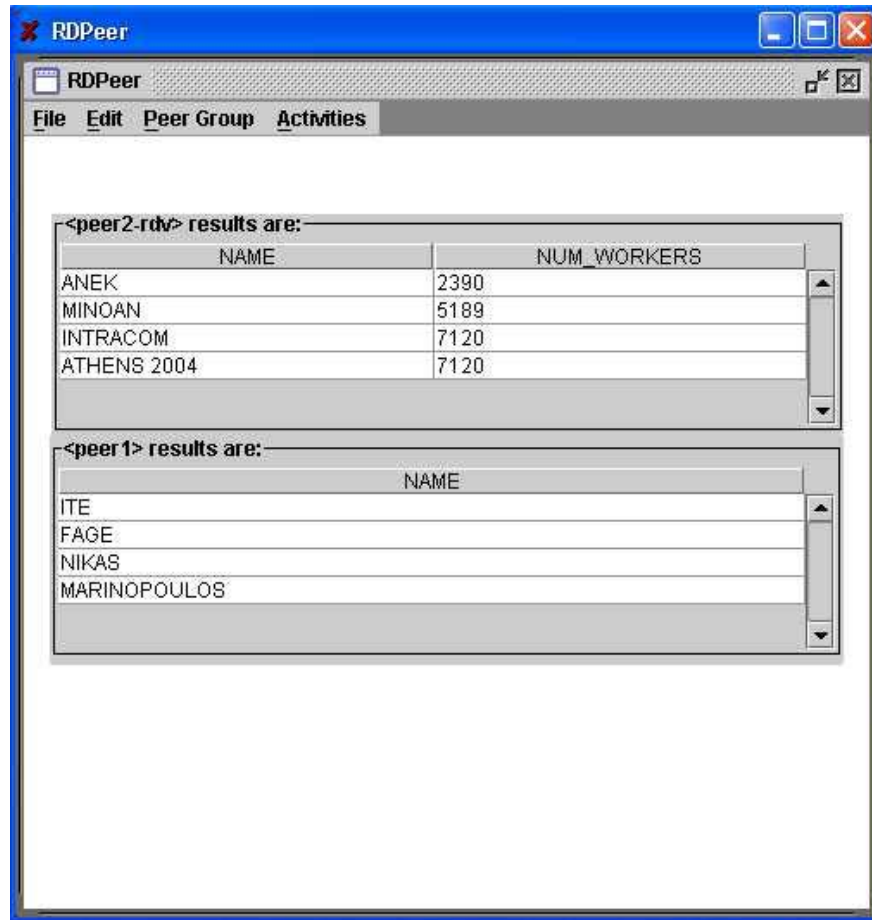
1. SELECT COMPANY.NAME  
FROM COMPANY;
2. SELECT ORGANIZATION.NAME, ORGANIZATION.NUM\_WORKERS FROM  
ORGANIZATION;

Η πρώτη επερώτηση αντιστοιχεί στον peer1, ενώ η δεύτερη στον peer2-rdn. Από το μετασχηματισμό συμπεραίνει κανείς ότι μπορεί η αρχική SQL επερώτηση να διαφέρει σημαντικά από τις μετασχηματισμένες. Παρά τις μεγάλες αποκλίσεις που μπορεί να υπάρχουν θεωρούμε κάθε μετασχηματισμένη SQL επερώτηση έγκυρη αν αποτελεί υποσύνολο της αρχικής. Στην περίπτωση που είναι υπερσύνολο απορρίπτεται. Ο ακριβής μηχανισμός για το μετασχηματισμό μιας SQL επερώτησης περιγράφεται λεπτομερώς σε επόμενη ενότητα.

Τα αποτελέσματα του μετασχηματισμού επιστρέφονται στο peer που έθεσε την επερώτηση. Επομένως, ο super-peer μεταβιβάζει στον peer τις μετασχηματισμένες επερωτήσεις και για κάθε μία σε ποιο peer αντιστοιχεί. Ο peer από το στάδιο της αρχικοποίησης γνωρίζει πως να κάνει επίκληση απομακρυσμένης μεθόδου, καθώς έχει συλλέξει τις δημοσιεύσεις όλων των peers που σχετίζονται με το web service που έχουν. Συνεπώς, καλώντας τις υπηρεσίες κάθε peer με παράμετρο την SQL επερώτηση που αντιστοιχεί στον εκάστοτε peer, καταφέρνει να συλλέξει τα δεδομένα που περιέχονται



στις ετερογενείς βάσεις δεδομένων και τα εμφανίζει στην επιφάνεια διεπαφής. Το τελικό αποτέλεσμα που προκύπτει από όλη την παραπάνω διαδικασία για το συγκεκριμένο παράδειγμα φαίνεται στην εικόνα που ακολουθεί. Πρώτα εμφανίζονται τα αποτελέσματα του peer για τον οποίο η μετασχηματισμένη SQL επερώτηση δε διαφέρει παρά μόνο γλωσσολογικά από την αρχική και έπειτα τα αποτελέσματα της επερώτησης που είναι υποσύνολο της αρχικής.



Εικόνα 12 Τα αποτελέσματα που εμφανίζονται σε έναν peer που έχει θέσει μια SQL επερώτηση.

Η υποβολή μιας SQL επερώτησης είναι και το στάδιο κατά το οποίο μπορούν να διεξαχθούν εκλογές μέσα στην κοινότητα των peers. Αυτό συμβαίνει, όταν ένας peer προσπαθεί να συνδεθεί στον super-peer και διαπιστώσει ότι η σύνδεση είναι ανέφικτη. Από τη στιγμή εκείνη και μετά διενεργούνται εκλογές για την ανάδειξη του νέου super-peer. Η διαδικασία αυτή της εκλογής θα αναλυθεί σε επόμενη ενότητα εκτενώς. Ο νέος super-peer που θα προκύψει είναι ουσιαστικά αυτός που έθεσε τις περισσότερες SQL

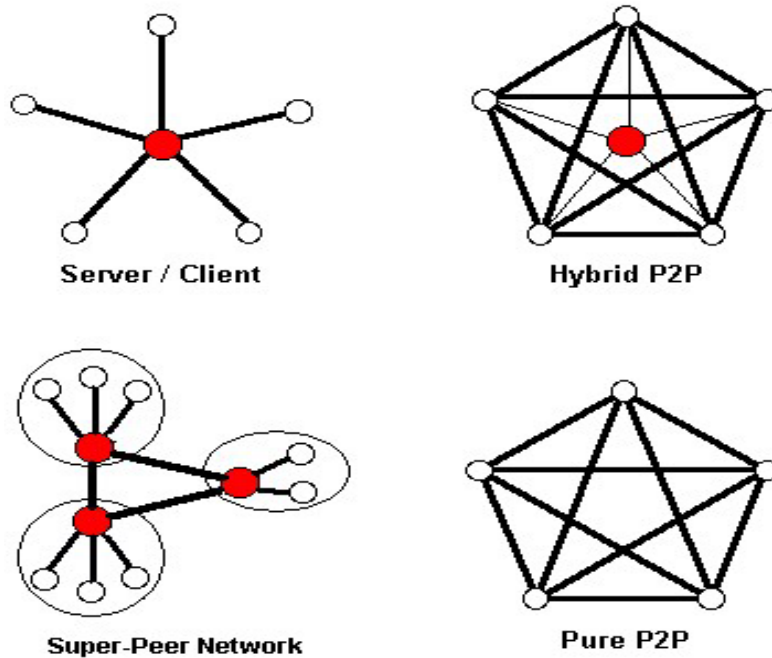
επερωτήσεις προς τον προηγούμενο super-peer. Αυτό συμβαίνει για να μειώσουμε την επιβάρυνση της κίνησης του δικτύου σε αριθμό μηνυμάτων, που θα προκύψει μετά τις εκλογές. Η επίτευξη της μείωσης συνεπάγεται, καθώς ο peer που δημιουργούσε τις περισσότερες επερωτήσεις δε θα χρειάζεται να επικοινωνεί για να θέτει τις επερωτήσεις του, αλλά θα τις επεξεργάζεται τοπικά. Στις επόμενες ενότητες περιγράφονται εκτενώς η αρχιτεκτονική του συστήματος, η αρχιτεκτονική του peer, η τεχνική της αντιστοίχισης σχημάτων που ακολουθήθηκε, ο μετασχηματισμός μιας SQL επερώτησης και η διαδικασία των εκλογών. Επιπλέον περιγράφονται οι τεχνολογίες που χρησιμοποιήθηκαν.

## 4.2 Αρχιτεκτονική του Συστήματος

Τα peer-to-peer συστήματα έχουν γίνει πρόσφατα το πιο δημοφιλές μέσο για το διαμοιρασμό μεγάλου πλήθους δεδομένων. Κύριο χαρακτηριστικό τους είναι η κατανομή του κύριου κόστους του διαμοιρασμού των αρχείων, που είναι ο χώρος στον δίσκο για την αποθήκευση των αρχείων και το bandwidth που απαιτείται για τη μεταφορά τους, στους peers που αποτελούν το δίκτυο και δίνεται έτσι η δυνατότητα να κλιμακώνονται χωρίς την ανάγκη για ακριβούς, δυνατούς servers. Επιπρόσθετα τα peer-to-peer συστήματα περιλαμβάνουν την αυτό-οργάνωση, την ισορροπία του φόρτου εργασίας, την προσαρμοστικότητα και την ανοχή στα λάθη. Αντίθετα με τα peer-to-peer συστήματα ο παγκόσμιος ιστός είναι κατασκευασμένο με βάση το μοντέλο server/client όπως φαίνεται στο παρακάτω σχήμα. Το μοντέλο αυτό αποτελείται από πολλούς πελάτες οι οποίοι προσπαθούν να εξυπηρετηθούν από μικρό σύνολο από εξυπηρετητές. Η κεντροποιημένη αυτή αρχιτεκτονική έχει σαν αποτέλεσμα τη δημιουργία μεγάλου φόρτου εργασίας στους servers και τον περιορισμό των πόρων του δικτύου με αποτέλεσμα να κρίνεται αναποτελεσματικός ο τρόπος διαχείρισης των δεδομένων, αν και ο τρόπος αναζήτησης, επειδή δεν είναι κατανεμημένος, διενεργείται με μεγαλύτερη ταχύτητα.

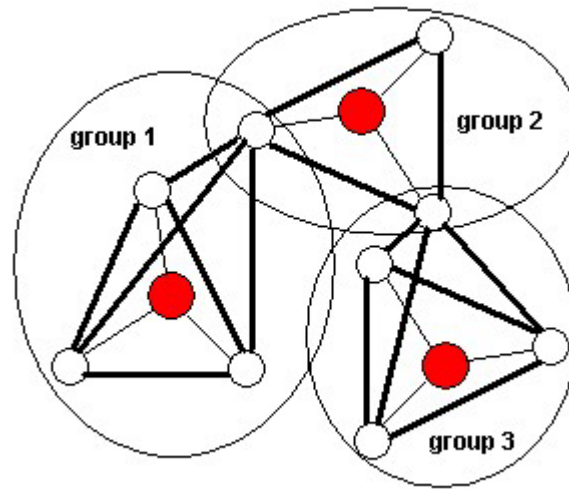
Υπάρχουν διάφοροι τύποι από peer-to-peer συστήματα όπως φαίνεται στο παρακάτω σχήμα και οι οποίοι χωρίζονται σε τρεις μεγάλες κατηγορίες. Στα «καθαρά» *pure* συστήματα όπως είναι το Gnutella, και το Freenet, όλοι οι peers έχουν ίσους ρόλους και ευθύνες σε όλα τα θέματα και χαρακτηρίζονται ως μη κεντροποιημένα καθώς όλοι είναι ισότιμοι και συμπεριφέρονται συγχρόνως ως πελάτες και εξυπηρετητές. Στα υβριδικά συστήματα όπως είναι το Napster η αναζήτηση πραγματοποιείται με κεντροποιημένο τρόπο, αλλά το download συμβαίνει με peer-to-peer τρόπο. Αυτή είναι άλλωστε είναι η ουσιώδης διαφορά που υπάρχει με το μοντέλο πελάτη / εξυπηρετητή. Τα *super-peer* δίκτυα όπως είναι το KaZaA παρουσιάζουν μια ενδιάμεση λογική. Ένας

super-peer είναι ένας κόμβος που ενεργεί σαν ένας κεντροποιημένος server για ένα υποσύνολο από peers. Οι super-peers όμως συνδέονται μεταξύ τους όπως κάνουν οι peers ενός pure συστήματος δρομολογώντας με αυτό τον τρόπο τα μηνύματα, υποβάλλοντας αιτήσεις και δεχόμενοι απαντήσεις.



Σχήμα 1 Αρχιτεκτονικές peer-to-peer συστημάτων και η αρχιτεκτονική Client / Server.

Η αρχιτεκτονική που υλοποιήθηκε στο RDPeer εμφανίζεται στο παρακάτω σχήμα και μοιάζει με την υβριδική αρχιτεκτονική. Το δίκτυο χωρίζεται σε συμπλέγματα από peers όπου κάθε ένα έχει ένα super-peer. Ο αριθμός των peers που έχει κάθε σύμπλεγμα δεν είναι προκαθορισμένος. Κάθε peer μπορεί να ανήκει σε περισσότερα από ένα συμπλέγματα ταυτόχρονα, αλλά να είναι ενεργός μόνο σε ένα, το οποίο είναι το τελευταίο στο οποίο έγινε δεκτή η αίτηση συμμετοχής. Ενεργός είναι ένας peer, όταν μπορεί να θέτει σε ένα super-peer SQL ερωτήσεις. Στα συμπλέγματα στα οποία ανήκει ένας peer χωρίς να χαρακτηρίζεται ως ενεργός τότε συμπεριφέρεται αποκλειστικά ως πηγή δεδομένων και όχι ως καταναλωτής. Η διαφορά της αρχιτεκτονικής με αυτή του super-peer είναι ότι δεν υπάρχει επικοινωνία μεταξύ των super-peers των συμπλεγμάτων, και ότι δεν έχουμε ισορροπημένο φόρτο εργασίας κάθε super-peer. Επίσης, κάθε peer μπορεί να συνδεθεί και να εξάγει δεδομένα από οποιονδήποτε peer ανήκει στο ίδιο σύμπλεγμα.



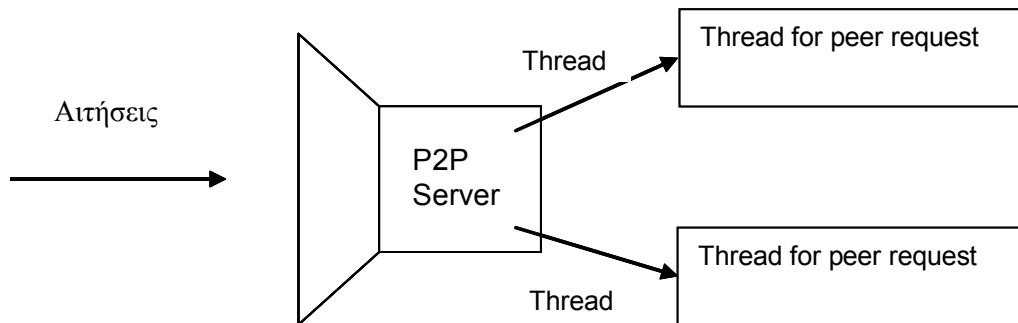
Σχήμα 2 Η αρχιτεκτονική του συστήματος

Η αρχιτεκτονική που υλοποιήθηκε έχει πλεονεκτήματα, αλλά και μειονεκτήματα. Η ύπαρξη σε κάθε σύμπλεγμα ενός super-peer έχει σαν αποτέλεσμα η αναζήτηση των σχημάτων, η εύρεση αντιστοιχίσεων μεταξύ των σχημάτων και ο μετασχηματισμός των SQL ερωτήσεων να γίνεται κεντριοποιημένα γεγονός που είναι θετικό, καθώς η απόδοση του συστήματος βελτιώνεται. Επιπλέον, πλεονέκτημα αποτελεί το γεγονός ότι δεν υπάρχει η επικοινωνία μεταξύ των super-peers, καθώς δεν αποκτούν επιπλέον φόρτο εργασίας. Ένα άλλο πλεονέκτημα που εμφανίζει η αρχιτεκτονική είναι η ανισότητα στους ρόλους και τις ευθύνες που υπάρχει μέσα σε κάθε σύμπλεγμα. Χρησιμοποιώντας τους super-peers σε κάθε σύμπλεγμα δεν εμφανίζεται το μειονέκτημα που υπάρχει στα pure συστήματα και το οποίο είναι η μη αποδοτικότητα εξαιτίας αρκετών peers που είναι περιορισμένων δυνατοτήτων, καθώς μπορεί να συνδέονται με συνδέσμους χαμηλής χωρητικότητας στο δίκτυο. Η ύπαρξη όμως των peers περιορισμένων δυνατοτήτων στα pure συστήματα επιφέρει και ένα άλλο μειονέκτημα που είναι η τμηματοποίηση του δικτύου κατά την αποχώρησή τους, γεγονός που δεν εμφανίζεται στην αρχιτεκτονική που επιλέξαμε. Θετικό στοιχείο αποτελεί ακόμα το γεγονός ότι όλοι οι peers είναι ισότιμοι στο download και δεν προσμένουν ένα κεντρικό εξυπηρετητή να το πράξει για αυτούς.

Αν και η ύπαρξη μιας κεντριοποιημένης διεργασίας είναι πιο αποδοτική από την κατανομημένη από την άποψη του συνολικού κόστους, ο φόρτος εργασίας που δέχεται ο super-peer είναι πολύ μεγάλος με αποτέλεσμα να υπάρχει ένα τελικό σημείο από πλευράς απόδοσης και κλιμακοσιμότητας.

### 4.3 Αρχιτεκτονική του Peer

Το peer-to-peer σύστημα που έχει υλοποιηθεί αποτελείται από peers, όπου κάθε ένας συνδέεται με μια βάση δεδομένων και μπορεί ταυτόχρονα να είναι πελάτης και εξυπηρετητής. Ο RDPeer είναι ένας πολυνηματικός εξυπηρετητής (multithreaded server) όπως είναι οι περισσότεροι αυτή τη στιγμή στο Διαδίκτυο. Αυτή είναι και η λύση που έχει ακολουθηθεί και στον παγκόσμιο ιστό για να αποφεύγεται να μονοπωλούνται οι servers από ένα και μόνο χρήστη [6].



