



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ

ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΣΥΛΛΟΓΗ ΚΑΙ ΔΙΑΝΟΜΗ ΠΛΗΡΟΦΟΡΙΑΣ
ΜΕ ΚΑΤΑΝΕΜΗΜΕΝΕΣ ΚΑΙ ΚΙΝΗΤΕΣ
ΣΥΝΙΣΤΩΣΕΣ**

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΕΙΡΗΝΗΣ Χ. ΣΥΓΚΟΥΝΑ

*Διπλωματούχου Ηλεκτρολόγου Μηχανικού και Μηχανικού
Υπολογιστών Ε.Μ.Π*

Αθήνα, Μάρτης 2006

ΠΕΡΙΛΗΨΗ

Η παρούσα διατριβή υποστηρίζει λύσεις που αποσκοπούν στη βελτιστοποίηση της επίδοσης συλλογής και διανομής πληροφορίας με χρήση κατανεμημένων και κινητών συνιστωσών. Εστιάζοντας στα συστήματα μεσιτείας, τα οποία αντιμετωπίζουν αποτελεσματικά ζητήματα σχετικά με την ετερογένεια, την αυτονομία και την κατανομή των πηγών πληροφορίας και επιλέγοντας την κατάλληλη αρχιτεκτονική, η οποία υπαγορεύεται από την κλίμακα και τη φύση της εκάστοτε εφαρμογής, η βελτιστοποίηση της επίδοσης αποτελεί κρίσιμο ζήτημα. Η πρόκληση είναι να εξασφαλιστεί διανομή πληροφορίας στους πελάτες με καλή ποιότητα υπηρεσίας, διατηρώντας αποδοτική και ισορροπημένη κατανάλωση πόρων του συστήματος. Προς αυτήν την κατεύθυνση μελετάται η αξιοποίηση του υποδείγματος κινητού κώδικα μέσω της παροχής ενός κατανεμημένου υπολογιστικού περιβάλλοντος με κινητές συνιστώσες και προτείνονται μηχανισμοί βελτιστοποίησης που προκύπτουν από την ανάλυση και επίλυση ορισμένων κρίσιμων προβλημάτων επίδοσης.

Αρχικά, παρουσιάζονται οι αρχιτεκτονικές των κατανεμημένων πληροφοριακών συστημάτων και οι τεχνολογίες διανομής δεδομένων, και αναλύεται ο ευρύτερος ρόλος της μεσιτείας. Στη συνέχεια, εστιάζοντας στον τρόπο κατανομής των συνιστωσών υπολογισμού της πληροφορίας, μελετάται το υπόδειγμα κινητού κώδικα και αναλύονται τα πιθανά οφέλη από την ενσωμάτωσή του στα πληροφοριακά συστήματα. Οι τεχνολογίες των Κινητών Πρακτόρων και των Ενεργών Δικτύων περιγράφονται ως χαρακτηριστικές υλοποιήσεις του υποδείγματος αυτού.

Σε σύστημα κεντρικής μεσιτείας μελετάται ένα πρόβλημα συλλογής κατανεμημένης πληροφορίας με παράλληλη αποστολή κινητών πρακτόρων. Οι κινητοί πράκτορες μετακινούν τη συνιστώσα υπολογισμού κοντά στα δεδομένα, εξαλείφοντας την ανάγκη μεταφοράς περιττών δεδομένων μέσω του δικτύου. Δεδομένου ότι ένας αυξημένος αριθμός πρακτόρων προκαλεί αυξημένη κατανάλωση πόρων, ο καθορισμός του ελάχιστου αριθμού πρακτόρων και των δρομολογίων τους που εξασφαλίζουν συλλογή της πληροφορίας στον ελάχιστο χρόνο αποτελεί ένα σύνθετο πρόβλημα βελτιστοποίησης, η λύση του οποίου προσεγγίζεται με ευριστικούς αλγορίθμους.

Σε σύστημα ομότιμων μεσιτών εξετάζεται η αξιοποίηση των Ενεργών Δικτύων σε επίπεδο ανάπτυξης των συνιστωσών υπολογισμού πληροφορίας στην υποδομή του δικτύου αλλά και λειτουργίας τους. Στα πλαίσια βελτιστοποίησης της επίδοσης σε επίπεδο λειτουργίας μελετάται ένα πρόβλημα αναζήτησης πληροφορίας. Σκοπός είναι να ελαχιστοποιηθεί ο χρόνος εντοπισμού οποιασδήποτε πληροφορίας μεταξύ των μεσιτών με αποδοτική χρήση των πόρων του συστήματος. Τα κριτήρια επίδοσης που αφορούν στο σύστημα αποτιμούν την ποσότητα των επικοινωνιακών και αποθηκευτικών πόρων που δαπανώνται, ενώ αυτά που αφορούν στο χρήστη αποτιμούν το χρόνο εντοπισμού της πληροφορίας. Σε αυτό το πλαίσιο, διερευνάται η αποδοτική εφαρμογή μηχανισμών που βασίζονται αποκλειστικά στη διάδοση των αιτημάτων αναζήτησης στο δίκτυο, αλλά και μηχανισμών που υποστηρίζουν διανομή διαφημίσεων από τις διαθέσιμες πηγές πληροφορίας.

Τέλος, παρατίθενται τα βασικά συμπεράσματα που προκύπτουν από τη μελέτη των παραπάνω προβλημάτων, επισημαίνοντας τους τρόπους αξιοποίησής τους από τα σύγχρονα πληροφοριακά συστήματα, ενώ παράλληλα συζητούνται ενδιαφέρουσες προεκτάσεις.

ABSTRACT

The scope of this thesis is to present solutions towards performance optimization of data collection and delivery with distributed and mobile components. Focusing on brokerage systems, which deal effectively with the heterogeneity, autonomy and distribution of the information sources, and having selected the most appropriate architecture, which is indicated by the scale and nature of each particular application, performance optimization becomes a critical issue. The challenge is to ensure data delivery to the users with high Quality of Service (QoS), while at the same time maintaining efficient and balanced consumption of system resources. In this direction, the exploitation of the mobile code paradigm is studied through the support of a distributed computing environment with mobile components, and several optimization techniques that arise from the analysis and solution to specific critical performance problems are proposed.

Initially, the distributed information systems architectures along with the data delivery technologies are presented, and the role of mediation is analyzed. Then, working on the distribution of information computing components, the mobile code paradigm is studied and the presumable benefits of integrating it into the information systems are analyzed. The technologies of Mobile Agents and Active Networks are described as representative implementations of the given paradigm.

In the framework of a centralized brokerage system, a mobile agent-based distributed information retrieval problem is studied. Mobile agents move the computation near to the data, thus eliminating the need of transferring redundant data through the network. Given that increasing the number of parallel agents increases the system resources consumption, determining the minimum number of agents and the routes they traverse for the completion of the information retrieval task in the minimum time is a complex optimization problem, the solution of which is approximated by heuristic algorithms.

In a Peer-to-Peer (P2P) brokerage system, Active Networks technology is considered for exploitation towards the deployment of information computing components on the network infrastructure and their execution. In the context of performance optimization in the execution plane, a search problem is studied. The objective is to minimize the time to locate any information source among the brokers with efficient use of system resources. The performance metrics from the system point of view estimate the amount of communication and storage resources consumed, whereas the ones from the user point of view estimate the time to locate the requested information. In this context, the efficient application of both request-broadcast-based and advertisement-based mechanisms is explored.

Finally, the conclusions coming from the thorough study of the above problems are presented, giving us useful insight to their potential exploitation by modern information systems, while at the same time bringing forward interesting extensions.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ.....	21
1.1	ΕΞΕΛΙΞΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ..... 21
1.2	ΠΡΟΚΛΗΣΕΙΣ ΣΤΗ ΣΥΛΛΟΓΗ ΚΑΙ ΔΙΑΝΟΜΗ ΠΛΗΡΟΦΟΡΙΑΣ 22
1.3	ΚΑΤΑΝΕΜΗΜΕΝΗ ΥΠΟΛΟΓΙΣΤΙΚΗ ΠΡΟΣ ΔΙΕΙΣΔΥΤΙΚΗ ΥΠΟΛΟΓΙΣΤΙΚΗ 24
1.4	ΚΑΙΝΟΤΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΔΙΑΤΡΙΒΗΣ..... 26
1.5	ΔΙΑΡΘΡΩΣΗ ΔΙΑΤΡΙΒΗΣ 28
1.6	ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ 31
ΚΕΦΑΛΑΙΟ 2: ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ	35
2.1	ΕΙΣΑΓΩΓΗ 35
2.2	ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ..... 36
2.2.1	Πελάτης-Εξυπηρετητής (Client/Server) 37
2.2.2	Σύστημα Ομότιμων Κόμβων (Peer-to-Peer System) 43
2.3	ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΝΟΜΗΣ ΔΕΔΟΜΕΝΩΝ 44
2.3.1	Πρωτόκολλα Διανομής Δεδομένων 44
2.3.2	Μέθοδοι Διανομής Δεδομένων 50
2.3.3	Συχνότητα Διανομής Δεδομένων 53
2.4	ΕΝΟΠΙΟΗΣΗ ΔΕΔΟΜΕΝΩΝ (DATA INTEGRATION) 54
2.4.1	Υπόδειγμα που Βασίζεται στη Διαχείριση Πολλαπλών Βάσεων Δεδομένων (Multidatabase Management-Based Paradigm) 56
2.4.2	Υπόδειγμα που Βασίζεται σε Μεσίτες (Mediator-Based Paradigm) 57
2.5	ΜΕΣΙΤΙΚΑ ΣΥΣΤΗΜΑΤΑ (BROKERAGE SYSTEMS) 59
2.5.1	Αρχιτεκτονικά Μοντέλα Μεσιτείας 59
2.5.2	Ρόλος Μεσιτείας..... 61
2.5.3	Προβλήματα Βελτιστοποίησης Επίδοσης 63
2.6	ΕΠΙΛΟΓΟΣ..... 69
2.7	ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ 71
ΚΕΦΑΛΑΙΟ 3: ΜΟΝΤΕΛΑ, ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΚΙΝΗΤΟΥ ΚΩΔΙΚΑ.....	77
3.1	ΕΙΣΑΓΩΓΗ 77
3.2	ΚΙΝΗΤΡΑ ΧΡΗΣΗΣ ΚΙΝΗΤΟΥ ΚΩΔΙΚΑ 78
3.3	ΜΟΝΤΕΛΑ ΕΠΙΚΟΙΝΩΝΙΑΣ 80
3.3.1	Κλήση Απομακρυσμένης Διαδικασίας (Remote Procedure Call – RPC) 81
3.3.2	Υποδείγματα Κινητού Κώδικα (Mobile Code Paradigms)..... 82
3.3.3	Διαδικασία Επιλογής Υποδείγματος..... 85
3.3.4	Βιβλιογραφική Έρευνα 86
3.4	ΥΠΟΛΟΓΙΣΤΙΚΑ ΜΟΝΤΕΛΑ 88
3.4.1	Υπολογιστική με Κινητούς Πράκτορες (Mobile Agent-based Computing)..... 89
3.4.2	Υπολογιστική με Ενεργά Δίκτυα (Active Networks-based Computing) 96
3.4.3	Σύγκριση..... 102
3.5	ΠΡΟΒΛΗΜΑ ΣΥΛΛΟΓΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕ ΚΙΝΗΤΟΥΣ ΠΡΑΚΤΟΡΕΣ 103
3.5.1	Επιλογή Μοντέλου Επικοινωνίας 103
3.5.2	Περιγραφή Συστήματος Πρακτόρων..... 105
3.5.3	Αρχιτεκτονική Συστήματος Σχεδιασμού Δρομολογίων..... 107
3.6	ΠΡΟΒΛΗΜΑ ΑΝΑΖΗΤΗΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕ ΕΝΕΡΓΑ ΔΙΚΤΥΑ..... 108
3.6.1	Εφαρμογή Ενεργών Δικτύων 108
3.6.2	Ενεργά Δίκτυα και Ομότιμοι Μεσίτες 110
3.7	ΕΠΙΛΟΓΟΣ..... 113
3.8	ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ 114
ΚΕΦΑΛΑΙΟ 4: ΣΧΕΔΙΑΣΜΟΣ ΔΡΟΜΟΛΟΓΙΩΝ ΓΙΑ ΣΥΛΛΟΓΗ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕ ΧΡΗΣΗ ΚΙΝΗΤΩΝ ΠΡΑΚΤΟΡΩΝ	121
4.1	ΕΙΣΑΓΩΓΗ 121
4.2	ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ..... 122
4.3	ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΔΙΑΤΥΠΩΣΗ ΠΡΟΒΛΗΜΑΤΟΣ 124

4.3.1	Διατύπωση με Ακέραιους Περιορισμούς (<i>Integer-constrained Formulation</i>).....	125
4.3.2	Γραφοθεωρητική Διατύπωση (<i>Graph-theoretic Formulation</i>)	127
4.4	ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ	128
4.4.1	Αλγόριθμος “Savings”	128
4.4.2	Αλγόριθμος “Insertion”	129
4.4.3	Αλγόριθμος “Serial Reverse-Constructed Route (SR-CR)”	131
4.5	ΠΟΛΥΠΛΟΚΟΤΗΤΑ ΑΛΓΟΡΙΘΜΩΝ.....	132
4.5.1	Αλγόριθμος “Savings”	133
4.5.2	Αλγόριθμος “Insertion”	134
4.5.3	Αλγόριθμος “SR-CR”	136
4.6	ΠΑΡΑΔΕΙΓΜΑΤΑ.....	136
4.6.1	Παράδειγμα 1: Πληροφορία σε Όλους τους Κόμβους του Γράφου	137
4.6.2	Παράδειγμα 2: Πληροφορία σε Υποσύνολο των Κόμβων του Γράφου.....	139
4.7	ΑΠΟΤΕΛΕΣΜΑΤΑ	141
4.7.1	Σύγκριση Αλγορίθμων	142
4.7.2	Μελέτη Επίδρασης Παραμέτρων	151
4.8	ΚΑΘΟΡΙΣΜΟΣ ΒΕΛΤΙΣΤΟΥ ΑΡΙΘΜΟΥ ΠΡΑΚΤΟΡΩΝ.....	157
4.9	ΠΑΡΑΤΗΡΗΣΕΙΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ	158
4.10	ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ.....	160
ΚΕΦΑΛΑΙΟ 5: ΜΗΧΑΝΙΣΜΟΙ ΑΝΑΖΗΤΗΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΣΕ ΣΥΣΤΗΜΑ ΟΜΟΤΙΜΩΝ ΜΕΣΙΤΩΝ.....		165
5.1	ΕΙΣΑΓΩΓΗ	165
5.2	ΕΠΙΣΚΟΠΗΣΗ ΣΧΕΤΙΚΗΣ ΒΙΒΛΙΟΓΡΑΦΙΑΣ.....	168
5.2.1	Αποκεντρωμένο και Μη-Δομημένο Μοντέλο	169
5.2.2	Αποκεντρωμένο και Δομημένο Μοντέλο.....	175
5.3	ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ	177
5.4	ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΚΡΙΤΗΡΙΑ ΕΠΙΔΟΣΗΣ	178
5.5	ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ	180
5.6	ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΟΜΙΩΣΗΣ	181
5.7	ΜΗΧΑΝΙΣΜΟΙ ΑΝΑΖΗΤΗΣΗΣ ΜΕ ΕΠΙΓΝΩΣΗ ΤΗΣ ΖΗΤΗΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ	182
5.7.1	Ορισμός και Μοντελοποίηση Ιδιοτήτων Εντοπιότητας	183
5.7.2	Μηχανισμοί Αναζήτησης	186
5.7.3	Προετοιμασία Προσομοίωσης και Αποτελέσματα Πειραμάτων	193
5.8	ΜΗΧΑΝΙΣΜΟΙ ΑΝΑΖΗΤΗΣΗΣ ΜΕ ΧΡΗΣΗ ΔΙΑΦΗΜΙΣΕΩΝ	201
5.8.1	Μοντέλο Μικρού-Κόσμου.....	202
5.8.2	Μηχανισμοί Αναζήτησης	205
5.8.3	Προετοιμασία Προσομοίωσης και Αποτελέσματα Πειραμάτων	210
5.9	ΠΑΡΑΤΗΡΗΣΕΙΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ	216
5.10	ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ.....	217
ΚΕΦΑΛΑΙΟ 6: ΕΠΙΛΟΓΟΣ.....		223
6.1	ΑΝΑΣΚΟΠΗΣΗ	223
6.2	ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ	225
ΠΑΡΑΡΤΗΜΑ.....		233
1. ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ.....		233
1.1	ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΔΙΑΤΥΠΩΣΗ ΠΡΟΒΛΗΜΑΤΟΣ	233
1.2	ΛΥΣΗ.....	235
1.2.1	Κλασικοί Ευριστικοί Αλγόριθμοι	235
1.2.2	Μετα-ευριστικοί Αλγόριθμοι.....	240
1.3	ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ	250

ΕΥΡΕΤΗΡΙΟ ΣΧΗΜΑΤΩΝ

Σχήμα 2.1: Λογική Διαστρωμάτωση Εφαρμογών	38
Σχήμα 2.2: Μοντέλο «Ενισχυμένου Πελάτη».....	39
Σχήμα 2.3: Μοντέλο «Αδύναμου Πελάτη».....	39
Σχήμα 2.4: Αρχιτεκτονική Τριών Επιπέδων	41
Σχήμα 2.5: Πλήρως Αποκεντρωμένο Σύστημα Ομότιμων Κόμβων	44
Σχήμα 2.6: Ημιαποκεντρωμένο Σύστημα Ομότιμων Κόμβων.....	44
Σχήμα 2.7: Πρωτόκολλο Αιτήματος-Απόκρισης	45
Σχήμα 2.8: Πρωτόκολλο Δημοσίευσης-Εγγραφής.....	46
Σχήμα 2.9: Σύστημα Ενοποίησης Δεδομένων.....	54
Σχήμα 2.10: Σύστημα Κεντρικής Μεσιτείας Smart-EC.....	64
Σχήμα 2.11: Γενική Αρχιτεκτονική Συστήματος Context	66
Σχήμα 2.12: Σύστημα Ομότιμων Μεσιτών Context	67
Σχήμα 3.1: Αναπαράσταση Συνιστωσών, Αλληλεπιδράσεων και Τόπων.....	81
Σχήμα 3.2: Υπόδειγμα Πελάτη-Εξυπηρετητή	82
Σχήμα 3.3: Υποδείγματα Κινητού Κώδικα	83
Σχήμα 3.4: Αποστολή Κινητών Πρακτόρων	90
Σχήμα 3.5: Κόμβος «φιλοξενίας» Κινητών Πρακτόρων.....	91
Σχήμα 3.6: Απλά και Ενεργά Πακέτα	98
Σχήμα 3.7: Στρατηγικές Υλοποίησης Επικοινωνίας Μεσίτη-Παρόχων	104
Σχήμα 3.8: Σύστημα Στατικών και Κινητών Πρακτόρων	106
Σχήμα 3.9: Σύστημα Σχεδιασμού Δρομολογίων.....	108
Σχήμα 3.10: Αρχιτεκτονική Ενεργού Κόμβου	109
Σχήμα 3.11: Ομότιμοι Μεσίτες Πληροφορίας πάνω από Ενεργά Δίκτυα.....	110
Σχήμα 3.12: Σύγκριση Μοντέλων Αναζήτησης Πληροφορίας	113
Σχήμα 4.1: Δρομολόγιο ενός πράκτορα	127
Σχήμα 4.2: Αλγόριθμος Savings	129
Σχήμα 4.3: Αλγόριθμος Insertion	130
Σχήμα 4.4: Αλγόριθμος SR-CR.....	132
Σχήμα 4.5: Γράφος δικτύου παραδείγματος 1.....	137
Σχήμα 4.6: Οι λύσεις των τριών αλγορίθμων στο παράδειγμα 1	139
Σχήμα 4.7: Γράφος δικτύου παραδείγματος 2.....	139
Σχήμα 4.8: Οι λύσεις των τριών αλγορίθμων στο παράδειγμα 2.....	141
Σχήμα 4.9: Χρόνος ολοκλήρωσης αποστολής συναρτήσεως του αριθμού πρακτόρων, για $N=20$	143
Σχήμα 4.10: Χρόνος ολοκλήρωσης αποστολής συναρτήσεως του αριθμού πρακτόρων, για $N=40$	143
Σχήμα 4.11: Χρόνος ολοκλήρωσης αποστολής συναρτήσεως του αριθμού πρακτόρων, για $N=60$	143
Σχήμα 4.12: Χρόνος ολοκλήρωσης αποστολής συναρτήσεως του αριθμού πρακτόρων, για $N=80$	144
Σχήμα 4.13: Χρόνος ολοκλήρωσης αποστολής συναρτήσεως του αριθμού πρακτόρων, για $N=20$ και $N_D=4, 8, 12, 16$	147
Σχήμα 4.14: Χρόνος ολοκλήρωσης αποστολής συναρτήσεως του αριθμού πρακτόρων, για $N=40$ και $N_D=8, 16, 24, 32$	148
Σχήμα 4.15: Χρόνος ολοκλήρωσης αποστολής συναρτήσεως του αριθμού πρακτόρων, για $N=60$ και $N_D=12, 24, 36, 48$	149
Σχήμα 4.16: Κατάταξη των αλγορίθμων με βάση την ποιότητα των λύσεων που παράγουν	150
Σχήμα 4.17: Κατάταξη των αλγορίθμων με βάση το χρόνο εκτέλεσής τους	151
Σχήμα 4.18: Χρόνος συναρτήσεως αριθμού πρακτόρων για διάφορα μεγέθη πληροφορίας	152
Σχήμα 4.19: MRI για διάφορα μεγέθη πληροφορίας.....	153
Σχήμα 4.20: Χρόνος συναρτήσεως αριθμού πρακτόρων για διάφορα αρχικά μεγέθη πρακτόρων.....	153
Σχήμα 4.21: MRI για διάφορα αρχικά μεγέθη πρακτόρων.....	154
Σχήμα 4.22: Χρόνος συναρτήσεως αριθμού πρακτόρων για διάφορες πυκνότητες δικτύου.....	155
Σχήμα 4.23: MRI για διάφορες πυκνότητες δικτύου	155
Σχήμα 4.24: Χρόνος συναρτήσεως αριθμού πρακτόρων για ομοιόμορφο και ανομοιόμορφο γράφο.....	156
Σχήμα 4.25: MRI για ομοιόμορφο και ανομοιόμορφο γράφο.....	157
Σχήμα 4.26: Συνολικό κόστος συναρτήσεως αριθμού πρακτόρων για διαφορετικές τιμές του c	158
Σχήμα 5.1: P2P Σύστημα Διανομής Πληροφορίας	166
Σχήμα 5.2: Γράφος μοντελοποίησης ενός P2P υπολογιστικού συστήματος.....	179

Σχήμα 5.3: Ομότιμες διεργασίες και τα αντίστοιχα threads του προσομοιωτή.....	182
Σχήμα 5.4: Εγχώρια Γεωγραφική Εντοπιότητα (HGL).....	185
Σχήμα 5.5: Εγχώρια Γεωγραφική Εντοπιότητα (HGL) έναντι Απομακρυσμένης Γεωγραφικής Εντοπιότητας (RGL).....	185
Σχήμα 5.6: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το χρόνο, για $\alpha=0.7$	196
Σχήμα 5.7: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το χρόνο, για $\alpha=0.95$	196
Σχήμα 5.8: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το βαθμό γεωγραφικής εντοπιότητας, για πυκνότητα δικτύου ίση με 0.035.....	198
Σχήμα 5.9: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το βαθμό γεωγραφικής εντοπιότητας, για πυκνότητα δικτύου ίση με 0.07.....	199
Σχήμα 5.10: Σχετική μείωση εύρους ζώνης σε σχέση με το βαθμό γεωγραφικής εντοπιότητας, για διαφορετικές πυκνότητες δικτύου.....	199
Σχήμα 5.11: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το βαθμό χωρικής εντοπιότητας.....	201
Σχήμα 5.12: Γειτονικές και μεγάλης ακτίνας συνδέσεις του κόμβου v στο διδιάστατο πλέγμα.....	204
Σχήμα 5.13: Πλήρες b -ary δένδρο ($b=2$) για τους σκοπούς του ιεραρχικού μοντέλου δικτύου.....	205
Σχήμα 5.14: Διάστημα πηγών κατανεμημένων γύρω από την τοπική πηγή S_x βάσει απόστασης.....	207
Σχήμα 5.15: Κατανομή πηγών στον RSD ενός κόμβου με 3 τοπικές πηγές (S_1, S_2, S_3) (α) Στοιχειώδης κατανομή των εγγραφών γύρω από τις τοπικές πηγές (β) Απώτερη κατανομή μετά την εφαρμογή του σχεδίου αντικατάστασης.....	208
Σχήμα 5.16: Κατανεμημένος αλγόριθμος αναζήτησης.....	209
Σχήμα 5.17: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=10$ (2000 πηγές συνολικά) και το πολύ 2 πηγές σε κάθε φύλλο.....	211
Σχήμα 5.18: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=20$ (4000 πηγές συνολικά) και το πολύ 4 πηγές σε κάθε φύλλο.....	212
Σχήμα 5.19: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=30$ (6000 πηγές συνολικά) και το πολύ 6 πηγές σε κάθε φύλλο.....	212
Σχήμα 5.20: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=5$ (1000 πηγές συνολικά) και το πολύ 1 πηγή σε κάθε φύλλο.....	214
Σχήμα 5.21: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=10$ (2000 πηγές συνολικά) και το πολύ 2 πηγές σε κάθε φύλλο.....	214
Σχήμα 5.22: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=20$ (4000 πηγές συνολικά) και το πολύ 4 πηγές σε κάθε φύλλο.....	215

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

1.1 ΕΞΕΛΙΞΗ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Πληροφοριακό σύστημα είναι ένα σύστημα που συλλέγει, αποθηκεύει και διανέμει πληροφορία. Η ανάπτυξη των πληροφοριακών συστημάτων βασίστηκε για μεγάλο χρονικό διάστημα σε συστήματα κεντρικών υπολογιστών, στα οποία οι χρήστες είχαν πρόσβαση με τη βοήθεια τερματικών με στοιχειώδεις δυνατότητες. Βασικός περιορισμός της εν λόγω *συγκεντρωτικής αρχιτεκτονικής (centralized architecture)* ήταν η αδυναμία της να υποστηρίξει εύκολα γραφικές διεπαφές χρηστών, καθώς και πρόσβαση σε πολλαπλές πηγές δεδομένων από γεωγραφικά διασκορπισμένες θέσεις. Στις αρχές της δεκαετίας του '80, οι χρήστες απέκτησαν πραγματική υπολογιστική ισχύ στο γραφείο τους με τη βοήθεια του προσωπικού υπολογιστή (Personal Computer - PC). Αυτή η τεχνολογική επανάσταση στο υλισμικό των υπολογιστών επηρέασε και την αρχιτεκτονική των πληροφοριακών συστημάτων με την εμφάνιση του μοντέλου *Πελάτη-Εξυπηρετητή (Client/Server)*, το οποίο αποτέλεσε μοντέλο αναφοράς για την αρχιτεκτονική των καταναμημένων συστημάτων [1, 2]. Ο «Πελάτης» και ο «Εξυπηρετητής» αποτελούν λογικές διεργασίες, με τον δεύτερο να παρέχει πρόσβαση στην πληροφορία και να την διανέμει στον πρώτο σαν απάντηση στα αιτήματά του [3, 4]. Το μοντέλο αυτό μπορεί να επεκταθεί με την προσθήκη ενδιάμεσων διεργασιών, και στην περίπτωση αυτή χαρακτηρίζεται περαιτέρω από έναν αριθμό επιπέδων, ανάλογα με τον τρόπο κατανομής του λογισμικού της εκάστοτε εφαρμογής στις λογικές διεργασίες (π.χ. 2-επιπέδων, 3-επιπέδων και πολλαπλών επιπέδων) [5, 6]. Περαιτέρω εξέλιξη των καταναμημένων συστημάτων αποτελεί το *σύστημα ομότιμων κόμβων (Peer-to-Peer System, P2P)*, στο οποίο δεν υπάρχει διάκριση μεταξύ του Πελάτη και του Εξυπηρετητή. Αντίθετα, υπάρχουν πολλές όμοιες διεργασίες, καθεμιά από τις οποίες μπορεί να καταστήσει πληροφορία διαθέσιμη για διανομή, αλλά και να εγκαταστήσει απευθείας σύνδεση με οποιοδήποτε άλλη προκειμένου να ανακτήσει πληροφορία [7, 8].

Η αρχιτεκτονική των πιο εξελιγμένων καταναμημένων πληροφοριακών συστημάτων, από το μοντέλο Πελάτη-Εξυπηρετητή 3-επιπέδων έως τα ομότιμα συστήματα, βασίζεται στην εισαγωγή ενός ενδιάμεσου στρώματος λογισμικού μεταξύ του υποκείμενου λειτουργικού συστήματος και του στρώματος εφαρμογών, το οποίο είναι γνωστό ως *μεσισμικό (middleware)*. Το μεσισμικό χειρίζεται θέματα πολυπλοκότητας που σχετίζονται με την κατανομή, την ετερογένεια και την αυτονομία των διαφορετικών συνιστωσών ενός καταναμημένου συστήματος. Σκοπός του είναι να διευκολύνει τη σύνδεση προγραμμάτων πελατών με προγράμματα εξυπηρετητών αλλά και την πρόσβαση στις ετερογενείς πηγές πληροφορίας μέσω μιας κοινής αφαίρεσης προγραμματισμού [9, 10].

Η συνεχώς αυξανόμενη δυνατότητα διασύνδεσης των υπολογιστών μέσω σταθερών, κινητών και ασύρματων δικτύων και η διανομή περιεχομένου υψηλού εύρους ζώνης έχουν οδηγήσει σε μια πληθώρα εφαρμογών διανομής πληροφορίας. Αυτές οι εφαρμογές δημιουργούν νέες προκλήσεις για τη διαχείριση των συνιστωσών ενός κατακευματισμένου πληροφοριακού συστήματος.

1.2 ΠΡΟΚΛΗΣΕΙΣ ΣΤΗ ΣΥΛΛΟΓΗ ΚΑΙ ΔΙΑΝΟΜΗ ΠΛΗΡΟΦΟΡΙΑΣ

Τα τελευταία χρόνια έχει παρατηρηθεί έκρηξη στην ποσότητα και ποικιλία των πληροφοριών που γίνονται διαθέσιμες μέσω των δικτύων [11]. Η εξάπλωση του διαδικτύου και των ενδοδικτύων καθώς και οι αυξανόμενες πρόοδοι στον παγκόσμιο ιστό (World Wide Web – WWW) έχουν πυροδοτήσει την ευρεία ανάπτυξη εντατικών, από πλευράς δεδομένων, υπηρεσιών και συστημάτων διανομής πληροφορίας. Στα πλαίσια αυτά, μελετώνται νέοι τρόποι για την αποτελεσματική διανομή της πληροφορίας σε χρήστες στο γραφείο, στο σπίτι ή στο δρόμο.

Ένα σύνθετο πρόβλημα που αντιμετωπίζουν οι επιχειρήσεις υπολογιστικών συστημάτων είναι η απαίτηση για ενιαία και κλιμακούμενη πρόσβαση στις πολλαπλές, ετερογενείς και διασκορπισμένες πηγές πληροφορίας και αποθήκες, συμπεριλαμβανομένων των βάσεων δεδομένων, των γνωστικών βάσεων, των συστημάτων αρχείων, των ψηφιακών βιβλιοθηκών και των συστημάτων ανάκτησης πληροφορίας. Οι πηγές πληροφορίας μεταβάλλονται συνεχώς, και οι χρήστες βρίσκονται αντιμέτωποι με τις δύσκολες προκλήσεις της πλοήγησης, της συλλογής, της αξιολόγησης και της επεξεργασίας δεδομένων στον πλήρως δυναμικό και ανοιχτό κόσμο της πληροφορίας. Οι υπεύθυνοι για τη λήψη αποφάσεων χρειάζονται συχνά πληροφορίες από πολλαπλές πηγές, αλλά δεν είναι σε θέση να ανακτήσουν και να συνδυάσουν πληροφορίες από διαφορετικές πηγές με τρόπο έγκαιρο και αποτελεσματικό, όχι μόνο εξαιτίας της απρόβλεπτης κατάστασης των δικτύων και του ανταγωνισμού μεταξύ των πηγών, αλλά και εξαιτίας της ετερογενούς και συνεχώς μεταβαλλόμενης φύσης των πηγών πληροφορίας.

Ένα «προηγμένο» σύστημα συλλογής και διανομής πληροφορίας θα πρέπει να υπακούει στις αρχές του ανοιχτού και διαλειτουργικού συστήματος. Δυο άμεσες λειτουργικές απαιτήσεις ενός τέτοιου συστήματος είναι η υποστήριξη εκτατής κατακευματισμένης διαχείρισης αντικειμένων καθώς και δυναμικής διαλειτουργικότητας, τόσο μεταξύ διαφορετικών πηγών πληροφορίας όσο και μεταξύ καταναλωτών και παραγωγών πληροφορίας. Σε όρους αντικειμενοστραφούς ορολογίας, η διαλειτουργικότητα αναφέρεται στη δυνατότητα ανταλλαγής αιτημάτων μεταξύ των αντικειμένων και στη δυνατότητα των αντικειμένων να αιτούνται υπηρεσίες άλλων αντικειμένων, ανεξάρτητα από τη γλώσσα στην οποία είναι ορισμένα και τη φυσική τους θέση (π.χ. πλατφόρμες υλισμικού, λειτουργικά συστήματα,

συστήματα διαχείρισης βάσεων δεδομένων). *Καταναμημένα διαλειτουργικά αντικείμενα* είναι τα αντικείμενα που υποστηρίζουν ένα επίπεδο διαλειτουργικότητας πέρα από τα συνηθισμένα όρια που επιβάλλονται από τις γλώσσες προγραμματισμού, τα μοντέλα δεδομένων και τις διεπαφές δικτύων [12]. Οι αφαιρέσεις των καταναμημένων διαλειτουργικών αντικειμένων χρησιμοποιούνται από τις σχετικές υπηρεσίες καταναμημένης διαχείρισης για τον προγραμματισμό και τον έλεγχο της πρόσβασης σε απομακρυσμένα δεδομένα και της διανομής τους.

Θεμελιώδεις στόχοι για την παροχή διαλειτουργικότητας και καταναμημένης διαχείρισης μέσα στο δυναμικό και ανοιχτό κόσμο της πληροφορίας είναι η *επεκτασιμότητα* και η *αποτελεσματικότητα* στην ανάκτηση και διανομή απομακρυσμένης πληροφορίας. Η επεκτασιμότητα αναφέρεται στη δυνατότητα των υπηρεσιών καταναμημένης διαχείρισης να επεκτείνουν τη διαδικασία διανομής της πληροφορίας από ένα σύνολο πηγών δεδομένων σε ένα τυπικά μεγαλύτερο σύνολο καταναλωτών, λαμβάνοντας υπόψη την ετερογένεια των πλατφορμών υλισμικού, των λειτουργικών συστημάτων, των μοντέλων δεδομένων και των γλωσσών προγραμματισμού, καθώς επίσης και την εξελισσόμενη φύση των πηγών πληροφορίας. Η αποτελεσματικότητα στην πρόσβαση και διανομή απομακρυσμένης πληροφορίας αναφέρεται στη δυνατότητα των υπηρεσιών καταναμημένης διαχείρισης να ενσωματώνουν στις στρατηγικές βελτιστοποίησης και εκτέλεσης των καταναμημένων αιτημάτων για πληροφορία τη συνεχή εξέλιξη της πληροφορίας, την κίνηση στο δίκτυο, τη διαθεσιμότητα των πηγών πληροφορίας και των συνδέσεων επικοινωνίας.

Η πρόσβαση σε πληροφορίες που προέρχονται από πηγές ευρέως καταναμημένες και ιδιαίτερα αυτόνομες θέτει νέες προκλήσεις στην καταναμημένη διαχείριση αντικειμένων για διάφορους λόγους. Κατ' αρχάς, προκύπτουν ζητήματα σημασιολογίας και επίδοσης εξαιτίας της ετερογενούς φύσης των πηγών δεδομένων. Δεύτερον, η ποσότητα και ποικιλομορφία της πληροφορίας που γίνεται διαθέσιμη διαμέσου των δικτύων αυξάνει ταχύτατα. Οι χρήστες σήμερα βρίσκονται αντιμέτωποι με μια ολοένα και αυξανόμενη δυσκολία που σχετίζεται με τη συλλογή, την επεξεργασία και το συνδυασμό πληροφοριών με τρόπο αποτελεσματικό και έγκαιρο. Δεδομένου ότι ο όγκος και ο ρυθμός αλλαγής της ενεργούς πληροφορίας αυξάνουν συνεχώς, η δυνατότητα αναζήτησης πληροφορίας με αποκλειστική πρωτοβουλία του χρήστη καθίσταται ανεπαρκής. Η ταυτόχρονη διανομή πληροφορίας με πρωτοβουλία των πηγών, καθώς επίσης και η μετάβαση από μοντέλα που προϋποθέτουν προεγκατεστημένα άκρα επικοινωνίας σε μοντέλα «πρόσβασης οπουδήποτε και οποτεδήποτε» καθίσταται αναγκαία. Τέλος, η πρόσβαση σε πηγές πληροφορίας διαμέσου δικτύων ευρείας περιοχής εξαρτάται σε μεγάλο βαθμό από την τρέχουσα κατάσταση του δικτύου και τις πηγές καθαυτές, συμπεριλαμβανομένων των ενδιάμεσων κόμβων, των συνδέσεων επικοινωνίας και της διαθεσιμότητας των πηγών.

1.3 ΚΑΤΑΝΕΜΗΜΕΝΗ ΥΠΟΛΟΓΙΣΤΙΚΗ ΠΡΟΣ ΔΙΕΙΣΔΥΤΙΚΗ ΥΠΟΛΟΓΙΣΤΙΚΗ

Ένα κατανεμημένο σύστημα αποτελείται από υπολογιστικούς κόμβους που επικοινωνούν μεταξύ τους μέσω ενός δικτύου. Η *κατανεμημένη υπολογιστική (distributed computing)* μελετά τη συντονισμένη χρήση των φυσικά κατανεμημένων υπολογιστικών κόμβων. Υπάρχουν πολλοί διαφορετικοί τύποι κατανεμημένων υπολογιστικών συστημάτων και για να σχεδιαστεί επιτυχώς ένα τέτοιο σύστημα θα πρέπει να αντιμετωπιστούν πολλές προκλήσεις. Ο κύριος στόχος ενός κατανεμημένου συστήματος υπολογισμού είναι να συνδεθούν οι χρήστες και οι πόροι με τρόπο διαφανή, ανοικτό και κλιμακούμενο [13]. Ιδανικά αυτή η διάταξη είναι πολύ περισσότερο ανεκτική σε σφάλματα αλλά και πιο αποτελεσματική από πολλούς συνδυασμούς ανεξάρτητων συστημάτων υπολογιστών.

Τα σημερινά συστήματα κρίσιμων αποστολών κάνουν εκτενή χρήση της κατανεμημένης υπολογιστικής με σκοπό τη διανομή της πληροφορίας πάνω από ετερογενή δίκτυα μεγάλης κλίμακας. Η ανάγκη για κατανεμημένες εφαρμογές καλύπτει ένα ευρύ φάσμα περιπτώσεων, από τη δημιουργία αντιγράφων βάσεων δεδομένων σε εμπορικές εφαρμογές μέχρι την υλοποίηση πρωτοκόλλων τύπου δημοσίευσης-εγγραφής (publish-subscribe), σε πιο εξελιγμένες εφαρμογές. Στο εσωτερικό αυτών των συστημάτων βρίσκονται συστήματα λογισμικού πραγματικού χρόνου, τα οποία μεταφέρουν πληροφορίες αξιόπιστα και με ασφάλεια, σε πραγματικό χρόνο, μεταξύ πολλών εξυπηρετητών εφαρμογών και υπολογιστών, όχι μόνο εντός των ορίων των τοπικών δικτύων αλλά και μέσω δικτύων ευρείας περιοχής και του διαδικτύου. Επιπλέον, η ανάπτυξη εφαρμογών για κατανεμημένα συστήματα βασίζεται όλο και περισσότερο σε λύσεις μεσισμικού, μέσω πλαισίων λογισμικού που παρέχουν αφαιρέσεις υψηλότερου επιπέδου [14].

Το διαδίκτυο αποτελεί το μεγαλύτερο κατανεμημένο σύστημα που έχει αναπτυχθεί. Η επικοινωνιακή υποδομή αναπτύσσεται με γρήγορο ρυθμό παρέχοντας ολοένα και μεγαλύτερο εύρος ζώνης με τη βοήθεια νέων τεχνολογιών και πρωτοκόλλων δικτύου. Επιπλέον, οι δυνατότητες που προσφέρονται από το διαδίκτυο έχουν ενθαρρύνει την έρευνα που αποσκοπεί στη δημιουργία ανάλογης υπολογιστικής υποδομής, η οποία εκμεταλλεύεται την ευρεία ανάπτυξη της επικοινωνιακής υποδομής. Με τον όρο *υπολογιστική υποδομή* εννοούμε τόσο τις τεχνολογίες που είναι διαθέσιμες για την υλοποίηση και την εκτέλεση δίκτυο-κεντρικών εφαρμογών όσο και τα υποδείγματα σχεδιασμού σύμφωνα με τα οποία αυτές οι εφαρμογές μπορούν να δομηθούν. Στα πλαίσια αυτά, το *υπόδειγμα κινητού υπολογισμού (mobile computation paradigm)* προτείνει μια νέα υπολογιστική υποδομή, η οποία υπερβαίνει το παραδοσιακό μοντέλο Πελάτη-Εξυπηρετητή αυξάνοντας τη δυναμικότητα και την ευελιξία των εφαρμογών [15]. Πιο συγκεκριμένα, η υπολογιστική υποδομή αποτελείται από ένα κατανεμημένο περιβάλλον με διάφορα υπολογιστικά

περιβάλλοντα, τα οποία υποστηρίζουν τους υπολογισμούς πολλαπλών μονάδων εκτέλεσης. Οι μονάδες εκτέλεσης μπορούν να αλλάζουν δυναμικά το υπολογιστικό τους περιβάλλον, ακόμη και τον κώδικα που εκτελούν. Χαρακτηριστικό παράδειγμα αποτελεί το *υπόδειγμα κινητού κώδικα (mobile code paradigm)*, το οποίο δημιουργεί νέες δυνατότητες για τα καταναμημένα συστήματα λογισμικού σε ένα ανοικτό και δυναμικά μεταβαλλόμενο περιβάλλον. Μπορεί να βελτιώσει την ταχύτητα, την ευελιξία, τη δομή, και τη δυνατότητα αντιμετώπισης αποσυνδέσεων, και θεωρείται ιδιαίτερα κατάλληλο εάν η προσαρμοστικότητα και η ευελιξία είναι μεταξύ των κύριων απαιτήσεων της εφαρμογής [16]. Πρόσφατες έρευνες σχετικά με την κινητικότητα κώδικα είχαν σαν αποτέλεσμα την παραγωγή αρκετών γλωσσών και συστημάτων που καθιστούν το δίκτυο μια πλήρη υπολογιστική υποδομή, στην οποία οι συνιστώσες μπορούν να κινούνται από μια θέση σε άλλη, ακόμη και κατά τη διάρκεια της εκτέλεσής τους [17].

Η συνεχής ανάπτυξη και διαθεσιμότητα νέων συσκευών υπολογισμού και επικοινωνίας, και η αυξανόμενη συνδετικότητα μεταξύ αυτών των συσκευών με τη βοήθεια των ενσύρματων και ασύρματων δικτύων, δημιουργούν νέες ευκαιρίες για την εκτέλεση των διαφόρων εργασιών οπουδήποτε και οποτεδήποτε. Επιπλέον, λόγω του υψηλού ποσοστού αποδοχής αυτών των συσκευών από το σύνολο των χρηστών, αναμένεται ότι αυτές οι συσκευές θα κυριαρχήσουν σε τέτοιο βαθμό που οι χρήστες θα τις θεωρούν ως δεδομένες. Βάσει αυτών των συνθηκών, το όραμα της *διδεισδυτικής υπολογιστικής (pervasive computing)* είναι να ωθήσει τις υπολογιστικές υπηρεσίες από τις συμβατικές διεπαφές υπολογιστών, σε περιβάλλοντα που χαρακτηρίζονται από διαφανείς μορφές αλληλεπίδρασης [18, 19]. Σε ένα περιβάλλον διεισδυτικής υπολογιστικής, ένας χρήστης θα μπορεί να αλληλεπιδρά με πολλαπλές υπολογιστικές συσκευές, οι οποίες θα μεταφέρονται από τους χρήστες ή θα είναι ενσωματωμένες σε άλλα συστήματα συνθέτοντας μια σφαιρική υποδομή δικτύωσης με καλύτερα επίπεδα ανοικτότητας και δυναμικής. Με τον τρόπο αυτό, οι συσκευές θα μπορούν να πραγματοποιούν υπολογισμούς κατά τρόπο σχετικά μη-παρεισφρητικό, επιδρώντας σε πολλές πτυχές της εργασίας και των καθημερινών δραστηριοτήτων [20, 21, 22]. Για να καταστεί αυτό δυνατό, τα διεισδυτικά συστήματα θα πρέπει να είναι συνεχώς ενήμερα σχετικά με το περιβάλλον τους, το οποίο διαμορφώνεται από την τρέχουσα κατάσταση του χρήστη, το τρέχον υπολογιστικό περιβάλλον και το τρέχον φυσικό περιβάλλον [23, 24], και να ρυθμίζουν-προσαρμόζουν δυναμικά τη λειτουργία και τη συμπεριφορά τους στις αλλαγές περιβάλλοντος. Η διαχείριση της πληροφορίας περιβάλλοντος (context) και η επίγνωση περιβάλλοντος (context-awareness) βρίσκονται στο επίκεντρο μελέτης της ερευνητικής κοινότητας που ασχολείται με την διεισδυτική υπολογιστική.

Προς την κατεύθυνση αυτή, το υπόδειγμα κινητού κώδικα θεωρείται πολλά υποσχόμενο για την πραγματοποίηση της «οποτεδήποτε, οπουδήποτε, με οποιαδήποτε

συσκευή, σε οποιοδήποτε περιβάλλον» πρόσβασης σε ψηφιακές πληροφορίες και υπηρεσίες. Προκειμένου να παραχθεί η πραγματική επίγνωση περιβάλλοντος στα συστήματα διεισδυτικής υπολογιστικής, ευφυή ενσωματωμένα προγράμματα μπορούν να μετακινούνται δυναμικά σε διαφορετικές θέσεις και να ανακτούν πληροφορίες σχετικές με το περιβάλλον [24, 25].

1.4 ΚΑΙΝΟΤΟΜΙΚΑ ΣΤΟΙΧΕΙΑ ΔΙΑΤΡΙΒΗΣ

Βασικός στόχος της διατριβής είναι η *βελτιστοποίηση στην επίδοση συλλογής και διανομής πληροφορίας σε κατανεμημένα περιβάλλοντα*. Η πρόκληση είναι να εξασφαλιστεί διανομή πληροφορίας στους πελάτες με καλή ποιότητα υπηρεσίας, διατηρώντας αποδοτική και ισορροπημένη κατανάλωση πόρων του συστήματος. Τα συστήματα μεσιτείας παρουσιάζουν ιδιαίτερο ενδιαφέρον ως προς την παροχή κλιμακούμενων και αποτελεσματικών υπηρεσιών διανομής πληροφορίας σε ανοιχτά και κατανεμημένα περιβάλλοντα. Ωστόσο, η παρεμβολή μεσιτών σε ένα πληροφοριακό σύστημα απαιτεί αυξημένη επίδοση ώστε να υποσκελιστεί το κόστος εισαγωγής τους. Εστιάζοντας λοιπόν σε λύσεις μεσιτικού που βασίζονται σε μεσιτικά συστήματα, μελετήσαμε ορισμένα προβλήματα που θεωρούνται κρίσιμα για την καλύτερη επίδοση ενός πληροφοριακού συστήματος.

Στα πλαίσια της βελτιστοποίησης της επίδοσης, ο σχεδιασμός κατάλληλων αλγορίθμων για συστήματα μεγάλης κλίμακας αποτελεί μια συνεχώς εξελισσόμενη περιοχή έρευνας. Προς αυτή την κατεύθυνση, η παρούσα ερευνητική εργασία επικεντρώθηκε στην ανάπτυξη *αλγορίθμων προγραμματισμού (planning algorithms)* και *αλγορίθμων αναζήτησης (search algorithms)* για αποδοτική και κλιμακούμενη συλλογή και διανομή πληροφοριών μέσω μεσιτών. Η πρόκληση στην ανάπτυξη των αλγορίθμων είναι να παρασχεθεί η απαιτούμενη ισορροπία μεταξύ ατομικής και γενικής απόδοσης (σε επίπεδο χρήστη και συστήματος, αντίστοιχα), μειώνοντας τη μέση καθυστέρηση παράδοσης πληροφοριών στους μεμονωμένους χρήστες, ανάλογα με το βαθμό ανοχής του συστήματος σε κατανάλωση πόρων του.

Επιλέγοντας την κατάλληλη αρχιτεκτονική, η οποία ικανοποιεί τις γενικές απαιτήσεις της εκάστοτε εφαρμογής, το πρόβλημα της βελτιστοποίησης επίδοσης αποσυντίθεται σε διαφορετικά υποπροβλήματα, των οποίων η λύση εξαρτάται τόσο από την ιδιαίτερη αρχιτεκτονική που επιλέγεται όσο και από τις ειδικές απαιτήσεις του εκάστοτε συστήματος. Στην παρούσα διατριβή, θεωρώντας σαν βάση αναφοράς μια αντιπροσωπευτική εφαρμογή πληροφοριακού συστήματος για καθεμιά από τις δυο βασικές αρχιτεκτονικές μεσιτικών συστημάτων, δηλαδή *κεντρικής μεσιτείας* και *ομότιμων μεσιτών*, μελετήθηκε ένα χαρακτηριστικό πρόβλημα βελτιστοποίησης για κάθε τύπο αρχιτεκτονικής. Πιο

συγκεκριμένα, για την περίπτωση κεντρικής μεσιτείας θεωρήσαμε ένα διεπιχειρησιακό σύστημα ηλεκτρονικού εμπορίου, το οποίο βασίζεται σε κεντρική διαμεσολάβηση, ενώ για την περίπτωση ομότιμων μεσιτών ένα σύστημα παροχής υπηρεσιών με επίγνωση του περιβάλλοντος. Η μελέτη επικεντρώθηκε στην επικοινωνία ενός μεσίτη με απομακρυσμένες πηγές δεδομένων και με άλλους μεσίτες (ανάλογα με το μοντέλο) αξιοποιώντας *τεχνολογίες κινητού κώδικα*, όπως Κινητούς Πράκτορες (Mobile Agents) και Ενεργά Δίκτυα (Active Networks), με σκοπό τη βελτιστοποίηση στην επίδοση συλλογής και διανομής της πληροφορίας. Ειδικότερα, τα προβλήματα που μελετήθηκαν έχουν ως εξής:

1. Σε σύστημα κεντρικής μεσιτείας, κινητοί πράκτορες αποστέλλονται από τον κεντρικό μεσίτη για να συλλέξουν κατανεμημένη πληροφορία. Δεδομένου ότι ένας αυξημένος αριθμός πρακτόρων συνεπάγεται αυξημένη κατανάλωση πόρων του συστήματος, σκοπός είναι να καθοριστούν ο ελάχιστος αριθμός πρακτόρων και τα δρομολογία τους που εξασφαλίζουν συλλογή της πληροφορίας στον ελάχιστο χρόνο. Για ένα σταθερό αριθμό πρακτόρων, το πρόβλημα είναι *NP-hard* και η λύση του προσεγγίστηκε με τρεις ευριστικούς αλγορίθμους, δυο γνωστούς και ένα νέο. Αναλύσεις χρονικής πολυπλοκότητας και δοκιμές προσομοιώσεων παρείχαν συγκριτικά αποτελέσματα σχετικά με το χρόνο υπολογισμού των αλγορίθμων και τις λύσεις που παράγουν. Με επιλεκτική παραμετρική μεταβολή του αριθμού πρακτόρων μπορεί επιπλέον να καθοριστεί ο αριθμός των πρακτόρων που εξασφαλίζει την επιθυμητή ισορροπία μεταξύ των πόρων που δαπανώνται και του χρόνου ολοκλήρωσης της αποστολής, η οποία αποτελεί αντικείμενο διαπραγμάτευσης μεταξύ του συστήματος και του χρήστη.
2. Σε σύστημα ομότιμων μεσιτών που σχηματίζουν υπερκείμενο δίκτυο, μελετήθηκε ένα πρόβλημα αναζήτησης πληροφορίας. Σκοπός είναι να ελαχιστοποιηθεί ο χρόνος εντοπισμού της ζητούμενης πληροφορίας μεταξύ των μεσιτών, με αποδοτική χρήση των πόρων του συστήματος. Εφαρμόστηκαν τόσο αντιδραστικοί μηχανισμοί, οι οποίοι βασίζονται στην ευρεία διάδοση των αιτημάτων αναζήτησης μέσω του δικτύου, όσο και προδραστικοί, οι οποίοι βασίζονται στη διάδοση διαφημίσεων των πηγών πληροφορίας. Στα πλαίσια των αντιδραστικών μηχανισμών, οι προτεινόμενοι μηχανισμοί βασίζονται σε κατάλληλα περιορισμένες εκδόσεις του αλγορίθμου της πλημμύρας, αξιοποιώντας *ιδιότητες εντοπιότητας αναφοράς (locality of reference properties)* που είναι πιθανό να επιδεικνύουν τα μοντέλα ζήτησης της πληροφορίας. Τα αποτελέσματα των προσομοιώσεων έδειξαν ότι οι προτεινόμενοι μηχανισμοί διατηρούν τον ίδιο καλό χρόνο εντοπισμού με τον αλγόριθμο της πλημμύρας, ενώ επιτυγχάνουν αποταμίευση εύρους ζώνης. Στα πλαίσια των προδραστικών

μηχανισμών μελετήθηκε η χρήση διαφημίσεων, η οποία υπονοεί ότι μόλις μια νέα πηγή πληροφορίας εισέλθει στο σύστημα, ο μεσίτης φιλοξενίας της μεταδίδει διαφήμιση της πηγής σε όλους τους άλλους μεσίτες. Ο σκοπός είναι να εξασφαλιστεί μικρός χρόνος εντοπισμού οποιασδήποτε πληροφορίας, δεδομένου ότι πραγματοποιείται επιλεκτική αποθήκευση των διαφημίσεων από κάθε κόμβο. Ο προτεινόμενος μηχανισμός αξιοποιεί το *μοντέλο μικρού-κόσμου (small-world model)* για τη διανομή των διαφημίσεων και στη συνέχεια εφαρμόζει έναν άπληστο αλγόριθμο για τον εντοπισμό της ζητούμενης πληροφορίας. Τα πειραματικά αποτελέσματα επιβεβαιώνουν τον περιορισμένο αριθμό βημάτων εντοπισμού της πληροφορίας που εγγυάται το θεωρητικό μοντέλο και διερευνούν τις δυνατότητες επίτευξης της επιθυμητής ισορροπίας μεταξύ χρόνου εντοπισμού και δαπάνης αποθηκευτικών πόρων.

Στα προβλήματα αυτά αξιοποιήθηκε το *υπόδειγμα κινητού κώδικα* για την καλύτερη επίδοση των συστημάτων, μέσω της παροχής ενός κατανεμημένου υπολογιστικού περιβάλλοντος με κινητές συνιστώσες. Ειδικότερα, στην περίπτωση της κεντρικής μεσιτείας, αναγνωρίστηκαν τα οφέλη που προέρχονται από τη μετακίνηση της συνιστώσας υπολογισμού κοντά στα δεδομένα προς επεξεργασία, και οδήγησαν στην υιοθέτηση του μοντέλου Κινητών Πρακτόρων αντί του μοντέλου Πελάτη-Εξυπηρετητή για την αποδοτικότερη συλλογή της κατανεμημένης πληροφορίας. Επίσης, στην περίπτωση του συστήματος ομότιμων μεσιτών, μελετήθηκε η αξιοποίηση της τεχνολογίας Ενεργών Δικτύων, τόσο σε επίπεδο γρήγορης ανάπτυξης των συνιστωσών υπολογισμού πληροφορίας στην υποδομή του δικτύου, όσο και σε επίπεδο λειτουργίας των υπηρεσιών.

1.5 ΔΙΑΡΘΡΩΣΗ ΔΙΑΤΡΙΒΗΣ

Το περιεχόμενο των κεφαλαίων που ακολουθούν περιγράφεται εν συντομία στη συνέχεια: Καταρχήν, το κεφάλαιο 2 ξεκινά με την περιγραφή των βασικών αρχιτεκτονικών των κατανεμημένων πληροφοριακών συστημάτων και συνεχίζει με την παρουσίαση των τεχνολογιών διανομής δεδομένων, εστιάζοντας στα πρωτόκολλα, τις μεθόδους και τη συχνότητα διανομής. Επίσης, αναλύεται το πρόβλημα της ενοποίησης δεδομένων που προέρχονται από ετερογενείς και διασκορπισμένες πηγές και παρουσιάζονται οι λύσεις του. Υιοθετώντας την προσέγγιση που βασίζεται στο υπόδειγμα μεσιτείας, περιγράφονται οι ειδικές αρχιτεκτονικές των μεσιτικών συστημάτων και ο ευρύτερος ρόλος της μεσιτείας στα συστήματα διανομής πληροφορίας. Τέλος, βάσει μιας αντιπροσωπευτικής εφαρμογής για καθεμιά από τις αρχιτεκτονικές μεσιτικών συστημάτων, εισάγονται τα ειδικότερα προβλήματα μελέτης της διατριβής, τα οποία επιλύονται στα κεφάλαια 4 και 5.

Ενώ στο κεφάλαιο 2 εξετάζονται τεχνολογίες δεδομενο-κεντρικές, οι οποίες αποσκοπούν στην αποδοτική διανομή της πληροφορίας, στο κεφάλαιο 3 παρουσιάζονται τεχνολογίες λειτουργικο-κεντρικές, οι οποίες εστιάζουν στον τρόπο κατανομής του υπολογισμού για την επίτευξη ενός συγκεκριμένου στόχου, και ειδικότερα το υπόδειγμα κινητού κώδικα. Αρχικά μελετώνται τα κίνητρα που οδηγούν στη χρήση κινητού κώδικα. Στη συνέχεια, παρουσιάζονται εναλλακτικά υποδείγματα κινητού κώδικα και συγκρίνονται με το παραδοσιακό μοντέλο απομακρυσμένης επικοινωνίας, ενώ στα πλαίσια υπολογιστικών μοντέλων μελετώνται οι τεχνολογίες των Κινητών Πρακτόρων και των Ενεργών Δικτύων. Τέλος, παρουσιάζεται η αξιοποίηση του υποδείγματος κινητού κώδικα στα ειδικότερα προβλήματα μελέτης της διατριβής. Πιο συγκεκριμένα, περιγράφεται η αξιοποίηση κινητών πρακτόρων στα πλαίσια της συλλογής κατανεμημένης πληροφορίας σε σύστημα κεντρικής μεσιτείας, και επίσης η χρήση Ενεργών Δικτύων στα πλαίσια της αναζήτησης πληροφορίας σε σύστημα ομότιμων μεσιτών.

Το αντικείμενο του κεφαλαίου 4 είναι η επίλυση του προβλήματος συλλογής κατανεμημένης πληροφορίας σε σύστημα κεντρικής μεσιτείας με παράλληλη αποστολή κινητών πρακτόρων. Αρχικά παρουσιάζονται οι σχετικές εργασίες. Ακολουθούν η γενική περιγραφή και η μαθηματική διατύπωση του προβλήματος. Στα πλαίσια της επίλυσής του, περιγράφονται οι προτεινόμενοι αλγόριθμοι, αναλύεται η χρονική τους πολυπλοκότητα και παρουσιάζονται δυο χαρακτηριστικά παραδείγματα εφαρμογής τους. Επιπλέον, παρουσιάζονται αποτελέσματα που προέκυψαν από δοκιμές με τη βοήθεια προσομοιωτή, τα οποία αφορούν τόσο στην αποδοτικότητα και αποτελεσματικότητα των αλγορίθμων όσο και στη μελέτη της επίδρασης συγκεκριμένων παραμέτρων του προβλήματος στη λύση του. Τέλος, προτείνεται μια μέθοδος προσδιορισμού του βέλτιστου αριθμού πρακτόρων που πρέπει να χρησιμοποιηθούν. Το κεφάλαιο ολοκληρώνεται με την ανάλυση των βασικών συμπερασμάτων που προέκυψαν.

Το κεφάλαιο 5 σχετίζεται με τη μελέτη του προβλήματος αναζήτησης πληροφορίας σε σύστημα ομότιμων μεσιτών. Καταρχήν, παρουσιάζεται η σχετική με το πρόβλημα βιβλιογραφία. Στη συνέχεια περιγράφεται λεπτομερώς το υπό εξέταση πρόβλημα, μοντελοποιείται το σύστημα και ορίζονται τα κριτήρια επίδοσης. Οι προτεινόμενοι μηχανισμοί για τη λύση του προβλήματος χωρίζονται σε δυο κατηγορίες: τους αντιδραστικούς και τους προδραστικούς. Στα πλαίσια των αντιδραστικών μηχανισμών, εξετάζονται ιδιότητες εντοπιότητας αναφοράς που είναι δυνατόν να παρουσιάζουν τα μοντέλα ζήτησης πληροφορίας. Αφού οριστούν και μοντελοποιηθούν οι ιδιότητες, περιγράφονται οι προτεινόμενοι μηχανισμοί που τις αξιοποιούν, και παρουσιάζονται τα αποτελέσματα των δοκιμών που πραγματοποιήθηκαν με χρήση ενός ανεξάρτητου εργαλείου προσομοίωσης. Στα πλαίσια των προδραστικών μηχανισμών, περιγράφεται ένας μηχανισμός

διανομής διαφημίσεων πηγών που βασίζεται στο μοντέλο μικρού-κόσμου, και παρουσιάζονται σχετικά πειραματικά αποτελέσματα που προέκυψαν από τον προσομοιωτή. Το κεφάλαιο ολοκληρώνεται με ανάλυση των συμπερασμάτων από την εφαρμογή των προτεινόμενων μηχανισμών.

Τέλος, στο κεφάλαιο 6 γίνεται μια ανακεφαλαίωση της διατριβής και παρατίθενται τα βασικά συμπεράσματα που προέκυψαν από την παρούσα εργασία, καθώς και μελλοντικές προεκτάσεις των συγκεκριμένων θεμάτων που μελετήθηκαν. Στο παράρτημα περιγράφονται οι αλγόριθμοι από τη βιβλιογραφία που έχουν εφαρμοστεί για την επίλυση του προβλήματος δρομολόγησης οχημάτων, το οποίο αποτελεί απλούστερη εκδοχή του προβλήματος κινητών πρακτόρων που μελετάται στο κεφάλαιο 4.

1.6 ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ

- [1] G. Coulouris, J. Dollimore, T. Kindberg, “Distributed Systems: Concepts and Design”, Publisher: Addison Wesley, ISBN: 0321263545, June 2005.
- [2] A. Tannenbaum, M. Steen, “Distributed Systems, Principles and Paradigms”, Publisher: Prentice Hall, ISBN: 0130888931, September 2001, Chapters 1 & 2.
- [3] R.M. Adler, “Distributed Coordination Models for Client/Server Computing”, IEEE Computer, Vol.28, No.4, pp. 14-22, April 1995.
- [4] A. Berson, “Client/Server Architecture”, Publisher: McGraw-Hill, ISBN: 0-07-005076-7, 1994.
- [5] L. Barnes and D. Shimberg, “Client/Server & Beyond: Strategies for the 21st Century”, Publisher: Prentice Hall, ISBN: 0135325161, April 1997.
- [6] A. Dickman, “Two-Tier versus Three-Tier Apps.”, Informationweek 553, pp.74-80, November 13, 1995.
- [7] M. Parameswaran, A. Susarla, A.B. Whinston, “P2P Networking: An Information-sharing Alternative”, IEEE Computer, Vol.34, No.7, pp.31-38, July 2001.
- [8] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, Z. Xu, “Peer-to-Peer Computing”, Technical Report HPL-2002-57, HP Laboratories Palo Alto, March 8, 2002.
- [9] P.A. Bernstein, “Middleware: A Model for Distributed Services”, Communications of the ACM, Vol.39, No.2, pp.86-97, February 1996.
- [10] “Middleware can Mask the Complexity of your Distributed Environment”, Client/Server Economics Letter, Vol.2, No.6, pp.1-5, June 1995.
- [11] L. Liu, L. Yan, and T. Ozsu, “Interoperability in Large-scale Distributed Information Delivery Systems”, in Advances in Workflow Systems and Interoperability, A. Dogac et.al (eds.), Springer-Verlag, 1998.
- [12] M. Betz, “Interoperable Objects: Laying the Foundation for Distributed Object Computing”, Dr. Dobb’s Journal: Software Tools for Professional Programmer, Vol.19, No.11, pp. 18-31, October 1994.
- [13] C. Hewitt, “Open Information Systems Semantics for Distributed Artificial Intelligence”, Journal of Artificial Intelligence, Vol.47, No.1, pp.79-106, January 1991.
- [14] A. Davies, “Computational Intermediation and the Evolution of Computation as a Commodity”, Applied Economics, Vol.36, No.11, pp.1131-1142, June 2004.
- [15] C. Ghezzi, G. Vigna, “Mobile Code Paradigms and Technologies: A Case Study”, Proceedings of the 1st International Workshop on Mobile Agents, Berlin, Germany, pp.39-49, April 1997.

- [16] F. Mattern, P. Sturm, “From Distributed Systems to Ubiquitous Computing – The State of the Art, Trends, and Prospects of Future Networked Systems”, Proceedings of KIVS (Kommunikation in Verteilten Systemen) 2003, Leipzig, pp.3-25, February 2003.
- [17] A. Fuggetta, G.P. Picco and G. Vigna, “Understanding Code Mobility”, IEEE Transactions on Software Engineering, Vol.24, No.5, pp.342-361, May 1998.
- [18] M. Weiser, “The Computer for the Twenty-First Century”, Scientific American, Vol.265, No.3, pp.66-75, September 1991.
- [19] M. Weiser, “Hot Topics: Ubiquitous Computing” IEEE Computer, pp.71-72, October 1993.
- [20] G.D. Abowd and E.D. Mynatt, “Charting Past, Present, and Future Research in Ubiquitous Computing”, ACM Transactions on Computer-Human Interaction, Vol.7, No.1, pp.29-58, March 2000.
- [21] C. Efstratiou, K. Cheverst, N. Davies, A. Friday, “An Architecture for the Effective Support of Adaptive Context-Aware Applications”, Proceedings of 2nd International Conference on Mobile Data Management, Hong-Kong, pp. 15-26, January 2001.
- [22] A. Zaslavsky, “Mobility in Enterprise Applications”, Proceedings of the 5th International Conference on Business Information Systems, Poland, April 2002.
- [23] G. Chen and D. Kotz “Context Aggregation and Dissemination in Ubiquitous Computing Systems”, Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002), Callicoon, NY, pp. 105-114, June 2002.
- [24] B. Schilit, N. Adams, and R. Want, “Context-aware Computing Applications”, Proceedings of the WMCSA 1994, Santa Cruz, CA, pp.85-90, 1994.
- [25] A. Zaslavsky, “Mobile Agents: Can They Assist with Context Awareness?”, Proceedings of the 2004 IEEE International Conference on Mobile Data Management (MDM’04), Berkeley, California, January 2004.

ΚΕΦΑΛΑΙΟ 2

***Αρχιτεκτονικές και Τεχνολογίες
Κατανεμημένων Πληροφοριακών
Συστημάτων***

ΚΕΦΑΛΑΙΟ 2: ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

2.1 ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια έχει παρατηρηθεί έκρηξη στη ποσότητα και ποικιλία των πληροφοριών που γίνονται διαθέσιμες ηλεκτρονικά. Εν τω μεταξύ, το περιβάλλον των πληροφοριών είναι ιδιαίτερα κατανεμημένο, δυναμικό και ετερογενές, υποκινώντας ενδιαφέροντα ζητήματα για τα πληροφοριακά συστήματα μεγάλης κλίμακας. Μια από τις μεγαλύτερες ερευνητικές προκλήσεις που τίθενται είναι η αποδοτική και αποτελεσματική διανομή της πληροφορίας. Το κεφάλαιο αυτό εστιάζει στην παρουσίαση των αρχιτεκτονικών και τεχνολογιών διανομής πληροφορίας των κατανεμημένων πληροφοριακών συστημάτων και εισάγει τα προβλήματα που πραγματεύεται η διατριβή σε επόμενα κεφάλαια, ορίζοντας τα γενικά πλαίσια μελέτης τους.

Ένα πληροφοριακό σύστημα αποτελεί υπολογιστικό σύστημα το οποίο παρέχει πληροφορίες στους πελάτες του, συχνά σε ρόλο υποστήριξης αποφάσεων. Οι βασικές απαιτήσεις που πρέπει να ικανοποιεί ένα προηγμένο πληροφοριακό σύστημα είναι η εξασφάλιση *επεκτασιμότητας* και *αποτελεσματικότητας* στη συλλογή και διανομή της πληροφορίας [1]. Η *επεκτασιμότητα* αναφέρεται στη δυνατότητα του συστήματος να επεκτείνει τη διαδικασία διανομής της πληροφορίας από ένα σύνολο πηγών δεδομένων σε ένα τυπικά μεγαλύτερο σύνολο καταναλωτών, υποστηρίζοντας διαλειτουργικότητα, τόσο μεταξύ διαφορετικών πηγών όσο και μεταξύ πηγών και καταναλωτών πληροφορίας. Η *αποτελεσματικότητα* στη συλλογή και διανομή πληροφορίας αναφέρεται στη δυνατότητα του συστήματος να ενσωματώνει στο σχέδιο διανομής τη συνεχή εξέλιξη της πληροφορίας, την κίνηση στο δίκτυο, τη διαθεσιμότητα των πηγών πληροφορίας και των συνδέσεων επικοινωνίας.

Έχοντας επιλέξει την κατάλληλη γενική αρχιτεκτονική, η οποία υπαγορεύεται από τη κλίμακα και τη φύση της εκάστοτε εφαρμογής, καθώς επίσης και τα κατάλληλα πρωτόκολλα διανομής σύμφωνα με τις λειτουργικές απαιτήσεις της εφαρμογής, η αναζήτηση και ο συνδυασμός πληροφοριών που προέρχονται από ετερογενείς και διασκορπισμένες πηγές αποτελεί σημαντική πρόκληση. Στην περίπτωση αυτή, τα συστήματα μεσιτείας πετυχαίνουν τη λογική ενοποίηση των πηγών με τρόπο κλιμακούμενο και αποτελεσματικό, αποκρύπτοντας την ετερογένεια και τη διανομή τους και δίνοντας στους χρήστες την αίσθηση ενός ενιαίου συστήματος. Ωστόσο, η παρεμβολή ενός μεσίτη σε ένα σύστημα απαιτεί αυξημένη επίδοση στη διανομή της πληροφορίας ώστε να υποσκελιστεί το υψηλό κόστος εισαγωγής του. Η βελτιστοποίηση της επίδοσης συνίσταται σε ελαχιστοποίηση του χρόνου

απόκρισης του συστήματος στα επικείμενα αιτήματα πελατών, και των δαπανών σε πόρους του συστήματος. Στο πλαίσιο αυτό, ο σχεδιασμός κατάλληλων αλγορίθμων για αποδοτική συλλογή και διανομή πληροφοριών μέσω μεσιτών αποτελεί μια συνεχώς εξελισσόμενη περιοχή έρευνας.

Τα κύρια σημεία αυτού του κεφαλαίου περιγράφονται εν συντομία στη συνέχεια. Καταρχήν, παρουσιάζονται οι βασικές αρχιτεκτονικές των κατανεμημένων πληροφοριακών συστημάτων. Δεύτερον, περιγράφονται οι τεχνολογίες διανομής δεδομένων, εστιάζοντας στα πρωτόκολλα, τις μεθόδους και τη συχνότητα διανομής. Τρίτον, συζητείται το πρόβλημα της ενοποίησης δεδομένων που προέρχονται από ετερογενείς και διασκορπισμένες σε διαφορετικούς κόμβους πηγές, και παρουσιάζονται οι λύσεις του. Στη συνέχεια, υιοθετώντας την προσέγγιση που βασίζεται στο υπόδειγμα μεσιτείας, εστιάζουμε στα συστήματα μεσιτών, περιγράφοντας τις ειδικές αρχιτεκτονικές και τον ευρύτερο ρόλο της μεσιτείας στα συστήματα διανομής πληροφορίας. Τέλος, θεωρώντας σαν βάση αναφοράς μια αντιπροσωπευτική εφαρμογή για καθεμιά από τις βασικές αρχιτεκτονικές μεσιτικών συστημάτων, εισάγουμε τα προβλήματα μελέτης της διατριβής, τα οποία επιλύονται στα κεφάλαια 4 και 5.

2.2 ΑΡΧΙΤΕΚΤΟΝΙΚΕΣ ΚΑΤΑΝΕΜΗΜΕΝΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Η *αρχιτεκτονική λογισμικού* περιγράφει τον τρόπο με τον οποίο η λειτουργικότητα μιας εφαρμογής κατανέμεται στις διάφορες συνιστώσες λογισμικού και πώς αυτές οι συνιστώσες κατανέμονται στους επεξεργαστές. Καθώς οι εφαρμογές εξελίσσονται σε κλίμακα και κρισιμότητα αποστολής, αυξάνονται αντίστοιχα οι απαιτήσεις για καλύτερη απόδοση, ευελιξία και επεκτασιμότητα. Στα πλαίσια ενός τόσο απαιτητικού σεναρίου, η επιλογή και υλοποίηση της πιο κατάλληλης αρχιτεκτονικής θεωρείται ιδιαίτερα κρίσιμη, αφού έχει επιπτώσεις σε όλες τις πτυχές του σχεδιασμού και της εφαρμοσμένης μηχανικής λογισμικού. Ο σχεδιαστής εξετάζει την πολυπλοκότητα της εφαρμογής, το επίπεδο ενοποίησης και διασύνδεσης που απαιτεί, τον αριθμό και τη διασπορά των χρηστών, τις συνολικές ανάγκες συναλλαγών που καλείται να εξυπηρετήσει και τη φύση του δικτύου και κατόπιν αποφασίζει σχετικά με τον τύπο της πιο κατάλληλης αρχιτεκτονικής.

Στα *κατανεμημένα συστήματα (distributed systems)* το λογισμικό της εφαρμογής κατανέμεται και εκτελείται σε σύνολο συνεργαζόμενων επεξεργαστών που συνδέονται διαμέσου δικτύου [2, 3]. Τα πιο σημαντικά χαρακτηριστικά αυτών των συστημάτων είναι τα εξής:

- *Μερισμός των πόρων.* Επιτρέπουν το μερισμό των πόρων υλισμικού και λογισμικού που σχετίζονται με διαφορετικούς υπολογιστές σε ένα δίκτυο.

- *Ανοικτότητα*. Είναι ανοικτά συστήματα, περιλαμβάνοντας υλισμικό και λογισμικό από διαφορετικούς προμηθευτές.
- *Παράλληλη δράση*. Διαφορετικές διεργασίες μπορούν να λειτουργούν συγχρόνως σε διαφορετικούς υπολογιστές στο δίκτυο, χωρίς απαραίτητα να επικοινωνούν μεταξύ τους.
- *Επεκτασιμότητα*. Οι δυνατότητές τους μπορούν να αυξηθούν με την προσθήκη νέων πόρων για την αντιμετώπιση των νέων απαιτήσεων.
- *Ανοχή βλαβών*. Σε περίπτωση βλάβης ή αποτυχίας, συνήθως μια υποβαθμισμένη υπηρεσία μπορεί να παρασχεθεί.
- *Διαφάνεια*. Οι χρήστες έχουν απολύτως διαφανή πρόσβαση στους πόρους και δεν απαιτείται να έχουν γνώση του συγκεκριμένου τρόπου κατανομής του συστήματος.

Ωστόσο, παρουσιάζουν μειονεκτήματα ως προς τα εξής:

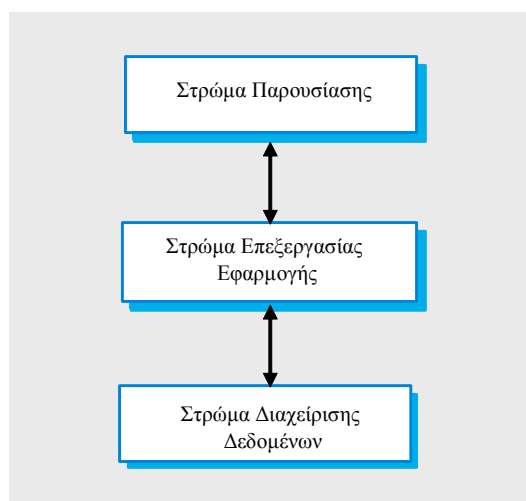
- *Πολυπλοκότητα*. Συνήθως τα συστήματα κατανεμημένης αρχιτεκτονικής είναι πιο σύνθετα από τα συστήματα συγκεντρωτικής αρχιτεκτονικής, καθιστώντας δυσκολότερη την κατανόηση των χαρακτηριστικών τους και τον έλεγχό τους.
- *Ασφάλεια*. Είναι περισσότερο επιρρεπή σε εξωτερικές επιθέσεις, αφού η πρόσβαση πραγματοποιείται από πολλαπλούς υπολογιστές και η διερχόμενη κίνηση εύκολα παρακολουθείται.
- *Διαχειρισιμότητα*. Απαιτούν μεγαλύτερη προσπάθεια για τη διαχείρισή τους, αφού οι υπολογιστές μπορεί να είναι διαφόρων τύπων και λειτουργικών συστημάτων, και ενδεχόμενες βλάβες σε ένα μηχάνημα είναι δυνατό να μεταδοθούν σε άλλα, με απροσδόκητες συνέπειες.
- *Προβλεψιμότητα*. Συχνά επιστρέφουν απρόβλεπτες αποκρίσεις, ανάλογα με το φορτίο στο δίκτυο και την οργάνωση του συστήματος.

Τα πιο αντιπροσωπευτικά μοντέλα της αρχιτεκτονικής κατανεμημένων συστημάτων είναι ο *Πελάτης-Εξυπηρετητής (Client/Server)* και το *Σύστημα Ομότιμων Κόμβων (Peer-to-Peer System)*, τα οποία περιγράφονται στη συνέχεια.

2.2.1 ΠΕΛΑΤΗΣ-ΕΞΥΠΗΡΕΤΗΤΗΣ (CLIENT/SERVER)

Σύμφωνα με την αρχιτεκτονική Πελάτη-Εξυπηρετητή, ένα πληροφοριακό σύστημα μοντελοποιείται ως ένα σύνολο υπηρεσιών που παρέχονται από εξυπηρετητές και ένα σύνολο πελατών που χρησιμοποιούν αυτές τις υπηρεσίες [4]. Οι πελάτες και οι εξυπηρετητές είναι λογικές διεργασίες, αλλά η μεταξύ τους απεικόνιση δεν είναι απαραίτητως ένα-προς-ένα. Το σύστημα Πελάτη-Εξυπηρετητή θα πρέπει να σχεδιάζεται με τρόπο ώστε να απεικονίζει τη λογική δομή της εφαρμογής. Μια κοινή λύση είναι να εφαρμοστεί η *τεχνοτροπία των διαδοχικών στρωμάτων (layered style)*. Στην περίπτωση αυτή ορίζονται τρία στρώματα

(Σχήμα 2.1). Το *Στρώμα Παρουσίασης (Presentation Layer)* σχετίζεται με τη συλλογή των δεδομένων εισόδου των χρηστών και την παρουσίαση των αποτελεσμάτων των υπολογισμών στους χρήστες. Το *Στρώμα Επεξεργασίας Εφαρμογής (Application Processing Layer)* αναλαμβάνει την παροχή της λειτουργικότητας της συγκεκριμένης εφαρμογής. Το *Στρώμα Διαχείρισης Δεδομένων (Data Management Layer)* ασχολείται με τη διαχείριση των βάσεων δεδομένων του συστήματος.



Σχήμα 2.1: Λογική Διαστρωμάτωση Εφαρμογών

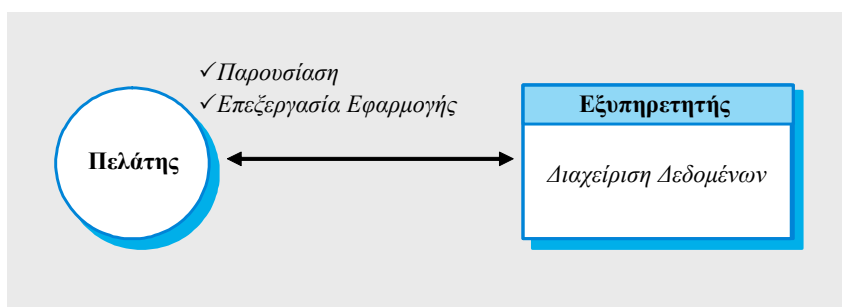
Ανάλογα με τον τρόπο που κατανέμονται τα στρώματα εφαρμογών σε σχέση με τις λογικές διεργασίες του συστήματος, η αρχιτεκτονική Πελάτη-Εξυπηρετητή χαρακτηρίζεται ειδικότερα ως *Δύο Επιπέδων (Two-tier)*, *Τριών Επιπέδων (Three-tier)* και *Πολλαπλών Επιπέδων (Multi-tier)*. Τα σχετικά μοντέλα περιγράφονται στη συνέχεια.

2.2.1.1 Αρχιτεκτονική Δύο Επιπέδων (Two-tier Architecture)

Η απλούστερη αρχιτεκτονική Πελάτη-Εξυπηρετητή είναι αυτή των δύο επιπέδων, σύμφωνα με την οποία μια εφαρμογή οργανώνεται ως ένας εξυπηρετητής και ένα σύνολο πελατών. Στην περίπτωση αυτή, οι πελάτες επικοινωνούν απευθείας με τον εξυπηρετητή, ο οποίος διαχειρίζεται τη βάση δεδομένων. Η λογική της εφαρμογής εδρεύει είτε στον πελάτη, είτε στον εξυπηρετητή βάσης δεδομένων υπό τη μορφή αποθηκευμένων διαδικασιών. Ανάλογα με την περίπτωση, το μοντέλο δύο επιπέδων χαρακτηρίζεται περαιτέρω ως *ενισχυμένου πελάτη (fat-client model)* ή *αδύναμου πελάτη (thin-client model)*, αντίστοιχα [5].

Συνήθως, δεν είναι προφανές πού θα πρέπει να εδρεύει η λογική της εφαρμογής. Αν οι πελάτες έχουν σημαντική υπολογιστική ισχύ, η λογική της εφαρμογής εδρεύει και εκτελείται στον πελάτη. Στην περίπτωση αυτή, ο πελάτης εκτελεί τη λογική της εφαρμογής και χειρίζεται τις αλληλεπιδράσεις με το χρήστη, ενώ ο εξυπηρετητής διατηρεί μόνο τους πόρους του συστήματος, δηλαδή τις βάσεις δεδομένων, οπότε λειτουργεί ουσιαστικά ως

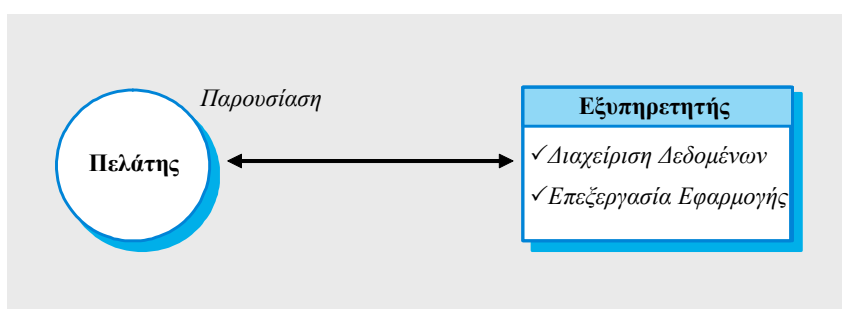
εξυπηρετητής βάσης δεδομένων. Αυτή η προσέγγιση Πελάτη-Εξυπηρετητή καλείται *αρχιτεκτονική δύο επιπέδων «Ενισχυμένου Πελάτη»* (Σχήμα 2.2).



Σχήμα 2.2: Μοντέλο «Ενισχυμένου Πελάτη»

Ωστόσο, η εκτέλεση της λογικής της εφαρμογής στον πελάτη δεν είναι πάντα αποδοτική. Από πλευρά επίδοσης του συστήματος, προκαλεί μεγάλο αριθμό αιτημάτων και μηνυμάτων μεταφοράς δεδομένων μέσα στο δίκτυο. Ένα άλλο πρόβλημα σχετίζεται με την ανάπτυξη και τη συντήρηση του συστήματος. Δεδομένου ότι υπόσταση του προγράμματος εφαρμογής εγκαθίσταται σε κάθε πελάτη, απαιτείται η ενημέρωση όλων των πελατών κάθε φορά που το πρόγραμμα εφαρμογής αλλάζει. Σε συστήματα μεγάλων επιχειρήσεων, αυτό μπορεί να προκαλέσει μεγάλα προβλήματα στην προσπάθεια να υπάρχει συνοχή μεταξύ των διαφορετικών εκδόσεων των προγραμμάτων, και αναπόφευκτα να οδηγήσει σε μεγάλα κόστη συντήρησης.