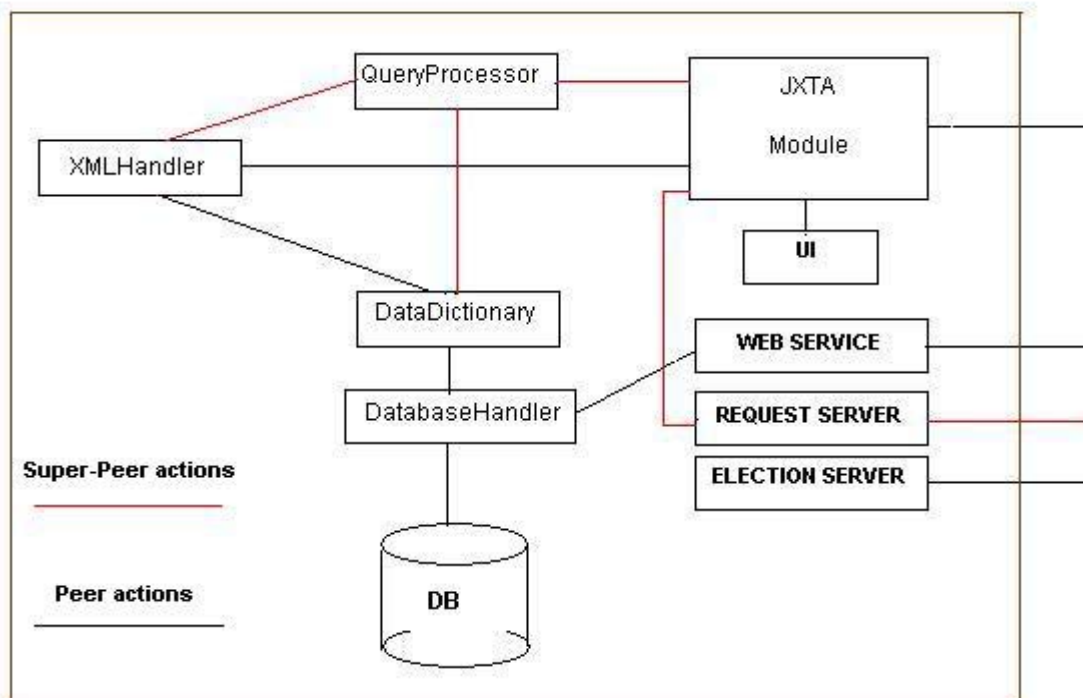
Σχήμα 3 Η αρχιτεκτονική του multithreaded server που διαθέτει κάθε peer.

Για κάθε αίτηση ενός peer ο RDPeer δημιουργεί ένα νήμα για να το εξυπηρετήσει. Ο server ενός peer δημιουργεί τόσα διαφορετικά νήματα όσες και οι αιτήσεις που δέχεται από τους άλλους peers. Στην περίπτωση που ένας peer γίνει super-peer για μια κοινότητα έχει ενεργοποιημένους δύο servers. Ο ένας χρησιμοποιείται για να δέχεται SQL επρωτήσεις, ενώ ο δεύτερος περιλαμβάνει την ανταλλαγή ring pong μηνυμάτων για την ανεύρεση των ενεργών peers της κοινότητας καθώς και την ανταλλαγή μηνυμάτων στην περίπτωση που διενεργούνται εκλογές μεταξύ των peers.

Ο RDPeer εκτός από τον πολυνηματικό εξυπηρετητή περιλαμβάνει και ένα σύνολο από modules, για να είναι ικανός να προσφέρει τη λειτουργικότητα για την οποία προορίζεται. Τα modules από τα οποία αποτελείται κάθε Peer είναι τα ακόλουθα:

- DatabaseHandler
- DataDictionary
- XmlHandler
- QueryProcessor
- JXTA module

- Webservice
- RequestServer
- ElectionServer
- UI



Σχήμα 4 Η αρχιτεκτονική του RDPeer

Στο παραπάνω σχήμα φαίνεται η διάταξη των modules και η επικοινωνία που υπάρχει μεταξύ τους. Με κόκκινο χρώμα τονίζονται οι επιπλέον λειτουργίες που διαθέτει ο super-peer σε σχέση με τον απλό peer. Λόγω της αρχιτεκτονικής του συστήματος είναι πολύ εύκολο να συμπεριληφθούν επιπρόσθετα modules, ώστε να επεκταθεί η λειτουργικότητα ενός peer. Αυτό συμβαίνει γιατί κάθε module είναι σχεδιασμένο και υλοποιημένο να ενεργεί ανεξάρτητα και να επιτελεί μια αυτοτελή διαδικασία. Παρακάτω θα αναλυθεί η λειτουργικότητα που προσφέρει το κάθε ένα module χωριστά, καθώς και ο τρόπος με τον οποίο επικοινωνούν μεταξύ τους οι peers.

### 4.3.1 DatabaseHandler

Η λειτουργία του module είναι να διαχειρίζεται τη βάση δεδομένων, και να εξάγει τα δεδομένα από αυτή. Είναι το μοναδικό τμήμα ενός peer που γνωρίζει την ύπαρξη της βάσης και τον τρόπο που θα επικοινωνήσει με αυτή. Το module DatabaseHandler είναι υπεύθυνο να επικοινωνήσει με τη βάση δεδομένων και να λάβει από αυτή το σχήμα της βάσης. Αφού αναλύσει το σχήμα κρατάει τα ονόματα των σχέσεων, πεδίων, τους τύπους δεδομένων που αυτά έχουν και τέλος τα πρωτεύον, εξωτερικά κλειδιά κάθε σχέσης. Η εμφάνιση του module στη λειτουργία ενός peer γίνεται όταν:

- ο peer θέλει να εκδώσει τη δημοσιοποίηση που περιέχει το σχήμα της βάσης του.
- ο peer δεχτεί μια SQL επερώτηση στο web service που διαθέτει. Τότε χρησιμοποιώντας τον διαχειριστή της βάσης δεδομένων θέτει την επερώτηση στη βάση και εξάγει τα δεδομένα.

### 4.3.2 DataDictionary

Η λειτουργία του module είναι η γλωσσολογική ανάλυση λέξεων και η δημιουργία ενός θησαυρού όρων. Αφού δεχτεί σαν είσοδο ένα σύνολο από λέξεις, βρίσκει τα συνώνυμα, υπερώνυμα για κάθε μια από αυτές. Ο θησαυρός όρων που δημιουργείται ενσωματώνεται σε μια κατάλληλη δομή δεδομένων, η οποία και επιστρέφεται στο module που κάλεσε το DataDictionary. Κατά τη διάρκεια λειτουργίας ενός peer το module αυτό καλείται σε δύο διαφορετικές περιπτώσεις. Η πρώτη είναι όταν θέλουμε να δημοσιοποιήσουμε το σχήμα της βάσης. Στην περίπτωση αυτή, έχοντας εξάγει το σχήμα καλώντας το module DatabaseHandler, προσπαθούμε να δημιουργήσουμε έναν θησαυρό όρων. Η δεύτερη περίπτωση είναι όταν ένας super-peer δέχεται μια SQL επερώτηση και θέλει για κάθε όνομα σχέσης, πεδίου να δημιουργήσει ένα θησαυρό όρων.

Για να γίνει εφικτή η δημιουργία του θησαυρού όρων το module πραγματοποιεί για κάθε όνομα σχέσης ή πεδίου δύο συνδέσεις μια σε κάθε online λεξικό. Αφού καλέσει κάθε λεξικό με παράμετρο τη λέξη που μας ενδιαφέρει θα επιστραφεί στην εφαρμογή μια ιστοσελίδα. Η ιστοσελίδα θα περιέχει από το μεν πρώτο λεξικό τα υπερώνυμα του ονόματος που ψάχναμε, ενώ η δεύτερη θα περιέχει τα συνώνυμα. Το module χρησιμοποιεί έναν αναλυτή της HTML γλώσσας (HTMLParser), ώστε να είναι εφικτή η ανάλυση της πληροφορία που περιέχεται σε μια ιστοσελίδα. Με την τεχνική αυτή καταφέρνουμε να απομονώσουμε από τις ιστοσελίδες μόνο τις λέξεις που μας ενδιαφέρουν.

Επειδή όμως ένας peer μπορεί να συνδεθεί σε πολλές κοινότητες και να κοινοποιήσει το σχήμα της βάσης προς τον εκάστοτε super peer πολλές φορές, για το λόγο αυτό έχει

υλοποιηθεί μια τοπική cache στην οποία φυλάσσονται οι θησαυροί όρων για κάθε όνομα πεδίου, και σχέσης. Με τον τρόπο αυτό πετυχαίνουμε λιγότερη επικοινωνία με τα ηλεκτρονικά λεξικά και επομένως μεγαλύτερη ταχύτητα στο σύστημά μας καθώς οι θησαυροί όρων θα δημιουργηθούν μόνο μια φορά, την πρώτη φορά που θα κοινοποιηθεί το σχήμα. Συνεπώς, κάθε κόμβος του δικτύου κερδίζει σε απόδοση και ταχύτητα, ενώ παράλληλα ο αποθηκευτικός χώρος που απαιτείται είναι πολύ μικρός. Η τοπική cache βοηθάει επίσης στη βελτίωση της απόδοσης του super-peer καθώς για τη δημιουργία του θησαυρού όρων της SQL επερώτησης μπορεί να τη συμβουλευτεί και να αποφύγει κάποιες χρονοβόρες συνδέσεις στα λεξικά.

### 4.3.3 XmlHandler

Το module XmlHandler χρησιμοποιείται για δύο λόγους. Ο πρώτος είναι η δημιουργία ενός XML αρχείου, το οποίο θα περιγράφει το σχήμα της βάσης μαζί με το θησαυρό όρων. Η XML χρησιμοποιείται για την ανταλλαγή των δεδομένων. Πολλές φορές τα υπολογιστικά συστήματα και οι βάσεις δεδομένων περιέχουν δεδομένα σε μη συμβατά σχήματα. Μια από της πιο χρονοβόρες διαδικασίες για τους μηχανικούς λογισμικού είναι η ανταλλαγή δεδομένων μεταξύ τέτοιων συστημάτων στο Διαδίκτυο. Μετατρέποντας τα δεδομένα σε XML μπορούμε να μειώσουμε σημαντικά την πολυπλοκότητα και να δημιουργήσουμε δεδομένα που να μπορούν να διαβαστούν από πολλούς διαφορετικούς τύπους εφαρμογών. Η γενική δομή που έχει το XML έγγραφο φαίνεται παρακάτω.

```
<?xml version="1.0"?>
<SCHEMA AUTHOR=peer-name>
  <TABLE> Table Name
    <SYN> synonym1 </SYN>
    <SYN> synonym2 </SYN>
    .
    .
    .
    <HYPER> hypernym1 </HYPER>
    <HYPER> hypernym2 </HYPER>
    .
    .
    .
    <KEYWORD0> keyword 0 </KEYWORD0>
    <KEYWORD0_SYN> synonym1 for keyword0 </KEYWORD0_SYN>
    <KEYWORD0_SYN> synonym2 for keyword0 </KEYWORD0_SYN>
    .
    .
```



```

.
<KEYWORD0_HYPER> hypernym1 for keyword0 </KEYWORD0_HYPER>
<KEYWORD0_HYPER> hypernym2 for keyword0 </KEYWORD0_HYPER>
.
.
.
<KEYWORD1> keyword 1 </KEYWORD1>
<KEYWORD1_SYN> synonym1 for keyword1 </KEYWORD1_SYN>
<KEYWORD1_SYN> synonym2 for keyword1 </KEYWORD1_SYN>
.
.
.
<KEYWORD1_HYPER> hypernym1 for keyword1 </KEYWORD1_HYPER>
<KEYWORD1_HYPER> hypernym2 for keyword1 </KEYWORD1_HYPER>
.
.
.
<ITEM> Attribute Name
  <ITEMTYPE> Domain of attribute </ITEMTYPE>
  <SYN> synonym1 </SYN>
  <SYN> synonym2 </SYN>
.
.
.
<HYPER> hypernym1 </HYPER>
<HYPER> hypernym2 </HYPER>
.
.
.
<KEYWORD0> keyword 0 </KEYWORD0>
<KEYWORD0_SYN> synonym1 for keyword0 </KEYWORD0_SYN>
<KEYWORD0_SYN> synonym2 for keyword0 </KEYWORD0_SYN>
.
.
.
<KEYWORD0_HYPER> hypernym1 for keyword0 </KEYWORD0_HYPER>
<KEYWORD0_HYPER> hypernym2 for keyword0 </KEYWORD0_HYPER>
.
.
.
<KEYWORD1> keyword 1 </KEYWORD1>
<KEYWORD1_SYN> synonym1 for keyword1 </KEYWORD1_SYN>
<KEYWORD1_SYN> synonym2 for keyword1 </KEYWORD1_SYN>
.
.
.
<KEYWORD1_HYPER> hypernym1 for keyword1 </KEYWORD1_HYPER>

```

```

<KEYWORD1_HYPER> hypernym2 for keyword1 </KEYWORD1_HYPER>
.
.
.
</ITEM>
.
.
.
<PRKEY> Attribute Name </PRKEY>
.
.
.
<FRKEY> Attribute Name <REF ITEM=" Attribute Name "> Table Name </REF> </FRKEY>
.
.
.
</TABLE>
.
.
.
</SCHEMA>

```

Η δομή που ακολουθείται έχει σαν πρώτο στάδιο τη δήλωση του συγγραφέα του εγγράφου. Αυτό είναι το στοιχείο «ρίζα» του εγγράφου. Στη συνέχεια, παρατίθενται διαδοχικά οι σχέσεις και τα πεδία που υπάρχουν στο σχήμα της βάσης δεδομένων με τη μορφή στοιχείων. Κάθε ένα από τα στοιχεία αυτά είναι μικτού περιεχομένου καθώς περιέχει κείμενο και άλλα στοιχεία. Αρχικά εμφανίζεται το στοιχείο TABLE. Το κείμενο που περιέχει είναι το όνομα της σχέσης. Κάθε στοιχείο TABLE έχει επιπλέον στοιχεία τα οποία περιγράφουν το θησαυρό όρων που έχει δημιουργηθεί και τα πεδία της σχέσης. Συνολικά υπάρχουν τα ακόλουθα στοιχεία:

- SYN. Το στοιχείο SYN περιέχει ένα συνώνυμο του ονόματος της σχέσης. Το πλήθος των στοιχείων SYN είναι ίσο με το πλήθος των συνωνύμων που έχουν βρεθεί.
- HYPER. Το στοιχείο HYPER περιέχει ένα υπερώνυμο του ονόματος της σχέσης. Το πλήθος των στοιχείων HYPER είναι ίσο με το πλήθος των υπερωνύμων που έχουν βρεθεί.
- KEYWORD0, KEYWORD1. Το στοιχείο KEYWORD0 περιέχει την πρώτη λέξη κλειδί που έδωσε ο χρήστης, ενώ το KEYWORD1 τη δεύτερη λέξη κλειδί.
- KEYWORD0\_SYN, KEYWORD1\_SYN. Τα στοιχεία KEYWORD0\_SYN περιέχει τα συνώνυμα της πρώτης λέξης κλειδιού, ενώ τα στοιχεία

KEYWORD1\_SYN τα συνώνυμα για τη δεύτερη λέξη κλειδί. Το πλήθος τους είναι ίσο με τον αριθμό των συνωνύμων που έχουν βρεθεί.

- KEYWORD0\_HYPER, KEYWORD1\_HYPER. Τα στοιχεία KEYWORD0\_HYPER περιέχει τα υπερώνυμα της πρώτης λέξης κλειδιού, ενώ η KEYWORD1\_HYPER τα υπερώνυμα για τη δεύτερη λέξη κλειδί. Το πλήθος τους είναι ίσο με τον αριθμό των υπερωνύμων που έχουν βρεθεί.
- ITEM. Κάθε στοιχείο ITEM περιέχει ένα πεδίο της σχέσης με τα μεταδεδομένα που έχουν βρεθεί για αυτό.

Τα στοιχεία με ετικέτα ITEM είναι αυτά που περιέχουν πληροφορία σχετικά με τα πεδία της σχέσης στην οποία ανήκουν. Τα στοιχεία αυτά είναι επίσης μεικτά και περικλείουν τα θυγατρικά στοιχεία που υπάρχουν στο στοιχείο με ετικέτα TABLE καθώς και ένα ακόμα στοιχείο με ετικέτα ITEMTYPE το οποίο περιγράφει την περιοχή τιμών του πεδίου. Το κείμενο του στοιχείου είναι το όνομα του πεδίου. Το πλήθος των στοιχείων είναι ίσο με το πλήθος των πεδίων της σχέσης. Στη συνέχεια ακολουθούν τα στοιχεία απλού περιεχομένου με ετικέτα PRKEY και τα στοιχεία μεικτού περιεχομένου με ετικέτα FRKEY. Τα στοιχεία με ετικέτα PRKEY δεν έχουν ιδιότητες και στο κείμενο δηλώνεται το όνομα του πρωτεύοντος κλειδιού της σχέσης. Ομοίως τα στοιχεία με ετικέτα FRKEY δεν έχουν ιδιότητες και περιγράφουν το όνομα του εξωτερικού κλειδιού. Περικλείουν όμως ένα ακόμα στοιχείο που περιέχει το όνομα της σχέσης στην οποία αναφέρεται το εξωτερικό κλειδί. Το πλήθος των στοιχείων με ετικέτα PRKEY, FRKEY είναι ίσο με το πλήθος των πρωτεύον και εξωτερικών κλειδιών αντίστοιχα που περιέχει μια σχέση. Αφού περιγραφούν και τα εξωτερικά κλειδιά η διαδικασία επαναλαμβάνεται για τις επόμενες σχέσεις της βάσης δεδομένων.

Ο δεύτερος λόγος που χρησιμοποιείται το module είναι στην περίπτωση που ο super peer πρέπει να κάνει ανάλυση των XML εγγράφων που του στέλνουν οι peers. Το module χρησιμοποιεί έναν αναλυτή της XML γλώσσας (XMLParser), ώστε να είναι εφικτή η ανάλυση της πληροφορία που περιέχεται μέσα στο έγγραφο. Ο super peer γνωρίζοντας τη δομή που ακολουθεί το XML έγγραφο και χρησιμοποιώντας τον αναλυτή, καταφέρνει να απομονώσει τα στοιχεία που τον ενδιαφέρουν και τα οποία είναι τα ονόματα των σχέσεων, πεδίων και οι θησαυροί όρων που υπάρχουν για κάθε ένα όνομα. Επιπλέον, διατηρεί τα πρωτεύον και εξωτερικά κλειδιά που έχει κάθε σχέση. Όλα τα δεδομένα ο super-peer τα διατηρεί σε εσωτερικές δομές δεδομένων, ενώ παράλληλα τα XML έγγραφα τα αποθηκεύει σε τοπική cache που διατηρεί για το σκοπό αυτό.

Συνεπώς, το module XmlHandler είναι δυνατόν να κληθεί τόσο από ένα απλό peer για την παροχή του XML εγγράφου το οποίο θα δημοσιοποιήσει αργότερα, όσο και από τον super peer για να αποσυνθέσει τα XML έγγραφα που συλλέγει από τους peers, και να εξάγει τα δεδομένα που χρειάζεται στο επόμενο στάδιο που είναι η αντιστοίχιση σχημάτων και ο μετασχηματισμός μιας SQL επερώτησης.