Μια εναλλακτική λύση είναι η εκτέλεση της λογικής της εφαρμογής στον εξυπηρετητή. Στην περίπτωση αυτή, τόσο η επεξεργασία εφαρμογής όσο και η διαχείριση δεδομένων πραγματοποιούνται στον εξυπηρετητή, ενώ ο πελάτης απλώς χειρίζεται τις αλληλεπιδράσεις με το χρήστη. Η σχετική προσέγγιση Πελάτη-Εξυπηρετητή είναι γνωστή ως *αρχιτεκτονική δύο επιπέδων «Αδύναμου Πελάτη»* (Σχήμα 2.3).



Σχήμα 2.3: Μοντέλο «Αδύναμου Πελάτη»

Η αρχιτεκτονική δύο επιπέδων χρησιμοποιείται εκτενώς σε συστήματα επεξεργασίας πληροφορίας μη κρίσιμου χρόνου, όπου η διαχείριση και οι λειτουργίες του συστήματος δεν είναι σύνθετες, καθώς και σε συστήματα χαμηλού φορτίου συναλλαγών. Γενικά ενδείκνυται για σχετικά ομοιογενή περιβάλλοντα, όπου οι κανόνες επεξεργασίας δεν αλλάζουν συχνά, και όταν ο αριθμός χρηστών είναι περιορισμένος, όπως σε μικρές επιχειρήσεις. Σε περιπτώσεις με

πολύ μεγάλο αριθμό χρηστών και μέτριο αριθμό εξυπηρετητών, η αρχιτεκτονική δύο επιπέδων, με ενισχυμένο ή αδύναμο πελάτη, μπορεί να παρουσιάσει σημαντικά προβλήματα επίδοσης [6, 7]. Πιο συγκεκριμένα, η αρχιτεκτονική δύο επιπέδων «Ενισχυμένου Πελάτη» παρουσιάζει προβλήματα ως προς τα εξής:

- *Μειωμένη δυνατότητα επαναχρησιμοποίησης και δύσκολη συντήρηση.* Καθώς τα προγράμματα πελατών καλούνται να καλύψουν πολλές διαφορετικές ανάγκες εφαρμογών, τείνουν να μετατραπούν σε τεράστιες μονολιθικές εφαρμογές, με μικρή δυνατότητα επαναχρησιμοποίησης. Καθώς επίσης οι εφαρμογές γίνονται πιο πολύπλοκες, καθίσταται δυσκολότερη και η συντήρησή τους, η οποία επιβαρύνεται περαιτέρω από το γεγονός ότι πρέπει να γίνει σε κατανεμημένο περιβάλλον.
- *Επίδοση.* Δεδομένου ότι οι εφαρμογές πελατών έχουν πρόσβαση σε πολλαπλές βάσεις δεδομένων μέσω απλών αιτημάτων, απαιτείται ένας μεγάλος αριθμός καναλιών μεταξύ των πελατών και των εξυπηρετητών, ενώ επίσης ανταλλάσσεται ένας μεγάλος αριθμός μηνυμάτων διαμέσου αυτών των καναλιών.
- *Ετερογένεια.* Καθώς ο αριθμός των πηγών πληροφορίας αυξάνεται, οι εφαρμογές αντιμετωπίζουν ολοένα και περισσότερα προβλήματα ετερογένειας, δεδομένου ότι δεν υπάρχει κάποιο ενδιάμεσο επίπεδο που να παρέχει τις απαραίτητες αφαιρέσεις για την κάλυψη κάθε είδους ετερογενέειας.

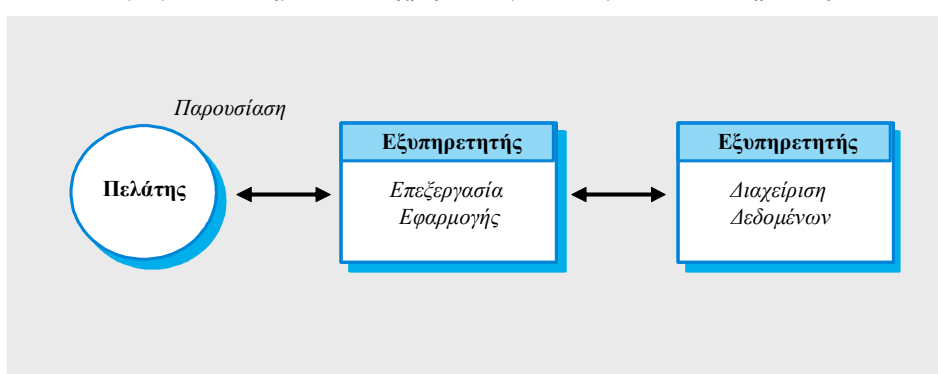
Εάν η λογική της εφαρμογής εκτελείται στον εξυπηρετητή (αρχιτεκτονική «Αδύναμου Πελάτη»), τα προβλήματα που παρουσιάζονται σχετίζονται με τα εξής:

- *Συναλλαγές με πολλαπλές βάσεις.* Δεδομένου ότι οι αποθηκευμένες διαδικασίες χρησιμοποιούνται για πρόσβαση σε συγκεκριμένη βάση δεδομένων, δεν μπορούν να χρησιμοποιηθούν για πρόσβαση σε άλλη βάση. Συνεπώς, περιορίζονται σημαντικά οι δυνατότητες υλοποίησης εφαρμογών που χρησιμοποιούν δεδομένα από πολλές διαφορετικές βάσεις.
- *Επίδοση.* Συνήθως, κάθε κλήση σε μια αποθηκευμένη διαδικασία κινεί μια νέα διεργασία. Κατά συνέπεια, με πολλαπλούς πελάτες, ο εξυπηρετητής μπορεί να παρουσιάσει προβλήματα επίδοσης.
- *Κλείδωμα προμηθευτών.* Οι αποθηκευμένες διαδικασίες μπορούν να χρησιμοποιηθούν μόνο σε περιβάλλοντα που προδιαγράφονται από τους αντίστοιχους προμηθευτές.

Συνεπώς, καμιά από τις δυο προσεγγίσεις της αρχιτεκτονικής δύο επιπέδων δεν θεωρείται κατάλληλη για εφαρμογές μεγάλης κλίμακας.

2.2.1.2 Αρχιτεκτονική Τριών Επιπέδων (Three-tier Architecture)

Η αρχιτεκτονική τριών επιπέδων εμφανίστηκε στο προσκήνιο τη δεκαετία του '90 με σκοπό να ξεπεράσει τους περιορισμούς της αρχιτεκτονικής δύο επιπέδων. Για το λόγο αυτό, εισήγαγε έναν εξυπηρετητή εφαρμογών (*application server*) ως μεσαίο επίπεδο μεταξύ του πελάτη και του εξυπηρετητή βάσης δεδομένων (Σχήμα 2.4). Το μεσαίο επίπεδο αποτελεί ανεξάρτητο τμήμα λογισμικού, το οποίο αναπτύσσεται συνήθως σε ξεχωριστό τμήμα υλισμικού και εκτελεί τη λογική της εφαρμογής, π.χ. θέτει σε εφαρμογή τους επιχειρησιακούς κανόνες, εκτελεί σύνθετους υπολογισμούς, παρέχει μηχανή ανάκτησης δεδομένων. Αξίζει να σημειωθεί ότι η συνδετικότητα μεταξύ των επιπέδων μπορεί να αλλάζει δυναμικά, ανάλογα με το αιτήματα των χρηστών για δεδομένα και υπηρεσίες [8].



Σχήμα 2.4: Αρχιτεκτονική Τριών Επιπέδων

Η αρχιτεκτονική τριών επιπέδων χρησιμοποιείται ως πιο αποτελεσματικός τρόπος σχεδίασης ενός συστήματος Πελάτη-Εξυπηρετητή, παρέχοντας αυξημένη απόδοση, ευελιξία, δυνατότητα επαναχρησιμοποίησης και επεκτασιμότητα, και αποκρύπτοντας την πολυπλοκότητα του κατανεμημένου τρόπου επεξεργασίας από το χρήστη. Αυτά τα χαρακτηριστικά καθιστούν την αρχιτεκτονική τριών επιπέδων αρκετά δημοφιλή επιλογή για εφαρμογές διαδικτύου και δίκτυο-κεντρικά πληροφοριακά συστήματα. Τα πλεονεκτήματα της αρχιτεκτονικής τριών επιπέδων σχετίζονται με αρκετά από τα ζητήματα που αναφέρθηκαν προηγουμένως και ειδικότερα αφορούν στα εξής [9, 10]:

- *Κεντρική διαχείριση των εφαρμογών.* Ο εξυπηρετητής εφαρμογών, ο οποίος περιέχει τη λογική της εφαρμογής, δεν είναι αποκλειστικά συνδεδεμένος ούτε στις υπάρχουσες βάσεις δεδομένων, ούτε στους πελάτες. Η κεντρική αυτή διαχείριση των εφαρμογών βελτιώνει τη διαχειριστικότητα και αυξάνει τις δυνατότητες επαναχρησιμοποίησης.
- *Συγκέντρωση και διανομή των αιτημάτων των πελατών.* Εάν μεγάλος αριθμός πελατών και εξυπηρετητών βάσεων δεδομένων πρέπει να συνδέονται μέσω προγραμμάτων εφαρμογής, ο αριθμός καναλιών που απαιτούνται είναι πολύ μικρότερος. Θεωρώντας ότι n πελάτες πρέπει να έχουν πρόσβαση σε m εξυπηρετητές,

χωρίς το μεσαίο επίπεδο, $n \cdot \mu$ συνδέσεις θα πρέπει να υποστηριχθούν. Χρησιμοποιώντας όμως το μεσαίο επίπεδο, μόνο $n + \mu$ συνδέσεις απαιτούνται: n συνδέσεις από τους πελάτες στο μεσαίο επίπεδο και μ από τη μεσαίο επίπεδο στους εξυπηρετητές βάσεων δεδομένων. Κατά συνέπεια, το μεσαίο επίπεδο προσφέρει καλύτερες δυνατότητες για επεκτασιμότητα.

- *Ευέλικτη αρχιτεκτονική υλισμικού.* Με βάση το βαθμό χρησιμοποίησης των διαφόρων συνιστωσών εφαρμογής ή βάσεων δεδομένων, αυτές αναπαράγονται και κατανέμονται κατάλληλα και ανεξάρτητα. Με τον τρόπο αυτό αποφεύγονται καταστάσεις συμφόρησης.
- *Ενσωμάτωση κληροδοτημένων συστημάτων.* Το μεσαίο επίπεδο μπορεί επίσης να χρησιμοποιηθεί προκειμένου να ενσωματωθούν κληροδοτημένες εφαρμογές. Με τον τρόπο αυτό, αποκρύπτονται πιθανές ετερογένειες κατά την πρόσβαση σε διαφορετικές εφαρμογές και βάσεις δεδομένων.
- *Βελτιωμένη επίδοση.* Ο πελάτης είναι απομονωμένος από τις λειτουργίες που αφορούν στις βάσεις δεδομένων και τη φύση του δικτύου. Μπορεί να έχει πρόσβαση στα δεδομένα εύκολα και γρήγορα χωρίς να πρέπει να γνωρίζει πού βρίσκονται τα δεδομένα ή πόσοι εξυπηρετητές υπάρχουν. Οι καταστάσεις συμφόρησης ελαχιστοποιούνται, αφού το μεσαίο επίπεδο μεταδίδει στον πελάτη ακριβώς την πληροφορία που χρειάζεται.

2.2.1.3 Αρχιτεκτονική Πολλαπλών Επιπέδων (Multi-tier Architecture)

Η αρχιτεκτονική τριών επιπέδων μπορεί να γενικευτεί στην αρχιτεκτονική N επιπέδων, όπου το N μπορεί να είναι αυθαίρετα μεγάλο. Σε αυτήν την περίπτωση, η λειτουργικότητα χωρίζεται σε διάφορα λογικά επίπεδα, τα οποία αναπτύσσονται και διατηρούνται χωριστά. Για παράδειγμα, στην περίπτωση εφαρμογής η οποία απαιτεί πρόσβαση σε δεδομένα που ανήκουν σε διαφορετικές βάσεις, ενσωματωμένος εξυπηρετητής μπορεί να τοποθετηθεί μεταξύ του εξυπηρετητή εφαρμογών και των εξυπηρετητών βάσεων δεδομένων. Ο εν λόγω εξυπηρετητής συλλέγει τα διασκορπισμένα δεδομένα και τα παρουσιάζει στον εξυπηρετητή εφαρμογών σαν να προέρχονταν από μια μοναδική βάση. Αυτή η αρχιτεκτονική θεωρείται πιο κατάλληλη για συστήματα μεγάλης κλίμακας, όπου τα δεδομένα και η εφαρμογή είναι ευμετάβλητα και απαιτείται συλλογή δεδομένων από πολλαπλές πηγές [11, 12].

Οδηγούμενοι από την αρχιτεκτονική ενός επιπέδου (συγκεντρωτική αρχιτεκτονική) στην αρχιτεκτονική N επιπέδων, με την προσθήκη κάθε νέου επιπέδου η αρχιτεκτονική κερδίζει σε ευελιξία, λειτουργικότητα και δυνατότητα κατανομής. Εντούτοις, εισάγει ένα πρόβλημα επίδοσης με την αύξηση του κόστους της επικοινωνίας, και αυξάνει το βαθμό πολυπλοκότητας του συστήματος από πλευράς διαχείρισης και συντονισμού. Με κατάλληλη

σχεδίαση όμως, μπορεί να πετύχει σημαντική μείωση της κίνησης στο δίκτυο, οδηγώντας σε γρηγορότερες επικοινωνίες, μεγαλύτερη αξιοπιστία, και καλύτερη γενική απόδοση.

Σήμερα η βιομηχανία κινείται γρήγορα προς την αρχιτεκτονική N επιπέδων. Είναι γεγονός πως η πλειοψηφία των νέων πληροφοριακών συστημάτων σχεδιάζεται σύμφωνα με το μοντέλο Πελάτη-Εξυπηρετητή N επιπέδων. Ωστόσο, η αρχιτεκτονική αυτή δεν αποκλείει τη χρήση του μοντέλου δύο ή τριών επιπέδων. Ανάλογα με την κλίμακα της εφαρμογής και τις ειδικές απαιτήσεις της πρόσβασης στα δεδομένα επιλέγεται ο κατάλληλος αριθμός επιπέδων.

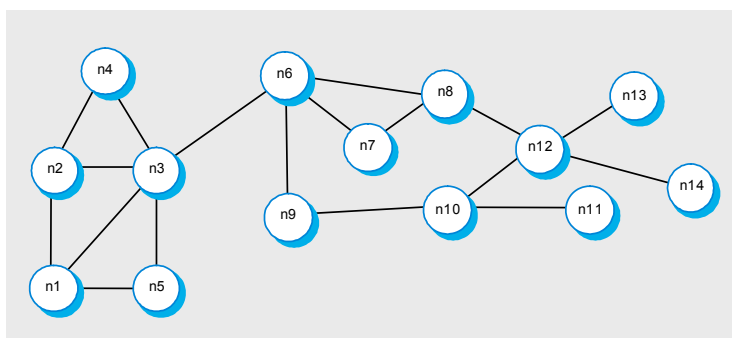
2.2.2 ΣΥΣΤΗΜΑ ΟΜΟΤΙΜΩΝ ΚΟΜΒΩΝ (PEER-TO-PEER SYSTEM)

Τα συστήματα ομότιμων κόμβων είναι κατανεμημένα συστήματα, στα οποία δεν υπάρχει διάκριση μεταξύ πελατών και εξυπηρετητών και οι υπολογισμοί μπορούν να πραγματοποιηθούν σε οποιοδήποτε κόμβο του δικτύου [13, 14]. Αποτελούνται από ομότιμες συνιστώσες λογισμικού, οι οποίες μοιράζονται και χρησιμοποιούν η μία τους πόρους της άλλης για να εκτελέσουν έναν κοινό στόχο. Οι κοινοί πόροι μπορεί να είναι η υπολογιστική ισχύς, ο αποθηκευτικός χώρος, το εύρος ζώνης, ακόμη και η ανθρώπινη παρουσία. Τα συστήματα αυτά βασίζονται σε τρεις θεμελιώδεις αρχές [13]:

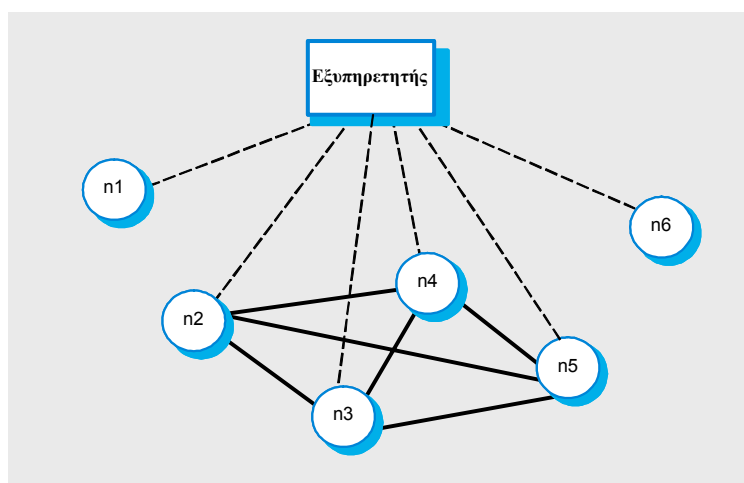
- Ο *μερισμός πόρων (resource sharing)* απαιτεί οι κόμβοι να μοιράζονται μερικούς από τους πόρους τους με άλλους κόμβους.
- Η *αποκέντρωση (decentralization)* συνεπάγεται ότι ένα σύστημα που αποτελείται από πολλούς κόμβους δεν ελέγχεται κεντρικά.
- Η *αυτο-οργάνωση (self-organization)* απαιτείται λαμβάνοντας υπόψη την αποκέντρωση, έτσι ώστε οι αυτόνομοι κόμβοι να συντονίζονται για να εκτελέσουν γενικές δραστηριότητες, οι οποίες βασίζονται σε τοπικούς κοινούς πόρους.

Έχουν αναγνωριστεί δυο βασικοί τύποι της αρχιτεκτονικής ομότιμων κόμβων. Στην *πλήρως αποκεντρωμένη αρχιτεκτονική ομότιμων κόμβων (decentralized P2P architecture)* (Σχήμα 2.5), οι κόμβοι είναι ταυτόχρονα λειτουργικά στοιχεία και μεταγωγείς επικοινωνίας. Ειδικότερα, κάθε κόμβος μπορεί να διατηρεί δεδομένα, να απαντά σε ερωτήσεις αναζήτησης, να προωθεί ερωτήσεις αναζήτησης σε άλλους κόμβους και να δρομολογεί τα κατάλληλα δεδομένα στους αιτούντες. Αντίθετα, στην *ημιαποκεντρωμένη αρχιτεκτονική ομότιμων κόμβων (semi-centralized P2P architecture)* (Σχήμα 2.6), ένας ή περισσότεροι κόμβοι ενεργούν ως εξυπηρετητές για να διευκολύνουν την επικοινωνία μεταξύ των κόμβων. Ειδικότερα, ο ρόλος ενός εξυπηρετητή στην περίπτωση αυτή είναι να βοηθά στη δημιουργία επαφής μεταξύ των κόμβων έτσι ώστε να εγκαθίσταται στη συνέχεια απευθείας σύνδεση μεταξύ αυτών, αλλά και να συντονίζει τα αποτελέσματα των υπολογισμών. Η αρχιτεκτονική

ομότιμων κόμβων θεωρείται ιδανική για συστήματα που απαιτούν αυξημένες δυνατότητες επέκτασης.



Σχήμα 2.5: Πλήρως Αποκεντρωμένο Σύστημα Ομότιμων Κόμβων



Σχήμα 2.6: Ημιαποκεντρωμένο Σύστημα Ομότιμων Κόμβων

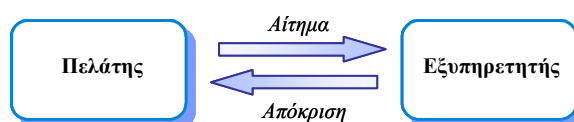
2.3 ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΝΟΜΗΣ ΔΕΔΟΜΕΝΩΝ

2.3.1 ΠΡΩΤΟΚΟΛΛΑ ΔΙΑΝΟΜΗΣ ΔΕΔΟΜΕΝΩΝ

Η διανομή δεδομένων ορίζεται ως η διαδικασία κατά την οποία διανέμονται δεδομένα από ένα σύνολο πηγών (εξυπηρετητές) σε ένα σύνολο πελατών. Υπάρχουν διάφοροι τρόποι επικοινωνίας μεταξύ των εξυπηρετητών και των πελατών για τη διανομή της πληροφορίας, όπως π.χ. οι πελάτες ρωτούν και οι εξυπηρετητές απαντούν, οι εξυπηρετητές δημοσιεύουν διαθέσιμες πληροφορίες και οι πελάτες εγγράφονται συνδρομητές σε αυτές που τους ενδιαφέρουν ή οι εξυπηρετητές εκπέμπουν δεδομένα. Καθένας από τους τρόπους αυτούς μπορεί να θεωρηθεί ως πρωτόκολλο μεταξύ των εξυπηρετητών και των πελατών, και παρουσιάζει πλεονεκτήματα και μειονεκτήματα, τα οποία καταγράφονται στη συνέχεια.

2.3.1.1 Πρωτόκολλο Αιτήματος-Απόκρισης (Request/Response Protocol)

Το πρωτόκολλο Αιτήματος-Απόκρισης ακολουθεί το μηχανισμό διανομής δεδομένων σύμφωνα με τον οποίο οι πελάτες στέλνουν τα αιτήματά τους στους εξυπηρετητές ώστε να μάθουν πληροφορίες του ενδιαφέροντός τους, και οι εξυπηρετητές αποκρίνονται, αποστέλλοντας τη ζητούμενη πληροφορία (Σχήμα 2.7). Πλεονέκτημα του πρωτοκόλλου θεωρείται η υψηλή ποιότητα διανομής πληροφορίας, δεδομένου ότι μόνο η πληροφορία που απαιτείται ρητά από τους πελάτες αποστέλλεται. Σε σύστημα όμως με μικρό αριθμό εξυπηρετητών και μεγάλο αριθμό πελατών, το εν λόγω πρωτόκολλο μπορεί να αποδειχτεί μη-αποδοτικό, αφού η ικανότητα επικοινωνίας και επεξεργασίας των εξυπηρετητών διαμοιράζεται μεταξύ των πελατών. Καθώς ο αριθμός των πελατών αυξάνει, οι εξυπηρετητές μπορεί να κατακλυστούν από τα πολλά αιτήματα, με αποτέλεσμα να αποκρίνονται αργά ή ακόμη και να απορρίπτουν αιτήματα [15].



Σχήμα 2.7: Πρωτόκολλο Αιτήματος-Απόκρισης

2.3.1.2 Πρωτόκολλο Δημοσίευσης-Εγγραφής (Publish/Subscribe Protocol)

Η δυναμική και αποσυζευγμένη φύση των εφαρμογών έχει υποκινήσει την ανάγκη για δημιουργία πιο προσαρμοστικών πρωτοκόλλων επικοινωνίας. Ιδιαίτερα κατάλληλο για την ασαφώς συζευγμένη φύση των κατανεμημένων αλληλεπιδράσεων στις μεγάλης κλίμακας εφαρμογές, το πρωτόκολλο Δημοσίευσης-Εγγραφής έχει λάβει πρόσφατα αυξημένο ενδιαφέρον. Αυτός ο μηχανισμός παρέχει στους συνδρομητές τη δυνατότητα να εκφράσουν το ενδιαφέρον τους για ένα γεγονός ή μια σειρά γεγονότων, προκειμένου να ειδοποιηθούν στη συνέχεια για οποιοδήποτε γεγονός παράγεται και συμφωνεί με το καταχωρημένο ενδιαφέρον τους. Στην περίπτωση αυτή οι πελάτες και οι εξυπηρετητές αποκαλούνται συχνά καταναλωτές και παραγωγοί πληροφορίας, αντίστοιχα. Συνεπώς, οι παραγωγοί δημοσιεύουν τις πληροφορίες που παράγουν και οι καταναλωτές εγγράφονται συνδρομητές στις πληροφορίες που θέλουν να λάβουν. Η πληροφορία χαρακτηρίζεται από τον όρο γεγονός (*event*) και η πράξη παράδοσής της από τον όρο ειδοποίηση (*notification*). Μια υπηρεσία γεγονότων (*event service*) αντιπροσωπεύει έναν ουδέτερο μεσολαβητή μεταξύ των εκδοτών (*publishers*), οι οποίοι ενεργούν ως παραγωγοί γεγονότων, και των συνδρομητών (*subscribers*), οι οποίοι ενεργούν ως καταναλωτές γεγονότων (Σχήμα 2.8).



Σχήμα 2.8: Πρωτόκολλο Δημοσίευσης-Εγγραφής

Δεδομένου ότι η κλίμακα και ο ρυθμός ανανέωσης των πληροφοριών αυξάνει συνεχώς, το πρωτόκολλο Δημοσίευσης-Εγγραφής γίνεται ολοένα και περισσότερο δημοφιλές. Όταν όμως τα ενδιαφέροντα των πελατών αλλάζουν ακανόνιστα, αποδεικνύεται λιγότερο αποδοτικό, επειδή σε τέτοιες καταστάσεις οι πελάτες θα διακόπτονται συνεχώς για να φιλτράρουν πληροφορίες που δεν τους ενδιαφέρουν. Ενώ επίσης είναι ιδιαίτερα χρήσιμο για διανομή νέων ή τροποποιημένων πληροφοριών στους πελάτες, δεν μπορεί να χρησιμοποιηθεί αποδοτικά για τη διανομή παλαιότερων πληροφοριών, όταν οι πελάτες συνειδητοποιούν αργότερα ότι τις χρειάζονται.

Η ισχύς αυτού του βασισμένου σε γεγονότα τύπου πρωτοκόλλου στηρίζεται στη δυνατότητα πλήρους αποσύζευξης σε επίπεδο χρόνου, χώρου αλλά και συγχρονισμού μεταξύ των καταναλωτών και των παραγωγών. Η *χρονική αποσύζευξη* επιτυγχάνεται, δεδομένου ότι τα μέρη που αλληλεπιδρούν δεν είναι αναγκαίο να συμμετέχουν ενεργά στην αλληλεπίδραση την ίδια χρονική στιγμή. Η *αποσύζευξη σε επίπεδο χώρου* εξασφαλίζεται, δεδομένου ότι τα αλληλεπιδρώντα μέρη δεν είναι απαραίτητο να ξέρουν το ένα για το άλλο. Οι εκδότες δεν είναι αναγκαίο να κρατούν αναφορές για τους συνδρομητές, αλλά και το αντίστροφο. Επιπλέον, η *αποσύζευξη συγχρονισμού* είναι εξασφαλισμένη επειδή οι εκδότες δεν μπλοκάρονται καθώς παράγουν γεγονότα, αλλά και οι συνδρομητές μπορούν να ειδοποιούνται ασύγχρονα για το συμβάν ενός γεγονότος, ενώ ταυτόχρονα εκτελούν κάποια άλλη δραστηριότητα [16]. Αυτή η προσέγγιση ασαφούς σύζευξης διευκολύνει την προσαρμοστικότητα και εκτατότητα, επειδή νέοι καταναλωτές και παραγωγοί μπορούν να εισέρχονται, να μετακινούνται ή να απομακρύνονται εύκολα.

Οι συνδρομητές ενδιαφέρονται συνήθως για συγκεκριμένα γεγονότα ή συνδυασμούς γεγονότων. Οι διαφορετικοί τρόποι προσδιορισμού των ενδιαφερόντων τους έχουν οδηγήσει σε διάφορα σχέδια εγγραφής συνδρομής, και ειδικότερα σε αυτά που βασίζονται στο *θέμα* και σε αυτά που βασίζονται στο *περιεχόμενο* των γεγονότων:

- *Πρωτόκολλο Δημοσίευσης-Εγγραφής που βασίζεται στο θέμα (topic-based publish-subscribe)*. Οι συμμετέχοντες μπορούν να δημοσιεύουν γεγονότα και να εγγράφονται συνδρομητές σε μεμονωμένα θέματα, τα οποία προσδιορίζονται από λέξεις-κλειδιά. Τα γεγονότα ταξινομούνται σε ομάδες θεμάτων και μπορούν να φιλτραριστούν μόνο με βάση το θέμα τους. Η εγγραφή συνδρομής αφορά σε κάποιο συγκεκριμένο θέμα, και ο χρήστης λαμβάνει όλα τα γεγονότα που σχετίζονται με αυτό το θέμα. Στην

πράξη, τα συστήματα Δημοσίευσης-Εγγραφής που βασίζονται στο θέμα εισάγουν μια «αφαίρεση» προγραμματισμού, η οποία αντιστοιχίζει μεμονωμένα θέματα σε ξεχωριστά κανάλια επικοινωνίας. Κάθε θέμα θεωρείται μια ξεχωριστή υπηρεσία γεγονότων από μόνη της, η οποία προσδιορίζεται από ένα μοναδικό όνομα και μια διεπαφή που ορίζει τις λειτουργίες «δημοσιεύω» και «εγγράφομαι συνδρομητής».

- *Πρωτόκολλο Δημοσίευσης-Εγγραφής που βασίζεται στο περιεχόμενο (content-based publish-subscribe)*. Στην περίπτωση αυτή, τα γεγονότα δεν ταξινομούνται σύμφωνα με κάποιο προκαθορισμένο εξωτερικό κριτήριο, αλλά σύμφωνα με τις καθαυτές ιδιότητες των γεγονότων. Τέτοιες ιδιότητες μπορούν να είναι εσωτερικές ιδιότητες των γεγονότων ή μετα-δεδομένα που συνδέονται με τα γεγονότα. Τα γεγονότα φιλτράρονται σύμφωνα με τις τιμές των ιδιοτήτων τους. Οι καταναλωτές εγγράφονται συνδρομητές σε γεγονότα, προσδιορίζοντας φίλτρα που ορίζουν περιορισμούς υπό τη μορφή ζευγαριών ονόματος-τιμής των ιδιοτήτων και βασικών τελεστών σύγκρισης. Το πλεονέκτημα ενός τέτοιου είδους συστήματος είναι η ευελιξία του. Με αυτόν τον τρόπο, παρέχεται στο συνδρομητή ακριβώς η πληροφορία που χρειάζεται, χωρίς να πρέπει να μάθει ένα σύνολο ονομάτων θέματος και τα αντίστοιχα περιεχόμενά τους πριν εγγραφεί. Ένα μειονέκτημα είναι το φορτίο που επιβάλλεται στο υποκείμενο σύστημα κατά τη διαδικασία ταιριάσματος των γεγονότων στις εγγραφές συνδρομής [16].

Τα συστήματα Δημοσίευσης-Εγγραφής πρέπει να εξυπηρετούν ένα μεγάλο αριθμό γεωγραφικά κατανεμημένων εκδοτών και συνδρομητών. Δεδομένου ότι τα συστήματα ανέρχονται συνεχώς σε κλίμακα και ο αριθμός συνδρομητών ολοένα αυξάνεται, προκύπτει η ανάγκη για **αποδοτική διανομή των γεγονότων**. Στα συστήματα που βασίζονται στο θέμα των γεγονότων, όπου ο αριθμός των θεμάτων είναι μικρός συγκριτικά με τον αριθμό των συνδρομητών, η λύση είναι συχνά μια μορφή *πολλαπλής διανομής (multicast)*. Σε αυτή την περίπτωση, ένας αλγόριθμος πολλαπλής διανομής καθορίζει μια ομάδα πολλαπλής διανομής και ένα δέντρο πολλαπλής διανομής ανά θέμα. Όταν ένας συνδρομητής εγγραφεί σε ένα θέμα, ένας νέος κόμβος προστίθεται στο αντίστοιχο δέντρο. Όταν δημοσιευτεί ένα γεγονός που αφορά στο συγκεκριμένο θέμα, αυτό μεταδίδεται χρησιμοποιώντας το δέντρο πολλαπλής διανομής. Στα συστήματα που βασίζονται στο περιεχόμενο των γεγονότων, δεδομένου ότι κάθε συνδρομητής μπορεί να έχει και μια ξεχωριστή συνδρομή, δεν είναι εύκολο να βρεθούν τα κοινά χαρακτηριστικά των συνδρομών ώστε αυτές να ομαδοποιηθούν. Συνεπώς, είναι σκόπιμο να επινοηθούν αποδοτικοί κατανεμημένοι αλγόριθμοι πολλαπλής διανομής [17]. Αναμφισβήτητα, τα συστήματα αυτά θεωρούνται τα πιο ευέλικτα, αλλά οι υλοποιήσεις που προσφέρουν εύκολη επεκτασιμότητα είναι αρκετά πιο δύσκολο να πραγματοποιηθούν.

Η τεχνική της *πλημμύρας* (*flooding*) είναι η απλούστερη προσέγγιση υλοποίησης μιας κατανεμημένης υπηρεσίας γεγονότων. Κάθε κόμβος προωθεί την ειδοποίηση ενός γεγονότος που δημοσιεύεται από τοπικό πελάτη σε όλους της γείτονές του, και εάν ένας κόμβος λάβει μια ειδοποίηση από έναν γείτονά του, την διαβιβάζει απλά σε όλους τους γείτονές και σε όλους τους τοπικούς πελάτες του που έχουν συνδρομές σχετικές με το γεγονός. Με την τεχνική της πλημμύρας, η δρομολόγηση είναι τετριμμένη, αλλά προφανώς ανταλλάσσονται πολλά περιττά μηνύματα μεταξύ των κόμβων, επειδή κάθε ειδοποίηση υποβάλλεται σε επεξεργασία από κάθε κόμβο.

Μια εναλλακτική λύση της τεχνικής της πλημμύρας είναι η *δρομολόγηση με βάση το περιεχόμενο των γεγονότων* (*content-based routing*). Κάθε κόμβος διατηρεί έναν πίνακα δρομολόγησης που χρησιμοποιείται για τη δρομολόγηση των ειδοποιήσεων με βάση το περιεχόμενό τους στους τοπικούς πελάτες και τους γείτονές του. Οι πίνακες δρομολόγησης διατηρούνται με την αποστολή πληροφοριών σχετικά με τις νέες και ακυρωμένες εγγραφές συνδρομής μέσω του δικτύου. Έναντι της πλημμύρας, η εν λόγω τεχνική μειώνει τον αριθμό των ειδοποιήσεων που προωθούνται, αλλά δυσχεραίνει τη διαδικασία της προώθησης και εισάγει την ανάγκη για ανανέωση των πινάκων δρομολόγησης, καθώς αλλάζουν συνεχώς οι εγγραφές συνδρομής. Δύο κατηγορίες δρομολόγησης με βάση το περιεχόμενο των γεγονότων διακρίνονται. Η *τέλεια δρομολόγηση* (*perfect routing*) εξασφαλίζει ότι μια ειδοποίηση διαβιβάζεται σε έναν γείτονα μόνο εάν στο αντίστοιχο υποδίκτυο υπάρχει καταναλωτής με συνδρομή σχετική με το γεγονός που δημοσιεύεται. Αντίθετα, με την *ατελή δρομολόγηση*, οι ειδοποιήσεις προωθούνται ακόμη και αν δεν υπάρχουν σχετικές συνδρομές. Δύο εκδόσεις της τέλειας δρομολόγησης είναι γνωστές, η *απλή δρομολόγηση* και αυτή που *βασίζεται σε αλληλοκαλύψεις*. Επιπλέον, *διαφημίσεις* των γεγονότων που παράγει ένας εκδότης μπορούν να χρησιμοποιηθούν για να βελτιστοποιήσουν περαιτέρω τη δρομολόγηση που βασίζεται στο περιεχόμενο των γεγονότων. Οι τεχνικές αυτές περιγράφονται στη συνέχεια:

- *Απλή δρομολόγηση*. Αυτό το σχέδιο δρομολόγησης απαιτεί την εισαγωγή μιας καταχώρησης για κάθε ενεργή εγγραφή συνδρομής στους πίνακες δρομολόγησης όλων των κόμβων, κατακλύζοντας το δίκτυο με όλες τις νέες και ακυρωμένες εγγραφές. Με τον τρόπο αυτό εξασφαλίζεται ότι κάθε ειδοποίηση που δημοσιεύεται, παραδίδεται σε όλους τους ενδιαφερόμενους καταναλωτές, ελαχιστοποιώντας τη κίνηση στο δίκτυο που προκαλείται από την κυκλοφορία των ειδοποιήσεων. Εντούτοις, οι πίνακες δρομολόγησης μπορεί να γίνουν υπερβολικά μεγάλοι, επειδή αυτό το σχέδιο δρομολόγησης απαιτεί όλοι οι κόμβοι να έχουν γνώση για όλες τις ενεργές συνδρομές. Επιπλέον, κάθε πίνακας δρομολόγησης επηρεάζεται εάν μια οποιαδήποτε συνδρομή αλλάξει. Προφανώς, αυτά είναι ανεπιθύμητα χαρακτηριστικά

γνωρίσματα σε συστήματα μεγάλης κλίμακας. Η απλή δρομολόγηση χρησιμοποιείται για παράδειγμα στο σύστημα *Gryphon* [18, 19].

- *Δρομολόγηση με αλληλοκαλύψεις.* Ως εναλλακτική λύση της απλής δρομολόγησης, ο Carzaniga [20] έδειξε ότι η σφαιρική γνώση για όλες τις ενεργές συνδρομές δεν είναι απαραίτητη για την υλοποίηση ενός τέλει αλγορίθμου δρομολόγησης. Οι αλγόριθμοι που προτείνει χρησιμοποιούν την επιλεκτική προώθηση συνδρομών, η οποία βασίζεται σε δοκιμές αλληλοκαλύψεων μεταξύ των συνδρομών, ώστε να αποφευχθεί η πλημμύρα όλων των συνδρομών στο δίκτυο. Παραδείγματος χάριν, μια νέα συνδρομή δεν διαβιβάζεται σε έναν γείτονα εάν προηγουμένως μια συνδρομή έχει διαβιβαστεί, η οποία καλύπτει την προηγούμενη και δεν έχει ακυρωθεί ακόμα. Δυστυχώς, αυτό επίσης υπονοεί ότι εάν μια συνδρομή ακυρώνεται, είναι απαραίτητο να διαβιβαστούν μερικές άλλες συνδρομές σε μερικούς γείτονες. Αυτό αφορά το υποσύνολο των συνδρομών που καλύπτεται από την ακυρωμένη συνδρομή. Αυτού του είδους η δρομολόγηση χρησιμοποιείται από τα συστήματα *Siena* [20] και *Jedi* [21].

- *Χρήση διαφημίσεων.* Οι διαφημίσεις είναι φίλτρα που εκδίδονται από τους παραγωγούς για να δείξουν την πρόθεσή τους να δημοσιεύσουν συγκεκριμένα γεγονότα. Οι διαφημίσεις μπορούν να χρησιμοποιηθούν ως πρόσθετος μηχανισμός για περαιτέρω βελτίωση της δρομολόγησης που βασίζεται στο περιεχόμενο των γεγονότων. Για το λόγο αυτό, ένας δεύτερος πίνακας δρομολόγησης υπάρχει σε κάθε κόμβο. Ο εν λόγω πίνακας περιέχει πληροφορία διαφημίσεων και διατηρείται με χρήση των ίδιων αλγορίθμων με τον πίνακα δρομολόγησης που περιέχει πληροφορία συνδρομών, δηλαδή με αποστολή των νέων και ακυρωμένων διαφημίσεων μέσω του δικτύου. Ενώ οι πίνακες δρομολόγησης συνδρομών χρησιμοποιούνται για τη δρομολόγηση των ειδοποιήσεων από τους παραγωγούς στους καταναλωτές, οι πίνακες δρομολόγησης διαφημίσεων χρησιμοποιούνται για τη δρομολόγηση των εγγραφών συνδρομής από τους καταναλωτές στους παραγωγούς: μια συνδρομή προωθείται σε έναν γείτονα μόνο εάν σχετίζεται με μια ενεργό διαφήμιση που έχει παραληφθεί από αυτόν τον γείτονα προηγουμένως. Οι περισσότεροι αλγόριθμοι δρομολόγησης που χρησιμοποιούν τεχνικές φιλτραρίσματος μπορούν να συνδυαστούν με διαφημίσεις.

2.3.1.3 Πρωτόκολλο Ευρείας Εκπομπής (*Broadcast Protocol*)

Το πρωτόκολλο Ευρείας Εκπομπής επιτρέπει σε πολλαπλούς πελάτες να λαμβάνουν δεδομένα που στέλνονται από μια πηγή. Πελάτες που απαιτούν πρόσβαση σε συγκεκριμένα δεδομένα χρειάζεται να περιμένουν μέχρι αυτά να εμφανιστούν. Το πρωτόκολλο αυτό τείνει να διαδραματίσει πρωταρχικό ρόλο στις εφαρμογές διανομής πληροφορίας, επειδή η

διαδικασία διανομής της πληροφορίας είναι εγγενώς I -προς- N (ή M -προς- N όπου $M \ll N$), που σημαίνει ότι οι πληροφορίες διανέμονται από έναν μικρό αριθμό πηγών σε έναν πολύ μεγαλύτερο αριθμό πελατών που έχουν επικαλυπτόμενα ενδιαφέροντα.

Υπάρχουν δυο χαρακτηριστικοί τύποι ευρείας εκπομπής: η *επιλεκτική* ή *πολλαπλή* (*multicast*) και η *τυχαία* (*random*) [22]. Η επιλεκτική εκπομπή μεταδίδει τα δεδομένα σε μια λίστα γνωστών πελατών και υλοποιείται συνήθως με χρήση δρομολογητή, ο οποίος διατηρεί τη λίστα με τους παραλήπτες. Αντίθετα, η τυχαία εκπομπή στέλνει τα δεδομένα σε μέσο όπου μπορεί να περιμένει απεριόριστος αριθμός πελατών, οι οποίοι δεν είναι γνωστοί εκ των προτέρων. Αξίζει να σημειωθεί ότι η διαφορά μεταξύ της επιλεκτικής εκπομπής και του πρωτοκόλλου Δημοσίευσης-Εγγραφής είναι ότι στην επιλεκτική εκπομπή η λίστα με τους παραλήπτες μπορεί να αλλάζει δυναμικά, χωρίς τη ρητή συνδρομή των πελατών.

Το πρωτόκολλο Ευρείας Εκπομπής προσπαθεί να πετύχει μείωση του φορτίου στο δίκτυο και ταυτόχρονα μείωση του χρόνου απόκρισης. Είναι λοιπόν προφανές ότι είναι περισσότερο αποδοτικό όταν ένας αυξημένος αριθμός πελατών ενδιαφέρονται για τα ίδια στοιχεία [22, 23]. Ο εκπομπός μπορεί να μειώσει το χρόνο απόκρισης εάν ωθεί τα στοιχεία στον κόμβο των δεκτών πριν ο χρήστης επιχειρήσει πρόσβαση σε αυτά, αλλά εάν ο χρήστης τελικά δεν έχει πρόσβαση στα στοιχεία, θα έχει καταναλωθεί άσκοπα εύρος ζώνης.

2.3.2 ΜΕΘΟΔΟΙ ΔΙΑΝΟΜΗΣ ΔΕΔΟΜΕΝΩΝ

Με τη γρήγορη ανάπτυξη τόσο σε όγκο όσο και σε ποικιλία των διαθέσιμων πληροφοριών, σε συνδυασμό με τη συνεχή αύξηση του αριθμού καταναλωτών πληροφορίας, δεν θεωρείται πλέον αποδοτική η χρήση ενός μόνο τρόπου διανομής. Ένα σύγχρονο και ευρείας κλίμακας πληροφοριακό σύστημα θα πρέπει να υποστηρίζει επαρκώς ένα σύνολο διαφορετικών τρόπων διανομής δεδομένων, ώστε να χειρίζεται αποτελεσματικά τους ποικίλους τρόπους επικοινωνίας μεταξύ των πελατών και των εξυπηρετητών, βελτιώνοντας τις δυνατότητες απόκρισής του στα αιτήματα που λαμβάνει. Επιπλέον, με τη παροχή πολλαπλών τρόπων διανομής, επιτρέπεται στο σύστημα να βελτιστοποιηθεί με βάση συγκεκριμένα κριτήρια και σύμφωνα με τις εκάστοτε απαιτήσεις. Στη συνέχεια, περιγράφονται οι πιο δημοφιλείς τρόποι διανομής δεδομένων, οι οποίοι συσχετίζονται με τα πρωτόκολλα διανομής που παρουσιάστηκαν προηγουμένως.

2.3.2.1 Μέθοδος Άντλησης (*Pull-only Mode*)

Με αυτή τη μέθοδο, η μεταφορά δεδομένων από τους εξυπηρετητές στους πελάτες πραγματοποιείται με πρωτοβουλία των πελατών. Οι πελάτες ζητούν ρητά συγκεκριμένη πληροφορία, αποστέλλοντας σχετικό μήνυμα στον εξυπηρετητή. Όταν το αίτημα ενός πελάτη λαμβάνεται από έναν εξυπηρετητή, ο εξυπηρετητής αποκρίνεται, αφού εντοπίσει τη

ζητούμενη πληροφορία. Συνεπώς, το πρωτόκολλο Αιτήματος-Απόκρισης βασίζεται σε αυτή τη μέθοδο.

Οι πληροφορίες που οι πελάτες μπορούν να λάβουν από έναν εξυπηρετητή περιορίζονται σε αυτές για τις οποίες είναι ενήμεροι να ζητήσουν. Κατά συνέπεια, νέα δεδομένα ή τροποποιήσεις των υπαρχόντων είναι δυνατόν να περάσουν απαρατήρητα. Από αυτή την άποψη, δεν μπορεί να εξασφαλιστεί πλήρης εγκυρότητα των στοιχείων που λαμβάνονται. Η εξασφάλιση της εγκυρότητας και η ανανέωση των στοιχείων από την πλευρά των πελατών μπορούν μόνο να επιτευχθούν υπό το κόστος των συχνών ερωτήσεων, οι οποίες δημιουργούν υψηλή κίνηση στο δίκτυο και ενδεχομένως έναν μεγάλο αριθμό περιττών μηνυμάτων, εάν τελικά τα δεδομένα δεν αλλάζουν συχνά. Επίσης, οι εξυπηρετητές θα πρέπει να διακόπτονται συνεχώς ώστε να επεξεργάζονται τα αιτήματα των πελατών και συνεπώς μπορεί εύκολα να παρουσιαστεί δυσχέρεια εξελισσιμότητας για μεγάλους πληθυσμούς πελατών. Παρά αυτές τις αδυναμίες, η μέθοδος άντλησης χρησιμοποιείται συχνά, δεδομένου ότι είναι απλή στην εφαρμογή της και αποφεύγει την αλόγιστη χρήση πόρων μέσω της αποκλειστικής λήψης των στοιχείων που απαιτούνται [15].

2.3.2.2 Μέθοδος Ωθησης (*Push-only Mode*)

Στην περίπτωση αυτή, η μεταφορά δεδομένων από τους εξυπηρετητές στους πελάτες γίνεται με πρωτοβουλία των εξυπηρετητών. Ελλείψει συγκεκριμένων αιτημάτων από τους πελάτες, η μέθοδος ώθησης αντιμετωπίζει το πρόβλημα της απόφασης σχετικά με τα στοιχεία που πρέπει να σταλούν στους πελάτες, δεδομένου ότι η αποστολή άσχετων στοιχείων οδηγεί σε απώλεια πόρων. Επιπλέον, είναι πιθανό οι εξυπηρετητές να μη παραδίδουν τα δεδομένα έγκαιρα. Κατά συνέπεια, η αποδοτικότητα της εν λόγω μεθόδου εξαρτάται σημαντικά από την ικανότητα του εξυπηρετητή να προβλέπει σωστά τις ανάγκες των πελατών. Μια λύση στο πρόβλημα θα ήταν να επιτρέπεται στους πελάτες να παρέχουν προφίλ των ενδιαφερόντων τους στους εξυπηρετητές. Το πρωτόκολλο Δημοσίευσης-Εγγραφής είναι ένας γνωστός τρόπος παροχής τέτοιου είδους προφίλ. Στην περίπτωση αυτή, οι πελάτες εγγράφονται συνδρομητές στις πληροφορίες που θέλουν να λαμβάνουν, παρέχοντας με τον τρόπο αυτό το προφίλ των ενδιαφερόντων τους. Εάν υπάρχουν νέες πληροφορίες, οι εξυπηρετητές θα τις κοινοποιήσουν στους πελάτες που έχουν δηλώσει σχετικό ενδιαφέρον. Αξίζει να σημειωθεί ότι και το πρωτόκολλο Ευρείας Εκπομπής Δεδομένων βασίζεται στη μέθοδο ώθησης. Η υλοποίηση του εν λόγω μηχανισμού με τη βοήθεια καναλιού ευρείας εκπομπής θεωρείται ιδιαίτερα επιτυχημένη για λόγους επεκτασιμότητας. Επειδή οι πελάτες, στην περίπτωση αυτή, είναι κατά ένα μεγάλο ποσοστό παθητικοί, η απόδοση οποιουδήποτε πελάτη που λαμβάνει δεδομένα από το κανάλι δεν επηρεάζεται άμεσα από άλλους πελάτες που ελέγχουν επίσης το κανάλι [15].

Μερικά από τα προβλήματα των κατανεμημένων συστημάτων διανομής πληροφορίας που μπορούν να αντιμετωπιστούν με τη μέθοδο ώθησης είναι τα παρακάτω:

- *Ανακάλυψη πληροφορίας.* Λόγω του μεγάλου όγκου δεδομένων που είναι διαθέσιμα στο δίκτυο, οι χρήστες αντιμετωπίζουν δυσκολίες στον εντοπισμό της επιθυμητής πληροφορίας. Παρά την ύπαρξη εξελιγμένων μηχανών αναζήτησης, η ποιότητα της πληροφορίας που εντοπίζεται εξαρτάται σε μεγάλο βαθμό από τις γνώσεις και τις δεξιότητες του χρήστη. Στα συστήματα που εφαρμόζεται η μέθοδος ώθησης, ειδικά κανάλια μπορούν να χρησιμοποιηθούν για την αναγγελία νέας διαθέσιμης πληροφορίας.
- *Επικαιρότητα πληροφορίας.* Στα συνήθη πληροφοριακά συστήματα ο πελάτης πρέπει συνεχώς να ρωτάει για να εξασφαλίσει ότι τα δεδομένα του είναι ενημερωμένα. Στα συστήματα ώθησης πληροφορίας, ο πελάτης μπορεί να ειδοποιείται όταν ενημερώνεται κάποιο στοιχείο στον εξυπηρετητή. Για να το πετύχει αυτό, μπορεί να χρησιμοποιεί τις γνωστές *υπηρεσίες γεγονότων (event services)*.
- *Πιστοποίηση πληροφορίας.* Η πιστοποίηση της ταυτότητας του προμηθευτή πληροφορίας καθώς και της ταυτότητας και ακεραιότητας της πληροφορίας απλοποιείται στα συστήματα ώθησης, λόγω της φάσης συνδρομής που πραγματοποιείται. Στη φάση αυτή, ανταλλάσσονται κλειδιά κρυπτογράφησης, τα οποία χρησιμοποιούνται αργότερα για την πιστοποίηση του εξυπηρετητή.
- *Εξατομίκευση.* Στα συστήματα ώθησης ο χρήστης μπορεί να δηλώσει τις προτιμήσεις του ρητά. Κατά συνέπεια είναι εύκολο να παρασχεθούν πληροφορίες προσαρμοσμένες στα ενδιαφέροντα του χρήστη.

2.3.2.3 Υβριδική Μέθοδος (Hybrid Mode)

Η υβριδική μέθοδος διανομής αποτελεί συνδυασμό των μεθόδων άντλησης και ώθησης. Για παράδειγμα, στην περίπτωση της μεθόδου άντλησης, ο εξυπηρετητής πρέπει να διακόπτεται συνεχώς για να εξετάζει τα αιτήματα των πελατών και συνεπώς μπορεί εύκολα να παρουσιάσει δυσχέρεια εξελισσιμότητας για μεγάλους πληθυσμούς πελατών. Αυτό θα μπορούσε να διορθωθεί με τη διανομή των στοιχείων μέσω ενός καναλιού ευρείας εκπομπής και παρέχοντας τη δυνατότητα στους πελάτες να «κατασκοπεύουν» το κανάλι για να λαμβάνουν στοιχεία που ζητήθηκαν από άλλους πελάτες [23].

Η προσέγγιση που προτείνεται στο [24] παρουσιάζει επίσης ένα πιθανό τρόπο συνδυασμού των δυο μεθόδων, ως εξής: η μεταφορά δεδομένων από τους εξυπηρετητές στους πελάτες ξεκινά με μια κίνηση άντλησης από την πλευρά του πελάτη, και στη συνέχεια η μετάδοση της ανανεωμένης πληροφορίας πραγματοποιείται με πρωτοβουλία των εξυπηρετητών και τη μέθοδο ώθησης. Η βασική διαφορά της υβριδικής μεθόδου από τη μέθοδο αποκλειστικής ώθησης είναι ότι στην υβριδική, προϋπάρχοντα δεδομένα τα οποία

συμφωνούν με το προφίλ του πελάτη που έστειλε το πρώτο αίτημα, είναι δυνατόν να αποσταλούν στον πελάτη αμέσως μετά την πρώτη κίνηση άντλησης. Αντίθετα, με τη μέθοδο αποκλειστικής ώθησης, ενώ νέα δεδομένα αποστέλλονται στους πελάτες των οποίων τα προφίλ συμφωνούν, είναι ιδιαίτερα δύσκολο να αποσταλούν παλαιότερα δεδομένα, τα οποία οι πελάτες συνειδητοποιούν αργότερα ότι τα χρειάζονται. Επίσης, η προσέγγιση που εφαρμόστηκε στο έργο *Ministrel* [25] ακολουθεί το υβριδικό μοντέλο, αφού οι πελάτες ειδοποιούνται σχετικά με τη διαθεσιμότητα νέων δεδομένων μέσω μικρών μηνυμάτων που αποστέλλονται με τη βοήθεια της μεθόδου ώθησης, ενώ κάνουν χρήση της μεθόδου άντλησης για ανάκτηση των πραγματικών δεδομένων.

2.3.3 ΣΥΧΝΟΤΗΤΑ ΔΙΑΝΟΜΗΣ ΔΕΔΟΜΕΝΩΝ

Η διανομή δεδομένων, ανάλογα με τη συχνότητα που εκτελείται, χαρακτηρίζεται ως *περιοδική, υπό συνθήκη ή κατόπιν αιτήματος* [15]. Οι κατηγορίες αυτές περιγράφονται στη συνέχεια.

2.3.3.1 Περιοδική Διανομή Δεδομένων (*Periodic Data Delivery*)

Τα δεδομένα διανέμονται περιοδικά από τους εξυπηρετητές στους πελάτες. Η περίοδος ορίζεται συνήθως από το σύστημα ή τους πελάτες, διαμέσου των προφίλ τους. Τόσο ο μηχανισμός άντλησης όσο και ο μηχανισμός ώθησης μπορούν να εκτελούνται σε περιοδική βάση. Σε αυτές τις περιπτώσεις η διανομή πραγματοποιείται επαναλαμβανόμενα, σε προκαθορισμένους χρόνους. Η μέθοδος περιοδικής άντλησης, γνωστή και ως *μέθοδος σφυγμομέτρησης (polling)*, αποτελεί την απλούστερη περίπτωση του πρωτοκόλλου Αιτήματος-Απόκρισης. Στην περίπτωση της περιοδικής ώθησης, η πληροφορία αποστέλλεται στους πελάτες περιοδικά, σύμφωνα με προκαθορισμένο σχέδιο. Εάν ένας πελάτης χάσει κάποια πληροφορία, είτε γιατί δεν ήταν διαθέσιμος την ώρα της αποστολής, είτε γιατί δεν πρόλαβε να αντιδράσει σε αυτό που έχει σταλεί, μπορεί να λάβει τη πληροφορία αυτή κατά την επόμενη αποστολή.

2.3.3.2 Υπό Συνθήκη Διανομή Δεδομένων (*Conditional Data Delivery*)

Τα δεδομένα αποστέλλονται από τους εξυπηρετητές όταν ικανοποιούνται συγκεκριμένες συνθήκες, οι οποίες έχουν οριστεί από τους πελάτες στα προφίλ τους. Η υπό συνθήκη διανομή εφαρμόζεται κυρίως στη μέθοδο ώθησης και την υβριδική. Στις περιπτώσεις αυτές, η πληροφορία αποστέλλεται εάν και μόνο ικανοποιείται μια προκαθορισμένη συνθήκη. Μια εφαρμογή που στέλνει ως πληροφορία τις τιμές των αποθεμάτων μόνο όταν αυτές αλλάζουν είναι ένα παράδειγμα διανομής με τη μέθοδο της ώθησης υπό συνθήκη. Αντίθετα, μια εφαρμογή που αποστέλλει την αναφορά ισολογισμού μόνο όταν ο συνολικός ισολογισμός ξεπεράσει το κάτω όριο κατά 5% είναι ένα παράδειγμα

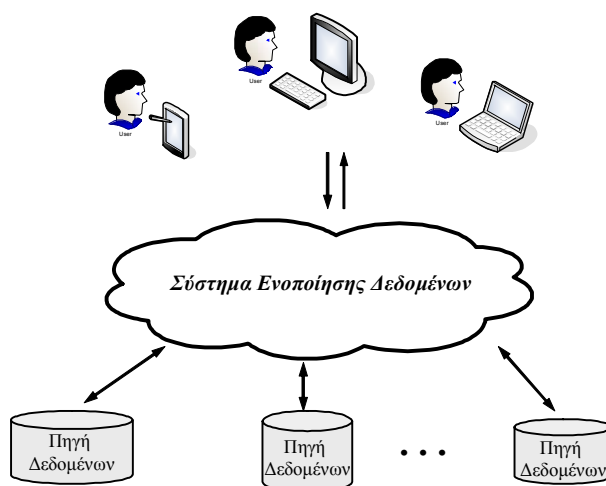
υβριδικής διανομής υπό συνθήκη. Η διανομή ώθησης υπό συνθήκη υποθέτει ότι οι αλλαγές είναι κρίσιμες για τους πελάτες και οι πελάτες απαιτείται να παρακολουθούν αυτά που αποστέλλονται. Αντίθετα, η υβριδική διανομή υπό συνθήκη υποθέτει ότι η απώλεια κάποιων πληροφοριών δεν είναι κρίσιμη για τους πελάτες.

2.3.3.3 Διανομή Δεδομένων Κατόπιν Αιτήματος (*Ad-Hoc Data Delivery*)

Κατόπιν αιτήματος του πελάτη, οι σχετικές πληροφορίες αποστέλλονται από τον εξυπηρετητή στον πελάτη σε λογικό χρόνο απόκρισης. Συνεπώς, αυτός ο τύπος διανομής πραγματοποιείται ακανόνιστα και συναντάται συνήθως σε συστήματα άντλησης, τα οποία χρησιμοποιούν το πρωτόκολλο Αιτήματος-Απόκρισης.

2.4 ΕΝΟΠΟΙΗΣΗ ΔΕΔΟΜΕΝΩΝ (DATA INTEGRATION)

Ο τομέας *Ενοποίησης Δεδομένων* ασχολείται με το πρόβλημα συνδυασμού δεδομένων που προέρχονται από διαφορετικές αυτόνομες πηγές, με σκοπό να παρέχει στους χρήστες ενοποιημένες και ενδεχομένως εμπλουτισμένες όψεις των πηγών, καθώς και τα μέσα για τον καθορισμό των αιτημάτων πληροφορίας που απαιτούν συσχετισμό δεδομένων από πολλές διαφορετικές πηγές. Ένα *Σύστημα Ενοποίησης Δεδομένων* παρέχει τα μέσα για τον καθορισμό των ενοποιημένων όψεων των πηγών και την επεξεργασία των σχετικών αιτημάτων. Σκοπός του είναι να αποκρύψει την πολυπλοκότητα των πολλαπλών διαφορετικών πηγών, παρουσιάζοντας στους χρήστες μια διεπαφή για ενιαία πρόσβαση στα δεδομένα όλων των πηγών. Όπως φαίνεται και στο Σχήμα 2.9, δεν υπάρχει συγκεκριμένη αρχιτεκτονική για τα συστήματα ενοποίησης δεδομένων, αλλά ούτε και τυποποιημένη τεχνολογία υλοποίησής τους.



Σχήμα 2.9: Σύστημα Ενοποίησης Δεδομένων

Οι πηγές δεδομένων αποτελούν συλλογές αποθηκευμένων δεδομένων ή δεδομένων που προκύπτουν βάσει υπολογισμών, για τις οποίες υπάρχει προγραμματιστική πρόσβαση και

επίσης είναι δυνατόν κανείς να ανακτήσει ή να συμπεράνει την περιγραφή της δομής των δεδομένων, η οποία είναι γνωστή ως *σχήμα (schema)*, καθώς και πρόσθετες πληροφορίες για τις πηγές. Όλες οι πληροφορίες που αφορούν στο περιεχόμενο μιας πηγής (σχήμα, μέγεθος δεδομένων, κ.λπ.), στις υπολογιστικές ικανότητές της (διεπαφή πρόσβασης στα δεδομένα), καθώς και πληροφορίες όπως η αξιοπιστία της πηγής ή η ποιότητα των πληροφοριών που παράγει καλούνται *μετα-δεδομένα πηγής (source meta-data)*.

Ένα σημαντικό χαρακτηριστικό των πηγών είναι ότι δεν υπάρχει μια γενική μέθοδος ανάκτησης του σχήματος μιας πηγής ή σύνδεσης ενός δεδομένου σχήματος με κάποια πηγή. Μερικές πηγές, όπως αυτές των συστημάτων RDBMS (*Relational Database Management System*), αποθηκεύουν και παρέχουν το σχήμα τους ως μέρος της ίδιας της πηγής, αλλά χωριστά από τα πραγματικά δεδομένα. Σε άλλες περιπτώσεις, όπως στα έγγραφα XML (*eXtensible Markup Language*) και RDF (*Resource Description Framework*), τα σύνολα δεδομένων είναι αυτοπεριγραφικά (*self-descriptive*) και η πληροφορία σχήματος ενυπάρχει μέσα στα σύνολα δεδομένων. Τέλος, πηγές όπως οι ιστοσελίδες μπορεί να μην παρέχουν σχήμα, αλλά είναι δυνατόν να αναπτυχθούν μέθοδοι για ανάλυση των δεδομένων τους και εξαγωγή της δομής τους. Τα πιο σημαντικά χαρακτηριστικά των πηγών περιγράφονται στη συνέχεια.

- **Ετερογένεια.** Η ετερογένεια των πηγών εκδηλώνεται σε διάφορα επίπεδα. Διακρίνουμε τα εξής τρία γενικά επίπεδα [26]:
 1. *Ετερογένεια πλατφορμών.* Οι πηγές διαφέρουν στο λειτουργικό σύστημα και το υλισμικό που χρησιμοποιούν, στη φυσική αναπαράσταση των δεδομένων, στις μεθόδους επίκλησης των λειτουργιών που παρέχουν την προγραμματιστική πρόσβαση στα δεδομένα τους και στα πρωτόκολλα δικτύου.
 2. *Ετερογένεια συστημάτων.* Σε αυτό το επίπεδο οι πηγές δεδομένων διαφέρουν κυρίως σε δύο σημεία. Πρώτον, είναι δυνατόν να χρησιμοποιούν διαφορετικά σύνολα εννοιών ή *μοντέλα δεδομένων (data models)* για την μοντελοποίηση των διαφόρων οντοτήτων στον πραγματικό κόσμο. Δεύτερον, ποικίλες μέθοδοι μπορούν να χρησιμοποιηθούν για την πρόσβαση στα δεδομένα και το χειρισμό τους. Αυτές οι μέθοδοι, γνωστές και ως *ικανότητες πηγών (source capabilities)*, μπορούν να ποικίλουν, από μια γλώσσα διατύπωσης ερωτήσεων, όπως *SQL*, έως μια μέθοδο σειριακής ανίχνευσης αρχείων.
 3. *Ετερογένεια πληροφορίας.* Αυτό το είδος ετερογένειας αφορά τα ίδια τα δεδομένα. Το περιεχόμενο των πηγών μπορεί να διαφέρει σε λογικό επίπεδο, επειδή υπάρχουν πολλοί τρόποι μοντελοποίησης του πραγματικού κόσμου. Η επίλυση αυτού του τύπου ετερογένειας καλείται *ενοποίηση σχημάτων (schema integration)*. Διακρίνονται δυο τύποι ετερογένειας πληροφορίας: η *σημασιολογική*

(*semantic*) και η *δομική* (*structural*). Ο πρώτος τύπος οφείλεται στο γεγονός ότι διαφορετικές έννοιες στον πραγματικό κόσμο μπορούν να συσχετιστούν με διαφορετικές έννοιες σε επίπεδο πηγών, όπως για παράδειγμα όταν χρησιμοποιούνται διαφορετικά ονόματα για το ίδιο πράγμα ή το ίδιο όνομα για διαφορετικά πράγματα. Αντίθετα, η δομική ετερογένεια σχετίζεται με τη χρήση διαφορετικών εννοιών σε επίπεδο μοντέλου δεδομένων, όπως διαφορετικών τύπων δεδομένων, για τη μοντελοποίηση των ίδιων οντοτήτων στον πραγματικό κόσμο.

- **Αυτονομία.** Εξαιτίας είτε οργανωτικών είτε τεχνικών λόγων, οι πηγές δεδομένων είναι συνήθως ανεξάρτητες και επιπλέον δεν γνωρίζουν η μία την ύπαρξη της άλλης. Αυτή η ανεξαρτησία αναφέρεται ως αυτονομία και σχετίζεται με την κατανομή του ελέγχου και όχι των δεδομένων. Υπό την οργανωτική έννοια, η αυτονομία συνεπάγεται ότι οι πηγές ελέγχονται από ανεξάρτητα πρόσωπα ή ομάδες, ενώ υπό την τεχνική της έννοια η αυτονομία σχετίζεται με την κατανομή του ελέγχου [27].
- **Κατανομή.** Οι πηγές δεδομένων συνήθως ανήκουν σε διαφορετικούς κόμβους και έτσι διανέμονται φυσικά. Δεδομένου ότι οι πηγές μπορούν όχι μόνο να αποθηκεύουν αλλά και να παράγουν δεδομένα βάσει υπολογισμών, η έννοια της κατανομής των πηγών αναφέρεται τόσο στη κατανομή δεδομένων όσο και στην κατανομή λειτουργίας.

Κατά συνέπεια, στα πλαίσια της ενοποίησης δεδομένων, καλούνται να αντιμετωπιστούν θέματα που αφορούν στην πρόσβαση στα δεδομένα, την ενοποίηση των δεδομένων σε διαφορετικά επίπεδα αφαίρεσης, την εξαγωγή των μετα-δεδομένων, και το συσχετισμό δεδομένων που προέρχονται από διαφορετικές πηγές. Φυσικά, όλες αυτές οι διαδικασίες πρέπει να εκτελούνται μέσα σε λογικά όρια από πλευράς χρόνου και εξάντλησης πόρων του συστήματος, και επομένως ένα σημαντικό ζήτημα για οποιαδήποτε λύση υιοθετηθεί είναι η αποδοτικότητα και η επεκτασιμότητα του συστήματος σε σχέση με το πλήθος των πηγών και τον όγκο των δεδομένων. Για λόγους απλότητας, οι προσεγγίσεις που ακολουθούνται κατηγοριοποιούνται σύμφωνα με τα εξής δύο υποδείγματα: το υπόδειγμα που βασίζεται στη *διαχείριση πολλαπλών βάσεων δεδομένων* και αυτό που βασίζεται σε *μεσίτες* [1].

2.4.1 ΥΠΟΔΕΙΓΜΑ ΠΟΥ ΒΑΣΙΖΕΤΑΙ ΣΤΗ ΔΙΑΧΕΙΡΙΣΗ ΠΟΛΛΑΠΛΩΝ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ (MULTIDATABASE MANAGEMENT-BASED PARADIGM)

Μια κλασική προσέγγιση [28, 29] της διαχείρισης πολλαπλών βάσεων βασίζεται στη δημιουργία ενός ενιαίου γενικού σχήματος, με σκοπό να καλύψει αυτό τις διαφορές μεταξύ των πολλαπλών τοπικών σχημάτων των επιμέρους βάσεων. Η μετατροπή από κάθε τοπικό

σχήμα στο ενιαίο εκφράζεται συχνά σε μια γλώσσα κοινότυπη της SQL. Αν και η επιβολή ενός ενιαίου γενικού σχήματος επιτυγχάνει πλήρη διαφάνεια για ενιαία πρόσβαση στα δεδομένα, οι επιμέρους βάσεις, στην περίπτωση αυτή, έχουν ιδιαίτερα περιορισμένη αυτονομία και επεκτασιμότητα, και η εξέλιξή τους πραγματοποιείται με δυσκολία.

Η προσέγγιση που περιγράφεται στο [29] βελτιώνει την αυτονομία και την ευελιξία της διαχείρισης πολλαπλών βάσεων, βασιζόμενη σε πολλαπλά σχήματα και την εξατομικευμένη ενσωμάτωσή τους σε διάφορα επίπεδα των πολλαπλών βάσεων. Εντούτοις, η ενσωμάτωση των επιμέρους σχημάτων σε κάθε επίπεδο πραγματοποιείται από το ίδιο το σύστημα. Το συνολικό σχήμα είναι στατικό και τα προβλήματα ετερογένειας επιλύονται στο στάδιο ενσωμάτωσης των επιμέρους σχημάτων. Αυτή η προσέγγιση παρουσιάζει προβλήματα επεκτασιμότητας όταν νέες πηγές πρέπει να προστεθούν σε ένα υπάρχον σύστημα πολλαπλών βάσεων. Επίσης, τα επιμέρους σχήματα δεν μπορούν να εξελιχθούν χωρίς τη συγκατάθεση του συνολικού σχήματος.

Η προσέγγιση της *κατανεμημένης διαχείρισης αντικειμένων (distributed object management)* [30, 31] γενικεύει την προηγούμενη προσέγγιση, μοντελοποιώντας τις ετερογενείς βάσεις ως αντικείμενα σε ένα χώρο κατανεμημένων αντικειμένων. Αυτό απαιτεί τον ορισμό ενός ενιαίου μοντέλου αντικειμένων και μιας κοινής γλώσσας διατύπωσης ερωτήσεων μεταξύ των αντικειμένων. Οι δραστηριότητες του προτύπου OMG και του ODMG [32], το οποίο επεκτείνει το OMG ώστε να καλύψει τη διαλειτουργικότητα μεταξύ διαφορετικών βάσεων δεδομένων, θεωρούνται σημαντικά ορόσημα στην κατανεμημένη διαχείριση αντικειμένων.

2.4.2 ΥΠΟΔΕΙΓΜΑ ΠΟΥ ΒΑΣΙΖΕΤΑΙ ΣΕ ΜΕΣΙΤΕΣ (MEDIATOR-BASED PARADIGM)

Το υπόδειγμα αυτό προέκυψε από μια αρχική πρόταση του Gio Wiederhold, η οποία περιγράφεται στο [33] και διαμορφώθηκε μέσω του προγράμματος ευφυούς ενοποίησης πληροφοριών (Intelligent Information Integration - I³) [34]. Ένα σύστημα μεσιτείας αποτελείται από σύνολο μεσιτών. Ο μεσίτης είναι μια μονάδα λογισμικού που εκμεταλλεύεται κωδικοποιημένη γνώση για ορισμένα σύνολα ή υποσύνολα δεδομένων με σκοπό να δημιουργήσει πληροφορία για ένα υψηλότερο επίπεδο εφαρμογών [33]. Διαισθητικά, κάθε μεσίτης προσφέρει μια συγκεκριμένη υπηρεσία δεδομένων και αποτελεί *μεσίτη πληροφορίας (information broker)* σε συγκεκριμένη περιοχή εφαρμογών.

Στη συνέχεια περιγράφονται τα πιο ενδιαφέροντα σημεία στα οποία διαφοροποιείται το υπόδειγμα που βασίζεται σε μεσίτες από αυτό της διαχείρισης πολλαπλών βάσεων δεδομένων.

Από πλευράς πηγών πληροφορίας:

- Ο κόσμος πληροφορίας στον οποίο απευθύνεται είναι ευρείας κλίμακας, δυναμικός και «ανοιχτός» στη φύση του, αντί μικρής κλίμακας, στατικός και «κλειστός».
- Οι πηγές πληροφορίας, εκτός από δομημένες (π.χ. σχεσιακές βάσεις δεδομένων, βάσεις γνώσης) μπορεί να είναι και ημιδομημένες (π.χ. HTML αρχεία, σελίδες του παγκόσμιου ιστού) ή μη-δομημένες (π.χ. κοινά έγγραφα).
- Οι πηγές είναι ιδιαίτερα αυτόνομες. Το περιεχόμενο, ο αριθμός και η συνδεσιμότητα των πηγών μπορούν να αλλάζουν συνεχώς, με αποτέλεσμα τα προβλήματα ετερογένειας να αποτελούν φυσική συνέπεια.

Από πλευράς χειρισμού διαλειτουργικότητας:

- Το σύνολο των λειτουργιών που σχετίζονται με το χειρισμό της διαλειτουργικότητας διαιρείται αρχιτεκτονικά σε δυο επίπεδα: το *επίπεδο μεσίτη (mediator-tier)* και το *επίπεδο μετατροπέα (wrapper-tier)*. Το επίπεδο μεσίτη είναι υπεύθυνο για την αλληλεπίδραση με τις εφαρμογές και τους τελικούς χρήστες, ενώ το επίπεδο μετατροπέα αναλαμβάνει το χειρισμό της αλληλεπίδρασης με τις υποκείμενες πηγές. Ο μετατροπέας θα μπορούσε να θεωρηθεί ειδικός τύπος μεσίτη που χειρίζεται τα ιδιαίτερα χαρακτηριστικά της κάθε πηγής, όπως τη μετάφραση ενός αιτήματος του μεσίτη σε σύνολο εκτελέσιμων εντολών στην πηγή.
- Δεν υπάρχει ένα μοναδικό, καθολικό σύστημα μεσιτείας που να μπορεί να εξυπηρετήσει όλες τις εφαρμογές. Αντίθετα, διαφορετικά συστήματα μεσιτείας πραγματοποιούν την πρόσβαση στις κοινές πηγές πληροφορίας. Η διαφορετικότητα των πηγών που προσπελάζονται από ένα συγκεκριμένο σύστημα μεσιτείας «καλύπτεται» χρησιμοποιώντας τη γλώσσα διεπαφής του συγκεκριμένου συστήματος. Συνεπώς, ο μετατροπέας χρησιμεύει ως πράκτορας ενός συγκεκριμένου συστήματος μεσιτείας που αναλαμβάνει να επικοινωνεί με τις υποκείμενες πηγές.
- Η αρχή της επεκτασιμότητας αποτελεί κύριο ενδιαφέρον και σημαντικό κριτήριο αποτίμησης του χειρισμού της διαλειτουργικότητας.
- Οντολογίες και ιεραρχίες ταξινόμησης βοηθούν στην επίλυση προβλημάτων ετερογένειας και διαλειτουργικότητας μεταξύ καταναλωτών και παραγωγών πληροφορίας.

Για τους παραπάνω λόγους, το υπόδειγμα που βασίζεται σε μεσίτες θεωρείται πιο ευέλικτο, υποστηρίζοντας αυτόνομες και δυναμικές πηγές, οι οποίες ενδεχομένως παράγουν δεδομένα βάσει υπολογισμών και κατόπιν απαίτησης και συνεπώς δεν τα αποθηκεύουν. Ο χειρισμός της διαλειτουργικότητας μέσω της προσέγγισης μεσίτη-μετατροπέα παρέχει καλύτερη επεκτασιμότητα, δεδομένου ότι ο κάθε μετατροπέας δομείται ανεξάρτητα και εξυπηρετεί κάθε είδους πρόσβαση στις πηγές που του αναλογούν. Υιοθετώντας λοιπόν το υπόδειγμα μεσιτείας για την ανάπτυξη συστημάτων ενοποίησης δεδομένων, οι αρχές και τα

προβλήματα των μεσιτικών συστημάτων περιγράφονται πιο αναλυτικά στην επόμενη ενότητα.

2.5 ΜΕΣΙΤΙΚΑ ΣΥΣΤΗΜΑΤΑ (BROKERAGE SYSTEMS)

Στα συστήματα διανομής πληροφορίας, ο χρήστης δεν θα πρέπει να επιβαρύνεται με καθήκοντα που σχετίζονται με την αναζήτηση και προσπέλαση των πηγών πληροφορίας. Τέτοιου είδους αποστολές, ειδικά εάν είναι επαναλαμβανόμενες, πραγματοποιούνται συνήθως μέσω της αλληλεπίδρασης των προγραμμάτων εφαρμογής με αυτοματοποιημένα προγράμματα διαμεσολάβησης. Πιο συγκεκριμένα, στα συστήματα μεσιτείας ή διαμεσολάβησης θεωρούνται τρεις διαφορετικές οντότητες:

1. *Πηγές δεδομένων*, οι οποίες παρέχουν τα βασικά δεδομένα προς διανομή.
2. *Πελάτες*, οι οποίοι αποτελούν τους καταναλωτές των πληροφοριών.
3. *Μεσίτες πληροφοριών* (ή πράκτορες, ή μεσολαβητές, κ.λπ.), οι οποίοι ανακτούν πληροφορίες από τις πηγές, προσθέτουν ενδεχομένως αξία σε αυτές (π.χ. κάποιο πρόσθετο υπολογισμό ή οργανωτική δομή), και έπειτα τις διανέμουν σε άλλους μεσίτες ή πελάτες.

Ο όρος «διαμεσολάβηση» περιλαμβάνει την επεξεργασία που απαιτείται ώστε οι διεπαφές μεταξύ πηγών και εφαρμογών χρηστών να λειτουργούν, τις γνωστικές δομές που καθοδηγούν τους μετασχηματισμούς των απλών δεδομένων σε πληροφορία, εναποθηκευτικούς μηχανισμούς, υπηρεσίες δικτύου, συνιστώσες προγραμματισμού πρόσβασης στις πηγές και μηχανές αναζήτησης [33]. Οι κύριες υπηρεσίες που προσφέρει ένας μεσίτης είναι οι εξής:

- Πρόσβαση και ανάκτηση πληροφορίας από πολλαπλές πηγές.
- Αφαίρεση και μετασχηματισμός της πληροφορίας που ανακτάται σε μια ενιαία απεικόνιση και σημασιολογία.
- Ενοποίηση και συναρμογή των ομογενοποιημένων στοιχείων.
- Μείωση του όγκου της συνενωμένης πληροφορίας βάσει αφαίρεσης.

2.5.1 ΑΡΧΙΤΕΚΤΟΝΙΚΑ ΜΟΝΤΕΛΑ ΜΕΣΙΤΕΙΑΣ

Τα συστήματα μεσιτείας ταξινομούνται με βάση τον αριθμό και την κατανομή των μεσιτών σε δυο βασικές κατηγορίες: στα συστήματα *κεντρικής μεσιτείας* και σε αυτά των *ομότιμων μεσιτών*, καθένα από τα οποία περιγράφεται στη συνέχεια.

2.5.1.1 Κεντρική Μεσιτεία

Η απλούστερη δυνατή προσέγγιση σχεδιασμού ενός στρώματος διαμεσολάβησης προκύπτει από την ενσωμάτωση ολόκληρης της λειτουργίας διαμεσολάβησης σε έναν κεντρικό μεσίτη, ο οποίος αλληλεπιδρά με τις πηγές μέσω των μετατροπών. Αυτός ο τύπος

μεσιτείας αναφέρεται στην αρχιτεκτονική τριών επιπέδων, όπου ο μεσίτης αντιστοιχεί στο μεσαίο επίπεδο.

Οι εφαρμογές που βασίζονται σε συστήματα κεντρικής μεσιτείας έχουν ισχυρές απαιτήσεις από πλευράς αξιοπιστίας και συνοχής δεδομένων ή υποστήριξης συναλλαγών, αλλά δεν απαιτούν υψηλούς ρυθμούς διέλευσης δεδομένων. Παραδείγματα τέτοιων εφαρμογών αποτελούν οι εφαρμογές ηλεκτρονικού εμπορίου και τραπεζικών συναλλαγών. Ένα σύστημα κεντρικής μεσιτείας έχει τα εξής κύρια χαρακτηριστικά:

- Μόνο η αυτονομία των ιδιοκτητών των πηγών μπορεί να διατηρηθεί, και όχι των εμπειρογνομόνων των επιμέρους περιοχών που ενοποιούν τις πηγές, οι οποίοι θα πρέπει να αποκαλύψουν όλη την εξειδικευμένη γνώση τους στον ιδιοκτήτη του μεσίτη.
- Απαιτεί κεντρική διαχείριση και επομένως συντονισμό μεταξύ πολλών ανεξάρτητων συμβαλλόμενων μερών, ο οποίος μπορεί να μην είναι πάντα εφικτός.
- Δεδομένου ότι ένας κεντρικός μεσίτης καλύπτει πολλές γνωστικές περιοχές και επιχειρεί πρόσβαση σε μεγάλο αριθμό πηγών διαφορετικών τύπων, είναι εξαιρετικά σύνθετο και δύσκολο να συντηρηθεί.
- Ο κεντρικός μεσίτης αποτελεί το μοναδικό σημείο πιθανής αποτυχίας.

Εν συντομία, η κεντρική μεσιτεία απαιτεί κεντρική διαχείριση και συγκέντρωση όλης της γνώσης που απαιτείται για ενοποίηση των πηγών σε ένα σημείο, κάτι που είναι εξαιρετικά δύσκολο να επιτευχθεί σε συστήματα μεγάλης κλίμακας. Ωστόσο, εγγυάται υψηλή συνέπεια και αξιοπιστία στη διανομή των δεδομένων.

2.5.1.2 Ομότιμοι Μεσίτες

Ένα σύστημα μεσιτείας μπορεί να αποτελείται από πολλούς κατανεμημένους μεσίτες, καθένας από τους οποίους ενοποιεί ένα μικρό υποσύνολο όλων των διαθέσιμων πηγών και μοιράζεται τις αφαιρέσεις των δεδομένων τους με εφαρμογές υψηλότερου επιπέδου. Η αρχιτεκτονική ομότιμων μεσιτών αποτελεί φυσικό τρόπο σχεδίασης ενός κατανεμημένου συστήματος μεσιτείας που απαιτεί αυτονομία και αποκέντρωση, παρέχοντας σύνολο αυτόνομων ομότιμων μεσιτών, οι οποίοι αλληλεπιδρούν μεταξύ τους, αλλά και με τις πηγές δεδομένων και τις εφαρμογές χρηστών.

Οι μεσίτες συνεργάζονται άμεσα ο ένας με τον άλλο και όλος ο έλεγχος, τα δεδομένα και τα μετα-δεδομένα κατανέμονται μεταξύ όλων των μεσιτών. Κάθε μεσίτης επιλέγει τους ομότιμους μεσίτες με τους οποίους θέλει να συνεργαστεί (ως πελάτες ή/και ως εξυπηρετητές του), διατηρώντας γνώση για ένα υποσύνολο όλων των διαθέσιμων μεσιτών. Επιπλέον, προγραμματίζει τοπικά τις ενέργειές του με βάση την τοπική γνώση του. Ο μόνος τρόπος απόκτησης δεδομένων και μετα-δεδομένων από διαφορετικούς μεσίτες είναι με την αποστολή αιτημάτων, συνήθως στη μορφή ερωτήσεων, σε γνωστούς μεσίτες, οι οποίοι με τη σειρά τους

μπορούν να προωθούν τις ερωτήσεις σε άλλους μεσίτες. Οι μεσίτες μπορούν να εισέρχονται στο σύστημα και να αποχωρούν από αυτό οποιαδήποτε στιγμή.

Επιπλέον, δεν υπάρχει μια και μοναδική ενοποιημένη όψη των πηγών, όπως στα συστήματα πολλαπλών βάσεων δεδομένων και κεντρικής μεσιτείας. Αντίθετα, κάθε μεσίτης προσδιορίζει τη δική του ενοποιημένη όψη για ένα υποσύνολο των πηγών και καθιστά μέρος αυτής της όψης διαθέσιμη σε άλλους μεσίτες και εφαρμογές για περαιτέρω ενοποίηση ή άντληση πληροφοριών.

Τα πλεονεκτήματα της αρχιτεκτονικής ομότιμων μεσιτών είναι ότι επιτρέπει στους εμπειρογνώμονες των επιμέρους περιοχών να ελέγχουν ανεξάρτητα τους μεσίτες τους, όπως ακριβώς οι ιδιοκτήτες πηγών έχουν το συνολικό έλεγχο των πηγών τους. Κάθε μεσίτης μπορεί να εξελιχθεί ανεξάρτητα, δεδομένου ότι διατηρεί τη δημόσια διεπαφή του. Η αρχιτεκτονική ομότιμων μεσιτών επιτρέπει κατανομή της προσπάθειας ενοποίησης πηγών μεταξύ πολλών αυτόνομων εμπειρογνομώνων, διευκολύνοντας την κλιμάκωση στη διαδικασία της ενοποίησης. Η κωδικοποιημένη γνώση των μεσιτών διαμοιράζεται, έτσι ώστε πιο σύνθετοι μεσίτες να μπορούν να συντεθούν από απλούστερους, ενσωματώνοντας δεδομένα από πολλές πηγές και γνωστικές περιοχές κατά τρόπο εξελικτικό. Επίσης, είναι δυνατή η επαναχρησιμοποίηση υπολογιστικών πόρων, όπως αποθηκευτικών, επεξεργασίας, εξειδικευμένου λογισμικού και υλισμικού, και επιπλέον κανένας μεσίτης δεν αποτελεί καθολικό σημείο συμφόρησης. Η αρχιτεκτονική ομότιμων μεσιτών θεωρείται ιδιαίτερα κατάλληλη για εφαρμογές που απαιτούν γρήγορη και αποδοτική διανομή ευμετάβλητων δεδομένων, όπως χρηματιστηριακές εφαρμογές ή ευρείας εκπομπής πολυμέσων [16].

2.5.2 ΡΟΛΟΣ ΜΕΣΙΤΕΙΑΣ

2.5.2.1 Διαλειτουργικότητα (*Interoperability*)

Η ενοποίηση των δεδομένων στα συστήματα μεσιτείας πραγματοποιείται σε δύο βασικά στάδια. Το πρώτο στάδιο, η *αντιστοίχιση των μοντέλων δεδομένων (data model mapping)*, προσδιορίζει τον τρόπο ανάκτησης των δεδομένων από κάθε μια από τις πηγές, καθώς και τον τρόπο μετατροπής των δεδομένων της κάθε πηγής στο μοντέλο δεδομένων του μεσιτικού συστήματος. Συνεπώς, αυτό το στάδιο χειρίζεται την ετερογένεια των πηγών σε επίπεδο συστήματος και παρέχει ομοιόμορφη αναπαράσταση όλων των πηγών με βάση το κοινό μοντέλο δεδομένων του μεσίτη. Από αρχιτεκτονικής πλευράς, το επίπεδο μετατροπέα (*wrapper-tier*) είναι υπεύθυνο για το στάδιο αυτό. Το δεύτερο στάδιο, η *ενοποίηση των σχημάτων (schema integration)*, χειρίζεται την ετερογένεια των πληροφοριών που παράγουν οι πηγές σε λογικό επίπεδο. Κατά τη διάρκεια αυτού του σταδίου, αντιστοιχίζονται ίδια αντικείμενα από διαφορετικές πηγές και επιλύονται συγκρούσεις σε επίπεδο σχημάτων ή δεδομένων με τη βοήθεια του επιπέδου μεσίτη (*mediator-tier*). Σε αυτή τη φάση, το κοινό

μοντέλο δεδομένων και η γλώσσα διατύπωσης ερωτήσεων του μεσίτη μπορούν να χρησιμοποιηθούν για να καθορίσουν τις ενοποιημένες όψεις των πηγών, παρέχοντας ενιαία διεπαφή σε όλες τις ενοποιημένες πηγές. Τα αιτήματα των χρηστών για πληροφορία εκφράζονται στη γλώσσα διατύπωσης ερωτήσεων του μεσίτη.

2.5.2.2 Διαφάνεια Δικτύου (*Network Transparency*)

Ο μεσίτης μπορεί να λειτουργεί σαν πελάτης σε κάποιες πηγές δεδομένων, συλλέγοντας και ενδεχομένως «ξαναπακετώντας» τα δεδομένα που λαμβάνει, και στη συνέχεια να λειτουργεί ως νέα πηγή δεδομένων του συστήματος. Η δυνατότητα των μεσιτών να λειτουργούν ταυτόχρονα ως πελάτες και πηγές δεδομένων παρέχει τη βάση για την έννοια της διαφάνειας δικτύου. Οι δέκτες πληροφοριών δεν μπορούν να ανιχνεύουν τις λεπτομέρειες των διασυνδέσεων πέρα από τον άμεσο προκάτοχό τους. Λόγω αυτής της διαφάνειας, ο μηχανισμός διανομής δεδομένων που χρησιμοποιείται μεταξύ δύο ή περισσότερων κόμβων μπορεί να αλλάζει, χωρίς να απαιτείται αλλαγή των μηχανισμών που χρησιμοποιούνται για κάποια άλλη επικοινωνία στο σύστημα. Για παράδειγμα, υποθέτουμε ότι ο κόμβος Β λαμβάνει τις τιμές στοιχείων από τον κόμβο Α μετά από αντίστοιχο αίτημα. Υποθέτουμε επιπλέον ότι ο κόμβος Γ λαμβάνει δεδομένα από τον κόμβο Β μέσω περιοδικής ευρείας εκπομπής, η οποία περιλαμβάνει τις τιμές που ο Β έχει προηγουμένως λάβει από τον Α. Εάν ο Α αρχίσει να ωθεί τις τιμές στον Β, δεν είναι απαραίτητο ο κόμβος Γ να αλλάξει τον τρόπο συλλογής δεδομένων του. Συνεπώς, αλλαγές που συμβαίνουν σε συγκεκριμένες συνδέσεις ενδιαφέρουν μόνο τους κόμβους που συνδέονται άμεσα. Επιπρόσθετα, αυτή η διαφάνεια επιτρέπει στον τρόπο διανομής των δεδομένων σε οποιοδήποτε κόμβο να διαφέρει από τον τρόπο που τα δεδομένα αυτά έχουν νωρίτερα διανεμηθεί στο δίκτυο.

2.5.2.3 Συνδυασμός Πολλαπλών Μεθόδων Διανομής Πληροφορίας (*Combination of Multiple Data Delivery Modes*)

Ένας μεσίτης έχει τη δυνατότητα να συνδυάζει πολλούς μηχανισμούς διανομής πληροφορίας με τρόπο ευέλικτο, ο οποίος υπαγορεύεται από τα χαρακτηριστικά της εκάστοτε εφαρμογής. Σκοπός είναι να πετύχει την αποδοτικότερη χρήση των διαθέσιμων πόρων από πλευράς εξυπηρετητών και επικοινωνίας, ώστε να ενισχύσει τη δυνατότητα κλιμάκωσης και την αποτελεσματικότητα της εφαρμογής. Δεδομένων των πηγών που ψάχνουν για ενδιαφερόμενους χρήστες και των χρηστών που αναζητούν σχετικές πληροφορίες, αναζητείται ένας αποδοτικός και αξιόπιστος τρόπος για την εκτέλεση της αντιστοίχισης και εγκατάσταση της ροής πληροφοριών μεταξύ πηγών και πελατών. Ένας απλός τρόπος είναι οι χρήστες να στέλνουν τα αιτήματά τους σε κάθε πηγή που είναι πιθανό να παράγει χρήσιμες πληροφορίες. Ωστόσο, ο μεγάλος αριθμός πηγών καθιστά αυτόν τον τρόπο ασύμφορο. Επιπλέον, θα ήταν πολύ δύσκολο να βρεθούν όλες οι ενδεχομένως σχετικές πηγές. Ο άλλος

ακραίος τρόπος είναι οι πηγές να στέλνουν τις πληροφορίες τους σε κάθε χρήστη, και οι χρήστες στη συνέχεια να απομακρύνουν τις άσχετες πληροφορίες. Και αυτός ο τρόπος όμως παρουσιάζει προβλήματα κλιμάκωσης, αφού πολύτιμο εύρος ζώνης χάνεται για τη μετάδοση άσχετων πληροφοριών και επιπλέον πραγματοποιείται άσκοπη τοπική επεξεργασία. Ο ρόλος του μεσίτη, στην περίπτωση αυτή, είναι να διευκολύνει την αποδοτική μετακίνηση της πληροφορίας από τις πηγές στους πελάτες, εξατομικεύοντας τον τρόπο διανομής ανά περίπτωση.

2.5.3 ΠΡΟΒΛΗΜΑΤΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ ΕΠΙΔΟΣΗΣ

Τα μοντέλα μεσιτείας παρουσιάζουν ιδιαίτερο ενδιαφέρον ως προς τη μελέτη μηχανισμών για την παροχή κλιμακούμενων και αποτελεσματικών υπηρεσιών διανομής πληροφορίας σε ανοιχτά, κατανεμημένα περιβάλλοντα. Ωστόσο, η παρεμβολή του στρώματος διαμεσολάβησης σε ένα πληροφοριακό σύστημα απαιτεί αυξημένη επίδοση ώστε να υποσκελιστεί το υψηλό κόστος εισαγωγής του. Η βελτιστοποίηση της επίδοσης σχετίζεται συχνά με την εφαρμογή τεχνικών για αποδοτική συλλογή και διανομή της πληροφορίας.

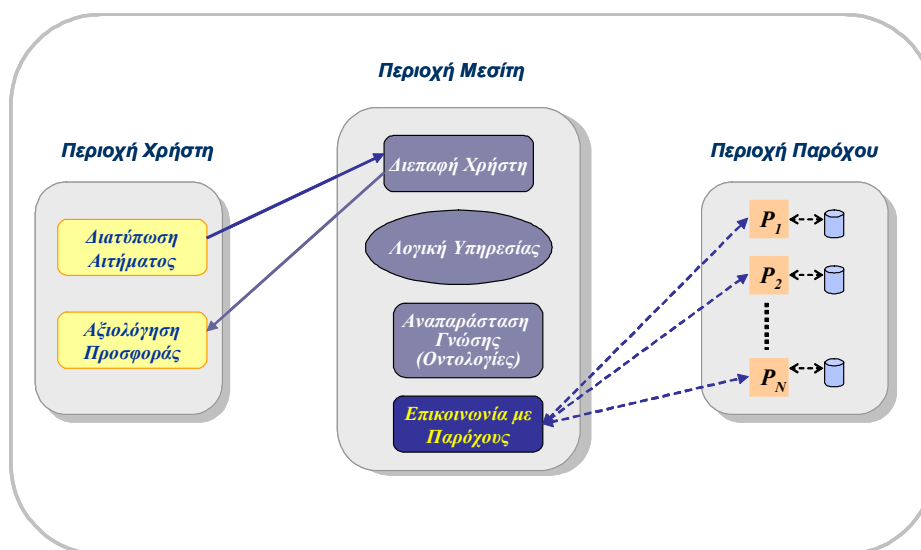
Επιλέγοντας την κατάλληλη αρχιτεκτονική, η οποία ικανοποιεί τις γενικές απαιτήσεις της εκάστοτε εφαρμογής, το πρόβλημα της βελτιστοποίησης επίδοσης αποσυντίθεται σε διαφορετικά υποπροβλήματα, των οποίων η λύση εξαρτάται τόσο από την ιδιαίτερη αρχιτεκτονική που επιλέγεται όσο και από τις ειδικές απαιτήσεις του εκάστοτε συστήματος. Στην παρούσα διατριβή, θεωρώντας σαν βάση αναφοράς μια αντιπροσωπευτική εφαρμογή πληροφοριακού συστήματος για καθεμιά από τις δυο αρχιτεκτονικές μεσιτικών συστημάτων, θα μελετηθεί ένα χαρακτηριστικό πρόβλημα βελτιστοποίησης για κάθε τύπο αρχιτεκτονικής. Πιο συγκεκριμένα, για την περίπτωση κεντρικής μεσιτείας θα θεωρήσουμε ένα διεπιχειρησιακό σύστημα ηλεκτρονικού εμπορίου, ενώ για την περίπτωση ομότιμων μεσιτών ένα σύστημα παροχής υπηρεσιών με επίγνωση του περιβάλλοντος. Η μελέτη εστιάζεται στην επικοινωνία ενός μεσίτη με απομακρυσμένες πηγές δεδομένων και με άλλους μεσίτες (ανάλογα με το μοντέλο), με σκοπό τη βελτιστοποίηση στην επίδοση συλλογής και διανομής της πληροφορίας. Στη συνέχεια περιγράφονται τα βασικά χαρακτηριστικά των αντιπροσωπευτικών αυτών συστημάτων και εντοπίζεται το σημείο μελέτης των αντίστοιχων προβλημάτων. Σημειώνεται ότι η λεπτομερής περιγραφή των συστημάτων δεν αποτελεί αντικείμενο της διατριβής και για το λόγο αυτό περισσότερες πληροφορίες παρέχονται μέσω των σχετικών αναφορών.

2.5.3.1 Πρώτο Πρόβλημα - Σύστημα Παροχής Υπηρεσιών Ηλεκτρονικού Εμπορίου

Στα πλαίσια του ερευνητικού προγράμματος Smart-EC [35] σχεδιάστηκε και υλοποιήθηκε μια διεπιχειρησιακή (*Business-to-Business - B2B*) πλατφόρμα διαμεσολάβησης για την παροχή σύνθετων υπηρεσιών ηλεκτρονικού εμπορίου. Το μεσιτικό μοντέλο του εν

λόγω συστήματος ακολουθεί τις σχεδιαστικές αρχές του **κεντρικού μεσίτη** ή **μεγαλομεσίτη (mega-broker)**. Ο μεγαλομεσίτης δρα σαν κεντρικός κόμβος ή πλατφόρμα διαμεσολάβησης για την εκτέλεση των περισσότερων συναλλαγών και για την επικοινωνία και το συντονισμό αγοραστή-προμηθευτή, έχοντας κοινά γνωρίσματα με τον κλασικό ηλεκτρονικό μεσίτη.

Η συνολική σχεδιαστική προσέγγιση του συστήματος ακολουθεί την αρχιτεκτονική τριών επιπέδων, όπου αναγνωρίζονται τα εξής τρία βασικά επίπεδα: η *περιοχή του χρήστη (User Domain)*, η *περιοχή του πάροχου (Provider Domain)* και η *περιοχή του μεσίτη (Broker Domain)*, η οποία αντιπροσωπεύει τη μεσιτική πλατφόρμα ή στρώμα διαμεσολάβησης (Σχήμα 2.10).



Σχήμα 2.10: Σύστημα Κεντρικής Μεσιτείας Smart-EC

Η *περιοχή του χρήστη* αντικατοπτρίζει τους πελάτες, οι οποίοι θέτουν τα αιτήματά τους για διαφορετικές υπηρεσίες μέσω κατάλληλης διεπαφής και επίσης διαχειρίζονται και αξιολογούν τις επιστρεφόμενες προσφορές.

Η *περιοχή του μεσίτη* αντιστοιχεί στην πλατφόρμα διαμεσολάβησης. Ο μεσίτης διαδραματίζει ουσιαστικό ρόλο σε εφαρμογές τριών επιπέδων. Χειρίζεται τα αιτήματα των χρηστών, απαλλάσσοντάς τους από την πολυπλοκότητα που υποκρύπτει η επίλυση των αιτημάτων τους μέσω προγραμματισμού της πρόσβασης στις εξωτερικές πηγές δεδομένων για τη σύνθεση της κατάλληλης πληροφορίας. Συγκεκριμένα, ο μεσίτης είναι υπεύθυνος για τη πρόσβαση στα δεδομένα που παρέχονται από τους διάφορους παρόχους και την εφαρμογή της σύνθετης κωδικοποιημένης γνώσης του συστήματος που παρέχεται από τις *Οντολογίες* [36]. Στη συνέχεια εφαρμόζει τους κανόνες της υπηρεσίας και επιστρέφει τα αποτελέσματα στους πελάτες.

Η *περιοχή του παρόχου* αντικατοπτρίζει τους εξωτερικούς παρόχους με τους οποίους μπορεί να επικοινωνήσει ο μεσίτης. Η μορφή στην οποία παρέχονται τα δεδομένα του κάθε παρόχου καθορίζεται κατά τη σύναψη συμφωνίας του παρόχου με το σύστημα.

Σημειώνεται ότι ο κύριος τρόπος διανομής πληροφορίας βασίζεται στο πρωτόκολλο Αιτήματος-Απόκρισης, ενώ χρησιμοποιείται και η υπό συνθήκη ώθηση, για αποστολή επιλεκτικής πληροφορίας (π.χ. δημοφιλής) από τον πάροχο στο μεσίτη όταν αυτή αλλάζει.

Η επικοινωνία του μεσίτη με τους εξωτερικούς παρόχους για ανάκτηση πληροφορίας είναι μια δαπανηρή φάση της επεξεργασίας πληροφορίας. Η ταχύτητα στη μεταφορά των δεδομένων από τη περιοχή των παρόχων στη περιοχή του μεσίτη παίζει σημαντικό ρόλο στην απόκριση του συστήματος. Για το λόγο αυτό κρίθηκε σκόπιμη η μελέτη αποδοτικών τεχνικών για την πραγματοποίηση της απευθείας και σε πραγματικό χρόνο επικοινωνίας του μεσίτη με τους παρόχους του συστήματος για ανάκτηση πληροφορίας. Οι διάφορες δυνατές στρατηγικές υλοποίησης της επικοινωνίας αυτής, οι οποίες διαφοροποιούνται ως προς το μοντέλο επικοινωνίας (*παραδοσιακό μοντέλο Πελάτη-Εξυπηρετητή* ή *μοντέλο Κινητών Πρακτόρων*) αλλά και τον τρόπο προσπέλασης των παρόχων (*σειριακός* ή *παράλληλος*), μελετώνται στο κεφάλαιο 3, ενώ το σχετικό πρόβλημα βελτιστοποίησης επιλύεται στο κεφάλαιο 4.

2.5.3.2 Δεύτερο Πρόβλημα - Σύστημα Παροχής Υπηρεσιών με Επίγνωση του Περιβάλλοντος

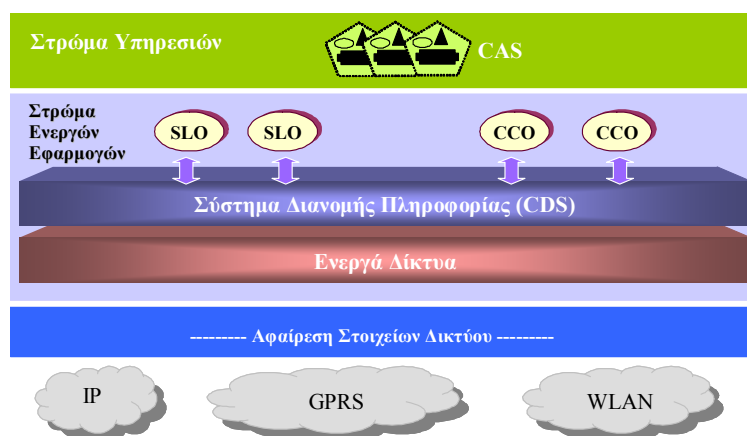
Στα πλαίσια του ερευνητικού προγράμματος Context [37] σχεδιάστηκε και υλοποιήθηκε πλατφόρμα μεσισμικού (*middleware platform*) με σκοπό την αυτόματη δημιουργία, διανομή και διαχείριση *υπηρεσιών με επίγνωση του περιβάλλοντος* (*Context Aware Services – CAS*). Με τον όρο *πληροφορία περιβάλλοντος* (*context*) εννοούμε οποιαδήποτε πληροφορία μπορεί να χρησιμοποιηθεί για να χαρακτηρίσει την κατάσταση μιας οντότητας. Μια οντότητα ορίζεται ως το πρόσωπο, η θέση, ή το αντικείμενο που θεωρείται σχετικό με την αλληλεπίδραση μεταξύ ενός χρήστη και μιας εφαρμογής, συμπεριλαμβανομένου του χρήστη και της εφαρμογής [38]. Η *επίγνωση περιβάλλοντος* (*context-awareness*) αναφέρεται στις ιδιότητες ενός συστήματος που το καθιστούν ενήμερο για την κατάσταση και το περιβάλλον του χρήστη του και το βοηθούν να προσαρμόζει τη συμπεριφορά του αναλόγως. Η χρησιμότητα της επίγνωσης περιβάλλοντος εντοπίζεται στο γεγονός ότι καθιστά ένα σύστημα ελάχιστα παρεισφρητικό, π.χ. σε δεδομένη υπηρεσία, μπορεί να ελαχιστοποιήσει τις ενέργειες ελέγχου που επιβαρύνουν το χρήστη.

Η επίγνωση περιβάλλοντος είναι αξιοποιήσιμη από μια νέα κατηγορία υπηρεσιών, οι οποίες αναπτύσσονται στα πλαίσια του σχετικά νέου τομέα έρευνας και ανάπτυξης που είναι γνωστός με τον όρο *διδεισδυτική υπολογιστική* (*pervasive computing*) ή *περιβαλλοντική νοημοσύνη* (*ambient intelligence*). Ένα σύστημα με επίγνωση περιβάλλοντος μπορεί να

θεωρηθεί ως «προσωπικός βοηθός», δεδομένου ότι το περιβάλλον του χρήστη συνυπολογίζεται ώστε οι αποφάσεις να λαμβάνονται από το βοηθό με τρόπο δυναμικό και προλαμβάνοντας τις ανάγκες και επιθυμίες των χρηστών. Στη λήψη αποφάσεων, ο «προσωπικός βοηθός» θα πρέπει να αποφεύγει να ενοχλεί το χρήστη, εκτός φυσικά από έκτακτη ανάγκη. Στόχος είναι η ανάπτυξη ενός συστήματος με επίγνωση περιβάλλοντος που θα μπορεί να εξομοιώσει έναν τέτοιο προσωπικό βοηθό.

2.5.3.2.1 Γενική Αρχιτεκτονική

Στη συνέχεια περιγράφεται η λογική διαστρωμάτωση του συστήματος σε σχέση με τις αλληλεπιδράσεις που εξελίσσονται κατά τη διάρκεια όλων των φάσεων του κύκλου ζωής μιας υπηρεσίας. Διακρίνονται τρία διαφορετικά στρώματα: το *Στρώμα Υπηρεσιών* (*Service Layer*), το *Στρώμα Ενεργών Εφαρμογών* (*Active Application Layer*), το οποίο αντιστοιχεί στην πλατφόρμα μεσισμικού, και το *Στρώμα Αφαίρεσης Στοιχείων Δικτύου* (*Network Element Abstraction Layer*) (Σχήμα 2.11).



Σχήμα 2.11: Γενική Αρχιτεκτονική Συστήματος Context

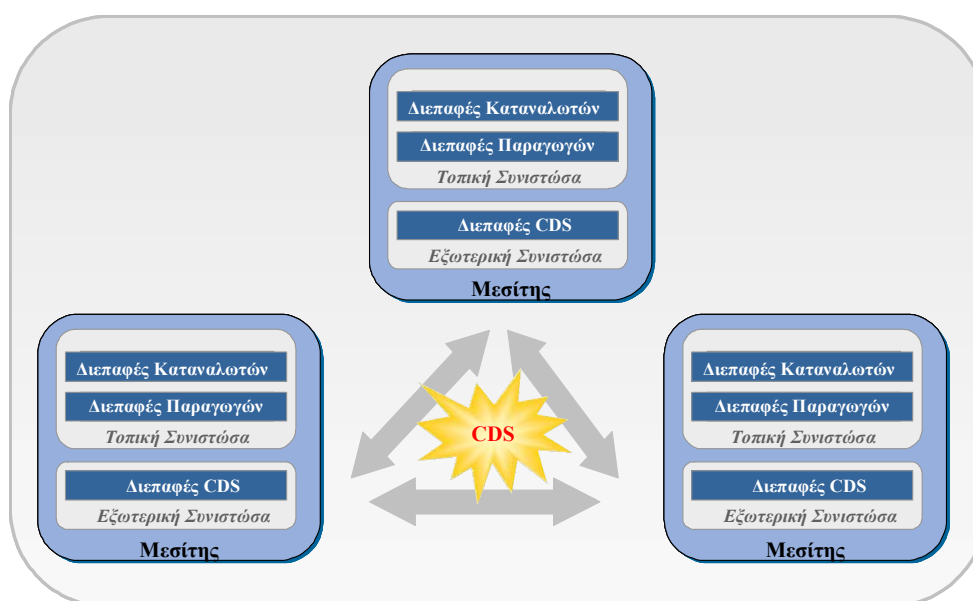
Το Στρώμα Υπηρεσιών παρέχει το πλαίσιο για τη δημιουργία υπηρεσιών με επίγνωση του περιβάλλοντος. Κατά τη φάση δημιουργίας υπηρεσίας, μια νέα υπηρεσία μοντελοποιείται με χρήση κανόνων, βάσει του μοντέλου PCIM (Policy Core Information Model). Ένα σύνολο προκαθορισμένων οντοτήτων πληροφορίας (Context Entities - CEs) και οντοτήτων δράσης (Action Entities - AEs) παρέχεται στους δημιουργούς υπηρεσιών, οι οποίοι μπορούν να κατασκευάσουν μια υπηρεσία χρησιμοποιώντας αυτές τις οντότητες ως δομικές μονάδες. Όταν ένας πελάτης εγγράφεται σε μια υπηρεσία, εισάγει τα χαρακτηριστικά που θα εξατομικεύσουν την υπηρεσία σύμφωνα με τις απαιτήσεις του και τελικά το αντικείμενο λογικής της υπηρεσίας (Service Logic Object – SLO) που αντιστοιχεί στο συγκεκριμένο πελάτη παράγεται και αναπτύσσεται στο περιβάλλον εκτέλεσης, το οποίο παρέχεται από το Στρώμα Ενεργών Εφαρμογών. Κατά τη φάση λειτουργίας της υπηρεσίας, πραγματοποιείται η συλλογή της πληροφορίας περιβάλλοντος που ορίζουν οι οντότητες πληροφορίας, και βάσει

της λογικής της υπηρεσίας που ορίζεται από τους σχετικούς κανόνες, οι συνθήκες αξιολογούνται και αναλόγως εκτελούνται οι επιθυμητές ενέργειες, όπως αυτές καθορίζονται στις οντότητες δράσης. Το Στρώμα Αφαίρεσης Στοιχείων Δικτύου παρέχει κοινό μηχανισμό μεταφοράς, περιβάλλοντας διαφορετικές τεχνολογίες δικτύου, όπως IPv4, IPv6, GPRS και WLAN μέσω κατάλληλων συνιστωσών προσαρμογής.

Κατά τη διάρκεια της φάσης λειτουργίας μιας CAS, η λογική της υπηρεσίας απαιτεί τη συλλογή πληροφορίας περιβάλλοντος από πολλαπλές, ετερογενείς και κατανεμημένες πηγές. Το Σύστημα Διανομής Πληροφορίας Περιβάλλοντος (*Context Distribution System - CDS*), το οποίο αποτελεί υπόστρωμα του Στρώματος Ενεργών Εφαρμογών, αναλαμβάνει τη διανομή της πληροφορίας, παρέχοντας όλους τους απαραίτητους μηχανισμούς [39]. Η κατανεμημένη λειτουργία των εν λόγω μηχανισμών παρέχεται από την τεχνολογία των Ενεργών Δικτύων, η εφαρμογή της οποίας μελετάται στο κεφάλαιο 3.

2.5.3.2.2 Σύστημα Διανομής Πληροφορίας – Τεχνικές Βελτιστοποίησης

Το αρχιτεκτονικό μοντέλο του CDS βασίζεται στις αρχές της διαμεσολάβησης, και συγκεκριμένα στο *μοντέλο ομότιμων μεσιτών*. Το ολοκληρωμένο CDS περιλαμβάνει παραγωγούς, καταναλωτές και μεσίτες. Οι μεσίτες επικοινωνούν και συνεργάζονται μεταξύ τους για να παρέχουν ένα ολοκληρωμένο σύστημα πληροφοριών (Σχήμα 2.12).



Σχήμα 2.12: Σύστημα Ομότιμων Μεσιτών Context

Οι *παραγωγοί* πληροφορίας περιβάλλοντος περιλαμβάνουν όλες τις πηγές που παράγουν στοιχειώδη πληροφορία περιβάλλοντος, όπως: πληροφορία δικτύου (π.χ. τοπικές μεταβλητές *MIB*), πληροφορία ασύρματης δικτυακής κάλυψης (π.χ. θέση χρήστη, χρόνος σύνδεσης μέσω *GPRS* και *WLAN*, ισχύς σήματος), πληροφορία χρηστών (π.χ. περιεχόμενο ατζέντας ή προφίλ χρήστη), πληροφορία συσκευών, πληροφορία που προέρχεται από ειδικές

εφαρμογές παροχής πληροφορίας περιβάλλοντος (π.χ. θερμοκρασία, καιρικές συνθήκες), πληροφορία χρόνου. Επιπλέον, οι παραγωγοί περιλαμβάνουν και εφαρμογές που παράγουν σύνθετη πληροφορία, η οποία προκύπτει μετά από συλλογή και κατάλληλη επεξεργασία στοιχειωδών πληροφοριών. Οι εν λόγω εφαρμογές αποκαλούνται *Αντικείμενα Υπολογισμού Πληροφορίας Περιβάλλοντος (Context Computational Objects – CCO)* (Σχήμα 2.11). Η λειτουργία των CCO έγκειται στη συλλογή πληροφορίας περιβάλλοντος από τις διαθέσιμες πηγές, τοπικές ή απομακρυσμένες, και στην εφαρμογή εξειδικευμένης λογικής με σκοπό την παραγωγή σύνθετης και ουσιαστικής πληροφορίας περιβάλλοντος.

Οι *καταναλωτές* αντιστοιχούν στα στιγμιότυπα των υπηρεσιών SLO, τα οποία είναι προσαρμοσμένα στις ανάγκες και επιθυμίες των χρηστών. Κατά τη διάρκεια της φάσης λειτουργίας, κάθε SLO εκδίδει αιτήματα για ανάκτηση πληροφορίας περιβάλλοντος προκειμένου η υπηρεσία να προσαρμόζεται στις αλλαγές που συμβαίνουν, απαιτώντας την εκτέλεση των κατάλληλων ενεργειών. Επίσης, τα CCO λειτουργούν και ως καταναλωτές καθώς εκδίδουν αιτήματα για συλλογή πληροφορίας από κατανεμημένες πηγές.

Οι *μεσίτες πληροφορίας (Context Brokers – CBs)*, αποτελούν τους διαμεσολαβητές μεταξύ των παραγωγών και των καταναλωτών, παρέχοντας ενιαία πρόσβαση στις ετερογενείς, κατανεμημένες και αυτόνομες πηγές δεδομένων. Οι *διεπαφές Παραγωγών* της τοπικής συνιστώσας ενός μεσίτη επιτρέπουν σε έναν παραγωγό να δηλώσει τον τύπο της πληροφορίας που παράγει στον συγκεκριμένο μεσίτη, τον οποίο και καθιστά υπεύθυνο για τη διαχείριση της πληροφορίας του. Με τον τρόπο αυτό, η εισαγωγή νέων παραγωγών ή η τροποποίηση των υπαρχόντων καθίσταται εύκολη, γιατί απαιτείται ενημέρωση μόνο του υπεύθυνου μεσίτη. Ένα SLO ή CCO διαβιβάζει το αίτημά του για ανάκτηση συγκεκριμένης πληροφορίας στον τοπικό μεσίτη μέσω των κατάλληλων *διεπαφών Καταναλωτών* της τοπικής συνιστώσας του μεσίτη. Αυτός με τη σειρά του είναι υπεύθυνος να επιστρέψει απάντηση, είτε αλληλεπιδρώντας με τοπικές είτε με απομακρυσμένες πηγές πληροφορίας. Συνεπώς, είναι απαραίτητη η συνεργασία μεταξύ των μεσιτών προκειμένου να παρασχεθεί απάντηση σε ένα αίτημα. Οι *διεπαφές της εξωτερικής συνιστώσας* ενός μεσίτη είναι υπεύθυνες για την επικοινωνία μεταξύ των μεσιτών ώστε να παρασχεθεί αποδοτική διανομή της πληροφορίας. Σημειώνεται ότι η κοινή αναφορά σε δεδομένη πληροφορία από καταναλωτές και παραγωγούς επιτυγχάνεται με τη χρήση των Οντοτήτων Πληροφορίας Περιβάλλοντος (CEs), όπως αναφέρθηκε νωρίτερα. Ένα CE αποτελεί στην ουσία XML περιγραφή ενός δεδομένου στοιχείου πληροφορίας και χαρακτηρίζεται από το όνομά του, τις παραμέτρους προσδιορισμού και τις παραμέτρους εξόδου.

Οι μεσίτες υποστηρίζουν διάφορους τρόπους διανομής πληροφορίας, ανάλογα με τις ανάγκες των υπηρεσιών. Τα αιτήματα των καταναλωτών μπορεί να είτε στη μορφή σύγχρονων ερωτήσεων (πρωτόκολλο Αιτήματος-Απόκρισης), είτε συνδρομών για λήψη

ασύγχρονων ειδοποιήσεων όταν συμβούν γεγονότα που συμφωνούν με το καταχωρημένο ενδιαφέρον των καταναλωτών (πρωτόκολλο Δημοσίευσης-Εγγραφής). Επιπλέον, επιλέγεται η υλοποίηση του πρωτοκόλλου Δημοσίευσης-Εγγραφής με βάση το περιεχόμενο (*content-based publish-subscribe system*), δεδομένου ότι η ενυπάρχουσα ευελιξία του εξυπηρετεί την εξατομικευμένη διαμόρφωση αιτημάτων με βάση τις ανάγκες των εκάστοτε υπηρεσιών.

Η εξασφάλιση αποδοτικότητας στο εν λόγω σύστημα διανομής πληροφορίας συνίσταται στη μείωση του χρόνου απόκρισης του στα επικείμενα αιτήματα των καταναλωτών, καθώς επίσης και στον περιορισμό των δαπανών σε επικοινωνιακούς ή αποθηκευτικούς πόρους του συστήματος. Σε ένα κατανεμημένο σύστημα μεσιτών, η αναζήτηση της πληροφορίας μεταξύ των μεσιτών αποτελεί συχνά το πιο χρονοβόρο στάδιο της επεξεργασίας πληροφορίας. Η απλή δρομολόγηση σύμφωνα με την οποία οποιοδήποτε αίτημα (είτε σύγχρονη ερώτηση του πρωτοκόλλου Αιτήματος-Απόκρισης, είτε συνδρομή για λήψη ασύγχρονης ειδοποίησης του πρωτοκόλλου Δημοσίευσης-Εγγραφής) εξαπλώνεται στο δίκτυο μεσιτών, δεδομένου ότι δεν είναι γνωστός εκ των προτέρων ο μεσίτης που «φιλοξενεί» την κατάλληλη πηγή, αποδεικνύεται μη-αποδοτική. Για το λόγο αυτό κρίθηκε σκόπιμη η μελέτη ενός προβλήματος αναζήτησης πληροφορίας σε σύστημα ομότιμων μεσιτών, εξετάζοντας διαφορετικούς αλγορίθμους δρομολόγησης, στα πλαίσια τόσο *προδραστικών* (*proactive*) όσο και *αντιδραστικών* (*reactive*) μηχανισμών. Διαφορετικές παραλλαγές δρομολόγησης θα μελετηθούν και θα συγκριθούν ως προς την επίδοσή τους στο κεφάλαιο 5.

2.6 ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό παρουσιάστηκαν οι γενικές σχεδιαστικές αρχές και τεχνολογίες διανομής δεδομένων των κατανεμημένων πληροφοριακών συστημάτων, η επιλογή των οποίων καθοδηγείται από τη φύση, τις απαιτήσεις και την κλίμακα της εκάστοτε εφαρμογής. Στη συνέχεια, προβλήθηκε ο ρόλος των μεσιτικών συστημάτων ως προς την παροχή επεκτασιμότητας και αποτελεσματικότητας για ενιαία πρόσβαση σε ετερογενείς, διασκορπισμένες και αυτόνομες πηγές δεδομένων. Εστιάζοντας στο υπόδειγμα μεσιτείας και μελετώντας τον ευρύτερο ρόλο του στα συστήματα διανομής πληροφορίας, τονίστηκε η ανάγκη για βελτιστοποίηση στην επίδοση συλλογής και διανομής της πληροφορίας μέσω μεσιτών. Τα συστήματα μεσιτείας ταξινομούνται με βάση τον αριθμό και την κατανομή των μεσιτών στα συστήματα κεντρικής μεσιτείας ή ομότιμων μεσιτών. Για κάθε τύπο μεσιτείας, παρουσιάστηκε μια χαρακτηριστική εφαρμογή πληροφοριακού συστήματος. Βάσει αυτών των εφαρμογών, εισήχθησαν τα προβλήματα μελέτης της διατριβής, τα οποία εστιάζουν στην επικοινωνία ενός μεσίτη με απομακρυσμένες πηγές δεδομένων ή με άλλους μεσίτες (ανάλογα

με το μοντέλο) με σκοπό τη βελτιστοποίηση στην επίδοση συλλογής και διανομής της πληροφορίας.

2.7 ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ

- [1] L. Liu, L. Yan, and T. Ozsu, “Interoperability in Large-scale Distributed Information Delivery Systems”, in *Advances in Workflow Systems and Interoperability*, A. Dogac et.al (eds.), Springer-Verlag, 1998.
- [2] R.M. Adler, “Distributed Coordination Models for Client/Server Computing”, *Computer*, Vol.28, No.4, pp. 14-22, April 1995.
- [3] M. Bever, K. Geihs, L. Heuser, M. Muhlhauser, A. Schil, “Distributed Systems, OSF DCE and Beyond”, *Proceedings of the International DCE Workshop on DCE - The OSF Distributed Computing Environment, Client/Server Model and Beyond*, LNCS, Vol.731, pp.1-20, Publisher: Springer-Verlag, ISBN: 3-540-57306-2, 1993.
- [4] D. Linthicum, “David Linthicum’s Guide to Client/Server and Intranet Development”, Publisher: John Wiley & Sons, pp.44-46, 1997.
- [5] H. Edelstein, “Unravelling Client/Server Architecture”, *DBMS*, Vol.34, p.7, May 1994.
- [6] A. Berson, “Client/Server Architecture”, Publisher: McGraw-Hill Inc., ISBN: 0-07-005076-7, pp. 39-41, 1994.
- [7] W.H. Inmon, “Developing Client/Server Application in an Architectural Environment”, Publisher: QED Technical Publishing Group, pp. 73-92, 1991.
- [8] W.W. Eckerson, “Three Tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications”, *Open Information Systems*, Vol.10, No.1, pp.1-12, January 1995.
- [9] A. Dickman, “Two-Tier versus Three-Tier Apps.”, *Informationweek* 553, pp.74-80, November 13, 1995.
- [10] J. Gallagher and S. Ramanathan, “Choosing a Client/Server Architecture. A Comparison of Two-Tier and Three-Tier Systems”, *Information Systems Management Magazine*, Vol.13, No.2, pp.7-13, Spring 1996.
- [11] R. Orfali, D. Harkey and J. Edwards, “Client/Server Survival Guide”, 3rd Edition, Publisher: Wiley, ISBN: 0-471-31615-6, January 1999.
- [12] L. Barnes and D. Shimberg, “Client/Server & Beyond: Strategies for the 21st Century”, Publisher: Prentice Hall, ISBN: 0135325161, April 1997.
- [13] K. Aberer and M. Hauswirth, “An Overview on Peer-to-Peer Information Systems”, *Proceedings of the Workshop on Distributed Data and Structures (WDAS-2002)*, Paris, France, 2002.
- [14] D.S. Milojevic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu “Peer-to-Peer Computing”, *Technical Report HPL-2002-57*, HP Laboratories, March 2002.

- [15] D. Aksoy, M. Altinel, R. Bose, U. Cetintemel, M. Franklin, J. Wang, S. Zdonik, "Research in Data Broadcast and Dissemination", Proceedings of the 1st International Conference on Advanced Multimedia Content Processing, Osaka, Japan, pp.196-211, November 1998.
- [16] P.Th. Eugster, P. Felber, R. Guerraoui, A.M. Kermarrec, "The Many Faces of Publish/Subscribe", ACM Computing Surveys, Vol.35, No.2, pp. 114-131, June 2003.
- [17] Y. Liu and B. Plale, "Survey of Publish-Subscribe Event Systems", Technical Report TR-574, May 2003.
- [18] G. Banavar, T. Chandra, B. Mukherjee, J. Nagarajarao, R.E. Strom, and D.C. Sturman: "An Efficient Multicast Protocol for Content-Based Publish-Subscribe Systems", Proceedings of 19th IEEE International Conference on Distributed Computing Systems (ICDCS), Austin, Texas, pp.262-272, May 1999.
- [19] L. Opyrchal, M. Astley, J. Auerbach, G. Banavar, R. Strom, and D. Sturman, "Exploiting IP Multicast in Content-Based Publish-Subscribe Systems", Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms, LNCS, Vol.1795, pp.185-207, Publisher: Springer-Verlag, ISBN: 3-540-67352-0, 2000.
- [20] A. Carzaniga, "Architectures for an Event Notification Service Scalable to Wide-area Networks", PhD thesis, Politecnico di Milano, Milano, Italy, December 1998.
- [21] G. Cugola, E. Di Nitto and A. Fuggetta, "The JEDI Event-based Infrastructure and its Application to the Development of the OPSS WFMS", IEEE Transactions on Software Engineering, Vol.27, No.9, pp.827-850, 2001.
- [22] M. Franklin and S. Zdonik, "Dissemination-Based Information Systems", IEEE Bulletin of the Technical Committee on Data Engineering, Vol.19, No.3, pp.20-30, September 1996.
- [23] S. Acharya, M. Franklin, and S. Zdonik, "Balancing Push and Pull for Data Broadcast", Proceedings of the ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, pp.183-194, May 1997.
- [24] L. Liu, C. Pu, R. Barga, and T. Zhou, "Differential Evaluation of Continual Queries", Proceedings of 16th IEEE International Conference on Distributed Computing Systems, Hong Kong, May 1996.
- [25] The Minstrel Push System Project website, Distributed Systems Group, Technical University of Vienna, 1999, Homepage URL: <http://www.infosys.tuwien.ac.at/Minstrel/>
- [26] A.M. Ouksel and A.P. Sheth, "Semantic Interoperability in Global Information Systems", ACM SIGMOD Record, Vol.28, No.1, pp.5-12, 1999.

- [27] M.T. Ozsu and P. Valduriez, "Principles of Distributed Database Systems", 2nd Edition, Publisher: Prentice Hall, ISBN: 0-13-659707-6, 1999.
- [28] S. Ram, "Heterogeneous Distributed Database Systems", IEEE Computer Magazine, Vol.24, No.12, pp.7-12, December 1991.
- [29] A. Sheth and J.A. Larson, "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", ACM Computing Surveys, Vol.22, No.3, pp.183-236, 1990.
- [30] F. Manola, S. Heidler, D. Georgakopoulos, M. Hornick and M. Brodie, "Distributed Object Management", International Journal of Intelligent and Cooperative Information Systems, Vol.1, No.1, pp.5-42, March 1992.
- [31] T. Ozsu, U. Dayal, and P. Valduriez, "Distributed Object Management", Publisher: Morgan Kaufmann, ISBN: 1-55860-256-9, May 1994.
- [32] R.G.G. Cattell, D.K. Barry, et al., "The Object Data Standard: ODMG 3.0", Publisher: Morgan Kaufmann, ISBN: 1558606475, 2000.
- [33] G. Wiederhold, "Mediators in the Architecture of Future Information Systems", IEEE Computer, Vol.25, pp.38-49, March 1992.
- [34] G. Wiederhold, "I³ Glossary", Draft 7, March 1995.
- [35] Smart-EC: Support for Mediation and Brokering for Electronic Commerce, IST-1999-10130-Smart-EC, Deliverable 4.1, "Smart-EC Architecture V1", Homepage URL: <http://www.telecom.ece.ntua.gr/smartec/>
- [36] M.F. Lopez, "Overview of Methodologies for Building Ontologies", Proceedings of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, pp.1-13, August 1999.
- [37] Context: Active Creation, Delivery and Management of Efficient Context-Aware Services, IST-2001-38142-CONTEXT, Deliverable D3.2, "Design and Implementation of Components for the Proof of Concept of Provisioning and Management of Context-aware Services", Homepage URL: <http://context.upc.es>
- [38] A.K. Dey and G.D. Abowd, "Towards a Better Understanding of Context and Context Awareness", Proceedings of Workshop on the What, Who, Where, When and How of Context-Awareness (CHI 2000), Hague, Netherlands, April 2000.
- [39] I. Sygkouna, M. Chantzara, S. Vrontis, S. Xynogalas, M. Anagnostou, E. Sykas, "Seamless Networking and QoS Provisioning for Context-Aware Services in Heterogeneous Environments", Proceedings of the 2nd International Workshop on Mobility Aware Technologies and Applications (MATA 2005), Montreal, Canada, pp.6-10, October 2005.

ΚΕΦΑΛΑΙΟ 3

***Μοντέλα, Τεχνολογίες και Εφαρμογές
Κινητού Κώδικα***

ΚΕΦΑΛΑΙΟ 3: ΜΟΝΤΕΛΑ, ΤΕΧΝΟΛΟΓΙΕΣ ΚΑΙ ΕΦΑΡΜΟΓΕΣ ΚΙΝΗΤΟΥ ΚΩΔΙΚΑ

3.1 ΕΙΣΑΓΩΓΗ

Καθώς το υπολογιστικό τοπίο μεταφέρεται από τα μεμονωμένα και αυτόνομα συστήματα υπολογιστών προς μια κατάσταση στην οποία η πραγματική ισχύς των υπολογιστών γίνεται αντιληπτή μέσω των κατανεμημένων, ανοικτών και δυναμικών συστημάτων, βρισκόμαστε αντιμέτωποι με νέες τεχνολογικές προκλήσεις και ευκαιρίες. Τα χαρακτηριστικά των δυναμικών και ανοικτών περιβαλλόντων στα οποία, για παράδειγμα, ετερογενή συστήματα πρέπει να αλληλεπιδράσουν ξεπερνώντας τα σύνορα των οργανισμών, και να λειτουργήσουν αποτελεσματικά μέσα στις γρήγορα μεταβαλλόμενες περιστάσεις και με τις εντυπωσιακά αυξανόμενες ποσότητες διαθέσιμων πληροφοριών, καθιστούν αναγκαία τη βελτίωση των παραδοσιακών μοντέλων και υποδειγμάτων υπολογισμού. Προς αυτή την κατεύθυνση, το *υπόδειγμα κινητού κώδικα* (*mobile code paradigm*) αποτελεί ένα σημαντικό υπόδειγμα προγραμματισμού, δημιουργώντας νέες δυνατότητες για τη σύνθεση κατανεμημένων συστημάτων σε ανοιχτά και δυναμικά μεταβαλλόμενα περιβάλλοντα, στα οποία η προσαρμοστικότητα και η ευελιξία αποτελούν κρίσιμες απαιτήσεις. Το υπόδειγμα αυτό αποτελεί αντικείμενο μελέτης του παρόντος κεφαλαίου.

Το υπόδειγμα κινητού κώδικα παρουσιάζεται ως εναλλακτική λύση του υποδείγματος Πελάτη-Εξυπηρετητή. Η κλασική προσέγγιση της μετακίνησης των δεδομένων στον κώδικα αντιστρέφεται για να υποστηρίξει τη μετακίνηση του κώδικα στα δεδομένα. Το βασικό κίνητρο για τη μετακίνηση του κώδικα είναι η μεταφορά του υπολογισμού σε έναν εξυπηρετητή δεδομένων ή έναν επικοινωνιακό εταίρο προκειμένου να μειωθεί το φορτίο του δικτύου μέσω της τοπικής αλληλεπίδρασης. Αναλόγως με τον τρόπο που ανακτάται ο κινητός κώδικας, δηλαδή με τη μέθοδο άντλησης (pull) ή ώθησης (push) και αναλόγως εάν μεταφέρεται και η κατάσταση εκτέλεσης του κώδικα μαζί με τον κώδικα, το υπόδειγμα κινητού κώδικα μπορεί να χαρακτηριστεί ως *κώδικας κατά απαίτηση*, *απομακρυσμένη αποτίμηση* ή *κινητός πράκτορας*.

Επιπλέον, η τεχνολογία Κινητών Πρακτόρων και Ενεργών Δικτύων, οι οποίες βασίζονται στη χρήση κινητού κώδικα, παρουσιάζονται ως δυνατές υλοποιήσεις του υπολογισμού *βασίζόμενου στο δίκτυο* (*network-based computation*). Σε αντίθεση με τις πολλαπλές ανεξάρτητες προσεγγίσεις που προσπαθούν να αντιμετωπίσουν τα προβλήματα που σχετίζονται με τα παθητικά δίκτυα, τα οποία παρέχουν απλώς μια υποδομή μεταφοράς και επιτρέπουν ελάχιστες δυνατότητες υπολογισμού μέσα σε αυτό, οι εν λόγω τεχνολογίες παρέχουν μηχανισμούς που επιτρέπουν την εκτέλεση υπολογισμών σε κόμβους μέσα στο

δίκτυο. Το βασικό κίνητρο είναι ότι η εκτέλεση υπολογισμών σχετικών με την εκάστοτε εφαρμογή σε κατάλληλα σημεία μέσα στο δίκτυο θα μπορούσε να βελτιώσει σημαντικά την επίδοση της εφαρμογής.

Τα κύρια σημεία αυτού του κεφαλαίου είναι τα εξής: Αρχικά μελετώνται τα κίνητρα που οδηγούν στη χρήση του κινητού κώδικα. Στη συνέχεια, παρουσιάζονται εναλλακτικά υποδείγματα κινητού κώδικα και συγκρίνονται με το παραδοσιακό μοντέλο απομακρυσμένης επικοινωνίας, ενώ στα πλαίσια υπολογιστικών μοντέλων μελετώνται οι τεχνολογίες των Κινητών Πρακτόρων και των Ενεργών Δικτύων. Τέλος, παρουσιάζεται η αξιοποίηση του κινητού κώδικα στα ειδικότερα προβλήματα μελέτης της διατριβής που εισήχθησαν στο κεφάλαιο 2. Πιο συγκεκριμένα, περιγράφεται η αξιοποίηση κινητών πρακτόρων στα πλαίσια ενός προβλήματος συλλογής κατανεμημένης πληροφορίας σε σύστημα κεντρικής μεσιτείας (πρώτο πρόβλημα), και η χρήση Ενεργών Δικτύων στα πλαίσια ενός προβλήματος αναζήτησης πληροφορίας σε σύστημα ομοτίμων μεσιτών (δεύτερο πρόβλημα).

3.2 ΚΙΝΗΤΡΑ ΧΡΗΣΗΣ ΚΙΝΗΤΟΥ ΚΩΔΙΚΑ

Τα δίκτυα υπολογιστών εξελίσσονται με γρήγορο ρυθμό, και αυτή η εξέλιξη πραγματοποιείται σε διάφορους άξονες. Η κλίμακα των δικτύων αυξάνει ταχύτατα και αυτό το φαινόμενο δεν περιορίζεται μόνο στο διαδίκτυο, του οποίου η τεράστια ανάπτυξη είναι γνωστή. Τα δίκτυα εντός και μεταξύ των οργανισμών βιώνουν επίσης μια αυξανόμενη ανάπτυξη, η οποία ενισχύεται από τη διαθεσιμότητα φτηνού υλικού και υποκινείται από την ανάγκη για ομοιόμορφα, ανοικτά, και αποτελεσματικά κανάλια πληροφορίας εντός και μεταξύ των οργανισμών. Μια παρενέργεια αυτής της ανάπτυξης είναι η σημαντική αύξηση της κίνησης στα δίκτυα, η οποία με τη σειρά της υποκινεί προσπάθειες για βελτίωση της επίδοσης της επικοινωνιακής υποδομής. Οι δικτυακές συνδέσεις βελτιώνονται συνεχώς και οι τεχνολογικές εξελίξεις οδηγούν σε αυξανόμενη υπολογιστική ισχύ, τόσο στους ενδιάμεσους κόμβους όσο και στους ακραίους.

Η αύξηση σε κλίμακα και επίδοση στα δίκτυα υπολογιστών είναι ταυτόχρονα η αιτία και το αποτέλεσμα ενός σημαντικού φαινομένου: τα δίκτυα γίνονται *διδεισδυτικά* (*pervasive*) και *πανταχού παρόντα* (*ubiquitous*). Με τον όρο «διδεισδυτικά» εννοούμε ότι η συνδετικότητα δεν είναι πλέον ένα ακριβό επιπρόσθετο χαρακτηριστικό, αλλά ένα βασικό γνώρισμα οποιουδήποτε υπολογιστικού στοιχείου. Με τον όρο «πανταχού παρόντα» αναφερόμαστε στη διαθεσιμότητα της συνδετικότητας των δικτύων ανεξάρτητα από τη φυσική θέση. Οι εξελίξεις στην ασύρματη τεχνολογία απαλλάσσουν τους κόμβους των δικτύων από τον περιορισμό της τοποθέτησής τους σε συγκεκριμένες θέσεις και επιτρέπουν την εμφάνιση της αποκαλούμενης *κινητής υπολογιστικής* (*mobile computing*). Στο νέο αυτό σενάριο, οι κινητοί χρήστες μπορούν να κινούνται μαζί με τους αντίστοιχους κόμβους σε

διαφορετικές φυσικές θέσεις και γεωγραφικές περιοχές, παραμένοντας συνδεδεμένοι στο δίκτυο μέσω ασύρματων συνδέσεων.

Ταυτόχρονα, η αυξημένη διαθεσιμότητα κατανεμημένων εφαρμογών και συστημάτων που είναι προσιτά στο ευρύ κοινό, έχει πυροδοτήσει το ενδιαφέρον των χρηστών και των αγορών, υποκινώντας την ανάπτυξη εντελώς νέων τύπων κατανεμημένων εφαρμογών μεγάλης κλίμακας. Ωστόσο, ο αριθμός των εφαρμογών που μπορούν να αξιοποιήσουν πλήρως την υποκείμενη επικοινωνιακή υποδομή είναι περιορισμένος. Ο λόγος είναι ότι η υπολογιστική υποδομή, δηλαδή το στρώμα λογισμικού το οποίο παρέχει τις αρχές και τους μηχανισμούς πάνω στους οποίους δομούνται οι κατανεμημένες εφαρμογές, δεν ακολουθεί το γρήγορο ρυθμό εξέλιξης της επικοινωνιακής υποδομής. Υπάρχουν αρκετά ζητήματα που πρέπει να διευθετηθούν. Η αυξανόμενη κλίμακα των δικτύων θίγει το πρόβλημα της επεκτασιμότητας. Τα περισσότερα αποτελέσματα τα οποία θεωρούνται σημαντικά για μικρά δίκτυα καταλήγουν να είναι μη εφαρμόσιμα σε μεγάλα δίκτυα όπως το διαδίκτυο. Η ασύρματη συνδετικότητα δημιουργεί ακόμη πιο έντονα προβλήματα [1, 2]. Οι κόμβοι μπορούν να κινούνται και να συνδέονται διακεκομμένα, και ως εκ τούτου η τοπολογία του δικτύου δεν καθορίζεται πλέον στατικά. Κατά συνέπεια, μερικές από τις βασικές αρχές των κατανεμημένων συστημάτων υπονομεύονται και προκύπτει η ανάγκη προσαρμογής των υπάρχοντων θεωρητικών και τεχνολογικών αποτελεσμάτων σε αυτό το νέο σενάριο. Ένα άλλο σχετικό ζήτημα είναι η διάχυση των υπηρεσιών και των εφαρμογών δικτύου σε πολύ μεγάλα τμήματα της κοινωνίας. Αυτό καθιστά απαραίτητη την αύξηση της δυνατότητας προσαρμογής των υπηρεσιών, έτσι ώστε διαφορετικές κατηγορίες χρηστών να μπορούν να προσαρμόζουν τη λειτουργία και τη διεπαφή μιας υπηρεσίας σύμφωνα με τις συγκεκριμένες ανάγκες και προτιμήσεις τους. Τέλος, η δυναμική φύση της επικοινωνιακής υποδομής και των απαιτήσεων της αγοράς απαιτεί αυξημένη ευελιξία και δυνατότητα επέκτασης.

Έχουν υπάρξει πολλές προσπάθειες δόμησης μιας υπολογιστικής υποδομής ισοδύναμης της επικοινωνιακής υποδομής. Οι περισσότερες από τις προτεινόμενες προσεγγίσεις προσπαθούν να προσαρμόσουν γνωστά μοντέλα και τεχνολογίες σε αυτό το νέο πλαίσιο, θεωρώντας συνήθως ως δεδομένη την παραδοσιακή αρχιτεκτονική Πελάτη-Εξυπηρετητή. Για παράδειγμα, η CORBA [3] ενσωματώνει το μηχανισμό RPC (Remote Procedure Call) στο αντικειμενοστραφές υπόδειγμα, προσπαθώντας να συνδυάσει τα οφέλη του δεύτερου από πλευράς δομοστοιχείωσης και επαναχρησιμοποίησης, με τον καθιερωμένο μηχανισμό επικοινωνίας του πρώτου. Ωστόσο, η επέκταση της υπάρχουσας τεχνολογίας Πελάτη-Εξυπηρετητή αποτελεί απλώς μια μερική λύση, γιατί αντιμετωπίζει τα προβλήματα της κλιμάκωσης και της συνδετικότητας, χωρίς όμως να παρέχει νέους μηχανισμούς και αφαιρέσεις που να επιτρέπουν σε έναν σχεδιαστή να δομήσει την εφαρμογή του με τρόπο ευέλικτο, προσαρμοστικό και εκτατό.

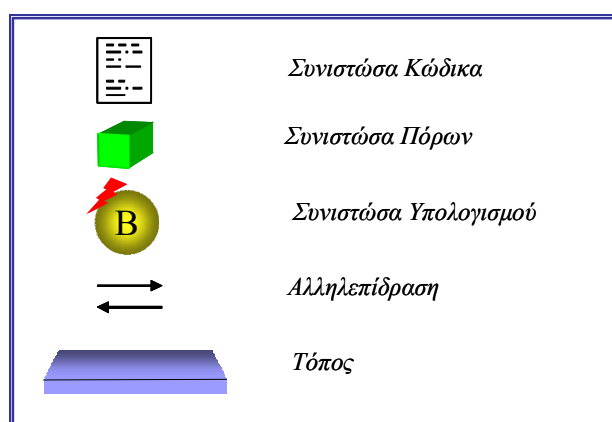
Μια διαφορετική προσέγγιση προέρχεται από την πολλά υποσχόμενη περιοχή έρευνας που εκμεταλλεύεται την έννοια του *κινητού κώδικα* (*mobile code*). Η *κινητικότητα κώδικα* μπορεί να οριστεί ως η ικανότητα δυναμικής αλλαγής των θέσεων όπου εκτελούνται τα τεμάχια κώδικα [4]. Η δυνατότητα μετακίνησης του κώδικα και εκτέλεσής του σε μια νέα θέση αποτελεί μια ισχυρή αρχή, η οποία προκάλεσε μια ενδιαφέρουσα σειρά εξελίξεων. Η συμβολή του κινητού κώδικα συνίσταται κυρίως στο γεγονός ότι παρέχει ένα ενιαίο, γενικό πλαίσιο, στο οποίο καταναμημένες, πληροφοριοστραφείς εφαρμογές μπορούν να υλοποιηθούν εύκολα και αποδοτικά, με το προγραμματιστικό φορτίο να κατανέμεται ομοιόμορφα στις πληροφορίες, το μεσισμικό και τους πελάτες. Στη συνέχεια εντοπίζονται οι κύριες διαφορές του κινητού κώδικα από άλλες σχετικές προσεγγίσεις, συγκρίνονται τα εναλλακτικά μοντέλα-τεχνολογίες κινητού κώδικα και παρουσιάζονται ενδιαφέρουσες εφαρμογές του.

3.3 ΜΟΝΤΕΛΑ ΕΠΙΚΟΙΝΩΝΙΑΣ

Η αλληλεπίδραση μεταξύ διαδικασιών που βρίσκονται σε διαφορετικές θέσεις σε ένα καταναμημένο υπολογιστικό σύστημα μπορεί να πραγματοποιηθεί με πολλούς διαφορετικούς τρόπους. Ένα υπόδειγμα επικοινωνίας αντιπροσωπεύει το σύνολο των κανόνων που ακολουθούνται για την ανταλλαγή δεδομένων και το συγχρονισμό της εκτέλεσης των διαδικασιών. Η καταναμημένη φύση των διαθέσιμων υπολογιστικών συστημάτων υποθέτει ότι οι υπολογιστικοί πόροι και τα δεδομένα μπορεί να μην βρίσκονται στο ίδιο μηχάνημα. Η μετακίνηση του κώδικα στα δεδομένα και των δεδομένων στον κώδικα αποτελούν εναλλακτικές στρατηγικές που μπορούν να υιοθετηθούν [5]. Η κλασική προσέγγιση Πελάτη-Εξυπηρετητή υποθέτει ότι η λειτουργικότητα του πελάτη διαχωρίζεται από την εκτελεστική ισχύ του εξυπηρετητή. Αυτό σημαίνει ότι διαδικασίες που εκτελούνται τοπικά στον πελάτη ανακτούν δεδομένα από απομακρυσμένες βάσεις, οι οποίες αποτελούν κοινούς πόρους του καταναμημένου συστήματος. Οι πρόσφατες τεχνολογικές βελτιώσεις επιτρέπουν εναλλακτικές λύσεις σε αυτό το υπόδειγμα επικοινωνίας [6]. Γενικά, η κλασική προσέγγιση μετακίνησης δεδομένων προς κώδικα συχνά αντιστρέφεται για να υποστηρίξει τη μετακίνηση κώδικα προς δεδομένα, στοχεύοντας στην προώθηση χρήσης απομακρυσμένων υπολογιστικών πόρων, οι οποίοι είναι ευρέως διαθέσιμοι και εύκολα προσβάσιμοι [4, 7]. Γλώσσες προγραμματισμού όπως η Java [8], υποστηρίζουν ισχυρά το υπόδειγμα επικοινωνίας που βασίζεται σε κινητό κώδικα, αφού επιτρέπουν να γραφτεί ο κώδικας μια φορά και να εκτελείται πολλές, σε ετερογενή περιβάλλοντα υλισμικού και λογισμικού.

Για τη μελέτη και σύγκριση των διαφόρων μοντέλων επικοινωνίας εισάγουμε τις εξής τρεις έννοιες, οι οποίες αναπαριστώνται στο Σχήμα 3.1: *συνιστώσες*, *αλληλεπιδράσεις*, και *τόπους*. Οι συνιστώσες είναι τα συστατικά μιας αρχιτεκτονικής λογισμικού. Μπορούν να

διαιρεθούν περαιτέρω στις *συνιστώσες κώδικα*, οι οποίες ενθυλακώνουν την τεχνογνωσία για την εκτέλεση συγκεκριμένων υπολογισμών, τις *συνιστώσες πόρων*, οι οποίες αντιπροσωπεύουν τα δεδομένα ή τις συσκευές που χρησιμοποιούνται κατά τη διάρκεια των υπολογισμών, και τις *συνιστώσες υπολογισμού*, οι οποίες αποτελούν ενεργούς εκτελεστές, ικανούς να πραγματοποιούν υπολογισμούς, όπως αυτοί προσδιορίζονται από μια συγκεκριμένη τεχνογνωσία. Οι αλληλεπιδράσεις αναφέρονται στα γεγονότα που ανταλλάσσονται μεταξύ δύο ή περισσότερων συνιστωσών, π.χ. ένα μήνυμα που ανταλλάσσεται μεταξύ δυο συνιστωσών υπολογισμού. Οι τόποι φιλοξενούν τις διάφορες συνιστώσες και υποστηρίζουν την εκτέλεση των συνιστωσών υπολογισμού. Ένας τόπος αντιπροσωπεύει διαισθητικά μια θέση. Οι αλληλεπιδράσεις μεταξύ συνιστωσών που εδρεύουν στον ίδιο τόπο θεωρούνται λιγότερο «ακριβές» από αυτές που πραγματοποιούνται μεταξύ συνιστωσών σε διαφορετικούς τόπους. Επιπλέον, ένας υπολογισμός πραγματοποιείται μόνο όταν η τεχνογνωσία που περιγράφει τον υπολογισμό, οι πόροι που χρησιμοποιούνται κατά τη διάρκεια του υπολογισμού και η συνιστώσα υπολογισμού που είναι αρμόδια για την εκτέλεση βρίσκονται στον ίδιο τόπο. Για τη μελέτη των εναλλακτικών στρατηγικών θα εξετάσουμε ένα σενάριο, όπου μια συνιστώσα υπολογισμού A η οποία βρίσκεται στον τόπο S_A , απαιτεί τα αποτελέσματα μιας υπηρεσίας. Υποθέτουμε επιπλέον την ύπαρξη ενός τόπου S_B , ο οποίος πρόκειται να περιληφθεί στην πραγματοποίηση της υπηρεσίας [9].

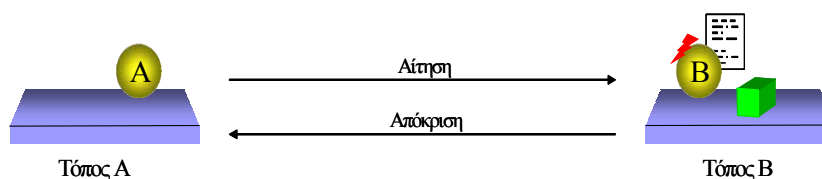


Σχήμα 3.1: Αναπαράσταση Συνιστωσών, Αλληλεπιδράσεων και Τόπων

3.3.1 ΚΛΗΣΗ ΑΠΟΜΑΚΡΥΣΜΕΝΗΣ ΔΙΑΔΙΚΑΣΙΑΣ (REMOTE PROCEDURE CALL – RPC)

Το κλασικό και ευρέως χρησιμοποιημένο σχέδιο επικοινωνίας, το οποίο συμφωνεί απόλυτα με το υπόδειγμα Πελάτη-Εξυπηρετητή, είναι το RPC [10]. Σύμφωνα με αυτό το μοντέλο, μια συνιστώσα λειτουργεί ως πελάτης όταν απαιτεί κάποια υπηρεσία από μια άλλη συνιστώσα, και ως εξυπηρετητής όταν αποκρίνεται σε αίτημα ενός πελάτη. Όπως φαίνεται στο Σχήμα 3.2, μια συνιστώσα υπολογισμού B (εξυπηρετητής) που προσφέρει ένα σύνολο υπηρεσιών τοποθετείται στον τόπο S_B . Η συνιστώσα πελάτη A, η οποία βρίσκεται στον τόπο

S_A , ζητά την εκτέλεση μιας υπηρεσίας μέσω αλληλεπίδρασης με τη συνιστώσα εξυπηρετητή B. Οι πόροι και η τεχνογνωσία (κώδικας) που απαιτούνται για την εκτέλεση της υπηρεσίας φιλοξενούνται στον τόπο S_B . Ως απάντηση, η B εκτελεί την αιτούμενη υπηρεσία, εκτελώντας την αντίστοιχη τεχνογνωσία και προσπελάζοντας τους σχετικούς πόρους του S_B . Η υπηρεσία παράγει κάποιο είδος αποτελέσματος, το οποίο θα παραδοθεί πίσω στον πελάτη με μια πρόσθετη αλληλεπίδραση. Παραδείγματα RPC είναι το πρωτόκολλο SOAP [11], το οποίο βασίζεται σε τεχνολογία XML, και το RMI [12], το οποίο ενσωματώνει το μοντέλο καταναμημένων αντικειμένων στη γλώσσα προγραμματισμού Java.



Σχήμα 3.2: Υπόδειγμα Πελάτη-Εξυπηρετητή

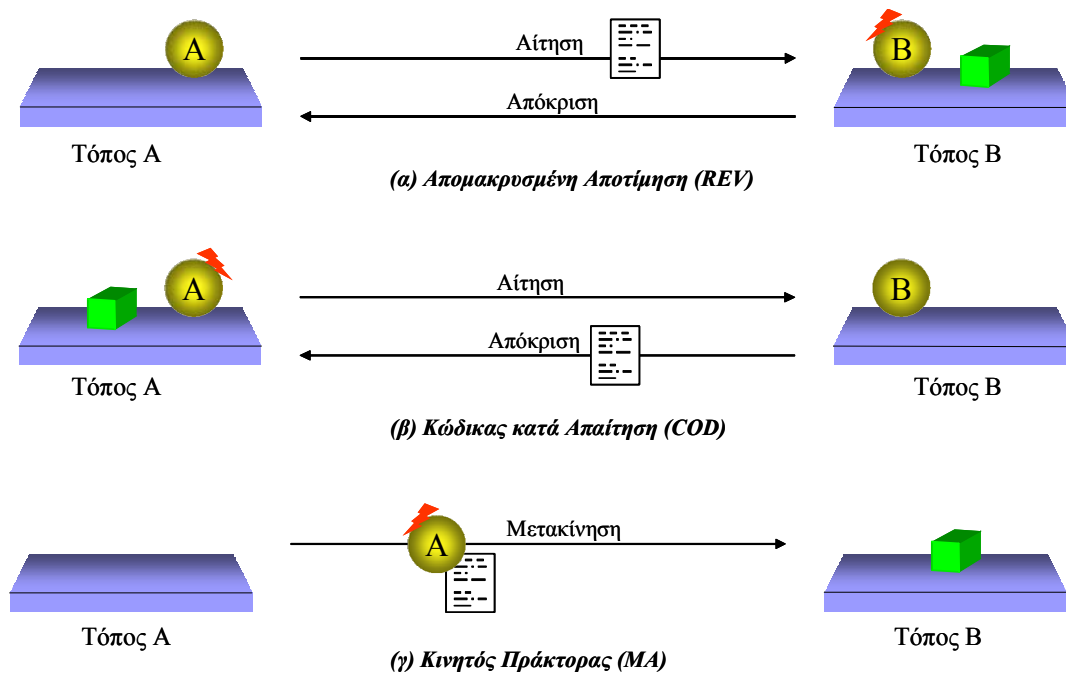
Στην πραγματικότητα, το έργο του εξυπηρετητή περιορίζεται στην εκτέλεση βασικών διαδικασιών για την αποθήκευση και την ανάκτηση δεδομένων. Εάν ο χρήστης έχει ειδικές απαιτήσεις σχετικά με τον τρόπο επεξεργασίας των δεδομένων και εάν ο εξυπηρετητής δεν υποστηρίζει ανάλογες λειτουργίες, μεγάλες ποσότητες αφιλτράριστων δεδομένων ανακτώνται στον πελάτη, ο οποίος και εκτελεί την ουσιαστική επεξεργασία των δεδομένων προκειμένου να εξάγει την απαραίτητη πληροφορία. Όλα αυτά όμως προκαλούν υπερφόρτωση στο σύστημα επικοινωνίας. Ωστόσο, σημαντικό πλεονέκτημα αυτού του μοντέλου είναι η δυνατότητα ελέγχου του τρόπου πρόσβασης στα δεδομένα του εξυπηρετητή, παρέχοντας υψηλό επίπεδο ασφάλειας.

3.3.2 ΥΠΟΔΕΙΓΜΑΤΑ ΚΙΝΗΤΟΥ ΚΩΔΙΚΑ (MOBILE CODE PARADIGMS)

Στο μοντέλο RPC, το σύνολο των υπηρεσιών που προσφέρονται από τον εξυπηρετητή είναι σταθερό, προσδιορίζεται εκ των προτέρων από το σχεδιαστή της εφαρμογής και είναι προσβάσιμο μέσω διεπαφών που καθορίζονται στατικά. Ωστόσο, οι παρεχόμενες υπηρεσίες ή οι συγκεκριμένες διεπαφές μπορεί να μην είναι κατάλληλες για τις διαφορετικές και απρόβλεπτες ανάγκες των χρηστών, ενώ η εξέλιξη της εφαρμογής απαιτεί συνήθως εργασίες υψηλού κόστους, οι οποίες μπορεί να περιορίσουν τη διαθεσιμότητα της εφαρμογής. Με άλλα λόγια δεν υπάρχει τρόπος να επεκταθούν δυναμικά οι δυνατότητες των εξυπηρετητών όσον αφορά τον κώδικα που περιγράφει την συμπεριφορά τους. Επιπλέον, ειδικές λειτουργίες, οι οποίες απαιτούν συνεχείς και εντατικές αλληλεπιδράσεις μεταξύ πελάτη και εξυπηρετητή μέσω του δικτύου, οδηγούν συνήθως σε μεγάλη σπατάλη εύρους ζώνης [6].

Τα υποδείγματα κινητού κώδικα δεν δεσμεύουν στατικά τον κώδικα που εκτελεί την υπηρεσία σε κάποιο συγκεκριμένο μηχάνημα, αλλά επιτρέπουν την μετακίνησή του σε διαφορετικά μηχανήματα. Σκοπός τους είναι να υπερνικήσουν τους περιορισμούς ευελιξίας των εξυπηρετητών, καθώς επίσης να αλλάζουν δυναμικά την ποιότητα μιας αλληλεπίδρασης μειώνοντας το κόστος της. Το βασικό κίνητρο για τη μετακίνηση είναι η μεταφορά του υπολογισμού σε έναν εξυπηρετητή δεδομένων ή έναν επικοινωνιακό εταίρο, προκειμένου να μειωθεί το φορτίο του δικτύου μέσω της τοπικής αλληλεπίδρασης.

Προσδιορίζονται τρία βασικά υποδείγματα που κάνουν χρήση της κινητικότητας κώδικα: η απομακρυσμένη αποτίμηση (*remote evaluation*), ο κώδικας κατά απαίτηση (*code on demand*), και ο κινητός πράκτορας (*mobile agent*) (Σχήμα 3.3). Αυτά χαρακτηρίζονται από τη θέση των συνιστωσών πριν και μετά την εκτέλεση της υπηρεσίας, από τη συνιστώσα υπολογισμού που είναι αρμόδια για την εκτέλεση του κώδικα, και από τη θέση όπου εκτελείται η υπηρεσία.



Σχήμα 3.3: Υποδείγματα Κινητού Κώδικα

3.3.2.1 Απομακρυσμένη Αποτίμηση (Remote Evaluation – REV)

Το υπόδειγμα REV [13] υπονοεί ότι ο εξυπηρετητής λαμβάνει όχι μόνο το αίτημα προς επεξεργασία από τον πελάτη, αλλά και ολόκληρο τον κώδικα που απαιτείται για την εκτέλεση των διαδικασιών που έχουν επιλεγεί στα αποθηκευμένα δεδομένα. Δεδομένου ότι ο πελάτης μπορεί να χρησιμοποιεί εξατομικευμένο κώδικα στον εξυπηρετητή, η απάντηση του εξυπηρετητή περιορίζεται στην αποστολή ακριβώς των πληροφοριών που απαιτούνται και τελικά χρησιμοποιούνται από τον πελάτη. Συνεπώς, τα δεδομένα που επιστρέφονται είναι

«έτοιμα προς χρήση» και χρειάζονται αμελητέα πρόσθετη επεξεργασία [14]. Σημειώνεται ότι η μετακίνηση κώδικα μεταξύ των εξυπηρετητών δεν υποστηρίζεται.

Σύμφωνα με το Σχήμα 3.3-α, η συνιστώσα A έχει την απαραίτητη τεχνογνωσία για την εκτέλεση της υπηρεσίας, αλλά στερείται τους απαιτούμενους πόρους, οι οποίοι βρίσκονται σε έναν απομακρυσμένο τόπο S_B . Συνεπώς, η A στέλνει την τεχνογνωσία στη συνιστώσα υπολογισμού B, η οποία βρίσκεται στον απομακρυσμένο τόπο. Η B, με τη σειρά της, εκτελεί τον κώδικα χρησιμοποιώντας τους τοπικά διαθέσιμους πόρους. Με μια πρόσθετη αλληλεπίδραση τα αποτελέσματα παραδίδονται πίσω στην A.

Το αρχικό κόστος επικοινωνίας είναι επομένως υψηλότερο σε σχέση με το μοντέλο RPC και οφείλεται στο γεγονός ότι ο κώδικας για την επεξεργασία των δεδομένων μπορεί να έχει αξιοσημείωτο μέγεθος, το οποίο είναι σαφώς μεγαλύτερο από ένα απλό αίτημα ανάκτησης πληροφορίας. Ωστόσο, αυτό το κόστος συνήθως αντισταθμίζεται από το στάδιο της απόκρισης, δεδομένου ότι ο όγκος των δεδομένων που επιστρέφονται είναι περιορισμένος.

3.3.2.2 Κώδικας κατά Απαίτηση (Code on Demand – COD)

Όπως αναφέρθηκε νωρίτερα, ο κώδικας που περιγράφει τη συμπεριφορά μιας συνιστώσας μπορεί να χρειαστεί να αλλάξει. Σύμφωνα με το υπόδειγμα COD, εάν ο πελάτης κατέχει τους πόρους που απαιτούνται για την εκτέλεση μιας υπηρεσίας, αλλά του λείπει μέρος του κώδικα εκτέλεσής της, μπορεί να καταφορτώσει τον κώδικα αυτό από μια απομακρυσμένη συνιστώσα, η οποία λειτουργεί ως εξυπηρετητής κώδικα [6].

Όπως φαίνεται στο Σχήμα 3.3-β, η συνιστώσα A έχει πρόσβαση στους πόρους που χρειάζεται, οι οποίοι βρίσκονται επίσης στον τόπο S_A . Εντούτοις, καμιά πληροφορία για το πώς να χειριστεί αυτούς τους πόρους δεν είναι διαθέσιμη στον S_A . Κατά συνέπεια, η A αλληλεπιδρά με τη B του S_B απαιτώντας την απαραίτητη τεχνογνωσία, η οποία βρίσκεται επίσης στον S_B . Μια δεύτερη αλληλεπίδραση πραγματοποιείται όταν η συνιστώσα B παραδίδει τον κώδικα στην A, η οποία στη συνέχεια τον εκτελεί. Χαρακτηριστική εφαρμογή του συγκεκριμένου υποδείγματος αποτελούν τα *Java Applets* [15].

3.3.2.3 Κινητός Πράκτορας (Mobile Agent - MA)

Ο κινητός πράκτορας αποτελεί αυτόνομο πρόγραμμα που μπορεί να κινείται μέσα στο δίκτυο και να ενεργεί εξ' ονόματος ενός χρήστη ή μιας άλλης οντότητας. Κατά τη μετακίνηση του πράκτορα μεταφέρεται τόσο ο κώδικας όσο και η κατάσταση εκτέλεσής του. Αυτό σημαίνει ότι συμβαίνει ένα είδος αναστολής της εκτέλεσης του προγράμματος, το οποίο θα συνεχίσει από το σημείο διακοπής του σε ένα απομακρυσμένο μηχάνημα [16, 17, 18].

Όπως φαίνεται στο Σχήμα 3.3-γ, η τεχνογνωσία της υπηρεσίας κατέχεται από τη συνιστώσα A, η οποία αρχικά φιλοξενείται στον τόπο S_A , αλλά μερικοί από τους

απαραίτητους πόρους βρίσκονται στον τόπο S_B . Ως εκ τούτου, η A μετακινείται στον τόπο S_B μεταφέροντας την τεχνογνωσία και ενδεχομένως κάποια ενδιάμεσα αποτελέσματα. Αφού εγκατασταθεί στον S_B , η A ολοκληρώνει την υπηρεσία χρησιμοποιώντας τους τοπικά διαθέσιμους πόρους. Το υπόδειγμα MA διαφέρει από τα άλλα υποδείγματα κινητού κώδικα, δεδομένου ότι οι σχετικές αλληλεπιδράσεις περιλαμβάνουν τη μετακίνηση μιας ολόκληρης συνιστώσας υπολογισμού. Ενώ τα υποδείγματα REV και COD εστιάζουν στη μεταφορά κώδικα μεταξύ των συνιστωσών, στο υπόδειγμα MA ολόκληρη η συνιστώσα υπολογισμού μετακινείται σε έναν απομακρυσμένο τόπο, μαζί με την κατάσταση εκτέλεσης, τον κώδικα, και κάποιους πόρους που ενδεχομένως απαιτούνται για την πραγματοποίηση του στόχου.

Το υπόδειγμα REV αποτελεί περιορισμένη προσέγγιση του υποδείματος MA . Στην πραγματικότητα, η μετακίνηση κώδικα ενυπάρχει επίσης στο REV , αλλά στην περίπτωση αυτή υπάρχει πάντα μια κατευθείαν αλληλεπίδραση μεταξύ του πελάτη και του εξυπηρετητή. Αυτό σημαίνει ότι ο κώδικας που στέλνεται από τον πελάτη επιστρέφει τα δεδομένα κατευθείαν στον πελάτη και ο κώδικας δεν μετακινείται μεταξύ των εξυπηρετητών.

Αντιθέτως, η αυτονομία των κινητών πρακτόρων μπορεί να χρησιμοποιηθεί προκειμένου να εκτελεστούν οι διαδικασίες πιο αποτελεσματικά. Ο πράκτορας περιέχει τις διαδικασίες για την αλληλεπίδραση με τις πηγές δεδομένων σύμφωνα με τους τρόπους που επιθυμεί ο χρήστης, αλλά μπορεί επίσης να λαμβάνει ανεξάρτητες αποφάσεις, όπως να μετακινείται σε άλλους κόμβους ή να επιστρέφει τα αποτελέσματα που έχει ήδη λάβει στον χρήστη, εάν τα θεωρεί ικανοποιητικά. Από αυτή την άποψη, η αλληλεπίδραση μεταξύ του χρήστη και του πράκτορα περιορίζεται στα στάδια της μετάδοσής του και της επιστροφής των αποτελεσμάτων. Αυτό που πραγματοποιείται στο ενδιάμεσο εξαρτάται μόνο από τον τρόπο που έχει σχεδιαστεί ο πράκτορας. Συνεπώς, μπορούμε να πούμε ότι ο χρήστης χάνει τον έλεγχο του πράκτορα μόλις αυτός αποσταλεί. Αυτό το χαρακτηριστικό γνώρισμα συνεπάγεται τα εξής: 1) το μέγεθος του κώδικα του πράκτορα είναι μεγαλύτερο από αυτό που απαιτείται για ένα απλό REV (η δομή του πράκτορα είναι πιο σύνθετη) 2) όταν ο πράκτορας μετακινείται, πρέπει να μεταφέρει τα ενδιάμεσα αποτελέσματα της επεξεργασίας που πραγματοποιήθηκε στους κόμβους που έχει ήδη επισκεφτεί. Συνεπώς, η ποσότητα των δεδομένων που μεταφέρεται σε κάθε μετακίνηση τείνει να αυξάνει. Ο πράκτορας μπορεί να περιορίζει δυναμικά τα δεδομένα που θεωρεί ενδιαφέροντα για τον χρήστη, ακόμη και απορρίπτοντας κάποια από αυτά που έχει ήδη συλλέξει σε προηγούμενους κόμβους. Ένας πράκτορας μπορεί επομένως να σχεδιάζεται έτσι ώστε να επιτρέπει ένα μέγιστο όγκο δεδομένων προς μεταφορά σε κάθε μετακίνησή του.

3.3.3 ΔΙΑΔΙΚΑΣΙΑ ΕΠΙΛΟΓΗΣ ΥΠΟΔΕΙΓΜΑΤΟΣ

Είναι γεγονός ότι δεν είναι απολύτως σαφές πότε πρέπει να χρησιμοποιούνται τα παραπάνω υποδείγματα και πώς κανείς μπορεί να επιλέξει το πιο κατάλληλο υπόδειγμα κατά

το σχεδιασμό μιας κατανεμημένης εφαρμογής. Είναι κοινά αποδεκτό ότι κανένα από τα υποδείγματα δεν θεωρείται με βεβαιότητα το καλύτερο. Σαν γενικό συμπέρασμα, θα λέγαμε ότι η κινητικότητα κώδικα θεωρείται συμφέρουσα όταν τα τοπικά δεδομένα που υποβάλλονται σε επεξεργασία είναι πολύ μεγαλύτερα σε όγκο από τον κώδικα επεξεργασίας, ώστε να υποσκελιστεί το κόστος αποστολής του κώδικα. Επιπλέον, η ικανότητα επεξεργασίας του αιτούντος κώδικα πρέπει να είναι συγκρίσιμη με αυτή του προμηθευτή κώδικα, ειδάλως, σε περίπτωση που το μηχάνημα του αιτούντος κώδικα είναι αργό, η εκτέλεση του σταλμένου κώδικα θα επιβραδύνει ολόκληρο τον υπολογισμό. Σε ένα κατανεμημένο σύστημα, παράμετροι όπως η ικανότητα επεξεργασίας των υπολογιστών, το εύρος ζώνης και η διέλευση του δικτύου, καθώς επίσης ο όγκος των δεδομένων και του κώδικα, θα πρέπει να λαμβάνονται υπόψη όταν ο υπεύθυνος προγραμματισμού καλείται να αποφασίσει πού και πώς θα εκτελέσει τον κινητό κώδικα.

Η επιλογή του πιο κατάλληλου υποδείγματος θα πρέπει να γίνεται ανά περίπτωση, σύμφωνα με τον τύπο της εκάστοτε εφαρμογής και την ειδικότερη λειτουργία που αυτή επιτελεί. Σε κάθε περίπτωση, συγκεκριμένες παράμετροι που περιγράφουν τη συμπεριφορά της εφαρμογής θα πρέπει να επιλέγονται, σε συνδυασμό με κάποια κριτήρια που αποτιμούν τις τιμές των παραμέτρων. Για παράδειγμα, κάποιος μπορεί να θέλει να ελαχιστοποιήσει τον αριθμό των αλληλεπιδράσεων ή το κόστος επεξεργασίας ή την κίνηση που προκαλείται στο δίκτυο. Επιπλέον, απαιτείται η υιοθέτηση ενός μοντέλου του υποκείμενου κατανεμημένου συστήματος που να δικαιολογεί τα κριτήρια. Για κάθε εξεταζόμενο υπόδειγμα, θα πρέπει να πραγματοποιείται ανάλυση ώστε τελικά να καθορίζεται ποιο υπόδειγμα βελτιστοποιεί τα κριτήρια που έχουν επιλεγεί. Προφανώς, αυτή η φάση δεν μπορεί να λάβει υπόψη όλα τα χαρακτηριστικά και τους περιορισμούς, οι οποίοι πιθανώς να γίνονται πλήρως αντιληπτοί μόνο μετά από λεπτομερή σχεδιασμό. Ωστόσο, παρέχει κάποιες ενδείξεις για το πιο «οικονομικό» υπόδειγμα [9].

3.3.4 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΈΡΕΥΝΑ

Η σύγκριση ως προς την επίδοση μεταξύ του παραδοσιακού υποδείγματος Πελάτη-Εξυπηρετητή και του υποδείγματος Κινητού Πράκτορα, το οποίο θεωρείται η πιο εξελιγμένη μορφή κινητού κώδικα, αποτέλεσε αντικείμενο έρευνας για αρκετές περιοχές εφαρμογών. Ωστόσο, τα περισσότερα ερευνητικά αποτελέσματα προέρχονται από την περιοχή *διαχείρισης δικτύων (network management)*. Στη συνέχεια παρουσιάζονται μερικά αντιπροσωπευτικά αποτελέσματα.