#### 4.3.4 QueryProcessor

Το module QueryProcessor περιλαμβάνει δύο λειτουργίες. Η πρώτη είναι η αναζήτηση και δημιουργία αντιστοιχίσεων μεταξύ των σχημάτων, κατά την οποία προσπαθεί να ανακαλύψει γλωσσολογικές ομοιότητες μεταξύ των ονομάτων των πεδίων, και των σχέσεων που περιέχει το σχήμα ενός peer και αυτών που περιλαμβάνονται σε μια SQL επερώτηση. Αναζητά επομένως τους peers που μπορούν να απαντήσουν την επερώτηση. Η δεύτερη λειτουργία είναι ο μετασχηματισμός της SQL επερώτησης, που βασίζεται στο σχήμα της βάσης το οποίο διαθέτει κάθε peer που μπορεί να απαντήσει στην επερώτηση.

Το module QueryProcessor χρησιμοποιείται μόνο από τον super peer και καλείται μόνο όταν φθάνει κάποια SQL επερώτηση στον server που διατηρεί για το σκοπό αυτό. Συνακόλουθα το module αυτό καλεί δύο άλλα modules. Το πρώτο είναι ο XmlHandler για να αποσυνθέσει τα έγγραφα που συλλέγει από τους peers και το δεύτερο είναι το DataDictionary για να μπορέσει να δημιουργήσει ένα θησαυρό όρων για την SQL επερώτηση. Για να καταλήξει όμως στη δημιουργία του θησαυρού όρων θα πρέπει να αποσυνθέσει την επερώτηση και να πάρει τα ονόματα των πεδίων, και των σχέσεων. Για το λόγο αυτό το module χρησιμοποιεί έναν αναλυτή της SQL γλώσσας (SQLParser-ZQL), έτσι ώστε να μπορέσει να εξάγει από την επερώτηση τα δεδομένα που επιθυμεί. Η χρήση του αναλυτή διευκολύνει τον RDPeer στη σύνθεση των SQL επερωτήσεων, που προκύπτουν από το μετασχηματισμό της αρχικής. Εκτός από τα ονόματα των σχέσεων, πεδίων συλλέγονται και οι εκφράσεις που περιέχονται στον όρο SELECT και WHERE για να είναι δυνατόν στη φάση του μετασχηματισμού να σχηματίσουμε τις SQL επερωτήσεις. Για να γίνει πιο γρήγορη η διαδικασία της δημιουργίας του θησαυρού όρων για την επερώτηση εκμεταλλευόμαστε την τοπική cache που έχει δημιουργήσει ο peer, έτσι ώστε να μη χρειαστεί άσκοπα να συνδεόμαστε στα ηλεκτρονικά λεξικά και να βελτιώσουμε συνεπώς την ταχύτητα της εφαρμογής.

#### 4.3.5 JXTA module

Το JXTA module χρησιμοποιείται για να μπορέσουμε να επωφεληθούμε από την λειτουργικότητα που προσφέρει η πλατφόρμα JXTA. Χάρη στο module αυτό κάθε peer μπορεί:

1. Να διαπιστώσει ποιοι peers συμμετέχουν σε μια κοινότητα.
2. Να διαπιστώσει ποιες κοινότητες έχουν δημιουργηθεί.
3. Να δημιουργήσει μια κοινότητα.

4. Να γίνει μέλος μιας κοινότητας.
5. Να εγκαταλείψει μια κοινότητα.
6. Να επικοινωνήσει με τους άλλους peers.
7. Να ανακαλύψει τον super peer.
8. Να ανακαλύψει δημοσιοποιήσεις που έχουν κάνει τόσο οι άλλοι peers όσο και ο ίδιος.

Επομένως, το module JXTA διευκολύνει στην ανακάλυψη και έκδοση δημοσιοποιήσεων. Επιπλέον, μέσω του module μπορούμε να επιτύχουμε τη δρομολόγηση των μηνυμάτων, τον έλεγχο της cache που διατηρεί κάθε peer, την επεξεργασία των δημοσιοποιήσεων, και τη δημιουργία σωληνώσεων μεταξύ των peers.

### 4.3.6 Webservice

Το module αυτό υλοποιεί το web service που φέρει κάθε peer και το οποίο αναλαμβάνει να εξυπηρετήσει την SQL επερώτηση που δέχεται από έναν άλλο peer. Για να εξυπηρετήσει την επερώτηση συνδέεται στη βάση δεδομένων, εκτελεί την επερώτηση, παίρνει τις απαντήσεις και τις επιστρέφει στον peer που έκανε τη σύνδεση στην υπηρεσία. Το module τρέχει στον jakarta-tomcat web server. Κάθε peer του συστήματος επομένως φέρει και έναν web server, έτσι ώστε να μπορεί να υποστηρίζει το web service που έχει. Η ύπαρξη του web server προσδίδει μια καλύτερη απόδοση στην εξυπηρέτηση των αιτήσεων. Αυτό συμβαίνει, γιατί κάθε peer επικαλείται την υπηρεσία μέσω του HTTP πρωτοκόλλου με αποτέλεσμα να μην έχει προβλήματα πρόσβασης λόγω της ύπαρξης firewall. Επιπλέον, ο web server έχει την ικανότητα να εξυπηρετήσει μεγάλο αριθμό ταυτόχρονων αιτήσεων. Η κλήση ενός web service είναι το τελικό στάδιο της διαδικασίας εύρεσης δεδομένων για μια SQL επερώτηση. Προϋποθέτει αρχικά ότι έχει γίνει η επεξεργασία του μετασχηματισμού της SQL επερώτησης από τον super-peer και έχουν επιστραφεί στον peer που αρχικοποιεί την επερώτηση οι μετασχηματισμένες επερωτήσεις και οι peers που αντιστοιχούν σ' αυτές. Με τον τρόπο αυτό ένας peer μπορεί να εξάγει τα δεδομένα που έχουν στις βάσεις δεδομένων οι υπόλοιποι peers μιας κοινότητας.

### 4.3.7 RequestServer

Το module RequestServer είναι ένας πολυνηματικός εξυπηρετητής που ως σκοπό έχει την αποδοχή και εξυπηρέτηση SQL επερωτήσεων, και ο οποίος είναι μέρος της λειτουργικότητας που έχει ο super-peer. Για να επιτευχθεί η εξυπηρέτηση μιας επερώτησης ο server αρχικά καλεί το JXTA module για να αναζητήσει και να συλλέξει

όλα τα σχήματα των βάσεων δεδομένων που έχουν κοινοποιήσει οι peers. Στη συνέχεια, το JXTA module καλεί το module QueryProcessor, το οποίο όπως αναφέραμε σε προηγούμενη ενότητα αναλαμβάνει να πράξει όλες τις απαραίτητες ενέργειες για να ανακαλύψει όλους τους peers που μπορούν να το απαντήσουν. Για να μπορέσει ένας peer να συνδεθεί στον server πρέπει να ανακαλύψει τη δημοσιοποίηση σωλήνα που έχει δημιουργήσει ο super-peer, και η οποία περιγράφει το port και τη διεύθυνση, στην οποία ο ακούει ο server. Η δομή της επικοινωνίας που έχει υλοποιηθεί για να επικοινωνήσει ένας peer με τον super-peer θα περιγραφεί σε επόμενη ενότητα.

### 4.3.8 ElectionServer

Το module ElectionServer είναι επίσης ένας πολυνηματικός εξυπηρετητής τον οποίο θέτει σε λειτουργία κάθε peer όταν εισέρχεται σε μια κοινότητα. Το module αυτό χρησιμεύει σε κάθε peer, για να μπορούν να επικοινωνήσουν οι peers μεταξύ τους και να διεξάγουν εκλογές για την ανάδειξη καινούργιου super-peer. Επιπλέον, βοηθάει στην ανακάλυψη των ενεργών peers που βρίσκονται μέσα σε μια κοινότητα με την ανταλλαγή ping pong μηνυμάτων.

Αν και η πλατφόρμα JXTA προσφέρει τη δυνατότητα εύρεσης των peers που ανήκουν σε μια κοινότητα, η λειτουργία αυτή κρίθηκε ανεπαρκής. Αυτό συμβαίνει, εξαιτίας της cache της πλατφόρμας που διατηρεί κάθε peer και στην οποία από τη στιγμή που θα γραφτεί μια δημοσιοποίηση, παραμένει ενεργή μέχρι που να παρέλθει το χρονικό όριο ζωής που φέρει (το προκαθορισμένο χρονικό όριο είναι μία μέρα), με αποτέλεσμα την ανακάλυψη peer που έχει πάψει να είναι μέλος της κοινότητας. Επακόλουθο της παραπάνω λειτουργίας είναι ότι για να καταφέρουμε να ανακαλύπτουμε ορθά τους peers που ανήκουν σε μια κοινότητα θα έπρεπε οι δημοσιοποιήσεις που γίνονταν να φέρουν χρονικό όριο ζωής και κάθε φορά που λήγει αυτό να επανεκδίδονται. Αυτό όμως θα επέφερε προβλήματα σε ότι αφορά την επιβάρυνση της κίνησης του δικτύου και στην ανακάλυψη των peers, καθώς για την πλατφόρμα JXTA αυτή είναι μια ασύγχρονη διαδικασία. Συνεπώς, επιλέχτηκε η χρήση του ElectionServer και συγκεκριμένα η ανταλλαγή των ping pong μηνυμάτων.

### 4.3.9 Τρόποι επικοινωνίας μεταξύ peers

Η επικοινωνία ενός peer με τους υπόλοιπους πραγματοποιείται σε τρία επίπεδα:

1. Αποστολή δημοσιοποιήσεων, όταν οι peers κάνουν κοινοποίηση του σχήματος των βάσεων δεδομένων προς τον super-peer, ή κοινοποιούν την υπηρεσία που έχουν.

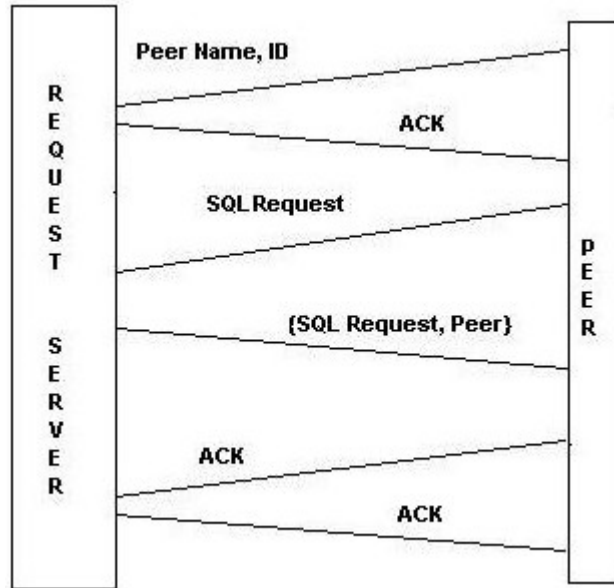
2. Με κλήση SOAP-RPC μπορεί ένας peer να στείλει τη μετασχηματισμένη SQL επερώτηση στον peer που διαθέτει δεδομένα στη βάση δεδομένων του.
3. Μέσω σωλήνων που εγκαθιδρύει ένας peer είτε με τον ElectionServer ενός άλλου peer της κοινότητας, είτε με τον RequestServer που έχει τεθεί σε λειτουργία από το super-peer.

Η δομή που έχει η επικοινωνία στο πρώτο επίπεδο εκμεταλλεύεται πλήρως την πλατφόρμα JXTA, την οποία παράλληλα επεκτείνει. Στο σημείο αυτό ο peer μπορεί πρώτον να δημιουργεί μια ξεχωριστή δημοσιοποίηση σε μορφή XML εγγράφου, ομοίως με την πλατφόρμα. Η δημοσιοποίηση περιέχει το όνομα του peer, ένα μοναδικό αναγνωριστικό, και το σχήμα της βάσης δεδομένων που διαχειρίζεται ο peer. Επιπρόσθετα, είναι δυνατόν να κατασκευάσει μια ακόμα δημοσιοποίηση που περιέχει το όνομα του peer, ένα μοναδικό αναγνωριστικό, το όνομα της υπηρεσίας και το URL στο οποίο μπορεί να καλεστεί η υπηρεσία. Ο peer δημιουργώντας τα XML έγγραφα μπορεί στη συνέχεια να τα κοινοποιήσει χρησιμοποιώντας το πρωτόκολλο ανακάλυψης peer που διαθέτει η πλατφόρμα. Με τον τρόπο αυτό ένας peer στέλνει πληροφορία τόσο στον super-peer, όσο και στους υπόλοιπους peers.

Στο δεύτερο επίπεδο έχουμε την κλήση απομακρυσμένης συνάρτησης με παράμετρο μια SQL επερώτηση και επιστροφή των δεδομένων που προκύπτουν από την εκτέλεση της SQL επερώτησης στη βάση δεδομένων.

#### 4.3.9.1 Δομή επικοινωνίας με RequestServer

Η δομή αυτή αφορά την επικοινωνία μεταξύ ενός peer μιας κοινότητας με το server που φέρει ο super-peer, προκειμένου να αποσταλεί μια επερώτηση. Αρχικά ο peer είναι υποχρεωμένος να αναζητήσει και να ανακαλύψει τη δημοσιοποίηση σωλήνα που έχει κοινοποιήσει ο super-peer και η οποία περιγράφει το port και τη διεύθυνση στην οποία μπορεί να εγκαθιδρυθεί η σύνδεση. Η δομή του πρωτοκόλλου είναι απλή και φαίνεται στο παρακάτω σχήμα.



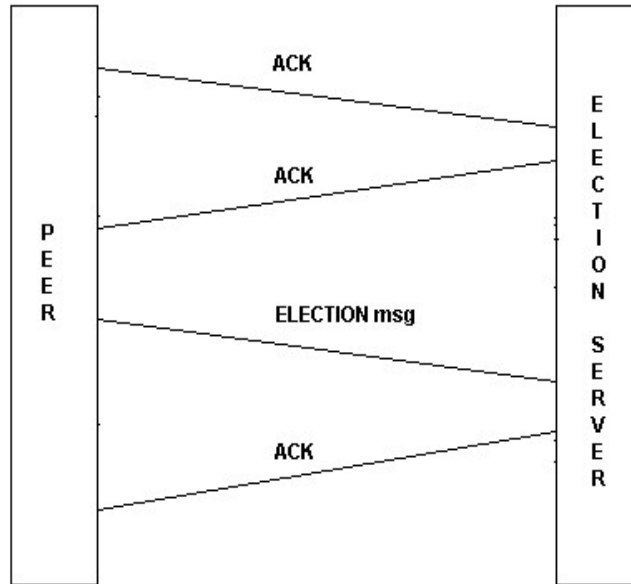
Σχήμα 5 Η δομή επικοινωνίας ενός peer με το RequestServer

Αρχικά ένας απλός peer στέλνει στον super-peer μέσω του σωλήνα ένα μήνυμα που περιέχει το όνομά του και το μοναδικό χαρακτηριστικό γνώρισμα που τον προσδιορίζει και ο super-peer απαντάει με μια βεβαίωση λήψης που είναι ένα απλό μήνυμα pong. Στη συνέχεια ο peer αποστέλλει ένα μήνυμα το οποίο περιέχει την SQL επερώτηση. Ο request server αφού επεξεργαστεί την επερώτηση απαντάει με ένα μήνυμα, το οποίο περιέχει ένα σύνολο από ζεύγη αντικειμένων. Κάθε ζεύγος αποτελείται από μια SQL επερώτηση και το όνομα του peer στον οποίο αντιστοιχεί. Με τη λήψη του συνόλου των ζευγών, ο peer απαντάει με ένα μήνυμα exit, το οποίο βεβαιώνει την περάτωση της διαδικασίας επικοινωνίας και ομοίως πράττει και ο server.

#### 4.3.9.2 Δομή επικοινωνίας με ElectionServer

Ο ElectionServer χρησιμοποιείται από κάθε peer για δύο λόγους. Ο πρώτος είναι για να διαπιστώσει τους ενεργούς peer που συμμετέχουν σε μια κοινότητα και ο δεύτερος είναι για να μπορέσουν οι peers να ανταλλάξουν μηνύματα, ώστε να μπορέσουν να εκλέξουν ένα καινούργιο super-peer. Στην πρώτη περίπτωση η δομή της επικοινωνίας είναι απλή και περιλαμβάνει την ανταλλαγή ring pong μηνυμάτων μεταξύ δύο peer. Στη δεύτερη περίπτωση η δομή είναι πιο σύνθετη. Στο σχήμα που ακολουθεί περιγράφεται η ενέργεια που διεξάγει ένας peer, όταν αντιληφθεί την απουσία του super-peer.





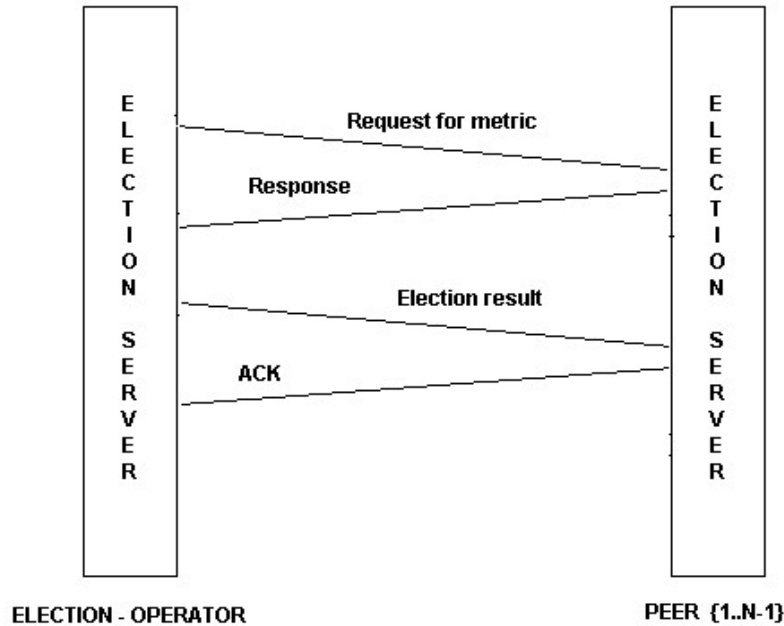
**Σχήμα 6** Η δομή της επικοινωνίας μεταξύ του peer, που ανακαλύπτει την απουσία του super-peer, με τους υπόλοιπους peers της κοινότητας.

Ο peer που διαπιστώνει την απουσία του super-peer επικοινωνεί με τους sever των άλλων peers για να τους ανακοινώσει τη διενέργεια των εκλογών. Αρχικά ανταλλάσσει ping pong μηνύματα για να διαπιστώσει ποιοι peers της κοινότητας είναι ενεργοί. Στη συνέχεια και αφού λάβει το μήνυμα pong αποστέλλει ένα μήνυμα εκλογών. Το μήνυμα αποτελείται από τα παρακάτω αντικείμενα:

- Το όνομα του peer
- Το αναγνωριστικό του peer
- Το όνομα, αναγνωριστικό του peer που θα διεξάγει τις εκλογές
- Το όνομα, αναγνωριστικό του αναπληρωματικού διαχειριστή των εκλογών
- Τη λέξη elections.