Παραδοσιακά, η διαχείριση δικτύων βασίζονταν στο SNMP (Simple Network Management Protocol) [19]. Το εν λόγω πρωτόκολλο έχει ως αρχή το παραδοσιακό μοντέλο Πελάτη-Εξυπηρετητή, όπου η κεντρική λογική διαχείρισης (διαχειριστής) βρίσκεται σε ένα κεντρικό σταθμό, ο οποίος συλλέγει και αναλύει τα δεδομένα που ανακτώνται από τους

φυσικά διανεμημένους εξυπηρετητές. Η διαδικασία αυτή τυπικά συνεπάγεται μεταφορά μεγάλου όγκου δεδομένων στον διαχειριστή, προκαλώντας δυσχέρεια διέλευσης και κατά συνέπεια καθυστέρηση στην επεξεργασία των δεδομένων [20]. Επιπλέον, η αύξηση της κλίμακας των δικτύων θέτει υπό αμφισβήτηση την επεκτασιμότητα του συγκεντρωτικού αυτού μοντέλου. Με βάση την αρχή απαίτησης της επεκτασιμότητας, το μοντέλο της κατανεμημένης διαχείρισης δικτύων εξετάστηκε ως εναλλακτικό του συγκεντρωτικού, στο οποίο ο κεντρικός διαχειριστής αντικαθίσταται από επί μέρους συνεργαζόμενα συστήματα διαχείρισης. Εντούτοις, αυτό έχει ως μειονέκτημα την απαίτηση σύνθετων μηχανισμών συντονισμού μεταξύ των επί μέρους διαχειριστών [21].

Η πιο πρόσφατη τάση είναι η χρησιμοποίηση κινητών πρακτόρων για τη διαχείριση μεγάλων ετερογενών δικτύων. Η τάση αυτή συνεπάγεται τον εφοδιασμό των πρακτόρων με ικανότητες διαχείρισης ώστε να είναι σε θέση να εκδίδουν τις απαιτούμενες αιτήσεις στους κόμβους που επισκέπτονται ενώ μετακινούνται αυτόνομα μεταξύ αυτών [21]. Η διαχείριση δικτύων που βασίζεται σε κινητούς πράκτορες επιτυγχάνει αποσυμφόρηση του κεντρικού διαχειριστή με την ανακατανομή του φορτίου επεξεργασίας και ελέγχου [22]. Ο απώτερος σκοπός είναι να ελαχιστοποιηθεί η κίνηση που αφορά στη διαχείριση και να επιταχυνθούν οι ενέργειες διαχείρισης με τη διανομή των διαδικασιών διαχείρισης στους πόρους του συστήματος.

Ο Rubinstein [22] συγκρίνει το συγκεντρωτικό μοντέλο SNMP με το μοντέλο Κινητών Πρακτόρων πραγματοποιώντας έναν αριθμό προσομοιώσεων με σκοπό την αξιολόγηση του χρόνου απόκρισης στην ανάκτηση τιμών παραμέτρων διαχείρισης. Στην ειδική περίπτωση που ένας μοναδικός πράκτορας χρησιμοποιείται για την εκτέλεση των ενεργειών διαχείρισης, τα αποτελέσματα δείχνουν ότι ο κινητός πράκτορας αποδίδει καλύτερα από το SNMP όταν ο αριθμός των διαχειριζόμενων στοιχείων κυμαίνεται μεταξύ δύο ορίων: ενός κατώτατου που καθορίζεται από τον αριθμό μηνυμάτων που διαπερνούν τυχόν συνδέσεις συμφόρησης που συνδέουν το διαχειριστή με τα διαχειριζόμενα στοιχεία, και ενός ανώτατου που καθορίζεται από το αυξημένο μέγεθος πράκτορα το οποίο δυσκολεύει τη μετακίνησή του. Πιο συγκεκριμένα, εξαιτίας του γεγονότος ότι το μέγεθος ενός πράκτορα δεν είναι μικρό συγκρινόμενο και με το μέγεθος ενός SNMP πακέτου, ένας κινητός πράκτορας δεν αποδίδει ικανοποιητικά όταν ο αριθμός των διαχειριζόμενων κόμβων που επισκέπτεται είναι πολύ μικρός. Η απόδοση του πράκτορα βελτιώνεται όταν υπάρχουν συνδέσεις συμφόρησης μεταξύ του διαχειριστή και των στοιχείων γιατί στη περίπτωση του SNMP, η απευθείας επικοινωνία του διαχειριστή με κάθε κόμβο χωριστά σημαίνει ότι αυξάνεται η πιθανότητα διέλευσης από συνδέσεις συμφόρησης. Ο κινητός πράκτορας όμως διασχίζει τέτοιου είδους συνδέσεις μόνο όταν πρόκειται να μεταφερθεί από τον διαχειριστή στην περιοχή των διαχειριζόμενων αντικειμένων και όταν επιστρέφει στην αρχική του θέση

αφού έχει προσπελάσει διαδοχικά τα διαχειριζόμενα στοιχεία. Ο καθορισμός ενός κατώτατου ορίου στον αριθμό στοιχείων που θα προσπελάσει πριν επιστρέψει στον κόμβο-διαχειριστή θα δώσει στον πράκτορα το προβάδισμα σε σχέση με το SNMP. Καθώς όμως ο αριθμός των κόμβων προσπέλασης αυξάνει, το μέγεθος του πράκτορα αυξάνει ανάλογα λόγω της μεταφοράς των ενδιάμεσων δεδομένων που ανακτώνται από κάθε κόμβο, με αποτέλεσμα η μετακίνησή του να γίνεται δυσκολότερη και άρα πιο χρονοβόρα. Συνεπώς, δικαιολογείται και η επιβολή ενός ανώτατου ορίου στον αριθμό των κόμβων που επισκέπτεται ο πράκτορας, με επιστροφή του στον κόμβο-διαχειριστή για εκφόρτωση των δεδομένων του.

Στην περίπτωση χρησιμοποίησης περισσότερων του ενός πράκτορα, ένα πιθανό πρόβλημα μελέτης σχετίζεται με τον καθορισμό του βέλτιστου αριθμού πρακτόρων. Ακολουθώντας το υπόδειγμα αυτό, ο Bieszczad [23] μελετά τη χρήση κινητών πρακτόρων με σκοπό την ανακάλυψη της τοπολογίας ενός δικτύου στα πλαίσια της διαχείρισής του, και επισημαίνει ότι η ταχύτητα με την οποία ανιχνεύονται οι αλλαγές που σχετίζονται με τη διάταξη του δικτύου εξαρτάται από την πυκνότητα των πρακτόρων: όσο περισσότεροι πράκτορες χρησιμοποιούνται, τόσο πιο γρήγορη θα είναι η διαδικασία ανίχνευσης. Υπάρχουν όμως ορισμένοι παράγοντες όπως π.χ. η διέλευση του δικτύου, που καθορίζουν ένα ανώτατο όριο στον αριθμό πρακτόρων που συμφέρει να χρησιμοποιηθούν.

Ο Gavalas [24] συγκρίνει την επίδοση από πλευράς χρόνου των κινητών πρακτόρων με αυτή του SNMP σε μια εφαρμογή ανάκτησης τιμών μεταβλητών από ένα σύνολο κόμβων, με σκοπό τον υπολογισμό μιας συνάρτησης φυσικής κατάστασης (*health function*). Τα αποτελέσματα δείχνουν ότι η τεχνική που βασίζεται σε έναν μοναδικό κινητό πράκτορα γίνεται λιγότερο αποδοτική καθώς ο αριθμός των κόμβων που προσπελάζονται αυξάνεται. Κατά συνέπεια, καθίσταται απαραίτητος ο διαμερισμός του δικτύου σε υποπεριοχές, με ανάθεση ενός πράκτορα σε κάθε υποπεριοχή, προκειμένου να επιτευχθούν χαμηλότεροι χρόνοι απόκρισης σε σχέση με το SNMP.

3.4 ΥΠΟΛΟΓΙΣΤΙΚΑ ΜΟΝΤΕΛΑ

Παραδοσιακά, η λειτουργία ενός δικτύου περιορίζονταν κυρίως στην απλή μεταφορά πακέτων από ένα άκρο του δικτύου σε κάποιο άλλο. Υπήρχε ένα διαχωριστικό σημείο μεταξύ των λειτουργιών που εκτελούνται μέσα στο δίκτυο και αυτών που γίνονται από τους χρήστες. Η επεξεργασία μέσα στο δίκτυο περιορίζονταν κυρίως στις λειτουργίες της δρομολόγησης, του ελέγχου συμφόρησης και της ποιότητας υπηρεσίας (Quality of Service - QoS). Αυτό το είδος δικτύου μπορεί να χαρακτηριστεί ως «παθητικό». Ένας αριθμός προβλημάτων που σχετίζονται με τα παθητικά δίκτυα είναι τα εξής: η δυσκολία ενσωμάτωσης νέων τεχνολογιών και προτύπων στην υποδομή του δικτύου, η μειωμένη απόδοση λόγω των περιττών λειτουργιών στα διάφορα στρώματα των πρωτοκόλλων και η δυσκολία προσαρμογής νέων

υπηρεσιών στο υπάρχον αρχιτεκτονικό μοντέλο. Μια πρόσθετη αδυναμία σχετίζεται με το γεγονός ότι, τελευταία, έχουν προκύψει εφαρμογές που απαιτούν μερικές φορές υπολογισμούς μέσα στο δίκτυο. Ελλείψει μιας αρχιτεκτονικής που να υποστηρίζει αυτές τις απαιτήσεις, οι εφαρμογές υιοθετούσαν ποικίλες ανεξάρτητες υπηρεσίες για την εκτέλεση υπολογισμών καθοδηγούμενων από τις ανάγκες των χρηστών στους κόμβους μέσα στο δίκτυο. Συνεπώς, έγινε αισθητή η ανάγκη αντικατάστασης των πολυάριθμων ειδικών προσεγγίσεων για τον **βασιζόμενο στο δίκτυο υπολογισμό** (*network-based computation*) από μια γενική δυνατότητα που θα επιτρέπει στους χρήστες να προγραμματίζουν τα δίκτυά τους. Αυτή η καινοτόμος ιδέα της παροχής στο χρήστη της δυνατότητας προγραμματισμού του δικτύου καλείται *ενεργή δικτύωση* (*active networking*) και οι σχετικές τεχνολογίες **Ενεργές Τεχνολογίες** (*Active Technologies*) [25].

Ένα σημαντικό βήμα προς τη κατεύθυνση ενός γενικευμένου μοντέλου υπολογισμού βασιζόμενου στο δίκτυο είναι η χρήση κινητού κώδικα. Σε ένα τέτοιο μοντέλο, οι έννοιες του πελάτη και του εξυπηρετητή αποδυναμώνονται και εμφανίζεται ένα σύστημα ομότιμων συνιστωσών, όπου όλοι οι υπολογιστικοί τύποι είναι ισοδύναμοι, διαδραματίζοντας δυναμικά το ρόλο αιτούντος κώδικα ή προμηθευτή κώδικα. Στα πλαίσια των Ενεργών Τεχνολογιών που κάνουν χρήση κινητού κώδικα και υποστηρίζουν το γενικευμένο μοντέλο υπολογισμού βασιζόμενου στο δίκτυο, θα παρουσιαστούν στη συνέχεια: 1) η τεχνολογία Κινητών Πρακτόρων, 2) η τεχνολογία Ενεργών Δικτύων.

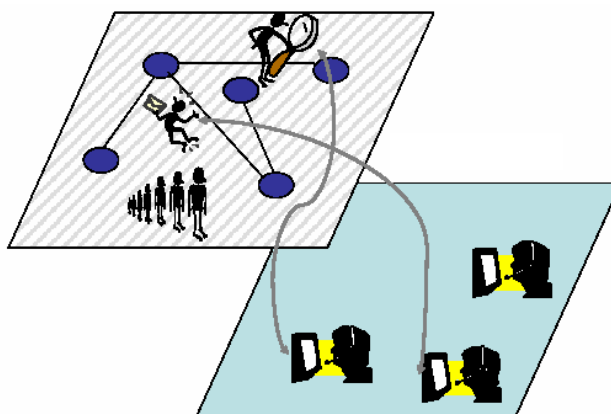
3.4.1 ΥΠΟΛΟΓΙΣΤΙΚΗ ΜΕ ΚΙΝΗΤΟΥΣ ΠΡΑΚΤΟΡΕΣ (MOBILE AGENT-BASED COMPUTING)

3.4.1.1 Γενικά

Η τεχνολογία Κινητών Πρακτόρων έχει λάβει μεγάλο ενδιαφέρον τα τελευταία χρόνια, παρέχοντας ένα νέο υπολογιστικό μοντέλο για τα ευρέως καταναμημένα και ετερογενή συστήματα. Οι βασικές έννοιες ενός συστήματος Κινητών Πρακτόρων είναι οι *θέσεις* (*locations*) και οι *πράκτορες* (*agents*). Ένα σύστημα Κινητών Πρακτόρων αποτελείται από διάφορες θέσεις όπου μπορεί να πραγματοποιηθεί κάποιος υπολογισμός και να παρασχεθούν υπηρεσίες. Οι κινητοί πράκτορες είναι ενεργές οντότητες που μπορούν να κινούνται από θέση σε θέση, για να συναντήσουν άλλους πράκτορες ή να έχουν πρόσβαση σε τοπικά παρεχόμενες υπηρεσίες. Η *κινητικότητα* (*mobility*) είναι το βασικό τους χαρακτηριστικό. Η *ευφνία* (*intelligence*) είναι το δεύτερο σημαντικό γνώρισμά τους, το οποίο αναφέρεται στην ικανότητα συνεργασίας τους με άλλους πράκτορες, χειρισμού ειδικών καταστάσεων ή προσαρμογής σε αλλαγές του δικτύου και απαιτεί προηγμένες τεχνικές τεχνητής νοημοσύνης.

Η κινητικότητα είναι μια ορθογώνια ιδιότητα των πρακτόρων, που σημαίνει ότι δεν είναι όλοι οι πράκτορες κινητοί. Ένας πράκτορας που δεν κινείται κατ' επιλογή ή δεν μπορεί να κινηθεί, παρά εκτελεί την αποστολή του μόνο στο σύστημα που ξεκίνησε η εκτέλεσή του χαρακτηρίζεται ως *στατικός (static)*. Έτσι, ένας πράκτορας μπορεί απλά να διαμένει σ' ένα μέρος και να επικοινωνεί με το περιβάλλον του μέσω συνηθισμένων μεθόδων, όπως για παράδειγμα με τη μέθοδο RPC ή με την αποστολή μηνυμάτων. Εάν δηλαδή χρειαστεί πληροφορία, η οποία δε βρίσκεται στο σύστημά του, ή χρειαστεί να αλληλεπιδράσει με πράκτορα άλλου συστήματος, χρησιμοποιεί τυπικούς μηχανισμούς επικοινωνίας Πελάτη-Εξυπηρετητή [26, 27].

Το Σχήμα 3.4 παριστάνει δύο διασυνδεδεμένους κόσμους, όπου χρήστες από τον πραγματικό κόσμο αναθέτουν σύνθετες εργασίες σε πράκτορες λογισμικού του δικτύου. Οι πράκτορες μετακινούνται από κόμβο σε κόμβο αυτόνομα, ερευνούν και ανακαλύπτουν τους διαθέσιμους πόρους και τις υπηρεσίες και επικοινωνούν με άλλους πράκτορες ή λειτουργούν ανεξάρτητα, ανάλογα με τις ανάγκες του εκάστοτε έργου [28]. Με αυτό τον τρόπο, η πολυπλοκότητα του έργου μειώνεται και η υπερφόρτωση πληροφοριών διαμοιράζεται μεταξύ πολλών οντοτήτων λογισμικού, οι οποίες εκτελούν πολλές εργασίες που διαφορετικά θα επιτελούσαν οι χρήστες.

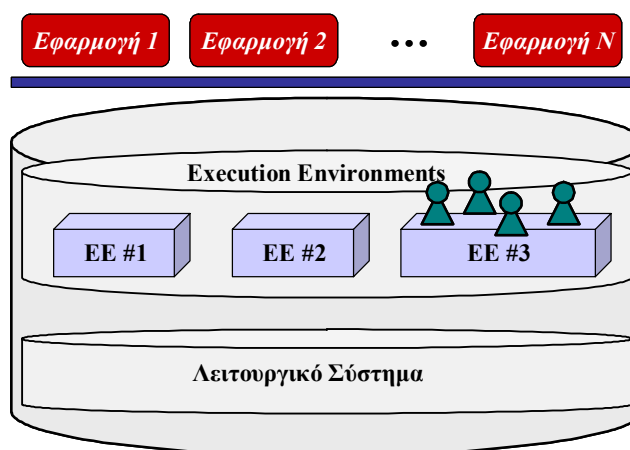


Σχήμα 3.4: Αποστολή Κινητών Πρακτόρων

Το υπόδειγμα Κινητών Πρακτόρων χειρίζεται το δίκτυο ως ένα περιβάλλον «φιλικό» προς τον πράκτορα, και τους πράκτορες ως προγραμματιστικές οντότητες που κινούνται από θέση σε θέση, εκτελώντας αποστολές για λογαριασμό των χρηστών ή άλλων οντοτήτων λογισμικού. Μια εφαρμογή κινητού πράκτορα απαιτεί δυο μέρη: ένα κινητό, δηλαδή τον κινητό πράκτορα, και ένα σταθερό, το οποίο παρέχει το *περιβάλλον εκτέλεσης του πράκτορα (Execution Environment - EE)*. Υπάρχει επίσης η έννοια του *σημείου εξυπηρέτησης (service point)*, το οποίο λειτουργεί ως μεσολαβητής μεταξύ του κινητού πράκτορα και των προσφερθέντων υπηρεσιών, και υλοποιείται ως στατικός πράκτορας. Το περιβάλλον

εκτέλεσης των πρακτόρων είναι γνωστό και ως *πλατφόρμα*. Οι περισσότερες πλατφόρμες Κινητών Πρακτόρων υλοποιούνται σαν εφαρμογές, οι οποίες «τρέχουν» πάνω από το λειτουργικό σύστημα (Σχήμα 3.5) και παρέχουν ειδικές δυνατότητες που σχετίζονται με τα εξής [29]:

- **Κινητικότητα:** παρέχει τους μηχανισμούς για τη μετακίνηση ενός πράκτορα μεταξύ των φυσικών κόμβων.
- **Επικοινωνία:** πραγματοποιεί την επικοινωνία ενός πράκτορα με τον εξωτερικό κόσμο.
- **Ονομασία και θέση:** οι κινητοί πράκτορες, όπως κάθε οντότητα, πρέπει να ονομάζονται. Επιπλέον, είναι σημαντικό να γνωρίζουμε σε ποιους κόμβους βρίσκονται οποιαδήποτε στιγμή.
- **Ασφάλεια:** οι πράκτορες πρέπει να προστατεύονται από τους κόμβους που τους φιλοξενούν, αλλά και οι κόμβοι από τους πράκτορες που τους επισκέπτονται.



Σχήμα 3.5: Κόμβος «φιλοξενίας» Κινητών Πρακτόρων

3.4.1.2 Πλεονεκτήματα Χρήσης Κινητών Πρακτόρων

Το ενδιαφέρον για τους κινητούς πράκτορες δεν υποκινείται μόνο από την τεχνολογία καθαυτή, αλλά και από τα πλεονεκτήματα που οι πράκτορες παρέχουν στη δόμηση καταναμημένων συστημάτων. Τα βασικότερα πλεονεκτήματα είναι τα εξής [26]:

- **Μειώνουν το φορτίο στο δίκτυο:** Τα καταναμημένα συστήματα συνήθως βασίζονται σε πρωτόκολλα επικοινωνίας που συνεπάγονται πολλαπλές αλληλεπιδράσεις για την πραγματοποίηση ενός δεδομένου έργου. Το αποτέλεσμα είναι να υπάρχει μεγάλη κίνηση στο δίκτυο. Οι κινητοί πράκτορες επιτρέπουν στους χρήστες να «πακετάρουν» μια συνδιάλεξη και στη συνέχεια να την αποστείλουν σε ένα προορισμό, όπου οι αλληλεπιδράσεις θα πραγματοποιηθούν τοπικά. Οι κινητοί πράκτορες είναι επίσης χρήσιμοι γιατί συνεισφέρουν στη μείωση της ροής των

δεδομένων στο δίκτυο. Όταν μεγάλες ποσότητες δεδομένων είναι καταχωρημένες σε απομακρυσμένα μηχανήματα, είναι προτιμότερο τα δεδομένα αυτά να επεξεργάζονται στο σημείο που βρίσκονται και όχι να μεταφέρονται μέσα στο δίκτυο. Η επεξεργασία δεδομένων βάσει πρακτόρων βασίζεται στη μετακίνηση του υπολογισμού στα δεδομένα και όχι των δεδομένων στον υπολογισμό.

- **Υποσκελίζουν την καθυστέρηση στο δίκτυο:** Κρίσιμα συστήματα πραγματικού χρόνου απαιτείται να ανταποκρίνονται σε πραγματικό χρόνο στις αλλαγές που συμβαίνουν στο περιβάλλον τους. Ο έλεγχος τέτοιων συστημάτων μέσω ενός δικτύου μεγάλης κλίμακας συνεπάγεται σημαντικές καθυστερήσεις. Στα συστήματα πραγματικού χρόνου όμως, τέτοιες καθυστερήσεις δεν είναι αποδεκτές. Οι κινητοί πράκτορες προσφέρουν λύση, γιατί μπορούν να αποσταλούν από ένα κεντρικό διαχειριστή ώστε να δράσουν τοπικά και να εκτελέσουν απευθείας τις οδηγίες του διαχειριστή.
- **Εμπεριέχουν πρωτόκολλα:** Όταν ανταλλάσσονται δεδομένα σε ένα κατανεμημένο σύστημα κάθε κόμβος φέρει τον κώδικα, ο οποίος υλοποιεί τα πρωτόκολλα που απαιτούνται για τη σωστή κωδικοποίηση των εξερχόμενων δεδομένων και τη μετάφραση των εισερχόμενων δεδομένων. Ωστόσο, καθώς τα πρωτόκολλα εξελίσσονται συνεχώς ώστε να εξυπηρετούν τις νέες απαιτήσεις για αποδοτικότητα και ασφάλεια, καθίσταται δύσκολη έως αδύνατη η σωστή αναβάθμιση του κώδικα του πρωτοκόλλου. Οι κινητοί πράκτορες όμως, μπορούν να μετακινούνται σε απομακρυσμένους κόμβους και να εγκαθιστούν «κανάλια» που βασίζονται στα κατάλληλα πρωτόκολλα.
- **Εκτελούνται ασύγχρονα και αυτόνομα:** Οι κινητές συσκευές βασίζονται συχνά σε ακριβές ή «εύθραυστες» δικτυακές συνδέσεις. Οι λειτουργίες που απαιτούν συνεχώς ανοιχτές συνδέσεις μεταξύ μιας κινητής συσκευής και ενός σταθερού δικτύου είναι συχνά ανέφικτες από οικονομικής ή τεχνικής σκοπιάς. Προκειμένου να λυθεί αυτό το πρόβλημα, οι λειτουργίες μπορούν να ενσωματώνονται μέσα σε κινητούς πράκτορες, οι οποίοι στη συνέχεια μπορούν να αποσταλούν στο δίκτυο. Μετά την αποστολή τους, οι πράκτορες γίνονται ανεξάρτητοι από τη διαδικασία δημιουργίας τους και μπορούν να λειτουργήσουν ασύγχρονα και αυτόνομα. Η κινητή συσκευή μπορεί να επανασυνδεθεί αργότερα για να συλλέξει τον πράκτορα.
- **Προσαρμόζονται δυναμικά:** Οι κινητοί πράκτορες είναι ικανοί να «διαισθάνονται» το περιβάλλον εκτέλεσής τους και να αντιδρούν αυτόνομα σε τυχόν αλλαγές. Πολλαπλοί κινητοί πράκτορες έχουν τη μοναδική ικανότητα να κατανέμονται ανάμεσα στους κόμβους του δικτύου έτσι ώστε να διατηρούν τη βέλτιστη διάταξη για τη λύση ενός συγκεκριμένου προβλήματος.

- **Είναι εκ φύσεως ετερογενείς:** Ο δικτυακός υπολογισμός είναι βασικά ετερογενής, συχνά τόσο από πλευράς υλισμικού όσο και λογισμικού. Επειδή οι κινητοί πράκτορες είναι γενικά ανεξάρτητοι του υπολογιστή και του στρώματος μεταφοράς (εξαρτώνται μόνο από το περιβάλλον εκτέλεσής τους), παρέχουν τις βέλτιστες συνθήκες για ομοιόμορφη ενοποίηση των συστημάτων.
- **Είναι αυτοδύναμοι και ανεκτικοί σε λάθη:** Η ικανότητα των κινητών πρακτόρων να αντιδρούν δυναμικά σε δυσμενείς καταστάσεις και γεγονότα κάνει πιο εύκολη τη δημιουργία αυτοδύναμων και ανεκτικών σε λάθη κατανεμημένων συστημάτων. Εάν ένας κόμβος τεθεί εκτός λειτουργίας, τότε όλοι οι πράκτορες που εκτελούνται σε αυτό τον κόμβο προειδοποιούνται, και τους δίνεται ο απαραίτητος χρόνος να αποσταλούν και να συνεχίσουν τη λειτουργία τους σε έναν άλλο κόμβο του δικτύου.

3.4.1.3 Εφαρμογές Κινητών Πρακτόρων

Οι πιο γνωστές εφαρμογές που μπορούν να επωφεληθούν από τη χρήση κινητών πρακτόρων είναι οι εξής [26]:

- **Ηλεκτρονικό Εμπόριο (E-Commerce)**

Οι κινητοί πράκτορες ταιριάζουν απόλυτα σε εφαρμογές ηλεκτρονικού εμπορίου. Μια εμπορική συναλλαγή μπορεί να απαιτεί πρόσβαση, σε πραγματικό χρόνο, σε απομακρυσμένους πόρους και ίσως διαπραγμάτευση μεταξύ πρακτόρων. Διαφορετικοί πράκτορες έχουν διαφορετικούς στόχους και υλοποιούν διαφορετικές στρατηγικές προκειμένου να τους πραγματοποιήσουν. Οι πράκτορες είναι ικανοί να εκφράζουν τις προθέσεις των δημιουργών τους ενεργώντας εκ μέρους των. Συνεπώς, η τεχνολογία Κινητών Πρακτόρων αποτελεί μια ελκυστική λύση για τέτοιου είδους προβλήματα.

- **Προσωπική Βοήθεια (Personal Assistance)**

Η ικανότητα των κινητών πρακτόρων να εκτελούνται σε απομακρυσμένους κόμβους τους καθιστά κατάλληλους ως βοηθούς στην εκτέλεση έργων στο δίκτυο για λογαριασμό των δημιουργών τους. Οι απομακρυσμένοι «βοηθοί» λειτουργούν ανεξάρτητα από την περιορισμένη δικτυακή τους σύνδεση, δηλαδή οι δημιουργοί τους μπορούν ακόμη και να θέσουν εκτός λειτουργίας τους υπολογιστές τους. Για παράδειγμα, για να προγραμματιστεί μια συνάντηση με πολλά άτομα, ένας χρήστης μπορεί να στείλει έναν κινητό πράκτορα να αλληλεπιδράσει με τους πράκτορες που εκπροσωπούν καθένα από τους ανθρώπους που είναι καλεσμένοι στη συνάντηση. Οι πράκτορες διαπραγματεύονται και τελικά καθορίζουν την ώρα της συνάντησης.

- **Ασφαλής Μεσιτεία (Secure Brokering)**

Μία ενδιαφέρουσα εφαρμογή των κινητών πρακτόρων είναι σε συνεργασίες, στις οποίες δεν είναι όλοι οι συνεργάτες αξιόπιστοι. Οι συμμετέχοντες μπορούν να αφήσουν τους κινητούς

πράκτορές τους σε ένα κοινά συμφωνημένο και ασφαλή κόμβο, όπου η συνεργασία πραγματοποιείται χωρίς ρίσκο του κόμβου που μπορεί να παίρνει το μέρος κάποιου από τους επισκέπτες πράκτορες.

□ ***Ανάκτηση Κατανεμημένης Πληροφορίας (Distributed Information Retrieval)***

Αντί να μετακινούνται μεγάλες ποσότητες πληροφορίας στη μηχανή αναζήτησης έτσι ώστε να δημιουργούνται ευρετήρια αναζήτησης, οι δημιουργοί των πρακτόρων μπορούν να αποστέλλουν τους πράκτορές τους σε απομακρυσμένες πηγές πληροφορίας, όπου δημιουργούν τοπικά ευρετήρια αναζήτησης. Οι κινητοί πράκτορες μπορούν επίσης να εκτελούν εκτεταμένες αναζητήσεις, οι οποίες δεν περιορίζονται από τις ώρες κατά τις οποίες ο υπολογιστής του δημιουργού είναι σε λειτουργία.

□ ***Υπηρεσίες Τηλεπικοινωνιακών Δικτύων (Telecommunication Network Services)***

Η υποστήριξη και η διαχείριση προηγμένων τηλεπικοινωνιακών υπηρεσιών χαρακτηρίζονται από δυναμική αναδιαμόρφωση του δικτύου και εξατομίκευση ανάλογα με τις ανάγκες του εκάστοτε χρήστη. Το μέγεθος αυτών των δικτύων και οι αυστηρές απαιτήσεις κάτω από τις οποίες λειτουργούν απαιτούν την τεχνολογία Κινητών Πρακτόρων ώστε να διατηρηθούν τα συστήματα ευέλικτα και αποδοτικά.

□ ***Εφαρμογές Ροής Εργασίας (Workflow applications and groupware)***

Η φύση των εφαρμογών ροής εργασίας περιλαμβάνει υποστήριξη για τη ροή της πληροφορίας μεταξύ των συνεργατών. Οι κινητοί πράκτορες είναι ιδιαίτερα χρήσιμοι στην περίπτωση αυτή, διότι επιπλέον της κινητικότητας, παρέχουν ένα βαθμό αυτονομίας στα στοιχεία ροής εργασίας. Μεμονωμένα στοιχεία ροής εργασίας ενσωματώνουν πλήρως την πληροφορία και τη συμπεριφορά που χρειάζονται για να μετακινούνται μέσα στον οργανισμό, ανεξάρτητα από οποιαδήποτε συγκεκριμένη εφαρμογή.

□ ***Παρακολούθηση και Ειδοποίηση (Monitoring and Notification)***

Αυτή η κλασική εφαρμογή κινητών πρακτόρων αξιοποιεί την ασύγχρονη φύση τους. Ένας πράκτορας μπορεί να παρακολουθεί μια δεδομένη πηγή πληροφορίας χωρίς να εξαρτάται από το σύστημα από το οποίο προήρθε. Οι πράκτορες λοιπόν μπορούν να αποσταλούν σε μια πηγή και να περιμένουν μέχρι να γίνουν διαθέσιμα συγκεκριμένα είδη πληροφορίας.

□ ***Διάδοση Πληροφορίας (Information Dissemination)***

Οι κινητοί πράκτορες ενσωματώνουν το μοντέλο ώθησης πληροφορίας (push model). Μπορούν να διαδίδουν πληροφορίες, όπως νέα και αυτόματες ενημερώσεις λογισμικού για πωλητές. Οι πράκτορες φέρουν τις νέες συνιστώσες λογισμικού, όπως επίσης και τις διαδικασίες εγκατάστασής τους κατευθείαν στους υπολογιστές των πελατών, όπου αυτόνομα ενημερώνουν και διαχειρίζονται το λογισμικό.

□ **Παράλληλη Επεξεργασία (Parallel Processing)**

Δεδομένου ότι οι κινητοί πράκτορες μπορούν να δημιουργήσουν μια σειρά από κλώνους στο δίκτυο, μια άλλη πιθανή χρήση της τεχνολογίας Κινητών Πρακτόρων είναι η διαχείριση έργων παράλληλης επεξεργασίας. Εάν ένας υπολογισμός απαιτεί τόσο μεγάλη ισχύ επεξεργασίας ώστε να πρέπει να κατανεμηθεί μεταξύ διαφόρων επεξεργαστών, μια υποδομή από κόμβους που φιλοξενούν κινητούς πράκτορες μπορεί να χρησιμοποιηθεί για την ανάθεση των σχετικών διαδικασιών.

3.4.1.4 Πλατφόρμες Κινητών Πρακτόρων

Σήμερα, οι περισσότερες πλατφόρμες Κινητών Πρακτόρων υλοποιούνται σαν εφαρμογές της Java. Οι πιο δημοφιλείς από αυτές είναι η *Aglet*, η *Voyager* και η *Grasshopper*.

- **Aglet** [30]: Η Aglet της IBM μοντελοποιεί τον κινητό πράκτορα σαν ένα *Java Applet*. Στην ουσία, η Aglet παρέχει ένα απλό πλαίσιο εργασίας, όπου ο προγραμματιστής επεκτείνει προκαθορισμένες μεθόδους για να προσθέσει την επιθυμητή λειτουργία. Ένα *aglet* ορίζεται ως ένα κινητό αντικείμενο της Java, το οποίο επισκέπτεται κόμβους συμβατούς με την Aglet σε ένα δίκτυο υπολογιστών. Ένα *aglet* «τρέχει» στο δικό του νήμα εκτέλεσης μόλις φτάσει στον κόμβο που θα το φιλοξενήσει, με αποτέλεσμα να χαρακτηρίζεται ως αυτόνομο. Είναι επίσης αναδραστικό γιατί αποκρίνεται στα εισερχόμενα μηνύματα. Το πλήρες μοντέλο αντικειμένου *aglet* περιλαμβάνει επιπρόσθετες αφαιρέσεις, όπως το περιβάλλον του, το πληρεξούσιό του (*proxy*), το μήνυμά του, το δρομολόγιό του, και το αναγνωριστικό του. Αυτές οι πρόσθετες αφαιρέσεις παρέχουν σε ένα *aglet* το πλήρες περιβάλλον στο οποίο μπορεί να εκτελέσει τις εργασίες του. Ένα *aglet* χρησιμοποιεί ένα απλό πληρεξούσιο αντικείμενο για τη μετάδοση των μηνυμάτων του και επίσης διατηρεί μια κλάση μηνύματος για να ενθυλακώνει το μήνυμα που ανταλλάσσεται μεταξύ των πρακτόρων. Εντούτοις, επικοινωνία προσανατολισμένη σε ομάδα δεν είναι διαθέσιμη, και η επιλογή χρησιμοποίησης ενός πληρεξούσιου για τη μετάδοση μηνύματος μπορεί να μην είναι κλιμακούμενη λύση σε καταστάσεις συχνών μεταφορών.
- **Voyager** [31]: Παρόλο που ένας αριθμός πλατφόρμων χρησιμοποιεί την Java, καμία από αυτές δεν την έχει ενσωματώσει απόλυτα, εκτός από την Voyager της ObjectSpace. Πιστοποιημένη ως *μεσίτης αιτήματος αντικειμένου (Object Request Broker - ORB)* της Java, ο οποίος έχει ενισχυθεί με την έννοια του πράκτορα, η Voyager προσφέρει διάφορους προηγμένους μηχανισμούς που θα μπορούσαν να χρησιμοποιηθούν για την υλοποίηση συστημάτων κινητών πρακτόρων. Το μοντέλο

πράκτορα βασίζεται σε μια συλλογή αντικειμένων της Java. Παρέχει μια κλάση *Πράκτορα*, την οποία οι προγραμματιστές χρησιμοποιούν για να υλοποιήσουν τον κινητό πράκτορα τύπου Voyager. Επίσης, παρέχει ένα πλήθος υπηρεσιών επικοινωνίας και υποδομής που μπορούν να χρησιμοποιηθούν για την υλοποίηση αυθαίρετων συστημάτων κινητών πρακτόρων. Ο πράκτορας Voyager είναι σχεδιασμένος ώστε να επωφελείται τα χαρακτηριστικά ORB, τα οποία κάνουν εκτενή χρήση του μηχανισμού απεικόνισης της Java. Επίσης, μπορεί να επικοινωνεί καλώντας μεθόδους ή χρησιμοποιώντας την τεχνολογία Voyager Space™. Όπως και στις άλλες πλατφόρμες, η κλάση *Πράκτορα* ενθυλακώνει ένα μοντέλο ελέγχου. Στην περίπτωση αυτή όμως, παρέχεται πιο εκτεταμένο σύνολο μηχανισμών ελέγχου, όπως πιο ευέλικτες οδηγίες για τον τρόπο που ο πράκτορας μπορεί να τερματίσει τη λειτουργία του μόνος του.

- **Grasshopper** [32]: Η πλατφόρμα Grasshopper της IKV++ συμφωνεί με τις αρχές τυποποίησης των πρακτόρων OMG (MASIF) και FIPA. Γενικά, αποτελεί βάση για την υλοποίηση υπηρεσιών που βασίζονται στην τεχνολογία των πρακτόρων στα πλαίσια των ευφυών δικτύων, των κινητών τηλεπικοινωνιών, της διαχείρισης των τηλεπικοινωνιών και των ενεργών δικτύων. Η Grasshopper υλοποιεί ένα κατανεμημένο περιβάλλον πρακτόρων, το οποίο αποτελείται από περιοχές (regions), θέσεις (places), αντιπροσωπείες (agencies) και διάφορους τύπους πρακτόρων (στατικούς και κινητούς). Οι βασικές υπηρεσίες που παρέχονται από μία κεντρική αντιπροσωπεία είναι οι ακόλουθες: επικοινωνίας, εγγραφής, διαχείρισης, ασφάλειας και διατήρησης. Εστιάζοντας στην υπηρεσία επικοινωνίας, επιτρέπονται αλληλεπιδράσεις ανεξάρτητα της θέσης μεταξύ πρακτόρων, αντιπροσωπειών αλλά και οντοτήτων που δεν είναι υλοποιημένοι σαν πράκτορες. Οι απομακρυσμένες αλληλεπιδράσεις εν γένει πραγματοποιούνται μέσω ενός συγκεκριμένου πρωτοκόλλου. Ωστόσο, η υπηρεσία επικοινωνίας της Grasshopper υποστηρίζει επικοινωνία μέσω των IIOP, RMI και socket συνδέσεων.

3.4.2 ΥΠΟΛΟΓΙΣΤΙΚΗ ΜΕ ΕΝΕΡΓΑ ΔΙΚΤΥΑ (ACTIVE NETWORKS-BASED COMPUTING)

3.4.2.1 Γενικά

Τα Ενεργά Δίκτυα είναι μια νέα προσέγγιση της αρχιτεκτονικής δικτύων, σύμφωνα με την οποία οι κόμβοι μπορούν να εκτελούν εξατομικευμένους υπολογισμούς στα πακέτα που περνούν από αυτούς. Ένα δίκτυο είναι ενεργό όταν περιλαμβάνει «νοημοσύνη» υπό τη μορφή μικρών προγραμμάτων - είτε μέσα στην υποδομή του δικτύου, είτε στα πακέτα που μεταδίδονται - που καθορίζουν τον τρόπο με τον οποίο το δίκτυο θα χειριστεί τα δεδομένα

των πακέτων. Επιπλέον, η επεξεργασία των δεδομένων μπορεί να προσαρμόζεται ανά χρήστη ή ανά εφαρμογή. Αντίθετα, ο ρόλος του υπολογισμού μέσα στα παραδοσιακά δίκτυα πακέτων είναι εξαιρετικά περιορισμένος. Αν και οι δρομολογητές μπορούν να τροποποιούν την επικεφαλίδα ενός πακέτου, περνούν τα δεδομένα των χρηστών χωρίς έλεγχο ή τροποποίηση. Επιπλέον, η επεξεργασία της επικεφαλίδας και οι σχετικές λειτουργίες των κόμβων πραγματοποιούνται ανεξάρτητα από το χρήστη ή την εφαρμογή που παράγει το πακέτο [33].

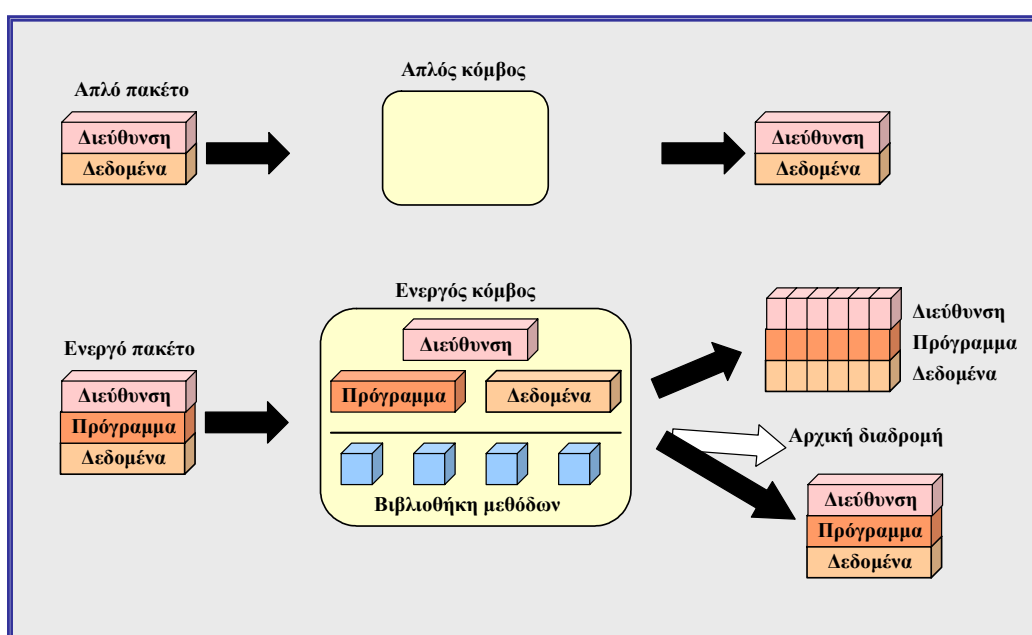
Εν συντομία, τα Ενεργά Δίκτυα υποστηρίζουν τη δυναμική τροποποίηση της συμπεριφοράς των δικτύων, αυξάνοντας την ευελιξία τους και την ικανότητα προσαρμογής τους στις απαιτήσεις των χρηστών. Παρέχοντας μια προγραμματισμένη διεπαφή στους κόμβους, «αποκαλύπτουν» τους πόρους, τους μηχανισμούς, και τις πολιτικές που κρύβονται κάτω από αυτή την αυξημένη λειτουργία και παρέχουν μηχανισμούς για τη δημιουργία ή τη βελτίωση υπηρεσιών βάσει αυτών των στοιχείων [34].

Τα βασικά στοιχεία μιας υποδομής Ενεργών Δικτύων είναι τα εξής: το περιβάλλον εκτέλεσης (*Execution Environment - EE*), ο ενεργός κώδικας (*Active Code - AC*) και ο φορέας ενεργού κώδικα (*Active Code Carrier - ACC*). Το περιβάλλον εκτέλεσης είναι το μέρος όπου εκτελείται ο ενεργός κώδικας και το οποίο παρέχει πρόσβαση στους πόρους του κόμβου. Ενεργός κώδικας είναι ο κώδικας που τελικά εκτελείται στο περιβάλλον εκτέλεσης του κόμβου. Μπορεί να είναι γραμμένος σε οποιαδήποτε γλώσσα γενικού σκοπού (π.χ. Java, C, κ.λπ.) δεδομένου ότι το περιβάλλον εκτέλεσης την υποστηρίζει και επιπλέον μπορεί να περιέχει αναφορές σε κώδικα ήδη εγκατεστημένο στον ενεργό κόμβο. Ο φορέας ενεργού κώδικα σχετίζεται με τον τρόπο μεταφοράς του ενεργού κώδικα. Αναλόγως εάν ο κώδικας και τα δεδομένα μεταφέρονται ξεχωριστά ή μαζί, διακρίνονται τρεις προσεγγίσεις Ενεργών Δικτύων, η *διακριτή*, η *ενοποιημένη* και η *μικτή*, οι οποίες αναλύονται στη συνέχεια [35].

1. Ενεργά Πακέτα (Active Packets) – Ενοποιημένη Προσέγγιση

Οι περισσότερες από τις πιο πρόσφατες αρχιτεκτονικές Ενεργών Δικτύων ακολουθούν την προσέγγιση «Ενεργών Πακέτων», η οποία χαρακτηρίζεται κυρίως από το γεγονός ότι ο κώδικας μεταφέρεται από τα πακέτα. Οι κόμβοι είναι επίσης ενεργοί, επειδή επιτρέπουν την πραγματοποίηση υπολογισμών μέχρι το στρώμα εφαρμογής, χωρίς όμως να «κατοικεί» ενεργός κώδικας σε αυτούς. Συνεπώς, ο λόγος που η δεδομένη προσέγγιση καλείται προσέγγιση «Ενεργών Πακέτων» είναι ότι ο ενεργός κώδικας φέρεται από τα πακέτα, είτε για να εκτελεσθεί πάνω στα δεδομένα του ίδιου πακέτου που φέρνει τον κώδικα, είτε για να εκτελεσθεί προκειμένου να αλλάξει την κατάσταση ή τη συμπεριφορά του κόμβου [25, 33]. Παραδείγματα τέτοιας αρχιτεκτονικής είναι το πρόγραμμα *Smart Packets* [36] του BBN, το πρόγραμμα *Active IP Option* [37] που προτάθηκε στο MIT και η αρχιτεκτονική *M0* [38], η οποία προτάθηκε από το Berkeley και το πανεπιστήμιο της Ζυρίχης.

Το Σχήμα 3.6 συγκρίνει ένα απλό με ένα ενεργό πακέτο. Ένα απλό πακέτο περιλαμβάνει ακριβώς τα δεδομένα και τη διεύθυνση αποστολής του. Ένας κόμβος απλά διαβιβάζει το πακέτο κατά μήκος της διαδρομής του ρεύματος δεδομένων. Ένα ενεργό πακέτο όμως περιλαμβάνει δεδομένα, πληροφορίες διευθύνσεων και ένα μικρό πρόγραμμα, με το οποίο υποδεικνύει στο δίκτυο πως να χειριστεί ένα ή περισσότερα πακέτα. Το πρόγραμμα μπορεί να έχει πρόσβαση στις λειτουργίες του κόμβου μέσω της βιβλιοθήκης μεθόδων του. Στο σχήμα, ο ενεργός κόμβος (επάνω μέρος) συμπιέζει, συγχωνεύει και διαβιβάζει διάφορα πακέτα, με σκοπό να εξοικονομήσει εύρος ζώνης και (κάτω μέρος) αλλάζει το δρομολόγιο ενός πακέτου για να αποφύγει τη συμφόρηση του δικτύου [39].



Σχήμα 3.6: Απλά και Ενεργά Πακέτα

2. Ενεργοί Κόμβοι (Active Nodes) – Διακριτή Προσέγγιση

Στην προσέγγιση «Ενεργών Κόμβων», τα πακέτα δεν φέρουν τον πραγματικό κώδικα, αλλά προσδιοριστικά ή αναφορές σε προκαθορισμένες λειτουργίες που «κατοικούν» στους ενεργούς κόμβους. Τα πακέτα είναι ενεργά υπό την έννοια ότι αποφασίζουν ποιες λειτουργίες πρόκειται να εκτελεστούν στα δεδομένα τους, και παρέχουν τις παραμέτρους γι' αυτές τις λειτουργίες. Εντούτοις, ο πραγματικός κώδικας υπάρχει στους ενεργούς κόμβους, δεν μεταφέρεται από τα πακέτα. Αυτός είναι και ο λόγος που η συγκεκριμένη προσέγγιση καλείται προσέγγιση «Ενεργών Κόμβων». Το κίνητρο για μια τέτοια αρχιτεκτονική είναι ότι η προσέγγιση «Ενεργών Πακέτων», αν και περισσότερο ευέλικτη, παρουσιάζει ορισμένα προβλήματα, είτε σχετικά με την επίδοση, επειδή οι απαιτήσεις ασφάλειας είναι τεράστιες,

είτε σχετικά με την αποτελεσματικότητα, επειδή ο μόνος τρόπος να ελαχιστοποιηθούν τα ζητήματα ασφάλειας είναι με τον περιορισμό των προγραμμάτων που φέρονται από τα πακέτα [25, 33, 39]. Παραδείγματα αρχιτεκτονικής «Ενεργών Κόμβων» είναι η αρχιτεκτονική που έχει προταθεί από το Georgia Institute of Technology [40], η αρχιτεκτονική *DAN* [41] που προτάθηκε από το πανεπιστήμιο της Ουάσιγκτον, και η αρχιτεκτονική *ANTS* [42] που προτάθηκε από το MIT.

3. Ενεργά Πακέτα και Ενεργοί Κόμβοι – Μικτή Προσέγγιση

Είναι προφανές ότι τα ενεργά πακέτα μπορούν να μεταφέρουν κώδικα αποτελεσματικά μόνο όταν ο κώδικας είναι σχετικά απλός και περιορισμένος. Από την άλλη μεριά, οι ενεργοί κόμβοι μπορούν να παρέχουν αποτελεσματικά οποιοδήποτε κώδικα. Εντούτοις, αυτός ο κώδικας είναι προκαθορισμένος επειδή πρέπει να υπάρχει στον ενεργό κόμβο ή τουλάχιστον σε έναν κόμβο από τον οποίο μπορεί να ανακτηθεί. Σύμφωνα με τη μικτή προσέγγιση, τα ενεργά πακέτα φέρουν τον πραγματικό κώδικα, ενώ άλλος, πιο σύνθετος κώδικας, μπορεί να υπάρχει στους ενεργούς κόμβους. Επομένως, τα πλεονεκτήματα των δύο προηγούμενων προσεγγίσεων υπάρχουν σε ένα σύστημα. Συνήθως, τέτοιες αρχιτεκτονικές επιτρέπουν στους χρήστες να επιλέξουν την πιο κατάλληλη προσέγγιση σύμφωνα με τη φύση της εκάστοτε εφαρμογής [25, 39]. Χαρακτηριστικά παραδείγματα είναι η αρχιτεκτονική *SwitchWare* [43] που προτάθηκε από το πανεπιστήμιο της Πενσυλβάνια, και η αρχιτεκτονική *NetScript* [44], η οποία προτάθηκε από το πανεπιστήμιο της Κολούμπια.

3.4.2.2 Πλεονεκτήματα Χρήσης Ενεργών Δικτύων

Τα βασικότερα πλεονεκτήματα από τη χρήση των Ενεργών Δικτύων είναι τα εξής:

- Από την πλευρά του παρόχου υπηρεσιών δικτύου, τα Ενεργά Δίκτυα έχουν τη δυνατότητα να μειώνουν το χρόνο δημιουργίας (creation) και ανάπτυξης (deployment) νέων υπηρεσιών δικτύου. Οι δικτυακές καινοτομίες εξαρτώνται συχνά από τη χρονοβόρα ανάπτυξη νέων τεχνολογιών και προτύπων υλισμικού. Ωστόσο, αυτή η διαδικασία δεν μπορεί να συμβαδίσει με την αυξανόμενη ζήτηση για βελτιώσεις στη δικτύωση. Με τα Ενεργά Δίκτυα όμως, νέες υπηρεσίες δικτύου υλοποιούνται πιο εύκολα επειδή βασίζονται σε λογισμικό. Συνεπώς, μέσω της δυνατότητας για δυναμική αλλαγή της συμπεριφοράς των κόμβων του δικτύου, οι πάροχοι υπηρεσιών μπορούν να αναπτύσσουν νέες υπηρεσίες γρήγορα, χωρίς να περνούν από μια μακρά διαδικασία τυποποίησης [34, 39].
- Τα Ενεργά Δίκτυα επιτρέπουν στους χρήστες να προσαρμόζουν τις υπηρεσίες στις ιδιαίτερες απαιτήσεις τους και στην τρέχουσα κατάσταση του δικτύου. Παρόλο δε που οι περισσότεροι τελικοί χρήστες δεν είναι σε θέση να γράφουν οι ίδιοι προγράμματα για το δίκτυο, είναι εύκολο να σκεφτούμε άτομα που προσαρμόζουν τις υπηρεσίες τους δοκιμάζοντας διαφορετικές επιλογές στον κώδικα που παρέχεται από τρίτους. Αυτή η

προοπτική είναι ελκυστική και για τους παρόχους δικτύου, αφού τους επιτρέπει να χρεώνουν περισσότερο τέτοιες υπηρεσίες προστιθεμένης αξίας [34, 33].

- Πραγματοποιώντας υπολογισμούς σε στρατηγικά σημεία μέσα στο δίκτυο, είτε μέσω της διακριτής είτε μέσω της ενοποιημένης προσέγγισης Ενεργών Δικτύων, είναι δυνατό να μειωθεί η κυκλοφορία σε περιορισμένου εύρους ζώνης μέρη του δικτύου [39].
- Δεδομένου ότι η ανάπτυξη και η διαχείριση ενός δικτύου είναι ακριβή διαδικασία, ένα δυναμικά προγραμματίσιμο δίκτυο δύναται να προσφέρει στους ερευνητές πλατφόρμα για πειραματισμό με νέες υπηρεσίες δικτύου και τα χαρακτηριστικά τους, σε πραγματική κλίμακα και χωρίς καμία αλλοίωση των υπηρεσιών [34].

3.4.2.3 Εφαρμογές Ενεργών Δικτύων

Μερικές από τις περιοχές εφαρμογών που επωφελούνται από τη χρήση Ενεργών Δικτύων είναι οι εξής:

□ Διαχείριση Δικτύων (*Network Management*)

Η κλασική προσέγγιση διαχείρισης δικτύων βασίζεται σε υποβολή ερωτήσεων στις διαχειριζόμενες συσκευές από το σταθμό διαχείρισης, ο οποίος ζητά τις τιμές συγκεκριμένων μεταβλητών και ελέγχει για ανωμαλίες. Αυτή η προσέγγιση στηρίζεται σε συγκέντρωση της ευφυΐας στο σταθμό διαχείρισης, με συνέπεια να προκαλεί συμφόρηση σε επίπεδο επεξεργασίας και επικοινωνίας, ενώ περιορίζει σημαντικά και τη δυνατότητα ανίχνευσης των προβλημάτων με τρόπο έγκαιρο και αποδοτικό. Εν τω μεταξύ, πολλές εργασίες διαχείρισης απαιτούν συλλογή και συσχέτισμό δεδομένων, όπως για παράδειγμα στην περίπτωση των ανιχνευτών γεγονότων. Για να παρέχονται χρήσιμα δεδομένα διαχείρισης, όπως ενδείξεις εξαιρέσεων, πρέπει να χρησιμοποιηθεί ευφυΐα μέσα στο δίκτυο, ώστε να απομακρύνονται τα άσχετα γεγονότα. Τα Ενεργά Δίκτυα μπορούν να χρησιμοποιηθούν για να εφαρμόσουν σύνθετες προσεγγίσεις στην παρακολούθηση δικτύων και φιλτράρισμα γεγονότων μέσα στο δίκτυο. Τα στοιχεία του δικτύου, όπως οι δρομολογητές, μπορούν ακόμη και να αναλάβουν έναν βαθμό ευθύνης για την παρακολούθησή τους, π.χ. με την έγχυση προσαρμοσμένων προγραμμάτων ελέγχου και διάγνωσης στους κοντινότερους γείτονές τους. Επίσης, τα Ενεργά Δίκτυα μπορούν να παρέχουν την απαραίτητη ευελιξία για βελτίωση της ανίχνευσης λαθών και ανανέωση των πολιτικών επιβίωσης που χαρακτηρίζουν την απόκριση των στοιχείων σε αποτυχίες, οι οποίες προκαλούνται, για παράδειγμα, από κακόβουλους εισβολείς [33, 34].

□ Προστασία Δικτύου με Επίγνωση του Χρήστη (*User-Aware Network Protection*)

Η προστασία των πληροφοριών σημαίνει ότι οι σωστές πληροφορίες διανέμονται στους σωστούς ανθρώπους, στο σωστό τόπο και χρόνο. Αν και η ασφάλεια δικτύων σε συνδυασμό με μηχανισμούς πιστοποίησης προτείνεται σε πολλά φόρουμ δικτύωσης, η ενεργή δικτύωση επιτρέπει το σχεδιασμό ενός ενιαίου μηχανισμού που ελέγχει όλους τους πόρους δικτύου και

τις πληροφορίες που διαπερνούν από αυτούς. Αυτό εξαλείφει την ανάγκη για πολλαπλά συστήματα ασφάλειας-πιστοποίησης που λειτουργούν ανεξάρτητα σε κάθε στρώμα πρωτοκόλλου επικοινωνίας και επιτρέπει τον προγραμματισμό πολιτικών ασφάλειας δικτύου ανά χρήστη ή ανά χρήση [33].

□ *Ποιότητα Υπηρεσίας (Quality of Service)*

Συνθήκες δικτύου όπως η παρουσία συμφόρησης ή μιας σύνδεσης με απώλειες μπορούν να υποβιάσουν σημαντικά την ποιότητα των ροών εφαρμογών. Οι μηχανισμοί που απαιτούν ο αποστολέας να προσαρμόζεται στην κατάσταση του δικτύου παρουσιάζουν περιορισμούς, συμπεριλαμβανομένου του χρόνου που απαιτεί ο αποστολέας να ανιχνεύσει την κατάσταση, να αντιδράσει, και να μεταδώσει τα αναπροσαρμοσμένα δεδομένα στο δέκτη. Κατά τη διάρκεια του διαστήματος προσαρμογής, ο δέκτης ενδέχεται να βιώσει ανεξέλεγκτες απώλειες (εάν οι συνθήκες επιδεινωθούν) ή μη-βέλτιστη απόδοση (εάν οι συνθήκες βελτιωθούν). Μετακινώντας όμως πληροφορία σχετική με τον τρόπο προσαρμογής του δικτύου στις εκάστοτε συνθήκες στους κόμβους του δικτύου, ο κατάλληλος τύπος προσαρμογής μπορεί να συμβεί όταν και όπου απαιτείται [34, 45]. Προσπάθειες προς αυτή την κατεύθυνση περιλαμβάνουν ευφυείς στρατηγικές απόρριψης για προστασία της ποιότητας MPEG βίντεο κατά τη διάρκεια συμφόρησης του δικτύου [40].

□ *Προσωρινή Φύλαξη Αντικειμένων (Caching)*

Ένα σημαντικό μέρος της κίνησης στα δίκτυα προέρχεται από εφαρμογές στις οποίες οι πελάτες ανακτούν αντικείμενα από τους εξυπηρετητές. Η προσωρινή φύλαξη των αντικειμένων σε σημεία κοντά στους πελάτες θεωρείται σημαντική τεχνική για τη μείωση της κίνησης στο δίκτυο, καθώς και του χρόνου απόκρισης. Τα σχέδια προσωρινής φύλαξης απαιτούν τη λήψη αποφάσεων σχετικών με τις θέσεις φύλαξης των αντικειμένων αλλά και τον τρόπο προώθησης των αιτημάτων μεταξύ των κρυπτών. Συνήθως οι κρύπτες διαρθρώνονται χειροκίνητα σε μια στατική ιεραρχία, προκαλώντας μεγάλο φορτίο διαχείρισης και περιορίζοντας την ικανότητα αντίδρασης σε δυναμικές καταστάσεις [34]. Οι προσπάθειες χρησιμοποίησης Ενεργών Δικτύων για τη φύλαξη των αντικειμένων προβλέπουν τη χρησιμοποίηση μηχανισμών για δρομολόγηση των αιτημάτων σε προκαθορισμένες θέσεις κρυπτών [46], και το συνδυασμό μικρών κρυπτών με πληροφορίες περιεχομένου για κοντινές κρύπτες σε κάθε κόμβο του δικτύου [47].

□ *Συγχώνευση και Διανομή Πληροφορίας (Merging and Distribution of Information)*

Υπάρχει πλήθος εφαρμογών, οι οποίες απαιτούν περίπλοκες υπηρεσίες που βασίζονται στο δίκτυο για να υποστηρίξουν τη συγχώνευση και τη διανομή πληροφοριών. Η πραγματοποίηση της συγχώνευσης των πληροφοριών μέσα στο δίκτυο μειώνει τις απαιτήσεις εύρους ζώνης των χρηστών, οι οποίοι βρίσκονται στην περιφέρεια του δικτύου.

Τοποθετώντας έναν υπολογισμό της εφαρμογής κοντά στο σημείο όπου αυτός απαιτείται, βοηθά επίσης να ξεπεραστούν περιορισμοί καθυστέρησης, μειώνοντας τους κρίσιμους βρόχους ανάδρασης των διαδραστικών εφαρμογών.

3.4.3 ΣΥΓΚΡΙΣΗ

Το υπόδειγμα Κινητών Πρακτόρων χειρίζεται το δίκτυο ως ένα σύνολο από περιβάλλοντα «φιλικά» προς τον πράκτορα, και τους πράκτορες ως προγραμματιστικές οντότητες που κινούνται από θέση σε θέση, εκτελώντας αποστολές για λογαριασμό των χρηστών ή άλλων οντοτήτων λογισμικού. Τα Ενεργά Δίκτυα θεωρούν το δίκτυο σαν μια συλλογή ενεργών κόμβων, οι οποίοι μπορούν να εκτελέσουν οποιοδήποτε υπολογισμό, και ενεργών πακέτων που φέρουν τον κώδικα. Η ομοιότητα μεταξύ των δυο τεχνολογιών είναι προφανής. Ένας κινητός πράκτορας θα μπορούσε να θεωρηθεί ως συγκεκριμένος τύπος ενεργού πακέτου, και ένας κόμβος συμβατός με κινητούς πράκτορες θα μπορούσε να θεωρηθεί ως συγκεκριμένος τύπος ενεργού κόμβου. Παρόλο που οι τεχνολογίες αυτές προέρχονται από διαφορετικές ερευνητικές κοινότητες, τείνουν σήμερα να μοιραστούν κοινές ιδέες, ξεπερνώντας το παραδοσιακό μοντέλο Πελάτη-Εξυπηρετητή και εφαρμόζοντας πλήρως καταναμημένα μοντέλα με σκοπό τη καλύτερη επίδοση των συστημάτων και την πιο αποδοτική χρήση των πόρων.

Ωστόσο, οι τεχνολογίες αυτές παρουσιάζουν ορισμένες αξιοσημείωτες διαφορές. Καταρχήν, εφαρμόζονται σε διαφορετικά επίπεδα του προτύπου OSI. Η τεχνολογία των Κινητών Πρακτόρων εφαρμόζεται στο επίπεδο εφαρμογών, ενώ αυτή των Ενεργών Δικτύων στο επίπεδο δικτύου. Συνεπώς, στη περίπτωση των Ενεργών Δικτύων καμιά ειδική μέριμνα δεν έχει ληφθεί για το μηχανισμό μεταφοράς του ενεργού κώδικα, δεδομένου ότι τα στοιχεία του δικτύου περιέχουν το κατάλληλο λογισμικό για την αποστολή των πακέτων, και έτσι η κύρια μέριμνα έγκειται στην επεξεργασία των πακέτων κατά τη λήψη τους από τους κόμβους του δικτύου. Η λειτουργία στο επίπεδο δικτύου αποκαλύπτει την πραγματική δυνατότητα των Ενεργών Δικτύων να εκμεταλλευτούν αμεσότερα και αποτελεσματικότερα τις υπηρεσίες δικτύου, πετυχαίνοντας γρήγορη ανάπτυξη καταναμημένων εφαρμογών. Αντιθέτως, το χαμηλότερο επίπεδο του προτύπου OSI με το οποίο η τεχνολογία των Κινητών Πρακτόρων ασχολείται προγραμματιστικά είναι το επίπεδο μεταφοράς, χρησιμοποιώντας απλά τις δυνατότητες επικοινωνίας που προσφέρει το περιβάλλον (π.χ. Java RMI). Το γεγονός λοιπόν ότι η τεχνολογία των Κινητών Πρακτόρων εντοπίζεται στο επίπεδο εφαρμογών, την καθιστά πιο αποδοτική όταν εμπλέκονται αλληλεπιδράσεις με συστήματα εφαρμογών.

Επίσης, η λειτουργία των Ενεργών Δικτύων απαιτεί την εφαρμογή ορισμένων επιπλέον κανόνων στους δρομολογητές, οι οποίοι τους καθιστούν ικανούς να αναγνωρίζουν τα ενεργά πακέτα και να τα μεταχειρίζονται αναλόγως, δηλαδή να τα προωθούν στο κατάλληλο

περιβάλλον εκτέλεσης προκειμένου να εκτελεστούν. Από αυτή την άποψη, η αναβάθμιση ενός κόμβου ώστε να γίνει αυτός συμβατός με τη τεχνολογία Ενεργών Δικτύων προϋποθέτει επιπλέον ρυθμίσεις.

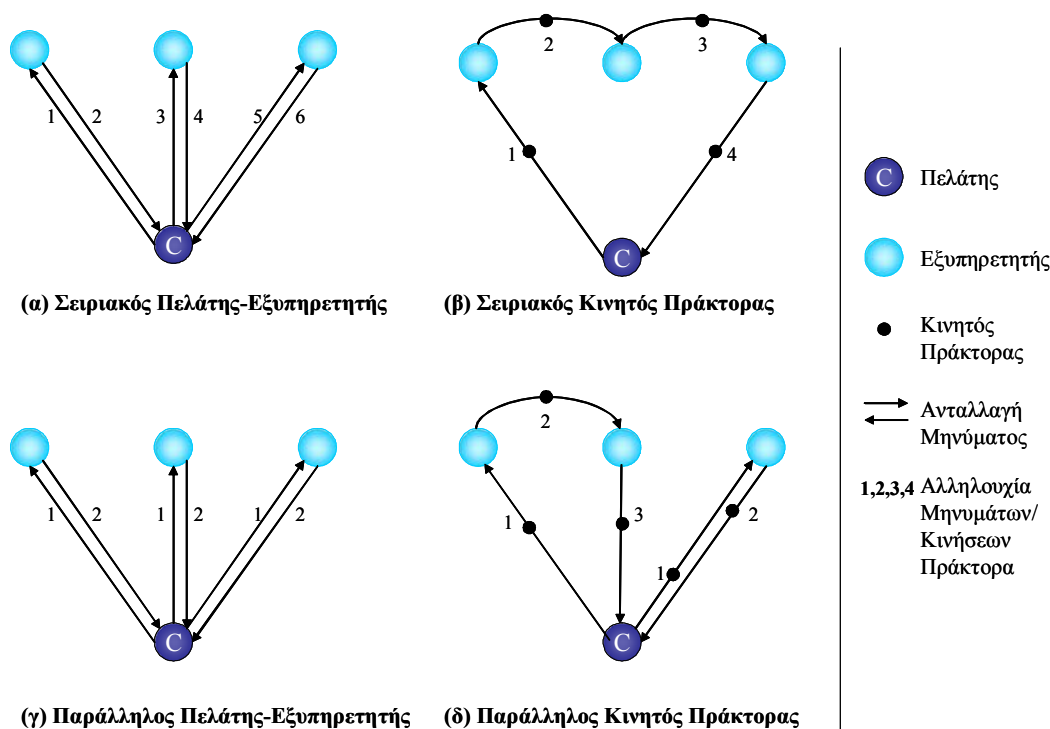
Τέλος, αξίζει να σημειωθεί ότι ενώ το ωφέλιμο φορτίο ενός ενεργού πακέτου στη περίπτωση των Ενεργών Δικτύων αποτελείται μόνο από το πρόγραμμα και τα δεδομένα, ένας κινητός πράκτορας μεταφέρει επιπλέον και την κατάστασή του καθώς μετακινείται μεταξύ των κόμβων του δικτύου. Η αυτονομία αυτή, επιτρέπει στον πράκτορα να λαμβάνει αποφάσεις βάσει της τρέχουσας κατάστασής του και να μετακινείται στο δίκτυο αποκλειστικά υπό τον έλεγχό του, δηλαδή να σταματάει την εκτέλεσή του σε έναν κόμβο και να τη συνεχίζει σε κάποιον άλλο, πράγμα το οποίο δεν υποστηρίζεται από ένα ενεργό πακέτο.

3.5 ΠΡΟΒΛΗΜΑ ΣΥΛΛΟΓΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕ ΚΙΝΗΤΟΥΣ ΠΡΑΚΤΟΡΕΣ

Στην παράγραφο αυτή μελετάται η χρήση κινητών πρακτόρων στα πλαίσια ενός προβλήματος συλλογής κατανεμημένης πληροφορίας σε σύστημα τριμερούς αρχιτεκτονικής [48, 49]. Σκοπός του προβλήματος είναι η ελαχιστοποίηση του συνολικού χρόνου συλλογής της απαιτούμενης πληροφορίας και βασίζεται στη χρήση κατάλληλων σχεδίων δρομολόγησης των κινητών πρακτόρων.

3.5.1 ΕΠΙΛΟΓΗ ΜΟΝΤΕΛΟΥ ΕΠΙΚΟΙΝΩΝΙΑΣ

Με βάση τη γενική αρχιτεκτονική Πελάτη-Εξυπηρετητή του συστήματος Smart-EC που παρουσιάστηκε στο κεφάλαιο 2, και λαμβάνοντας υπόψη τα μοντέλα επικοινωνίας που αναπτύχθηκαν στην παράγραφο 3.3, η επικοινωνία του μεσίτη με τους εξωτερικούς παρόχους για ανάκτηση πληροφορίας μπορεί να επιτευχθεί σύμφωνα με τις στρατηγικές που απεικονίζονται στο Σχήμα 3.7. Σημειώνεται ότι ο πελάτης αντιστοιχεί στο μεσίτη και κάθε εξυπηρετητής σε έναν πάροχο.



Σχήμα 3.7: Στρατηγικές Υλοποίησης Επικοινωνίας Μεσίτη-Παρόχων

α. Σειριακός Πελάτης-Εξυπηρετητής

Βασίζεται στο παραδοσιακό μοντέλο Πελάτη-Εξυπηρετητή. Ο πελάτης υποβάλλει το αίτημά του στον πρώτο εξυπηρετητή και αφού επεξεργαστεί την απάντηση, υποβάλλει το αίτημα στον δεύτερο, και ούτω καθ'εξής, μέχρι να εξαντληθεί ο αριθμός των υποψήφιων εξυπηρετητών (Σχήμα 3.7-α).

β. Σειριακός Κινητός Πράκτορας

Στην περίπτωση αυτή ο κινητός πράκτορας μετακινείται από την πηγή προέλευσής του στη θέση του πρώτου εξυπηρετητή. Στη συνέχεια μετακινείται στη θέση του δεύτερου και ούτω καθεξής, μέχρι να εξαντληθούν όλοι (Σχήμα 3.7-β).

γ. Παράλληλος Πελάτης-Εξυπηρετητής

Βασίζεται επίσης στο μοντέλο Πελάτη-Εξυπηρετητή. Ωστόσο, αντί για διαδοχικά αιτήματα, ο πελάτης πραγματοποιεί παράλληλες ακολουθίες εκτέλεσης, όπου η κάθε ακολουθία περιλαμβάνει υποβολή του αιτήματος σε δεδομένο εξυπηρετητή και επεξεργασία της επιστρεφόμενης απάντησης (Σχήμα 3.7-γ).

δ. Παράλληλος Κινητός Πράκτορας

Στην περίπτωση αυτή ο πελάτης αποστέλλει πολλούς κινητούς πράκτορες ταυτόχρονα, καθένας από τους οποίους επισκέπτεται ένα υποσύνολο των εξυπηρετητών. Οι πράκτορες στη

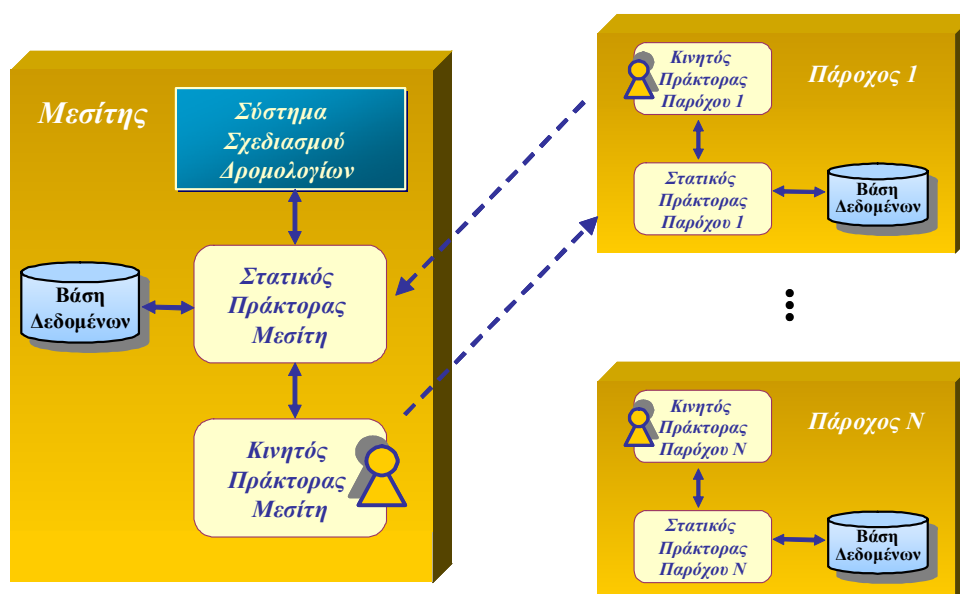
συνέχεια επιστρέφουν στο σημείο εκκίνησής τους, όπου ταξινομούν και συγχωνεύουν τις απαντήσεις τους για την περάτωση της αποστολής τους (Σχήμα 3.7-δ).

Εάν υποθέσουμε ότι το σύνολο των κόμβων-εξυπηρετητών που πρόκειται να προσπελαστούν είναι προκαθορισμένο και επιπλέον η σειρά επίσκεψής τους είναι στατική και γνωστή εκ των προτέρων, οι δυνατές στρατηγικές υλοποίησης είναι η α και η β . Εάν όμως η σειρά επίσκεψης δεν έχει σημασία και θα μπορούσε να αποφασιστεί και δυναμικά, τότε και οι τέσσερις στρατηγικές είναι δυνατές. Στην περίπτωση που το σύνολο των κόμβων που προσπελάζονται καθορίζεται δυναμικά, τότε οι σειριακές και μόνο υλοποιήσεις έχουν νόημα, δηλαδή η α και η β .

Στο πρόβλημα που μελετάμε θεωρούμε ότι η περιοχή του μεσίτη γνωρίζει εκ των προτέρων το σύνολο των κόμβων από τους οποίους πρόκειται να ανακτηθεί η επιθυμητή πληροφορία και επιπλέον δεν απαιτείται συγκεκριμένη σειρά επίσκεψης των κόμβων αυτών. Συνεπώς, όλες οι παραπάνω στρατηγικές είναι δυνατές. Ωστόσο, η στρατηγική δ παρουσιάζει ιδιαίτερο ενδιαφέρον για τους εξής λόγους: Καταρχάς, ο κινητός πράκτορας μετακινεί τη συνιστώσα υπολογισμού κοντά στα δεδομένα προς επεξεργασία, με αποτέλεσμα να εκτελεί τις απαιτούμενες διαδικασίες αναζήτησης τοπικά, εξαλείφοντας την ανάγκη μεταφοράς περιττών δεδομένων μέσω του δικτύου. Εάν η εφαρμογή απαιτεί επεξεργασία μεγάλου όγκου πληροφορίας στον απομακρυσμένο κόμβο, η σημαντική μείωση της πληροφορίας που θα επιστραφεί μέσω του κατάλληλου φιλτραρίσματος που εκτελεί ο πράκτορας υποσκελίζει την επιβάρυνση που προκαλείται από την αποστολή του πράκτορα στον εξυπηρετητή. Επιπλέον, με την παράλληλη αποστολή πολλών πρακτόρων, μπορούν να επιλέγονται διαφορετικές στρατηγικές μετακίνησης των πρακτόρων ανάλογα με το έργο και την τρέχουσα κατάσταση του δικτύου και με σκοπό την ταχύτερη εκπλήρωση του έργου. Επιπρόσθετα, με τη μετανάστευση στην άλλη πλευρά μιας σύνδεσης, ο πράκτορας μπορεί να συνεχίσει να λειτουργεί ακόμα και αν η σύνδεση διακοπεί. Το γεγονός αυτό καθιστά τους κινητούς πράκτορες ιδιαίτερα ελκυστικούς σε περιβάλλον κινητής υπολογιστικής (mobile computing environment), το οποίο συχνά αντιμετωπίζει αναξιόπιστες συνδέσεις χαμηλού εύρους ζώνης και υψηλής καθυστέρησης.

3.5.2 ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ ΠΡΑΚΤΟΡΩΝ

Στη συνέχεια περιγράφεται το σύστημα κινητών και σταθερών πρακτόρων που αλληλεπιδρούν μεταξύ τους στα πλαίσια διανομής της πληροφορίας στο σύστημα Smart-EC, αντιστοιχίζοντας τους πράκτορες σε γνωστούς τύπους [50].



Σχήμα 3.8: Σύστημα Στατικών και Κινητών Πρακτόρων

Στην πλευρά του μεσίτη ορίζονται οι εξής δυο τύποι πρακτόρων (Σχήμα 3.8):

1. Στατικός Πράκτορας Μεσίτη (Static Broker Agent - SBA)

Ο πράκτορας αυτός εδρεύει μόνιμα στην πλευρά του μεσίτη και έχει τις εξής αρμοδιότητες:

- 1) Να αρχικοποιεί και να αποστέλλει τους κινητούς πράκτορες στους παρόχους για ανάκτηση της απαιτούμενης πληροφορίας, αφού προηγουμένως έχει επικοινωνήσει με το Σύστημα Σχεδιασμού Δρομολογίων, ώστε να λάβει οδηγίες για τη δρομολόγηση των κινητών πρακτόρων. Στην περίπτωση αυτή λειτουργεί ως *πράκτορας σχεδιασμού (planning agent)*
- 2) Δέχεται πληροφορίες από τους εξωτερικούς παρόχους και τις ενσωματώνει στο σύστημα Smart-EC, επιδεικνύοντας χαρακτηριστικά *πράκτορα ενσωμάτωσης (integration agent)*
- 3) Επιτελεί λειτουργίες πιστοποίησης, ως *πράκτορας πιστοποίησης (authentication agent)*, δεδομένου ότι πριν την εισαγωγή οποιασδήποτε πληροφορίας στη βάση του συστήματος ελέγχει τα παρεχόμενα διαπιστευτήρια για την πιστοποίηση της πηγής από όπου προήλθε η πληροφορία.

2. Κινητός Πράκτορας Μεσίτη (Mobile Broker Agent - MBA)

Πρόκειται για τον κινητό πράκτορα που αναλαμβάνει να αντιπροσωπεύσει το μεσίτη και να μεταφέρει το αίτημά του στην πλευρά των παρόχων. Κατά την επίσκεψη σε έναν πάροχο, αναζητά την απαιτούμενη πληροφορία αλληλεπιδρώντας με έναν σταθερό πράκτορα, όπως περιγράφεται στη συνέχεια, και αφού εξαντλήσει τον αριθμό των υποψηφίων παρόχων επιστρέφει στην αρχική του θέση με τα αποτελέσματα των ερευνών του. Ο πράκτορας αυτός συγκαταλέγεται στην κατηγορία των *πρακτόρων αναζήτησης (search agent)*.

Αντίστοιχα, στην πλευρά κάθε παρόχου ορίζονται οι εξής δυο τύποι πρακτόρων (Σχήμα 3.8):

1. Στατικός Πράκτορας Παρόχου (Static Provider Agent - SPA)

Ο πράκτορας αυτός υπάρχει μόνιμα σε κάθε πάροχο και εκτελεί τις εξής λειτουργίες: 1) Αλληλεπιδρά με τον κινητό πράκτορα MBA, ο οποίος έχει αποσταλεί από τον μεσίτη, και στη συνέχεια επικοινωνεί με την τοπική βάση δεδομένων, αντλεί την κατάλληλη πληροφορία και την προωθεί στον MBA για την επίλυση του αιτήματός του. Στα πλαίσια αυτής της λειτουργίας χαρακτηρίζεται ως *πράκτορας μετάφρασης (translation agent)*, δεδομένου ότι αντιστοιχεί στο επίπεδο μετατροπέα (wrapper-tier) ενός μεσιτικού συστήματος, προσδιορίζοντας τον τρόπο ανάκτησης των δεδομένων από μια συγκεκριμένη πηγή καθώς και τον τρόπο μετατροπής τους στο κοινό μοντέλο που έχει υιοθετηθεί από το σύστημα 2) Αρχικοποιεί και αποστέλλει κινητό πράκτορα (ο οποίος περιγράφεται στη συνέχεια) στην πλευρά του μεσίτη, στα πλαίσια της υπό συνθήκη ώθησης πληροφορίας.

2. Κινητός Πράκτορας Παρόχου (Mobile Provider Agent - MPA)

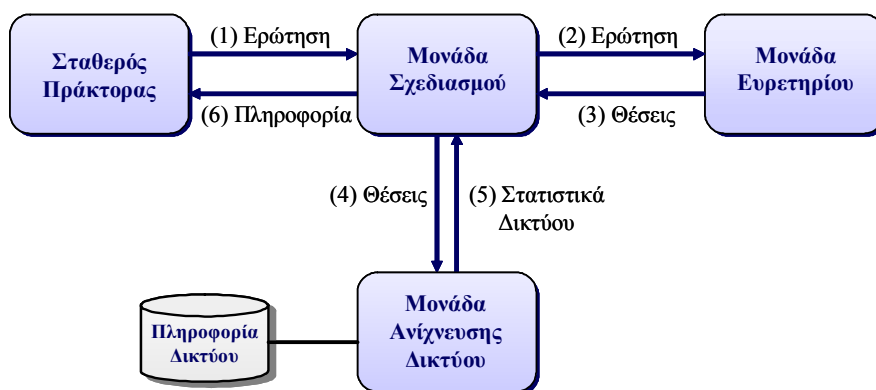
Είναι εξουσιοδοτημένος να λειτουργεί σαν αντιπρόσωπος του εκάστοτε παρόχου όταν αυτός επιθυμεί να αλληλεπιδράσει με το μεσίτη, και για το λόγο αυτό κατατάσσεται στην κατηγορία των *προσωπικών πρακτόρων (personal agents)*. Οι λειτουργίες που αναλαμβάνει σχετίζονται με την εισαγωγή, την ανανέωση ή τη διαγραφή πληροφοριών από τη βάση δεδομένων του συστήματος.

Στα πλαίσια της συλλογής κατανεμημένης πληροφορίας από πολλαπλούς παρόχους βάσει της στρατηγικής Παράλληλου Κινητού Πράκτορα, ο Στατικός Πράκτορας Μεσίτη θα μπορούσε απλά να στέλνει έναν κινητό πράκτορα αναζήτησης σε κάθε κόμβο που παρέχει σχετική πηγή πληροφορίας. Εάν όμως επιθυμούμε να καταναλωθούν λιγότεροι πόροι δικτύου, θα πρέπει να αποσταλεί μικρότερος αριθμός πρακτόρων από τον αριθμό πηγών πληροφορίας. Στην περίπτωση αυτή προκύπτει η ανάγκη για ένα μηχανισμό προγραμματισμού που θα αποφασίζει σχετικά με την ανάθεση κόμβων σε πράκτορες και τη βέλτιστη σειρά επίσκεψης των κόμβων για κάθε πράκτορα, έτσι ώστε η διαδικασία συλλογής της επιθυμητής πληροφορίας να πραγματοποιείται στον ελάχιστο χρόνο. Ο μηχανισμός αυτός υλοποιείται από το **Σύστημα Σχεδιασμού Δρομολογίων**. Συνεπώς, ο Στατικός Πράκτορας Μεσίτη αλληλεπιδρά με το εν λόγω σύστημα και αντλεί τις απαιτούμενες οδηγίες έτσι ώστε να μπορεί να επιλέξει την αποδοτικότερη στρατηγική μετακίνησης ενός συνόλου κινητών πρακτόρων με βάση τις ανάγκες του συγκεκριμένου έργου και την τρέχουσα κατάσταση του δικτύου. Η αρχιτεκτονική του συστήματος αυτού περιγράφεται στη συνέχεια.

3.5.3 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ ΣΧΕΔΙΑΣΜΟΥ ΔΡΟΜΟΛΟΓΙΩΝ

Η αρχιτεκτονική του Συστήματος Σχεδιασμού Δρομολογίων των κινητών πρακτόρων απεικονίζεται στο Σχήμα 3.9. Αποτελείται από τρεις βασικές συνιστώσες: τη *μονάδα σχεδιασμού*, τη *μονάδα ανίχνευσης δικτύου* και τη *μονάδα ευρετηρίου*. Όταν ο Στατικός

Πράκτορας Μεσίτη αναλαμβάνει να συλλέξει συγκεκριμένη πληροφορία, απευθύνεται αρχικά στη μονάδα σχεδιασμού. Η μονάδα σχεδιασμού συμβουλευεται τη μονάδα ευρετηρίου για να μάθει τις θέσεις των κόμβων που παρέχουν την πληροφορία που ζητήθηκε. Αφού λάβει τη λίστα με τις σχετικές θέσεις, την προωθεί στη μονάδα ανίχνευσης δικτύου, η οποία με τη σειρά της επιστρέφει πληροφορίες σχετικές με την τρέχουσα κατάσταση του δικτύου, όπως τη συνδετικότητα των γραμμών, τη λειτουργικότητα των κόμβων ή την καθυστέρηση και το εύρος ζώνης των συνδέσεων. Η μονάδα ανίχνευσης δικτύου παρέχει αυτά τα στατιστικά εξετάζοντας το δίκτυο σε τακτά χρονικά διαστήματα. Βάσει των λαμβανόμενων στατιστικών δικτύου, η μονάδα σχεδιασμού υπολογίζει τα δρομολόγια των πρακτόρων που ελαχιστοποιούν τον συνολικό αναμενόμενο χρόνο εκτέλεσης, κάνοντας χρήση αλγορίθμων βέλτιστου σχεδιασμού δρομολογίων, οι οποίοι εξετάζονται στο κεφάλαιο 4.