Ο εκάστοτε peer που λαμβάνει το μήνυμα απαντάει με μια βεβαίωση λήψης και στη συνέχεια στο πλαίσιο της προετοιμασίας των εκλογών καθαρίζει την τοπική cache που διαθέτει από τυχόν δημοσιοποιήσεις που είχε αποθηκεύσει. Η προετοιμασία αυτή συμβαίνει μόνο στο πρώτο μήνυμα εκλογών που θα λάβει. Αν λάβει και δεύτερο δεν επιτελεί καμία ενέργεια. Συνεπώς, ο peer που στέλνει το μήνυμα των εκλογών δε σημαίνει ότι θα τις διεξάγει κιόλας. Η επικοινωνία με τα άλλα μέλη της κοινότητας πραγματοποιείται μόνο αφού ανακαλύψει τον peer που θα διεξάγει τις εκλογές και ο οποίος προσδιορίζεται με βάση το μοναδικό αναγνωριστικό που φέρει κάθε peer. Ο peer που έχει τη μεγαλύτερη τιμή αναγνωριστικού θα είναι ο διαχειριστής των εκλογών.

Ο peer που διεξάγει τις εκλογές θα πρέπει μέσω του server να επικοινωνήσει με τους άλλους peers της κοινότητας για να διαπιστώσει ποιος είναι ο καταλληλότερος για να γίνει νέος super-peer. Στο παρακάτω σχήμα φαίνεται η δομή που θα έχει η επικοινωνία.



Σχήμα 7 Η δομή της επικοινωνίας του διαχειριστή των εκλογών με τους υπόλοιπους peers της κοινότητας.

Η επικοινωνία μεταξύ του peer που διαχειρίζεται τις εκλογές και όλων των υπολοίπων που είναι μέλη της κοινότητας είναι δύο επιπέδων. Στην αρχή αποστέλλεται ένα μήνυμα με το οποίο ζητείται από κάθε peer να δηλώσει τον αριθμό των SQL επερωτήσεων που είχε υποβάλλει. Ο εκάστοτε peer δεχόμενος το μήνυμα αποκρίνεται με ένα μήνυμα που περιέχει το όνομά του, το αναγνωριστικό του και το πλήθος των επερωτήσεων που είχε θέσει. Ο διαχειριστής των εκλογών ανακαλύπτει από τις απαντήσεις που έλαβε τον peer που είχε θέσει τον μεγαλύτερο αριθμό επερωτήσεων και στέλνει το αποτέλεσμα των εκλογών, δηλαδή ένα μήνυμα που περιέχει το όνομα, αναγνωριστικό του νέου super-peer. Κάθε peer απαντάει στέλνοντας μια βεβαίωση λήψης και η επικοινωνία τερματίζεται.

## 4.4 Αντιστοίχιση σχημάτων και μετασχηματισμός SQL επερώτησης

Ένας από τους κύριους σκοπούς του RDPeer είναι να επιτρέψει στους χρήστες να διαχειριστούν και να διαμοιράσουν τα δεδομένα τους χρησιμοποιώντας ένα σύστημα διαχείρισης βάσης δεδομένων. Όπως έχει αναφερθεί δεν υπάρχει ένα προκαθορισμένο και ομοιόμορφο σχήμα το οποίο να μοιράζονται οι κόμβοι του δικτύου. Στην εφαρμογή που υλοποιήθηκε θεωρούμε ότι τα δεδομένα μπορούν να οριστούν με διαφορετικό τρόπο από διαφορετικούς χρήστες ακόμα και αν αυτοί έχουν κοινά ενδιαφέροντα. Για τον κατανεμημένο διαμοιρασμό των δεδομένων χρησιμοποιούμε μια κεντρικοποιημένη αρχή για να μπορέσουμε να διαχειριστούμε πολλαπλές πηγές δεδομένων και θεωρούμε ότι κατά τη διάρκεια αυτής της διαδικασίας δεν υπάρχει ανάμειξη των χρηστών. Επιπλέον, υπήρχαν δύο παράμετροι κλειδιά στο διαμοιρασμό ετερογενών δεδομένων, οι οποίοι ήταν αφενός ότι ο αριθμός των σχημάτων ήταν πολύ μεγάλος και αφετέρου ότι η δημιουργία ενός κεντρικοποιημένου ενδιάμεσου σχήματος δεν είναι δυνατή. Η δημιουργία του κεντρικοποιημένου ενδιάμεσου σχήματος έχει ως σκοπό την ολοκλήρωση των δεδομένων πολλαπλών πηγών δεδομένων.

Με κάθε κόμβο ενός peer-to-peer συστήματος να συμμετέχει και να εγκαταλείπει το δίκτυο οποιαδήποτε χρονική στιγμή, η θεώρηση ενός καθολικού σχήματος σε ένα τέτοιο δυναμικό περιβάλλον δεν είναι πρακτική, κλιμακώσιμη, και επεκτάσιμη. Μία πιθανή προσέγγιση είναι να επιτρέπεται σε κάθε κόμβο να προκαθορίσει τις αντιστοιχίσεις μεταξύ του σχήματός του και των σχημάτων των υπόλοιπων κόμβων πριν τεθεί η SQL επερώτηση [3,12,13,16,24]. Μια εναλλακτική προσέγγιση είναι η δημιουργία αντιστοιχίσεων κατά τη διάρκεια της επερώτησης [30]. Η δημιουργία αντιστοιχίσεων μεταξύ των σχημάτων έχει ως σκοπό την εύρεση των κόμβων που μπορούν να απαντήσουν μια SQL επερώτηση και την ορθή δρομολόγηση της επερώτησης σ' αυτούς τους κόμβους του δικτύου.

Η πρώτη προσέγγιση δεν υιοθετήθηκε, καθώς περικλείει αρκετά προβλήματα. Όταν ο αριθμός των σχημάτων είναι μεγάλος, η δημιουργία και διατήρηση των αντιστοιχίσεων είναι μια χρονοβόρα και δαπανηρή σε πόρους διαδικασία. Επιπλέον, εξαιτίας της μεταβαλλόμενης σύνθεσης ενός peer-to-peer συστήματος, η απόδοση του συστήματος μειώνεται προκειμένου να επαναδιαμορφωθούν οι αντιστοιχίσεις. Και οι δύο προσεγγίσεις παρουσιάζουν το μειονέκτημα σχετικά με το πως μπορούν αποτελεσματικά να δημιουργηθούν και να διατηρηθούν οι αντιστοιχίσεις. Επίσης, βασικό μειονέκτημα είναι ότι ενώ μια επερώτηση θα μπορούσε να απαντηθεί από έναν κόμβο του δικτύου, σε περίπτωση που το σχήμα του δεν έχει αντιστοιχίσεις με το σχήμα του κόμβου στον οποίο τίθεται η επερώτηση, τότε αποκλείεται να δρομολογηθεί προς αυτόν η επερώτηση. Αυτό συμβαίνει, επειδή για πολλά πεδία και σχέσεις διαφορετικών σχημάτων δεν είναι

δυνατόν να δημιουργηθούν αντιστοιχίσεις βασισμένες σε κάποια προσέγγιση αντιστοίχισης σχημάτων χωρίς την παρέμβαση των χρηστών. Εξίσου σημαντικό μειονέκτημα είναι ότι για να τεθεί από ένα χρήστη ενός κόμβου μια SQL επερώτηση θα πρέπει να γνωρίζει το σχήμα της βάσης που διαθέτει.

Στην εφαρμογή που υλοποιήσαμε δεν υιοθετήσαμε καμία από τις δύο προσεγγίσεις, αλλά επιλέξαμε τη δυναμική δημιουργία αντιστοιχίσεων μεταξύ του σχήματος μιας SQL επερώτησης και των σχημάτων που φέρουν οι peers, κατά τη διάρκεια που τίθεται η επερώτηση. Η διαδικασία εύρεσης των αντιστοιχίσεων γίνεται κεντρικοποιημένα, ώστε να επιτευχθεί καλύτερη απόδοση. Η δημιουργία των αντιστοιχίσεων έχει ως σκοπό το μετασχηματισμό της αρχικής SQL επερώτησης, ώστε να μπορεί να απαντηθεί από τους peers για τους οποίους βρέθηκαν αντιστοιχίσεις. Επομένως, ένας peer μιας κοινότητας αναλαμβάνει το ρόλο του ενδιάμεσου φορέα των σχημάτων της κοινότητας και κάθε φορά που τίθεται μια SQL επερώτηση κοινοποιείται σε αυτόν, ώστε να βρει αρχικά τις αντιστοιχίσεις και να πραγματοποιήσει έπειτα το μετασχηματισμό. Ο RDPeer συνεπώς έχει υλοποιηθεί έτσι ώστε, αντί να απαιτεί μια συνολική συμφωνία σε ένα μοναδικό ενοποιημένο σχήμα, να παρέχει τη δυνατότητα απάντησης επερωτήσεων πάνω από ένα αυθαίρετο δίκτυο τοπικών σχημάτων.

#### 4.4.1 Δημιουργία αντιστοιχίσεων

Για να επιτευχθεί η δημιουργία των αντιστοιχίσεων μεταξύ του σχήματος μιας SQL επερώτησης και των σχημάτων των peers επιλέχθηκε η γλωσσολογική προσέγγιση [34]. Η προσέγγιση αυτή εφαρμόζεται μόνο σε επίπεδο σχήματος. Η γλωσσολογική προσέγγιση χρησιμοποιεί ονόματα και κείμενο για να βρει σημασιολογικά όμοια στοιχεία σχημάτων και μπορεί να διακριθεί σε δύο επιμέρους τεχνικές. Η πρώτη είναι η αντιστοίχιση ονομάτων και η δεύτερη η αντιστοίχιση περιγραφών. Για λόγους απλότητας και καλύτερης απόδοσης υιοθετήθηκε και υλοποιήθηκε η πρώτη τεχνική. Σύμφωνα με την πρώτη τεχνική τα ονόματα των σχέσεων και των πεδίων ενός σχήματος συνταιριάζονται με ίσα ή παρόμοια ονόματα άλλων σχημάτων. Η ομοιότητα των ονομάτων μπορεί να οριστεί και να μετρηθεί με διάφορους τρόπους όπως:

- Με την ισότητα των ονομάτων
- Με την ισότητα των συνωνύμων
- Με την ισότητα των υπερωνύμων
- Με την παροχή από το χρήστη όμοιων ονομάτων
- Με την ομοιότητα των ονομάτων βασισμένη είτε σε κοινά υποσύνολα αλφαριθμητικών, είτε στην προφορά, είτε στον τρόπο που ηχούν.

- Με την ισότητα των κανονικών αντιπροσωπεύσεων ονομάτων ύστερα από επεξεργασία. (π.χ. CName -> Customer Name, EmpNo -> employee number)

Από τους παραπάνω τρόπους αυτοί που χρησιμοποιήσαμε στην υλοποίηση είναι οι τέσσερις πρώτοι.

Την ομοιότητα των ονομάτων βασισμένη είτε σε κοινά υποσύνολα αλφαριθμητικών, είτε στην προφορά, είτε στον τρόπο που ηχούν δεν τη χρησιμοποιήσαμε όπως επίσης την ισότητα των κανονικών αντιπροσωπεύσεων ονομάτων ύστερα από επεξεργασία, καθώς θα μπορούσαν να μας οδηγήσουν σε λανθασμένα συμπεράσματα, ενώ παράλληλα θα απαιτούσε πολύ περισσότερους πόρους καθώς το πλήθος των ερμηνειών που υπάρχουν είναι μεγάλο και επομένως η πολυπλοκότητα αυξάνεται. Αν για παράδειγμα θεωρήσουμε το όνομα CSDName τότε το ακρωνύμιο CSD μπορεί να πάρει πολλές διαφορετικές ερμηνείες με αποτέλεσμα να μας οδηγήσει σε λάθος συμπέρασμα σχετικά με το τι εννοεί ο χρήστης.

Η εκμετάλλευση των συνωνύμων, υπερωνύμων απαιτεί τη χρήση είτε θησαυρών όρων, είτε λεξικών. Για το λόγο αυτό υλοποιήσαμε την εφαρμογή μας με τρόπο που να μπορεί να συνδέεται σε δύο online λεξικά και με τη χρήση ενός αναλυτή της HTML γλώσσας, να λαμβάνει ένας peer τα συνώνυμα και υπερώνυμα των ονομάτων των πεδίων, των σχέσεων και των λέξεων κλειδιών που τον ενδιαφέρουν. Επιπλέον, μέσω της διεπιφάνειας χρήσης που υλοποιήθηκε προσφέρουμε στο χρήστη τη δυνατότητα να επικοινωνεί με το σχήμα της βάσης δεδομένων και να εφοδιάζει την εφαρμογή με λέξεις κλειδιά, δηλαδή δικά του συνώνυμα για κάθε όνομα που εμφανίζεται στο σχήμα. Με τον τρόπο αυτό επεκτείνεται η ερμηνεία των ονομάτων που βρίσκονται στο σχήμα και προσδιορίζεται σαφέστερα το νόημα τους.

Επομένως, για να πραγματοποιηθεί ο κατανομημένος διαμοιρασμός των δεδομένων, προϋποθέτει αρχικά από κάθε κόμβο του δικτύου τη δημιουργία και διατήρηση μεταδεδομένων για κάθε όνομα σχέσης και πεδίου. Τα μεταδεδομένα αποτελούν ένα σύνολο από συνώνυμα, υπερώνυμα και λέξεις κλειδιά τα οποία περιγράφουν κάθε όνομα πεδίου, και σχέσης. Η χρήση των θησαυρών όρων είναι μείζονος σημασίας για την εφαρμογή καθώς επιτρέπει την εννοιολογική γειτνίαση όρων που διαφορετικά δε θα μπορούσαν να ταυτιστούν. Αφού δημιουργηθούν οι θησαυροί όροι από κάθε peer αποθηκεύονται τοπικά για να βελτιώσουμε την απόδοση κάθε peer και αποστέλλονται στο super-peer. Έτσι καταφέρνουμε να έχουμε έναν κεντροποιημένο φορέα των επιμέρους σχημάτων και θησαυρών όρων, ο οποίος αναλαμβάνει να αναζητήσει και να βρει τις αντιστοιχίσεις που υπάρχουν μεταξύ των ονομάτων των πεδίων και των σχέσεων που βρίσκονται σε μια SQL επερώτηση και αυτών που υπάρχουν στα σχήματα και τους θησαυρούς όρων.

Όταν ο super-peer δέχεται μια SQL επερώτηση, πραγματοποιεί αποσύνθεση της επερώτησης χρησιμοποιώντας έναν αναλυτή της SQL γλώσσας, ώστε να μπορέσει να λάβει τα ονόματα των πεδίων, και των σχέσεων, καθώς και τις εκφράσεις που υπάρχουν

είτε στον όρο WHERE, είτε στον όρο SELECT. Η αποσύνθεση της επερώτησης στην περίπτωση των εκφράσεων είναι μια αναδρομική διαδικασία. Εξάγοντας τα ονόματα μπορεί πλέον να δημιουργήσει ένα νέο θησαυρό όρων για αυτά. Για κάθε ένα όνομα δημιουργεί ένα αρχείο στο οποίο αποθηκεύονται τα συνώνυμα και ένα άλλο για τα υπερώνυμα. Με τον τρόπο αυτό ο super-peer μπορεί να επιταχύνει τη λειτουργία του σε μελλοντικές SQL επερωτήσεις, τα οποία θα περιέχουν όμοια ονόματα, αφού δε θα χρειάζεται να δημιουργεί για αυτά θησαυρούς όρων. Έχοντας συλλέξει ο super-peer τα σχήματα και τους θησαυρούς όρων των μελών της κοινότητας και δημιουργήσει το θησαυρό όρων για την SQL επερώτηση είναι έτοιμος να αναζητήσει τις ομοιότητες που υπάρχουν.

Η διαδικασία αναζήτησης των ομοιοτήτων ξεκινάει από τα ονόματα των σχέσεων και ακολουθεί τα παρακάτω βήματα.

- Βήμα 1<sup>ο</sup>: Για κάθε μία σχέση ενός σχήματος αναζητούμε ομοιότητες, σε επίπεδο ονομάτων, συνωνύμων, υπερωνύμων, λέξεων κλειδιών με κάθε μία σχέση που περιγράφεται στον όρο FROM της SQL επερώτησης. Αν δε βρεθεί τουλάχιστον μία ομοιότητα ο peer θεωρείται ότι δε διαθέτει κατάλληλη πληροφορία για να απαντήσει στην επερώτηση και η διαδικασία μεταβαίνει στο τέταρτο βήμα. Αντίθετα εάν υπάρχουν ομοιότητες, τότε δημιουργούνται αντιστοιχίσεις.
- Βήμα 2<sup>ο</sup>: Για κάθε μία σχέση του σχήματος, για την οποία βρέθηκε ομοιότητα με κάποια σχέση του όρου FROM, αναζητούμε ομοιότητες μεταξύ των πεδίων της σχέσης του σχήματος και των πεδίων της σχέσης του όρου FROM που περιγράφονται στον όρο SELECT. Στην περίπτωση που δε βρεθεί τουλάχιστον μία ομοιότητα για ένα πεδίο μιας σχέσης, τότε η αντιστοίχιση που δημιουργήθηκε στο προηγούμενο βήμα διαγράφεται. Αν διαγραφούν όλες οι αντιστοιχίσεις ο peer θεωρείται ότι δε διαθέτει κατάλληλη πληροφορία για να απαντήσει στην επερώτηση και η διαδικασία μεταβαίνει στο τέταρτο βήμα. Αντίθετα εάν έχουν βρεθεί ομοιότητες μεταξύ των πεδίων των σχέσεων του σχήματος και των πεδίων των σχέσεων που περιγράφονται στον όρο SELECT, τότε δημιουργούνται για αυτά αντιστοιχίσεις.
- Βήμα 3<sup>ο</sup>: Ομοίως με το δεύτερο βήμα για κάθε μια σχέση του σχήματος για την οποία βρέθηκε ομοιότητα με κάποια σχέση του όρου FROM, αναζητούμε ομοιότητες μεταξύ των πεδίων της σχέσης του σχήματος και των πεδίων της σχέσης του όρου FROM που περιγράφονται στον όρο WHERE. Στην περίπτωση εύρεσης ομοιοτήτων μεταξύ πεδίων δημιουργούνται αντιστοιχίσεις.
- Βήμα 4<sup>ο</sup>: Εάν τα σχήματα που απέστειλαν οι peers δεν έχουν εξαντληθεί η διαδικασία μεταβαίνει στο πρώτο βήμα και επαναλαμβάνεται για κάποιο άλλο σχήμα, διαφορετικά τερματίζει.

Επομένως αρχικό μέλημα είναι να βρούμε ομοιότητες στο επίπεδο των σχέσεων και εφόσον βρεθούν ψάχνουμε για ομοιότητες στο επίπεδο των πεδίων. Αυτό έχει σαν

αποτέλεσμα την ταχύτερη αναζήτηση ομοιοτήτων και όχι μόνο. Μας προσφέρει επιπλέον τη δυνατότητα να μην οδηγηθούμε σε λάθη. Δηλαδή, αποκλείεται να δημιουργήσουμε αντιστοίχιση για ένα πεδίο μιας σχέσης του σχήματος με ένα πεδίο μιας διαφορετικής σχέσης της επερώτησης. Για παράδειγμα έστω ότι έχουμε ένα σχήμα το οποίο έχει μια σχέση ονόματι COMPANY με ένα από τα πεδία να ονομάζεται ADDRESS και φθάνει στον super-peer η ακόλουθη SQL επερώτηση:

```
SELECT WORKER.ADDRESS
FROM WORKER;
```

Στην περίπτωση αυτή και θεωρώντας ότι οι θησαυροί όρων που έχουν δημιουργηθεί για τα ονόματα των σχέσεων δεν έχουν καμία ομοιότητα, το πεδίο COMPANY.ADDRESS του σχήματος δε θα ταυτιστεί με το πεδίο WORKER.ADDRESS της επερώτησης, επειδή δεν υπάρχει ομοιότητα μεταξύ των δύο σχέσεων.

Οι αντιστοιχίσεις που δημιουργούνται έχουν την εξής μορφή:

*\$S: query-element – peer-element*

*\$F: query-element – peer-element*

*\$W: query-element – peer-element*

Η πρώτη μορφή δηλώνει ότι έχουμε στον όρο SELECT ένα στοιχείο της SQL επερώτησης το οποίο ταυτίζεται με ένα στοιχείο του σχήματος του peer. Ομοίως η δεύτερη μορφή περιγράφει τις ταυτίσεις που υπάρχουν στον όρο FROM, ενώ η τρίτη μορφή περιγράφει τις ταυτίσεις που βρέθηκαν στον όρο WHERE. Τα στοιχεία τόσο της επερώτησης όσο και του σχήματος του peer αναφέρονται στις αντιστοιχίσεις με βάση τα πραγματικά τους ονόματα και όχι αυτά των θησαυρών όρων. Το πλήθος των αντιστοιχίσεων δεν είναι απαραίτητο να είναι ίσο με το πλήθος των στοιχείων που έχει η SQL επερώτηση. Όμως είναι αναγκαίο, για να θεωρούμε έναν peer ότι μπορεί να απαντήσει μια SQL επερώτηση, να υπάρχει τουλάχιστον μία αντιστοίχιση για ένα στοιχείο του όρου FROM και μία αντιστοίχιση για ένα στοιχείο στον όρο SELECT. Σε κάθε περίπτωση στις αντιστοιχίσεις που δημιουργούνται για πεδία είτε του όρου SELECT, είτε του όρου WHERE, το όνομα της σχέσης στην οποία ανήκει το πεδίο πρέπει να είναι όμοιο με το όνομα της σχέσης που υπάρχει σε κάποια αντιστοίχιση που δημιουργήθηκε για τον όρο FROM.

Ας θεωρήσουμε για παράδειγμα ότι ο super-peer έχει τα σχήματα από τρεις peers, με ονόματα peer1, peer2 και peer3. Έστω ότι το σχήμα του peer1 είναι το COMPANY(NAME, ADDRESS, CITY), το σχήμα του peer2 είναι ORGANIZATION(NAME, TOWN, ADDRESS, NUM\_WORKERS), και το σχήμα του peer3 είναι WORKER(NAME, ADDRESS, TOWN). Στον πρώτο πίνακα που ακολουθεί παρατηρούμε τους θησαυρούς όρων για τα σχήματα. Στο δεύτερο πίνακα που ακολουθεί

στο πρώτο κελί διακρίνεται η SQL επερώτηση που δέχεται ο super-peer και στο δεύτερο κελί οι αντιστοιχίσεις που δημιουργεί για κάθε peer. Όπως παρατηρούμε για τον peer3 δε δημιουργούνται αντιστοιχίσεις καθώς δεν υπάρχει καμία ομοιότητα μεταξύ του θησαυρού όρων της σχέσης του σχήματός με το όνομα της σχέσης που υπάρχει στον όρο FROM. Αντίθετα τόσο τα ονόματα των σχέσεων COMPANY και ORGANIZATION, όσο και τα ονόματα των πεδίων CITY και TOWN έχουν ταύτιση στο επίπεδο των συνωνύμων.

<b>Peer1</b>	COMPANY	CALLER,COMPANION,COMPANIONSHIP,COMRADESHIP, FELLOWSHIP,INSTITUTION,ESTABLISHMENT, ORGANIZATION,ORGANISATION,SOCIAL GROUP"
	NAME	ADVERT,APPOINT,BRING UP,CALL,CITE,LANGUAGE UNIT,LINGUISTIC UNIT,PORTION,COMPONENT PART
	ADDRESS	ACCOST,COME UP TO,COMPUTER,ADDRESS,COVER, DEAL,CODE,COMPUTER CODE,CODING SYSTEM WRITING,WRITTEN COMMUNICATION
	CITY	METROPOLIS,URBAN CENTER,MUNICIPALITY,URBAN AREA,POPULATED AREA,GEOGRAPHICAL AREA, GEOGRAPHIC AREA
<b>Peer2</b>	ORGANIZATION	ARRANGEMENT,BRASS, FELLOWSHIP,INSTITUTION, CONSTITUTION,GROUP, GOVERNANCE, ESTABLISHMENT,GROUPING
	NAME	ADVERT,APPOINT,BRING UP,CALL,CITE,LANGUAGE UNIT,LINGUISTIC UNIT,PORTION,COMPONENT PART
	TOWN	TOWNSFOLK,TOWNSHIP,TOWNSPEOPLE, MUNICIPALITY,URBAN AREA,POPULATED AREA, GEOGRAPHICAL AREA,GEOGRAPHIC AREA
	ADDRESS	ACCOST,COME UP TO,COMPUTER,ADDRESS,COVER, DEAL,CODE,COMPUTER CODE,CODING SYSTEM WRITING,WRITTEN COMMUNICATION
	NUM_WORKERS	
<b>Peer3</b>	WORKER	ACTOR,DOER,PROLE,PROLETARIAN,PERSON, INDIVIDUAL,SOMEONE,SOMEBODY,MORTAL
	NAME	ADVERT,APPOINT,BRING UP,CALL,CITE,LANGUAGE UNIT,LINGUISTIC UNIT,PORTION,COMPONENT PART
	ADDRESS	ACCOST,COME UP TO,COMPUTER,ADDRESS,COVER, DEAL,CODE,COMPUTER CODE,CODING SYSTEM WRITING,WRITTEN COMMUNICATION



TOWN	TOWNSFOLK,TOWNSHIP,TOWNSPEOPLE, MUNICIPALITY,URBAN AREA,POPULATED AREA, GEOGRAPHICAL AREA,GEOGRAPHIC AREA
------	---

**Πίνακας 1** Θησαυρός όρων για ονόματα πεδίων και σχέσεων.

<b>SELECT</b> COMPANY.TOWN, COMPANY.NUM_WORKERS <b>FROM</b> COMPANY;
<u><b>Peer1</b></u> \$\$: COMPANY.TOWN – COMPANY.CITY \$\$F: COMPANY – COMPANY
<u><b>Peer2</b></u> \$\$: COMPANY.TOWN – ORGANIZATION.TOWN \$\$: COMPANY.NUM_WORKERS – ORGANIZATION.NUM_WORKERS \$\$F: COMPANY – ORGANIZATION

**Πίνακας 2** Παράδειγμα παραγωγής αντιστοιχίσεων για δύο peers.

#### 4.4.2 Μετασχηματισμός SQL επερώτησης

Η δημιουργία των αντιστοιχίσεων προσφέρει στη συνέχεια τη δυνατότητα στο super-peer να μετασχηματίσει την αρχική SQL επερώτηση, έτσι ώστε να μπορεί κάθε peer για τον οποίο υπάρχουν αντιστοιχίσεις να την κατανοήσει. Ο μετασχηματισμός της αρχικής επερώτησης έχει ως αποτέλεσμα τη δημιουργία ενός πλήθους επερωτήσεων μικρότερου ή ίσου με το πλήθος των διαφορετικών σχημάτων που έχει ο super-peer.

Οι επερωτήσεις που δημιουργούνται ορισμένες φορές μπορεί να διαφέρουν από την αρχική, δηλαδή να είναι υποσύνολα της αρχικής. Αυτό συμβαίνει γιατί ένας peer μπορεί να είναι σε θέση να απαντήσει μόνο ένα μέρος της επερώτησης και όχι το σύνολο της. Για το λόγο αυτό τα αποτελέσματα εμφανίζονται με βάση το βαθμό γειτνίασης που έχει μια μετασχηματισμένη επερώτηση με την αρχική. Ο βαθμός γειτνίασης ορίζεται ως το άθροισμα των αντιστοιχίσεων που έχουν βρεθεί για το σχήμα ενός peer. Στο παράδειγμα του πίνακα 2 η μετασχηματισμένη SQL επερώτηση που θα προκύψει για τον peer1 θα έχει βαθμό γειτνίασης 2, ενώ αυτό που θα προκύψει για τον peer2 θα έχει βαθμό γειτνίασης 3. Με τη δημιουργία επερωτήσεων που είναι υποσύνολα ή ισοδύναμα σύνολα της αρχικής πετυχαίνουμε τη μεγαλύτερη συνεισφορά δεδομένων από peers χωρίς να υστερούν οι απαντήσεις σε ορθότητα και εγκυρότητα. Αντίθετα, η δημιουργία υπερσυνόλων της αρχικής επερώτησης δεν επιτρέπεται από το σύστημά μας, καθώς με τον τρόπο αυτό αποφεύγουμε να προσφέρουμε στον χρήστη δεδομένα που δεν τον

ενδιαφέρουν και μειώνουμε επίσης και την επιβάρυνση της κίνησης στο δίκτυο. Υιοθετώντας επομένως αυτή την τακτική επεκτείνουμε το πλήθος των peers που συμμετέχουν στη διαδικασία, χωρίς να μειώνουμε την ποιότητα και εγκυρότητα των προσφερομένων αποτελεσμάτων.

Η φάση του μετασχηματισμού δέχεται ως είσοδο τις αντιστοιχίσεις που δημιουργήθηκαν και την αρχική SQL επερώτηση και δίνει ως αποτέλεσμα τις μετασχηματισμένες επερωτήσεις. Ο RDPeer χρησιμοποιώντας την αρχική επερώτηση ως οδηγό αντικαθιστά με βάση τις αντιστοιχίσεις τα στοιχεία της επερώτησης (query-elements) με τα στοιχεία του σχήματος κάθε peer (peer-elements). Στοιχεία της επερώτησης που δεν περιλαμβάνονται σε ένα σύνολο από αντιστοιχίσεις, τα οποία αναφέρονται σε ένα συγκεκριμένο peer, δε θα συμπεριληφθούν στην επερώτηση που δημιουργείται. Αυτό βέβαια συνεπάγεται ότι κάποιες εκφράσεις που περιλαμβάνονται στον όρο SELECT ή στον όρο WHERE δε συμμετέχουν στη μετασχηματισμένη επερώτηση.

Για να αποφύγουμε την εξαγωγή αποτελεσμάτων, με την εκτέλεση των μετασχηματισμένων επερωτήσεων, που αποτελούν υπερσύνολο των αποτελεσμάτων της αρχικής επερώτησης εκτελούμε μια ειδική επεξεργασία στα στοιχεία του όρου WHERE. Σύμφωνα με τον αναλυτή της SQL γλώσσας που χρησιμοποιούμε ο τελεστής AND έχει μεγαλύτερη προτεραιότητα από τον τελεστή OR. Αποτέλεσμα της προτεραιότητας είναι να γίνεται αρχικά η αξιολόγηση των εκφράσεων που συνδέονται με τον τελεστή AND. Αν μια από τις δύο εκφράσεις δεν είναι έγκυρη τότε στα πλαίσια της βελτιστοποίησης αφαιρούνται από τον όρο WHERE και οι δύο εκφράσεις, χωρίς να παραβιάζεται η ορθότητα της επερώτησης που παράγεται. Αντίθετα, όταν έχουμε δύο εκφράσεις που συνδέονται με τον τελεστή OR, τότε η μη εγκυρότητα της μιας δε συνεπάγεται την αφαίρεση και των δύο από τον όρο WHERE. Στην περίπτωση που στον όρο WHERE μια μετασχηματισμένη επερώτηση αντίθετα με την αρχική δεν έχει καμία έκφραση, τότε απορρίπτεται ως ανεπαρκής, γιατί διαφορετικά η σχέση που θα παραγόταν από την εκτέλεσή της θα αποτελούσε υπερσύνολο αυτής που θα εξάγονταν από την αρχική επερώτηση. Η διαδικασία που ακολουθείται είναι αναδρομική και έχει σαν αποτέλεσμα η συνθήκη που θα παραχθεί να μην οδηγεί κατά την εκτέλεση της επερώτησης στη δημιουργία ενός υπερσυνόλου αποτελεσμάτων.

Βασισμένοι στο παράδειγμα της προηγούμενης ενότητας και τις αντιστοιχίσεις που δημιουργήθηκαν για τους δύο peers οι μετασχηματισμένες επερωτήσεις που παράγονται είναι οι ακόλουθες:

<b>Peer1</b>	<b>SELECT COMPANY.CITY FROM COMPANY;</b>
<b>Peer2</b>	<b>SELECT ORGANIZATION.TOWN, ORGANIZATION.NUM_WORKERS FROM ORGANIZATION;</b>

**Πίνακας 3 Παράδειγμα μετασχηματισμένων επερωτήσεων.**

Παρατηρούμε ότι η μετασχηματισμένη επερώτηση που απευθύνεται στον peer1 θα παράγει ως αποτέλεσμα μια σχέση που θα είναι υποσύνολο της σχέσης που θα παραγόταν από την αρχική. Αντίθετα ο peer2 θα δεχτεί μια μετασχηματισμένη επερώτηση όμοια με την αρχική.

Όπως είχαμε αναφέρει για να φτάσουμε στο σημείο να θεωρούμε έναν peer υποψήφιο να απαντήσει μια επερώτηση πρέπει να υπάρχει τουλάχιστον μια αντιστοίχιση για ένα στοιχείο του όρου SELECT και τουλάχιστον μια αντιστοίχιση για ένα στοιχείο του όρου FROM. Η ύπαρξη αντιστοιχίσεων που ταυτίζουν τα ονόματα πεδίων, και σχέσεων μιας SQL επερώτησης με αυτά του σχήματος ενός peer, δε συνεπάγεται και τη δημιουργία μιας μετασχηματισμένης επερώτησης. Αυτό εκπορεύεται από την αναγκαιότητα να υπάρχει μια έγκυρη συνθήκη στον όρο WHERE εφόσον υπάρχει και στην αρχική επερώτηση. Στον παρακάτω πίνακα φαίνονται οι αντιστοιχίσεις και οι μετασχηματισμένες επερωτήσεις που παράγονται δεδομένου μιας νέας SQL επερώτησης και θεωρώντας τα σχήματα των peer1, peer2, και peer3 του προηγούμενου παραδείγματος.

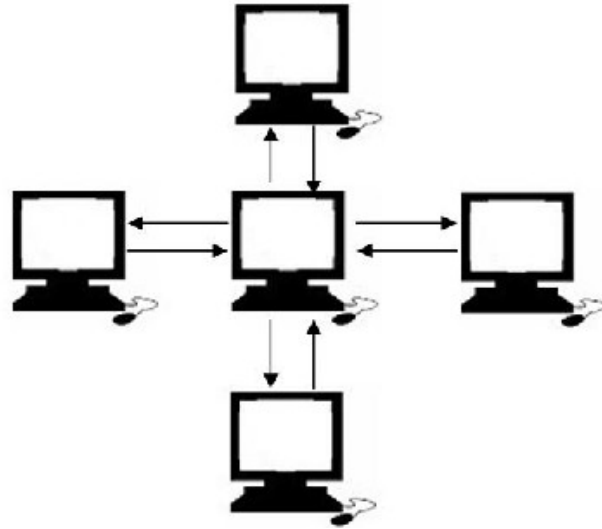
SQL query	<b>SELECT C.TOWN FROM COMPANY C, COMPANY E WHERE C.TOWN = "HRAKLIO" AND E.NUM_WORKERS &gt; 10;</b>
Mappings	<b><u>Peer1 (mappings)</u></b> \$\$: C.TOWN – C.CITY \$F: COMPANY C – COMPANY C \$F: COMPANY E – COMPANY E \$W: C.TOWN – C.CITY  <b><u>Peer2 (mappings)</u></b> \$\$: C.TOWN – C.TOWN \$F: COMPANY C – ORGANIZATION C \$F: COMPANY E – ORGANIZATION E \$W: C.TOWN – C.TOWN \$W: E.NUM_WORKERS – E.NUM_WORKERS
Transformed queries	<b><u>Peer1 (transformed query)</u></b> <b>Does not exist</b>  <b><u>Peer2 (transformed query)</u></b> <b>SELECT C.TOWN FROM ORGANIZATION C, ORGANIZATION E WHERE C.TOWN = "HRAKLIO" AND E.NUM_WORKERS &gt; 10;</b>

#### Πίνακας 4 Δημιουργία αντιστοιχίσεων και μετασχηματισμός SQL επερώτησης

Σύμφωνα με τη βελτιστοποίηση που διενεργούμε αν και δημιουργούνται αντιστοιχίσεις και για τους δύο peers και επομένως θεωρούνται υποψήφιοι να απαντήσουν την επερώτηση για τον peer1 δε δημιουργείται μετασχηματισμένη επερώτηση και κρίνεται ανεπαρκής. Αυτό συμβαίνει γιατί δε μπορεί να υπάρξει συνθήκη στον όρο WHERE. Η μορφή που έχει ο όρος WHERE της αρχικής επερώτησης είναι <expression> AND <expression>. Στην περίπτωση όμως του peer1 η έγκυρη συνθήκη που μπορεί να υπάρξει έχει τη μορφή <expression> και πιο συγκεκριμένα θα ήταν C.CITY = "HRAKLIO". Αν επιτρέπαμε τη δημιουργία της επερώτησης για τον peer1 τότε με την έλλειψη του τελεστή AND και της έκφρασης που ακολουθεί μετά από αυτόν, η εκτέλεση της επερώτησης θα επέφερε ένα σύνολο αποτελεσμάτων που θα ήταν υπερσύνολο του αρχικού. Αυτό συμβαίνει γιατί απουσιάζουν ο τελεστής και η δεύτερη έκφραση που περιορίζουν το σύνολο των πλειάδων που λαμβάνουμε κατά την εκτέλεση της επερώτησης. Ο peer1 θα μπορούσε εν τέλει να απαντήσει την επερώτηση αν ο τελεστής ήταν το OR. Τότε θα μπορούσε η συνθήκη να έχει μόνο τη μία έκφραση χωρίς τον κίνδυνο λήψης υπερσυνόλου.