Σχήμα 3.9: Σύστημα Σχεδιασμού Δρομολογίων

3.6 ΠΡΟΒΛΗΜΑ ΑΝΑΖΗΤΗΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕ ΕΝΕΡΓΑ ΔΙΚΤΥΑ

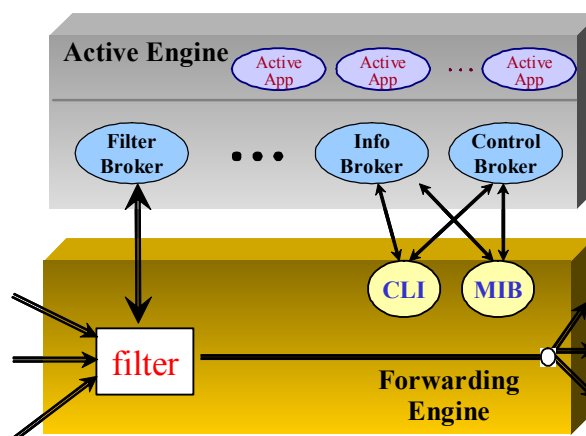
Στη συνέχεια μελετάται η αξιοποίηση της τεχνολογίας Ενεργών Δικτύων από ένα σύστημα ομότιμων μεσιτών. Στα πλαίσια βελτίωσης της επίδοσης του συστήματος κρίνεται σκόπιμη η μελέτη ενός προβλήματος αναζήτησης πληροφορίας.

3.6.1 ΕΦΑΡΜΟΓΗ ΕΝΕΡΓΩΝ ΔΙΚΤΥΩΝ

Η ανάπτυξη υπηρεσιών με επίγνωση του περιβάλλοντος έχει καθυστερήσει εξαιτίας της έλλειψης εξειδικευμένης υποδομής πληροφορίας για κάθε εφαρμογή. Στην προσπάθεια να ξεπεράσει αυτή τη δυσκολία, το CONTEXT κατέφυγε στην τοποθέτηση λειτουργικότητας σχετικής με την πληροφορία μέσα στη δικτυακή υποδομή υπό τη μορφή ενεργών συνιστωσών, κάνοντας χρήση της τεχνολογίας Ενεργών Δικτύων. Σκοπός είναι η δυναμική και αυτόματη προσαρμογή των υπηρεσιών προς όφελος του χρήστη, μέσω της κατανεμημένης λειτουργίας και της δυνατότητας εκτέλεσης κώδικα μέσα στο δίκτυο. Το όφελος της κατανεμημένης λειτουργίας γίνεται αντιληπτό από το γεγονός ότι οι διάφορες πηγές πληροφορίας περιβάλλοντος είναι κατανεμημένες στο δίκτυο. Στην περίπτωση αυτή,

ένα κεντρικό σχέδιο ανάκτησης και διανομής της πληροφορίας δεν θα λειτουργούσε αποδοτικά. Από την άλλη πλευρά, η δυνατότητα εκτέλεσης κώδικα μέσα στο δίκτυο παρέχει τη δυνατότητα εύκολης και γρήγορης προσαρμογής των υπηρεσιών σε τυχόν αλλαγές της πληροφορίας περιβάλλοντος χωρίς την παρέμβαση των χρηστών [51].

Η πλατφόρμα Ενεργών Δικτύων DINA αναπτύχθηκε στα πλαίσια του CONTEXT ως επέκταση της πλατφόρμας ABLE (Active Bell Labs Engine) [52]. Η DINA είναι μια κλιμακούμενη αρχιτεκτονική λογισμικού, η οποία επιτρέπει την ανάπτυξη, τον έλεγχο και τη διαχείριση ενεργών εφαρμογών πάνω από στοιχεία δικτύου όπως δρομολογητές, σημεία πρόσβασης WLAN, πύλες μέσων, και εξυπηρετητές που υποστηρίζουν τέτοιες εφαρμογές σε δίκτυα IP. Επιπλέον, παρέχει διεπαφές που μπορούν να χρησιμοποιηθούν από τις ενεργές εφαρμογές για τη διαχείριση, τον έλεγχο και την ανάκτηση πληροφορίας ή την εκτέλεση άλλων λειτουργιών τοπικά σε κάποιο κόμβο. Σημειώνεται ότι η εν λόγω αρχιτεκτονική ακολουθεί της αρχές της μικτής προσέγγισης (Ενεργά Πακέτα και Ενεργοί Κόμβοι), δεδομένου ότι επιτρέπει ο ενεργός κώδικας να μεταφέρεται μέσω ενεργών πακέτων ή να αποθηκεύεται σε συγκεκριμένα σημεία μέσα στο δίκτυο.

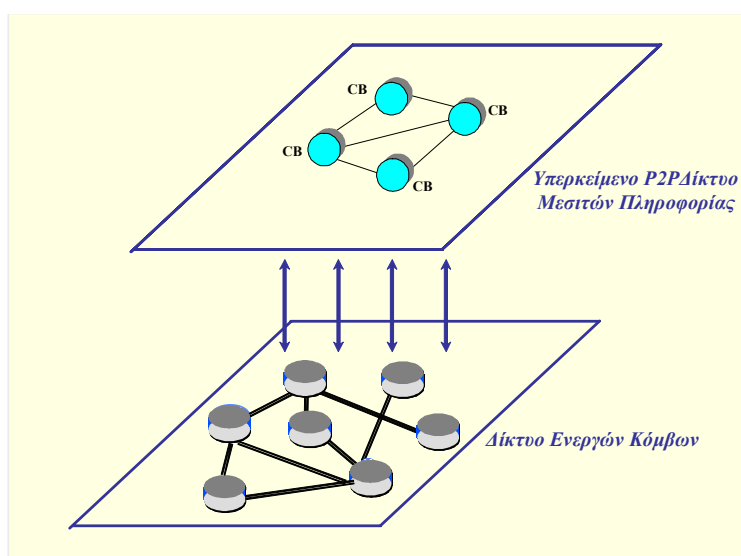


Σχήμα 3.10: Αρχιτεκτονική Ενεργού Κόμβου

Κάθε ενεργός κόμβος διαχωρίζεται αρχιτεκτονικά στη Μηχανή Προώθησης (*Forwarding Engine*) και την Ενεργό Μηχανή (*Active Engine*) (Σχήμα 3.10). Η Μηχανή Προώθησης είναι η μονάδα που προωθεί τα πακέτα στο στρώμα IP, όπως δρομολογητής, σημείο πρόσβασης WLAN, πύλη μέσων, κ.λπ.. Διαθέτει ένα φίλτρο (*filter*), το οποίο χρησιμοποιείται για τη σύλληψη των ενεργών πακέτων και την προώθησή τους στην Ενεργό Μηχανή, όπου εκτελούνται. Η Ενεργός Μηχανή υποδιαιρείται στους μεσίτες (*brokers*), οι οποίοι αποτελούν οντότητες επεξεργασίας που επιτρέπουν στις ενεργές εφαρμογές (*active applications*) να ανακτούν πληροφορίες και να εκτελούν λειτουργίες στο τοπικό περιβάλλον μέσω καθορισμένων με σαφήνεια διεπαφών.

3.6.2 ΕΝΕΡΓΑ ΔΙΚΤΥΑ ΚΑΙ ΟΜΟΤΙΜΟΙ ΜΕΣΙΤΕΣ

Σε ένα υψηλότερο επίπεδο αφαίρεσης, υπάρχει ο Μεσίτης Πληροφορίας (Context Broker - CB), ο οποίος επικοινωνεί με άλλους ομότιμους μεσίτες σχηματίζοντας υπερκείμενο δίκτυο ομότιμων κόμβων (overlay P2P network), όπως έχει περιγραφεί στο κεφάλαιο 2. Απώτερος σκοπός κάθε Μεσίτη Πληροφορίας είναι να παράσχει οποιαδήποτε πληροφορία στις υπερκείμενες υπηρεσίες μέσω των κατάλληλων διεπαφών. Οι διεπαφές αυτές προσφέρουν στις υπηρεσίες τη δυνατότητα ανάκτησης πληροφοριών που αφορούν σε χρήστες, κατάσταση δικτύου ή γενικότερες συνθήκες περιβάλλοντος. Αυτές οι πληροφορίες ανακτώνται από το υποκείμενο δίκτυο, τους χρήστες, εξωτερικά συστήματα ή από τις ενεργές δυνατότητες της πλατφόρμας Ενεργών Δικτύων (Σχήμα 3.11).



Σχήμα 3.11: Ομότιμοι Μεσίτες Πληροφορίας πάνω από Ενεργά Δίκτυα

Η βελτίωση της επίδοσης του συστήματος αξιοποιώντας την τεχνολογία των Ενεργών Δικτύων μπορεί να πραγματοποιηθεί τόσο στο πλαίσιο ανάπτυξης της υπηρεσίας στην υποδομή του δικτύου (deployment framework) όσο και στο πλαίσιο λειτουργίας της (execution framework). Το πλαίσιο ανάπτυξης σχετίζεται με τον τρόπο που οι συνιστώσες της υπηρεσίας εγκαθίστανται κατά απαίτηση στους κόμβους του δικτύου, ενώ το πλαίσιο λειτουργίας περιγράφει τον τρόπο που οι συνιστώσες εκτελούνται και συνεργάζονται μεταξύ τους ώστε να παρέχουν ένα αξιόπιστο και αποδοτικό δίκτυο. Η συγχώνευση και διανομή της πληροφορίας μέσω ειδικών εφαρμογών που αναπτύσσονται δυναμικά μέσα στο δίκτυο αποτελεί αντικείμενο του πρώτου πλαισίου, ενώ οι τεχνικές αναζήτησης πληροφορίας μεταξύ των μεσιτών είναι αντικείμενο του δεύτερου πλαισίου.

3.6.2.1 Συγχώνευση και Διανομή Πληροφορίας

Οι ενσωματωμένες δυνατότητες των Ενεργών Δικτύων για μετακίνηση του ενεργού κώδικα και ομοιόμορφη επικοινωνία παρέχουν τις προϋποθέσεις για συγχώνευση και διανομή πληροφορίας με αποδοτικό και αποτελεσματικό τρόπο. Αυτό μπορεί να επιτευχθεί επιτρέποντας στους υπολογισμούς να εκτελούνται κοντά στις πηγές πληροφορίας που χρησιμοποιούν. Ειδικότερα, η συγχώνευση πληροφορίας μέσα στο δίκτυο προκαλεί μείωση των απαιτήσεων σε εύρος ζώνης των χρηστών, οι οποίοι βρίσκονται στη χαμηλού εύρους ζώνης «περιφέρεια» του δικτύου. Αυτό συμβαίνει γιατί μειώνεται ο αριθμός των απαραίτητων συναλλαγών μέσω του δικτύου, αξιοποιώντας τις δυνατότητες για τοπική επεξεργασία.

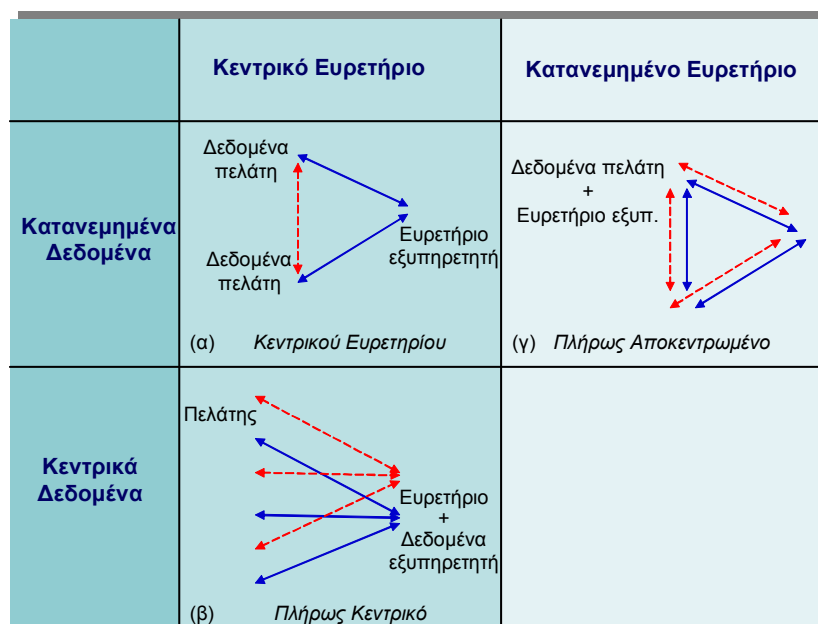
Συνεπώς, η παραγωγή σύνθετης πληροφορίας πραγματοποιείται από ενεργές εφαρμογές (active applications), οι οποίες συλλέγουν πρωτογενείς πληροφορίες από διαφορετικές πηγές και τις επεξεργάζονται κατάλληλα ώστε να προκύψει πιο σύνθετη πληροφορία. Οι ενεργές αυτές εφαρμογές αναπτύσσονται δυναμικά και κατά απαίτηση (on-demand deployment) στο περιβάλλον εκτέλεσης των ενεργών κόμβων, σε στρατηγικά σημεία μέσα στο δίκτυο, τα οποία επιλέγονται με γνώμονα συγκεκριμένα κριτήρια επίδοσης, όπως το εύρος ζώνης που καταναλώνεται ή η χρονική απόκριση στα επικείμενα αιτήματα για πληροφορία. Κάθε τέτοια εφαρμογή αποτελεί μια νέα πηγή πληροφορίας και καταχωρείται στον τοπικό μεσίτη (CB), ο οποίος αναλαμβάνει τη διανομή της πληροφορίας της.

3.6.2.2 Αναζήτηση Πληροφορίας

Κάθε υπηρεσία διαβιβάζει οποιοδήποτε αίτημα για πληροφορία στον τοπικό CB. Οι CBs είναι κατανεμημένοι στους ενεργούς κόμβους και συνεργάζεται ο ένας με τον άλλο προκειμένου να βρεθεί η ζητούμενη πληροφορία. Ωστόσο, σε ένα κατανεμημένο σύστημα μεσιτών, η αναζήτηση της πληροφορίας αποτελεί συχνά το πιο χρονοβόρο στάδιο της επεξεργασίας πληροφορίας. Για το λόγο αυτό κρίθηκε σκόπιμη η μελέτη ενός προβλήματος αναζήτησης πληροφορίας σε σύστημα ομότιμων μεσιτών, αξιοποιώντας τις δυνατότητες που προσφέρουν τα Ενεργά Δίκτυα με τον κινητό κώδικα.

Κάνοντας μια σύντομη αναφορά στους γνωστούς μηχανισμούς αναζήτησης, σημειώνουμε τα εξής: Οι παραδοσιακές μηχανές αναζήτησης Ιστού (π.χ. η μηχανή Yahoo) βασίζονται σε αυτόματα προγράμματα λογισμικού, τους *σαρωτές δικτύου* (*Web crawlers*), οι οποίοι εξερευνούν συστηματικά το διαδίκτυο. Οι πληροφορίες που προκύπτουν αποθηκεύονται σε κεντρική, γιγαντιαία βάση δεδομένων, η οποία αποτελεί το ευρετήριο της μηχανής αναζήτησης. Ωστόσο, το ευρετήριο στην περίπτωση αυτή περιλαμβάνει πληροφορίες μόνο από εξυπηρετητές περιεχομένου που λειτουργούν δημόσια, ενώ επιπλέον δεν υποβάλλεται σε άμεση ανανέωση όταν αυτό απαιτείται, π.χ. όταν κάποιος εξυπηρετητής αποσυνδεθεί. Το σχετικό μοντέλο αναζήτησης πληροφορίας χαρακτηρίζεται ως *πλήρως*

κεντρικό, αφού τόσο τα δεδομένα όσο και το ευρετήριο είναι κεντρικά αποθηκευμένα. Σε μια αρχική απόπειρα να ξεπεραστούν τα προβλήματα που αυτό παρουσιάζει, προέκυψε το μοντέλο **κεντρικού ευρετηρίου**, το οποίο έγινε γνωστό από το σύστημα μερισμού αρχείων Napster [53]. Σύμφωνα με αυτό το μοντέλο, μόνο το ευρετήριο είναι κεντρικά αποθηκευμένο, ενώ τα δεδομένα παραμένουν καταναμημένα στους κόμβους. Το γεγονός όμως ότι το ευρετήριο συγκεντρώνει πληροφορία θέσης για όλες τις πηγές μπορεί να προκαλέσει προβλήματα επεκτασιμότητας στο σύστημα. Ο εξυπηρετητής ενδέχεται να υπερφορτωθεί όταν καλείται να εξυπηρετήσει πολλά αιτήματα, και επίσης για μεγάλο αριθμό εγγραφών θα πρέπει να διαθέτει μεγάλο αποθηκευτικό χώρο. Αντίθετα, το **πλήρως αποκεντρωμένο** μοντέλο, μέσω της P2P αναζήτησης, αποφεύγει την εξάρτηση από κεντρικά σημεία, επιτρέποντας τόσο στα δεδομένα όσο και στο ευρετήριο να είναι καταναμημένα στους κόμβους. Βασίζεται στην προώθηση μηνυμάτων αναζήτησης μεταξύ των κόμβων μέχρι να εντοπιστεί η επιθυμητή πληροφορία. Αυτό όμως μπορεί να οδηγήσει σε μεγάλη σπατάλη εύρους ζώνης αλλά και καθυστέρηση απόκρισης εάν δεν χρησιμοποιηθούν κατάλληλοι αλγόριθμοι δρομολόγησης. Το Σχήμα 3.12 συγκρίνει τα τρία αυτά μοντέλα αναζήτησης. Σημειώνεται ότι οι συμπαγείς γραμμές συμβολίζουν μεταφορά δεδομένων, ενώ οι διακεκομμένες μεταφορά μηνυμάτων αναζήτησης.



Σχήμα 3.12: Σύγκριση Μοντέλων Αναζήτησης Πληροφορίας

Η P2P αναζήτηση θα μπορούσε να επωφεληθεί από τα Ενεργά Δίκτυα ως εξής: Μπορούμε εύκολα να σκεφτούμε ένα ενεργό πακέτο να μεταφέρει μια P2P ερώτηση για τον εντοπισμό συγκεκριμένης πληροφορίας. Επειδή οι P2P ερωτήσεις μπορούν να είναι πιο «ελαφριές» από τις ερωτήσεις των καταναμημένων βάσεων δεδομένων, παρακάμπτοντας ζητήματα διαλειτουργικότητας, ο αντίστοιχος κινητός κώδικας θέτει ένα ελάχιστο

υπολογιστικό φορτίο και επιβάλει ελάχιστη επιβάρυνση στο δίκτυο. Επιπλέον, ένα ενεργό πακέτο θα μπορούσε να φέρει και τον κώδικα για τον καθορισμό των μοντέλων ζήτησης πληροφορίας σε κάθε κόμβο, συνεισφέροντας στη βελτίωση της επίδοσης αναζήτησης [54]. Στα πλαίσια τόσο *προδραστικών* (*proactive*) όσο και *αντιδραστικών* (*reactive*) μηχανισμών αναζήτησης, οι δυνατότητες αυτές αξιοποιούνται από μια σειρά αλγορίθμων δρομολόγησης που αναπτύσσονται στο κεφάλαιο 5.

3.7 ΕΠΙΛΟΓΟΣ

Στο κεφάλαιο αυτό παρουσιάστηκε η χρήση του υποδείγματος κινητού κώδικα, μέσα από τη μελέτη σχετικών μοντέλων, τεχνολογιών και εφαρμογών. Ειδικότερα, προβλήθηκε ο ρόλος εναλλακτικών υποδειγμάτων κινητού κώδικα έναντι του μοντέλου απομακρυσμένης επικοινωνίας, και διερευνήθηκαν οι περιπτώσεις που ο κινητός κώδικας αποδεικνύεται πιο αποδοτικός. Στη συνέχεια, μελετήθηκε η χρήση κινητού κώδικα προς την κατεύθυνση ενός γενικευμένου μοντέλου υπολογισμού βασισμένου στο δίκτυο. Σε ένα τέτοιο μοντέλο, οι έννοιες του πελάτη και του εξυπηρετητή αποδυναμώνονται και εμφανίζεται ένα σύστημα ομότιμων συνιστωσών, όπου όλοι οι υπολογιστικοί τόποι είναι ισοδύναμοι, διαδραματίζοντας δυναμικά το ρόλο αιτούντος κώδικα ή προμηθευτή κώδικα. Η υπολογιστική με Κινητούς Πράκτορες και με Ενεργά Δίκτυα παρουσιάστηκαν ως δυνατές προσεγγίσεις του μοντέλου αυτού. Η παρουσίαση ορισμένων από τους τομείς εφαρμογών της τεχνολογίας Κινητών Πρακτόρων και αυτής των Ενεργών Δικτύων αποδεικνύει την πρακτική τους χρήση. Τέλος, παρουσιάστηκε η αξιοποίηση του υποδείγματος κινητού κώδικα στα ειδικότερα προβλήματα μελέτης της διατριβής που εισήχθησαν στο κεφάλαιο 2 και θα αναπτυχθούν λεπτομερώς στα κεφάλαια 4 και 5.

3.8 ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ

- [1] G.H. Forman and J. Zahorjan, “The Challenges of Mobile Computing”, IEEE Computer, Vol.27, No.4, pp.38-47, 1994.
- [2] T. Imielinsky and B.R. Badrinath, “Wireless Computing: Challenges in Data Management,” Communication of the ACM, Vol.37, No.10, pp.18-28, 1994.
- [3] Object Management Group, “The Common Object Request Broker: Architecture and Specification”, Revision 2.5, September 2001.
- [4] A. Carzaniga, G.P. Picco and G. Vigna, “Designing Distributed Applications with Mobile Code Paradigms”, Proceedings of 19th International Conference on Software Engineering (ICSE’97), Boston, Ma., pp.22-32, May 1997.
- [5] C. Ghezzi and G. Vigna, “Mobile Code Paradigms and Technologies: A Case Study”, Proceedings of 1st International Workshop on Mobile Agents (MA’97), Berlin, Germany, April 1997.
- [6] M. Baldi, S. Gai and G.P. Picco, “Exploiting Code Mobility in Decentralized and Flexible Network Management”, Proceedings of 1st International Workshop on Mobile Agents (MA’97), Berlin, Germany, pp.13-26, April 1997.
- [7] M.A. Hamilton, “Java and the Shift to Net-Centric Computing”, IEEE Computer, Vol.29, No.8, pp.31-39, August 1996.
- [8] www.java.sun.com
- [9] A. Fuggetta, G.P. Picco and G. Vigna, “Understanding Code Mobility”, IEEE Transactions on Software Engineering, Vol.24, No.5, pp.342-361, May 1998.
- [10] A.D. Birrell and B.J. Nelson, “Implementing Remote Procedure Calls”, ACM Transactions on Computer Systems, Vol.2, No.1, pp.39-59, February 1984.
- [11] R. Englander, “Java and SOAP”, Publisher: O’Reilly, ISBN: 0-596-00175-4, May 2002.
- [12] W. Grosso, “Java RMI”, Publisher: O’Reilly, ISBN: 1-56592-452-5, October 2001.
- [13] J.W. Stamos and D.K. Gifford, “Remote Evaluation”, ACM Transactions on Programming Languages and Systems, Vol.12, No.4, pp.537-565, October 1990.
- [14] A Puliafito, S. Riccobene, M. Scarpa, “An Analytical Comparison of the Client-Server, Remote Evaluation and Mobile Agents Paradigms”, Proceedings of the 1st Joint Symposium on Agent Systems and Applications (ASA '99), October 1999.
- [15] J. Zukowski, Chapter 18 of “Java AWT Reference”, Publisher: O’Reilly, ISBN: 1-56592-240-9, March 1997.

- [16] S. Papastavrou, G. Samaras and E. Pitoura, "Mobile Agents for World Wide Web Distributed Database Access", IEEE Transactions on Knowledge and Data Engineering, Vol.12, No.5, pp.802-820, 2000.
- [17] R. Gray, G. Cybenko, D. Kotz, and D. Rus, "Mobile Agents: Motivations and State of the Art", Handbook of Agent Technology, Publisher: AAAI/MIT Press, 2002.
- [18] C.G. Harrison, D.M. Chess, and A. Kershenbaum, "Mobile Agents: Are they a good idea?", Technical Report, IBM Research Division, T.J. Watson Research Center, 1995.
- [19] S. Feit, "SNMP: A Guide to Network Management", Publisher: McGraw-Hill, ISBN: 0070203598, September 1994.
- [20] D. Gavalas, D. Greenwood, M. Ghanbari, M. O'Mahony, "An Infrastructure for Distributed and Dynamic Network Management based on Mobile Agent Technology", Proceedings of the IEEE International Conference on Communications (ICC'99), pp.1362-1366, 1999.
- [21] M.K. Kona and C. Xu, "A Framework for Network Management Using Mobile Agents", International Parallel and Distributed Processing Symposium (IPDPS'2002), Florida, April 15 - 19, 2002,
- [22] M.G. Rubinstein, O. Carlos and M.B. Duarte, "Evaluating Tradeoffs of Mobile Agents in Network Management", Networking and Information Systems Journal, Vol.2, No.2, pp.237-252, 1999.
- [23] A. Bieszczad, B. Pagurek and T. White, "Mobile Agents for Network Management", IEEE Communications Surveys, Vol.1, No.1, pp.2-9, Fourth Quarter 1998.
- [24] D. Gavalas, D. Greenwood, M. Ghanbari, M. O'Mahony, "Complimentary Polling Modes for Network Performance Management Employing Mobile Agents", Proceedings of the IEEE Global Communications Conference, Rio de Janeiro, Brazil, pp.401-405, December 1999.
- [25] K. Psounis, "Active Networks: Applications, Security, Safety, and Architectures", IEEE Communications Surveys, Vol.2, No.1, pp.2-16, First Quarter 1999.
- [26] D.B. Lange, M. Oshima, "Seven Good Reasons for Mobile Agents", Communications of the ACM, Vol.42, No.3, pp. 88-89, 1999.
- [27] A. Kaltabani, I. Sygkouna, M. Chantzara, M. Strimpakou, P. Demestichas, M. Anagnostou, "Strategies for Deploying Static and Mobile Components in Future Distributed Processing Environments", Proceedings of the 8th International Conference on Advances in Communications and Control (ComCon8 2001), Crete, Greece, June 2001.
- [28] A. Kaltabani, M. Strimpakou, I. Sygkouna, M. Chantzara, M. Anagnostou, "Agent Mobility in Open Telecommunication Environments", Proceedings of the 1st

- International Conference on Evolutionary Computing for Industrial Applications (Eurogen 2001), Athens, Greece, September 2001.
- [29] R.H. Glitho, E. Olougouna, S. Pierre, "Mobile Agents and their Use for Information Retrieval: A Brief Overview and an Elaborate Case Study", IEEE Network Magazine, Vol.16, No.1, pp.34-41, January/February 2002.
- [30] <http://www.trl.ibm.com/aglets>
- [31] <http://www.objectspace.com/voyager>
- [32] <http://www.grasshopper.de>
- [33] D.L. Tennenhouse, J.M. Smith, W.D. Sincoskie, D.J. Wetherall, G.J. Minden, "A Survey of Active Network Research", IEEE Communications Magazine, Vol.35, No.1, pp.80-86, January 1997.
- [34] K.L. Calvert, S. Bhattacharjee and E. Zegura, "Directions in Active Networks", IEEE Communications Magazine, Vol.36, No.10, pp. 72-78, October 1998.
- [35] S. Karnouskos, "Security Implications of Implementing Active Network Infrastructures Using Agent Technology", Computer Networks, Vol.36, No.1, pp.87-100, 2001.
- [36] B. Schwartz, W. Zhou, A.W. Jackson, W.T. Strayer, D. Rockwell, and C. Partridge, "Smart Packets for Active Networks", Proceedings of the IEEE Operarch'99, pp.90-97, March 1999.
- [37] D.J. Wetherall and D.L. Tennenhouse, "The ACTIVE_IP Option", Proceedings of the 7th ACM SIGOPS European Workshop, September 1996.
- [38] A. Banchs et al., "Multicasting Multimedia Streams with Active Networks", ICSI Technical Report 97-050.
- [39] S. Ortiz, "Active Networks: The Programmable Pipeline", IEEE Computer Magazine, Vol.31, No.8, pp.19-21, August 1998.
- [40] S. Bhattacharjee, K.L. Calvert, and E.W. Zegura, "An Architecture for Active Networking", Proceedings of High Performance Networking (HPN'97), White Plains, NY, pp.265-279, April 1997.
- [41] D. Decasper and B. Plattner, "DAN: Distributed Code Caching for Active Networks", Proceedings of the IEEE INFOCOM '98, San Francisco, CA, April 1998.
- [42] D.J. Wetherall, J.V. Guttag, and D.L. Tennenhouse, "ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols", Proceedings of the IEEE OPENARCH '98, pp.117-129, April 1998.
- [43] C.A. Gunter, S.M. Nettles, and J.M. Smith, "The SwitchWare Active Network Architecture", IEEE Network Magazine, Special Issue on Active and Programmable Networks, Vol.12, No.3, pp.29-36, May/June 1998.

- [44] Y. Yemini and Sushil da Silva, "Towards Programmable Networks", Proceedings of the IFIP/IEEE International Workshop on Distributed Systems, Operations, and Management, L'Aquila, Italy, October 1996.
- [45] S. Vrontis, I. Sygkouyna, M. Chantzara, E. Sykas, "Enabling Distributed QoS Management Utilizing Active Network Technology", Network Control and Engineering for QoS, Security and Mobility II, Publisher: Kluwer Academic Publishers, ISBN: 1-4020-7616-9, pp.139-151, September 2003.
- [46] U. Legedza, D.J. Wetherall, and J. Gutttag, "Improving the Performance of Distributed Applications Using Active Networks", Proceedings of the IEEE Infocom '98, San Francisco, CA, pp.590-599, April 1998.
- [47] S. Bhattacharjee, K.L. Calvert, and E.W. Zegura, "Self-organizing Wide-Area Network Caches", Proceedings of the IEEE Infocom '98, San Francisco, CA, pp.600-608, April 1998.
- [48] I. Sygkouyna, M. Strimpakou, F. Valera, A. Kaltabani, L. Bellido, E. Vázquez, M. Anagnostou, "Seamless Incorporation of Agents in an E-Commerce Intermediation Platform", Mobile Agents for Telecommunication Applications, Publisher: Springer-Verlag, ISBN: 3-540-00021-6, LNCS 2521, pp.292-301, October 2002.
- [49] F. Valera, A. Kaltabani, E. Vazquez, L. Bellido, I. Sygkouyna, M.Strimpakou, M. Anagnostou, "Experiences with Middleware and Mobile Agents in an E-Commerce European Project", International Journal of Internet and Enterprise Management, Vol.2, No.2, pp.163-177, Spring 2004.
- [50] M. Papazoglou, "Agent-oriented Technology in Support of E-Business", Communications of the ACM, Vol.44, No.4, pp.71-77, April 2001.
- [51] I. Sygkouyna, S. Vrontis, M. Chantzara, M. Anagnostou, E. Sykas, "Context-Aware Services Provisioning on Top of Active Technologies", Mobile Agents for Telecommunication Applications, Publisher: Springer-Verlag, ISBN: 3-540-20298-6, LNCS 2881, pp.67-76, October 2003.
- [52] ABLE: The Active Bell Labs Engine, Homepage URL: <http://www.cs.bell-labs.com/who/ABLE/>
- [53] <http://www.napster.com>
- [54] M. Parameswaran, A.Susarla, A.B. Whinston, "P2P Networking: An Information-Sharing Alternative", IEEE Computer, Vol.34 No.7, pp.31-38, July 2001.

ΚΕΦΑΛΑΙΟ 4

***Σχεδιασμός Δρομολογίων για Συλλογή
Πληροφορίας με Χρήση Κινητών Πρακτόρων***

ΚΕΦΑΛΑΙΟ 4: ΣΧΕΔΙΑΣΜΟΣ ΔΡΟΜΟΛΟΓΙΩΝ ΓΙΑ ΣΥΛΛΟΓΗ ΠΛΗΡΟΦΟΡΙΑΣ ΜΕ ΧΡΗΣΗ ΚΙΝΗΤΩΝ ΠΡΑΚΤΩΡΩΝ

4.1 ΕΙΣΑΓΩΓΗ

Η ανάκτηση πληροφορίας αποτελεί έναν σημαντικό τομέα εφαρμογών που θα μπορούσε να ωφεληθεί από τη χρήση του υποδείγματος κινητών πρακτόρων [1]. Για παράδειγμα, μια εμπορική συναλλαγή ηλεκτρονικού εμπορίου απαιτεί συχνά πρόσβαση, σε πραγματικό χρόνο, σε απομακρυσμένες πηγές πληροφοριών, όπως καταλόγους προϊόντων με σκοπό την ανάκτηση μεγάλων ποσοτήτων πληροφοριών για λογαριασμό ενός μεσίτη, ο οποίος είναι υπεύθυνος να εξυπηρετεί έγκαιρα τα αιτήματα των πελατών [2]. Σε αυτήν την περίπτωση, ένας κινητός πράκτορας μπορεί να μετακινηθεί από τον κόμβο του μεσίτη στον απομακρυσμένο κόμβο που φέρει μια πηγή πληροφορίας. Οι διαδικασίες φιλτραρίσματος που εκτελούνται από τον πράκτορα τοπικά στην πηγή αναμένεται να μειώσουν την πληροφορία που τελικά επιστρέφεται στο μεσίτη. Προκειμένου να επιταχυνθεί η συλλογή κατανεμημένης πληροφορίας, ο μεσίτης θα μπορούσε να αποστείλει ταυτόχρονα έναν αριθμό ταυτόσημων πρακτόρων, καθένας από τους οποίους επισκέπτεται ένα υποσύνολο των κόμβων που φιλοξενούν τη ζητούμενη πληροφορία και επιστρέφει στο μεσίτη με τα αποτελέσματα της αναζήτησής του. Σε ένα σωστά κατανεμημένο σύστημα οι πράκτορες μπορούν να είναι ανεξάρτητοι ο ένας από τον άλλο και θα πρέπει να είναι σε θέση να επιλέξουν την καλύτερη δυνατή στρατηγική μετακίνησής τους για μια δεδομένη αποστολή και την τρέχουσα κατάσταση του δικτύου.

Ο καθορισμός του ελάχιστου αριθμού πρακτόρων και των δρομολογίων τους που εξασφαλίζουν συλλογή της κατανεμημένης πληροφορίας στον ελάχιστο χρόνο αποτελεί ένα σύνθετο πρόβλημα βελτιστοποίησης [3], το οποίο υπάγεται στην περιοχή έρευνας που ασχολείται με τον *προγραμματισμό της δράσης κινητών πρακτόρων (Mobile Agent planning)* [4, 5, 6]. Μια προφανής παρατήρηση είναι ότι μια αύξηση του αριθμού παράλληλων πρακτόρων μπορεί να μειώσει το χρόνο ολοκλήρωσης της αποστολής, δηλαδή το χρόνο που απαιτείται από τον πράκτορα που ολοκληρώνει τελευταίος την αποστολή του. Από την άλλη μεριά, οι κινητοί πράκτορες έχουν κάποιο κόστος δημιουργίας και επικοινωνίας και καταναλώνουν εύρος ζώνης κατά τη μετακίνησή τους μέσα στο δίκτυο. Κατά συνέπεια, η χρήση περισσότερων πρακτόρων μπορεί να επιβαρύνει την απόδοση του συστήματος. Είναι γεγονός ότι ακόμα και αν καμιά ειδική μέριμνα για την απόδοση δεν απαιτείται, ο μέγιστος αριθμός πρακτόρων, ο οποίος ισούται με τον αριθμό κόμβων που πρέπει να εξυπηρετηθούν, θα μπορούσε να μειωθεί σε τέτοιο βαθμό ώστε ο χρόνος ολοκλήρωσης της αποστολής να διατηρηθεί στο ελάχιστο, υπό τον όρο ότι έχει προηγουμένως πραγματοποιηθεί προσεκτικός

σχεδιασμός των δρομολογίων των πρακτόρων. Εάν όμως απαιτούνται περαιτέρω εγγυήσεις απόδοσης, η επιθυμητή ισορροπία μεταξύ της απόδοσης του συστήματος και του χρόνου ολοκλήρωσης της αποστολής αποτελεί αντικείμενο διαπραγματεύσεως μεταξύ του συστήματος και του χρήστη. Μέσα στο πλαίσιο αυτό, προσπαθούμε αρχικά να βρούμε τα δρομολόγια που εκτελούνται από ένα δεδομένο αριθμό πρακτόρων έτσι ώστε ο χρόνος ολοκλήρωσης της αποστολής συλλογής της πληροφορίας να ελαχιστοποιηθεί. Προκειμένου να καταστεί το σύστημα πιο ευέλικτο, επιδιώκουμε στη συνέχεια να καθορίσουμε το βέλτιστο αριθμό πρακτόρων που πρέπει να χρησιμοποιηθεί, δηλαδή τον ελάχιστο αριθμό που εξασφαλίζει την επιθυμητή ισορροπία μεταξύ του χρόνου ολοκλήρωσης της αποστολής και των πόρων που καταναλώνει το σύστημα.

Στην παρούσα μελέτη προσεγγίζουμε τη λύση του προβλήματος αρχικά με δύο γνωστούς ευριστικούς αλγορίθμους, τον *Savings* και τον *Insertion*, οι οποίοι προέρχονται από την περιοχή έρευνας που ασχολείται με το πρόβλημα της δρομολόγησης οχημάτων, αλλά έχουν προσαρμοστεί κατάλληλα στις απαιτήσεις του προβλήματος που μελετάμε. Αφετέρου, εισάγουμε έναν καινούριο ευριστικό, τον *Serial Reverse-Constructed Route (SR-CR)*, τον όποιο σχεδιάσαμε ειδικά για το πρόβλημα των πρακτόρων. Τα αποτελέσματα της προσομοίωσης έδειξαν ότι ο SR-CR σε συνδυασμό με τον *Savings* παράγουν λύσεις που πλησιάζουν τη βέλτιστη σε λογικό χρόνο. Με επιλεκτική παραμετρική μεταβολή του αριθμού των πρακτόρων, μπορεί επιπλέον να καθοριστεί ο αριθμός των πρακτόρων που εξασφαλίζει την απαραίτητη ισορροπία μεταξύ του χρόνου ολοκλήρωσης της αποστολής και των πόρων που δαπανώνται από τη χρήση των πρακτόρων.

Η δομή του κεφαλαίου είναι η εξής: Αρχικά παρουσιάζονται οι σχετικές με την παρούσα εργασίες. Στη συνέχεια δίνεται η γενική περιγραφή αλλά και η μαθηματική διατύπωση του προβλήματος που μελετάμε. Στα πλαίσια της επίλυσής του, περιγράφονται οι προτεινόμενοι αλγόριθμοι, αναλύεται η χρονική τους πολυπλοκότητα και παρουσιάζονται δυο χαρακτηριστικά παραδείγματα εφαρμογής τους. Επιπλέον, παρουσιάζονται αποτελέσματα που προέκυψαν από δοκιμές με τη βοήθεια προσομοίωσης, τα οποία αφορούν τόσο στην αποδοτικότητα και αποτελεσματικότητα των αλγορίθμων όσο και στη μελέτη της επίδρασης συγκεκριμένων παραμέτρων του προβλήματος στη λύση του. Τέλος, προτείνεται μια μέθοδος προσδιορισμού του βέλτιστου αριθμού πρακτόρων που πρέπει να χρησιμοποιηθούν. Το κεφάλαιο ολοκληρώνεται με την ανάλυση των βασικών παρατηρήσεων και συμπερασμάτων που προέκυψαν από τη μελέτη του προβλήματος.

4.2 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Μετά από σχετική βιβλιογραφική έρευνα προέκυψε ότι ένας αριθμός ερευνητών έχει ασχοληθεί με τεχνικές σχεδιασμού της δράσης πρακτόρων για ανάκτηση κατανεμημένης

πληροφορίας. Το *Πρόβλημα του Περιοδεύοντος Πωλητή (Traveling Salesman Problem - TSP)* [7] είναι το πιο γνωστό πρόβλημα αυτής της κατηγορίας. Μια κοντινή παραλλαγή αυτού, το *Πρόβλημα του Περιοδεύοντος Πράκτορα (Travelling Agent Problem - TAP)* [8, 1], προσπαθεί να βρει την καλύτερη σειρά επίσκεψης των κόμβων από έναν πράκτορα, έτσι ώστε ο αναμενόμενος χρόνος ολοκλήρωσης μιας δεδομένης αποστολής να ελαχιστοποιείται, υπό τον όρο ότι η αποστολή μπορεί να ολοκληρωθεί σε οποιοδήποτε κόμβο, βάσει γνωστής πιθανότητας. Μόλις η αποστολή ολοκληρωθεί, ο πράκτορας επιστρέφει στον κόμβο από τον οποίο ξεκίνησε.

Το *Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem - VRP)* [7] είναι πιθανώς το πλησιέστερο στο πρόβλημα πρακτόρων που μελετάμε. Η ιδέα στο κλασικό VRP είναι η συλλογή φορτίου από πελάτες, δεδομένου ότι υπάρχει ένας κόμβος-σταθμός, από τον οποίο αποστέλλεται ένας αριθμός οχημάτων περιορισμένης χωρητικότητας προκειμένου να συλλέξουν το φορτίο όλων των πελατών. Αν υποθέσουμε ότι κάθε πελάτης πρέπει να εξυπηρετηθεί ακριβώς από ένα όχημα, σκοπός είναι να εξυπηρετηθούν όλοι οι πελάτες με ελάχιστο κόστος, δηλαδή να βρεθεί ένα δρομολόγιο για κάθε όχημα έτσι ώστε το άθροισμα από τα κόστη όλων των ακμών που διασχίζονται να είναι ελάχιστο. Διάφορες επεκτάσεις και πιο σύνθετες παραλλαγές της κλασικής έκδοσης έχουν επίσης μελετηθεί, συμπεριλαμβανομένων μεταξύ άλλων της ύπαρξης χρονικών παραθύρων, πολλαπλών κόμβων-σταθμών, ή περιορισμών προτεραιότητας στην εξυπηρέτηση των πελατών [9, 10]. Μια βασική διαφορά μεταξύ του προβλήματος πρακτόρων που μελετάμε και του κλασικού VRP είναι ότι στο πρώτο, ο χρόνος για τη μετάβαση του πράκτορα από έναν κόμβο σε έναν άλλο δεν είναι σταθερός, αλλά εξαρτάται από το συσσωρευμένο όγκο στοιχείων που ο πράκτορας έχει ανακτήσει μέχρι τη στιγμή της μετάδοσής του. Αντιθέτως, στο κλασικό VRP το κόστος μιας σύνδεσης, το οποίο αντιπροσωπεύει συνήθως απόσταση, είναι σταθερό. Επιπλέον, το υπό μελέτη πρόβλημα λύνεται για ένα γενικό γράφο, ενώ το VRP για έναν πλήρη. Αξίζει επίσης να σημειωθεί ότι στην περίπτωσή μας η τριγωνική ανισότητα δεν ισχύει, και ένας πράκτορας μπορεί να επισκεφτεί έναν κόμβο περισσότερες της μιας φορές [11, 12]. Ωστόσο, η αλγοριθμική προσέγγιση της λύσης του VRP μας παρείχε πολύτιμες ενδείξεις για το πρόβλημα των πρακτόρων και για το λόγο αυτό παραθέτουμε τα αποτελέσματα της βιβλιογραφικής έρευνας που αφορά στη λύση του VRP στο Παράρτημα.

Η εργασία του Baek [13] αναφέρεται στο ίδιο πρόβλημα σχεδιασμού αλλά από μια ελαφρώς διαφορετική σκοπιά. Θεωρώντας τον αριθμό πρακτόρων και το συνολικό χρόνο εκτέλεσης (άθροισμα των χρόνων όλων των πρακτόρων) ως τους δυο βασικούς παράγοντες επίδοσης του συστήματος, προσπαθεί να τους ελαχιστοποιήσει διατηρώντας στο ελάχιστο το χρόνο ολοκλήρωσης της αποστολής. Εντούτοις, προκειμένου ο συγγραφέας να μειώσει την πολυπλοκότητα του προβλήματος, κάνει την υπόθεση ότι τα μεγέθη των κινητών πρακτόρων

παραμένουν σταθερά και αγνοεί το χρόνο μετάδοσής τους, ο οποίος είναι ένας σημαντικός παράγοντας καθυστέρησης στα προβλήματα αυτά.

4.3 ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΔΙΑΤΥΠΩΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

Μια υψηλού επιπέδου περιγραφή του προβλήματος προσδιορισμού των βέλτιστων δρομολογίων των πρακτόρων είναι η ακόλουθη: Στα πλαίσια ενός συστήματος πρακτόρων θεωρούμε ότι ένας πελάτης αποστέλλει ταυτόχρονα έναν αριθμό ταυτόσημων πρακτόρων, οι οποίοι αναλαμβάνουν να ανακτήσουν δεδομένα από συγκεκριμένους κόμβους του δικτύου. Κάθε πράκτορας επισκέπτεται ένα υποσύνολο των κόμβων και στη συνέχεια επιστρέφει στον πελάτη, ολοκληρώνοντας την αποστολή του. Σκοπός του προβλήματος είναι να προσδιοριστούν τα δρομολόγια των πρακτόρων, έτσι ώστε ο συνολικός χρόνος ολοκλήρωσης της αποστολής να ελαχιστοποιηθεί. Υποθέτουμε ότι τα δεδομένα ενός κόμβου ανακτώνται από έναν και μόνο πράκτορα. Επίσης, εάν ένας κόμβος προσπελάζεται από τον ίδιο πράκτορα περισσότερες από μια φορές, οικονομικοί λόγοι υπαγορεύουν ανάκτηση των δεδομένων του κατά την τελευταία προσπέλαση.

Υποθέτουμε ότι οι καθυστερήσεις προέρχονται αποκλειστικά από τη μετάδοση των πρακτόρων στους κόμβους που επισκέπτονται. Σημειώνουμε επίσης ότι στην τελική λύση δύο ή περισσότερα δρομολόγια μπορεί να έχουν κοινό μονοπάτι, το οποίο ξεκινάει από τον κόμβο του κεντρικού αποστολέα των πρακτόρων. Στην περίπτωση αυτή, έχουμε θεωρήσει αμελητέα την επιπλέον καθυστέρηση που οφείλεται στη σειριακή μετάδοση των πρακτόρων. Μια προηγμένη υλοποίηση θα μπορούσε να θεωρήσει ότι ένας και μόνο πράκτορας διασχίζει το κοινό μονοπάτι, ο οποίος όταν φτάσει στον τελευταίο κόμβο του κοινού μονοπατιού δημιουργεί έναν αριθμό αντιγράφων του, ίσο με τον αριθμό των διαφορετικών διαδρομών που ξεκινούν από αυτόν τον κόμβο. Τέλος, πληροφορίες δικτύου όπως το εύρος ζώνης των συνδέσεων, είναι γνωστές από στατιστικές μετρήσεις, και επίσης θεωρούμε ότι ίσος όγκος δεδομένων ανακτάται από κάθε κόμβο του δικτύου.

Το πρόβλημα ορίζεται σε μη-κατευθυνόμενο γράφο $G(V, E)$, όπου $V = \{0, 1, \dots, N\}$ είναι το σύνολο των κόμβων και E το σύνολο των ακμών του γράφου. Επίσης συμβολίζουμε με $V' \subseteq V - \{0\}$ το υποσύνολο των κόμβων που φέρουν την αιτούμενη από τον πελάτη πληροφορία. Χωρίς βλάβη της γενικότητας θεωρούμε ότι ο κόμβος 0 αντιπροσωπεύει τον κόμβο από τον οποίο αποστέλλονται οι πράκτορες. Επίσης χρησιμοποιούμε τους ακόλουθους συμβολισμούς: L παριστάνει το αρχικό μέγεθος πράκτορα, l το μέγεθος της πληροφορίας που ανακτάται από έναν κόμβο, τ_{ij} το ρυθμό μετάδοσης της γραμμής $(i, j) \in E$, και K τον αριθμό των πρακτόρων που αποστέλλονται. Ορίζουμε επίσης τη βοηθητική μεταβλητή L_{ik} , η οποία αντιπροσωπεύει το συσσωρευμένο μέγεθος του πράκτορα k μετά από την επίσκεψη του κόμβου $i \in V$.

4.3.1 ΔΙΑΤΥΠΩΣΗ ΜΕ ΑΚΕΡΑΙΟΥΣ ΠΕΡΙΟΡΙΣΜΟΥΣ (INTEGER-CONSTRAINED FORMULATION)

Για τις ανάγκες μιας τυπικής μαθηματικής διατύπωσης του προβλήματος, επιχειρούμε μια διατύπωση ακέραιων περιορισμών. Για το σκοπό αυτό ορίζουμε τις ακόλουθες μεταβλητές:

$$z_{ik} = \begin{cases} 1, & \text{αν τα δεδομένα του κόμβου } i \in V' \text{ ανακτώνται από τον πράκτορα } k \\ 0, & \text{διαφορετικά} \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & \text{αν ο πράκτορας } k \text{ διασχίζει την } (i,j) \in E \text{ από τον } i \text{ προς τον } j \\ 0, & \text{διαφορετικά} \end{cases}$$

Ο σκοπός του προβλήματος για ελαχιστοποίηση του χρόνου που απαιτεί ο πράκτορας που τελειώνει τελευταίος την αποστολή του, υπό τους περιορισμούς που αναφέρθηκαν, περιγράφεται από τις παρακάτω εξισώσεις:

Ελαχιστοποίησε:

$$\max_k \sum_{i=0}^N \sum_{j=0}^N \frac{1}{\tau_{ij}} \cdot L_{ik} \cdot x_{ijk}, \quad k = 1, \dots, K \quad (1)$$

$$L_{ik} = \begin{cases} L, & \text{αν } i = 0 \\ L_{hk} + l \cdot z_{ik}, & \text{αν } i \in V' \text{ και } x_{hik} = 1 \\ L_{hk}, & \text{αν } i \notin V' \text{ και } x_{hik} = 1 \end{cases} \quad (2)$$

Περιορισμοί:

$$\sum_{k=1}^K z_{ik} = 1, \quad \forall i \in V' \quad (3)$$

$$z_{ik} = 0 \text{ ή } 1, \quad \forall i \in V' \\ k = 1, \dots, K \quad (4)$$

$$\left. \begin{aligned} \sum_{i=1}^N x_{ijk} &= 1, & j &= 0 & (5) \\ \sum_{j=1}^N x_{ijk} &= 1, & i &= 0 & (6) \\ \sum_{i=0}^N x_{ijk} &= \sum_{i=0}^N x_{jik}, & j &\in V - \{0\} & (7) \\ \sum_{ij \in S \times S} x_{ijk} &\leq |S| - 1, & S &\subseteq \{1, \dots, N\} \\ & & & 2 \leq |S| \leq N - 1 & (8) \\ x_{ijk} &= 0 \text{ ή } 1, & \forall (i, j) &\in E & (9) \end{aligned} \right\} k = 1, \dots, K$$

Οι περιορισμοί (3)-(4) διασφαλίζουν ότι η πληροφορία ενός κόμβου ανακτάται ακριβώς από έναν πράκτορα, ενώ οι (5)-(6) δηλώνουν ότι σε κάθε δρομολόγιο υπάρχει μια εισερχόμενη και μια εξερχόμενη ακμή στον κόμβο 0, δεδομένου ότι κάθε δρομολόγιο αρχίζει και τελειώνει στον 0. Για κάθε άλλο κόμβο, ο αριθμός των εισερχόμενων ακμών είναι ίσος με τον αριθμό των εξερχόμενων (περιορισμός (7)). Σημειώνεται επίσης ότι, όπως και στη διατύπωση του TSP, οι περιορισμοί (5)-(7) δεν αποκλείουν τη δημιουργία δυο ασύνδετων κύκλων για έναν πράκτορα και συνεπώς απαιτείται ένας επιπλέον περιορισμός που να το απαγορεύει. Αυτός είναι ο (8), ο οποίος περιγράφει το εξής: για κάθε πράκτορα k , ο υπογράφος με σύνολο κόμβων $V_k = \{i, j \in V \mid x_{ijk} = 1\}$ και σύνολο ακμών $E_k = \{(i, j) \in E \mid x_{ijk} = 1\}$ θα πρέπει να είναι συνδεδεμένος. Τελικά, οι τιμές των μεταβλητών x_{ijk}, z_{ik} μαζί με το L_{ik} προσδιορίζουν τη λύση του προβλήματος.

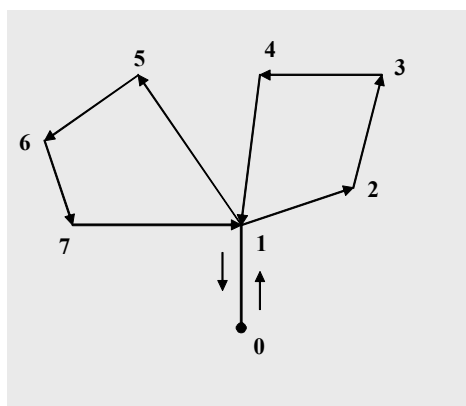
Ωστόσο, στην περίπτωση που το άθροισμα των εισερχόμενων και εξερχόμενων ακμών σε έναν κόμβο που περιλαμβάνεται σε κάποιο δρομολόγιο είναι μεγαλύτερο του 2, τότε η παραπάνω διατύπωση του L_{ik} (εξίσωση (2)) παρουσιάζει ασάφεια, αφού δεν το ορίζει μονοσήμαντα. Για παράδειγμα, εάν υποθέσουμε ότι στο Σχήμα 4.1 για δεδομένο πράκτορα k ισχύει:

$$x_{01k} = x_{10k} = x_{12k} = x_{23k} = x_{34k} = x_{41k} = x_{15k} = x_{56k} = x_{67k} = x_{71k} = 1$$

και

$$z_{3k} = z_{4k} = z_{6k} = z_{7k} = 1$$

τότε δεν είναι σαφές εάν το δρομολόγιό του θα είναι το $\langle 0, 1, 2, 3, 4, 1, 5, 6, 7, 1, 0 \rangle$ ή το $\langle 0, 1, 5, 6, 7, 1, 2, 3, 4, 1, 0 \rangle$ και συνεπώς δεν είναι σαφές εάν το L_{3k} , για παράδειγμα, θα είναι ίσο με l ή $3l$, αν υποθέσουμε ότι το μέγεθος της πληροφορίας που ανακτάται από ένα κόμβο είναι ίσο με l . Για το λόγο αυτό, καταφεύγουμε στη γραφοθεωρητική διατύπωση του προβλήματος.



Σχήμα 4.1: Δρομολόγιο ενός πράκτορα

4.3.2 ΓΡΑΦΟΘΕΩΡΗΤΙΚΗ ΔΙΑΤΥΠΩΣΗ (GRAPH-THEORETIC FORMULATION)

Για τους σκοπούς μιας γραφοθεωρητικής διατύπωσης του προβλήματος θεωρούμε ότι κάθε ακμή $(i, j) \in E$ που διασχίζεται από έναν πράκτορα k χαρακτηρίζεται από καθυστέρηση μετάδοσης του πράκτορα $t_{ijk}(L_{ik})$, η οποία μεταβάλλεται με το L_{ik} σύμφωνα

με $t_{ijk}(L_{ik}) = \frac{1}{\tau_{ij}} \cdot L_{ik}$, με το L_{ik} να αυξάνει με τον αριθμό των κόμβων που ο πράκτορας έχει

εξυπηρετήσει (δηλαδή από τους οποίους έχει ανακτήσει πληροφορία) μέχρι τη στιγμή της μετάδοσής του. Η λύση του προβλήματος απαιτεί τον προσδιορισμό των κόμβων που θα εξυπηρετηθούν από κάθε πράκτορα k και της σειράς επίσκεψης των κόμβων αυτών, δηλαδή του μονοπατιού $\langle n_k(1), n_k(2), \dots, n_k(N_k) \rangle$, όπου N_k ο συνολικός αριθμός κόμβων από τους οποίους θα περάσει ο πράκτορας k , συμπεριλαμβανομένων αυτών από τους οποίους έχει ξαναπεράσει. Ο χρόνος που απαιτείται για την ανάκτηση της πληροφορίας σε αυτό το μονοπάτι είναι:

$$T_k = \sum_{i=1}^{N_k-1} t_{n_k(i), n_k(i+1), k}(L_{n_k(i), k})$$

$$L_{n_k(i), k} = \begin{cases} L, & \text{αν } i = 1 \\ L_{n_k(i-1), k} + l, & \text{αν } n_k(i) \in V' \text{ εξυπηρετείται} \\ L_{n_k(i-1), k}, & \text{διαφορετικά} \end{cases}$$

Σκοπός του προβλήματος είναι να προσδιοριστούν K μονοπάτια έτσι ώστε το $\max_{k=1, \dots, K} T_k$ να ελαχιστοποιηθεί, δεδομένου ότι κάθε μονοπάτι αρχίζει και τελειώνει στον κόμβο 0 (δηλαδή $n_k(1) \equiv n_k(N_k) \equiv 0$), και το φορτίο ενός κόμβου ανακτάται από έναν πράκτορα μονομιάς. Σημειώνεται επίσης ότι δεν απαιτείται ένα μονοπάτι να είναι απλό, ούτε τα K μονοπάτια να είναι μεταξύ τους ασύνδετα.

4.4 ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ

Το υπό μελέτη πρόβλημα είναι *NP-hard* δεδομένου ότι ανάγεται στο *TSP* για $k = 1$ και $l = 0$, δηλαδή όταν χρησιμοποιείται ένας πράκτορας, ο οποίος επισκέπτεται τους κόμβους χωρίς να ανακτά οποιαδήποτε πληροφορία. Για το λόγο αυτό εφαρμόσαμε τρεις ευριστικούς αλγορίθμους [14]. Οι δυο από αυτούς, ο *Savings* και ο *Insertion*, οι οποίοι προέρχονται από την περιοχή έρευνας *VRP*, κατασκευάζουν βαθμιαία μια δυνατή λύση παρακολουθώντας ταυτόχρονα το συνολικό κόστος της λύσης. Ο τρίτος, δηλαδή ο *Serial Reverse-Constructed Route*, σχεδιάστηκε ειδικά για το υπό μελέτη πρόβλημα και κατασκευάζει τα δρομολόγια σειριακά, ενώ το καθένα από αυτά δομείται από το τέλος προς την αρχή. Λεπτομερής περιγραφή των αλγορίθμων δίνεται στις επόμενες ενότητες.

Σημειώνεται ότι σε καθέναν από αυτούς τους αλγορίθμους, το ελάχιστο σε χρόνο μονοπάτι από έναν κόμβο σε οποιοδήποτε άλλο, υπό τον όρο ότι δεν εκτελείται ανάκτηση πληροφορίας στο ενδιάμεσο, υπολογίζεται με τη βοήθεια του αλγορίθμου *Dijkstra Shortest Path* [7], υποθέτοντας ότι το βάρος κάθε ακμής $(i, j) \in E$ ισούται με $1/\tau_{ij}$.

4.4.1 ΑΛΓΟΡΙΘΜΟΣ “SAVINGS”

Ο εν λόγω αλγόριθμος βασίζεται σε μια παραλλαγή του κλασικού αλγορίθμου *Savings* των *Clarke and Wright* [15], ο οποίος σχεδιάστηκε ειδικά για το *VRP*. Ενώ η κλασική έκδοση του αλγορίθμου βασίζεται στην έννοια της πραγματικής αποταμίευσης απόστασης που προκαλείται από τη συγχώνευση δύο δρομολογίων, στην περίπτωση μας το κριτήριο είναι το μέγεθος κατά το οποίο αυξάνει ο συνολικός χρόνος συλλογής της πληροφορίας εξαιτίας μιας δεδομένης συγχώνευσης δυο δρομολογίων. Επιπλέον, θεωρούμε ότι όταν ένα δρομολόγιο A συγχωνεύεται με ένα δρομολόγιο B , το νέο δρομολόγιο σχηματίζεται συνδέοντας τον τελευταίο κόμβο του δρομολογίου A από τον οποίο ανακτάται πληροφορία, με τον πρώτο κόμβο του δρομολογίου B από τον οποίο επίσης ανακτάται πληροφορία, μέσω του ελαχίστου σε χρόνο μονοπατιού. Ο αλγόριθμος λειτουργεί ως εξής:

Βήμα 1 (Αρχικοποίηση Δρομολογίων): Για κάθε κόμβο $i \in V'$, βρείτε το συντομότερο από πλευράς χρόνου δρομολόγιο, το οποίο ξεκινά από το 0, τελειώνει στο 0 και περιλαμβάνει ανάκτηση πληροφορίας από τον κόμβο i .

Βήμα 2 (Συγχώνευση Δρομολογίων): Κάθε δρομολόγιο συγχωνεύεται με κάθε άλλο. Από τα δρομολόγια που προκύπτουν, επιλέξτε αυτό με το μικρότερο χρόνο, και απομακρύνετε τα δυο δρομολόγια από τα οποία αυτό προήλθε.

Το βήμα 2 επαναλαμβάνεται έως ότου ο αριθμός των δρομολογίων που έχουν απομείνει γίνει ίσος με K . Το ακόλουθο διάγραμμα (Σχήμα 4.2) απεικονίζει τη λειτουργία του αλγορίθμου.



Σχήμα 4.2: Αλγόριθμος Savings

4.4.2 ΑΛΓΟΡΙΘΜΟΣ “INSERTION”

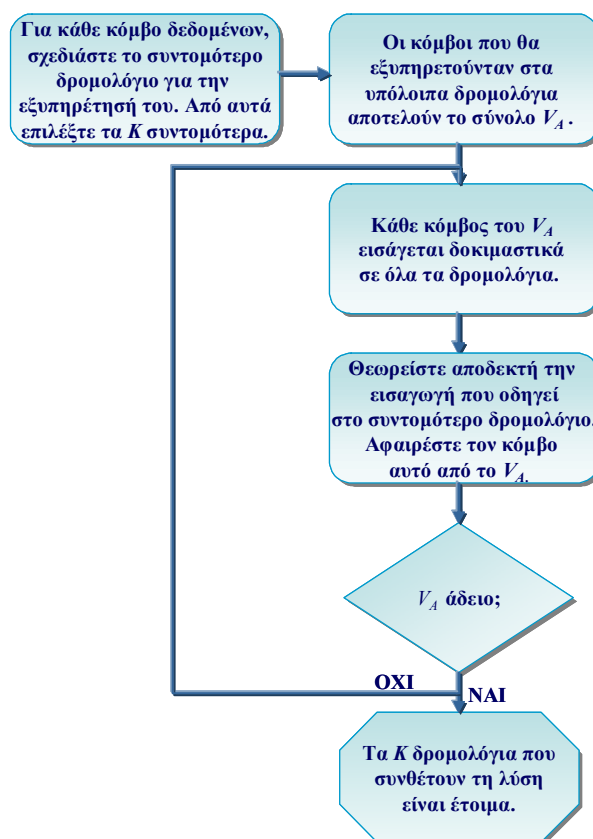
Η βασική ιδέα του αλγορίθμου *Insertion* είναι η βαθμιαία ανάθεση κόμβων με δεδομένα σε υπάρχοντα δρομολόγια. Ενώ στον αλγόριθμο *Insertion* που έχει προταθεί για το *VRP* [16] το κριτήριο για την εισαγωγή ενός κόμβου σε ένα δρομολόγιο είναι το κόστος εισαγωγής του, στην περίπτωσή μας κριτήριο είναι το μέγεθος κατά το οποίο αυξάνει ο συνολικός χρόνος συλλογής της πληροφορίας εξαιτίας της δεδομένης εισαγωγής. Για κάθε ακμή (i, j) ενός δρομολογίου R , όπου ο i ή ο j εξυπηρετείται μέσα στο R , η εισαγωγή ενός κόμβου $m \in V'$ στο R είναι δυνατή κατά τους ακόλουθους τρόπους: αν ο i εξυπηρετείται και ο m είναι γειτονικός του, το μονοπάτι που προκύπτει από τη δεδομένη ακμή μετά την εισαγωγή είναι το $\langle i, m, i, j \rangle$. Κατά τρόπο ανάλογο, αν ο j εξυπηρετείται και ο m είναι γειτονικός του, το προκύπτων μονοπάτι είναι το $\langle i, j, m, j \rangle$. Σε καθεμιά από αυτές τις περιπτώσεις, αν ο m δεν είναι γειτονικός σε κάποιον από τους i, j , η εισαγωγή εξακολουθεί να θεωρείται σύμφωνα με τα αντίστοιχα ελάχιστα μονοπάτια. Επιπλέον, για κάθε ζευγάρι κόμβων που εξυπηρετούνται διαδοχικά στο R (ενδέχεται να μεσολαβούν κόμβοι που απλώς προσπελάζονται), η εισαγωγή του m θεωρείται επίσης μεταξύ αυτών, σύμφωνα με τα

αντίστοιχα ελάχιστα μονοπάτια, ως $i \xrightarrow{p_1} m \xrightarrow{p_2} j$, όπου τα p_1 και p_2 συμβολίζουν τα ελάχιστα μονοπάτια από το i στο m και από το m στο j , αντίστοιχα. Η ίδια περίπτωση θεωρείται επίσης μεταξύ του κόμβου 0 και του πρώτου ή του τελευταίου κόμβου που εξυπηρετείται στο R . Ο αλγόριθμος έχει ως εξής:

Βήμα 1 (Αρχικοποίηση Δρομολογίων): Για κάθε κόμβο $i \in V'$, βρείτε το συντομότερο σε χρόνο δρομολόγιο, το οποίο ξεκινά από το 0, τελειώνει στο 0 και περιλαμβάνει ανάκτηση πληροφορίας από τον κόμβο i . Κατατάξτε τα δρομολόγια κατά αύξουσα σειρά των χρόνων τους και επιλέξτε τα K πρώτα. Οι κόμβοι που θα εξυπηρετούνταν στα δρομολόγια που δεν έχουν επιλεγεί αποτελούν το σύνολο των υποψηφίων για εισαγωγή κόμβων V_A ($V_A \subseteq V'$).

Βήμα 2 (Εισαγωγή Κόμβων): Κάθε κόμβος $i \in V_A$ τοποθετείται δοκιμαστικά σε κάθε δυνατή θέση από κάθε δρομολόγιο, όπως περιγράφηκε νωρίτερα, και θεωρείται αποδεκτή η εισαγωγή εκείνου του κόμβου που οδηγεί στο συντομότερο, από πλευράς χρόνου, δρομολόγιο. Ο κόμβος που εισάγεται αφαιρείται από το V_A .

Το βήμα 2 επαναλαμβάνεται μέχρι να εξαντληθούν τα στοιχεία του V_A . Τα βήματα του αλγορίθμου παριστάνονται στο επόμενο διάγραμμα (Σχήμα 4.3).



Σχήμα 4.3: Αλγόριθμος Insertion

4.4.3 ΑΛΓΟΡΙΘΜΟΣ “SERIAL REVERSE-CONSTRUCTED ROUTE (SR-CR)”

Ο εν λόγω αλγόριθμος αναθέτει παρόμοιο αριθμό κόμβων με δεδομένα στα δρομολόγια που κατασκευάζει, εκμεταλλευόμενος την υπόθεση ότι ίσος όγκος πληροφορίας ανακτάται από κάθε κόμβο. Κάθε δρομολόγιο δομείται με τη χρήση ενός βοηθητικού ευριστικού, του *Reverse-Construction*, ο οποίος βασίζεται στην εξής ιδέα: όσο αυξάνει το φορτίο που μεταφέρει ο πράκτορας, τόσο γρηγορότερες συνδέσεις θα πρέπει να διασχίζει. Με άλλα λόγια, όσο αυξάνει ο αριθμός των κόμβων που έχει εξυπηρετήσει ο πράκτορας, τόσο πιο αποδοτικό είναι να επιλέγει συνδέσεις υψηλότερου ρυθμού μετάδοσης. Εάν υποθέσουμε ότι $V_R \subseteq V'$, $n \leq |V_R|$, και $f \in V_R$, οι βοηθητικοί ευριστικοί και ο γενικός αλγόριθμος λειτουργούν ως εξής:

Reverse-Construction (V_R, n, f)

Ξεκινώντας τη δόμηση ενός δρομολογίου από πίσω προς τα εμπρός, η δόμηση ξεκινά από τον τελευταίο κόμβο (κόμβος 0) και συνεχίζει προσθέτοντας ένα κόμβο $i \in V_R$ κάθε φορά. Ο τελευταίος κόμβος από τον οποίο ανακτάται πληροφορία είναι ο κόμβος f , οπότε ο f συνδέεται με τον κόμβο 0 μέσω ελαχίστου μονοπατιού. Ο προτελευταίος κόμβος από τον οποίο ανακτάται πληροφορία είναι εκείνος ο κόμβος από το V_R , ο οποίος δεν έχει επιλεγεί ακόμη και συνδέεται με το f μέσω του συντομότερου μονοπατιού. Η διαδικασία εισαγωγής κόμβων συνεχίζει με τον ίδιο τρόπο και σταματάει όταν θα έχουν συνολικά εισαχθεί n κόμβοι από το V_R , συμπεριλαμβανομένου του f . Προφανώς, το δρομολόγιο «κλείνει» στον κόμβο 0 μέσω ελαχίστου μονοπατιού από τον κόμβο που εισάγεται τελευταίος.

Reverse-Construction (V_R, n)

Αυτός αποτελεί παραλλαγή του προηγούμενου. Διαφέρει μόνο στο γεγονός ότι ο κόμβος που εξυπηρετείται τελευταίος δεν είναι προκαθορισμένος.

Ο αλγόριθμος SR-CR λειτουργεί συνολικά ως εξής:

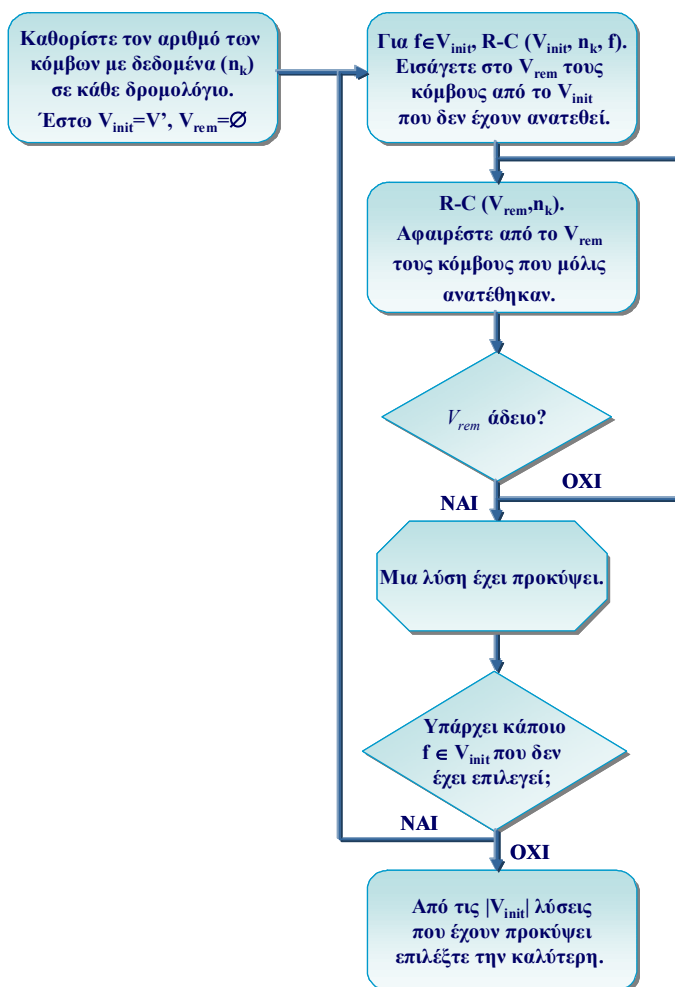
Βήμα 1 (Εξισορρόπηση Φορτίου): Θα δημιουργηθούν διαδοχικά K δρομολόγια με τον ίδιο αριθμό κόμβων με δεδομένα προς εξυπηρέτηση. Εάν αυτό δεν είναι δυνατό, τότε δυο οποιαδήποτε δρομολόγια θα διαφέρουν το πολύ κατά ένα κόμβο, ενώ τα δρομολόγια με τους περισσότερους κόμβους θα δομηθούν πρώτα. Για ευκολία θα θεωρήσουμε ότι σε κάθε δρομολόγιο εξυπηρετούνται n_k κόμβοι με δεδομένα.

Βήμα 2 (Διαδοχική Κατασκευή Δρομολογίων): Υποθέστε ότι $V_{init} = V'$ και επιλέξτε $f \in V_{init}$.

Βήμα 2.1: Εφαρμόστε τον *Reverse-Construction* (V_{init}, n_k, f), οπότε προκύπτει το πρώτο δρομολόγιο. Συμβολίζουμε με V_{rem} ($V_{rem} \subseteq V_{init}$) το σύνολο των κόμβων από το V_{init} , οι οποίοι δεν έχουν ακόμη ανατεθεί σε κάποιο δρομολόγιο.

Βήμα 2.2: Εφαρμόστε τον *Reverse-Construction* (V_{rem}, n_k) διαδοχικά $(K-1)$ φορές. Μετά από την κάθε εκτέλεση, το V_{rem} ανανεώνεται, απομακρύνοντας από αυτό τους κόμβους που εισήχθησαν στο δρομολόγιο που μόλις κατασκευάστηκε. Με αυτό τον τρόπο, τα επόμενα $(K-1)$ δρομολόγια δημιουργούνται διαδοχικά.

Το βήμα 2 επαναλαμβάνεται για κάθε $f \in V_{init}$. Με τον τρόπο αυτό προκύπτουν συνολικά $|V_{init}|$ λύσεις. Η βέλτιστη λύση είναι προφανώς αυτή με το μικρότερο χρόνο. Η λειτουργία του αλγορίθμου απεικονίζεται στο επόμενο διάγραμμα (Σχήμα 4.4).



Σχήμα 4.4: Αλγόριθμος SR-CR

4.5 ΠΟΛΥΠΛΟΚΟΤΗΤΑ ΑΛΓΟΡΙΘΜΩΝ

Ο χρόνος εκτέλεσης των αλγορίθμων σε δεδομένο γράφο $G(V, E)$ θα υπολογιστεί μετρώντας το μέγεθος της εισόδου σε σχέση με τον αριθμό των κόμβων $|V|$ και ακμών $|E|$ του γράφου (για συντομία θα συμβολίζονται με V και E , αντίστοιχα), τον αριθμό των κόμβων που περιέχουν δεδομένα N_D (όπου $N_D = |V'|$) και τον αριθμό των κινητών πρακτόρων K . Ο γράφος

$G(V, E)$ αναπαριστάται ως μια συλλογή από λίστες γειτνίασης (*adjacency-list representation*). Πιο συγκεκριμένα, χρησιμοποιείται ένας πίνακας Adj από $|V|$ λίστες, μία λίστα για κάθε κόμβο. Για κάθε $u \in V$, η λίστα $Adj[u]$ περιλαμβάνει δείκτες, ένα σε κάθε κόμβο v για τον οποίο υπάρχει ακμή $(u, v) \in E$ [17].

Ο κοινός όρος του χρόνου εκτέλεσης των τριών αλγορίθμων προέρχεται από το πρόβλημα εύρεσης των ελαχίστων μονοπατιών όλων των ζευγών κόμβων (*all-pairs shortest paths problem*), το οποίο επιλύσαμε με χρήση του αλγορίθμου *Dijkstra*. Ο *Dijkstra* τυπικά επιλύει το πρόβλημα εύρεσης των ελαχίστων μονοπατιών από μια πηγή (*single-source shortest path problem*). Στην παρούσα περίπτωση εφαρμόστηκε θεωρώντας σαν πηγή κάθε κόμβο του γράφου χωριστά. Δεδομένου ότι για μη-κατευθυνόμενο γράφο ο συνολικός αριθμός των ακμών στην αναπαράσταση με λίστες γειτνίασης είναι $2E$, ο χρόνος εκτέλεσης του *Dijkstra* είναι $O(V^2+2E)$ [17]. Συνεπώς, ο συνολικός χρόνος, δηλαδή για όλα τα δυνατά ζεύγη, είναι $T_{SP} = O(V^3 + 2EV)$.

4.5.1 ΑΛΓΟΡΙΘΜΟΣ “SAVINGS”

Κατά τη διάρκεια του βήματος 1, κάθε λειτουργία για τη κατασκευή ενός στοιχειώδους δρομολογίου απαιτεί σταθερό χρόνο $O(1)$. Επομένως, για αριθμό τέτοιων λειτουργιών ίσο με N_D απαιτείται συνολικός χρόνος $T_{init} = O(N_D)$.

Κατά τη διάρκεια του βήματος 2 πραγματοποιούνται λειτουργίες συγχώνευσης. Κάθε λειτουργία συγχώνευσης απαιτεί σταθερό χρόνο $O(1)$, δεδομένου ότι τα ελάχιστα μονοπάτια έχουν ήδη υπολογιστεί. Αρχικά, κάθε στοιχειώδες δρομολόγιο συγχωνεύεται με κάθε άλλο, για ένα σύνολο $2 \binom{N_D}{2}$ τέτοιων λειτουργιών. Ο παράγοντας 2 οφείλεται στο γεγονός ότι κάθε συγχώνευση πραγματοποιείται και προς τις δυο κατευθύνσεις. Επιλέγοντας το δρομολόγιο ελαχίστου χρόνου, αυτό θα συγχωνευτεί με καθένα από τα άλλα δρομολόγια κατά τη δεύτερη επανάληψη του βήματος, για ένα σύνολο $2(N_D - 2)$ λειτουργιών. Σημειώνεται ότι οι επαναλήψεις ίδιων συγχωνεύσεων αποφεύγονται. Για όλες τις επαναλήψεις του βήματος 2, ο συνολικός αριθμός των λειτουργιών συγχώνευσης είναι λοιπόν:

$$\begin{aligned} N_{\text{merge}} &= 2 \binom{N_D}{2} + 2(N_D - 2) + 2(N_D - 3) + \dots + 2K \\ &= N_D(N_D - 1) + 2 \sum_{i=K}^{N_D-2} i \\ &= 2N_D^2 - 4N_D - K^2 + K + 2 \end{aligned}$$

Ο αντίστοιχος χρόνος εκτέλεσης είναι: $T_{\text{merge}} = O(2N_D^2 - 4N_D - K^2 + K + 2)$

Η διαδικασία *EXTRACT-MIN*, η οποία εντοπίζει το στοιχείο με την ελάχιστη τιμή σε ένα σύνολο στοιχείων πλήθους N απαιτεί χρόνο εκτέλεσης $O(N)$ [17]. Κατά τη διάρκεια κάθε επανάληψης του βήματος 2, η *EXTRACT-MIN* εφαρμόζεται στο εκάστοτε σύνολο δρομολογίων που έχει προκύψει. Κατά την πρώτη επανάληψη, υπάρχουν $2^{\binom{N_D}{2}}$ δρομολόγια, κατά τη δεύτερη $2^{\binom{N_D-1}{2}}$, και προχωρώντας με τον ίδιο τρόπο, στην τελευταία επανάληψη απομένουν $2^{\binom{K+1}{2}}$ δρομολόγια. Επομένως, η συνολική συνεισφορά της διαδικασίας *EXTRACT-MIN* στο συνολικό χρόνο εκτέλεσης του αλγορίθμου είναι:

$$\begin{aligned} T_{min} &= O\left(2^{\binom{N_D}{2}}\right) + O\left(2^{\binom{N_D-1}{2}}\right) + \dots + O\left(2^{\binom{K+1}{2}}\right) \\ &= O\left(-\frac{1}{3}(K - N_D)(-1 + K^2 + KN_D + N_D^2)\right) \end{aligned}$$

Η συνάρτηση πολυπλοκότητας διατυπώνεται τελικά ως εξής:

$$\begin{aligned} T_{savings} &= T_{SP} + T_{init} + T_{merge} + T_{min} \\ &= O(V^3 + 2EV) + O\left(\frac{1}{3}N_D^3 + 2N_D^2 - \frac{10}{3}N_D - \frac{1}{3}K^3 - K^2 + \frac{4}{3}K + 2\right) \\ &= O\left(V^3 + 2EV + \frac{1}{3}N_D^3 - \frac{1}{3}K^3\right) \end{aligned}$$

4.5.2 ΑΛΓΟΡΙΘΜΟΣ “INSERTION”

Κατά τη διάρκεια του βήματος 1, η κύρια συνεισφορά στο χρόνο εκτέλεσης προέρχεται από τον βοηθητικό αλγόριθμο *Insertion Sort* [17], ο οποίος χρησιμοποιείται για την ταξινόμηση κατά αύξουσα σειρά ενός συνόλου στοιχείων. Για την ταξινόμηση των N_D διαδρομών αυτός απαιτεί χρόνο $T_{init} = O(N_D^2)$.

Κατά τη διάρκεια του βήματος 2 πραγματοποιούνται λειτουργίες εισαγωγής κόμβων. Η κατασκευή ενός νέου δρομολογίου από την εισαγωγή ενός κόμβου σε συγκεκριμένη θέση ενός δρομολογίου απαιτεί σταθερό χρόνο $O(1)$. Εάν υποθέσουμε ότι n_d κόμβοι δεδομένων εξυπηρετούνται σε δεδομένο δρομολόγιο, τότε ο μέγιστος δυνατός αριθμός δρομολογίων που προκύπτουν από την εισαγωγή ενός κόμβου στο δρομολόγιο αυτό είναι $(2n_d+1)$. Κατά την πρώτη επανάληψη του βήματος, $(N_D - K)$ κόμβοι εισάγονται στα K στοιχειώδη δρομολόγια δημιουργώντας $3K(N_D - K)$ δρομολόγια. Κατά τη δεύτερη επανάληψη, $(N_D - K - 1)$ κόμβοι δεδομένων εισάγονται στο νέο δρομολόγιο, το οποίο επιλέχτηκε κατά την πρώτη επανάληψη ως το συντομότερο. Δεδομένου ότι ο μέγιστος δυνατός αριθμός κόμβων που αυτό εξυπηρετεί είναι 2, ο μέγιστος αριθμός λειτουργιών εισαγωγής κατά τη δεύτερη επανάληψη

είναι $(N_D - K - 1)(2 \cdot 2 + 1)$. Σημειώνεται και πάλι ότι οι επαναλήψεις των ίδιων λειτουργιών εισαγωγής αποφεύγονται. Συνεχίζοντας με τον ίδιο τρόπο, προκύπτει ότι ο μέγιστος συνολικός αριθμός λειτουργιών εισαγωγής είναι:

$$\begin{aligned} N_{insert} &= 3K(N_D - K) + (N_D - K - 1)(2 \cdot 2 + 1) + (N_D - K - 2)(2 \cdot 3 + 1) + \dots + \\ &\quad + 1 \cdot [2(N_D - K) + 1] \\ &= 3K(N_D - K) + \sum_{i=1}^{N_D - K + 1} (N_D - K - i)(2i + 3) \\ &= \frac{1}{6}(N_D - K)(N_D - K - 1)(2N_D - 2K + 11) + 3K(N_D - K) \end{aligned}$$

Ο αντίστοιχος χρόνος εκτέλεσης είναι:

$$T_{insert} = O\left(\frac{1}{6}(N_D - K)(N_D - K - 1)(2N_D - 2K + 11) + 3K(N_D - K)\right)$$

Κατά τη διάρκεια κάθε επανάληψης του βήματος 2, η *EXTRACT-MIN* εφαρμόζεται στο εκάστοτε σύνολο δρομολογίων που έχει προκύψει. Στην πρώτη επανάληψη θα έχουν δημιουργηθεί $3K(N_D - K)$ δρομολόγια. Στη δεύτερη, το σύνολο των δρομολογίων θα έχει προκύψει από την εισαγωγή $(N_D - K - 1)$ κόμβων σε $[2(K + 1) + K]$ δυνατές θέσεις, στην τρίτη, από την εισαγωγή $(N_D - K - 2)$ κόμβων σε $[2(K + 2) + K]$ θέσεις και ούτω καθεξής, μέχρι την τελευταία, όπου τα διαθέσιμα δρομολόγια έχουν προκύψει από την εισαγωγή ενός κόμβου σε $[2(N_D - 1) + K]$ θέσεις. Συνεπώς, η συνολική συνεισφορά της διαδικασίας *EXTRACT-MIN* στο χρόνο εκτέλεσης του αλγορίθμου είναι:

$$\begin{aligned} T_{min} &= O(3K(N_D - K)) + O((N_D - K - 1)[2(K + 1) + K]) + \\ &\quad + O((N_D - K - 2)[2(K + 2) + K]) + \dots + O(2(N_D - 1) + K) \\ &= O\left(3K(N_D - K) + \sum_{i=1}^{N_D - K - 1} i[2(N_D - i) + K]\right) \\ &= O\left(3K(N_D - K) + \frac{1}{6}(N_D - K)(N_D - K - 1)(2N_D + 7K + 2)\right) \end{aligned}$$

Προσθέτοντας τους παραπάνω όρους, η συνάρτηση πολυπλοκότητας διαμορφώνεται ως εξής:

$$\begin{aligned}
T_{insertion} &= T_{SP} + T_{init} + T_{insert} + T_{min} \\
&= O(V^3 + 2EV) + \\
&+ O\left(\frac{2}{3}N_D^3 - K^2N_D - \frac{1}{2}KN_D^2 + \frac{5}{6}K^3 + \frac{5}{2}N_D^2 + \frac{3}{2}KN_D - 3K^2 - \frac{13}{6}N_D + \frac{13}{6}K\right) \\
&= O(V^3 + 2EV + \frac{2}{3}N_D^3 - K^2N_D - \frac{1}{2}KN_D^2 + \frac{5}{6}K^3)
\end{aligned}$$

4.5.3 ΑΛΓΟΡΙΘΜΟΣ “SR-CR”

Σε κάθε επανάληψη του βήματος 2, η κύρια λειτουργία που συνεισφέρει στο χρόνο εκτέλεσης είναι η *EXTRACT-MIN*. Αυτή εφαρμόζεται διαδοχικά σε πλήθος στοιχείων ίσο με $N_D - 1, N_D - 2, \dots, 1$, καταναλώνοντας χρόνο:

$$\begin{aligned}
T_{min} &= O(N_D - 1) + O(N_D - 2) + \dots + O(1) \\
&= O(N_D \frac{N_D - 1}{2}) \\
&= O(N_D^2)
\end{aligned}$$

Για σύνολο επαναλήψεων ίσο με N_D , ο συνολικός χρόνος που καταναλώνει η *EXTRACT-MIN* είναι $O(N_D^3)$. Επιπλέον, «κλείνοντας» ένα δρομολόγιο στον κόμβο 0 απαιτεί χρόνο $O(1)$. Για σύνολο τέτοιων λειτουργιών ίσο με KN_D , η συνεισφορά στο χρόνο εκτέλεσης είναι $O(KN_D)$.

Η συνάρτηση χρονικής πολυπλοκότητας γίνεται λοιπόν:

$$T_{SR-CR} = O(V^3 + 2EV + N_D^3 + KN_D)$$

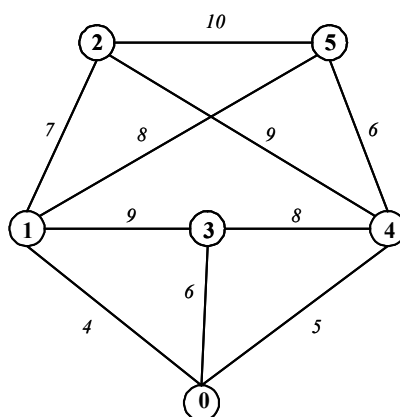
Δεδομένου ότι αυτό που τελικά ενδιαφέρει είναι ο τρόπος που η συνάρτηση πολυπλοκότητας μεταβάλλεται, από την παραπάνω ανάλυση προκύπτει ότι η πολυπλοκότητα και των τριών υπό εξέταση αλγορίθμων μεταβάλλεται *πολυωνυμικά* με το μέγεθος της εισόδου.

4.6 ΠΑΡΑΔΕΙΓΜΑΤΑ

Η ενότητα αυτή περιλαμβάνει δυο παραδείγματα. Η εφαρμογή καθενός από τους αλγορίθμους που παρουσιάστηκαν προηγουμένως περιγράφεται αναλυτικά με σκοπό να γίνει κατανοητή η λειτουργία τους και ταυτόχρονα να μελετηθούν οι συμπεριφορές τους και οι λύσεις που αυτοί παράγουν. Στο πρώτο παράδειγμα, η ζητούμενη πληροφορία βρίσκεται σε όλους τους κόμβους του δικτύου, ενώ στο δεύτερο σε ένα υποσύνολο αυτών.

4.6.1 ΠΑΡΑΔΕΙΓΜΑ 1: ΠΛΗΡΟΦΟΡΙΑ ΣΕ ΌΛΟΥΣ ΤΟΥΣ ΚΟΜΒΟΥΣ ΤΟΥ ΓΡΑΦΟΥ

Ο γράφος δικτύου στο υπό μελέτη παράδειγμα απεικονίζεται στο Σχήμα 4.5. Η επιγραφή μιας ακμής υποδηλώνει το ρυθμό μετάδοσης της αντίστοιχης σύνδεσης, μετρούμενη σε Mbps. Υποθέτουμε ότι ο κόμβος 0, ο οποίος αντιστοιχεί στον πελάτη, εξαπολύει δυο πράκτορες παράλληλα. Το αρχικό μέγεθος κάθε πράκτορα ισούται με 20 Kbytes, ενώ το μέγεθος της πληροφορίας που ανακτάται από έναν κόμβο ισούται με 5 Kbytes. Σκοπός είναι να προσδιοριστούν τα δρομολόγια των δυο πρακτόρων, έτσι ώστε ο χρόνος που απαιτείται για συλλογική ανάκτηση της πληροφορίας από τους κόμβους 1, 2, 3, 4 και 5 και παράδοσή της στον κόμβο 0 να ελαχιστοποιηθεί.



Σχήμα 4.5: Γράφος δικτύου παραδείγματος 1

4.6.1.1 Εφαρμογή του “Savings”

Το βήμα αρχικοποίησης έχει σαν αποτέλεσμα τη δημιουργία των δρομολογίων $\langle 0, \underline{1}, 0 \rangle$, $\langle 0, \underline{4}, \underline{2}, 4, 0 \rangle$, $\langle 0, \underline{3}, 0 \rangle$, $\langle 0, \underline{4}, 0 \rangle$ και $\langle 0, \underline{4}, \underline{5}, 4, 0 \rangle$, με αντίστοιχους χρόνους 90 ms, 112 ms, 60 ms, 72 ms, και 132 ms. Σημειώνεται ότι η υπογράμμιση του αριθμού ενός κόμβου συμβολίζει ανάκτηση πληροφορίας από τον κόμβο αυτό.

Στη συνέχεια, καθένα από τα παραπάνω δρομολόγια συγχωνεύεται με κάθε άλλο. Για παράδειγμα, η συγχώνευση του δρομολογίου $\langle 0, \underline{1}, 0 \rangle$ με το $\langle 0, \underline{4}, 0 \rangle$ παράγει το δρομολόγιο $\langle 0, \underline{1}, \underline{3}, \underline{4}, 0 \rangle$ με χρόνο 135 ms, ενώ η συγχώνευση του $\langle 0, \underline{4}, 0 \rangle$ με το $\langle 0, \underline{1}, 0 \rangle$ καταλήγει στο δρομολόγιο $\langle 0, \underline{4}, \underline{3}, \underline{1}, 0 \rangle$ με χρόνο 139 ms. Προφανώς, η κατεύθυνση κατά την οποία πραγματοποιείται ένα δρομολόγιο είναι σημαντική. Επιλέγοντας την πιο αποδοτική οικονομικά συγχώνευση, η οποία στην περίπτωση αυτή προέρχεται από τη σύζευξη του $\langle 0, \underline{4}, 0 \rangle$ με το $\langle 0, \underline{3}, 0 \rangle$, έχει σαν αποτέλεσμα τη διαμόρφωση των δρομολογίων $\langle 0, \underline{1}, 0 \rangle$, $\langle 0, \underline{4}, \underline{2}, 4, 0 \rangle$, $\langle 0, \underline{4}, \underline{3}, 0 \rangle$, και $\langle 0, \underline{4}, \underline{5}, 4, 0 \rangle$ με χρόνους 90 ms, 112 ms, 97 ms και 132 ms, αντίστοιχα.

Ομοίως, η δεύτερη επανάληψη συγχώνευσης έχει σαν αποτέλεσμα τα δρομολόγια $\langle 0, \underline{4}, \underline{2}, \underline{1}, 0 \rangle$, $\langle 0, \underline{4}, \underline{3}, 0 \rangle$, και $\langle 0, \underline{4}, \underline{5}, 4, 0 \rangle$ με χρόνους 138 ms, 97 ms, και 132 ms, αντίστοιχα.

Πραγματοποιώντας άλλη μια συγχώνευση, το τελικό σύνολο δρομολογίων είναι το $\{<0, 4, \underline{2}, \underline{1}, 0>, <0, 4, \underline{5}, \underline{4}, \underline{3}, 0>\}$, με χρόνους 138 ms και 168 ms, αντίστοιχα. Συνεπώς, ο συνολικός χρόνος που απαιτείται για συλλογή της πληροφορίας είναι 168 ms και τα δρομολόγια των πρακτόρων απεικονίζονται στο Σχήμα 4.6-α.

4.6.1.2 Εφαρμογή του “Insertion”

Στην περίπτωση αυτή, το βήμα αρχικοποίησης δημιουργεί τα στοιχειώδη δρομολόγια $<0, \underline{3}, 0>$ και $<0, \underline{4}, 0>$, με αντίστοιχους χρόνους 60 και 72 ms. Στη συνέχεια, καθένας από τους κόμβους 1, 2, και 5 ελέγχεται για εισαγωγή σε κάθε δυνατή θέση των στοιχειωδών δρομολογίων. Για παράδειγμα, εισάγοντας τον κόμβο 1 στο δρομολόγιο $<0, \underline{3}, 0>$ μπορεί να οδηγήσει σε κάποιο από τα δρομολόγια $<0, \underline{1}, \underline{3}, 0>$, $<0, \underline{3}, \underline{1}, 0>$ και $<0, 3, \underline{1}, \underline{3}, 0>$, με χρόνους 102 ms, 108 ms και 106 ms, αντίστοιχα. Τελικά, η εισαγωγή του κόμβου 1 στο δρομολόγιο $<0, \underline{3}, 0>$ αποδεικνύεται ως η πιο συμφέρουσα και επομένως τα δρομολόγια που προκύπτουν μετά την πρώτη εισαγωγή είναι το $<0, \underline{1}, \underline{3}, 0>$ με χρόνο 102 ms και το $<0, \underline{4}, 0>$ με χρόνο 72 ms.

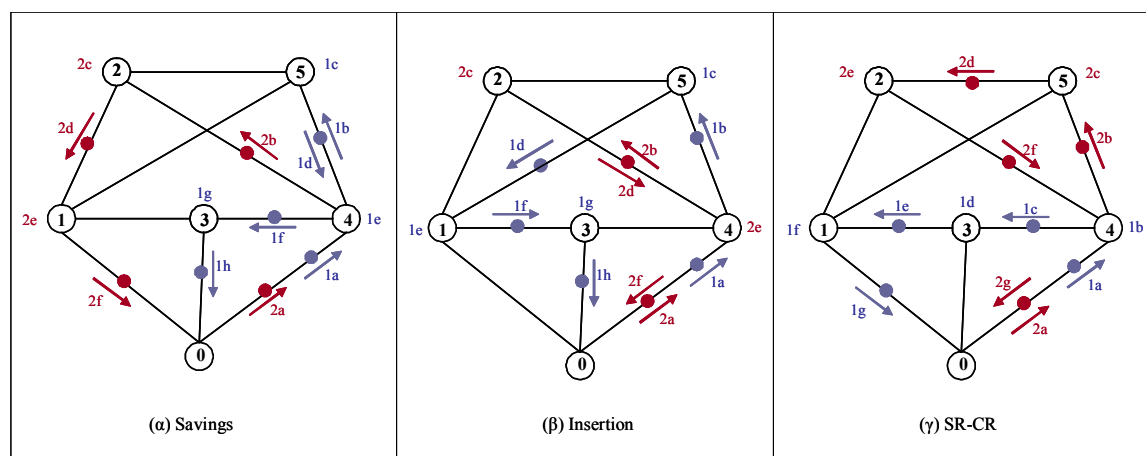
Ακολουθώντας την ίδια διαδικασία, ο επόμενος κόμβος που εισάγεται είναι ο 2, ενώ τα δρομολόγια όπως διαμορφώνονται μετά την εισαγωγή του είναι το $<0, \underline{1}, \underline{3}, 0>$ με χρόνο 102 ms και το $<0, 4, \underline{2}, \underline{4}, 0>$ με χρόνο 120 ms. Τέλος, η εισαγωγή του εναπομένου κόμβου 5 καθορίζει το τελικό σύνολο των δρομολογίων ως $\{<0, 4, \underline{5}, \underline{1}, \underline{3}, 0>, <0, 4, \underline{2}, \underline{4}, 0>\}$ με χρόνους 157 ms και 120 ms, αντίστοιχα. Συνεπώς, ο Insertion βρίσκει καλύτερη λύση από τον Savings, πετυχαίνοντας χρόνο ολοκλήρωσης της αποστολής ίσο με 157 ms. Το Σχήμα 4.6-β δείχνει τα δρομολόγια των πρακτόρων.

4.6.1.3 Εφαρμογή του “SR-CR”

Από το βήμα εξισορρόπησης φορτίου προκύπτει ότι ο ένας πράκτορας θα εξυπηρετήσει 3 κόμβους και ο άλλος 2. Τα δρομολόγια θα κατασκευαστούν διαδοχικά, ξεκινώντας από αυτό που περιλαμβάνει 3 κόμβους δεδομένων. Θεωρώντας με τη σειρά κάθε κόμβο από το σύνολο $\{1, 2, 3, 4, 5\}$ ως τον σταθερό από τον οποίο ξεκινά η δόμηση του πρώτου δρομολογίου από το τέλος του, θα προκύψουν τελικά τα επόμενα σύνολα δρομολογίων, καθένα από τα οποία αποτελεί μια ξεχωριστή λύση:

1. Το σύνολο $\{<0, \underline{4}, \underline{3}, \underline{1}, 0>, <0, 4, \underline{5}, \underline{2}, 4, 0>\}$, με μέγιστο χρόνο 153 ms.
2. Το σύνολο $\{<0, \underline{1}, \underline{5}, \underline{2}, 4, 0>, <0, \underline{4}, \underline{3}, 0>\}$, με μέγιστο χρόνο 176 ms.
3. Το σύνολο $\{<0, 4, \underline{5}, \underline{1}, \underline{3}, 0>, <0, 4, \underline{2}, \underline{4}, 0>\}$, με μέγιστο χρόνο 157 ms.
4. Το σύνολο $\{<0, 4, \underline{5}, \underline{2}, \underline{4}, 0>, <0, \underline{1}, \underline{3}, 0>\}$, με μέγιστο χρόνο 161 ms.
5. Το σύνολο $\{<0, \underline{4}, \underline{2}, \underline{5}, 4, 0>, <0, \underline{1}, \underline{3}, 0>\}$ μετατρέπεται στο $\{<0, 4, \underline{2}, \underline{5}, \underline{4}, 0>, <0, \underline{1}, \underline{3}, 0>\}$, με μέγιστο χρόνο 165 ms.

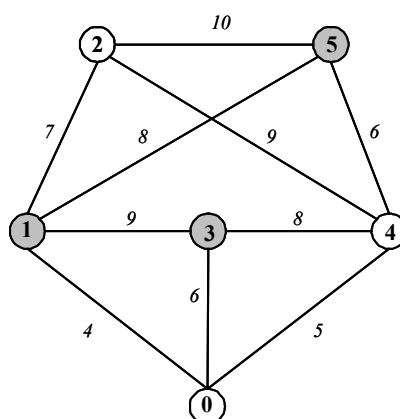
Η αποδεκτή λύση είναι αυτή με το μικρότερο χρόνο, δηλαδή η πρώτη (153 ms). Συμπερασματικά, συγκρίνοντας με τους άλλους δυο αλγορίθμους, ο SR-CR πετυχαίνει την καλύτερη λύση στο συγκεκριμένο παράδειγμα (Σχήμα 4.6-γ).



Σχήμα 4.6: Οι λύσεις των τριών αλγορίθμων στο παράδειγμα 1

4.6.2 ΠΑΡΑΔΕΙΓΜΑ 2: ΠΛΗΡΟΦΟΡΙΑ ΣΕ ΥΠΟΣΥΝΟΛΟ ΤΩΝ ΚΟΜΒΩΝ ΤΟΥ ΓΡΑΦΟΥ

Σκοπός του τρέχοντος παραδείγματος είναι να αναλύσουμε την εφαρμογή των αλγορίθμων στην περίπτωση που η ζητούμενη πληροφορία διαμένει σε ένα υποσύνολο των κόμβων του γράφου. Υποθέτουμε λοιπόν ότι οι κόμβοι 1, 3 και 5 παρέχουν την πληροφορία που ζητάται. Για διαφοροποίηση από την προηγούμενη περίπτωση, οι κόμβοι αυτοί έχουν χρωματιστεί (Σχήμα 4.7). Το αρχικό μέγεθος του πράκτορα ισούται με 20 Kbytes, ενώ το μέγεθος της πληροφορίας που ανακτάται από έναν κόμβο ισούται με 5 Kbytes. Αποστέλλοντας ταυτόχρονα δυο πράκτορες από τον κόμβο 0, σκοπός είναι να σχεδιαστούν τα δρομολόγια των πρακτόρων έτσι ώστε η ανάκτηση της πληροφορίας και η παράδοσή της στον κόμβο 0 να ολοκληρωθούν στον ελάχιστο χρόνο.



Σχήμα 4.7: Γράφος δικτύου παραδείγματος 2

4.6.2.1 Εφαρμογή του “Savings”

Από το βήμα αρχικοποίησης, το οποίο ασχολείται μόνο με τους κόμβους που φέρουν τη ζητούμενη πληροφορία, δημιουργούνται τα προκαταρκτικά δρομολόγια $\langle 0, \underline{1}, 0 \rangle$, $\langle 0, \underline{3}, 0 \rangle$ και $\langle 0, 4, \underline{5}, 4, 0 \rangle$ με αντίστοιχους χρόνους 90 ms, 60 ms, και 132 ms. Για να καταλήξουμε σε δυο δρομολόγια, όσα δηλαδή και οι πράκτορες, απαιτείται μια ακριβώς συγχώνευση. Δοκιμάζοντας κάθε δυνατή συγχώνευση μεταξύ δυο οποιονδήποτε δρομολογίων, όπως έχει περιγραφεί, επιλέγεται τελικά η πιο συμφέρουσα. Για παράδειγμα, συγχωνεύοντας το $\langle 0, \underline{3}, 0 \rangle$ με το $\langle 0, 4, \underline{5}, 4, 0 \rangle$ παράγεται το $\langle 0, \underline{3}, 1, \underline{5}, 4, 0 \rangle$ με χρόνο 161 ms, ενώ πραγματοποιώντας τη συγχώνευση με αντίθετη φορά προκύπτει το $\langle 0, 4, \underline{5}, 1, \underline{3}, 0 \rangle$ με χρόνο 145 ms. Επειδή όμως η συγχώνευση του $\langle 0, \underline{1}, 0 \rangle$ με το $\langle 0, \underline{3}, 0 \rangle$ αποδεικνύεται ως η πιο συμφέρουσα, το τελικό σύνολο δρομολογίων διαμορφώνεται ως $\{\langle 0, \underline{1}, \underline{3}, 0 \rangle, \langle 0, 4, \underline{5}, 4, 0 \rangle\}$ με χρόνους 102 ms και 132 ms, αντίστοιχα. Άρα, ο χρόνος ολοκλήρωσης της αποστολής είναι ίσος με 132 ms και τα δρομολόγια των πρακτόρων φαίνονται στο Σχήμα 4.8-α.

4.6.2.2 Εφαρμογή του “Insertion”

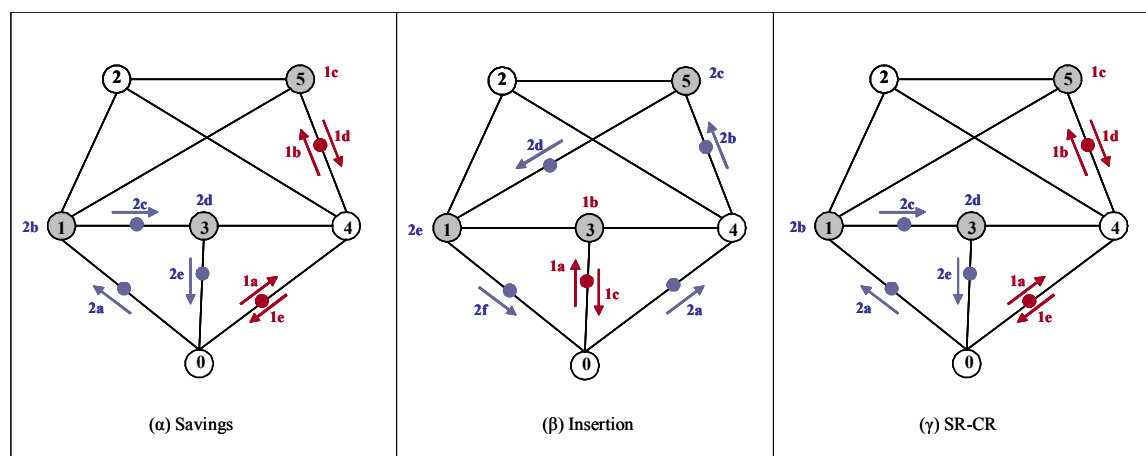
Το αρχικό σύνολο δρομολογίων είναι το $\{\langle 0, \underline{1}, 0 \rangle, \langle 0, \underline{3}, 0 \rangle\}$ με χρόνους 90 ms και 60 ms, αντίστοιχα. Δεδομένου ότι ο κόμβος 5 είναι ο μόνος που δεν έχει ανατεθεί σε κάποιο δρομολόγιο, αναζητείται η πιο συμφέρουσα θέση εισαγωγής του σε κάποιο από τα παραπάνω δρομολόγια. Για παράδειγμα, τα δυνατά δρομολόγια που μπορούν να προκύψουν από την εισαγωγή του 5 στο δρομολόγιο $\langle 0, \underline{3}, 0 \rangle$ είναι τα $\langle 0, 4, \underline{5}, 1, \underline{3}, 0 \rangle$, $\langle 0, \underline{3}, 1, \underline{5}, 4, 0 \rangle$ και $\langle 0, 3, 1, \underline{5}, 1, \underline{3}, 0 \rangle$, τα οποία απαιτούν 145 ms, 161 ms και 151 ms, αντίστοιχα. Από την άλλη μεριά, η εισαγωγή του 5 στο $\langle 0, \underline{1}, 0 \rangle$ έχει σαν αποτέλεσμα τα δυνατά δρομολόγια $\langle 0, 4, \underline{5}, \underline{1}, 0 \rangle$, $\langle 0, \underline{1}, \underline{5}, 4, 0 \rangle$ και $\langle 0, 1, \underline{5}, \underline{1}, 0 \rangle$, με αντίστοιχους χρόνους 143 ms, 153 ms και 145 ms. Συνεπώς, επιλέγεται η εισαγωγή του 5 στο $\langle 0, \underline{1}, 0 \rangle$ ως η πιο συμφέρουσα, και το τελικό σύνολο δρομολογίων γίνεται $\{\langle 0, 4, \underline{5}, \underline{1}, 0 \rangle, \langle 0, \underline{3}, 0 \rangle\}$ με χρόνους 143 ms και 60 ms, αντίστοιχα. Προφανώς, σε αυτή την περίπτωση ο Insertion παράγει χειρότερη λύση από τον Savings, πετυχαίνοντας συνολικό χρόνο αποστολής ίσο με 143 ms. Το Σχήμα 4.8-β δείχνει τα δρομολόγια των πρακτόρων.

4.6.2.3 Εφαρμογή του “SR-CR”

Σε αυτή την περίπτωση ο ένας πράκτορας θα εξυπηρετήσει 2 κόμβους με δεδομένα και ο άλλος 1 κόμβο, ενώ η κατασκευή των δρομολογίων αρχίζει με τον πρώτο. Θεωρώντας με τη σειρά κάθε κόμβο δεδομένων ως τον σταθερό από τον οποίο ξεκινά η δόμηση του πρώτου δρομολογίου από το τέλος του, τα επόμενα σύνολα δρομολογίων τελικά παράγονται:

1. Το σύνολο $\{\langle 0, \underline{3}, \underline{1}, 0 \rangle, \langle 0, 4, \underline{5}, 4, 0 \rangle\}$, με μέγιστο χρόνο 132 ms.
2. Το σύνολο $\{\langle 0, \underline{1}, \underline{3}, 0 \rangle, \langle 0, 4, \underline{5}, 4, 0 \rangle\}$, με μέγιστο χρόνο 132 ms.
3. Το σύνολο $\{\langle 0, \underline{1}, \underline{5}, 4, 0 \rangle, \langle 0, \underline{3}, 0 \rangle\}$, με μέγιστο χρόνο 153 ms.

Η πρώτη και η δεύτερη λύση απαιτούν τον ίδιο συνολικό χρόνο, αλλά η δεύτερη έχει μικρότερο άθροισμα χρόνων των συνιστωσών διαδρομών, με αποτέλεσμα να γίνει αυτή αποδεκτή. Άρα λοιπόν ο SR-CR βρίσκει την ίδια λύση με τον Savings (Σχήμα 4.8-γ).



Σχήμα 4.8: Οι λύσεις των τριών αλγορίθμων στο παράδειγμα 2

4.7 ΑΠΟΤΕΛΕΣΜΑΤΑ

Προκειμένου να πειραματιστούμε με τη συμπεριφορά των προτεινόμενων αλγορίθμων και συγχρόνως με το ίδιο το πρόβλημα, υλοποιήσαμε μια προσομοίωση σε γλώσσα προγραμματισμού *Java*. Οι δοκιμές πραγματοποιήθηκαν σε δίκτυα, τα οποία παρήχθησαν με βάση το μοντέλο *Waxman* [18]. Αυτό το μοντέλο διανέμει τυχαία τους κόμβους πάνω σε ορθογώνιο πλέγμα. Η ευκλείδεια απόσταση χρησιμοποιείται για τον προσδιορισμό της απόστασης μεταξύ δυο οποιονδήποτε κόμβων. Δύο κόμβοι u, v συνδέονται βάσει πιθανότητας $P(u, v)$, η οποία εξαρτάται από την απόστασή τους ως εξής:

$$P(u, v) = a e^{-\frac{d(u, v)}{L\beta}}$$

όπου το $d(u, v)$ παριστάνει την απόσταση μεταξύ των u και v , το L είναι η μέγιστη απόσταση μεταξύ δυο οποιονδήποτε κόμβων, και a, β είναι παράμετροι που παίρνουν τιμές στο διάστημα $(0, 1)$. Αύξηση της τιμής του a συνεπάγεται αύξηση του αριθμού των ακμών στο γράφο, ενώ αύξηση του β σημαίνει αύξηση της αναλογίας των μακριών ακμών σε σχέση με τις κοντύτερες.

Αξίζει να σημειωθεί ότι οι γράφοι που χρησιμοποιούνται για τις δοκιμές έχουν όλοι την ίδια πυκνότητα ($|E| \cong 1/3|V|^2$), εκτός από ειδικές περιπτώσεις που αναφέρονται ρητά, και επίσης ο ρυθμός μετάδοσης κάθε γραμμής παίρνει τιμές στο διάστημα $(1, 10)$ Mbps.

Οι προτεινόμενοι αλγόριθμοι συγκρίνονται τόσο από πλευράς ποιότητας των λύσεων που παράγουν όσο και χρόνου υπολογισμού που απαιτούν. Σημειώνεται ότι ο *Βέλτιστος* αλγόριθμος, ο οποίος ψάχνει εξαντλητικά το χώρο των δυνατών λύσεων, απαιτεί εξαιρετικά

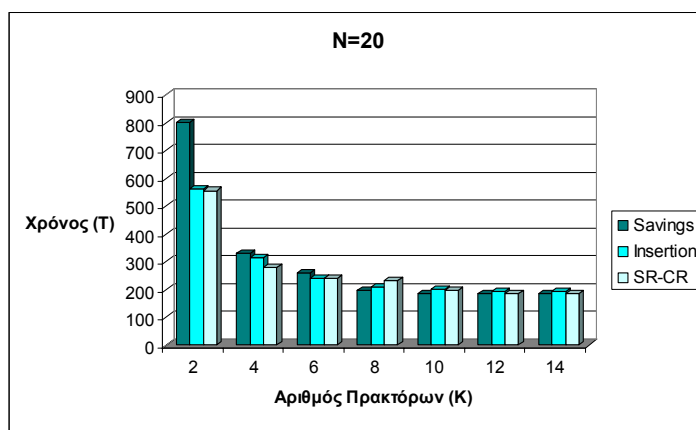
μεγάλο χρόνο υπολογισμού, για $N > 10$. Προκειμένου να παρασχεθεί μια εκτίμηση της προσέγγισης στο Βέλτιστο που επιτυγχάνουν οι προτεινόμενοι αλγόριθμοι, αναφέρεται η μέγιστη απόκλιση τους από αυτόν σε περιπτώσεις μικρών γράφων ($N < 10$). Επίσης αναφέρονται και ορισμένα αποτελέσματα σχετικά με την επίδραση διαφόρων παραμέτρων του προβλήματος στη λύση του.

4.7.1 ΣΥΓΚΡΙΣΗ ΑΛΓΟΡΙΘΜΩΝ

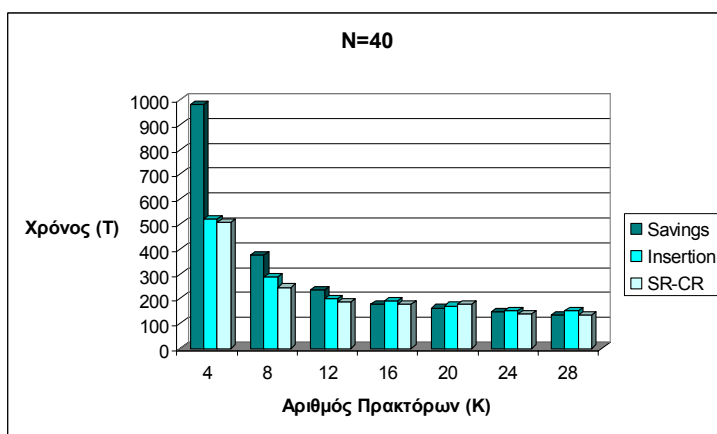
4.7.1.1 Περίπτωση I^1 : Πληροφορία σε Όλους τους Κόμβους του Γράφου

Αρχίζουμε τις δοκιμές χρησιμοποιώντας ένα γράφο που αποτελείται από 20 κόμβους και μεταβάλλουμε τον αριθμό των πρακτόρων από 2 έως 14. Το μέγεθος κάθε πράκτορα είναι ίσο με 20Kbytes ενώ το μέγεθος της πληροφορίας που ανακτάται από έναν κόμβο ισούται με 5Kbytes. Αυξάνοντας σταδιακά το μέγεθος του προβλήματος, δηλαδή τον αριθμό των κόμβων του γράφου, και αναλόγως αυξάνοντας τον αριθμό των πρακτόρων που αποστέλλονται μέχρι το σημείο που αντιστοιχεί στο χαμηλότερο χρόνο ολοκλήρωσης της αποστολής (το οποίο φαίνεται ότι επιτυγχάνεται χρησιμοποιώντας έως και 30% λιγότερους πράκτορες από τον αριθμό των κόμβων), παρατηρούμε τα ακόλουθα:

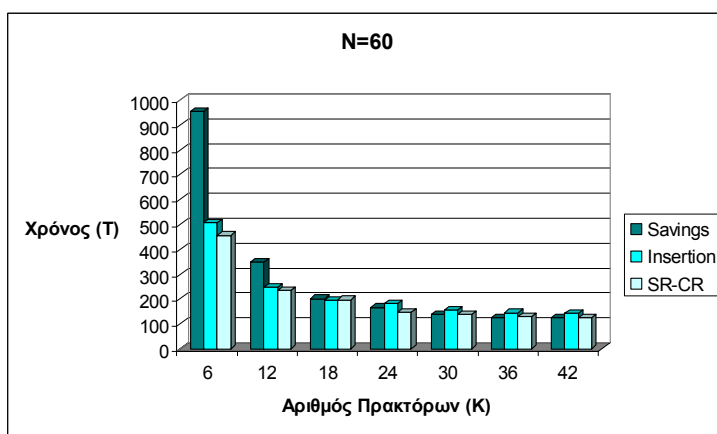
Όταν ο αριθμός των πρακτόρων είναι υψηλός σε σχέση με τον αριθμό των κόμβων (προσεγγιστικά $K/N > 0,4$), ο Savings παράγει καλύτερες λύσεις από τον Insertion ή τον SR-CR, ενώ ο SR-CR παράγει επίσης ενδιαφέρουσες λύσεις. Καθώς όμως ο αριθμός πρακτόρων μειώνεται, οι λύσεις που παράγονται από τον Savings χειροτερεύουν σταδιακά, καθιστώντας απαγορευτική την εφαρμογή του για πολύ μικρό αριθμό πρακτόρων. Κατά συνέπεια, ένα μειονέκτημα του Savings είναι ότι γενικά παράγει καλά δρομολόγια στην αρχή, αλλά λιγότερο ενδιαφέροντα προς το τέλος, όπου ο αριθμός των δρομολογίων έχει μειωθεί λόγω των πολλαπλών συγχωνεύσεων. Για την τελευταία περίπτωση, όταν δηλαδή ο αριθμός των πρακτόρων είναι μικρός, ο SR-CR φαίνεται να παράγει τις καλύτερες λύσεις ενώ ο Insertion έρχεται δεύτερος. Τα Σχήματα 4.9, 4.10, 4.11, και 4.12 δείχνουν τις τιμές των λύσεων που παράγουν και οι τρεις υπό εξέταση αλγόριθμοι, για τιμή του N ίση με 20, 40, 60 και 80, αντίστοιχα και μεταβλητό αριθμό πρακτόρων.



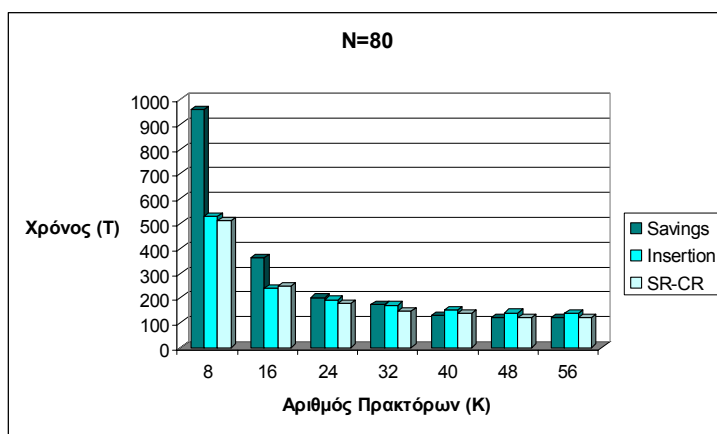
Σχήμα 4.9: Χρόνος ολοκλήρωσης αποστολής συναρτήσει του αριθμού πρακτόρων, για N=20



Σχήμα 4.10: Χρόνος ολοκλήρωσης αποστολής συναρτήσει του αριθμού πρακτόρων, για N=40



Σχήμα 4.11: Χρόνος ολοκλήρωσης αποστολής συναρτήσει του αριθμού πρακτόρων, για N=60



Σχήμα 4.12: Χρόνος ολοκλήρωσης αποστολής συναρτήσει του αριθμού πρακτόρων, για $N=80$

Επίσης πειραματιστήκαμε με την αποτελεσματικότητα των αλγορίθμων σε σχέση με τον Βέλτιστο σε μικρά μόνο δίκτυα εξαιτίας του εξαντλητικού χρόνου εκτέλεσης του Βέλτιστου. Για μέγεθος δικτύου έως 10 κόμβους, παρατηρήσαμε ότι για $K/N > 0,4$ ο Savings πράγματι παράγει την καλύτερη λύση συγκριτικά με τους άλλους δυο αλγορίθμους, παρουσιάζοντας μέγιστη απόκλιση από τον Βέλτιστο ίση με 4%. Αντίθετα, για $K/N < 0,4$, ο SR-CR προσεγγίζει ακόμη πιο πολύ τον Βέλτιστο, σημειώνοντας μέγιστη απόκλιση 2%.

Μέχρι τώρα έχουμε αξιολογήσει τους αλγορίθμους βάσει μόνο της ποιότητας των λύσεων που παράγουν. Εντούτοις, ο χρόνος εκτέλεσης ενός αλγορίθμου θεωρείται επίσης σημαντικός παράγοντας αποτίμησης κατά την επιλογή ενός αλγορίθμου αντί κάποιου άλλου. Για το λόγο αυτό καταγράψαμε επίσης το χρόνο εκτέλεσης που απαιτούν οι αλγόριθμοι στα παραπάνω πειράματα και παραθέτουμε ενδεικτικά τις σχετικές τιμές για $N = 40$ και $N = 80$ στους Πίνακες 4.1 και 4.2, αντίστοιχα. Σημειώνουμε ωστόσο, ότι ουσιαστικά δεν ενδιαφερόμαστε για τον ακριβή χρόνο εκτέλεσης των αλγορίθμων, αλλά για τον τρόπο που αυτός μεταβάλλεται με το μέγεθος της εισόδου, προκειμένου να επαληθεύσουμε την ισχύ της ανάλυσης χρονικής πολυπλοκότητας που αναπτύχθηκε στην ενότητα 4.5.

Μια απευθείας σύγκριση των αλγορίθμων είναι δυνατή δεδομένου ότι όλοι εκτελούνται στο ίδιο ακριβώς περιβάλλον. Συγκεκριμένα, όλα τα πειράματα πραγματοποιήθηκαν σε έναν υπολογιστή Windows με ταχύτητα επεξεργαστή 1.4 GHz (Centrino) και μνήμη RAM 512MB. Μελετώντας τους πίνακες, παρατηρούμε ότι ο χρόνος εκτέλεσης και των τριών αλγορίθμων μεταβάλλεται με τον ίδιο τρόπο με τον αριθμό κόμβων του γράφου. Επιπλέον, ο χρόνος εκτέλεσης του Savings και του Insertion μειώνεται με τον αριθμό πρακτόρων, ενώ του SR-CR αυξάνεται. Είναι επίσης αξιοπρόσεχτο ότι οι υπολογιστικές απαιτήσεις του Insertion γίνονται απαγορευτικές για έναν σχετικά υψηλό αριθμό κόμβων και έναν χαμηλό αριθμό πρακτόρων. Είναι λοιπόν προφανές ότι τα πειραματικά αποτελέσματα συμφωνούν με τη θεωρητική περιγραφή της χρονικής πολυπλοκότητας των αλγορίθμων που δόθηκε νωρίτερα. Σημειώνεται ότι ο Savings και ο

Insertion έχουν υποστεί τη βασική βελτιστοποίηση κώδικα που απαιτείται προκειμένου να αποφευχθούν οι περιττοί υπολογισμοί που είναι πιθανό να προκύψουν λόγω των επαναλαμβανόμενων συγχωνεύσεων ή εισαγωγών σε διαφορετικές επαναλήψεις του αλγορίθμου. Καμία ειδική τεχνική για περαιτέρω βελτιστοποίηση του κώδικα δεν έχει εφαρμοστεί. Εντούτοις, είναι δυνατό να μειωθούν σημαντικά οι χρόνοι εκτέλεσης των αλγορίθμων με την ανάπτυξη εξειδικευμένου κώδικα, τη χρησιμοποίηση έξυπνων δομών δεδομένων ή την εφαρμογή παράλληλου υπολογισμού.

Σε σχέση και με την ποιότητα των λύσεων, τα συμπεράσματα που προκύπτουν είναι τα εξής: ο SR-CR παράγει συγκριτικά με τους άλλους δυο αλγορίθμους τα καλύτερα αποτελέσματα και μάλιστα στο συντομότερο χρόνο, όταν ο αριθμός των πρακτόρων είναι σχετικά μικρός. Αντίθετα, ένας αυξημένος αριθμός πρακτόρων ευνοεί τον Savings, ο οποίος στην περίπτωση αυτή παράγει καλύτερες λύσεις από τους άλλους δυο, επιδεικνύοντας ταυτόχρονα εντυπωσιακή υπολογιστική απόδοση. Τέλος, η ικανότητα του Insertion να παράγει σχετικά καλές λύσεις όταν ο αριθμός των πρακτόρων είναι μικρός συνοδεύεται από εκτεταμένο χρόνο εκτέλεσης. Η υπολογιστική του απόδοση βελτιώνεται σημαντικά καθώς ο αριθμός των πρακτόρων αυξάνει, αλλά υπό το κόστος της παραγωγής λύσεων κατώτερης ποιότητας. Για το λόγο αυτό, ο Insertion θεωρείται λιγότερο κατάλληλος για το υπό μελέτη πρόβλημα, ενώ η συνδυασμένη εφαρμογή του SR-CR και του Savings υπόσχεται να λύσει τις περισσότερες περιπτώσεις του προβλήματος αποτελεσματικά και αποδοτικά.

Αξίζει όμως να σημειωθεί ότι το αυξημένο υπολογιστικά κόστος του Insertion θα μπορούσε να μειωθεί σημαντικά, εάν η εισαγωγή ενός κόμβου σε ένα δεδομένο δρομολόγιο θεωρούνταν μόνο στην περίπτωση που αυτός ήταν γειτονικός σε κάποιον από τους κόμβους του δρομολογίου. Αυτό όμως είναι δυνατό μόνο στην υπό μελέτη περίπτωση, δηλαδή όταν όλοι οι κόμβοι φιλοξενούν τη ζητούμενη πληροφορία. Με αυτή την παραλλαγή, οι λύσεις που παράγει ο Insertion χειροτερεύουν ελαφρώς, επιτρέποντας στην κατάταξη των αλγορίθμων σε σχέση με την ποιότητα των λύσεων να παραμείνει η ίδια, ενώ από πλευράς χρόνου εκτέλεσης ο Insertion γίνεται ο πιο αποδοτικός από τους τρεις αλγορίθμους.

		Savings	Insertion	SR-CR
N=40	K=4	63	154	54
	K=8	62	112	60
	K=12	59	101	67
	K=16	57	91	73
	K=20	54	82	79
	K=24	51	71	86
	K=28	45	57	92

Πίνακας 4.1: Χρόνοι εκτέλεσης (σε msec) των αλγορίθμων, για N=40

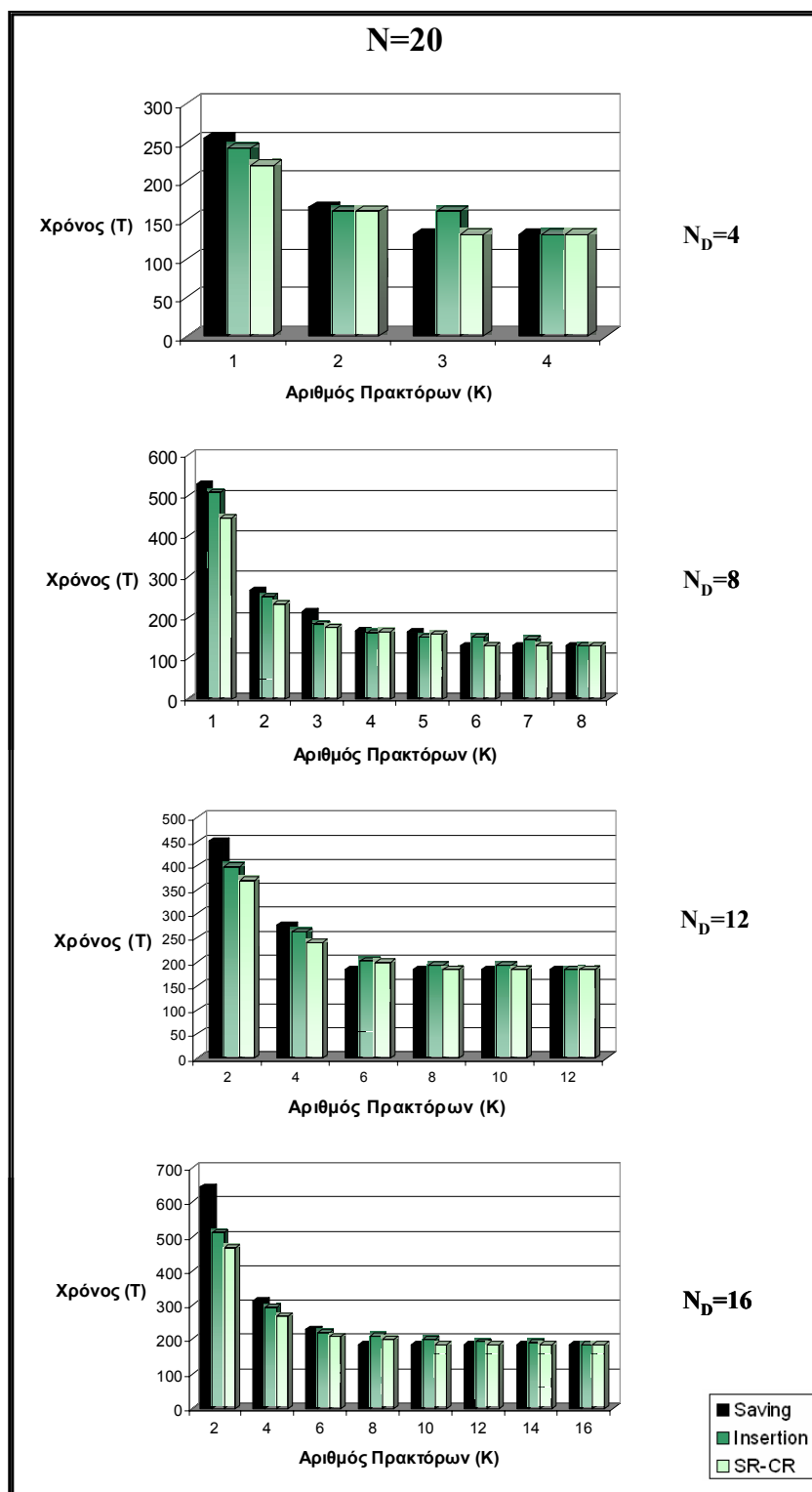
		Savings	Insertion	SR-CR
N=80	K=8	1037	2583	931
	K=16	1016	1826	1020
	K=24	986	1648	1105
	K=32	942	1472	1193
	K=40	889	1289	1277
	K=48	826	1085	1363
	K=56	753	855	1450

Πίνακας 4.2: Χρόνοι εκτέλεσης (σε msec) των αλγορίθμων, για $N=80$

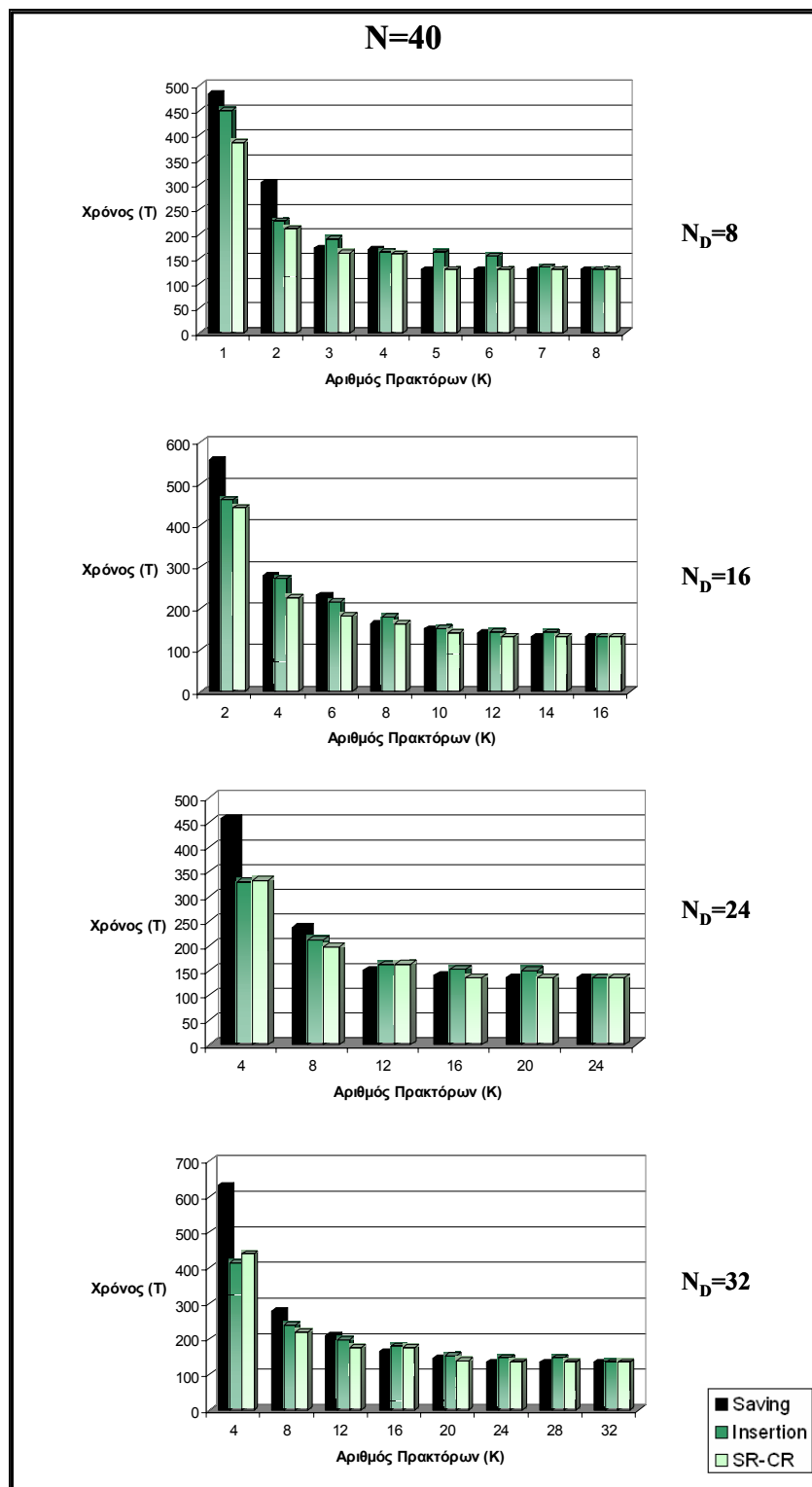
4.7.1.2 Περίπτωση 2^η: Πληροφορία σε Υποσύνολο των Κόμβων του Γράφου

Αρχίζουμε τις δοκιμές με ένα γράφο που αποτελείται από 20 κόμβους και μεταβάλλουμε τον αριθμό των κόμβων που περιέχουν τη ζητούμενη πληροφορία, καθώς και τον αριθμό των πρακτόρων που αποστέλλονται για να ανακτήσουν αυτή την πληροφορία. Το μέγεθος κάθε πράκτορα είναι ίσο με 20Kbytes, ενώ το μέγεθος της πληροφορίας που ανακτάται από έναν κόμβο ισούται με 5Kbytes. Αυξάνοντας βαθμιαία το μέγεθος του προβλήματος, δηλαδή τον αριθμό των κόμβων του γράφου, και αναλόγως μεταβάλλοντας τον αριθμό των κόμβων που περιέχουν την πληροφορία και τον αριθμό των πρακτόρων που αποστέλλονται, παρατηρούμε τα εξής:

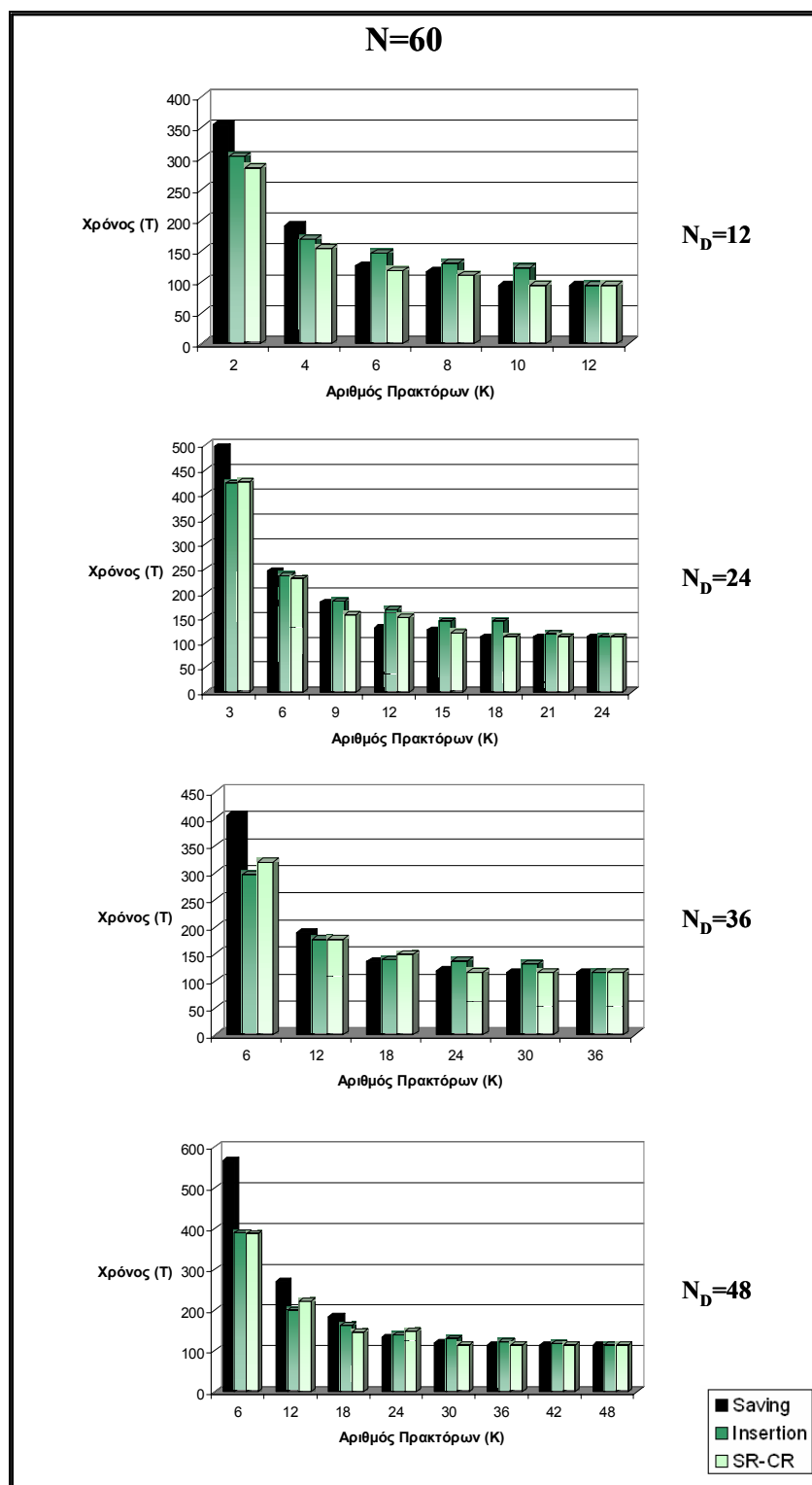
Για γράφο με τους περισσότερους από τους κόμβους του να φιλοξενούν τη ζητούμενη πληροφορία (προσεγγιστικά $N_D/N > 0,5$, όπου $N_D = |V'|$), τα αποτελέσματα συμφωνούν με αυτά της πρώτης περίπτωσης, όπου ο γράφος είναι «πλημμυρισμένος» με τη ζητούμενη πληροφορία. Εντούτοις, μια ελαφριά διαφοροποίηση παρατηρείται καθώς οι κόμβοι που φιλοξενούν τη ζητούμενη πληροφορία γίνονται λιγότεροι. Σε αυτή την περίπτωση, ο SR-CR φαίνεται να υπερέχει συνολικά του Savings και του Insertion όσον αφορά στην ποιότητα της λύσης. Πιο συγκεκριμένα, όπως και στην πρώτη περίπτωση, για ένα χαμηλό αριθμό πρακτόρων συγκριτικά με τον αριθμό κόμβων πληροφορίας, ο SR-CR θεωρείται συγκριτικά με τους άλλους δυο ο καλύτερος, λαμβάνοντας υπόψη τόσο την ποιότητα της λύσης όσο και το χρόνο εκτέλεσης. Σε αντίθεση με την πρώτη περίπτωση όμως, για έναν υψηλό αριθμό πρακτόρων ο Savings συναγωνίζεται με τον SR-CR για την καλύτερη λύση, με τον SR-CR να είναι ο επικρατέστερος. Η αποτελεσματικότητα όμως του SR-CR σε αυτή την περίπτωση συνοδεύεται από υψηλές υπολογιστικές απαιτήσεις, αφού για $K/N_D > 0,4$, ο SR-CR είναι ο πιο χρονοβόρος, ενώ ο Savings καταφέρνει να παράγει ανταγωνιστικές λύσεις σε λιγότερο χρόνο. Τα Σχήματα 4.13, 4.14 και 4.15 απεικονίζουν με συγκριτικό τρόπο τις λύσεις που παράγονται από τους τρεις αλγορίθμους για $N=20, 40$, και 60 , $N_D/N=0,2, 0,4, 0,6, 0,8$, και μεταβλητό αριθμό πρακτόρων.



Σχήμα 4.13: Χρόνος ολοκλήρωσης αποστολής συναρτήσει του αριθμού πρακτόρων, για $N=20$ και $N_D=4, 8, 12, 16$



Σχήμα 4.14: Χρόνος ολοκλήρωσης αποστολής συναρτήσει του αριθμού πρακτόρων, για $N=40$ και $N_D=8, 16, 24, 32$

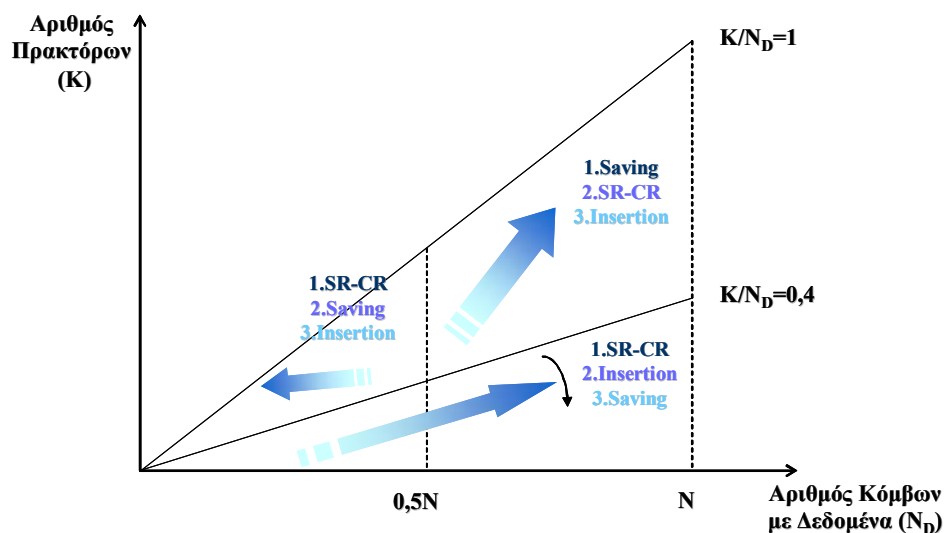


Σχήμα 4.15: Χρόνος ολοκλήρωσης αποστολής συναρτήσει του αριθμού πρακτόρων, για $N=60$ και $N_D=12, 24, 36, 48$

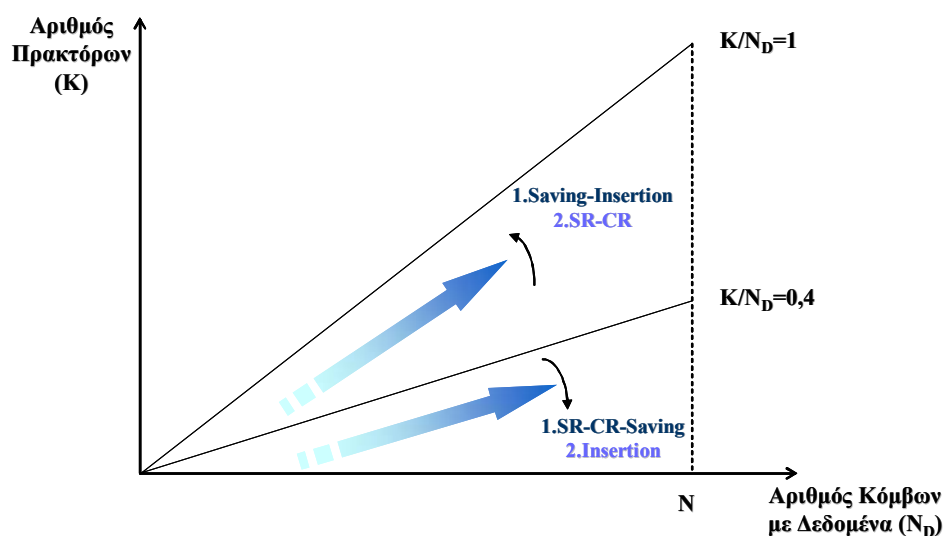
4.7.1.3 Γενική Αποτίμηση των Αλγορίθμων

Με βάση τα αποτελέσματα που προέκυψαν από τα παραπάνω πειράματα, επιχειρείται στη συνέχεια η κατάταξη των αλγορίθμων σε σχέση με την ποιότητα των λύσεων που παράγουν και το χρόνο εκτέλεσης που απαιτούν, σε μια προσπάθεια να εντοπιστεί ποιος από αυτούς είναι καταλληλότερος στις διάφορες περιοχές του παραμετρικού χώρου. Θεωρώντας σαν όριο τη γραμμή $K/N_D=0,4$ και μειώνοντας την τιμή του K , ο SR-CR υπερέχει σαφέστατα, ενώ ο Insertion παράγει επίσης ενδιαφέρουσες λύσεις. Αυξάνοντας την τιμή του K , ο Savings έρχεται στο προσκήνιο και συναγωνίζεται με τον SR-CR όσον αφορά τον εντοπισμό της καλύτερης λύσης, με τον πρώτο να υπερέχει σε γράφους στους οποίους η πληροφορία είναι διασκορπισμένη στην πλειοψηφία των κόμβων, και τον δεύτερο στην αντίθετη περίπτωση (Σχήμα 4.16). Θεωρώντας ως κριτήριο το χρόνο εκτέλεσης, καθώς η τιμή του K μειώνεται, ο SR-CR και ο Savings αποδεικνύονται ως οι πιο κατάλληλοι, ενώ για υψηλότερες τιμές του K , οι υπολογιστικές απαιτήσεις του SR-CR αυξάνονται σημαντικά. Στην περίπτωση αυτή ο Insertion συναγωνίζεται με τον Savings (Σχήμα 4.17).

Συνοψίζοντας, μια συνολική αποτίμηση των αλγορίθμων λαμβάνοντας υπόψη τόσο την ποιότητα των λύσεων που πετυχαίνουν όσο και τους υπολογιστικούς πόρους που καταναλώνουν υποδεικνύει ότι ο SR-CR είναι ο πιο κατάλληλος από τους τρεις για $K/N_D < 0,4$, ενώ ο Savings έρχεται πρώτος για $K/N_D > 0,4$.



Σχήμα 4.16: Κατάταξη των αλγορίθμων με βάση την ποιότητα των λύσεων που παράγουν



Σχήμα 4.17: Κατάταξη των αλγορίθμων με βάση το χρόνο εκτέλεσής τους

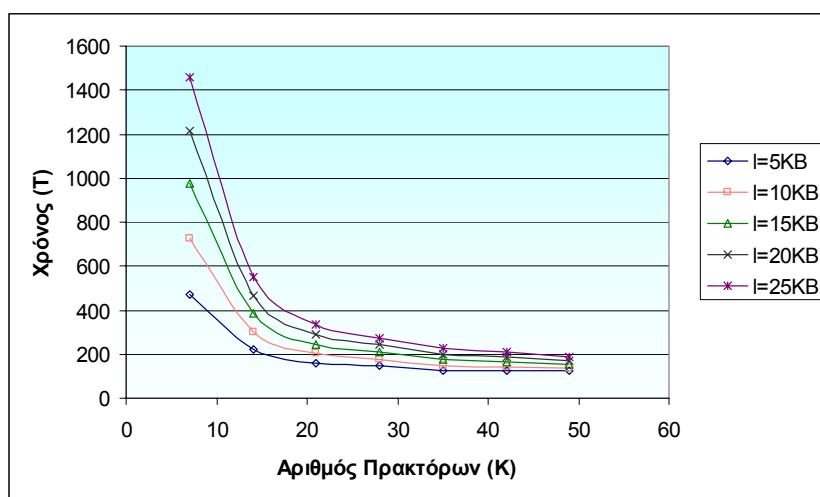
4.7.2 ΜΕΛΕΤΗ ΕΠΙΔΡΑΣΗΣ ΠΑΡΑΜΕΤΡΩΝ

Σε αυτή την ενότητα θα μελετήσουμε την επιρροή στο συνολικό χρόνο συλλογής της πληροφορίας που ασκεί το μέγεθος της πληροφορίας που ανακτάται από τυχαίο κόμβο, το αρχικό μέγεθος των κινητών πρακτόρων, η πυκνότητα του δικτύου και η ύπαρξη συνδέσεων συμφόρησης στον κόμβο του κεντρικού αποστολέα. Για το σκοπό αυτό, χρησιμοποιούμε γράφο αποτελούμενο από 70 κόμβους, μεταβάλλουμε τον αριθμό των πρακτόρων από 7 έως 49 με βήμα ίσο με 7, και εξετάζουμε καθεμιά από τις παραπάνω παραμέτρους χωριστά. Ο αλγόριθμος που εφαρμόζουμε είναι ο SR-CR εξαιτίας της αποτελεσματικότητάς του στο σύνολο των περιπτώσεων. Επιπλέον, για υψηλότερη κλίμακα, θεωρούμε την περίπτωση όπου όλοι οι κόμβοι περιέχουν τη ζητούμενη πληροφορία. Σκοπός αυτών των πειραμάτων είναι να μελετηθεί πιο προσεκτικά η σχέση μεταξύ κάθε παραμέτρου του προβλήματος και του αριθμού πρακτόρων, ώστε να προκύψουν χρήσιμα συμπεράσματα σχετικά με την αποτελεσματικότητα της χρήσης κινητών πρακτόρων στα πλαίσια οποιασδήποτε εφαρμογής συλλογής κατανεμημένης πληροφορίας [19].

Κατά την επεξεργασία των αποτελεσμάτων, το κριτήριο επίδοσης που χρησιμοποιείται είναι η μέση σχετική βελτίωση του χρόνου ανά στοιχειώδη αύξηση του πληθυσμού πρακτόρων (*Mean Relative Improvement - MRI*), η οποία προκύπτει υπολογίζοντας το μέσο όρο των σχετικών βελτιώσεων σε κάθε διάστημα μεταβολής του πληθυσμού πρακτόρων. Η σχετική βελτίωση σε κάποιο διάστημα μεταβολής υπολογίζεται για σταθερή αύξηση του πληθυσμού πρακτόρων και ίση με 7, ως $(T_{init} - T_{fin}) / T_{init}$, όπου T_{init} και T_{fin} είναι η τιμή του χρόνου για αριθμό πρακτόρων που αντιστοιχεί στην αρχή και το τέλος του δεδομένου διαστήματος μεταβολής, αντίστοιχα.

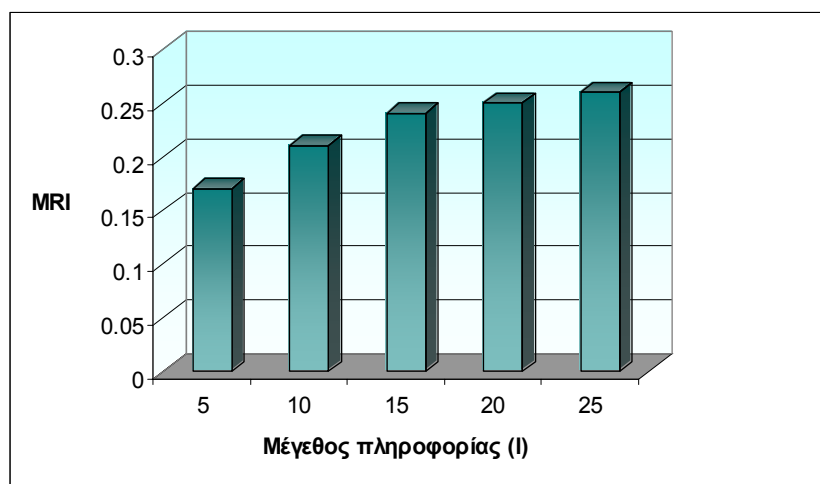
4.7.2.1 Επίδραση μεγέθους πληροφορίας

Αρχικά θα μελετήσουμε την επίδραση του μεγέθους της πληροφορίας που ανακτάται από κάθε κόμβο στο συνολικό χρόνο ανάκτησης της πληροφορίας. Το αρχικό μέγεθος των πρακτόρων είναι ίσο με 20Kbytes και επιλέγουμε για το μέγεθος της πληροφορίας που ανακτάται τις τιμές 5, 10, 15, 20 και 25Kbytes. Από το Σχήμα 4.18 παρατηρούμε τα εξής: καθώς ο αριθμός πρακτόρων αυξάνει, ο χρόνος ολοκλήρωσης της αποστολής μειώνεται, αλλά μετά από κάποιο συγκεκριμένο αριθμό πρακτόρων η βελτίωση αυτή γίνεται αμελητέα. Αυξάνοντας βαθμιαία το μέγεθος της πληροφορίας που ανακτάται από κάθε κόμβο, η προκύπτουσα γραφική παράσταση γίνεται πιο απότομη, δηλαδή καθώς ο αριθμός πρακτόρων αυξάνει, η βελτίωση γίνεται όλο και πιο εμφανής και συνεπώς το σημείο στον άξονα x πέρα από το οποίο δεν παρατηρείται σημαντική μείωση του χρόνου μετατοπίζεται προς τα δεξιά.



Σχήμα 4.18: Χρόνος συναρτήσει αριθμού πρακτόρων για διάφορα μεγέθη πληροφορίας

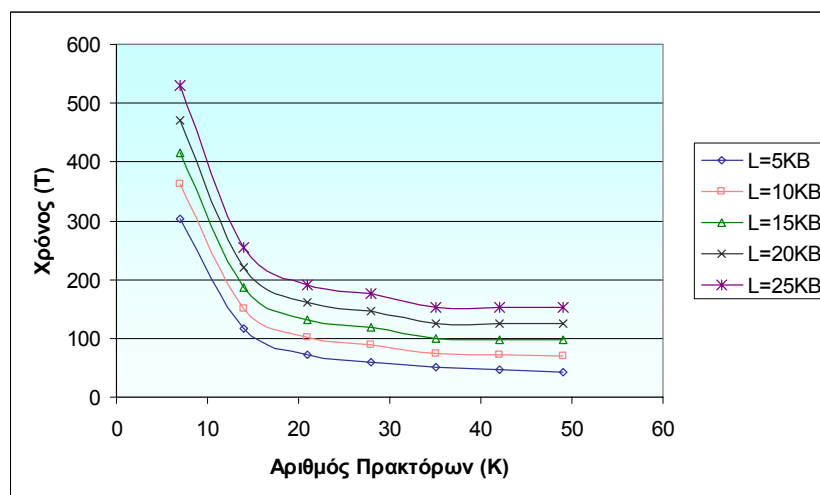
Θεωρώντας ως κριτήριο επίδοσης το MRI, το οποίο με τον τρόπο που ορίστηκε παρέχει για μια συγκεκριμένη τιμή του μεγέθους πληροφορίας μια ένδειξη της μέσης σχετικής βελτίωσης που επιτυγχάνεται από μια αύξηση του αριθμού πρακτόρων κατά 7, παρατηρούμε ότι όσο αυξάνει ο όγκος της ανακτηθείσας πληροφορίας τόσο πιο ευεργετική είναι μια αύξηση του αριθμού πρακτόρων (Σχήμα 4.19). Κατά συνέπεια, οι εφαρμογές που συνεπάγονται μεταφορά μεγάλου όγκου πληροφοριών αναμένεται να επωφελούνται περισσότερο από μεγάλους πληθυσμούς πρακτόρων.



Σχήμα 4.19: MRI για διάφορα μεγέθη πληροφορίας

4.7.2.2 Επίδραση αρχικού μεγέθους κινητού πράκτορα

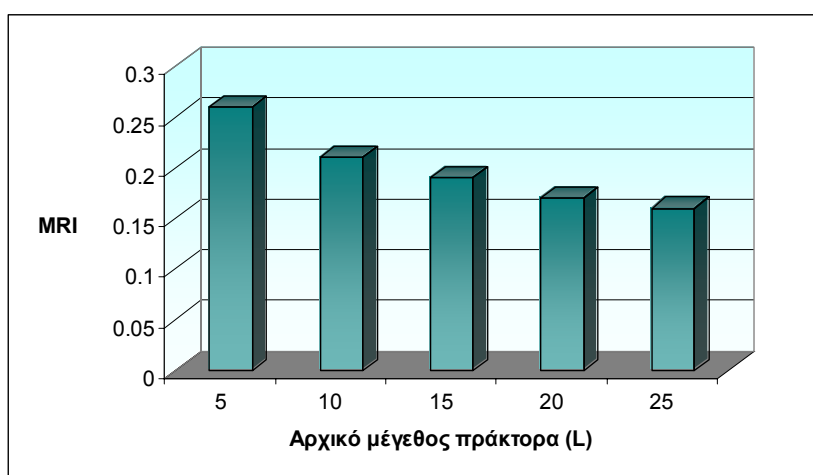
Για τη μελέτη της επίδρασης του αρχικού μεγέθους των κινητών πρακτόρων, χρησιμοποιούνται 5 διαφορετικά αρχικά μεγέθη (5, 10, 15, 20 και 25Kbytes), ενώ το μέγεθος της πληροφορίας που ανακτάται από κάθε κόμβο ρυθμίζεται στην τιμή 5Kbytes. Η μεταβολή του χρόνου ολοκλήρωσης της αποστολής σε σχέση με τον αριθμό πρακτόρων για τις διαφορετικές τιμές του αρχικού μεγέθους πρακτόρων απεικονίζεται στο Σχήμα 4.20.



Σχήμα 4.20: Χρόνος συναρτήσει αριθμού πρακτόρων για διάφορα αρχικά μεγέθη πρακτόρων

Με βάση το κριτήριο επίδοσης MRI, παρατηρούμε ότι για αυξανόμενο αρχικό μέγεθος πρακτόρων, η σχετική βελτίωση του χρόνου εξαιτίας της αύξησης του αριθμού πρακτόρων μειώνεται (Σχήμα 4.21). Αυτό εξηγείται ως εξής: είναι προφανές ότι καθώς ο αριθμός των επισκεπτόμενων κόμβων από έναν πράκτορα αυξάνει, το μέγεθος του κινητού πράκτορα θα αυξάνει επίσης, καθιστώντας τη μετακίνησή του πιο αργή. Αυτό όμως που

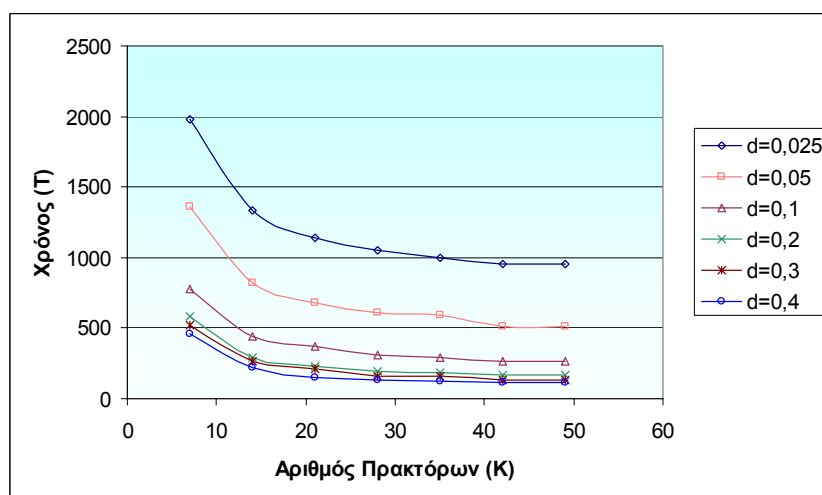
τελικά επηρεάζει τη λύση είναι ο λόγος του όγκου της πληροφορίας που ανακτάται από ένα πράκτορα προς το αρχικό μέγεθος του πράκτορα. Εάν υποθεθεί ότι ένας συγκεκριμένος όγκος πληροφορίας ανακτάται από κάθε κόμβο, ένας σχετικά «βαρύς» κινητός πράκτορας θα «άντεχε» να εξυπηρετήσει έναν υψηλό αριθμό κόμβων, αφού σε αυτή την περίπτωση ο χρόνος του δρομολογίου του θα αυξάνονταν αργά με τους κόμβους που θα εξυπηρετούσε. Αντίθετα, ένας σχετικά «ελαφρύς» κινητός πράκτορας θα γινόταν πιο γρήγορα «αργοκίνητος», αφού ο χρόνος ολοκλήρωσης της διαδρομής του θα αυξάνονταν με πιο γρήγορο ρυθμό με τον αριθμό των κόμβων που θα εξυπηρετούσε. Καθώς όμως ενδιαφερόμαστε για χαμηλό συνολικό χρόνο ολοκλήρωσης της αποστολής, όσο πιο μεγάλο είναι το αρχικό μέγεθος των πρακτόρων, τόσο πιο πολλούς κόμβους θα μπορούσε να εξυπηρετήσει καθένας από αυτούς και συνεπώς τόσο λιγότεροι πράκτορες θα απαιτούνταν. Επομένως, η αποταμίευση χρόνου που επιτυγχάνεται από τη χρησιμοποίηση περισσότερων πρακτόρων γίνεται πιο εμφανής καθώς το αρχικό μέγεθος των πρακτόρων μειώνεται.



Σχήμα 4.21: MRI για διάφορα αρχικά μεγέθη πρακτόρων

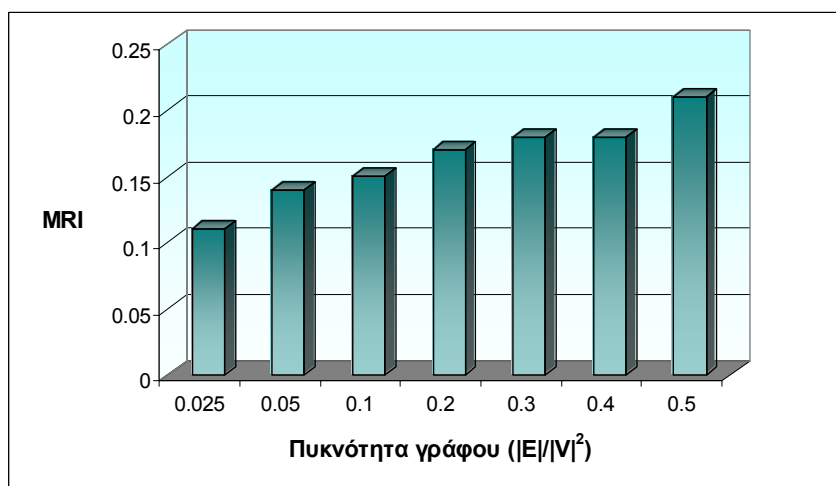
4.7.2.3 Επίδραση πυκνότητας δικτύου

Ο πειραματισμός με την πυκνότητα του δικτύου αποτελεί αντικείμενο της επόμενης δοκιμής. Υποθέτουμε ότι το μέγεθος πρακτόρων είναι ίσο με 20Kbytes, ενώ το μέγεθος της πληροφορίας που ανακτάται από κάθε κόμβο ισούται με 5Kbytes. Αυξάνουμε βαθμιαία την πυκνότητα του γράφου, αυξάνοντας την αναλογία των ακμών του σε σχέση με το τετράγωνο του αριθμού των κόμβων του ($|E|/|V|^2$). Η μεταβολή του χρόνου ολοκλήρωσης της αποστολής συναρτήσει του αριθμού πρακτόρων για πυκνότητα γράφου ίση με 0.025, 0.05, 0.1, 0.2, 0.3, και 0.4 απεικονίζεται στο Σχήμα 4.22.



Σχήμα 4.22: Χρόνος συναρτήσει αριθμού πρακτόρων για διάφορες πυκνότητες δικτύου

Είναι φανερό ότι καθώς η πυκνότητα του γράφου αυξάνει, ο χρόνος ολοκλήρωσης της συλλογής της πληροφορίας με συγκεκριμένο αριθμό πρακτόρων μειώνεται, επειδή υπάρχουν περισσότερες απευθείας συνδέσεις μεταξύ δυο οποιονδήποτε κόμβων, διευκολύνοντας τη γρηγορότερη εξυπηρέτηση των κόμβων με δεδομένα. Επιπλέον, το Σχήμα 4.23 δείχνει ότι ένας πυκνότερος γράφος επωφελείται περισσότερο από μια αύξηση στον πληθυσμό πρακτόρων σε σχέση με έναν αραιό.

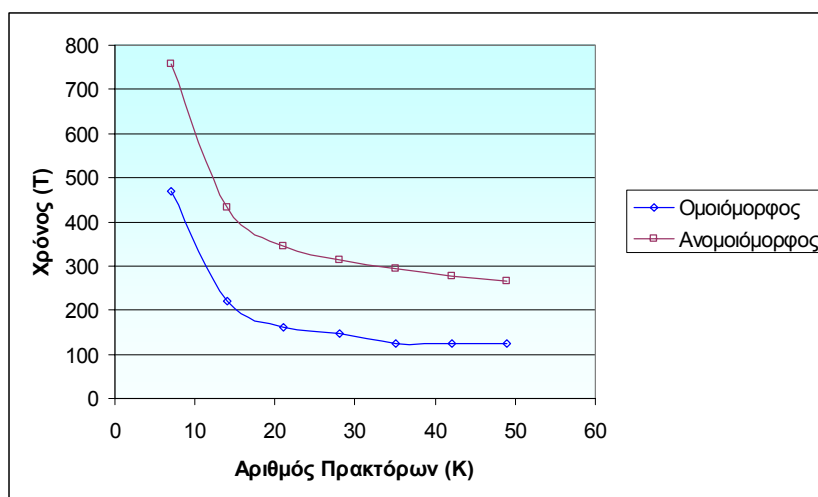


Σχήμα 4.23: MRI για διάφορες πυκνότητες δικτύου

4.7.2.4 Επίδραση συνδέσεων συμφόρησης στον κεντρικό αποστολέα

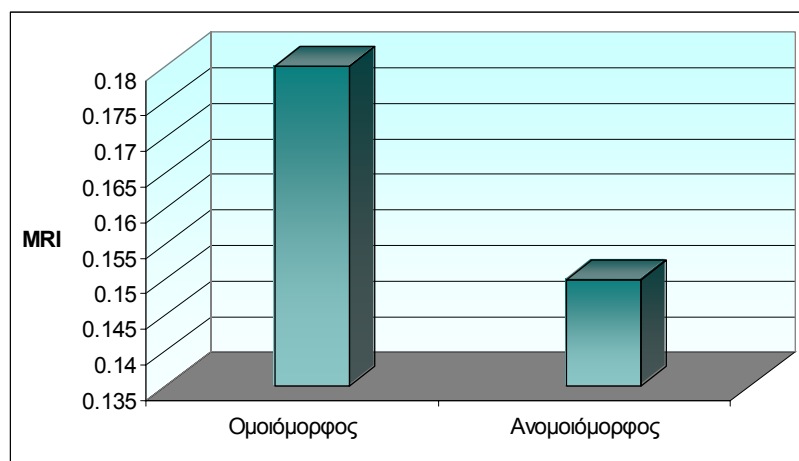
Ο σκοπός του επόμενου πειράματος είναι να μελετηθεί πώς η ταχύτητα συλλογής της πληροφορίας με πράκτορες επηρεάζεται από την ύπαρξη συνδέσεων συμφόρησης στον κόμβο του κεντρικού αποστολέα. Θέτουμε το αρχικό μέγεθος των πρακτόρων ίσο με 20Kbytes, και το μέγεθος της πληροφορίας που ανακτάται από κάθε κόμβο ίσο με 5Kbytes. Θα εξετάσουμε

έναν «ομοιόμορφο» γράφο, στον οποίο ο ρυθμός μετάδοσης κάθε γραμμής ανήκει στο διάστημα (1, 10) Mbps, καθώς και έναν «ανομοιόμορφο», στον οποίο οι συνδέσεις του κόμβου 0 χαρακτηρίζονται ως υπερφορτωμένες. Προσομοιώνουμε την τελευταία περίπτωση θέτοντας το ρυθμό μετάδοσης μιας σύνδεσης συμφόρησης σε μια τιμή του διαστήματος (1, 2) Mbps. Αξίζει να σημειωθεί ότι μια τέτοια κατάσταση δεν είναι καθόλου ασυνήθιστη, αφού ο κόμβος 0, ως κεντρικός αποστολέας των πρακτόρων και παραλήπτης της συλλεγόμενης πληροφορίας, είναι επιρρεπής σε καταστάσεις συμφόρησης εξαιτίας του μεγάλου όγκου πληροφορίας και του υπερβολικού φορτίου επεξεργασίας που προκαλείται από τις εργασίες συντονισμού και διαχείρισης. Το Σχήμα 4.24 δείχνει πώς ο χρόνος ολοκλήρωσης της αποστολής μεταβάλλεται με τον αριθμό πρακτόρων και για τους δύο τύπους γράφου.



Σχήμα 4.24: Χρόνος συναρτήσει αριθμού πρακτόρων για ομοιόμορφο και ανομοιόμορφο γράφο

Προφανώς, η ύπαρξη των συνδέσεων συμφόρησης καθυστερεί την αποστολή της ανάκτησης πληροφορίας. Ωστόσο, όπως δείχνει και το Σχήμα 4.25, ο ανομοιόμορφος γράφος δεν ευνοείται από μια αύξηση του αριθμού πρακτόρων στο βαθμό που ευνοείται ο ομοιόμορφος. Αυτό εξηγείται ως εξής: όταν υπάρχουν συνδέσεις συμφόρησης στον κόμβο του κεντρικού αποστολέα, όσο περισσότεροι είναι οι πράκτορες, τόσο πιο συχνά θα διασχίζονται οι συνδέσεις συμφόρησης. Αυτό όμως συνεπάγεται αυξημένη καθυστέρηση στην ολοκλήρωση της αποστολής. Αντίθετα, ένας περιορισμένος αριθμός πρακτόρων θα ανάγκαζε τους πράκτορες να διαπερνούν συχνότερα τις εσωτερικές συνδέσεις, οι οποίες είναι γρηγορότερες από αυτές που συνδέουν οποιοδήποτε κόμβο με τον αποστολέα, αφού σε αυτή την περίπτωση ένας υψηλότερος αριθμός κόμβων θα εξυπηρετούταν από κάθε πράκτορα.