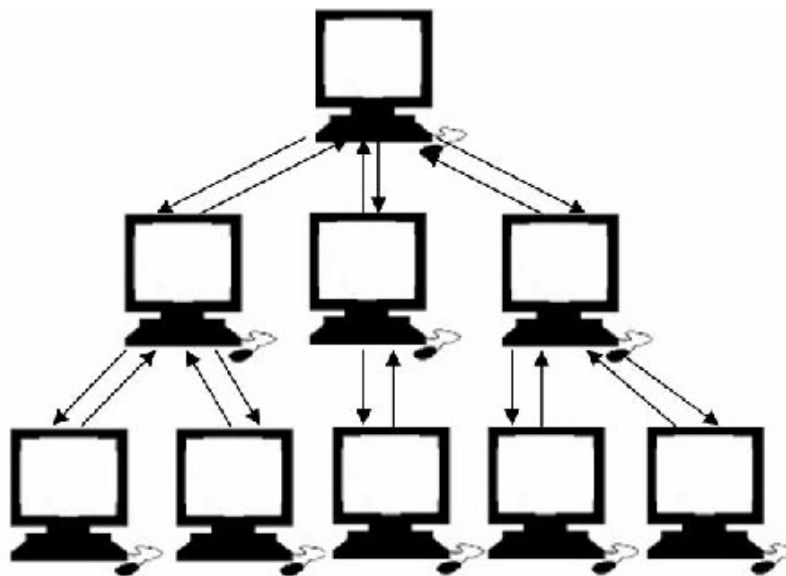
Στην εφαρμογή που έχουμε υλοποιήσει δεν έχουμε συμπεριλάβει την περίπτωση της σύνθεσης της απάντησης, δηλαδή της επερώτησης που μπορεί να απαντηθεί με τη συνεργασία περισσότερων του ενός peer. Υπάρχουν περιπτώσεις που μια SQL επερώτηση είναι δυνατόν για να απαντηθεί να απαιτείται η χρήση δύο ή περισσότερων διαφορετικών σχημάτων. Αυτό συμβαίνει όταν η SQL επερώτηση περιλαμβάνει περισσότερες της μία σχέσεις και αυτές ανήκουν σε διαφορετικούς peers. Στην περίπτωση αυτή είναι αναγκαίο να μεταφερθούν μεταξύ των peers μεγάλος όγκος δεδομένων και να συγκεντρωθούν σε ένα μόνο peer για να τεθεί η επερώτηση.

Για να μπορέσει να δημιουργηθεί ένα σύστημα που μπορεί να εξυπηρετεί τη σύνθεση της απάντησης πρέπει αρχικά να βρεθούν οι peers που συμμετέχουν στην επερώτηση και στη συνέχεια να ανατεθεί σε ένα εξ αυτών η διαδικασία συλλογής των δεδομένων που απαιτούνται από τους υπόλοιπους peers για να απαντηθεί η επερώτηση τοπικά. Η διαδικασία συλλογής των δεδομένων μπορεί να είναι είτε κεντροποιημένη είτε κατακεντρωμένη. Στα παρακάτω σχήματα φαίνονται οι δύο προσεγγίσεις, ενώ στο παράδειγμα που ακολουθεί αναλύεται ο τρόπος που πραγματοποιούνται.



Σχήμα 8 Κεντρικοποιημένο σύστημα για σύνθεση της απάντησης σε μία SQL ερώτηση

Σύμφωνα με την πρώτη προσέγγιση ένας peer αναλαμβάνει να επεξεργαστεί την αρχική ερώτηση και να δημιουργήσει τόσες ερωτήσεις όσοι και οι peers που συμμετέχουν στην ερώτηση. Στη συνέχεια όλες οι ερωτήσεις στέλνονται σε κάθε peer του οποίου το σχήμα περιλαμβάνεται στην ερώτηση και αφού επιστραφούν και αποθηκευτούν οι απαντήσεις τίθεται η αρχική SQL ερώτηση.



Σχήμα 9 Κατανεμημένο σύστημα για σύνθεση της απάντησης σε μία SQL ερώτηση

Η δεύτερη προσέγγιση είναι πιο ευέλικτη καθώς υπάρχουν ενδιάμεσοι peers που κάνουν επιπρόσθετη σύνθεση των απαντήσεων, ώστε να γίνει πιο κατανοητή η διαδικασία συλλογής των δεδομένων. Όμως, δε σημαίνει ότι είναι καλύτερη. Σε κάθε περίπτωση η προσέγγιση που πρέπει να επιλεγεί, ώστε να απαντηθεί μια επερώτηση με αξιοπιστία, αποδοτικότητα και μικρότερη επιβάρυνση του δικτύου. εξαρτάται από την επερώτηση καθώς και από το πλήθος των δεδομένων που πρέπει να μεταφερθούν.

Ας θεωρήσουμε για παράδειγμα ότι ο super-peer μιας κοινότητας έχει συλλέξει τρία σχήματα από τους peer1, peer2, και peer3. Το σχήμα του peer1 είναι COMPANY (COMPANY\_NAME, ADDRESS, CITY), το σχήμα του peer2 είναι WORKER (NAME, ADDRESS, COMPANY\_NAME) και το σχήμα του peer3 είναι EMPLOYEE (NAME, ADDRESS, COMPANY\_NAME). Ο super-peer δέχεται μια SQL επερώτηση από τον peer4 που είναι η εξής:

```
SELECT WORKER.ADDRESS, COMPANY.ADDRESS
FROM COMPANY, WORKER
WHERE WORKER.COMPANY_NAME = COMPANY.COMPANY_NAME;
```

Έχοντας τους θησαυρούς όρων ο super-peer αντιλαμβάνεται ότι υπάρχει συσχέτιση των ονομάτων WORKER και EMPLOYEE. Για να απαντηθεί η επερώτηση σύμφωνα με όσα αναφέραμε προηγούμενα πρέπει αρχικά να δημιουργηθούν τρεις επερωτήσεις οι οποίες είναι:

<b>Peer1</b>	SELECT COMPANY.ADDRESS, COMPANY.COMPANY_NAME FROM COMPANY;
<b>Peer2</b>	SELECT WORKER.ADDRESS, WORKER.COMPANY_NAME FROM WORKER;
<b>Peer3</b>	SELECT EMPLOYEE.ADDRESS, EMPLOYEE.COMPANY_NAME FROM EMPLOYEE;

**Πίνακας 5 Παράδειγμα μετασηματισμένων επερωτήσεων για την κεντρικοποιημένη σύνθεση της απάντησης**

Ο super-peer μπορεί να στείλει σε κάθε peer την επερώτηση που του αντιστοιχεί. Αφού κάθε peer εκτελέσει την επερώτηση που του αντιστοιχεί αποστέλλει τα αποτελέσματα στο super-peer, ο οποίος αφού τα αποθηκεύσει στη βάση δεδομένων που έχει το σχήμα που εμφανίζεται στην SQL επερώτηση εκτελεί την επερώτηση. Για λόγους βελτιστοποίησης θα μπορούσε να αποσταλεί η αρχική επερώτηση στο peer1 και όχι κάποια επερώτηση και επιπλέον να αναλάμβανε ο ίδιος τη συλλογή των αποτελεσμάτων

από τους άλλους δύο. Έτσι ο peer1 θα κατασκεύαζε μόνο έναν πίνακα στη βάση του για το σχήμα WORKER (ADDRESS,COMPANY\_NAME) στο οποίο θα αποθήκευε τα δεδομένα των άλλων δύο και θα εκτελούσε την αρχική επερώτηση. Με τον τρόπο αυτό θα επιτυγχάνονταν καλύτερη απόδοση του συστήματος.

Σύμφωνα με τη δεύτερη προσέγγιση θα μπορούσαν να δημιουργηθούν αρχικά οι παρακάτω επερωτήσεις.

<b>Peer1</b>	SELECT COMPANY.ADDRESS, COMPANY.COMPANY_NAME FROM COMPANY;
<b>Peer2</b>	SELECT WORKER.ADDRESS, COMPANY.ADDRESS FROM COMPANY, WORKER WHERE WORKER.COMPANY_NAME = COMPANY.COMPANY_NAME;
<b>Peer3</b>	SELECT EMPLOYEE.ADDRESS, COMPANY.ADDRESS FROM COMPANY, EMPLOYEE WHERE EMPLOYEE.COMPANY_NAME = COMPANY.COMPANY_NAME;

**Πίνακας 6 Παράδειγμα μετασχηματισμένων επερωτήσεων για την καταναμημένη σύνθεση της απάντησης**

Αφού αποσταλεί η επερώτηση στον peer1 και εκτελεστεί στέλνονται τα αποτελέσματα που παράγονται στον peer2 και στον peer3. Αφού αποθηκευτούν και εκτελεστούν οι επερωτήσεις που αντιστοιχούν σε κάθε ένα από αυτούς στέλνονται στο super-peer τα αποτελέσματα που τα συνθέτει και τα αποστέλλει στον peer που έθεσε την SQL επερώτηση.

Η περίπτωση της σύνθεσης της απάντησης που περιγράφηκε δεν υλοποιήθηκε επειδή εμφανίζονται σημαντικά μειονεκτήματα τα οποία είναι:

- Η ραγδαία αύξηση της κίνησης στο δίκτυο. Μεγάλος όγκος δεδομένων που βρίσκεται αποθηκευμένος στις βάσεις δεδομένων των peers πρέπει να μεταφερθεί μεταξύ των peers, ώστε να μπορέσει να συγκεντρωθεί σε έναν μόνο για να μπορέσει να τεθεί η επερώτηση. Όσο αυξάνονται τα σχήματα που εμπλέκονται στην επερώτηση, και τα δεδομένα που πρέπει να μεταφερθούν, τόσο περισσότερο επιβαρύνεται και η κίνηση στο δίκτυο.
- Η μείωση της απόδοσης απόκρισης του συστήματος στην επερώτηση. Η μείωση της απόδοσης οφείλεται στο μεγάλο όγκο δεδομένων που πρέπει να ανταλλαχθούν μεταξύ των peers.
- Το caching μεγάλου όγκου πληροφορίας. Οι peers θα αναγκάζονταν να αποθηκεύουν μεγάλο όγκο δεδομένων που δεν τους ανήκει, γεγονός που δεν είναι αυτονόητο για peer-to-peer σύστημα.

- Η μείωση στην απόδοση επεξεργασίας των ερωτήσεων. Οι peers που χρησιμοποιούν συνδέσμους χαμηλής χωρητικότητας για τη διασύνδεση τους με το Διαδίκτυο θα μειώνουν την απόδοση του συστήματος.
- Η δυσκολία στην επιλογή των ερωτήσεων τα οποία πρέπει να δρομολογηθούν προς κάθε peer.

## 4.5 Εκλογές

Ένας αλγόριθμος για την επιλογή μιας μοναδικής διεργασίας, η οποία καλείται να διαδραματίσει ένα ιδιαίτερο ρόλο, ονομάζονται αλγόριθμος εκλογής. Ο αλγόριθμος εκλογής έχει ως στόχο την επιλογή μιας διεργασίας  $p_i$ ,  $i = 1, 2, \dots, N$ , η οποία είναι απαραίτητο να χρησιμοποιεί την κρίσιμη περιοχή και να αποτελεί τον εξυπηρετητή για τις υπόλοιπες. Βασικό είναι επίσης ότι όλες οι διεργασίες θα συμφωνούν στην επιλογή. Εάν η διεργασία που παίζει το ρόλο του εξυπηρετητή επιθυμεί να αποσυρθεί, τότε μια νέα εκλογή απαιτείται για να πραγματοποιηθεί η αντικατάσταση.

Μια διεργασία διενεργεί εκλογές αν αρχικοποιεί τον αλγόριθμο εκλογής. Μία μοναδική διεργασία δε μπορεί να διεξάγει περισσότερες από μια εκλογές σε μια δεδομένη χρονική στιγμή, αλλά γενικότερα οι  $N$  διεργασίες μπορούν να διεξάγουν  $N$  ταυτόχρονες εκλογές. Η πιο σημαντική απαίτηση είναι ότι η επιλεγείσα από τις εκλογές διεργασία πρέπει να είναι μοναδική, ακόμα και αν πολλές διεργασίες διενεργούν εκλογές συγχρόνως. Για παράδειγμα δύο peers μπορούν να αντιληφθούν συγχρόνως την απουσία του super-peer και να ξεκινήσουν τη διεξαγωγή των εκλογών.

Επομένως, σε κάθε peer-to-peer σύστημα στο οποίο υπάρχει η έννοια του συντονιστή, δηλαδή του super-peer είναι αναγκαία η ύπαρξη ενός αλγορίθμου εκλογών, μέσω του οποίου θα μπορεί να εκλέγεται ο νέος αντικαταστάτης στην περίπτωση που αποσυρθεί ο υπάρχων. Στην εφαρμογή που υλοποιήσαμε δε χρησιμοποιήσαμε κανένα από τους δύο πιο γνωστούς αλγορίθμους εκλογών που προτείνονται στη βιβλιογραφία [41] και οι οποίοι είναι ο *ring-based election algorithm* και ο *bully algorithm*.

Ο πρώτος αλγόριθμος δεν υλοποιήθηκε γιατί θεωρεί ότι οι διεργασίες έχουν μια εκ των προτέρων γνώση η μία της άλλης, και επιπλέον ορίζεται η γειτνίαση μεταξύ των διεργασιών, αντίθετα με το σύστημα που υλοποιήσαμε όπου μια διεργασία δε γνωρίζει εκ των προτέρων τις υπόλοιπες που συμμετέχουν στην ίδια κοινότητα και δεν έχει γνώση για τις γειτονικές προς αυτή διεργασίες. Μια ακόμα σημαντική διαφορά που μας ώθησε στη μη υιοθέτηση του *ring-based election algorithm* είναι η μη ανεκτικότητά του στις αποτυχίες. Ο *bully algorithm* από την άλλη δεν υιοθετήθηκε, επειδή θεωρεί ότι κάθε διεργασία γνωρίζει σχετικά με το ποιες διεργασίες έχουν τα μεγαλύτερα αναγνωριστικά. Η έννοια όμως του αναγνωριστικού στο RDPeer με βάση το οποίο εκλέγεται ο νέος



συντονιστής είναι μια μεταβλητή, η οποία για κάθε διεργασία δεν είναι σταθερή αλλά μεταβάλλεται ανάλογα με το πλήθος των SQL επερωτήσεων που θέτει μια διεργασία, ενώ επιπλέον όπως προαναφέραμε δεν υπάρχει εκ των προτέρων γνώση των άλλων διεργασιών. Συνεπώς, για να επιτυγχάναμε κάθε διεργασία να γνωρίζει τα αναγνωριστικά που κατέχουν οι υπόλοιπες κάθε χρονική στιγμή θα έπρεπε να επιβαρύνουμε σημαντικά την κίνηση στο δίκτυο.

Στο RDPeer σχεδιάσαμε και υλοποιήσαμε έναν κατανεμημένο αλγόριθμο εκλογής ο οποίος είναι ανεκτικός σε αποτυχίες και δεν προϋποθέτει από μια διεργασία τη γνώση των υπολοίπων. Η απόδοση του αλγορίθμου εκλογής, δηλαδή η χρήση του δικτύου που απαιτεί και η χρονική διάρκεια περάτωσής του, δεν αποτέλεσε το κύριο κριτήριο στη σχεδίαση και υλοποίηση, όσο η ορθή τελικά εκλογή του νέου συντονιστή. Οι εκλογές λαμβάνουν χώρα μεταξύ των peers όταν τεθεί από κάποιον peer μια SQL επερώτηση και διαπιστωθεί ότι για οποιοδήποτε λόγο έχει πάψει να υπάρχει ο super-peer μέσα στην κοινότητα. Αυτό μπορεί να συμβεί είτε λόγω αποχώρησης του super-peer είτε επειδή παύει ξαφνικά να λειτουργεί. Συνεπώς, είναι δυνατόν μια κοινότητα να βρεθεί ένα χρονικό διάστημα χωρίς την ύπαρξη ενός super-peer και παρόλα αυτά να μην έχει καλεστεί ο αλγόριθμος εκλογής, εξαιτίας της μη δημιουργίας μιας SQL επερώτησης από κάποιον peer.

Στον τρόπο λειτουργίας του RDPeer είχαμε αναφέρει ότι όταν ένας peer δημιουργήσει μια κοινότητα και εισέλθει σ' αυτή αυτόματα ανακηρύσσεται super-peer. Από εκείνη τη στιγμή και μετά κάθε peer που θέλει να εισέλθει μέσα στην κοινότητα πρέπει απαραίτητα να αναζητήσει και να βρει τον super-peer, διαφορετικά δεν μπορεί να συμμετάσχει στην κοινότητα. Επομένως, όλοι οι peers μιας κοινότητας γνωρίζουν τον ίδιο super-peer, ενώ σε περίπτωση που πάψει να υφίσταται ο super-peer και μέχρι να εκλεγεί ο νέος, κανένας peer δε μπορεί να ενταχθεί στην κοινότητα.

Η αποστολή μίας SQL επερώτησης γίνεται μέσω ενός σωλήνα που δημιουργεί ένας peer με τον RequestServer που διαθέτει ο super-peer. Ο server είναι το module του super-peer που αναλαμβάνει να δεχτεί την επερώτηση και να την εξυπηρετήσει. Στην περίπτωση που ένας peer δεν καταφέρει να εγκαθιδρύσει τη σύνδεση με τον server τότε πραγματοποιεί μια δεύτερη προσπάθεια σύνδεσης, η οποία αν είναι και αυτή αποτυχημένη τότε ο peer αναλαμβάνει να συνδεθεί με όλους τους peers της κοινότητας για να τους ανακοινώσει τη διενέργεια εκλογών. Κάθε peer διαθέτει έναν πολυνηματικό εξυπηρετητή (ElectionServer) για να δέχεται τις αιτήσεις που αφορούν τη διενέργεια των εκλογών. Συνεπώς, αντιλαμβανόμενος ένας peer την απουσία του super-peer και αφού αναζητήσει τους ενεργούς peers της κοινότητας στέλνει σ' αυτούς ένα μήνυμα εκλογών.

Πριν αποστείλει το μήνυμα όμως υπολογίζει ποιος peer πρέπει να αναλάβει τη διεξαγωγή των εκλογών. Ο peer που αναλαμβάνει τη διεξαγωγή των εκλογών είναι μοναδικός κάθε φορά και υπολογίζεται με βάση το μοναδικό αναγνωριστικό που φέρει κάθε peer. Η πλατφόρμα JXTA αναθέτει σε κάθε peer του δικτύου μια μοναδική

διεύθυνση, δηλαδή ένα αναγνωριστικό, που αποτελείται από ένα δεκαεξαδικό αριθμό. Ο peer που αντιλαμβάνεται την απουσία του super-peer ύστερα από επεξεργασία και σύγκριση επιλέγει τον peer που διαθέτει το μεγαλύτερο αναγνωριστικό και το δεύτερο μεγαλύτερο αναγνωριστικό. Ο peer που διαθέτει το δεύτερο μεγαλύτερο αναγνωριστικό είναι ουσιαστικά ο αναπληρωματικός διαχειριστής σε περίπτωση αποτυχίας του πρώτου. Επομένως κάθε φορά ένας peer θα είναι αυτός που θα επωμίζεται το ρόλο του διαχειριστή των εκλογών.

Το μήνυμα που αποστέλλεται αποτελείται από το αναγνωριστικό, όνομα του peer που στέλνει το μήνυμα, το αναγνωριστικό, όνομα του peer που θα διεξάγει τις εκλογές και του αναπληρωματικού διαχειριστή και τέλος τη λέξη elections. Η αποστολή των μηνυμάτων γίνεται με την τεχνική του *basic multicast* [8] και ο πρώτος peer που θα λάβει το μήνυμα θα είναι ο νέος διαχειριστής. Η τεχνική αυτή ακολουθήθηκε, ώστε να αποφύγουμε τυχόν αποτυχία του peer, που αποστέλλει τα μηνύματα, κατά τη διάρκεια της επικοινωνίας με όλους τους υπόλοιπους. Κάθε peer που λαμβάνει το μήνυμα αποκρίνεται με μια βεβαίωση λήψης. Η διαδικασία αυτή διενεργείται για να αποφευχθούν οι περιπτώσεις ασυνέπειας. Επομένως, ακόμα και αν αντιληφθούν ταυτόχρονα την απουσία του super-peer περισσότεροι του ενός peers όλοι θα επιλέξουν τον ίδιο διαχειριστή των εκλογών. Με τον τρόπο αυτό αποκλείεται να διενεργηθούν παράλληλες εκλογές μέσα σε μια κοινότητα και να εκλεγούν δύο peers στο ρόλο του super-peer. Ταυτόχρονα μειώνονται σημαντικά οι πόροι που απαιτούνται.

Κάθε peer που λαμβάνει ένα μήνυμα εκλογών καθαρίζει την cache που διαθέτει από παλιότερες δημοσιοποιήσεις. Αυτές που διαγράφονται είναι οι δύο δημοσιοποιήσεις που είχε εκδώσει ο super-peer, δηλαδή η δημοσιοποίηση του σωλήνα στην οποία περιγραφόταν η διεύθυνση και το port μέσω του οποίου θα μπορούσαν να επικοινωνήσουν με τον RequestServer, και η δημοσιοποίηση που περιγράφει τον super-peer. Συνακόλουθα, διαγράφονται και οι δημοσιοποιήσεις όλων των peers της κοινότητας. Τέλος, κάθε peer που λαμβάνει το μήνυμα των εκλογών αφού το αποσυνθέσει συγκρίνει το αναγνωριστικό του διαχειριστή και του αναπληρωματικού διαχειριστή των εκλογών με το δικό του. Εάν υπάρχει ταύτιση με το αναγνωριστικό του διαχειριστή τότε αναλαμβάνει να διεξάγει τις εκλογές, ενώ η ύπαρξη ταύτισης με το αναγνωριστικό του αναπληρωματικού διαχειριστή υποδεικνύει το νέο ρόλο που έχει ο συγκεκριμένος peer και είναι η «αστυνόμηση» του διαχειριστή.

Η πρώτη ενέργεια που επιτελεί ο διαχειριστής των εκλογών είναι να ανακαλύψει τους ενεργούς peers της κοινότητας. Η ανακάλυψη γίνεται μέσω της ανταλλαγής ring pong μηνυμάτων. Στη συνέχεια για να υπολογίσει τον peer που πρέπει να εκλεγεί επικοινωνεί με τους υπόλοιπους peers για να τους ζητήσει τον αριθμό των SQL ερωτήσεων που έχουν θέσει. Ο αριθμός των ερωτήσεων είναι μια μετρική ενδεικτική για την κίνηση που παράγει ένας peer στο δίκτυο καθώς και για τους πόρους που ξοδεύονται για αυτόν από τον εκάστοτε super-peer. Το μήνυμα που επιστρέφεται στο διαχειριστή περιέχει το

όνομα, αναγνωριστικό του peer που στέλνει το μήνυμα και τον αριθμό των επερωτήσεων. Ο διαχειριστής των εκλογών συλλέγοντας τους αριθμούς των επερωτήσεων κάθε peer, τους συγκρίνει και επιλέγει για νέο super-peer αυτόν που έθεσε τις περισσότερες επερωτήσεις. Στην περίπτωση που έχουμε περισσότερους του ενός peer να ισοβαθμούν τότε επιλέγεται αυτός που έχει το μεγαλύτερο αναγνωριστικό με διαδικασία ανάλογη αυτής που διενεργήθηκε για την επιλογή του διαχειριστή των εκλογών.

Ο διαχειριστής αφού βρει το νέο super-peer αποστέλλει μήνυμα προς όλα τα μέλη της κοινότητας, χρησιμοποιώντας την τεχνική basic multicast κατά την οποία ο πρώτος peer που θα λάβει το μήνυμα θα είναι ο super-peer. Το μήνυμα περιέχει το όνομα και το αναγνωριστικό του νέου super-peer. Κάθε peer που λαμβάνει το μήνυμα αποκρίνεται με μια βεβαίωση λήψης για να πιστοποιήσει τη γνώση του αποτελέσματος των εκλογών. Ο αναπληρωματικός διαχειριστής έχει μόνο μια αρμοδιότητα η οποία είναι να ανταλλάσσει περιοδικά ring ring μηνύματα με το διαχειριστή. Σε περίπτωση που σταλεί μήνυμα ring και δε ληφθεί απάντηση η διαδικασία επαναλαμβάνεται. Ύστερα από τρεις ανεπιτυχείς προσπάθειες ο διαχειριστής θεωρείται ότι έχει αποτύχει και ο αναπληρωματικός διαχειριστής αναλαμβάνει τη διεξαγωγή των εκλογών. Το χρονικό όριο μεταξύ των ανεπιτυχών προσπαθειών επικοινωνίας με το διαχειριστή δεν είναι σταθερό, αλλά ορίζεται τυχαία και είναι πολλαπλάσιο του δευτερολέπτου.

Κάθε ένας από τους peers της κοινότητας που λαμβάνει το μήνυμα των αποτελεσμάτων το αποσυνθέτει και συγκρίνει το αναγνωριστικό του νέου super-peer με το δικό του. Με τον τρόπο αυτό ένα από τα μέλη της κοινότητας αντιλαμβάνεται το νέο ρόλο που καλείται να έχει μέσα στην κοινότητα. Οι άμεσες ενέργειες που διεξάγει είναι:

- Να αρχικοποιήσει το module RequestServer που διαθέτει, ώστε να αρχίσει να δέχεται και να εξυπηρετεί SQL επερωτήσεις.
- Να δημιουργήσει και να εκδώσει μια δημοσιοποίηση η οποία τον χαρακτηρίζει ως super-peer για τη συγκεκριμένη κοινότητα και την οποία πρέπει υποχρεωτικά να κατέχει κάθε μέλος της κοινότητας.
- Να εκδώσει μια δημοσιοποίηση σωλήνα. Η κοινοποίηση αυτή περιγράφει ένα κανάλι επικοινωνίας και χρησιμοποιείται από την υπηρεσία σωλήνωσης για να δημιουργήσει τα σχετιζόμενα σημεία κατάληξης εισόδου, εξόδου της σωλήνωσης. Κάθε δημοσιοποίηση σωλήνα περιέχει ένα προαιρετικό συμβολικό αναγνωριστικό, ένα τύπο διασωλήνωσης και ένα μοναδικό αναγνωριστικό διασωλήνωσης. Με τη χρήση αυτή της δημοσιοποίησης θα μπορεί κάθε μέλος της κοινότητας να συνδέεται στο super-peer και να του θέτει επερωτήσεις. Η εύρεση της συγκεκριμένης δημοσιοποίησης είναι υποχρεωτική από κάθε μέλος της κοινότητας.

Τα υπόλοιπα μέλη της κοινότητας με τη σειρά τους χρησιμοποιώντας το πρωτόκολλο ανακάλυψης του JXTA κάνουν αναζήτηση των δημοσιοποιήσεων που έχει εκδώσει ο

νέος super-peer. Στη συνέχεια ο peer που αρχικά είχε ανακαλύψει την απουσία του super-peer στέλνει την SQL επερώτηση που εκκρεμεί στον RequestServer, ώστε να εξυπηρετηθεί.

Στον αλγόριθμο εκλογής υπάρχουν τρία κρίσιμα σημεία. Το πρώτο είναι η αποτυχία του αρχικού peer που αντιλαμβάνεται την απουσία του super-peer, κατά τη διάρκεια εύρεσης του διαχειριστή των εκλογών. Στην περίπτωση αυτή η κοινότητα παραμένει χωρίς νέο συντονιστή και το σύστημα προσμένει κάποιο νέο peer να ανιχνεύσει την απουσία του super-peer, έτσι ώστε να διεξαχθούν ξανά εκλογές. Η αποτυχία εύρεσης νέου super-peer δεν επηρεάζει το σύστημα καθώς είναι επιτακτική η ανάγκη της εκλογής όταν έχει τεθεί μια επερώτηση. Αποτυγχάνοντας αυτός που θέτει την επερώτηση συνεπάγεται ότι το σύστημα δε θα επηρεαστεί από αποτυχία του αλγορίθμου εκλογής. Το δεύτερο κρίσιμο σημείο είναι η αποτυχία του διαχειριστή των εκλογών είτε κατά τη διάρκεια υπολογισμού του νέου super-peer, είτε κατά τη διάρκεια αποστολής των μηνυμάτων. Εάν αυτό συμβεί τότε τη διενέργεια των εκλογών αναλαμβάνει ο αναπληρωματικός διαχειριστής. Η μόνη περίπτωση όπου ο αλγόριθμος είναι μη ανεκτικός σε αποτυχία είναι όταν έχουμε αποτυχία τόσο του διαχειριστή, όσο και του αναπληρωματικού του. Τότε δε θα αναδειχθεί καινούργιος super-peer και κυρίως θα μείνει αναπάντητη η SQL επερώτηση.

## 4.6 Τεχνολογίες του συστήματος RDPeer

Οι τεχνολογίες που χρησιμοποιήθηκαν κατά την υλοποίηση του συστήματος επιτρέπουν γρήγορη ανάπτυξη, επεκτασιμότητα (προσθήκη νέων *modules*) και ευελιξία σε τυχόν αλλαγές.

Όλη η εφαρμογή είναι υλοποιημένη στη γλώσσα προγραμματισμού Java [19]. Ένας από τους λόγους που προτιμήθηκε η Java είναι επειδή μας παρέχει αρκετά εργαλεία τα οποία σε μια άλλη γλώσσα προγραμματισμού όπως η C θα ήταν αρκετά χρονοβόρα να υλοποιηθούν. Επιπλέον είναι ανεξάρτητη πλατφόρμας, ενώ σημαντικό ήταν το γεγονός ότι υποστηριζόταν από την πλατφόρμα JXTA. Η πλατφόρμα JXTA χρησιμοποιήθηκε γιατί μας προσφέρει τα δομικά στοιχεία για την ανάπτυξη της peer-to-peer εφαρμογής.

Όλα τα δεδομένα που λαμβάνει ο RDPeer από τους servers των online λεξικών είναι σε μορφή HTML. Ο κώδικας HTML είναι αυτός από τον οποίο και παράγονται οι ιστοσελίδες που υπάρχουν στο Διαδίκτυο. Μετά από ανάλυση αυτών των δεδομένων ο RDPeer κρατά μόνο την πληροφορία που χρειάζεται (συνώνυμα, υπερώνυμα) για να δημιουργήσει τους θησαυρούς όρων.

Για τη δημιουργία των XML [44] αρχείων τα οποία περιέχουν τους θησαυρούς όρων χρησιμοποιήθηκε ο αναλυτής της XML γλώσσας (XMLParser) που περικλύει η Java και ονομάζεται DOM. Η XML χρησιμοποιήθηκε γιατί:

- Η XML είναι το κύριο δομικό στοιχείο που χρησιμοποιεί η πλατφόρμα για την ανταλλαγή πληροφοριών. Όλες οι δημοσιοποιήσεις υλοποιούνται με τη μορφή XML εγγράφων.
- Η XML είναι ουδέτερη γλώσσα. Κάθε γλώσσα προγραμματισμού που μπορεί να χειριστεί κείμενο είναι ικανή να αναλύσει και να σχηματοποιήσει XML δεδομένα.
- Η XML είναι απλή. Η XML χρησιμοποιεί σήμανση κειμένου για να δομήσει τα δεδομένα με τον ίδιο τρόπο που η HTML δομεί έγγραφο κειμένου για να εμφανιστούν σε φυλλομετρητές του ιστού. Η απλότητα της XML έχει σαν πλεονέκτημα ότι οι μηχανικοί λογισμικού μπορούν να την καταλάβουν ευκολότερα.
- Η XML είναι αυτό-περιγραφική. Ένα XML έγγραφο αποτελείται από δεδομένα τα οποία έχουν δομηθεί χρησιμοποιώντας ετικέτες μεταδεδομένων και ιδιότητες για να περιγραφούν τα δεδομένα.
- Η XML είναι επεκτάσιμη. Αντίθετα με την HTML, η XML επιτρέπει στους συγγραφείς του εγγράφου να ορίσουν το δικό τους σύνολο από ετικέτες σήμανσης για να δομήσουν τα δεδομένα.
- Η XML είναι πρότυπο. Το *World Wide Web Consortium* είναι υπεύθυνο για τη διατήρηση και υποστήριξη του XML προτύπου και έχει ευρέως υιοθετηθεί από όλες τις περιοχές της βιομηχανίας υπολογιστών.

Μία άλλη τεχνολογία που χρησιμοποιήθηκε στο RDPeer είναι το πρωτόκολλο SOAP (Simple Object Access Protocol) [39]. Το SOAP είναι ένα απλό πρωτόκολλο, βασισμένο στη ιδέα ότι σε κάποιο σημείο σε μια καταναμεμημένη αρχιτεκτονική, θα είναι απαραίτητη η ανταλλαγή πληροφοριών. Επιπρόσθετα, σε ένα σύστημα που είναι υπερφορτωμένο και έχει βαριές διεργασίες, αυτό το πρωτόκολλο είναι ελαφρύ, και απαιτεί ένα ελάχιστο ποσό επιπλέον κόστους. Επιπλέον, προσφέρει την ανταλλαγή της πληροφορίας πάνω από το HTTP πρωτόκολλο, γεγονός που επιτρέπει τη διάσχιση του firewall. Το SOAP σε αντίθεση με το XML-RPC [45] προσφέρει τη δυνατότητα του ορισμού καινούργιων τύπων δεδομένων. Στο RDPeer χρησιμοποιήσαμε το SOAP για κλίσεις απομακρυσμένων μεθόδων σε κάποιο server και όχι για επικοινωνία μέσω μηνυμάτων, στην οποία ένας πελάτης στέλνει κομμάτια πληροφορίας σε ένα server. Επομένως, ένας peer επικαλείται μια απομακρυσμένη μέθοδο ενός server οπουδήποτε και λαμβάνει μια απάντηση. Το SOAP ενεργεί απλά σαν ένα περισσότερο επεκτάσιμο XML-RPC σύστημα, το οποίο επιτρέπει καλύτερη διαχείριση των λαθών και το πέρασμα σύνθετων τύπων διαμέσου του δικτύου. Η χρήση του πρωτοκόλλου έγινε γιατί κάθε peer έπρεπε να διαθέτει μια

υπηρεσία την οποία θα μπορούσαν όλοι να καλούν για να πάρουν απαντήσεις στην SQL επερώτηση που έθεταν.

Για να είναι ικανός κάθε peer να διαθέτει ένα σύνολο από SOAP modules τα οποία να αποτελούν συστατικά server, και τα οποία να καλούν όλοι οι υπόλοιποι μέσω SOAP-RPC, χρειάζεται ένα servlet-engine. Κατά τη διαδικασία της ανάπτυξης του συστήματος χρησιμοποιήθηκε ο Tomcat application server. Αυτός είναι γραμμένος εξ' ολοκλήρου σε Java και ειδικότητά του είναι να εκτελεί τα java servlets. Έχει ικανοποιητική απόδοση στην εκτέλεση των servlets αλλά υστερεί στην απόδοση όταν πρόκειται για στατικές σελίδες.

Στην εφαρμογή που υλοποιήθηκε χρησιμοποιήθηκε ένας αναλυτής της γλώσσας SQL (SQLParser) για να είναι δυνατή η αποσύνθεση μιας SQL επερώτησης και η λήψη από αυτή των ονομάτων των πεδίων, σχέσεων καθώς και των εκφράσεων. Ο αναλυτής που χρησιμοποιήθηκε ονομάζεται ZQL [49] και είναι γραμμένος σε Java.

Κάθε peer του συστήματος είναι αναγκαίο να διαθέτει δεδομένα τα οποία να διαχειρίζεται. Για το σκοπό αυτό επιλέχτηκε η βάση δεδομένων Oracle9i [31], στην οποία κάθε peer είναι ικανός να κατασκευάζει ή να διαγράφει πίνακες, να εισάγει, διαγράφει δεδομένα. Επιπλέον, παρέχεται η δυνατότητα να κάνει επερωτήσεις στη βάση δεδομένων που έχει δημιουργήσει και να εξάγει το σχήμα της βάσης. Για να μπορεί να συνδέεται στη βάση δεδομένων και να εκτελεί αυτές τις ενέργειες ένας peer χρησιμοποιήθηκαν οι βιβλιοθήκες JDBC της Java, και ένας οδηγός που παρέχει η ORACLE.