Σχήμα 4.25: MRI για για ομοιόμορφο και ανομοιόμορφο γράφο

4.8 ΚΑΘΟΡΙΣΜΟΣ ΒΕΛΤΙΣΤΟΥ ΑΡΙΘΜΟΥ ΠΡΑΚΤΩΡΩΝ

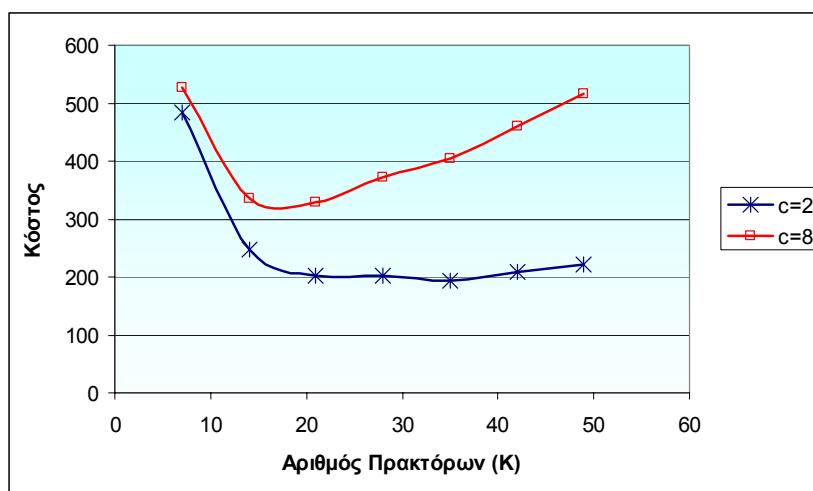
Στη συνέχεια πειραματιζόμαστε αποκλειστικά με την παραμετρική μεταβολή του αριθμού πρακτόρων προκειμένου να καθοριστεί η βέλτιστη τιμή του. Στη μέχρι τώρα μελέτη δεν έχουμε λάβει υπόψη την απώλεια πόρων που προκαλείται από τη χρήση πρακτόρων. Ο απώτερος σκοπός του προβλήματος είναι να επιτευχθεί μια «ικανοποιητική» για το χρήστη λύση καταναλώνοντας μια «λογική» ποσότητα πόρων του συστήματος. Εντούτοις, η επιθυμητή ισορροπία μεταξύ του χρόνου ολοκλήρωσης της αποστολής και των πόρων που καταναλώνονται αποτελεί κρίσιμη απαίτηση του προβλήματος. Στα πλαίσια αυτά, εκτός από το χρόνο ολοκλήρωσης της αποστολής, θεωρούμε και το κόστος ενοικίασης των πρακτόρων, ως ένα άλλο παράγοντα κόστους που αντιπροσωπεύει τα γενικά έξοδα του συστήματος και υποθέτουμε ότι αυξάνει γραμμικά με τον αριθμό πρακτόρων K , ως:

$$\text{renting_cost} = c \cdot K$$

όπου c είναι μια παράμετρος που παίρνει τιμές στο διάστημα $(1, 10)$. Στην πραγματικότητα, ο καθορισμός της τιμής του c ρυθμίζει την επιθυμητή ισορροπία μεταξύ του χρόνου ολοκλήρωσης της αποστολής και της ποσότητας πόρων που καταναλώνονται. Σύμφωνα και με τη φιλοσοφία της αγοράς, ένας χρήστης απολαμβάνει μια γρήγορη συναλλαγή εφ' όσον συμφωνεί να καταβάλει την τιμή της υψηλής ποιότητας υπηρεσίας που του παρέχεται, η οποία συνήθως σχετίζεται με τη δαπάνη αυξημένων πόρων του συστήματος. Ο καθορισμός της τιμής του c υπαγορεύεται συχνά από τα χαρακτηριστικά της εκάστοτε εφαρμογής. Για παράδειγμα, σε μια εφαρμογή, η οποία χαρακτηρίζεται ως ιδιαίτερα χρονικά κρίσιμη, απαιτείται να δοθεί προτεραιότητα στην ικανοποίηση του χρήστη, επιλέγοντας για το c μια τιμή που πλησιάζει το κατώτατο όριο, ενώ σε εφαρμογή που αναπτύσσεται σε σύστημα περιορισμένων διαθέσιμων πόρων, απαιτείται να επιλεγεί μια τιμή κοντά στο ανώτατο όριο, ώστε να «υποβοηθηθεί» το σύστημα. Μεταβάλλοντας λοιπόν παραμετρικά τον αριθμό των

πρακτόρων σε ένα περιορισμένο διάστημα, το οποίο προσδιορίζεται διαισθητικά από την τιμή που έχει επιλεγεί για το c , και επιδιώκοντας ελαχιστοποίηση της συνάρτησης κόστους, η οποία περιλαμβάνει και τους δύο όρους, προσδιορίζεται ο βέλτιστος αριθμός πρακτόρων.

Για παράδειγμα, χρησιμοποιώντας τα δεδομένα των προηγούμενων πειραμάτων και θέτοντας $c = 8$, ο βέλτιστος αριθμός πρακτόρων βρίσκεται ότι είναι 21, με χρόνο ολοκλήρωσης της αποστολής ίσο με 161 ms. Θέτοντας $c = 2$, ο βέλτιστος αριθμός πρακτόρων γίνεται 35, αλλά ο συνολικό χρόνος μειώνεται στα 125 ms (Σχήμα 4.26). Σημειώνεται ότι ο ελάχιστος χρόνος που μπορεί να επιτευχθεί είναι 123 ms, χρησιμοποιώντας 42 πράκτορες.



Σχήμα 4.26: Συνολικό κόστος συναρτήσει αριθμού πρακτόρων για διαφορετικές τιμές του c

4.9 ΠΑΡΑΤΗΡΗΣΕΙΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ

Σε αυτό το κεφάλαιο μελετήθηκε ένα πρόβλημα βελτιστοποίησης, το οποίο προκύπτει όταν ένας αριθμός κινητών πρακτόρων αποστέλλονται ταυτόχρονα από κεντρικό κόμβο προκειμένου να συλλέξουν πληροφορία κατανομημένη στο δίκτυο. Σκοπός είναι να προσδιοριστούν τα δρομολόγια των πρακτόρων που ελαχιστοποιούν το χρόνο ολοκλήρωσης της αποστολής, δεδομένου ότι ικανοποιείται ένα σύνολο περιορισμών. Το πρόβλημα αποδεικνύεται ότι είναι *NP-hard*, και για το λόγο αυτό προτείνονται τρεις ευριστικοί αλγόριθμοι για τη λύση του. Με χρήση προσομοίωσης πραγματοποιήθηκε σύγκριση των αλγορίθμων και προέκυψε ότι καθένας από αυτούς είναι πιο αποτελεσματικός σε συγκεκριμένο παραμετρικό χώρο, παράγοντας λύσεις που προσεγγίζουν τη βέλτιστη. Επιπλέον, η ανάλυση χρονικής πολυπλοκότητας έδειξε ότι οι αλγόριθμοι αυτοί είναι πολυωνυμικοί, ενώ οι χρόνοι εκτέλεσης που καταγράφηκαν στις δοκιμές επαλήθευσαν αυτή την αποδοτικότητα. Ένας αριθμός πειραμάτων εκτελέστηκε επίσης προκειμένου να μελετηθεί η επίδραση διαφόρων παραμέτρων του προβλήματος στη λύση του. Αξίζει να σημειωθεί ότι η

αποδοτική εφαρμογή των προτεινόμενων αλγορίθμων σε ένα σύστημα ανάκτησης κατανεμημένης πληροφορίας ευνοείται ιδιαίτερα από τις εξής συνθήκες: όταν απαιτείται να χρησιμοποιηθεί περιορισμένος αριθμός πρακτόρων, όταν οι πράκτορες έχουν σχετικά μεγάλο αρχικό μέγεθος (όπως για παράδειγμα όταν περιλαμβάνουν σύνθετο κώδικα για την εκτέλεση εξειδικευμένων λειτουργιών φιλτραρίσματος πληροφορίας), όταν μεγάλος όγκος πληροφορίας ανακτάται από κάθε κόμβο ή η κατάσταση του δικτύου είναι κακή. Ο λόγος είναι ότι σε αυτές τις περιπτώσεις ο χρόνος υπολογισμού των δρομολογίων θα είναι αμελητέος συγκριτικά με το χρόνο ολοκλήρωσης της αποστολής ανάκτησης της πληροφορίας.

Η εφαρμογή της προτεινόμενης μεθόδου επεκτείνεται και σε έναν μεταβλητό αριθμό πρακτόρων, παρέχοντας ευελιξία στο σύστημα. Σε αυτή την περίπτωση, η επιθυμητή ισορροπία μεταξύ του χρόνου ολοκλήρωσης της αποστολής και της ποσότητας των πόρων του συστήματος που καταναλώνονται αποτελεί κρίσιμη απαίτηση για τον καθορισμό του βέλτιστου αριθμού πρακτόρων. Αξίζει να σημειωθεί ότι η αποτίμηση ενός συνόλου παραμέτρων, όπως αυτές που εξετάστηκαν για τον τρόπο που επηρεάζουν τη λύση, μέσω μιας σύνθετης συνάρτησης με βάρη θα μπορούσε να αποδειχτεί ιδιαίτερα ενδιαφέρουσα για τον ακριβέστερο προσδιορισμό του βέλτιστου αριθμού πρακτόρων.

Η παρούσα μελέτη βασίστηκε στην υπόθεση ότι η πληροφορία για την κατάσταση του δικτύου είναι γνωστή από στατιστικά δεδομένα. Στα πραγματικά δίκτυα όμως, αυτή η πληροφορία υπόκειται σε διακυμάνσεις, και συνεπώς οι πράκτορες θα πρέπει να είναι σε θέση να αντιδρούν σε μη-αναμενόμενες συνθήκες ώστε τελικά να πραγματοποιούν τη λύση που είχε αρχικά υπολογιστεί. Η ενσωμάτωση μιας συνιστώσας, η οποία θα λαμβάνει υπόψη την αστάθεια του δικτύου, στο προτεινόμενο μοντέλο παρουσιάζει ιδιαίτερο ενδιαφέρον και για το λόγο αυτό θα μπορούσε να αποτελέσει αντικείμενο μελέτης σε μελλοντική εργασία.

4.10 ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ

- [1] B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko and D. Rus, “Mobile Agents in Distributed Information Retrieval”, Intelligent Information Agents, Publisher: Springer Verlag, pp.355–395, 1999.
- [2] F. Valera, A. Kaltabani, E. Vazquez, L. Bellido, I. Sygkouna, M. Strimpakou, M. Anagnostou, “Experiences with Middleware and Mobile Agents in an E-Commerce European Project”, International Journal of Internet and Enterprise Management, Vol.2, No.2, pp.163-177, Spring 2004.
- [3] C. Papadimitriou, K. Steiglitz, “Combinatorial Optimization”, Publisher: Dover Publications, ISBN: 0-486-40258-4, 1998.
- [4] D. Moizumi, “Mobile Agent Planning Problems”, PhD thesis, Dartmouth College, 1998.
- [5] A. Ohsuga, Y. Nagai, Y. Irie, M. Hattori, and S. Honiden, “Plangent: An Approach to Making Mobile Agents Intelligent, IEEE Internet Computing, Vol.1, No.4, pp.50-57, 1997.
- [6] J. Bredin, “Market-based Mobile-agent Planning”, Technical Report, Dartmouth College, 1999.
- [7] D.P. Bertsekas, “Network Optimization: Continuous and Discrete Models”, Publisher: Athena Scientific, ISBN: 1-886529-02-7, 1998.
- [8] K. Moizumi and G. Cybenko, “The Travelling Agent Problem”, Mathematics of Control, Signals and Systems, Vol.14, No.3, pp.213-232, 2001.
- [9] N. Christofides, A. Mingozzi, P. Toth, “The Vehicle Routing Problem”, Combinatorial Optimization, Publisher: John Wiley & Sons, ISBN: 0-471-99749-8, Chichester, 1979.
- [10] B.L. Golden, A.A. Assad, “Vehicle Routing: Methods and Studies”, Publisher: Elsevier, ISBN: 0-444-70407-8, 1991.
- [11] I. Sygkouna and M. Anagnostou, “Efficient Information Retrieval Using Mobile Agents”, Proceedings of the 4th ACM International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2005), Utrecht, Netherlands, July 2005.
- [12] I. Sygkouna and M. Anagnostou, “Speeding-up Information Retrieval with the Employment of a Multi-Agent System”, accepted for publication in the International Journal of Wireless and Mobile Computing, October 2004.
- [13] J.W. Baek, J.H. Yeo, G.T. Kim and H.Y. Yeom, “CEMAP: Cost-effective Mobile Agent Planning”, International Journal of Cooperative Information Systems, Vol.13, No.2, pp.159-181, 2004.

- [14] D.T. Pham, D. Karaboga, “Intelligent Optimisation Techniques”, Publisher: Springer, ISBN: 1-85233-028-7, 2000.
- [15] G. Clarke and J.W. Wright, “Scheduling of Vehicles from a Central Depot to a Number of Delivery Points”, Operations Research, Vol.12, pp.568-581, 1964.
- [16] R.H. Mole and S.R. Jameson, “A Sequential Route-Building Algorithm Employing a Generalized Savings Criterion”, Operational Research Quarterly, Vol.27, pp.503-511, 1976.
- [17] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, “Introduction to Algorithms”, Publisher: MIT Press, ISBN: 0-262-03141-8, 1999.
- [18] B.M. Waxman, “Routing of Multipoint Connections”, IEEE Journal on Selected Areas in Communications, Vol. 6, No. 9, pp. 1617-1622, December 1988.
- [19] I. Sygkouna and M. Anagnostou, “Route Planning for Agent-based Information Retrieval”, submitted to IEEE/ACM Transactions on Networking, September 2005.

ΚΕΦΑΛΑΙΟ 5

***Μηχανισμοί Αναζήτησης Πληροφορίας σε
Σύστημα Ομότιμων Μεσιτών***

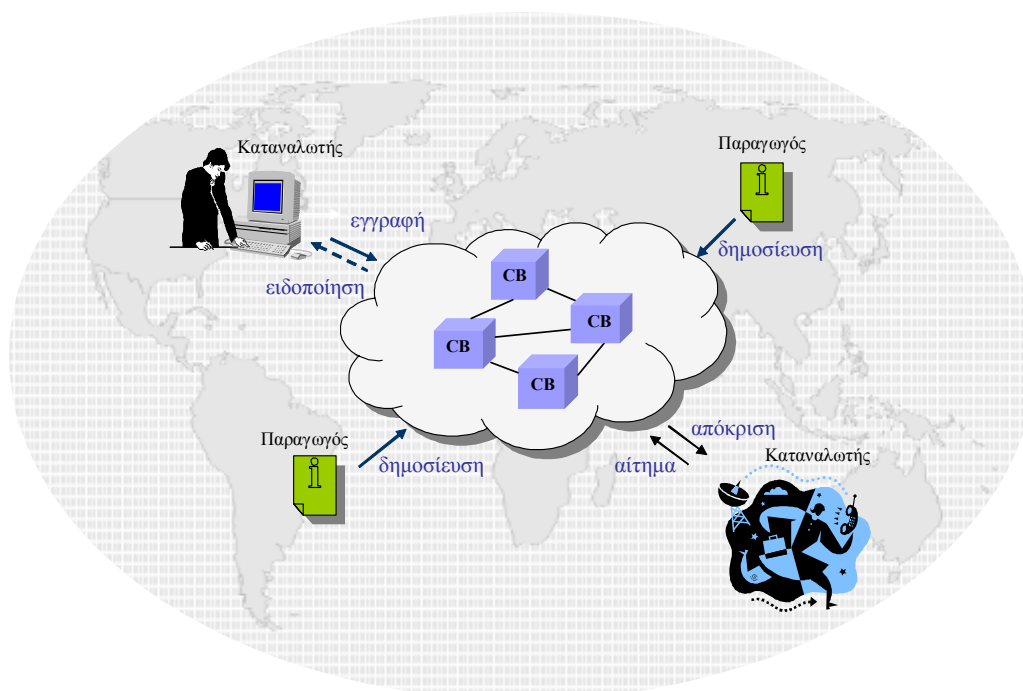
ΚΕΦΑΛΑΙΟ 5: ΜΗΧΑΝΙΣΜΟΙ ΑΝΑΖΗΤΗΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΣΕ ΣΥΣΤΗΜΑ ΟΜΟΤΙΜΩΝ ΜΕΣΙΤΩΝ

5.1 ΕΙΣΑΓΩΓΗ

Οι πρόσφατες τεχνολογικές εξελίξεις μαρτυρούν μια τεράστια πρόοδο στην ανάπτυξη εφαρμογών για περιβάλλοντα κινητής υπολογιστικής. Πολλοί ερευνητές προβλέπουν το μέλλον να κυριαρχείται από μια υπολογιστική και επικοινωνιακή υποδομή που θα έχει επίγνωση του περιβάλλοντος και θα επιτρέπει στους χρήστες να χρησιμοποιούν υπηρεσίες και να έχουν πρόσβαση σε δεδομένα οποτεδήποτε και οπουδήποτε. Η συνεχώς αυξανόμενη αλληλεπίδραση μεταξύ της επικοινωνίας και του υπολογισμού, η οποία τείνει να αλλάξει τον τρόπο προσπέλασης της πληροφορίας από αντιδραστικό τύπου «οποτεδήποτε και οπουδήποτε» σε προδραστικό τύπου «όλη την ώρα και παντού», υποκινεί ενδιαφέροντα αλγοριθμικά ζητήματα σχετικά με την πανταχού παρούσα πληροφορία.

Σε αυτό το πλαίσιο οι υπηρεσίες με επίγνωση του περιβάλλοντος (*Context-Aware Services - CASs*) απαιτούν μια συνεχή ροή πληροφορίας από και σχετικά με το περιβάλλον τους προκειμένου να είναι σε θέση να προσαρμόζονται σε αυτό δυναμικά. Το Σύστημα Διανομής Πληροφορίας Περιβάλλοντος (*Context Distribution System - CDS*), το οποίο περιγράψαμε στο κεφάλαιο 2, χρησιμοποιείται για τη συλλογή πληροφοριών από πολυάριθμες και διασκορπισμένες στο δίκτυο πηγές, και τη διανομή τους σε CASs σε διαφορετικά σημεία του δικτύου. Βάσει ενός κατάλληλου επιπέδου αφαίρεσης, το CDS περιλαμβάνει παραγωγούς πληροφορίας, καταναλωτές πληροφορίας και μεσίτες, με τους τελευταίους να λειτουργούν ως μεσολαβητές μεταξύ των παραγωγών και των καταναλωτών. Οι μεσίτες παρέχουν τις διεπαφές των πρωτοκόλλων Αιτήματος-Απόκρισης και Δημοσίευσης-Εγγραφής, απαλλάσσοντας τους παραγωγούς και τους καταναλωτές από τις λεπτομέρειες των υποκείμενων μηχανισμών διανομής της πληροφορίας.

Η ανάγκη παροχής CASs με καλές δυνατότητες κλιμάκωσης σε σχέση με τον αριθμό των κόμβων και των πηγών πληροφορίας υποδεικνύει την αρχιτεκτονική ομότιμων κόμβων ως την πιο κατάλληλη [1]. Σε αντίθεση με τις συγκεντρωτικές αρχιτεκτονικές, η πληροφορία δεν είναι συγκεντρωμένη σε έναν εξυπηρετητή, αλλά κατανέμεται μεταξύ των κόμβων-μελών. Κάθε μέλος μπορεί να διαθέσει πληροφορία για διανομή αλλά και να επικοινωνήσει με άλλα μέλη για να λάβει απομακρυσμένη πληροφορία. Στην περίπτωση που μελετάμε, το CDS βασίζεται στη συνεργασία πολλαπλών ομότιμων μεσιτών πληροφορίας (*Context Brokers - CBs*), όπως φαίνεται στο Σχήμα 5.1. Κάθε μεσίτης διαχειρίζεται τις τοπικές πηγές του, συνεργάζεται με άλλους μεσίτες, και επίσης παρέχει τα σημεία πρόσβασης των καταναλωτών και των παραγωγών για να αλληλεπιδράσουν με το σύστημα.



Σχήμα 5.1: P2P Σύστημα Διανομής Πληροφορίας

Ένα τέτοιο σύστημα αντιμετωπίζει την πρόκληση να εξασφαλίσει αποδοτική και κλιμακούμενη διανομή της πληροφορίας. Εξαιτίας της αποκεντρωμένης φύσης του μοντέλου P2P, η επίδοση εξαρτάται κυρίως από τρία είδη πόρων: επεξεργασίας, αποθήκευσης, και δικτύωσης. Ειδικότερα, η δικτυακή καθυστέρηση μπορεί να είναι σημαντική σε δίκτυα μεγάλης κλίμακας. Το εύρος ζώνης είναι επίσης σημαντικός παράγοντας επίδοσης όταν μεταδίδεται ένας μεγάλος αριθμός μηνυμάτων στο δίκτυο. Και οι δύο αυτοί παράγοντες περιορίζουν την επεκτασιμότητα του συστήματος. Συνεπώς, ζητήματα επίδοσης προσπαθούν να απαντήσουν σε ερωτήσεις, όπως πόσος χρόνος απαιτείται για να ανακτηθεί κάποια πληροφορία ή πόσο εύρος ζώνης καταναλώνει ένα αίτημα ανάκτησης πληροφορίας (είτε σύγχρονη ερώτηση του πρωτοκόλλου Αιτήματος-Απόκρισης, είτε συνδρομή για λήψη ασύγχρονης ειδοποίησης του πρωτοκόλλου Δημοσίευσης-Εγγραφής). Μεταξύ των δημοφιλέστερων προσεγγίσεων που εφαρμόζονται προκειμένου να βελτιωθεί η επίδοση του συστήματος είναι η *χρήση αντιγράφων (replication)*, η *προσωρινή φύλαξη αντικειμένων (caching)*, και η *ευφυής δρομολόγηση (intelligent routing)* [2]. Στο παρόν πλαίσιο οι δυο πρώτες προσεγγίσεις θεωρούνται λιγότερο κατάλληλες εξαιτίας της δυναμικής φύσης της πληροφορίας περιβάλλοντος, και συνεπώς επικεντρωνόμαστε σε **μηχανισμούς δρομολόγησης που εφαρμόζονται κάθε φορά που ζητείται κάποια πληροφορία και απαιτείται να πραγματοποιηθεί αναζήτηση σε πραγματικό χρόνο προκειμένου να εντοπιστεί.**

Στην απλούστερή του μορφή, το μοντέλο P2P χρησιμοποιεί μηχανισμούς προώθησης μηνυμάτων-αιτημάτων, γνωστούς με τον όρο *κανόνας διάδοσης αιτήματος (request*

propagation rule), για τον εντοπισμό της απαιτούμενης πληροφορίας, δεδομένου ότι δεν υπάρχει ένας κεντρικός εξυπηρετητής με σφαιρική γνώση για όλους τους κόμβους του δικτύου ή τις πληροφορίες που παρέχουν. Τα αιτήματα συχνά ακολουθούν πολλαπλά βήματα καθώς προωθούνται από τον έναν κόμβο στον άλλο, ενώ κάθε κόμβος αποκρίνεται σε αιτήματα για πληροφορίες που διατηρεί τοπικά. Το πρόβλημα όμως με αυτά τα συστήματα είναι ότι καταλήγουν να στέλνουν έναν μεγάλο αριθμό μηνυμάτων. Κάθε βήμα που εκτελεί ένα αίτημα συμβάλλει στην αύξηση του εύρους ζώνης που καταναλώνεται στις συνδέσεις του δικτύου, αλλά και του χρόνου που απαιτείται για τη λήψη των αποτελεσμάτων. Συνεπώς, η βελτίωση της επεκτασιμότητας των συστημάτων αυτών θεωρείται ιδιαίτερα σημαντική, ειδικότερα όταν δεν επιβαρύνει άλλα επιθυμητά χαρακτηριστικά, όπως εγγυήσεις απόδοσης ή ντετερμινισμού.

Προς αυτήν την κατεύθυνση, οι αλγοριθμικές καινοτομίες στην περιοχή της αναζήτησης πληροφοριών λαμβάνουν όλο και μεγαλύτερο ενδιαφέρον, με τους σχετικούς αλγορίθμους τυπικά να υλοποιούνται στη μορφή ενός πρωτοκόλλου του επιπέδου εφαρμογής. Εκτός από την αυξανόμενη διάσταση μεταξύ του όγκου πληροφοριών που γίνονται διαθέσιμες και της ικανότητας ανάκτησης πληροφοριών, επιπλέον οι CASs είναι λιγότερο πρόθυμες να περιμένουν έως ότου ανακτηθεί η απαιτούμενη πληροφορία σε σχέση με μια τυπική εφαρμογή αναζήτησης μουσικών αρχείων, για παράδειγμα. Οι υπάρχουσες τεχνικές όμως, καταλήγουν συνήθως σε αντίθετα άκρα όσον αφορά την κατανάλωση εύρους ζώνης ή αποθηκευτικού χώρου και του χρόνου απόκρισης. Για το λόγο αυτό προτείνουμε εναλλακτικές λύσεις στο πρόβλημα της αναζήτησης πληροφορίας με τον εξής σκοπό: να εγγυηθούμε το χαμηλότερο δυνατό χρόνο εντοπισμού οποιασδήποτε πληροφορίας ζητηθεί, περιορίζοντας ταυτόχρονα τις δαπάνες σε επικοινωνιακούς ή αποθηκευτικούς πόρους του συστήματος.

Η δομή του κεφαλαίου έχει ως εξής: Καταρχήν, παρουσιάζεται η σχετική με το πρόβλημα βιβλιογραφία. Στη συνέχεια περιγράφεται λεπτομερώς το υπό εξέταση πρόβλημα, μοντελοποιείται το σύστημα και ορίζονται τα κριτήρια επίδοσης που θα χρησιμοποιηθούν. Οι προτεινόμενοι μηχανισμοί για τη λύση του προβλήματος θα χωριστούν σε δυο κατηγορίες: τους *αντιδραστικούς* και τους *προδραστικούς*. Στα πλαίσια των αντιδραστικών μηχανισμών εξετάζονται ιδιότητες εντοπιότητας αναφοράς που είναι δυνατόν να παρουσιάζουν τα μοντέλα ζήτησης της πληροφορίας. Αφού οριστούν και μοντελοποιηθούν οι ιδιότητες, θα περιγραφούν οι προτεινόμενοι μηχανισμοί που τις αξιοποιούν, ενώ τα οφέλη από τη χρήση των μηχανισμών αυτών θα δειχτούν μέσα από πειράματα που πραγματοποιήθηκαν με χρήση ενός ανεξάρτητου εργαλείου προσομοίωσης που υλοποιήσαμε. Στα πλαίσια των προδραστικών μηχανισμών, θα μελετηθεί ένας μηχανισμός διανομής διαφημίσεων πηγών που βασίζεται στο μοντέλο μικρού-κόσμου, και θα παρουσιαστούν σχετικά πειραματικά

αποτελέσματα που προέκυψαν από τον προσομοιωτή. Το κεφάλαιο ολοκληρώνεται με ανάλυση των βασικών συμπερασμάτων που απορρέουν από τη μελέτη του προβλήματος.

5.2 ΕΠΙΣΚΟΠΗΣΗ ΣΧΕΤΙΚΗΣ ΒΙΒΛΙΟΓΡΑΦΙΑΣ

Το Πρόβλημα Εντοπισμού Πόρων (*Resource Location Problem*) αποτελεί ένα θεμελιώδες πρόβλημα στα κατανεμημένα συστήματα μεγάλης κλίμακας και μολονότι η εύρεση πόρων σε ένα δίκτυο υπολογιστών είναι ένα πρόβλημα πιθανώς τόσο παλιό όσο και η ίδια η κατανεμημένη υπολογιστική, οι διαφορετικές συνθήκες και απαιτήσεις των τρεχουσών εφαρμογών μεγάλης κλίμακας έχουν οδηγήσει σε μια πληθώρα νέων προσεγγίσεων. Το πρόβλημα της αναζήτησης πληροφοριών στα P2P δίκτυα μπορεί να αντιμετωπιστεί με διαφορετικούς τρόπους, από μηχανισμούς κεντρικών ευρετηρίων, όπως το σύστημα μερισμού αρχείων Napster [3], ως πλήρως αποκεντρωμένους μηχανισμούς που εξερευνούν το υποκείμενο δίκτυο χωρίς συνολική γνώση της δομής του. Στην περίπτωση του Napster, ένα ευρετήριο αρχείων διατηρείται σε κεντρικό σημείο, ενώ τα πραγματικά αρχεία κατανέμονται μεταξύ των κόμβων. Οι κόμβοι εγγράφονται στο κεντρικό ευρετήριο και αναφέρουν τα τοπικά αποθηκευμένα αρχεία τους. Οι μεμονωμένοι κόμβοι επικοινωνούν με τον κόμβο του κεντρικού ευρετηρίου για να εντοπίσουν τον κόμβο που φέρει το αρχείο που αναζητούν, και έπειτα πραγματοποιούν μια P2P αλληλεπίδραση για να ανακτήσουν το περιεχόμενο απευθείας από τον σχετικό κόμβο. Το βασικό πλεονέκτημα αυτού του μοντέλου είναι η ύπαρξη ενός κεντρικού ευρετηρίου, με το οποίο εντοπίζονται τα αρχεία γρήγορα και αποτελεσματικά. Δεδομένου όμως ότι ο εξυπηρετητής του ευρετηρίου θα πρέπει να διατηρεί πληροφορίες για όλους τους συμμετέχοντες του συστήματος, είναι πιθανό το μοντέλο αυτό να παρουσιάσει προβλήματα επεκτασιμότητας, επειδή θα απαιτεί μεγαλύτερο εξυπηρετητή όταν αυξάνεται ο αριθμός των αιτημάτων και μεγαλύτερη αποθηκευτική ικανότητα όταν αυξάνεται ο αριθμός των μελών. Επιπλέον, ο κεντρικός εξυπηρετητής αποτελεί μοναδικό σημείο πιθανής αποτυχίας και ορατό στόχο για επιθέσεις μέσω του δικτύου. Τέλος, επειδή ο κεντρικός εξυπηρετητής του ευρετηρίου ανανεώνεται συνήθως περιοδικά, είναι πιθανό ένας πελάτης να λαμβάνει παλιές πληροφορίες.

Το μεγαλύτερο ενδιαφέρον στη βιβλιογραφία εστιάζεται στα πρωτόκολλα αναζήτησης που εφαρμόζονται στο αποκεντρωμένο P2P υπόδειγμα επικοινωνίας: το όνομα ενός αντικειμένου που αναζητείται μπορεί να είναι γνωστό, αλλά ο κόμβος που φιλοξενεί το αντικείμενο να μην είναι γνωστός, μέχρι να διενεργηθεί αναζήτηση σε πραγματικό χρόνο. Οι σχετικοί αλγόριθμοι αναζήτησης προσπαθούν γενικά να βελτιστοποιήσουν το μονοπάτι που ακολουθεί το μήνυμα αναζήτησης καθώς μεταδίδεται από κόμβο σε κόμβο. Οι βασικοί στόχοι είναι να μειωθεί ο αριθμός των P2P βημάτων που απαιτούνται ώστε να εντοπιστεί η πληροφορία που αναζητείται, και επιπλέον να μειωθεί η πληροφορία δρομολόγησης που

απαιτείται να αποθηκεύεται από κάθε ομότιμο κόμβο. Η δυνατότητα καλής επεκτασιμότητας είναι επίσης ένα ιδιαίτερα επιθυμητό χαρακτηριστικό σε αυτά τα συστήματα. Δύο βασικά αποκεντρωμένα μοντέλα για τη μελέτη μηχανισμών P2P αναζήτησης έχουν προσδιοριστεί: το *μη-δομημένο (unstructured)* και το *δομημένο (structured)*.

5.2.1 ΑΠΟΚΕΝΤΡΩΜΕΝΟ ΚΑΙ ΜΗ-ΔΟΜΗΜΕΝΟ ΜΟΝΤΕΛΟ

Στα συστήματα αυτά δεν υπάρχει η έννοια του κεντρικού ευρετηρίου, ούτε κανένας ακριβής έλεγχος της τοπολογίας του δικτύου ή της θέσης των αντικειμένων. Η τοπολογία του δικτύου έχει ορισμένες ιδιότητες, αλλά η τοποθέτηση των αντικειμένων στους κόμβους δεν βασίζεται σε γνώση της τοπολογίας. Αντί να εξετάζει ένα κεντρικό ευρετήριο για να εντοπίσει τη θέση των αντικειμένων, ένας πελάτης που αναζητά συγκεκριμένη πληροφορία σε ένα P2P δίκτυο ερευνά τις διεσπαρμένες συλλογές πληροφορίας που βρίσκονται στους πολυάριθμους κόμβους, με διαδοχική προώθηση του αιτήματός του στους γειτονικούς του κόμβους, μέχρι να βρεθεί αυτό που αναζητά. Σε ένα μη-δομημένο μοντέλο υπάρχουν πολλοί βαθμοί ελευθερίας στην επιλογή του κανόνα προώθησης. Διάφορες στρατηγικές μπορούν να υιοθετηθούν, οι οποίες χαρακτηρίζονται από τον αριθμό των γειτονικών κόμβων στους οποίους προωθείται ένα αίτημα, αλλά και από τον τρόπο με τον οποίο επιλέγονται αυτοί οι γειτονικοί κόμβοι. Κάθε ομότιμος κόμβος μπορεί να διατηρεί ένα μερικό ευρετήριο των κόμβων-μελών που γνωρίζει. Το ψάξιμο για ένα συγκεκριμένο αντικείμενο μπορεί να αρχίσει από αυτό το ευρετήριο και να επεκταθεί σε ευρετήρια άλλων κόμβων. Είναι γεγονός ότι ένα τέτοιο σύστημα είναι λιγότερο αποδοτικό όσον αφορά την αναζήτηση, αλλά είναι λιγότερο ευαίσθητο στην αποτυχία ενός μεμονωμένου κόμβου, και επίσης δεν επιβαρύνει υπερβολικά ένα συγκεκριμένο κόμβο. Επιπλέον, υποστηρίζει αναζήτηση βάσει αναγνωριστικού ή περιεχομένου. Μερικά αντιπροσωπευτικά συστήματα που συμμορφώνονται με αυτό το μοντέλο περιγράφονται στη συνέχεια.

- Το σύστημα *Gnutella* [4] είναι ένα παγκόσμια κατανεμημένο σύστημα πληροφοριών, το οποίο χρησιμοποιεί τον *αλγόριθμο της πλημμύρας (flooding algorithm)* για την υλοποίηση του κανόνα διάδοσης αιτημάτων αναζήτησης: κάθε μήνυμα αιτήματος από έναν κόμβο αποστέλλεται σε όλους τους γείτονές του (δηλαδή στους άμεσα συνδεδεμένους κόμβους), καθένας από τους οποίους το προωθεί με τη σειρά του σε όλους τους γείτονές του κ.ο.κ., μέχρι το αίτημα να απαντηθεί. Το μήνυμα της απάντησης οδηγείται κατά μήκος της διαδρομής από την οποία έφθασε το μήνυμα του αιτήματος, με τη βοήθεια δυναμικά ενημερωμένων πινάκων δρομολόγησης που διατηρούνται σε κάθε ομότιμο κόμβο. Για να περιοριστεί η εκθετική διάδοση των μηνυμάτων μέσω του δικτύου, κάθε αίτημα έχει έναν μετρητή του αριθμού των βημάτων που πραγματοποιεί, ο οποίος συμβολίζεται με TTL (Time-To-Live). Συνήθως, το TTL τίθεται από το σύστημα σε μια μέγιστη αρχική τιμή, και η

τιμή αυτή μειώνεται από κάθε κόμβο που λαμβάνει το μήνυμα, πριν το αναμεταδώσει. Μόλις η τιμή του TTL γίνει μηδέν, το μήνυμα απορρίπτεται. Εάν η ποιότητα των αποτελεσμάτων σε ένα σύστημα μετρούνταν από τον αριθμό των αποτελεσμάτων και όχι απλά από την ικανοποίηση του αιτήματος, τότε η τεχνική της πλημμύρας θα ήταν ιδανική αφού στέλνει το αίτημα σε κάθε δυνατό κόμβο (δηλαδή σε όλους τους κόμβους σε απόσταση, μετρούμενη σε αριθμό βημάτων, ίση με TTL), όσο το δυνατόν γρηγορότερα. Εντούτοις, εάν η ικανοποίηση ήταν το κριτήριο επιλογής, τότε είναι φανερό ότι η τεχνική της πλημμύρας σπαταλά μεγάλες ποσότητες εύρους ζώνης και ισχύος επεξεργασίας γιατί τα περισσότερα αιτήματα συνήθως ικανοποιούνται από τις αποκρίσεις σχετικά λίγων κόμβων. Στο άρθρο [5], οι συγγραφείς ανέλυσαν την τοπολογία του δικτύου Gnutella και παρατήρησαν ισχυρά χαρακτηριστικά *μικρού-κόσμου* (*small-world*) και κατανομή *power-law* στον βαθμό των κόμβων, θέτοντας τη βάση για το σχεδιασμό νέων αλγορίθμων που θα μπορούσαν να εκμεταλλευθούν αυτές τις ιδιότητες. Ο Adamic [6] πρότεινε έναν αλγόριθμο αναζήτησης για δίκτυα με *power-law* κατανομή στο βαθμό των κόμβων τους, όπως το Gnutella, τα οποία αποτελούνται από λίγους κόμβους υψηλού βαθμού και πολλούς χαμηλού. Εάν υποθεθεί ότι δεν υπάρχει γενική πληροφορία για τη θέση του στόχου και ως εκ τούτου δεν είναι δυνατό να καθοριστεί εάν ένα βήμα αποτελεί μια κίνηση προς ή μακριά από το στόχο, ο αλγόριθμος που προτείνει, απαιτεί κατά την επίσκεψη των κόμβων, να προηγούνται οι κόμβοι με υψηλό αριθμό συνδέσεων. Αυτή η προσέγγιση σιωπηρά υποθέτει ότι οι κόμβοι με τον μεγαλύτερο αριθμό συνδέσεων είναι πιο πιθανό να απαντούν στις ερωτήσεις. Διαφορετικά, κινδυνεύει να «πλημμυρίσει» το δίκτυο με μηνύματα.

- Ο Lv [7] μελέτησε δύο εναλλακτικές του αλγορίθμου της πλημμύρας, σε μια προσπάθεια να βελτιώσει την επεκτασιμότητά του: τον *διευρυνόμενο δακτύλιο* (*expanding ring*) και τον *τυχαίο περίπατο* (*random walk*). Με την τεχνική του διευρυνόμενου δακτυλίου προσπαθεί να λύσει το πρόβλημα της επιλογής κατάλληλης τιμής για το TTL που υπάρχει στον αλγόριθμο της πλημμύρας. Μέσω προσομοιώσεων έχει δείχτει ότι διαφορετικά TTLs απαιτούνται για διαφορετικές τοπολογίες δικτύων, και συνεπώς η επιλογή κατάλληλης τιμής του TTL δεν είναι τετριμμένη. Ο διευρυνόμενος δακτύλιος εφαρμόζει επαναλαμβανόμενα τον αλγόριθμο της πλημμύρας με διαδοχικά αυξανόμενη τιμή του TTL. Ειδικότερα, ένας κόμβος εφαρμόζει αρχικά τον αλγόριθμο της πλημμύρας με μικρό TTL, και περιμένει να δει εάν η αναζήτηση είναι επιτυχής. Εάν είναι, ο κόμβος σταματάει. Διαφορετικά, αυξάνει το TTL και εφαρμόζει άλλη μια φορά τον αλγόριθμο της πλημμύρας. Η διαδικασία αυτή επαναλαμβάνεται έως ότου βρεθεί το αντικείμενο. Συγκριτικά με τον αλγόριθμο της πλημμύρας, ο διευρυνόμενος δακτύλιος επιτυγχάνει αποταμίευση στον αριθμό των μηνυμάτων που στέλνονται αλλά αυξάνει την καθυστέρηση για τον εντοπισμό ενός αντικειμένου. Για την αντιμετώπιση του προβλήματος μετάδοσης πολλαπλών αντιγράφων

ενός μηνύματος σε κάποιο κόμβο από τους πολλαπλούς γείτονές του, το οποίο εμφανίζεται στον αλγόριθμο της πλημμύρας, προτείνεται ο μηχανισμός του τυχαίου περιπάτου. Τα αντίγραφα αυτά προκαλούν καθαρή σπατάλη εύρους ζώνης χωρίς να συμβάλλουν σε αύξηση της πιθανότητας εντοπισμού ενός αντικειμένου. Ο τυχαίος περίπατος διαβιβάζει σε κάθε βήμα το αίτημα σε έναν γείτονα που επιλέγεται τυχαία, έως ότου βρεθεί το αντικείμενο. Οι συγγραφείς πειραματίστηκαν με τη χρήση πολλαπλών τυχαίων περιπάτων, δηλαδή αντί ένας κόμβος να στέλνει ακριβώς ένα μήνυμα με το αίτημα, να στέλνει πολλαπλά μηνύματα, και κάθε μήνυμα να πραγματοποιεί τον δικό του τυχαίο περίπατο. Με περισσότερους περιπατητές, τα αντικείμενα μπορούν να βρεθούν γρηγορότερα, αλλά υπό το κόστος της αύξησης της κυκλοφορίας στο δίκτυο. Οι προσομοιώσεις έδειξαν ότι ο «έλεγχος» με τον αρχικό κόμβο είναι η κατάλληλη προσέγγιση για τη λήξη των αναζητήσεων στους τυχαίους περιπάτους. Ο «έλεγχος» υπονοεί ότι ένας περιπατητής περιοδικά επικοινωνεί με τον αιτούντα κόμβο και ελέγχει εάν έχει βρεθεί το αντικείμενο πριν αποστείλει το μήνυμα στον επόμενο κόμβο. Τα αποτελέσματα έδειξαν ότι, έναντι του αλγορίθμου της πλημμύρας, ο τυχαίος περίπατος 32-περιπατητών μειώνει τον αριθμό των μηνυμάτων κατά προσέγγιση δύο τάξεις μεγέθους στα δίκτυα που εξετάστηκαν, υπό το κόστος όμως μιας μικρής αύξησης στον αριθμό των βημάτων που απαιτούνται για τον εντοπισμό του αντικειμένου, αλλά υπερτερεί του διευρυνόμενου δακτυλίου.

- Οι Yang και Garcia-Molina [8] πρότειναν τρεις τεχνικές αναζήτησης σε P2P συστήματα. Η πρώτη από αυτές, η *επαναληπτική εμβάθυνση (iterative deepening)*, είναι παρόμοια με την τεχνική του διευρυνόμενου δακτυλίου: μειώνει τον αριθμό των κόμβων που ρωτούνται στέλνοντας επαναληπτικά το αίτημα σε όλο και περισσότερους κόμβους, μέχρι αυτό να απαντηθεί. Αυτή η τεχνική προτείνεται σε συστήματα όπου η ικανοποίηση είναι το κριτήριο επιλογής. Εάν η ελαχιστοποίηση του χρόνου απόκρισης είναι σημαντική για μια εφαρμογή τότε η τεχνική αυτή είναι λιγότερο κατάλληλη εξαιτίας του χρόνου που απαιτούν οι πολλαπλές επαναλήψεις. Στη συνέχεια εξετάζεται η τεχνική της *κατευθυνόμενης αναζήτησης ως προς το εύρος (Directed Breadth-First Search - DBFS)*, σύμφωνα με την οποία ο αρχικός κόμβος ρωτά ένα περιορισμένο σύνολο γειτονικών κόμβων, οι οποίοι επιλέγονται έξυπνα ώστε να μεγιστοποιηθεί η πιθανότητα να απαντηθεί το αίτημα. Οι γείτονες που λαμβάνουν το αίτημα συνεχίζουν την προώθηση του αιτήματος σε όλους τους γείτονες, όπως με τον αλγόριθμο της πλημμύρας. Προκειμένου να επιλέγονται έξυπνα οι γειτονικοί κόμβοι στους οποίους στέλνεται αρχικά το αίτημα, κάθε κόμβος διατηρεί στατιστικές πληροφορίες σχετικές με τους γείτονές του. Αυτές οι στατιστικές μπορεί να είναι πολύ απλές, όπως ο αριθμός των αποτελεσμάτων που έχουν ληφθεί μέσω ενός γείτονα σε προηγούμενα αιτήματα ή η καθυστέρηση της σύνδεσης με έναν γείτονα. Τα πειραματικά αποτελέσματα έδειξαν ότι ο χρόνος απόκρισης που πετυχαίνει ο DBFS είναι ελαφρώς

καλύτερος από το χρόνο απόκρισης της επαναληπτικής εμβάθυνσης. «Θυσιάζοντας» όμως χρόνο απόκρισης, η επαναληπτική εμβάθυνση πετυχαίνει χαμηλότερο κόστος από τον DBFS. Τέλος, με την τεχνική των τοπικών ευρετηρίων (*local indices*), ένας κόμβος διατηρεί ευρετήριο με τα στοιχεία των κόμβων που βρίσκονται σε μέγιστη απόσταση (μετρούμενη σε βήματα) από αυτόν ίση με r , όπου r είναι μια γενική μεταβλητή του συστήματος, γνωστή ως ακτίνα του ευρετηρίου. Όταν ένας κόμβος λαμβάνει ένα αίτημα, μπορεί να το επεξεργαστεί εκ μέρους κάθε άλλου κόμβου που βρίσκεται σε απόσταση r από τον ίδιο. Κατ' αυτόν τον τρόπο, οι συλλογές πολλών κόμβων μπορούν να ερευνηθούν με την επεξεργασία του αιτήματος από λίγους κόμβους, πετυχαίνοντας ένα υψηλό ποσοστό ικανοποίησης και αριθμό αποτελεσμάτων, και ταυτόχρονα διατηρώντας τις δαπάνες χαμηλές. Ένας κανόνας προσδιορίζει το βάθος στο οποίο ένα αίτημα πρέπει να υποβάλλεται σε επεξεργασία. Όλοι οι κόμβοι που βρίσκονται σε βάθη που δεν αναφέρονται στον κανόνα απλά προωθούν το αίτημα στους γειτονικούς τους κόμβους. Για τη δημιουργία και τη διατήρηση των τοπικών ευρετηρίων σε κάθε κόμβο απαιτούνται πρόσθετα βήματα κάθε φορά που ένας κόμβος εισέρχεται στο δίκτυο ή εξέρχεται από αυτό και κάθε φορά που ένας χρήστης ανανεώνει την τοπική συλλογή του. Σε κάθε περίπτωση, το κατάλληλο μήνυμα στέλνεται με TTL ίσο με r . Τα πειραματικά αποτελέσματα έδειξαν ότι η επιλογή κατάλληλης τιμής για το r ανά περίπτωση είναι κρίσιμη και καθορίζει την αποδοτικότητα του αλγορίθμου.

- Η αποτελεσματικότητα του τυχαίου περιπάτου μελετήθηκε επίσης από τον Gkantsidi στο [9]. Συγκεκριμένα, έχοντας υποθέσει ότι πολλαπλά αντίγραφα ενός αντικειμένου είναι κατανομημένα σε ένα P2P δίκτυο, επιχείρησε σύγκριση μεταξύ του τυχαίου περιπάτου και του αλγορίθμου της πλημμύρας. Θεωρώντας ως βασικό κριτήριο επίδοσης το μέσο αριθμό αντιγράφων που εντοπίζονται κατά την αναζήτηση ενός αντικειμένου και υποθέτοντας ότι μεταδίδεται ο ίδιος αριθμός μηνυμάτων και στις δυο περιπτώσεις, ο τυχαίος περίπατος με έναν περιπατητή βρέθηκε ότι έχει παρόμοια επίδοση με τον αλγόριθμο της πλημμύρας στην περίπτωση επίπεδων τακτικών γράφων (δηλαδή γράφων με ισόβαθμους κόμβους). Αυξάνοντας τον αριθμό των περιπατητών αλλά διατηρώντας σταθερό τον αριθμό των μηνυμάτων, τα αποτελέσματα παρέμειναν σχεδόν ίδια. Στην περίπτωση όμως γράφων με ομαδοποίηση κόμβων, όπου η τοπολογία ιεραρχείται σε δυο επίπεδα, ένα χαμηλότερο, το οποίο χαρακτηρίζεται από τους ομαδοποιημένους κόμβους και ένα υψηλότερο, το οποίο συνδέει αντιπροσώπους από κάθε ομάδα, βρέθηκαν διαφορετικά αποτελέσματα. Ειδικότερα, ο τυχαίος περιπατητής υπερέχει του αλγορίθμου της πλημμύρας, σημειώνοντας υψηλότερο ποσοστό επιτυχίας αλλά και υψηλότερο αριθμό αποτελεσμάτων ανά αναζήτηση.

- Στο [10] προτείνονται υβριδικά σχήματα, τα οποία συνδυάζουν αλγορίθμους πλημμύρας με τυχαίους περιπάτους, με σκοπό να υποσκελίσουν τις αδυναμίες των

μεμονωμένων αλγορίθμων. Με βάση την παρατήρηση ότι στην περίπτωση γράφων με μεγάλες διαφορές στους βαθμούς των κόμβων η απλή πλημμύρα παρουσιάζει κακή επίδοση εξαιτίας των πολλαπλών μηνυμάτων που στέλνει, προτείνεται ο αλγόριθμος της κανονικοποιημένης πλημμύρας (*normalized flooding*). Ο αλγόριθμος αυτός παρουσιάζει βελτιωμένη επίδοση (ως προς τον αριθμό μηνυμάτων που μεταδίδονται) σε σχέση με την απλή πλημμύρα στην περίπτωση αραιών γράφων με λίγους κόμβους υψηλού βαθμού, και συνεπώς έχει καλύτερες δυνατότητες κλιμάκωσης. Σύμφωνα με αυτόν, ένα κόμβος μικρού βαθμού προωθεί ένα λαμβανόμενο μήνυμα σε όλους τους γείτονές του, ενώ ένα κόμβος υψηλού βαθμού το προωθεί σε ένα μικρό υποσύνολο (σταθερού μεγέθους) των γειτόνων του, οι οποίοι επιλέγονται τυχαία. Τα αποτελέσματα έδειξαν επιπλέον ότι σε γράφους υψηλών βαθμών ο αλγόριθμος κανονικοποιημένης πλημμύρας είναι προτιμότερος από τον τυχαίο περιπατητή, ενώ σε τοπολογίες με ομαδοποίηση κόμβων υπερέχει ελαφρώς ο δεύτερος. Δεύτερον, προτείνεται ο *τυχαίος περίπατος με τοπική πλημμύρα* (*random walk with local flooding*), ο οποίος βελτιώνει τον απλό τυχαίο περίπατο γιατί μειώνει το χρόνο που απαιτείται για τη λήψη ενός συγκεκριμένου αριθμού απαντήσεων. Για την ακρίβεια, εκτελεί συντομότερους περιπάτους, αλλά σε κάθε βήμα εφαρμόζει μια «ρηχή» πλημμύρα (π.χ. με TTL=1). Σε τακτικούς γράφους, η αποταμίευση χρόνου που πετυχαίνει σε σχέση με τον απλό τυχαίο περίπατο είναι σταθερή για σταθερό βάθος πλημμύρας, ενώ η αποταμίευση είναι πολύ μεγαλύτερη σε τυχαίους γράφους με μερικούς κόμβους υψηλού βαθμού. Τέλος, προτείνεται ένας γενικός μηχανισμός αναζήτησης, όπου η απλή πλημμύρα, ο απλός τυχαίος περίπατος και ο τυχαίος περίπατος με τοπική πλημμύρα αποτελούν ειδικές περιπτώσεις. Πιο συγκεκριμένα, υποθέτουμε ότι ένας κόμβος ξεκινάει μια αναζήτηση εάν του ανατεθεί ένας «προϋπολογισμός», ο οποίος αποτελεί άνω όριο στον αριθμό των μηνυμάτων που θα μεταδοθούν συνολικά κατά την αναζήτηση. Ο κόμβος αυτός μοιράζει το ποσό αυτό σε διάφορα μέρη και στέλνει κάθε μέρος σε ένα γείτονα. Ο γείτονας θα μειώσει το ποσό που έλαβε κατά ένα και θα συνεχίσει την ίδια διαδικασία εάν το προκύπτον ποσό είναι μεγαλύτερο από μηδέν. Ο μηχανισμός αυτός έχει νόημα όταν για παράδειγμα η *κρισιμότητα των ακμών* είναι γνωστή, και είναι χρήσιμος κυρίως σε τοπολογίες που παρουσιάζουν ομαδοποίηση κόμβων: μια ακμή μεγάλης κρισιμότητας θα «μετατοπίσει» μεγάλο μέρος του προϋπολογισμού προς ένα σύνολο ομαδοποιημένων κόμβων ώστε να καλυφθεί η πλειοψηφία αυτών. Σε τακτικούς γράφους, η επίδοσή του είναι παρόμοια με αυτή της απλής πλημμύρας, ενώ σε άλλα είδη γράφων παίζει σημαντικό ρόλο στην επιτυχία του αλγορίθμου η κρισιμότητα που έχει επιλεγεί για την κάθε ακμή.

- Οι συγγραφείς του [11] ανέπτυξαν ένα πρωτόκολλο για το *Πιθανολογικό Πρόβλημα Εντοπισμού Πόρων* (*Probabilistic Resource Location Problem*), το οποίο διαπραγματεύεται την απόδοση και την επεκτασιμότητα του πρωτοκόλλου με την πιθανότητα εντοπισμού ενός

πόρου. Ο στόχος είναι να επιτευχθεί μια πιθανότητα κοντά στη μονάδα με κόστος πολύ χαμηλότερο από αυτό της ντετερμινιστικής περίπτωσης. Θεωρούν ότι κάθε κόμβος διατηρεί έναν τοπικό κατάλογο (Local Directory - LD) που υποδεικνύει τους τοπικά διαθέσιμους πόρους, και έναν κατάλογο κρύπτης (Directory Cache - DC) που υποδεικνύει τη θέση μερικών από τους πόρους που παρέχονται από άλλους κόμβους. Κάθε πόρος φέρει ένα μοναδικό, γενικό προσδιοριστικό, το οποίο είναι ανεξάρτητο από τη θέση του πόρου (Globally Unique Identifier - GUID). Μια καταχώρηση στο DC είναι ένα ζευγάρι (id , loc), όπου id είναι το GUID του πόρου και loc είναι η δικτυακή διεύθυνση του κόμβου που διαθέτει το συγκεκριμένο πόρο. Οι καταχωρήσεις των καταλόγων μπορούν μετακινούνται από τα LDs σε DCs άλλων κόμβους ώστε να προσαρμόζονται στα μοντέλα ζήτησης της πληροφορίας. Επιπλέον, κάθε κόμβος s έχει μια τοπική γειτονιά $N(s)$, η οποία ορίζεται ως το σύνολο των κόμβων που βρίσκονται κοντά του – σε απόσταση ενός βήματος από τον s ή στο ίδιο τοπικό δίκτυο με τον s . Το πρωτόκολλο λειτουργεί ως εξής: Όταν ένας κόμβος λάβει ένα αίτημα αναζήτησης ψάχνει αρχικά στον κατάλογο LD και στη συνέχεια στον DC. Εάν βρει τον πόρο στον LD, στέλνει ένα μήνυμα εντοπισμού του πόρου κατά μήκος της διαδρομής που είχε ακολουθήσει το μήνυμα του αιτήματος αναζήτησης μέχρι να εντοπίσει τον πόρο. Το μήνυμα εντοπισμού ενημερώνει τον DC σε κάθε κόμβο που επισκέπτεται. Εάν βρει τον πόρο στον DC, στέλνει το μήνυμα του αιτήματος αναζήτησης στον κόμβο που υποδεικνύει ο DC. Αυτός ο κόμβος όμως μπορεί να μην έχει πλέον τον πόρο, και έτσι η αναζήτηση συνεχίζεται με τον ίδιο τρόπο. Εάν ένας κόμβος δεν βρει τον πόρο ούτε στον LD ούτε στον DC, θα στείλει το αίτημα σε κάθε κόμβο στη γειτονιά του με μια πιθανότητα p , η οποία ονομάζεται πιθανότητα ευρείας εκπομπής. Αυτή η πιθανότητα μεταβάλλεται με το μήκος της διαδρομής που έχει ακολουθήσει το αίτημα. Ωστόσο το πρωτόκολλο δεν εξετάζει πολιτικές αντικατάστασης καταχωρήσεων στον DC. Η αποδοτικότητα του πρωτοκόλλου μελετήθηκε με χρήση ενός απλοποιημένου αναλυτικού μοντέλου.

- Στο [12] το *Πιθανολογικό Πρόβλημα Εντοπισμού Πόρων* διατυπώνεται με τη χρήση *πιθανολογικών ερωτήσεων*. Μια πιθανολογική ερώτηση φέρει ένα κατηγορημα (predicate) ως παράμετρο και επιστρέφει ένα σύνολο κόμβων στους οποίους θεωρείται ότι το κατηγορημα ισχύει. Η ερώτηση είναι πιθανολογική επειδή υπάρχουν μερικές πιθανότητες το κατηγορημα να μην ισχύει σε όλους, ή ακόμα και σε κανέναν από τους κόμβους που έχουν επιστραφεί. Για την υλοποίηση των πιθανολογικών ερωτήσεων, οι συγγραφείς εισήγαγαν τον *Psearch*, έναν *επιδημικό αλγόριθμο* που χρησιμοποιεί βασικές έννοιες της Μπεϋζιανής στατιστικής λογικής και εκτελεί αναζήτηση με βάση το περιεχόμενο. Προσπαθεί να εντοπίσει πληροφορίες βάσει των μοντέλων χρήσης της πληροφορίας: εάν, για παράδειγμα, συγκεκριμένη πληροφορία εντοπίζεται σε κάποιο μηχάνημα, θεωρείται πιθανό αυτό το μηχάνημα να έχει αποθηκευμένες και άλλες ενδιαφέρουσες πληροφορίες του ίδιου τύπου. Εν

συντομία, οι κόμβοι ανταλλάσσουν μεταξύ τους πληροφορίες σχετικές με την εκτέλεση προηγούμενων ερωτήσεων και χρησιμοποιούν αυτές τις πληροφορίες για να προωθούν τις ερωτήσεις στους κόμβους όπου είναι πιο πιθανό να ισχύει το αιτούμενο κατηγορημα. Οι κόμβοι χρησιμοποιούν έναν μηχανισμό κουτσομπολιού (*gossip mechanism*) για να ανταλλάσσουν αυτούς τους πίνακες, και ανανεώνουν τις καταχωρήσεις των πινάκων σύμφωνα με τις αιτιολογικές σχέσεις μεταξύ των καταχωρήσεων και της Μπεϋζιανής στατιστικής λογικής. Η αξιολόγηση του αλγορίθμου αναζήτησης έγινε με τη χρήση ενός απλού αναλυτικού μοντέλου και ενός μοντέλου προσομοίωσης. Τα αποτελέσματα έδειξαν ότι μόνο εάν μερικοί από τους κόμβους περιέχουν πολλούς πόρους, τα αποτελέσματα που παράγονται από την αναζήτηση Psearch μπορούν να είναι αρκετά ακριβή.

5.2.2 ΑΠΟΚΕΝΤΡΩΜΕΝΟ ΚΑΙ ΔΟΜΗΜΕΝΟ ΜΟΝΤΕΛΟ

Τα συστήματα που υπακούουν σε αυτό το μοντέλο δεν έχουν κεντρικό ευρετήριο αλλά διατηρούν ένα σημαντικό ποσοστό δομής. Με τον όρο «δομή» εννοούμε ότι η P2P τοπολογία (δηλαδή το σύνολο συνδέσεων μεταξύ των P2P μελών) ελέγχεται αυστηρά και τα αντικείμενα δεν τοποθετούνται σε τυχαίους κόμβους αλλά σε καθορισμένες θέσεις, καθιστώντας ευκολότερη την ικανοποίηση των αιτημάτων αναζήτησης και προσφέροντας συνεχείς δυνατότητες αποθήκευσης. Αυτά τα συστήματα χρησιμοποιούν μηχανισμούς *hashing* για την αποθήκευση και τον εντοπισμό των αντικειμένων. Στα «χαλαρά δομημένα» συστήματα ο εντοπισμός των αντικειμένων βασίζεται σε υποδείξεις. Το σύστημα Freenet [13] είναι ένα παράδειγμα τέτοιου συστήματος και περιγράφεται αμέσως μετά. Στα «ισχυρά δομημένα» συστήματα, τόσο η P2P τοπολογία όσο και η τοποθέτηση των αντικειμένων καθορίζονται ακριβώς. Αυτή η αυστηρά ελεγχόμενη δομή επιτρέπει στο σύστημα να ικανοποιεί τα αιτήματα πολύ αποτελεσματικά. Κάθε κόμβος λαμβάνει ένα συγκεκριμένο προσδιοριστικό, το οποίο δημιουργείται π.χ. με μηχανισμό *hashing* στη διεύθυνση IP και τη θύρα του. Στα συστήματα Chord [14], Pastry [15] και Tapestry [16], εκχωρείται στους κόμβους ένα αριθμητικό προσδιοριστικό, ενώ στο CAN [17] εκχωρούνται στους κόμβους ολόκληρες περιοχές ενός d -διάστατου διαστήματος προσδιοριστικών. Ένας κόμβος είναι έπειτα αρμόδιος να κατέχει αντικείμενα ή δείκτες σε αντικείμενα, των οποίων τα προσδιοριστικά αντιστοιχούν στο προσδιοριστικό ή την περιοχή προσδιοριστικών του ίδιου του κόμβου. Οι γείτονες στους οποίους προωθείται ένα μήνυμα δεν επιλέγονται με τυχαία διαδικασία. Δεδομένου ότι η θέση των κόμβων και των αντικειμένων είναι προκαθορισμένη, η αναζήτηση ενός αντικειμένου ωφελείται από τη γνώση της θέσης όπου το συγκεκριμένο αντικείμενο πρέπει να βρίσκεται. Τα μηνύματα των αιτημάτων αναζήτησης προωθούνται ακολουθώντας μια σχεδόν απευθείας διαδρομή προς το στόχο και δεν υπάρχει η ανάγκη κατακλυσμού του δικτύου με μηνύματα. Οι κόμβοι σχηματίζουν επίσης συνδέσεις με βάση τις ιδιότητες των προσδιοριστικών τους. Με αυτές τις σκόπιμα σχηματισμένες συνδέσεις και

με ευφυή δρομολόγηση, ένα σύστημα μπορεί να εντοπίσει ένα αντικείμενο με βάση το προσδιοριστικό του μέσα σε περιορισμένο αριθμό βημάτων. Ωστόσο, σε δίκτυα όπου κόμβοι εισέρχονται και εξέρχονται συχνά, οι συνεχείς μετακινήσεις των στοιχείων διακυβεύουν τη σταθερότητα της δομής και τη συνοχή των στοιχείων. Επιπλέον, τα συστήματα αυτά υποστηρίζουν μόνο αναζήτηση με βάση το προσδιοριστικό ενός αντικειμένου και όχι το περιεχόμενό του.

- Το Freenet [13] είναι ένα καταναμημένο σύστημα αποθήκευσης και ανάκτησης πληροφοριών, το οποίο χρησιμοποιεί αναζήτηση ως προς το βάθος (*Depth-First Search - DFS*) με όριο βάθους D . Κάθε κόμβος προωθεί το αίτημα αναζήτησης σε έναν και μόνο γείτονα και περιμένει μια απάντηση πριν συνεχίσει να προωθείται το αίτημα (εφόσον το αίτημα δεν ικανοποιηθεί), ή προωθεί τα αποτελέσματα πίσω στην πηγή του αιτήματος (εφόσον το αίτημα ικανοποιηθεί). Με την αναζήτηση DFS, επειδή κάθε κόμβος επεξεργάζεται το αίτημα διαδοχικά, η αναζήτηση τερματίζεται μόλις ικανοποιηθεί το αίτημα, ελαχιστοποιώντας το κόστος της αναζήτησης. Ωστόσο, η διαδοχική εκτέλεση μεταφράζεται σε αυξημένο χρόνο απόκρισης, με τη χειρότερη περίπτωση να είναι εκθετική όσον αφορά το D . Ο πραγματικός χρόνος απόκρισης στο Freenet είναι μέτριος επειδή χρησιμοποιείται ευφυής δρομολόγηση. Ειδικότερα, τα αντικείμενα προσδιορίζονται από δυαδικά κλειδιά, τα οποία προκύπτουν από την εφαρμογή μιας συνάρτησης *hash* σε ένα αλφαριθμητικό που περιγράφει το περιεχόμενο του αντικειμένου και δηλώνεται από τον προμηθευτή. Κάθε κόμβος διατηρεί επίσης έναν πίνακα δρομολόγησης, ο οποίος περιέχει ένα σύνολο ζευγαριών $\langle \text{κλειδί}, \text{δείκτης} \rangle$, όπου ο *δείκτης* υποδεικνύει τον κόμβο που έχει ένα αντίγραφο του αντικειμένου με το προσδιοριστικό *κλειδί*. Οι μόνες λειτουργίες που μπορούν να εκτελεστούν στο σύστημα είναι η αναζήτηση και η εισαγωγή αντικειμένων. Για να ανακτήσει ένα αντικείμενο ένας χρήστης, θα πρέπει πρώτα να λάβει το δυαδικό κλειδί του αντικειμένου και έπειτα να στείλει ένα μήνυμα αιτήματος στον τοπικό του κόμβο, προσδιορίζοντας το κλειδί και μια τιμή του TTL. Όταν ένας κόμβος λάβει ένα αίτημα, ελέγχει πρώτα στην αποθήκη του μήπως υπάρχει το αντικείμενο που ζητήθηκε και αν το βρει το επιστρέφει, μαζί με ένα σημείωμα που λέει ότι αυτός ήταν η πηγή του αντικειμένου. Εάν δεν το βρει, ανατρέχει στον πίνακα δρομολόγησης για να βρει το κοντινότερο κλειδί σε αυτό που ζητήθηκε και στη συνέχεια προωθεί το αίτημα στον αντίστοιχο κόμβο. Εάν το αίτημα τελικά επιλυθεί, το σχετικό αντικείμενο επιστρέφεται ακολουθώντας την αλυσίδα των αιτούντων κόμβων, και καθένας από τους αιτούντες κόμβους αποθηκεύει τοπικά ένα αντίγραφο (οπότε δημιουργούνται αντίγραφα των δημοφιλέστερων αντικειμένων στο σύστημα) και επίσης δημιουργεί μια νέα καταχώρηση στον πίνακα δρομολόγησης του, αντιστοιχίζοντας την πραγματική πηγή του αντικειμένου με το ζητούμενο κλειδί. Η εισαγωγή στοιχείων ακολουθεί παρόμοια στρατηγική με αυτή της αναζήτησης και θα μπορούσε να χρησιμοποιηθεί από τους

νέους κόμβους και ως συμπληρωματικός τρόπος αναγγελίας της ύπαρξής τους στο υπόλοιπο του δικτύου. Μόλις εισέλθει στο δίκτυο ένας νέος κόμβος, του εκχωρείται αυθαίρετα ένα κλειδί, στο οποίο θα «ειδικευτεί». Καθώς περνάει ο χρόνος, λόγω των μηχανισμών εναποθήκευσης και δρομολόγησης, ο κόμβος θα περιέχει αντικείμενα των οποίων τα κλειδιά είναι κοντά στο κλειδί που του έχει ανατεθεί. Η λογική στην οποία βασίζεται το Freenet είναι ότι, βαθμιαία το διάστημα των κλειδιών γίνεται αυτόματα όλο και περισσότερο δομημένο, εξαιτίας του μηχανισμού αναζήτησης στοιχείων και οι πίνακες δρομολόγησης συγκλίνουν στην κατάσταση όπου οι περισσότερες από τις ερωτήσεις απαντώνται επιτυχώς και γρήγορα. Τέλος, όταν ένα νέο αντικείμενο φθάνει σε έναν κόμβο (είτε από μια νέα εισαγωγή, είτε από την ικανοποίηση ενός αιτήματος) που θα ανάγκαζε την αποθήκη των στοιχείων να υπερβεί ένα καθορισμένο μέγεθος, τα λιγότερο πρόσφατα χρησιμοποιημένα (Least Recently Used - LRU) αντικείμενα εκδιώκονται έως ότου υπάρξει χώρος. Οι καταχωρήσεις στους πίνακες δρομολόγησης αντικαθίστανται επίσης χρησιμοποιώντας μια πολιτική LRU.

5.3 ΠΕΡΙΓΡΑΦΗ ΠΡΟΒΛΗΜΑΤΟΣ

Τα ιδιαίτερα χαρακτηριστικά του συγκεκριμένου προβλήματος αναζήτησης πηγών πληροφορίας οδηγούν στη θεώρηση των ακόλουθων βασικών υποθέσεων:

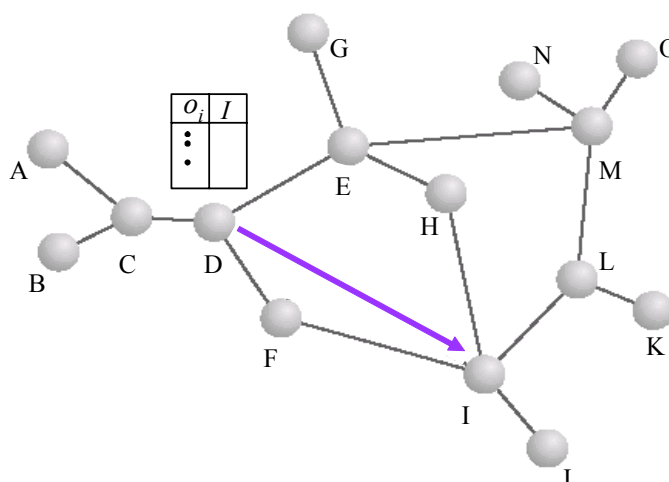
- Σε αντίθεση με τα συστήματα διανομής αρχείων όπως το Gnutella, η ποικιλομορφία πηγών στο CDS απαιτεί τον προσδιορισμό της κάθε πηγής βάσει ενός συνόλου χαρακτηριστικών ιδιοτήτων της, αφού τα γενικά προσδιοριστικά που βασίζονται σε ονόματα δεν εξυπηρετούν. Κατά συνέπεια, κάθε αίτημα για αναζήτηση πληροφορίας προσδιορίζει ένα σύνολο επιθυμητών χαρακτηριστικών για την πηγή που αναζητείται και όχι ένα γενικό προσδιοριστικό. Σημειώνεται επίσης ότι ένα *αντικείμενο πληροφορίας (context object)* είναι το αποτέλεσμα ενός αιτήματος αναζήτησης, το οποίο παράγεται από την κατάλληλη πηγή. Δεδομένου ότι εκτελείται *αναζήτηση βάσει χαρακτηριστικών ιδιοτήτων (attribute-based search)* και όχι γενικών προσδιοριστικών των πηγών, θα εστιάσουμε σε μηχανισμούς αναζήτησης για μη-δομημένα συστήματα.
- Κάθε μεσίτης διατηρεί έναν κατάλογο τοπικών πηγών (Local Sources Directory - LSD), με καταχωρήσεις που υποδεικνύουν τις τοπικά διαθέσιμες πηγές. Θεωρούμε ότι κάθε πηγή είναι μοναδική και δεν υποστηρίζεται η ύπαρξη αντιγράφων της.
- Κάθε μεσίτης διατηρεί επιπλέον έναν κατάλογο απομακρυσμένων πηγών (Remote Sources Directory - RSD), του οποίου οι καταχωρήσεις υποδεικνύουν θέσεις πηγών που ανήκουν σε άλλους μεσίτες. Το είδος των καταχωρήσεων του RSD εξαρτάται από το συγκεκριμένο πρωτόκολλο αναζήτησης πηγών που χρησιμοποιείται. Μια καταχώρηση στον RSD είναι ένα ζευγάρι (*source_info, loc*), όπου το *source_info* παρέχει μια

περιγραφή της πληροφορίας που παράγει μια πηγή και *loc* είναι η διεύθυνση του μεσίτη που θεωρείται ότι φιλοξενεί τη συγκεκριμένη πηγή.

- Εξαιτίας της ασταθούς φύσης της πληροφορίας περιβάλλοντος, ασχολούμαστε μόνο με δυναμικές πηγές πληροφοριών. Κατά συνέπεια, δεν υποστηρίζεται μηχανισμός αποθήκευσης των αντικειμένων πληροφορίας.
- Κάθε ομότιμος μεσίτης n έχει μια τοπική γειτονιά, η οποία συμβολίζεται ως $N(n)$, και ορίζεται ως το σύνολο των κόμβων που βρίσκονται κοντά στον n (π.χ. σε απόσταση ενός βήματος ή στο ίδιο τοπικό δίκτυο).
- Η συνολική τοπολογία του υπερκείμενου δικτύου είναι άγνωστη και ένας μεσίτης επικοινωνεί μόνο με γειτονικούς μεσίτες ή μεσίτες που υποδεικνύονται από τον RSD του.

5.4 ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΚΡΙΤΗΡΙΑ ΕΠΙΔΟΣΗΣ

Το P2P υπερκείμενο δίκτυο μοντελοποιείται ως συνδεδεμένος γράφος, του οποίου οι κορυφές αντιστοιχούν στους κόμβους-μεσίτες του δικτύου και οι ακμές στις ανοικτές συνδέσεις που διατηρούνται μεταξύ των κόμβων. Όπως αναφέρθηκε και νωρίτερα, κάθε κόμβος διατηρεί συνδέσεις με τους γείτονές του, καθώς επίσης και με τους κόμβους που υποδεικνύονται από τον RSD του. Κατά συνέπεια, η τοπολογία του υπερκείμενου δικτύου διαμορφώνεται από τις συνδέσεις μεταξύ γειτονικών κόμβων και τις συνδέσεις που υποδεικνύονται από τους RSDs, και δεν απαιτείται να έχει οποιαδήποτε σχέση με την τοπολογία του υποκείμενου φυσικού δικτύου. Οι τυχαίοι γράφοι [18] χρησιμοποιούνται συνήθως για να περιγράψουν την τοπολογία των P2P δικτύων. Το Σχήμα 5.2 απεικονίζει βασικά την αρχική τοπολογία ενός P2P δικτύου, όπως αυτή διαμορφώνεται από τις συνδέσεις μεταξύ γειτονικών κόμβων. Σημειώνεται ότι ο ομότιμος κόμβος D διατηρεί συνδέσεις με τους γείτονές του C , E , F και επίσης με τον κόμβο I , δεδομένου ότι ο τελευταίος υποδεικνύεται στον RSD του.



Σχήμα 5.2: Γράφος μοντελοποίησης ενός P2P υπολογιστικού συστήματος

Κάθε κόμβος χαρακτηρίζεται από ένα μοναδικό προσδιοριστικό (π.χ. τη διεύθυνση IP του) και μπορεί να εκτελέσει μια από τις ακόλουθες ενέργειες: να ενημερώσει έναν τοπικό κατάλογο (π.χ. LSD ή RSD), να στείλει ένα μήνυμα ή να λάβει ένα μήνυμα. Για να μεταφερθεί ένα μήνυμα από έναν κόμβο σε κάποιον άλλο, θα πρέπει αυτό να διασχίσει ένα μονοπάτι στο γράφο. Το μήκος αυτού του μονοπατιού είναι γνωστό ως *αριθμός βημάτων* που εκτελεί το μήνυμα. Το σύστημα είναι ασύγχρονο, δηλαδή δεν τίθεται όριο στο χρόνο που χρειάζεται ένας κόμβος για να εκτελέσει μια ενέργεια, ούτε στο χρόνο που απαιτείται για τη μετάδοση ενός μηνύματος.

Για τους σκοπούς της ανάλυσης που ακολουθεί ορίζουμε την ακόλουθη σημειογραφία και ορολογία: Δεδομένου ενός συνόλου κόμβων του γράφου ($n \in N$) και ενός συνόλου αντικειμένων πληροφορίας ($o \in O$), παριστάνουμε με $r(o, t, n)$ ένα αίτημα ($r \in R$), όπου o ($o \in O$) είναι το αιτούμενο αντικείμενο πληροφορίας, t ο χρόνος λήψης του αιτήματος από έναν κόμβο και n ($n \in N$) ο κόμβος από τον οποίο προήλθε αρχικά το δεδομένο αίτημα, ο οποίος στη συνέχεια θα αποκαλείται *εναρκτήριο κόμβος* του αιτήματος. Επιπλέον, θα χρησιμοποιούμε τον όρο *κόμβο φιλοξενίας* ενός αντικειμένου για να αναφερόμαστε στον κόμβο που διαχειρίζεται την πηγή του δεδομένου αντικειμένου.

Προκειμένου να αξιολογηθεί η αποτελεσματικότητα και η αποδοτικότητα των προτεινόμενων τεχνικών τόσο από πλευράς δικτύου όσο και από πλευράς χρηστών θεωρούμε ένα σύνολο κριτηρίων επίδοσης. Όταν ένα αίτημα διαδίδεται μέσω του δικτύου, αυτό περνάει από έναν αριθμό κόμβων. Κάθε κόμβος καταναλώνει εύρος ζώνης καθώς στέλνει το μήνυμα με το αίτημα. Το κύριο κόστος των αιτημάτων μπορεί επομένως να περιγραφεί σε σχέση με την κατανάλωση εύρους ζώνης. Στην περίπτωση αυτή ορίζουμε το ακόλουθο κριτήριο:

- **Μέσος αριθμός μηνυμάτων ανά αίτημα** (*Average Number of Messages per Request*): ο λόγος του συνολικού αριθμού μηνυμάτων που στέλνονται για όλα τα αιτήματα προς τον συνολικό αριθμό των αιτημάτων.

Ενώ επιδιώκουμε να μειώσουμε τους πόρους που καταναλώνονται από το σύστημα για την επεξεργασία ενός αιτήματος, θα πρέπει επιπλέον να εξασφαλίσουμε ότι οι προτεινόμενες τεχνικές δεν υποβιβάζουν την εμπειρία του χρήστη. Μια σημαντική πτυχή της εμπειρίας του χρήστη είναι ο χρόνος που θα πρέπει να περιμένει μέχρι να επιστραφεί το αποτέλεσμα του αιτήματός του. Οι χρόνοι απόκρισης τείνουν να είναι μεγάλοι στα P2P δίκτυα, δεδομένου ότι ένα αίτημα εκτελεί πολλά βήματα μέσα στο δίκτυο. Διαισθητικά η επίδοση, όπως γίνεται αντιληπτή από ένα χρήστη, μπορεί να περιγραφεί από το μέσο μήκος του μονοπατιού που ακολουθεί ένα αίτημα μέχρι να εντοπίσει την κατάλληλη πηγή πληροφορίας. Ειδικότερα, ορίζουμε τα εξής κριτήρια:

- **Ποσοστό επιτυχίας** (*Request Hit Ratio*): ο λόγος του αριθμού των επιτυχόντων αιτημάτων προς το συνολικό αριθμό των αιτημάτων.
- **Μέσο μήκος μονοπατιού ανά αίτημα** (*Average Path-length per Request*): ο λόγος του συνολικού αριθμού βημάτων που εκτελέστηκαν σε όλα τα αιτήματα προς το συνολικό αριθμό των αιτημάτων.
- **Μέσο μήκος μονοπατιού ανά επιτυχές αίτημα** (*Average Path-length per Successful Request*): ο λόγος του συνολικού αριθμού βημάτων που εκτελέστηκαν σε όλα τα επιτυχή αιτήματα προς το συνολικό αριθμό των επιτυχόντων αιτημάτων.

Σημειώνεται ότι ένα αίτημα θεωρείται αποτυχημένο όταν δεν έχει εντοπίσει την πηγή σε αριθμό βημάτων μικρότερο ή ίσο του TTL. Στην περίπτωση αυτή θεωρούμε ότι έχει εκτελέσει αριθμό βημάτων ίσο με TTL.

5.5 ΠΡΟΤΕΙΝΟΜΕΝΗ ΛΥΣΗ

Με βάση το κατανεμημένο υπολογιστικό περιβάλλον που παρέχεται από τα Ενεργά Δίκτυα (ενότητα 3.6) προτείνονται στη συνέχεια του κεφαλαίου συγκεκριμένες αλγοριθμικές λύσεις, οι οποίες εφαρμόζονται πάνω από το P2P υπόδειγμα επικοινωνίας και στοχεύουν να ενισχύσουν την αποτελεσματικότητα και αποδοτικότητα ενός συστήματος παροχής χρονικά κρίσιμων υπηρεσιών με επίγνωση του περιβάλλοντος (CASs). Πιο συγκεκριμένα, παρέχονται *αντιδραστικοί* (*reactive*) και *προδραστικοί* (*proactive*) μηχανισμοί, οι οποίοι λαμβάνουν μέτρα προκειμένου να βελτιωθεί ο τρόπος αναζήτησης των πληροφοριών κατά τη διάρκεια και πριν την εκτέλεση των αιτημάτων, αντίστοιχα.

Στα πλαίσια των αντιδραστικών μηχανισμών, οι προτεινόμενοι αλγόριθμοι αναζήτησης εκμεταλλεύονται επίγνωση της ζήτησης πληροφορίας, αξιοποιώντας *ιδιότητες*

εντοπιότητας αναφοράς (*locality of reference properties*) που είναι πιθανό να επιδεικνύουν τα μοντέλα ζήτησης πληροφορίας. Το προφίλ ζήτησης της πληροφορίας παρακολουθείται από τους κατανεμημένους ομότιμους κόμβους, οι οποίοι ανταλλάσσουν την τοπική γνώση τους με αποδοτικά μέσα που παρέχονται από τα Ενεργά Δίκτυα. Οι σχετικοί αλγόριθμοι αποφεύγουν την εξαντλητική σπατάλη πόρων των συνήθων αλγορίθμων πλημμύρας, και ένα ζητούμενο αντικείμενο πληροφορίας που παρουσιάζει μια χαρακτηριστική ιδιότητα εντοπιότητας μπορεί να βρεθεί γρήγορα και οικονομικά με την εφαρμογή ενός κατάλληλα περιορισμένου αλγορίθμου πλημμύρας, ο οποίος προσαρμόζεται ανάλογα με τη συγκεκριμένη ιδιότητα.

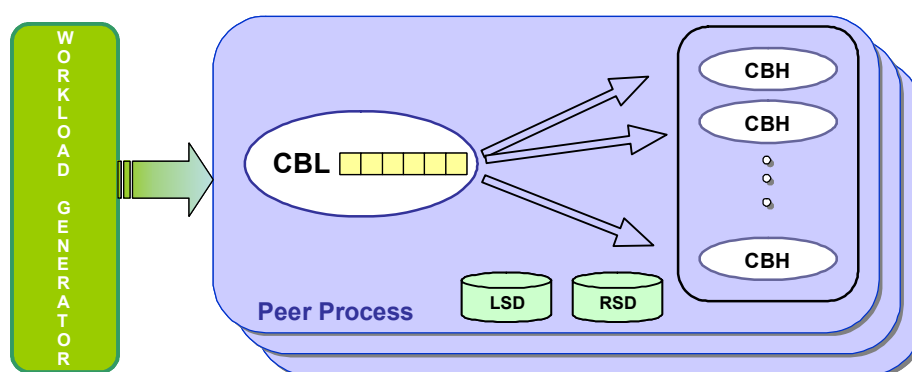
Στα πλαίσια των προδραστικών μηχανισμών μελετάται η χρήση διαφημίσεων, η οποία υπονοεί ότι μόλις μια νέα πηγή πληροφορίας εισέλθει στο σύστημα, ο κόμβος φιλοξενίας της μεταδίδει πληροφορία σχετική με την περιγραφή και τη θέση της δεδομένης πηγής στους υπόλοιπους κόμβους. Ο προτεινόμενος μηχανισμός εκμεταλλεύεται το μοντέλο μικρού-κόσμου (*small-world model*) για τη διανομή των διαφημίσεων και στη συνέχεια εφαρμόζει ειδική αναζήτηση για τον εντοπισμό μιας πηγής. Λεπτομερείς περιγραφές των μηχανισμών δίνονται στη συνέχεια και συγκεκριμένα στις ενότητες 5.7 και 5.8.

5.6 ΠΕΡΙΒΑΛΛΟΝ ΠΡΟΣΟΜΙΩΣΗΣ

Οι δοκιμές των προτεινόμενων μηχανισμών πραγματοποιήθηκαν με χρήση ενός εργαλείου προσομοίωσης ασύγχρονων γεγονότων, το οποίο υλοποιήσαμε σε *Java* και ειδικότερα με χρήση της τεχνολογίας των *Threads*. Οι ομότιμοι μεσίτες ρυθμίστηκαν ώστε να τρέχουν ως *threads* σε μια εικονική μηχανή της *Java* (*Java Virtual Machine – Java VM*), ενώ η επικοινωνία μεταξύ των μεσιτών προσομοιώθηκε με επικλήσεις σε τοπικά αντικείμενα. Αν και υπάρχουν πολλά έτοιμα εργαλεία προσομοίωσης, προτιμήσαμε να υλοποιήσουμε ένα ανεξάρτητο που να προσεγγίζει όσο το δυνατόν καλύτερα το κατανεμημένο περιβάλλον που έχουμε περιγράψει. Είναι γεγονός ότι οι δημοφιλείς προσομοιωτές όπως ο *NS-2* [19] ή ο *Opnet Modeler* [20] παρέχουν ένα προηγμένο περιβάλλον προσομοίωσης για τη μελέτη κάθε είδους δικτυακού πρωτοκόλλου, αλλά υπό το κόστος της αυξημένης προσοχής που απαιτεί η ρύθμιση ενός συνόλου παραμέτρων για τα χαμηλά στρώματα, το οποίο θα μπορούσε να αποφευχθεί στην περίπτωσή μας. Επιπλέον, έχει αποδειχθεί ότι αυτό το στάδιο της ρύθμισης παραμέτρων έχει σοβαρές επιπτώσεις στην απόδοση ενός πρωτοκόλλου και μπορεί να οδηγήσει σε αρκετά διαφορετικά αποτελέσματα ανάλογα με τον συγκεκριμένο προσομοιωτή που χρησιμοποιείται, δεδομένου ότι καθένας απαιτεί διαφορετικά επίπεδα λεπτομέρειας [21].