## 5 Συμπεράσματα της εργασίας και μελλοντικές επεκτάσεις

Καθώς το Διαδίκτυο συνεχώς μεγαλώνει σε μέγεθος, αυξάνονται και τα δεδομένα τα οποία διαμοιράζονται μεταξύ των χρηστών. Για το λόγο αυτό, τα peer-to-peer συστήματα έχουν αυξηθεί δραματικά τα τελευταία χρόνια, καθώς προσφέρουν χαμηλό κόστος διαμοιρασμού της πληροφορίας, αυτονομία, κλιμακοσιμότητα, έλλειψη κεντρικού ελέγχου, και μυστικότητα. Επιπλέον, κατανέμουν το κύριο κόστος του διαμοιρασμού των δεδομένων διαμέσου των κόμβων του δικτύου με αποτέλεσμα να μπορούν να κλιμακώνονται χωρίς την ανάγκη ισχυρών εξυπηρετητών. Σε ένα peer-to-peer κατανεμημένο σύστημα, ένας μεγάλος αριθμός από κόμβους μπορούν να τοποθετηθούν μαζί για να μοιραστούν τους πόρους, τις πληροφορίες και τις υπηρεσίες. Οι κόμβοι, οι οποίοι μπορούν και να καταναλώνουν αλλά και να συνεισφέρουν δεδομένα και υπηρεσίες, έχουν τη δυνατότητα να συμμετέχουν ή να εγκαταλείπουν το peer-to-peer δίκτυο ανά πάσα χρονική στιγμή, οδηγώντας έτσι σε ένα πραγματικά δυναμικό και *ad-hoc* περιβάλλον.

Παρά το γεγονός ότι τα peer-to-peer δίκτυα έχουν ένα μεγάλο εύρος από εφαρμογές, υπάρχουν ακόμα κάποια σημεία τα οποία δεν έχουν εξερευνηθεί πολύ. Ένα από αυτά είναι η συσχέτιση των peer-to-peer συστημάτων και των τεχνολογιών των βάσεων δεδομένων. Στα πρόσφατα συστήματα διαμοιρασμού των δεδομένων αγνοούνται σημαντικά οι έννοιες των δεδομένων. Στην παρούσα εργασία RDPeer παρουσιάσαμε ένα peer-to-peer σύστημα για κατανεμημένο διαμοιρασμό των δεδομένων. Ο διαμοιρασμός των δεδομένων, που βρίσκονται στη σχεσιακή βάση δεδομένων κάθε peer, γίνεται χωρίς τη γνώση ενός συνολικού σχήματος. Ο RDPeer επιτρέπει στο χρήστη να συμμετέχει σε κοινότητες από peers, μέσα στις οποίες μπορεί να θέτει SQL αιτήματα και να παίρνει απαντήσεις από τους peers που έχουν σχετικά με την επερώτηση δεδομένα.

Ο RDPeer προσπαθεί να επιλύσει κάποιες σημαντικές διαστάσεις που επηρεάζουν το πρόβλημα του διαμοιρασμού σχεσιακών δεδομένων με τη δυναμική δημιουργία των super-peers, δηλαδή τη χρήση ιεραρχικής οργάνωσης των peers. Οι super-peers συμβάλλουν στην επεκτασιμότητα του διαμοιρασμού γνώσης, στην αξιοπιστία του συστήματος, και στην καλύτερη απόδοση της αναζήτησης δεδομένων.

Επιπλέον, με τη χρήση μεταδεδομένων, τα οποία περιγράφουν τα δεδομένα που βρίσκονται αποθηκευμένα σε κάθε βάση δεδομένων και τα οποία δημιουργούνται από κάθε κόμβο του δικτύου, σε συνδυασμό με την εφαρμογή της γλωσσολογικής προσέγγισης επιτύχαμε τη δημιουργία αντιστοιχίσεων, στο επίπεδο του super-peer, μεταξύ των σχημάτων των peers και των SQL επερωτήσεων. Οι αντιστοιχίσεις δημιουργούνταν, ώστε ο super-peer να διενεργεί μετασχηματισμό των SQL επερωτήσεων, με σκοπό να μην γίνεται το download της πληροφορίας κεντροποιημένα αλλά με peer-to-peer τρόπο.

Έτσι ο εκάστοτε peer μπορεί να συνδεθεί απευθείας σε κάθε ένα peer για τον οποίο έχει δημιουργηθεί μια SQL επερώτηση, να υποβάλλει την επερώτηση και να λάβει τα αποτελέσματα. Η δημιουργία αντιστοιχίσεων μεταξύ των επερωτήσεων και των σχημάτων και όχι μεταξύ των σχημάτων είχε σαν αποτέλεσμα την πληρέστερη, εγκυρότερη, και αποδοτικότερη αντιμετώπιση στην ανακάλυψη των πληροφοριών που ζητούσαν οι peers μέσω των SQL επερωτήσεων που έθεταν.

Για την υλοποίηση του RDPeer χρησιμοποιήσαμε την πλατφόρμα JXTA. Η πλατφόρμα JXTA πετυχαίνει αναμφίβολα να λύσει το εύρος των συσκευών που μπορούν να είναι συνδεδεμένες σε ένα peer-to-peer δίκτυο, αφού μια οποιαδήποτε συσκευή επιτρέπεται να λειτουργεί σαν κόμβος του δικτύου, και παράλληλα προσφέρει όλα τα στοιχεία που απαιτούνται για να υλοποιηθεί μια peer-to-peer εφαρμογή. Επιπρόσθετα, η χρησιμοποίηση της XML είχε σαν αποτέλεσμα ένα σχήμα για δομημένα δεδομένα το οποίο είναι εύκολα κατανοητό, καλά υποστηριζόμενο, και εύκολα προσαρμόσιμο σε μεγάλη ποικιλία από μεταφορές.

Με την χρήση της πλατφόρμας κάθε peer μπορούσε εύκολα να ανακαλύψει, και να επικοινωνήσει με τους υπόλοιπους. Επίσης, είχε τη δυνατότητα να εκδώσει τα μεταδεδομένα που θέλει, να δημιουργήσει κοινότητες στις οποίες μπορούσε να ενταχθεί ή να εγκαταλείψει. Ο RDPeer επεκτείνοντας τις λειτουργίες της πλατφόρμας JXTA επέτρεψε σε κάθε κόμβο του δικτύου να διαχειρίζεται μια βάση δεδομένων, να συνδέεται σε ηλεκτρονικά λεξικά και να δημιουργεί ένα θησαυρό όρων βασισμένο στο σχήμα της βάσης δεδομένων που κατέχει, έτσι ώστε να επιτύχει την καλύτερη περιγραφή της. Επιπλέον, παρείχε σε κάθε peer τη δυνατότητα να κάνει τις δικές του δημοσιοποιήσεις και να φέρει ένα web service για την καλύτερη εξυπηρέτηση των επερωτήσεων των υπολοίπων μελών του δικτύου.

Το σύστημά που υλοποιήθηκε είναι υβριδικό peer-to-peer σύστημα στην κορυφή του JXTA όπου κάθε peer είναι υλοποιημένος με βάση το μοντέλο πελάτη / εξυπηρετητή. Η επιλογή της υβριδικής αρχιτεκτονικής πρόσφερε τη δυνατότητα της αποδοτικότερης αναζήτησης και δημιουργίας αντιστοιχίσεων με τη μικρότερη επιβάρυνση στην κίνηση του δικτύου.

Στα peer-to-peer συστήματα που υπάρχει η έννοια του συντονιστή είναι απαραίτητη η ύπαρξη ενός αλγορίθμου εκλογής. Για το λόγο αυτό, υλοποιήσαμε έναν αλγόριθμο εκλογής ο οποίος συμβάλει στην αξιοπιστία, συνέπεια και διατήρηση της αρχιτεκτονικής του peer-to-peer συστήματος. Επιπλέον, οι εκλογές σχεδιάστηκαν και υλοποιήθηκαν, έτσι ώστε να επιτευχθεί η καλύτερη μελλοντική απόδοση του συστήματος σε ότι αφορά το πλήθος των μηνυμάτων που ανταλλάσσονται.

Μια μελλοντική έκδοση του συστήματος θα μπορούσε να περιλαμβάνει τη σχεδίαση και υλοποίηση ενός super-peer δικτύου, έτσι ώστε να μην υπάρχει ο φόρτος εργασίας που παρατηρείται όταν για κάθε σύμπλεγμα από peers έχουμε μόνο ένα συντονιστή. Συνεπώς, αντί σε μια κοινότητα από peers να ανατίθενται μόνο ένας super-peer όπως



υλοποιήθηκε στην παρούσα εργασία θα μπορούσε να σχεδιαστεί και υλοποιηθεί ένας αλγόριθμος για την ύπαρξη περισσότερων super-peers, ο αριθμός των οποίων θα ορίζεται με βάση είτε το πλήθος των SQL επερωτήσεων που εκδίδουν οι peers, ή τον αριθμό των peers που συμμετέχουν σε μια κοινότητα. Η χρήση περισσότερων του ενός super-peer μέσα σε μια κοινότητα θα πρέπει να συνδυαστεί με τεχνικές συντονισμού και συνεργασίας μεταξύ τους, ώστε να μπορεί να απαντηθεί πλήρως μια επερώτηση. Εναλλακτικά, σχετικά με τον τρόπο οργάνωσης των peers θα μπορούσε να μελετηθεί η βελτίωση της απόδοσης ενός peer-to-peer συστήματος με την ομαδοποίηση των κόμβων, με βάση την ομοιότητα του περιεχομένου τους σε εννοιολογικά επικαλυπτόμενα δίκτυα.

Στην παρούσα εργασία αν και έχει υλοποιηθεί ο μετασχηματισμός και βελτιστοποίηση μιας SQL επερώτησης έχει παραμείνει ανοιχτό το θέμα της επερώτησης που μπορεί να απαντηθεί με συνδυασμό βάσεων δεδομένων που ανήκουν σε διαφορετικούς peers. Επιπλέον, σε μια επόμενη εργασία θα μπορούσε να εξεταστεί η χρήση κάποιας άλλης τεχνικής και όχι της γλωσσολογικής για ταύτιση των μεταδεδομένων που περιγράφονται σε κάθε σχήμα.

Στα πλαίσια του RDPeer θα μπορούσε επίσης να περιληφθεί η δυνατότητα να μπορεί ένας χρήστης να διαπιστώσει τις έννοιες πάνω στις οποίες θα θέσει τις επερωτήσεις ορίζοντας ένα κατώφλι σχετικότητας μεταξύ των εννοιών των σχημάτων που βρίσκονται στο super-peer. Με τον τρόπο αυτό, ο εκάστοτε χρήστης θα γνώριζε εκ των προτέρων το ποσοστό των peers που μπορεί να φέρουν σχετική πληροφορία με αυτή που πρόκειται να αναζητήσει.

Στο RDPeer έχει υλοποιηθεί μια υπηρεσία η οποία έχει ενσωματωθεί στη λειτουργικότητα κάθε peer και την οποία μπορεί να καλέσει απομακρυσμένα οποιοσδήποτε με παράμετρο μια SQL επερώτηση προκειμένου να εξάγει δεδομένα. Η διαδικασία αυτή είναι αδιαφανής προς τον χρήστη και αποτελεί το τελικό στάδιο απάντησης μιας επερώτησης. Θα αποτελούσε σημαντική βελτίωση η παροχή πρώτα στον χρήστη των ονομάτων των peers που έχουν την ικανότητα να απαντήσουν στην επερώτηση και η δυνατότητα του χρήστη να λάβει τα δεδομένα επιλεκτικά από τους peers που θέλει. Με την τεχνική αυτή θα επιτυγχάνονταν η μικρότερη επιβάρυνση της κίνησης του δικτύου καθώς και η ελευθερία επιλογής του χρήστη, ο οποίος μπορεί να μην έχει τη διάθεση να «βομβαρδίζεται» με πλήθος πληροφοριών. Συνακόλουθα, με τον τρόπο αυτό θα μπορεί κάθε peer να γίνει αρκετά πιο «έξυπνος», κατατάσσοντας τους υπόλοιπους peers με βάση το βαθμό προτίμησης που δείχνει ο χρήστης. Η λογική αυτή θα μπορούσε να βοηθήσει στον καλύτερο χρόνο απόκρισης του συστήματος, καθώς θα μπορεί να δρομολογηθεί μια επερώτηση εκ των προτέρων σε κάποιους peers πριν ακόμα φθάσει στο super-peer για να αναλυθεί και να μετασχηματιστεί.

## 6 Αναφορές και Βιβλιογραφία

- [1]. Aberer, K., Cudre-Mauroux, P., Hauswirth, M.: A framework for semantic gossiping. In SIGMOD RECORD, Special Section on Semantic Web and Data Management, Vol. 31, Nr. 4, 2002.
- [2]. Adar, E., and Huberman, B.: Free riding on gnutella. First Monday 5, October 2000
- [3]. Arenas, M., Kantere, V., Kementsietsidis, A., Kiringa, I., Miller, R. J., Mylopoulos, J.: The Hyperion Project: From Data Integration to Data Coordination. In SIGMOD Record, Special Issue on Peer-to-Peer Data Management, 32(3):53-58, 2003
- [4]. Baeza-Yates, R. A., Ribeiro-Neto, B. A.: Modern Information Retrieval. ACM Press/Addison-Wesley, 1999.
- [5]. Bernstein, P.A., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zaihrayeu, I.: Data management for peer-to-peer computing: A vision. In WebDB Workshop, 2002.
- [6]. Breg, F., Bik, A. and Gannon, D.: Exploiting Implicit Loop Parallelism using Multiple Multithreaded Servers in Java. December 1997.
- [7]. Brunkhorst, I., Dhraief, H., Kemper, A., Nejd, W., Wiesner, C.: Distributed Queries and Query Optimization in Schema-Based P2P-Systems. In Proceedings of the International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P), Berlin, Germany, 2003.
- [8]. Coulouris, G., Dollimore, J., Kindberg, T., editors: Distributed Systems: Concepts and Design, 3rd edition Addison-Wesley, (c) Pearson Education 2001.
- [9]. Crespo, A., Garcia-Molina H.: Semantic Overlay Networks for P2P Systems. Stanford Technical Report, 2003.
- [10]. Crespo, A., Garcia-Molina, H.: Routing indices for peer-to-peer systems. Submitted to ICDCS 2002, October 2001.
- [11]. FreeNet website. <http://freenet.sourceforge.com>.
- [12]. Garcia-Molina, H., Papakonstantinou, Y., Quass, D., Rajaraman, A., Sagiv, Y., Ullman, J., Wilson, J.: The TSIMMIS project: Integration of heterogeneous information sources. Journal of Intelligent Information Systems, 8(2), March 1997.

- [13]. Giunchiglia, F., Zaihrayeu, I.: Making peer databases interact – a vision for an architecture supporting data coordination. 6<sup>th</sup> International Workshop on Cooperative Information Agents (CIA-2002), Madrid, Spain, September 18-20, 2002.
- [14]. Gribble, S., Halevy, A., Ives, Z., Rodrig, M., Suci, D.: What can databases do for peer-to-peer? WebDB Workshop on Databases and the Web, June 2001.
- [15]. Gnutella website. <http://www.gnutelliums.com>.
- [16]. Halevy, A., Ives, Z., Mork, P., Tatarinov, I.: Piazza: Data management infrastructure for semantic web applications. In Proceedings of the 12<sup>th</sup> International Conference on World Wide Web, pages 556-567, ACM Press, 2003.
- [17]. Halevy, A., Ives, Z., Mork, P., Tatarinov, I.: Schema Mediation in Peer Data Management System. In ICDE, 2003.
- [18]. Jakarta website – Apache Project, <http://jakarta.apache.org/tomcat/> .
- [19]. Java Tutorial, A practical guide for programmers, <http://www.java.sun.com/docs/books/tutorial>
- [20]. Jini website, <http://www.jini.org>.
- [21]. JXTA website, <http://www.jxta.org>
- [22]. Kantere, V.: A rule mechanism for p2p data management. Technical report, University of Toronto, 2003. CSRG-469
- [23]. KaZaA website. <http://www.kazaa.com>.
- [24]. Kementsietsidis, A., Arenas, M., Miller, R. J.: Mapping data in peer-to-peer systems: Algorithms and issues. In ACM SIGMOD International Conference on Management of Data, 2003.
- [25]. Levy, A., Rajaraman, A., Ordille, J.: Querying Heterogeneous Information Sources Using Source Descriptions. In VLDB, 2003.
- [26]. Magnini, B., Serafini, L., Speranza, M.: Linguistic Based Matching of Local Ontologies. MeaN – 02 – AAI Workshop on Meaning Negotiation. Edmonton, Alberta, Canada, 2002.
- [27]. Miller, R. J., Haas, L. M., Hernandez, M.: Schema Mapping as Query Discovery. In VLDB, pages 77-88, Cairo, Egypt, Sept. 2000.
- [28]. Napster website. <http://www.napster.com>.

- [29]. NET website, <http://www.microsoft.com/net/>.
- [30]. Ng, W. S., Ooi, B. C., Tan, K. L., Zhou, A.: Peerdb: A p2p-based system for distributed data sharing. In Proceedings of the 19<sup>th</sup> International Conference on Data Engineering, 2003.
- [31]. ORACLE website, <http://www.oracle.com>.
- [32]. Oram, A., editor. Peer-to-Peer : Harnessing the Power of Disruptive Technologies. O' Reilly & Associates, 2001.
- [33]. Popa, L., Velegakis, Y., Miller, R. J., Hernandez, M. A., Fagin R.: Translating Web Data. In VLDB, pages 598-609, 2002.
- [34]. Rahm, E., Bernstein, P. A.: A survey of approaches to automatic schema matching. The VLDB Journal 10: 334-350, 2001.
- [35]. Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. In Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM), 2001.
- [36]. Rownstron, A., Druschel, P.: Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), 2001.
- [37]. Saroiu, S., Gummadi, P., Gribble, S.: A measurement study of peer-to-peer file sharing systems. Technical Report UW-CSE-01-06-22, University of Washington, Department of Computer Science and Engineering, July 2001.
- [38]. SETI@home website. <http://setiathome.ssl.berkeley.edu>
- [39]. SOAP website – Apache Project, <http://ws.apache.org/soap/>.
- [40]. Stoica, I., Morris, R., Karger, D., Kaashoek, F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of the ACM Special Interest Group on Data Communication (SIGCOMM), 2001.
- [41]. Tanenbaum, A. S. editor: Distributed Operating Systems. Prentice-Hall, 1995.
- [42]. Truelove K.: To the bandwidth barrier and beyond. <http://web.archive.org/web/200101107185800/dss.clip2.com/gnutella.html>, 2000.
- [43]. Web Service Definition Language website, <http://www.w3.org/TR/wsdl>.
- [44]. XML website, <http://www.w3.org/XML/>.
- [45]. XML-RPC website, <http://www.xml-rpc.com>.

- [46]. Yang, B., Garcia-Molina, H.: Comparing hybrid peer-to-peer systems. In VLDB' 2001, 2001.
- [47]. Yang, B., Garcia-Molina, H.: Improving efficiency of peer-to-peer search. In Proc. of the 27<sup>th</sup> International Conference on Very Large Databases, September 2001.
- [48]. Zhao, B.Y., Kubiawicz, J. D., Joseph, A. D.: Tapestry: An infrastructure for fault-tolerant wide-area location and routing. In UC Berkley Technical Report, UCB/CSD-01-1141, 2001.
- [49]. ZQL website, <http://dyade.inrialpes.fr/membres/gibello/ZQL/Zql.html>.