Στον ανεξάρτητο προσομοιωτή που υλοποιήσαμε, κάθε ομότιμος μεσίτης είναι ένα *thread*, το οποίο παράγει άλλα *threads*. Ειδικότερα, κάθε ομότιμη διεργασία (μεσίτης) υλοποιείται ως το βασικό *thread ContextBrokerListener (CBL)*, όπως φαίνεται στο Σχήμα 5.3, το οποίο λαμβάνει μηνύματα από άλλες ομότιμες διεργασίες και δημιουργεί ένα νέο *thread*,

το *ContextBrokerHandler (CBH)* για κάθε μήνυμα αιτήματος που λαμβάνει. Κάθε CBH προσπελάζει τους τοπικούς καταλόγους LSD και RSD και καθορίζει σε ποιες ομότιμες διεργασίες θα πρέπει να στείλει το μήνυμα. Στη συνέχεια τοποθετεί το μήνυμα στην ουρά αναμονής του CBL των αντίστοιχων διεργασιών. Η *γεννήτρια αιτημάτων (Workload Generator)* είναι μια εξωτερική διεργασία που παράγει τα αιτήματα αναζήτησης πηγών. Οι παράμετροι που χρησιμοποιούνται κατά τη διαδικασία παραγωγής των αιτημάτων αποθηκεύονται σε έναν πίνακα (*Workload Table*), ο οποίος αρχικοποιείται πριν την εκτέλεση ενός πειράματος. Αυτές οι παράμετροι καθορίζουν την κατανομή των ζητούμενων πηγών. Σημειώνεται ότι τα παραγόμενα αιτήματα στέλνονται για εκτέλεση διαδοχικά στις διάφορες ομότιμες διεργασίες.



Σχήμα 5.3: Ομότιμες διεργασίες και τα αντίστοιχα threads του προσομοιωτή

Υποθέτουμε ότι κατά την εξέλιξη ενός πειράματος ο αριθμός των ομότιμων μεσιτών είναι σταθερός, ότι δεν συμβαίνουν αποτυχίες στο σύστημα, και ότι οι πηγές δεν μετακινούνται μεταξύ των ομότιμων μεσιτών. Επιπλέον, θεωρούμε ότι τα αιτήματα είναι απλά και υπάρχει ακριβώς μια πηγή που ικανοποιεί πλήρως κάθε αίτημα. Το εργαλείο προσομοίωσης θα χρησιμοποιηθεί για την αποτίμηση των κριτηρίων επίδοσης που έχουν οριστεί προκειμένου να αξιολογηθούν οι προτεινόμενοι αλγόριθμοι.

Αξίζει να σημειωθεί ότι ο μεγάλος αριθμός διεργασιών που απαιτούνται από τα σενάρια προσομοίωσης συστημάτων μεγάλης κλίμακας θίγει το πρόβλημα της έλλειψης πόρων. Για το λόγο αυτό, έχουμε εκμεταλλευτεί μερικές από τις ευκολίες που παρέχουν οι βιβλιοθήκες της Java, οι οποίες σχετίζονται με ζητήματα συγχρονισμού, προγραμματισμού των threads και λήψης μέτρων για περιορισμό των πόρων που χρησιμοποιούνται (π.χ. *Thread pools*).

5.7 ΜΗΧΑΝΙΣΜΟΙ ΑΝΑΖΗΤΗΣΗΣ ΜΕ ΕΠΙΓΝΩΣΗ ΤΗΣ ΖΗΤΗΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ

Σε αυτή την ενότητα η μελέτη εστιάζεται σε ιδιότητες εντοπιότητας που είναι πιθανό να παρουσιάζουν τα μοντέλα ζήτησης πληροφορίας και καθοδηγείται από τη σημασία

αξιοποίησης αυτών των ιδιοτήτων από έναν μηχανισμό αναζήτησης πληροφορίας [22]. Οι ιδιότητες εντοπιότητας με τις οποίες θα ασχοληθούμε είναι οι εξής τρεις: η *χρονική* (*temporal*), η *γεωγραφική* (*geographical*) και η *χωρική* (*spatial*). Στη συνέχεια προτείνονται ορισμένες οικονομικά αποδοτικές εναλλακτικές λύσεις του αλγορίθμου της πλημμύρας, οι οποίες αξιοποιούν αυτές τις ιδιότητες με έναν προσαρμοστικό τρόπο. Αξίζει να σημειωθεί ότι για τις πρώτες δύο ιδιότητες, οι μηχανισμοί αναζήτησης αποτιμούν τη ζήτηση της πληροφορίας όπως αυτή γίνεται αντιληπτή από την πλευρά ενός εξυπηρετητή, ενώ για την τρίτη ιδιότητα, από τη πλευρά των πελατών.

5.7.1 ΟΡΙΣΜΟΣ ΚΑΙ ΜΟΝΤΕΛΟΠΟΙΗΣΗ ΙΔΙΟΤΗΤΩΝ ΕΝΤΟΠΙΟΤΗΤΑΣ

5.7.1.1 Χρονική Εντοπιότητα Αναφοράς

Η *χρονική εντοπιότητα αναφοράς* (*temporal locality of reference*) υπονοεί ότι αντικείμενα που ζητήθηκαν πρόσφατα είναι πιθανό να ζητηθούν ξανά στο μέλλον [23]. Γενικά, η χρονική εντοπιότητα χαρακτηρίζεται τόσο από τη συχνότητα όσο και από τις χρονικές συσχετίσεις των επαναλαμβανόμενων αιτημάτων για συγκεκριμένη πληροφορία [24]. Κατά συνέπεια, η δημοτικότητα δεν είναι ο μόνος παράγοντας της χρονικής εντοπιότητας. Οι χρονικοί συσχετισμοί των επαναλαμβανόμενων αιτημάτων για συγκεκριμένα αντικείμενα συνθέτουν ένα άλλο σημαντικό προσδιοριστικό της [25]. Εντούτοις, για τους σκοπούς των μηχανισμών αναζήτησης που θα μελετήσουμε, εστιάζουμε στη χρονική εντοπιότητα που οφείλεται στη δημοτικότητα.

Ένα ποσοτικό μέτρο της χρονικής εντοπιότητας λόγω δημοτικότητας είναι η συχνότητα με την οποία ζητείται ένα αντικείμενο. Επομένως, ορίζουμε τη δημοτικότητα f ενός αντικειμένου o , ως:

$$f_o = \sum_{n \in N} |R_n^o| \quad (1)$$

όπου N είναι το σύνολο των κόμβων του γράφου και R_n^o το σύνολο των αιτημάτων για το αντικείμενο o με εναρκτήριο κόμβο τον $n \in N$, δηλαδή: $R_n^o = \{\forall r(o', t', n') \in R \mid o' = o, n' = n\}$.

Παρατηρώντας τη ζήτηση της πληροφορίας σε πραγματικά σενάρια υπηρεσιών με επίγνωση περιβάλλοντος [26] καταλήξαμε στο συμπέρασμα ότι η χρονική εντοπιότητα αναφοράς είναι εμφανής, αφού ορισμένα αντικείμενα ήταν πιο δημοφιλή από άλλα και έτσι ζητούνταν επανειλημμένα από την πλειοψηφία των μεσιτών. Αυτά τα αντικείμενα αναφέρονται συνήθως σε βασικούς τύπους πληροφορίας, οι οποίοι αποτελούν βασικά συστατικά άλλων πιο σύνθετων τύπων πληροφορίας και έτσι ζητούνται συχνά. Για παράδειγμα, μια πηγή που παρέχει πληροφορίες θέσης των κινητών χρηστών για μια ευρεία

γεωγραφική περιοχή είναι πολύ πιθανό να προσπελάζεται επανειλημμένα, δεδομένου ότι η πληροφορία θέσης αποτελεί βασικό συστατικό πολλών σύνθετων και πιο εξειδικευμένων τύπων πληροφορίας.

5.7.1.2 Γεωγραφική Εντοπιότητα Αναφοράς

Η *γεωγραφική εντοπιότητα αναφοράς* (*geographical locality of reference*) λαμβάνει υπόψη τη θέση των κόμβων από τους οποίους ξεκινάει ένα επαναλαμβανόμενο αίτημα. Υπονοεί ότι ένα αντικείμενο που ζητήθηκε από συγκεκριμένο πελάτη είναι πιθανό να ζητηθεί ξανά στο μέλλον από κοντινούς πελάτες [23]. Ένα περιβάλλον χαρακτηρίζεται από γεωγραφική εντοπιότητα αναφοράς πληροφοριών όταν χρήστες σε κοντινή γεωγραφική απόσταση είναι πιθανό να ζητούν τα ίδια αντικείμενα πληροφορίας [24]. Με άλλα λόγια, η γεωγραφική εντοπιότητα αναφοράς εστιάζει στη γεωγραφική περιοχή στην οποία εμφανίζεται ένα επαναλαμβανόμενο αίτημα και δείχνει εάν επαναλαμβανόμενα αιτήματα προέρχονται από κοντινούς κόμβους.

Προκειμένου να περιγράψουμε τα κριτήρια της γεωγραφικής εντοπιότητας ορίζουμε το σύνολο $N(n_o, l) \subseteq N$, το οποίο αποτελείται από τους κόμβους που βρίσκονται σε μέγιστη απόσταση l βημάτων από έναν κόμβο $n_o \in N$, ως:

$$N(n_o, l) = \{ \forall n \in N \mid \text{dist}(n, n_o) < l \} \quad (2)$$

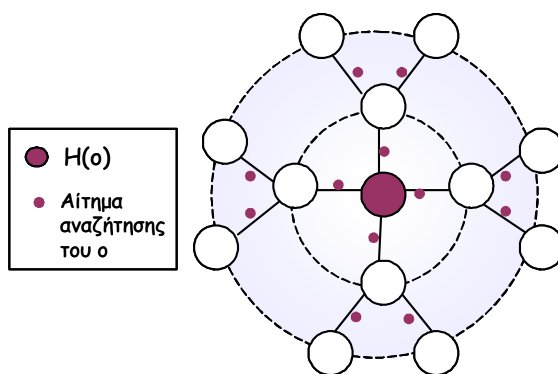
όπου $\text{dist}(n, n_o)$ το μήκος του ελάχιστου μονοπατιού μεταξύ των n, n_o .

Αναλόγως εάν η γεωγραφική περιοχή που παρουσιάζονται τα επαναλαμβανόμενα αιτήματα για ένα δεδομένο αντικείμενο o βρίσκεται κοντά ή μακριά από τον κόμβο φιλοξενίας του o , χαρακτηρίζουμε περαιτέρω τη γεωγραφική εντοπιότητα ως *εγχώρια* ή *απομακρυσμένη*. Ένα αντικείμενο πληροφορίας o παρουσιάζει *εγχώρια γεωγραφική εντοπιότητα* (*Home Geographical Locality - HGL*) (Σχήμα 5.4) ακτίνας L , εάν ένας σημαντικός αριθμός επαναλαμβανόμενων αιτημάτων για το o προέρχεται από την ίδια γεωγραφική περιοχή, η οποία εκτείνεται γύρω από τον κόμβο φιλοξενίας του o , δηλαδή:

$$\sum_{n \in N} \frac{|R_n^o| \cdot x(n, N(H(o), L))}{f_o} > T_{home} \quad (3)$$

όπου f_o η δημοτικότητα του o , $H(o)$ ο κόμβος φιλοξενίας του o , R_n^o το σύνολο των επαναλαμβανόμενων αιτημάτων για το o με εναρκτήριο κόμβο τον $n \in N$, T_{home} ένα ορισμένο κατώφλι και:

$$x(n, N) = \begin{cases} 1, & \text{αν } n \in N \\ 0, & \text{αν } n \notin N \end{cases} \quad (4)$$

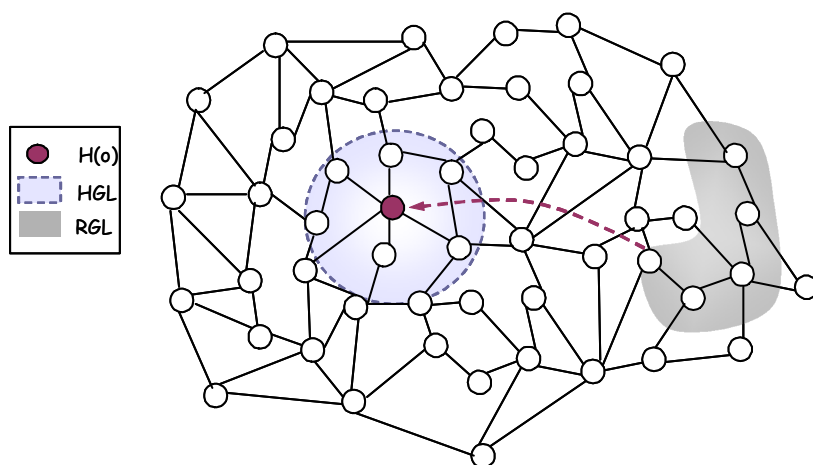


Σχήμα 5.4: Εγχώρια Γεωγραφική Εντοπιότητα (HGL)

Ένα αντικείμενο o παρουσιάζει απομακρυσμένη γεωγραφική εντοπιότητα (*Remote Geographical Locality - RGL*) (Σχήμα 5.5) ακτίνας L , εάν υπάρχει κόμβος $n_o \in N$, ($n_o \neq H(o)$), έτσι ώστε ένας σημαντικός αριθμός επαναλαμβανόμενων αιτημάτων για το o να προέρχεται από την ίδια γεωγραφική περιοχή, η οποία εκτείνεται γύρω από τον n_o , δηλαδή:

$$\sum_{n \in N} \frac{|R_n^o| \cdot x(n, N(n_o, L))}{f_o} > T_{remote} \tag{5}$$

όπου T_{remote} ένα ορισμένο κατώφλι.



Σχήμα 5.5: Εγχώρια Γεωγραφική Εντοπιότητα (HGL) έναντι Απομακρυσμένης Γεωγραφικής Εντοπιότητας (RGL)

Η γεωγραφική εντοπιότητα παρουσιάζεται συχνά στα μοντέλα ζήτησης πληροφορίας περιβάλλοντος, δεδομένου ότι οι υπηρεσίες με επίγνωση του περιβάλλοντος αναπτύσσονται συνήθως για να παρασχεθούν σε συγκεκριμένες περιοχές, συχνά κοντά στις πηγές πληροφοριών που χρησιμοποιούν, αλλά μερικές φορές μακριά από αυτές [26]. Για παράδειγμα, σε μια πανεπιστημιούπολη, μια πηγή που παρέχει πληροφορίες ακαδημαϊκού τύπου χρησιμοποιείται συνήθως από σχετικές υπηρεσίες που προσφέρονται μέσα στην ίδια την πανεπιστημιούπολη (περίπτωση εγχώριας γεωγραφικής εντοπιότητας). Από την άλλη

μεριά, μια τέτοια πηγή θα μπορούσε να χρησιμοποιηθεί από υπηρεσία που παρέχεται σε μια διαφορετική πανεπιστημιούπολη, η οποία μπορεί να συνδυάζει πληροφορίες ακαδημαϊκού τύπου από πολλά διαφορετικά πανεπιστήμια (περίπτωση απομακρυσμένης γεωγραφικής εντοπιότητας).

5.7.1.3 Χωρική Εντοπιότητα Αναφοράς

Η *χωρική εντοπιότητα αναφοράς* (*spatial locality of reference*) υπονοεί ότι ένα αντικείμενο που «γειτονεύει» με αντικείμενο που ζητήθηκε πρόσφατα είναι πιθανό να ζητηθεί στο μέλλον [27]. Μια τέτοια ιδιότητα αναζητά εξαρτήσεις μεταξύ των ζητούμενων αντικειμένων. Ειδικότερα, διακρίνονται δύο τύποι εξαρτήσεων. Μια *ισχυρή εξάρτηση* (*embedding dependency*) παρατηρείται όταν ένα αντικείμενο o_j ζητείται πάντα όταν ζητείται ένα άλλο αντικείμενο o_i . Μια *χαλαρή εξάρτηση* (*traversal dependency*) εμφανίζεται όταν ένα αντικείμενο o_j ζητείται μερικές φορές όταν ζητείται ένα άλλο αντικείμενο o_i [23].

Ένα αντικείμενο o παρουσιάζει *χωρική εντοπιότητα*, εάν υπάρχει ακολουθία αιτημάτων s_o , η οποία ξεκινάει με το αντικείμενο o , και ισχύει:

$$\frac{f_{s_o}}{f_o} > T_{spat} \quad (6)$$

όπου f_o η δημοτικότητα του αντικειμένου o , f_{s_o} η δημοτικότητα της ακολουθίας s_o (στο σύνολο των πελατών) και T_{spat} ένα ορισμένο κατώφλι. Ρυθμίζοντας την τιμή του T_{spat} μπορούμε να περιορίσουμε ή να χαλαρώσουμε το είδος της εξάρτησης. Ειδικότερα, ρυθμίζοντας αυτό κοντά στο 1, περιορίζουμε το είδος της εξάρτησης σε ισχυρή, διαφορετικά διευρύνουμε το είδος της εξάρτησης για να περιλάβει επίσης τις χαλαρές εξαρτήσεις.

Η χωρική εντοπιότητα αναφοράς είναι πιθανό να εμφανίζεται στα μοντέλα ζήτησης πληροφορίας περιβάλλοντος, λόγω του δομοστοιχειωτού τρόπου με τον οποίο δομούνται συνήθως τα αντικείμενα πληροφορίας [26]. Ειδικότερα, ένα αντικείμενο πληροφορίας μπορεί να αποτελείται από άλλα απλούστερα αντικείμενα σε διαδοχική ακολουθία, απαιτώντας σειριακή ανάθεση τιμών στα επιμέρους αντικείμενα. Αυτό προκαλεί τη δημιουργία ισχυρών εξαρτήσεων μεταξύ των αντικειμένων που ζητούνται. Από την άλλη μεριά, είναι επίσης πιθανό να υπάρχουν χαλαρές εξαρτήσεις, δεδομένου ότι μερικές δημοφιλείς υπηρεσίες με επίγνωση του περιβάλλοντος επιβάλλουν το σχηματισμό ευδιάκριτων ομάδων αντικειμένων, τα οποία ζητούνται συνήθως με μια συγκεκριμένη σειρά.

5.7.2 ΜΗΧΑΝΙΣΜΟΙ ΑΝΑΖΗΤΗΣΗΣ

Το πρωτόκολλο αναζήτησης που χρησιμοποιείται ευρέως είναι αυτό της *πλημμύρας* (*Flooding*), κατά το οποίο ένα αίτημα αναζήτησης διαδίδεται σε όλους τους γείτονες μέσα σε μια ορισμένη ακτίνα. Ειδικότερα, κάθε κόμβος διαβιβάζει κάθε μήνυμα αναζήτησης που λαμβάνει σε όλους τους γείτονές του, εκτός από εκείνον από τον οποίο παρέλαβε το μήνυμα.

Κάθε μήνυμα περιέχει το προσδιοριστικό του αιτήματος που μεταφέρει, ώστε να απορρίπτονται πολλαπλά μηνύματα του ίδιου αιτήματος που λαμβάνονται από έναν συγκεκριμένο κόμβο. Για να αποφευχθεί η απεριόριστη διάδοση μηνυμάτων μέσω του δικτύου, κάθε μήνυμα περιέχει έναν μετρητή TTL. Όταν ένας κόμβος λάβει ένα μήνυμα, μειώνει κατά μια μονάδα την τιμή TTL αυτού του μηνύματος. Όταν το TTL γίνει μηδέν, το μήνυμα απορρίπτεται. Επίσης, κάθε μήνυμα απόκρισης καθοδηγείται κατά μήκος της διαδρομής από την οποία έφθασε το μήνυμα του αιτήματος.

Ο πολύ καλός χρόνος απόκρισης του Flooding σε ένα αίτημα αναζήτησης, μετρούμενος σε αριθμό βημάτων που απαιτούνται για τον εντοπισμό της κατάλληλης πηγής, επιτυγχάνεται με αντίτιμο την εξαντλητική χρήση του δικτύου, από πλευράς κατανάλωσης πόρων επικοινωνίας και επεξεργασίας. Ενώ λοιπόν ο Flooding έχει πολύ καλή επίδοση για μικρές τιμές του TTL, για μεγάλες τιμές του TTL κάθε μεμονωμένο αίτημα προκαλεί μεγάλη κίνηση στο δίκτυο και τα μεγάλα συστήματα κατακλύζονται πολύ γρήγορα από το υπερβολικό φορτίο των αιτημάτων. Είναι ενδεικτικό ότι μια γραμμική αύξηση στην τιμή του TTL προκαλεί εκθετική αύξηση στον αριθμό των μηνυμάτων. Ένα άλλο πρόβλημα είναι τα πολλαπλά αντίγραφα ενός αιτήματος που στέλνονται σε έναν κόμβο από τους πολλαπλούς γείτονές του, και τα οποία αποτελούν καθαρή σπατάλη εύρους ζώνης. Το πρόβλημα αυτό γίνεται εντονότερο σε δίκτυα με υψηλό βαθμό συνδετικότητας. Αυτοί οι περιορισμοί αυξάνονται καθώς αυξάνει η κλίμακα του δικτύου και ο ρυθμός παραγωγής αιτημάτων και συνεπώς ο Flooding δεν έχει καλές δυνατότητες κλιμάκωσης.

Υποκινούμενοι από τους περιορισμούς του Flooding, ασχολούμαστε με την ανάπτυξη εναλλακτικών λύσεων με καλύτερες δυνατότητες κλιμάκωσης, **με σκοπό την επίλυση των αιτημάτων αναζήτησης σε χρόνο συγκρίσιμο με του Flooding αλλά μειώνοντας τον αριθμό των μηνυμάτων στο δίκτυο**. Οι προτεινόμενες τεχνικές βασίζονται σε κατάλληλα περιορισμένες εκδόσεις του Flooding, αξιοποιώντας ιδιότητες εντοπιότητας αναφοράς που είναι πιθανό να παρουσιάζουν τα μοντέλα ζήτησης της πληροφορίας. Θεωρούμε ότι κατάλληλοι πίνακες που διατηρούνται σε κάθε ομότιμο κόμβο παρέχουν πληροφορίες για τα αντικείμενα που επιδεικνύουν ιδιότητες εντοπιότητας ως προς τη ζήτησή τους: ο *T-Table* για τη χρονική εντοπιότητα, ο *G-Table* για τη γεωγραφική εντοπιότητα και ο *S-Table* για τη χωρική εντοπιότητα. Πριν ένας κόμβος ξεκινήσει την αποστολή ενός αιτήματος αναζήτησης, ψάχνει στους πίνακές του για πηγές που πιθανόν να ικανοποιούν το αίτημα. Σε περίπτωση που βρεθεί κάποια, εφαρμόζει μια κατάλληλα περιορισμένη έκδοση του Flooding, ανάλογα με τον τύπο της ιδιότητας. Διαφορετικά, εφαρμόζει απλό Flooding. Σε κάθε περίπτωση, μόλις βρεθεί το ζητούμενο αντικείμενο, το μήνυμα της απάντησης ακολουθεί την αντίστροφη πορεία για να φθάσει στον εναρκτήριο κόμβο, και μια νέα καταχώρηση με την πηγή που εντοπίστηκε εισάγεται στον RSD του εναρκτήριου κόμβου.

Οι προτεινόμενοι μηχανισμός αντιμετωπίζει ένα σύνολο κρίσιμων προκλήσεων: κατ' αρχάς, οι ιδιότητες εντοπιότητας θα πρέπει να προσδιορίζονται σωστά. Δεύτερον, η κίνηση στο δίκτυο για τη διατήρηση των πινάκων θα πρέπει να είναι χαμηλή, και τέλος, οι πίνακες θα πρέπει να καταλαμβάνουν μικρό χώρο. Όσον αφορά τις δυο πρώτες προκλήσεις, οι ιδιότητες εντοπιότητας προσδιορίζονται τοπικά από κάθε κόμβο, ο οποίος καταγράφει τα λαμβανόμενα αιτήματα που επιλύει τοπικά. Οι καταχωρήσεις των πινάκων ανανεώνονται με αποδοτικούς μηχανισμούς επωμισμού (*riggybacking*) [28] που παρέχονται από τα Ενεργά Δίκτυα και θα περιγραφούν στη συνέχεια. Τέλος, κάθε καταχώρηση στους πίνακες καταλαμβάνει μικρό χώρο, δεδομένου ότι απλώς υποδεικνύει ορισμένα χαρακτηριστικά της κάθε ιδιότητας εντοπιότητας.

Η λογική πίσω από την προτεινόμενη προσέγγιση είναι βασισμένη σε δύο παρατηρήσεις: πρώτον, μόνο ένα μικρό ποσοστό όλων των διαθέσιμων αντικειμένων είναι πιθανό να παρουσιάζει κάποια ιδιότητα εντοπιότητας. Δεύτερον, τα περισσότερα αιτήματα αναφέρονται σε τέτοια αντικείμενα. Επομένως, αναμένουμε ότι η πλειοψηφία των αιτημάτων μπορεί να ικανοποιηθεί ανατρέχοντας στους πίνακες και εφαρμόζοντας μια περιορισμένη έκδοση του Flooding, ενώ οι πληροφορίες εντοπιότητας καταλαμβάνουν μικρό αποθηκευτικό χώρο και προσπελάζονται γρήγορα. Με τον τρόπο αυτό, οι πίνακες βοηθούν να βελτιωθεί η απόδοση της αναζήτησης για τα δημοφιλή αντικείμενα που χαρακτηρίζονται από κάποια ιδιότητα εντοπιότητας, με αμελητέα επιβάρυνση. Ο βαθμός κατά τον οποίο μειώνονται οι δαπάνες της αναζήτησης σχετίζεται στενά με το βαθμό της ιδιότητας που επιδεικνύεται. Σημειώνεται ότι η αναζήτηση στους πίνακες γίνεται μια φορά για κάθε αίτημα, δηλαδή στον εναρκτήριο κόμβο του αιτήματος, ενώ η αναζήτηση στους καταλόγους RSDs σε κάθε κόμβο που μεταδίδεται το μήνυμα.

5.7.2.1 Πλημμύρα περιορισμένου εύρους (*Range-limited Flooding*)

Ο *Range-limited Flooding* διαφοροποιείται από τον απλό Flooding στο γεγονός ότι ένα αίτημα που λαμβάνεται από έναν κόμβο δεν στέλνεται σε όλους τους γείτονές του αλλά σε έναν ορισμένο αριθμό, και οι γείτονες αυτοί επιλέγονται τυχαία. Ένας τέτοιος μηχανισμός θα μπορούσε να αποταμιεύσει εύρος ζώνης, διατηρώντας ταυτόχρονα την ίδια μέση καθυστέρηση εντοπισμού ενός ζητούμενου αντικειμένου με τον Flooding, εάν χρησιμοποιηθεί υπό συγκεκριμένες συνθήκες. Ειδικότερα, η ύπαρξη χρονικής εντοπιότητας θεωρείται σημαντική στην αποτελεσματικότητα του εν λόγω μηχανισμού.

Ένα δημοφιλές αντικείμενο χαρακτηρίζεται από επαναλαμβανόμενα αιτήματα που μπορεί να προέρχονται από διαφορετικούς κόμβους και έτσι είναι πιθανό να έχει καταχωρηθεί στους RSDs ενός σχετικά μεγάλου αριθμού κόμβων. Επομένως, όσο πιο δημοφιλές είναι ένα αντικείμενο, τόσο περισσότερες πιθανότητες υπάρχουν να βρεθεί γρήγορα η θέση της αντίστοιχης πηγής. Προκειμένου λοιπόν να μειωθούν οι περιττές δαπάνες

εύρους ζώνης στην περίπτωση αντικειμένων υψηλής δημοτικότητας, ο Range-limited Flooding προτείνει περιορισμό της πλημμύρας σε βαθμό που καθορίζεται από τη δημοτικότητα ενός ζητούμενου αντικειμένου, έτσι ώστε να αποταμιευτεί όσο το δυνατόν περισσότερο εύρος ζώνης χωρίς να αυξηθεί ο χρόνος εντοπισμού της κατάλληλης πηγής. Σύμφωνα με αυτή τη λογική, όσο πιο δημοφιλές είναι το αντικείμενο που ζητείται, τόσο μικρότερος θα πρέπει να είναι ο αριθμός των γειτόνων K , στους οποίους θα διαβιβάζεται το αίτημα σε κάθε βήμα, δεδομένου ότι η δημοτικότητα παρέχει μια ένδειξη για τον αριθμό των προηγούμενων επαναλαμβανόμενων αιτημάτων για το αντικείμενο, τα οποία θα πρέπει να έχουν εντοπίσει και αποθηκεύσει τη θέση της αντίστοιχης πηγής. Εάν υποθεθεί ότι ο κόμβος $i \in N$ λαμβάνει ένα αίτημα $r(o, t, n)$, η τιμή του K καθορίζεται από:

$$K \propto \frac{1}{f_o} \cdot d_i \quad (7)$$

όπου f_o η δημοτικότητα του o και d_i ο βαθμός του κόμβου i .

Υποθέτουμε ότι ο πίνακας T-Table φέρει πληροφορίες δημοτικότητας για τα δημοφιλέστερα αντικείμενα. Ειδικότερα, μια εγγραφή (o, f_o) δείχνει τη δημοτικότητα f_o του αντικειμένου o . Η δημοτικότητα f_o ενός αντικειμένου o μετράται από τον κόμβο φιλοξενίας του o . Κάθε κόμβος διατηρεί αρχεία των αιτημάτων που έχει επιλύσει μέσω των τοπικών πηγών του και έτσι είναι σε θέση να προσδιορίσει τη δημοτικότητα κάθε πηγής που διατηρεί. Ένα αντικείμενο o καταχωρείται στον T-Table μόνο αν η δημοτικότητά του είναι μεγαλύτερη από ένα δεδομένο κατώφλι a (δηλαδή $f_o > a$), ενώ μια λειτουργία ενημέρωσης της τιμής της δημοτικότητας του o πραγματοποιείται μόνο εάν καταγραφεί μια αξιοπρόσεχτη αλλαγή στην τιμή αυτή από τον κόμβο φιλοξενίας του o , η οποία καθορίζεται από ένα δεδομένο κατώφλι ποσοστού b . Αξίζει να σημειωθεί ότι οι τιμές των παραμέτρων a, b ρυθμίζουν την ευαισθησία του μηχανισμού: όσο υψηλότερη είναι η τιμή του a , τόσο μικρότερο θα είναι το μέγεθος του T-Table και συνεπώς ένας μικρότερος αριθμός αντικειμένων θα μπορεί να επωφεληθεί από τον μηχανισμό, αλλά από την άλλη τόσο πιο γρήγορη θα είναι η αναζήτηση στον T-Table. Επιπλέον, όσο υψηλότερη είναι η τιμή του b , τόσο λιγότερες αναπροσαρμογές του T-Table εκτελούνται και έτσι μειώνεται το κόστος λόγω των συχνών ενημερώσεων, αλλά από την άλλη ο μηχανισμός γίνεται λιγότερο ευαίσθητος στις μεταβολές της δημοτικότητας των αντικειμένων.

Υποθέτουμε ότι ένα αντίγραφο του T-Table διατηρείται από κάθε κόμβο του υπερκείμενου δικτύου, σύμφωνα και με την πλήρως κατανεμημένη αρχιτεκτονική που έχει υιοθετηθεί. Στην περίπτωση που ένας κόμβος αποφασίσει να εκτελέσει μια λειτουργία εγγραφής ή ενημέρωσης στον T-Table, η σχετική πληροφορία θα ενσωματωθεί στο αμέσως επόμενο μήνυμα αιτήματος που θα ξεκινήσει από αυτόν τον κόμβο. Σε αυτήν την περίπτωση, το αντίστοιχο ενεργό πακέτο θα πρέπει να «φορτώσει» έναν τροποποιημένο κώδικα, ο οποίος

θα εκτελέσει την ερώτηση αναζήτησης αλλά και τη λειτουργία εγγραφής/ενημέρωσης στον T-Table σε κάθε επισκεπτόμενο κόμβο. Προκειμένου να εξασφαλιστεί ότι οι T-Tables όλων των κόμβων θα ενημερωθούν, ο κόμβος που αποφασίζει την εκτέλεση της λειτουργίας εγγραφής/ενημέρωσης θα πρέπει να επιλέξει ως «μεταφορέα» ένα μήνυμα αιτήματος αναζήτησης για ένα μη-δημοφιλές αντικείμενο, δεδομένου ότι ένα τέτοιο μήνυμα είναι πιο πιθανό να κατακλύσει το δίκτυο και συνεπώς να ενημερώσει όλους τους T-Tables.

5.7.2.2 Πλημμύρα περιορισμένου βάθους (*Depth-limited Flooding*)

Ο *Depth-limited Flooding* προτείνεται ως εναλλακτική λύση του Flooding, η οποία στοχεύει να αποταμιεύσει εύρος ζώνης περιορίζοντας το βάθος της πλημμύρας. Η επιτυχία του αλγορίθμου στην αναζήτηση ενός δεδομένου αντικειμένου βασίζεται σε μια ισχυρή ένδειξη ότι το δεδομένο αντικείμενο είναι αρκετά πιθανό να εντοπιστεί ακολουθώντας ένα περιορισμένο αριθμό βημάτων και οφείλεται στην αποταμίευση εύρους ζώνης που μπορεί να επιτύχει. Κατά συνέπεια, η εμφάνιση γεωγραφικής εντοπιότητας αναφοράς θεωρείται σημαντική ως προς την αποτελεσματική και αποδοτική εφαρμογή του αλγορίθμου, δηλαδή να εντοπιστεί το αντικείμενο που ζητείται στον ίδιο χρόνο όπως με τον Flooding αλλά να περιοριστεί η κατανάλωση εύρους ζώνης. Σημειώνεται ότι για τους σκοπούς του συγκεκριμένου μηχανισμού θεωρούμε μόνο την περίπτωση της εγχώριας γεωγραφικής εντοπιότητας.

Με τον Depth-limited Flooding, σκοπός είναι να εκμεταλλευτούμε την πιθανή συγκέντρωση στο φυσικό χώρο πελατών με τις ίδιες πληροφοριακές ανάγκες, ώστε να βελτιώσουμε την επίδοση της αναζήτησης. Συνεπώς, εάν ένα αντικείμενο επιδεικνύει εγχώρια γεωγραφική εντοπιότητα αναφοράς με ακτίνα L , υπάρχει αυξημένη πιθανότητα το αντικείμενο αυτό να ζητείται από κοντινούς κόμβους στο μέλλον. Σε μια τέτοια περίπτωση, αυτοί οι κοντινοί κόμβοι θα πρέπει γνωρίζουν ότι το συγκεκριμένο αντικείμενο εμφανίζει την ιδιότητα, ώστε να περιορίσουν το βάθος της πλημμύρας σε επόμενα αιτήματα για το δεδομένο αντικείμενο που ξεκινούν από τους ίδιους στην τιμή $D=L$, με την προσδοκία ότι είτε η ίδια η πηγή θα εντοπιστεί κάπου κοντά, είτε η θέση της πηγής θα βρεθεί αποθηκευμένη στον RSD ενός κοντινού κόμβου.

Η εμφάνιση γεωγραφικής εντοπιότητας αναφοράς για ένα δεδομένο αντικείμενο πιστοποιείται από τον κόμβο φιλοξενίας του. Υποθέτοντας ότι κάθε κόμβος διατηρεί αρχεία των αιτημάτων που έχει επιλύσει ο ίδιος, είναι σε θέση να «αποφανθεί» σχετικά με την εμφάνιση της ιδιότητας για τις πηγές που ελέγχει και έχουν ήδη χαρακτηριστεί ως δημοφιλείς. Για το σκοπό αυτό, διατηρεί πληροφορίες σχετικές με την ευρύτερη γειτονιά του, δηλαδή τους κόμβους που βρίσκονται σε κοντινή γεωγραφική απόσταση (μετρούμενη σε αριθμό βημάτων), και εξετάζει περιοδικά το ιστορικό των αιτημάτων για κάθε τοπική πηγή του, ώστε να συσχετίσει τους εναρκτήριους κόμβους αυτών. Στη συνέχεια καταχωρεί κάθε αντικείμενο

που δείχνει να εμφανίζει γεωγραφική εντοπιότητα σύμφωνα με την ανισότητα (3) στον G-Table κάθε ομότιμου κόμβου που ανήκει στην ευρύτερη γειτονιά του, αφού τα επόμενα αιτήματα για το δεδομένο αντικείμενο, τα οποία θα ξεκινούν πιθανότατα από αυτήν την γειτονιά, αξίζει να εκμεταλλευτούν αυτή την πληροφορία. Επιπλέον, η τιμή του T_{home} καθορίζει την επιθυμητή ισορροπία μεταξύ του βαθμού εκμετάλλευσης της ιδιότητας και της επιβάρυνσης που επιφέρεται: όσο μεγαλύτερη είναι η τιμή του T_{home} τόσο λιγότερα αιτήματα θα εκμεταλλευτούν την πληροφορία που διατηρείται στους G-Tables, δεδομένου ότι μόνο τα αντικείμενα που παρουσιάζουν πολύ ισχυρή γεωγραφική εντοπιότητα θα καταχωρούνται στους πίνακες. Ωστόσο, σε αυτή την περίπτωση ο G-Table θα περιέχει λιγότερα αντικείμενα και έτσι ο χρόνος προσπέλασής του θα είναι μικρότερος.

Εάν ένα αντικείμενο καταχωρηθεί σε έναν G-Table, δεν συνεπάγεται ότι η διατήρησή του στον πίνακα θα είναι μόνιμη δεδομένου ότι μια λειτουργία διαγραφής είναι πιθανό να ζητηθεί στο μέλλον από τον κόμβο φιλοξενίας του, ο οποίος περιοδικά ελέγχει την επαλήθευση της ανισότητας (3). Σημειώνεται ότι μια λειτουργία εγγραφής/διαγραφής προτείνεται να εκτελεστεί ενσωματώνοντας τη σχετική πληροφορία σε ένα επόμενο μήνυμα αιτήματος που ξεκινάει από αυτόν τον κόμβο. Σε αυτή την περίπτωση, το αντίστοιχο ενεργό πακέτο θα πρέπει να «φορτώσει» έναν τροποποιημένο κώδικα, ο οποίος θα εκτελεί την ερώτηση αναζήτησης αλλά και τη λειτουργία εγγραφής/διαγραφής, ρυθμίζοντας το TTL της τελευταίας λειτουργίας στην κατάλληλη τιμή ώστε να ενημερωθεί ο G-Table κάθε κόμβου που ανήκει στην ευρύτερη γειτονιά του εναρκτήριου κόμβου, χωρίς να τροποποιηθεί η τιμή TTL του αρχικού μηνύματος αναζήτησης.

5.7.2.3 Πλημμύρα με πρόωρη ανάκτηση (*Prefetch-oriented Flooding*)

Ο *Prefetch-oriented Flooding* εκμεταλλεύεται και ενσωματώνει στον Flooding την ιδέα της πρόωρης ανάκτησης, δηλαδή της αναζήτησης περισσότερων του ενός αντικειμένων κατά τη διάρκεια μιας αναζήτησης, σε αναμονή μελλοντικών αιτημάτων. Το βασικό κέρδος επίδοσης από αυτή την εναλλακτική λύση προέρχεται από το μειωμένο χρόνο που αναμένεται να πετύχει στο κατάλληλο περιβάλλον. Ειδικότερα, η χωρική εντοπιότητα αναφοράς θεωρείται σημαντική για τους σκοπούς της πρόωρης ανάκτησης. Εντούτοις, ο αλγόριθμος αποφεύγει μια πιθανή αρνητική επίδραση στην επίδοση, μόνο εάν η συγκεκριμένη ιδιότητα τεθεί υπό συντηρητική εκμετάλλευση και για το λόγο αυτό θα θεωρήσουμε μόνο την περίπτωση ισχυρών εξαρτήσεων. Διαφορετικά, τόσο ο χρόνος απόκρισης, όσο και το εύρος ζώνης που καταναλώνεται είναι πολύ πιθανό να χειροτερεύσουν.

Τα μοντέλα ζήτησης πληροφορίας που επιδεικνύουν χωρική εντοπιότητα μπορούν να ωφεληθούν από την πρόωρη ανάκτηση κατά τον εξής τρόπο: υποθέτουμε ότι όταν ένας κόμβος αποστέλλει ένα αίτημα για ένα αντικείμενο, το οποίο έχει αποδειχτεί ότι εμφανίζει χωρική εντοπιότητα αναφοράς, θα ενσωματώσει στο μήνυμά του αιτήματος ένα σύνολο

ερωτήσεων για τα εξαρτώμενα αντικείμενα, τα οποία «φαντάζεται» ότι θα ζητηθούν από τον πελάτη στο εγγύς μέλλον, προκειμένου να αναζητηθούν όλα κατά τη διάρκεια της ίδιας πλημμύρας αναζήτησης. Με αυτό τον τρόπο, θα χρειαστεί προφανώς περισσότερος χρόνος για την επίλυση αυτού του σύνθετου αιτήματος, αλλά θα αποφευχθούν νέες πλημμύρες αναζήτησης για τα εξαρτώμενα αντικείμενα, οπότε αναμένεται συνολικά να επιτευχθεί αποταμίευση χρόνου και εύρους ζώνης. Ωστόσο, μια τέτοια εικασία για τις εξαρτήσεις μεταξύ των αντικειμένων θα μπορούσε να κοστίζει σε χρόνο και εύρος ζώνης πολύ ακριβά εάν θεωρήσουμε επίσης τις περιπτώσεις χαλαρών εξαρτήσεων. Επιδιώκοντας λοιπόν να μεγιστοποιήσουμε την πιθανότητα να αποδειχτεί χρήσιμη η πρόωρη ανάκτηση στα πλαίσια μιας αναζήτησης, περιοριζόμαστε να θεωρήσουμε μόνο τις ισχυρές εξαρτήσεις.

Σε αντίθεση με τις προηγούμενες ιδιότητες, η ύπαρξη των οποίων ανιχνεύεται από τη πλευρά του εξυπηρετητή, αφού ο κόμβος φιλοξενίας ενός αντικειμένου καθορίζει εάν το αντικείμενο αυτό εμφανίζει την ιδιότητα, η ανίχνευση της χωρικής ιδιότητας απαιτεί συνολική ανάλυση των σχεδίων ζήτησης πληροφορίας των πελατών. Υποθέτοντας ότι κάθε κόμβος διατηρεί αρχεία με τα αιτήματα που έχει ξεκινήσει, είναι σε θέση να ελέγξει τοπικά την ύπαρξη της ιδιότητας για κάθε αντικείμενο που ζητείται συχνά από τον ίδιο, βάσει της ανισότητας (6). Προκειμένου να εξασφαλιστεί ένας σωστός έλεγχος για την ύπαρξη της ισχυρής εξάρτησης επιλέγουμε μια τιμή για το T_{spat} κοντά στο 1. Σε περίπτωση επαλήθευσης, το υπό εξέταση αντικείμενο καταχωρείται στον S-Table κάθε κόμβου. Σημειώνεται ότι κάθε εγγραφή (o, s_o) του πίνακα δείχνει το αντικείμενο o που εμφανίζει τη χωρική εντοπιότητα και την αντίστοιχη ακολουθία s_o .

Κάθε κόμβος που αποστέλλει συχνά αιτήματα για ένα συγκεκριμένο αντικείμενο, το οποίο είναι καταχωρημένο στον S-Table, είναι υπεύθυνος να επιβεβαιώνει ότι αυτό το αντικείμενο πράγματι παρουσιάζει χωρική εντοπιότητα. Εάν «αποφανθεί» ότι δεν ισχύει η ιδιότητα, θα πρέπει να εκτελέσει μια λειτουργία διαγραφής προκειμένου να αφαιρεθεί η αντίστοιχη εγγραφή από όλους τους S-Tables. Συνεπώς, η διατήρηση ενός αντικειμένου στους S-Tables είναι μια σύνθετη διαδικασία, η οποία απαιτεί τη συγκατάθεση πολλών κόμβων. Όπως και στις προηγούμενες περιπτώσεις, η λειτουργία εγγραφής/διαγραφής εκτελείται από έναν κόμβο ενσωματώνοντας τη σχετική πληροφορία σε επόμενο μήνυμα αιτήματος που αποστέλλεται από αυτόν τον κόμβο.

5.7.2.4 Συνολικά προσαρμοσμένη πλημμύρα με βάση την εντοπιότητα

Οι ιδιότητες εντοπιότητας που επιδεικνύονται από τα σχέδια ζήτησης πληροφορίας μπορούν να χρησιμοποιηθούν προκειμένου να προσαρμοστεί ο αλγόριθμος αναζήτησης κατά τον πιο αποτελεσματικό τρόπο. Ειδικότερα, όταν ένας κόμβος ξεκινά ένα αίτημα συμβουλευεται αρχικά τους πίνακες. Εάν εντοπίσει σε αυτούς ένα αντικείμενο που ικανοποιεί το αίτημά του, προσαρμόζει τον αλγόριθμο αναζήτησης σύμφωνα με την ιδιότητα

εντοπιότητας που επιδεικνύει το αντικείμενο αυτό. Δεδομένου ότι ένα αντικείμενο μπορεί να υπάρχει ταυτόχρονα σε περισσότερους από έναν πίνακες επιδεικνύοντας περισσότερες από μια ιδιότητες εντοπιότητας, οι πίνακες ερευνώνται με συγκεκριμένη σειρά, η οποία καθορίζεται από το μέγεθος του πιθανού κέρδους που επιτυγχάνονται σε κάθε περίπτωση. Πιο συγκεκριμένα, ο S-Table είναι ο πρώτος που διερευνάται αφού ο Prefetch-oriented Flooding είναι ο μόνος που μειώνει το χρόνο, και ο χρόνος είναι το πιο κρίσιμο κριτήριο επίδοσης στις χρονικά-κρίσιμες εφαρμογές που μελετάμε. Σε περίπτωση μη-εντοπισμού, ο G-Table ερευνάται δεύτερος, επειδή η αποταμίευση εύρους ζώνης που επιτυγχάνεται από τον Depth-limited Flooding είναι μεγαλύτερη από αυτή στον Range-limited Flooding, όπως δείχνουν τα αποτελέσματα που παραθέτονται στην επόμενη ενότητα. Συνεπώς, ο T-Table ερευνάται τρίτος κατά σειρά. Στη χειρότερη περίπτωση, δηλαδή εάν το αντικείμενο δεν χαρακτηρίζεται από κάποια ιδιότητα εντοπιότητας, εφαρμόζεται ο απλός Flooding.

5.7.3 ΠΡΟΕΤΟΙΜΑΣΙΑ ΠΡΟΣΟΜΟΙΩΣΗΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΕΙΡΑΜΑΤΩΝ

Σε αυτή την ενότητα παρουσιάζονται τα αποτελέσματα των πειραμάτων που σχετίζονται με την ποσοτικοποίηση του κόστους και του κέρδους των μηχανισμών αναζήτησης που περιγράψαμε παραπάνω. Σαν αναφορά για τη σύγκριση χρησιμοποιούμε τον αλγόριθμο της *πλημμύρας (Flooding)*, ο οποίος πετυχαίνει τον καλύτερο χρόνο εντοπισμού μιας πηγής, υπό το κόστος όμως του υπερβολικού αριθμού μηνυμάτων που κατακλύζουν το δίκτυο. Επίσης, η σύγκριση γίνεται και με τον *τυχαίο περίπατο με τοπική πλημμύρα (Random Walk with Local Flooding)*. Ο υβριδικός αυτός μηχανισμός αποτελεί συνδυασμό του απλού τυχαίου περιπάτου με τοπικές πλημμύρες μικρού βάθους, δηλαδή εκτελεί έναν τυχαίο περίπατο, αλλά σε κάθε βήμα του εφαρμόζει πλημμύρα με TTL=1. Όπως αναφέρθηκε και στην επισκόπηση της βιβλιογραφικής έρευνας, ο μηχανισμός αυτός προτάθηκε πρόσφατα ως παραλλαγή του τυχαίου περιπάτου, η οποία βελτιώνει το χρόνο απόκρισής του [10]. Για λόγους συντομίας στην περιγραφή των αποτελεσμάτων, ο αλγόριθμος αυτός θα αποκαλείται απλά *Random Walk*. Στη συνέχεια, αφού αναφέρουμε αρχικά τις βασικές υποθέσεις που γίνονται για την προσομοίωση, θα παρουσιαστούν τα σχετικά αποτελέσματα.

Η αρχική τοπολογία του υπερκείμενου δικτύου (όπως αυτή διαμορφώνεται από τις συνδέσεις μεταξύ γειτονικών κόμβων) μοντελοποιείται ως τυχαίος γράφος, ο οποίος παράγεται με βάση το μοντέλο Waxman [29]. Αυτό το μοντέλο διανέμει τυχαία τους κόμβους πάνω σε ορθογώνιο πλέγμα. Η ευκλείδεια απόσταση χρησιμοποιείται για τον προσδιορισμό της απόστασης μεταξύ δυο οποιονδήποτε κόμβων. Δυο κόμβοι u , v συνδέονται βάσει πιθανότητας $P(u,v)$, η οποία εξαρτάται από την απόστασή τους ως εξής:

$$P(u,v) = a e^{-\frac{d(u,v)}{L\beta}} \quad (8)$$

όπου το $d(u, v)$ παριστάνει την απόσταση μεταξύ των u και v , το L είναι η μέγιστη απόσταση μεταξύ δυο οποιονδήποτε κόμβων, και α, β είναι παράμετροι που παίρνουν τιμές στο διάστημα $(0, 1)$. Αύξηση της τιμής του α συνεπάγεται αύξηση του αριθμού των ακμών στο γράφο, ενώ αύξηση του β σημαίνει αύξηση της αναλογίας των μακριών ακμών σε σχέση με τις κοντύτερες.

Θα ακολουθήσουμε την *αναλυτική μέθοδο παραγωγής φορτίου αιτημάτων* (*analytic workload generation method*), η οποία θεωρεί συγκεκριμένα μοντέλα προσομοίωσης διαφόρων χαρακτηριστικών του φορτίου (π.χ. τις ιδιότητες εντοπιότητας αναφοράς στην περίπτωση μας), και στη συνέχεια παράγει εξόδους που υπακούουν σε αυτά τα μοντέλα [30]. Σε αντίθεση με τη *μέθοδο παραγωγής φορτίου αιτημάτων βάσει καταγεγραμμένων ιχνών* (*trace-based workload generation method*), η οποία βασίζεται σε αναπαραγωγή ενός καταγεγραμμένου φορτίου αιτημάτων, η μέθοδος που επιλέξαμε να ακολουθήσουμε προσδιορίζει αρχικά τις ιδιότητες εντοπιότητας που θεωρεί σημαντικές για μοντελοποίηση και έχουν παρατηρηθεί εμπειρικά, και στη συνέχεια δημιουργεί εξόδους που χαρακτηρίζονται από αυτές τις ιδιότητες. Δεδομένου ότι καθεμία από τις ιδιότητες που αναφέρθηκαν προτείνεται να χρησιμοποιηθεί με διαφορετικό τρόπο, παράγουμε διαφορετικά αρχεία αιτημάτων για κάθε ιδιότητα προκειμένου να πειραματιστούμε αποκλειστικά με τα πιθανά οφέλη που επιτυγχάνονται σε κάθε περίπτωση.

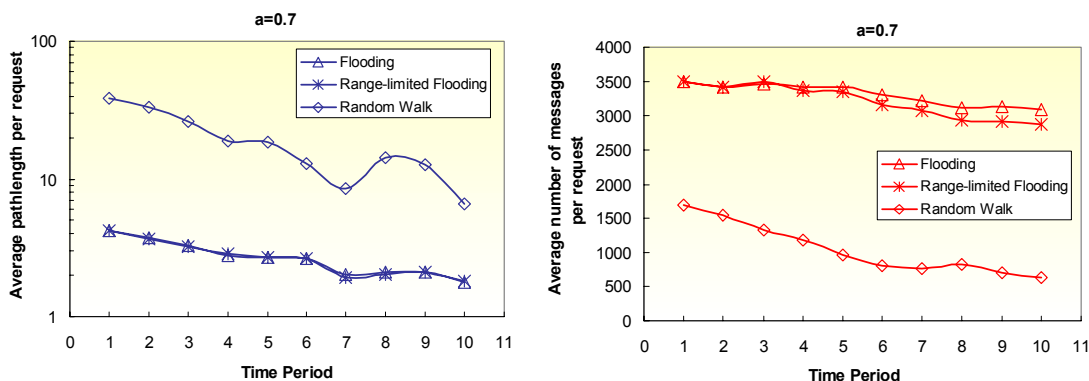
Υποθέτουμε μια ισορροπημένη κατανομή των πηγών, με όλους τους κόμβους να διατηρούν τον ίδιο αριθμό πηγών. Σχετικά με την κατανομή των αιτημάτων, στις πραγματικές εφαρμογές μερικά αντικείμενα είναι πιο δημοφιλή από κάποια άλλα. Η κατανομή *Zipf* αποδεικνύεται ότι χαρακτηρίζει τη δημοτικότητα των αιτημάτων για τα διάφορα αντικείμενα [31] και συνεπώς χρησιμοποιούμε τον κανόνα *Zipf* για να μοντελοποιήσουμε την κατανομή των αιτημάτων πληροφορίας. Σύμφωνα με αυτόν τον κανόνα και θεωρώντας ανεξάρτητα αιτήματα, η σχετική πιθανότητα να προκύψει ένα αίτημα για το i^{th} δημοφιλέστερο αντικείμενο είναι ανάλογη προς $1/i^a$, με τον εκθέτη να παίρνει τιμές στο διάστημα $0 < a \leq 1$. Η τιμή του a μπορεί να είναι διαφορετική για διαφορετικά αρχεία αιτημάτων, δηλαδή η κατανομή των αιτημάτων δεν ακολουθεί τον ακριβή νόμο του *Zipf* (κατά τον οποίο $a=1$), αλλά μια γενικότερη κατανομή *Zipf* με μεταβλητό a . Η πρακτική σημασία αυτής της κατανομής είναι ότι ένας μικρός αριθμός ανεξάρτητων αιτημάτων εμφανίζεται συχνά στο σύνολο των θεωρούμενων αιτημάτων, και ειδικότερα η συγκέντρωση των αιτημάτων στα πιο δημοφιλή αντικείμενα εξαρτάται από την ακριβή τιμή του a . Τέλος, υποθέτουμε απεριόριστο μέγεθος για τον κατάλογο RSD.

5.7.3.1 Δοκιμές Χρονικής Εντοπιότητας

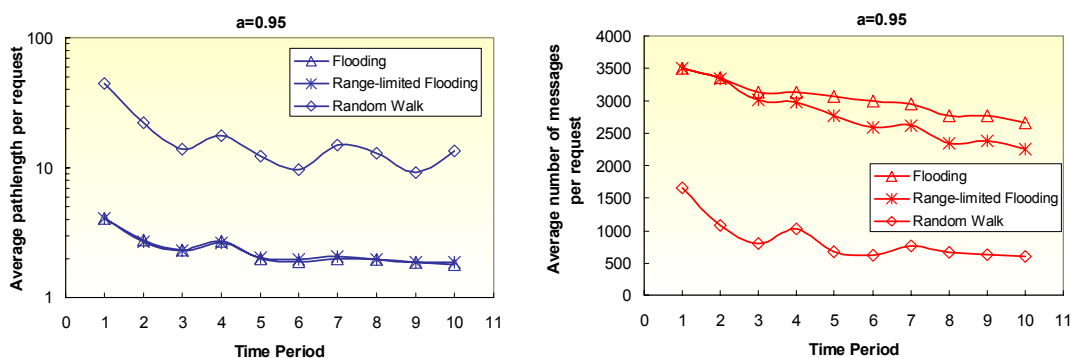
Για τους σκοπούς του παρόντος πειράματος, η γεννήτρια αιτημάτων παράγει ένα τεχνητό αρχείο αιτημάτων το οποίο επιδεικνύει χρονική εντοπιότητα αναφοράς. Όπως έχει

περιγραφεί νωρίτερα, θα εφαρμόσουμε τον Range-limited Flooding, ο οποίος ρυθμίζει το εύρος της πλημμύρας σύμφωνα με τη δημοτικότητα των αντικειμένων. Ένα κρίσιμο θέμα είναι η μεταβολή του εύρους ζώνης που θα μπορούσε να αποταμειωθεί διατηρώντας την ίδια μέση καθυστέρηση εντοπισμού πηγών με τον Flooding, σε σχέση με το ποσοστό ασυμμετρίας στην κατανομή της δημοτικότητας των αντικειμένων που ζητούνται.

Θεωρούμε ένα δίκτυο που αποτελείται από 200 κόμβους και φέρει διάμετρο ίση με 10. Υποθέτουμε ότι κάθε ομότιμος κόμβος ελέγχει μια πηγή, οπότε υπάρχουν συνολικά 200 πηγές στο σύστημα. Κάθε κόμβος ελέγχει τη δημοτικότητα των τοπικών πηγών του και ενημερώνει τους T-Tables ανάλογα. Για λόγους απλότητας στην εκτέλεση του πειράματος θεωρούμε ότι οι T-Tables ενημερώνονται περιοδικά, δηλαδή κάθε φορά που το υπερκείμενο δίκτυο έχει εκτελέσει 100 νέα αιτήματα, για συνολικό αριθμό αιτημάτων ίσο με 1000. Επίσης υποθέτουμε ότι υπάρχει ένα διάστημα «προθέρμανσης» (τα πρώτα 100 αιτήματα), έτσι ώστε η εκτίμηση της δημοτικότητας να είναι αρκετά ακριβής. Τα σχήματα που ακολουθούν απεικονίζουν τη μεταβολή του μέσου μήκους μονοπατιού ανά αίτημα (*average pathlength per request*) και του μέσου αριθμού μηνυμάτων ανά αίτημα (*average number of messages per request*) σε σχέση με το χρόνο, εφαρμόζοντας τον Range-limited Flooding, τον Flooding και τον Random Walk, για δύο διαφορετικές τιμές της Zipf παραμέτρου a . Η τιμή της παραμέτρου TTL ρυθμίζεται στην τιμή της διαμέτρου του δικτύου (TTL=10) για τον Flooding και τον Range-limited Flooding, οπότε το ποσοστό αποτυχίας τους γίνεται αμελητέο. Για να είναι η σύγκριση με τον Random Walk δίκαιη, θέτουμε το TTL για τον συγκεκριμένο αλγόριθμο σε μια τιμή ώστε να δαπανάται περίπου ο ίδιος αριθμός μηνυμάτων με τον Range-limited Flooding. Συνεπώς, για μέσο αριθμό μηνυμάτων στην περίπτωση του Range-limited Flooding ίσο με 3200 (όπως φαίνεται στη συνέχεια) και μέσο βαθμό ενός κόμβου ίσο με 19, θέτουμε το TTL στην τιμή 170. Στην περίπτωση αυτή, το ποσοστό αποτυχίας του Random Walk είναι της τάξης του 2%. Η εξέλιξη του χρόνου δίνεται με βάση διαδοχικά χρονικά διαστήματα, καθένα από τα οποία αντιστοιχεί στην εκτέλεση 100 αιτημάτων.



Σχήμα 5.6: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το χρόνο, για $\alpha=0.7$



Σχήμα 5.7: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το χρόνο, για $\alpha=0.95$

Τα αποτελέσματα δείχνουν ότι το μέσο μήκος μονοπατιού ανά αίτημα γενικά μειώνεται καθώς αυξάνει ο χρόνος, αφού ο αριθμός των αιτημάτων που επαναλαμβάνονται αυξάνει και επιπλέον ο Range-limited Flooding ακολουθεί τον Flooding πολύ στενά. Ομοίως, ο μέσος αριθμός μηνυμάτων ανά αίτημα μειώνεται με το χρόνο, αλλά σε αυτήν την περίπτωση ο Range-limited Flooding υπερτερεί ξεκάθαρα του Flooding, πετυχαίνοντας μικρότερη απώλεια εύρους ζώνης. Επιπλέον, η μείωση του εύρους ζώνης που επιτυγχάνεται από τον πρώτο γίνεται πιο εμφανής καθώς περνάει ο χρόνος, δεδομένου ότι το εύρος της πλημμύρας μειώνεται περαιτέρω λόγω των υψηλότερων τιμών δημοτικότητας που καταγράφονται.

Συγκρίνοντας τον Range-limited Flooding με τον Random Walk παρατηρούμε ότι ο δεύτερος πετυχαίνει μεγάλη αποταμίευση εύρους ζώνης αυξάνοντας όμως σημαντικά το μήκος μονοπατιού. Είναι επίσης χαρακτηριστικό ότι η επίδοση του Random Walk βελτιώνεται αισθητά με το χρόνο. Ειδικότερα, η αποταμίευση εύρους ζώνης που πετυχαίνει σε σχέση με τον Range-limited Flooding αυξάνεται σταδιακά από 53% σε 78%, ενώ η

αποταμίευση μήκους μονοπατιού που πετυχαίνει ο Range-limited Flooding σε σχέση με τον Random Walker μειώνεται σταδιακά από 89% σε 73% (Σχήμα 5.6).

Επιπλέον, είναι φανερό ότι η τιμή της παραμέτρου a επηρεάζει το βαθμό της αποταμίευσης εύρους ζώνης που επιτυγχάνει ο Range-limited Flooding σε σχέση με τον Flooding. Επιλέγοντας τιμή για το a ίση με 0.95 (Σχήμα 5.7) επιβάλλουμε την ύπαρξη μεγάλης ασυμμετρίας στην κατανομή της δημοτικότητας, δηλαδή τα περισσότερα αιτήματα συγκεντρώνονται σε λίγα αντικείμενα, ενώ θέτοντας το a στην τιμή 0.7 (Σχήμα 5.6) σημαίνει ότι η δημοτικότητα των αιτημάτων κατανέμεται πιο ομοιόμορφα, με τα περισσότερα αιτήματα να συγκεντρώνονται σε έναν μεγαλύτερο αριθμό αντικειμένων. Από τις αντίστοιχες γραφικές παραστάσεις γίνεται φανερό ότι όσο μεγαλύτερη είναι η τιμή του a τόσο μεγαλύτερο είναι το χάσμα μεταξύ του Range-limited Flooding και του Flooding, δηλαδή η αποταμίευση εύρους ζώνης που πετυχαίνει ο πρώτος αυξάνεται γρηγορότερα με το χρόνο. Κατά συνέπεια, τα σχέδια ζήτησης πληροφορίας με μεγαλύτερη ασυμμετρία στην κατανομή της δημοτικότητας των αντικειμένων επωφελούνται σε μεγαλύτερο βαθμό από την αποταμίευση εύρους ζώνης που πετυχαίνει ο Range-limited Flooding. Αντίθετα, τα ποσοστά αποταμίευσης εύρους ζώνης και αύξησης μήκους μονοπατιού του Random Walk σε σχέση με τον Range-limited Flooding δείχνουν να μην επηρεάζονται από την τιμή του a .

Τέλος, αξίζει να σημειωθεί ότι οι μικρές τοπικές αυξομειώσεις των γραφημάτων είναι πιο έντονες στην περίπτωση του Random Walk εξαιτίας της επιλεκτικής προώθησης αιτημάτων σε κάθε βήμα, η οποία επιφέρει ελαφρώς μεγαλύτερο ποσοστό αποτυχίας σε σχέση με τους άλλους δυο αλγορίθμους.

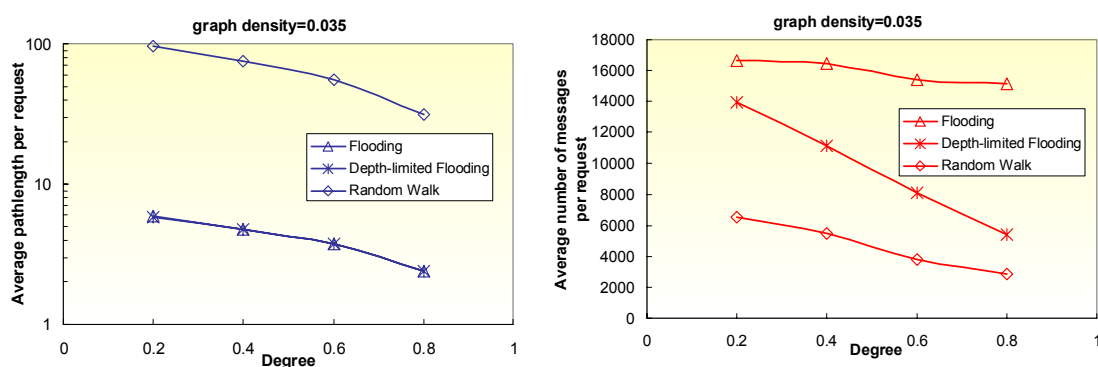
5.7.3.2 Δοκιμές Γεωγραφικής Εντοπιότητας

Στο παρόν πείραμα μεταβάλλουμε το βαθμό γεωγραφικής εντοπιότητας που είναι δυνατόν να εμφανίσει ένα σχέδιο ζήτησης πληροφορίας και καταγράφουμε το κέρδος που προκύπτει από την εφαρμογή του Depth-limited Flooding αντί του Flooding, αλλά και τα αποτελέσματα της σύγκρισης με τον Random Walk. Ορίζουμε τον βαθμό της γεωγραφικής εντοπιότητας ως το ποσοστό του συνολικού αριθμού αιτημάτων που περιλαμβάνονται στο αρχείο, τα οποία επιδεικνύουν γεωγραφική εντοπιότητα αναφοράς. Επιπλέον πειραματιζόμαστε με τυχαίους γράφους διαφορετικής πυκνότητας.

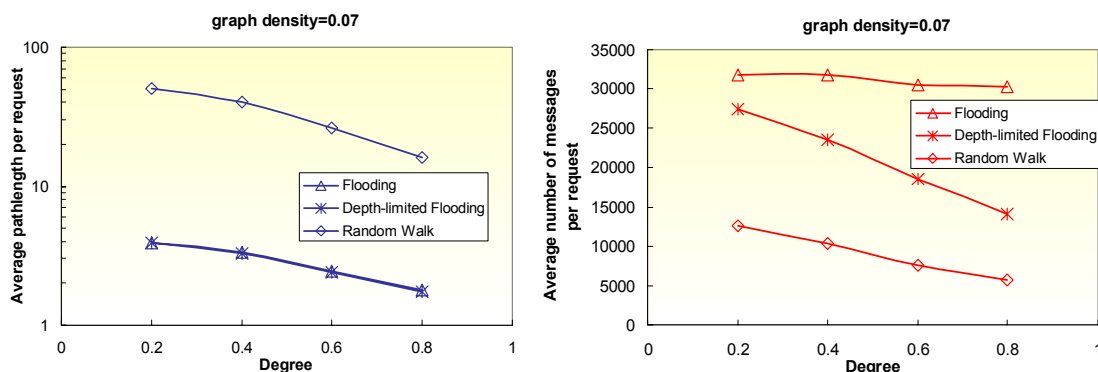
Θεωρούμε αρχικά έναν γράφο των 500 κόμβων και πυκνότητας 0.035. Η διάμετρος του γράφου είναι 17 ενώ ο μέσος βαθμός κόμβου είναι 35. Κάθε κόμβος διατηρεί 10 τοπικές πηγές και συνεπώς υπάρχουν συνολικά 5000 πηγές στο σύστημα. Η γεννήτρια αιτημάτων παράγει έναν αριθμό αρχείων των 400 αιτημάτων σύμφωνα με την κατανομή Zipf (με $a=0.7$), τα οποία επιδεικνύουν γεωγραφική εντοπιότητα σε διαφορετικό βαθμό. Ειδικότερα, ο βαθμός έχει οριστεί στις τιμές 0.2, 0.4, 0.6, και 0.8, με ποσοστό 0.48%, 1.28%, 2.72% και 4.32% των δημοφιλέστερων αντικειμένων να επιδεικνύουν την ιδιότητα, αντίστοιχα. Θα αξιολογήσουμε

την επίδοση του Depth-limited Flooding συγκριτικά με αυτή του Flooding και του Random Walk, υποθέτοντας ότι ο πρώτος έχει θέσει την ακτίνα της εγχώριας γεωγραφικής εντοπιότητας στην τιμή 3. Τα σχετικά αποτελέσματα απεικονίζονται στο Σχήμα 5.8. Στη συνέχεια επαναλαμβάνουμε το πείραμα για γράφο πυκνότητας 0.07, διαμέτρου 10 και μέσου βαθμού κόμβου 69. Τα αντίστοιχα αποτελέσματα φαίνονται στο Σχήμα 5.9. Σημειώνεται ότι η τιμή του TTL έχει προσδιοριστεί όπως και στην προηγούμενη περίπτωση για κάθε αλγόριθμο. Δηλαδή για τον Flooding και τον Depth-limited Flooding έχουμε ορίσει το TTL ίσο με τη διάμετρο του δικτύου οπότε το ποσοστό αποτυχίας είναι αμελητέο, ενώ στη περίπτωση του Random Walk έχουμε θέσει το TTL στην τιμή 290, ώστε ο μέγιστος αριθμός μηνυμάτων να είναι περίπου ίδιος με του Depth-limited Flooding. Στην τελευταία αυτή περίπτωση το ποσοστό αποτυχίας κυμαίνεται μεταξύ 6%-16% στον αραιό γράφο, ενώ στον πυκνό είναι της τάξης του 2%.

Σύμφωνα με τα σχήματα, καθώς ο βαθμός γεωγραφικής εντοπιότητας αυξάνει το μέσο μήκος μονοπατιού ανά αίτημα μειώνεται. Αυτό οφείλεται στο γεγονός ότι ξεκινούν όλο και περισσότερα επαναλαμβανόμενα αιτήματα από κόμβους που βρίσκονται κοντά στην πηγή που ικανοποιεί το αίτημα. Επιπλέον, ο Depth-limited Flooding καταφέρνει να διατηρήσει το μήκος μονοπατιού για τον εντοπισμό της κατάλληλης πηγής στις ίδιες τιμές με τον Flooding. Από την άλλη μεριά, ενώ ο μέσος αριθμός μηνυμάτων ανά αίτημα παραμένει σχεδόν αμετάβλητος στην περίπτωση του Flooding καθώς ο βαθμός γεωγραφικής εντοπιότητας αυξάνει, ο Depth-limited Flooding πετυχαίνει αυξανόμενες αποταμιεύσεις. Ο λόγος είναι ότι ο δεύτερος καταφεύγει σε περιορισμό του βάθους της πλημμύρας όλο και πιο συχνά καθώς ο βαθμός γεωγραφικής εντοπιότητας αυξάνει.



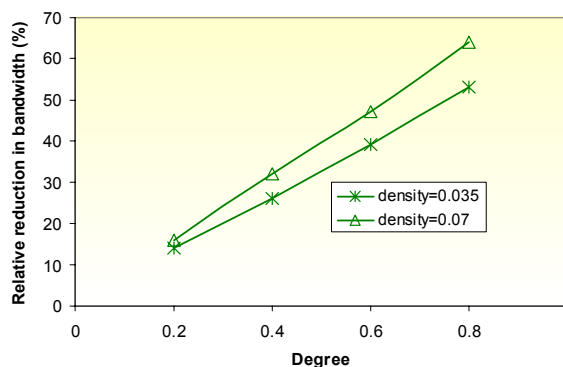
Σχήμα 5.8: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το βαθμό γεωγραφικής εντοπιότητας, για πυκνότητα δικτύου ίση με 0.035



Σχήμα 5.9: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το βαθμό γεωγραφικής εντοπιότητας, για πυκνότητα δικτύου ίση με 0.07

Συγκρίνοντας τον Depth-limited Flooding με τον Random Walk, παρατηρούμε ότι η αποταμίευση εύρους ζώνης που πετυχαίνει ο δεύτερος σε σχέση με τον πρώτο είναι της τάξης του 50% και παραμένει σχεδόν σταθερή σε μεταβολές του βαθμού γεωγραφικής εντοπιότητας. Αντίστοιχα, η αποταμίευση μήκους μονοπατιού που πετυχαίνει ο Depth-limited Flooding σε σχέση με τον Random Walk είναι της τάξης του 90% και επίσης δεν δείχνει να επηρεάζεται από το βαθμό της ιδιότητας (Σχήμα 5.8).

Ενώ επίσης η πυκνότητα του γράφου δεν επηρεάζει τα παραπάνω ποσοστά που αφορούν στην επίδοση του Depth-limited Flooding σε σχέση με αυτή του Random Walk, δεν ισχύει το ίδιο όσον αφορά τον Flooding (Σχήμα 5.9). Το Σχήμα 5.10 παριστάνει τη σχετική μείωση του μέσου αριθμού μηνυμάτων ανά αίτημα που πετυχαίνει ο Depth-limited Flooding ως προς τον Flooding σε σχέση με το βαθμό γεωγραφικής εντοπιότητας, για τις δυο διαφορετικές τιμές της πυκνότητας δικτύου. Είναι φανερό ότι η σχετική μείωση αυξάνει σχεδόν γραμμικά με το βαθμό της ιδιότητας και επιπλέον ο ρυθμός αύξησης εξαρτάται από την πυκνότητα του δικτύου. Πιο συγκεκριμένα, όσο πυκνότερο είναι το δίκτυο, τόσο πιο επικερδής θα είναι η εφαρμογή του Depth-limited Flooding σε σχέση με του Flooding.



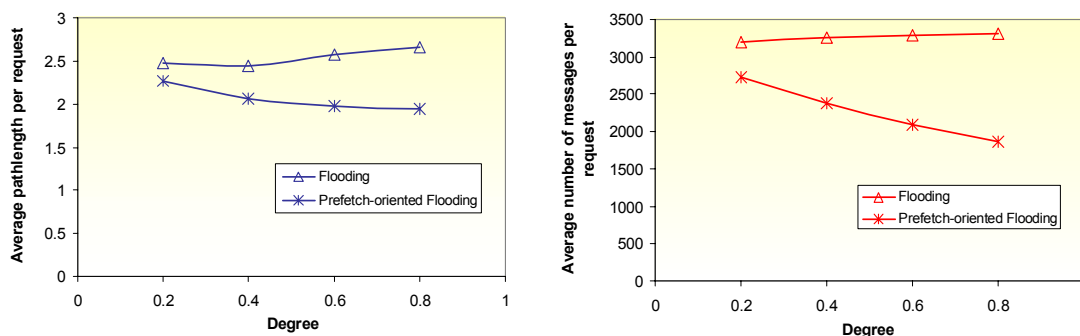
Σχήμα 5.10: Σχετική μείωση εύρους ζώνης σε σχέση με το βαθμό γεωγραφικής εντοπιότητας, για διαφορετικές πυκνότητες δικτύου

5.7.3.3 Δοκιμές Χωρικής Εντοπιότητας

Σκοπός του τρίτου πειράματος είναι να ποσοτικοποιήσουμε το κόστος του Prefetch-oriented Flooding σε σχέση με το βαθμό χωρικής εντοπιότητας που επιδεικνύεται από ένα σχέδιο ζήτησης πληροφορίας. Όπως και στο προηγούμενο πείραμα, ορίζουμε το βαθμό χωρικής εντοπιότητας ως το ποσοστό του συνολικού αριθμού αιτημάτων, τα οποία επιδεικνύουν χωρική εντοπιότητα.

Ο γράφος του δικτύου αποτελείται από 200 κόμβους, καθένας από τους οποίους διατηρεί 1 πηγή, οπότε συνολικά το σύστημα διαθέτει 200 πηγές. Η διάμετρος του δικτύου είναι ίση με 10. Η γεννήτρια αιτημάτων παρήγαγε 4 διαφορετικά συνθετικά αρχεία αιτημάτων σύμφωνα με την κατανομή Zipf ($a=0.7$), καθένα από τα οποία περιλαμβάνει 1000 αιτήματα. Ο βαθμός χωρικής εντοπιότητας σε καθένα από αυτά θα ρυθμιστεί στην τιμή 0.2, 0.4, 0.6 και 0.8, αντίστοιχα, με το 2.5%, 10%, 26% και 54.5% των δημοφιλέστερων αντικειμένων να επιδεικνύουν την ιδιότητα, αντίστοιχα. Εάν θεωρήσουμε ότι ακολουθίες 2 αιτημάτων προκαλούν την ιδιότητα, αυτό επιτυγχάνεται ως εξής: προκειμένου να διατηρηθεί αναλλοίωτο όσο το δυνατόν περισσότερο το προφίλ δημοτικότητας των διαθέσιμων αντικειμένων και συγχρόνως να απεικονιστεί η επιθυμητή χωρική εντοπιότητα αναφοράς των αντικειμένων, αλλάζουμε τα αρχεία αιτημάτων που παρήχθησαν αρχικά, αναμειγνύοντας το σύνολο των 1000 αιτημάτων που είχαν κατανεμηθεί σύμφωνα με τον κανόνα του Zipf με έναν αριθμό αιτημάτων, τα οποία συσχετίζονται με αυτά που θεωρούμε ότι επιδεικνύουν χωρική εντοπιότητα. Αυτό έχει σαν αποτέλεσμα ο συνολικός αριθμός των διαθέσιμων αντικειμένων και αιτημάτων που εκτελούνται να είναι τελικά μεγαλύτερος από αυτόν που είχε υπολογιστεί αρχικά. Θεωρώντας τα ίδια κριτήρια επίδοσης με τα προηγούμενα πειράματα, οι γραφικές παραστάσεις που προκύπτουν απεικονίζονται στο Σχήμα 5.11.

Τα αποτελέσματα δείχνουν ότι ενώ ο απλός Flooding δεν επηρεάζεται αισθητά από τη μεταβολή του βαθμού της χωρικής εντοπιότητας, ο Prefetch-oriented Flooding επιτυγχάνει σημαντική αποταμίευση, τόσο από πλευράς μήκους μονοπατιού όσο και αριθμού μηνυμάτων ανά αίτημα, η οποία γίνεται όλο και πιο εμφανής καθώς ο βαθμός αυξάνεται. Ειδικότερα, η αποταμίευση μήκους μονοπατιού φτάνει έως και 27%, ενώ η αποταμίευση αριθμού μηνυμάτων έως και 44%.



Σχήμα 5.11: Μέσο μήκος μονοπατιού και αριθμός μηνυμάτων ανά αίτημα σε σχέση με το βαθμό χωρικής εντοπιότητας

5.8 ΜΗΧΑΝΙΣΜΟΙ ΑΝΑΖΗΤΗΣΗΣ ΜΕ ΧΡΗΣΗ ΔΙΑΦΗΜΙΣΕΩΝ

Οι *προδραστικοί μηχανισμοί (proactive mechanisms)* αναφέρονται στους μηχανισμούς που εφαρμόζονται πριν από την εκτέλεση ενός αιτήματος αναζήτησης πληροφορίας προκειμένου να βελτιωθεί η επίδοση της αναζήτησης. Ένας τέτοιος μηχανισμός είναι η *διανομή διαφημίσεων*, σύμφωνα με τον οποίο μόλις μια νέα πηγή εισέλθει στο σύστημα, ο κόμβος φιλοξενίας της μεταδίδει μια *διαφήμιση* της πηγής σε όλους τους άλλους κόμβους του υπερκείμενου δικτύου σύμφωνα με έναν απλό αλγόριθμο πλημμύρας. Η διαφήμιση περιλαμβάνει πληροφορία που περιγράφει την πηγή καθώς και πληροφορία σχετική με τη θέση του κόμβου φιλοξενίας της. Λαμβάνοντας μια διαφήμιση, ένας κόμβος μπορεί να αποθηκεύσει την πληροφορία της διαφήμισης στον RSD του. Εάν υποθέσουμε ότι το μέγεθος του RSD είναι απεριόριστο, κάθε κόμβος μπορεί να αποθηκεύει κάθε διαφήμιση που λαμβάνει. Σε αυτήν την περίπτωση, κάθε κόμβος διατηρεί γνώση για όλες τις διαθέσιμες πηγές του συστήματος και έτσι μπορεί να εντοπίσει αμέσως την κατάλληλη πηγή για κάθε αίτημα. Από υπολογιστική πλευρά όμως, μπορεί γρήγορα να αποτελέσει σημείο συμφόρησης εξαιτίας της διαμόρφωσης μεγάλων καταλόγων, οι οποίοι απαιτούν χρονοβόρα αναζήτηση. Επιπλέον, οι ομότιμοι κόμβοι μπορεί να μην είναι διατεθειμένοι να αφιερώσουν μεγάλους αποθηκευτικούς χώρους για να αποθηκεύσουν δεδομένα ή και μετα-δεδομένα άλλων κόμβων. Εν τω μεταξύ, καθώς η κλίμακα του συστήματος αυξάνει με την εισαγωγή νέων πηγών, το μήκος των καταλόγων μπορεί να αυξηθεί ανεξέλεγκτα. Καθοδηγούμενοι από αυτές τις παρατηρήσεις, πρόθεσή μας είναι να περιορίσουμε την κατάσταση που διατηρείται από κάθε κόμβο, δηλαδή τις αποθηκευτικές απαιτήσεις του, διατηρώντας ταυτόχρονα μικρό το μήκος μονοπατιού οποιουδήποτε μηνύματος που μεταφέρει ένα αίτημα αναζήτησης.

Η ιδέα που θα χρησιμοποιήσουμε βασίζεται στην εφαρμογή του *μοντέλου μικρού-κόσμου* όσον αφορά την αποθήκευση των διαφημίσεων στους RSDs, έτσι ώστε να διατηρείται περιορισμένη κατάσταση σε κάθε κόμβο και με κατάλληλη αναζήτηση να εντοπίζεται

γρήγορα οποιαδήποτε πηγή. Η επιλεκτική αποθήκευση διαφημίσεων στους RSDs αποσκοπεί στο να διαμορφωθούν οι κατάλογοι με τέτοιο τρόπο ώστε οι κόμβοι που διατηρούν πηγές με παρόμοιο περιεχόμενο να συνεργάζονται μεταξύ τους προκειμένου να επιλύσουν ένα αίτημα σε λογικό χρόνο. Πιο συγκεκριμένα, μόλις ληφθεί ένα μήνυμα αιτήματος αναζήτησης από έναν κόμβο, ο οποίος δε διαθέτει τοπική πηγή που να ικανοποιεί το αίτημα αλλά ούτε γνωρίζει για κάποια απομακρυσμένη πηγή, αποστέλλει το μήνυμα σε ένα γειτονικό στο χώρο του περιεχομένου κόμβο, δηλαδή σε κόμβο που διαθέτει πηγή παρόμοιου περιεχομένου με αυτή που θα ικανοποιούσε το αίτημα, όπως του υποδεικνύει η τοπική γνώση του [32, 33]. Η προσέγγιση που θα ακολουθήσουμε αρχίζει από τη μελέτη ενός θεωρητικού μοντέλου μικρού-κόσμου, στο οποίο βασίζεται στη συνέχεια η σχεδίαση του προτεινόμενου πρωτοκόλλου.

Πολλές αρχιτεκτονικές ανακάλυψης υπηρεσιών (service discovery architectures), όπως για παράδειγμα η Jini [34], η Salutation και Salutation-lite [35], η UPnP [36], και η Service Location Protocol [37] έχουν αναπτυχθεί, με σκοπό την αποδοτική ανακάλυψη υπηρεσιών μέσω ενσύρματων και ασύρματων πλατφόρμων. Οι περισσότερες από αυτές βασίζονται σε ένα κεντρικό εξυπηρετητή, στον οποίο εγγράφονται όλες οι υπηρεσίες, και συνεπώς παρουσιάζουν προβλήματα επεκτασιμότητας. Η διανομή διαφημίσεων ανέρχεται τελευταία ως ενδιαφέρουσα ιδέα, η οποία προτείνεται και για περιβάλλοντα διεισδυτικής υπολογιστικής [38, 39], χωρίς όμως να έχει διερευνηθεί η αντιμετώπιση των προβλημάτων επίδοσης που εμφανίζει. Από την άλλη, ο Zhang [40] βασίστηκε στην ιδέα του μοντέλου μικρού-κόσμου και τα θεωρητικά αποτελέσματα που είχαν βρεθεί ως τότε από τον Kleinberg [41] και πρότεινε ένα σχέδιο αντικατάστασης αποθηκών ενισχυμένης συγκέντρωσης για το Freenet, το οποίο δημιουργεί στους πίνακες δρομολόγησης σχέσεις γειτόνων βάσει ενός γράφου γνωριμιών μικρού-κόσμου, επιβάλλοντας ομαδοποίηση με ελαφρά τυχαιότητα.

Στην παρούσα μελέτη ασχολούμαστε με την αξιοποίηση του μοντέλου μικρού-κόσμου στη διανομή των διαφημίσεων. Ο προτεινόμενος μηχανισμός βασίζεται σε ένα πιο πρόσφατο μοντέλο του Kleinberg, το *ιεραρχικό μοντέλο δικτύου (hierarchical network model)*, το οποίο θα περιγραφεί στην επόμενη ενότητα.

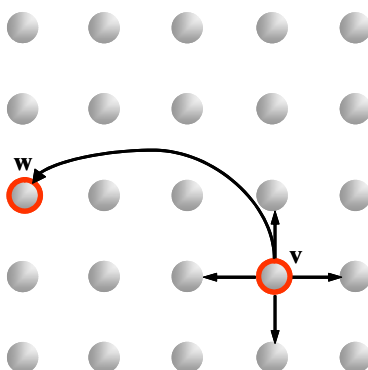
5.8.1 ΜΟΝΤΕΛΟ ΜΙΚΡΟΥ-ΚΟΣΜΟΥ

Το φαινόμενο του μικρού-κόσμου, η αρχή δηλαδή ότι όλοι οι άνθρωποι συνδεόμαστε μέσω μικρών αλυσίδων γνωστών, εγκαινιάστηκε ως τομέας πειραματικής μελέτης στις κοινωνικές επιστήμες με την πρωτοποριακή εργασία του Stanley Milgram στη δεκαετία του '60 [42, 43, 44]. Ο Milgram επιδίωξε να καθορίσει εάν τα περισσότερα ζευγάρια ανθρώπων στην κοινωνία συνδέονται μέσω κοινών γνωστών, και για το λόγο αυτό ανέθεσε σε έναν άνθρωπο στη Νεμπράσκα να μεταφέρει μια επιστολή σε έναν άλλο άνθρωπο στη Μασαχουσέτη. Στον αρχικό αποστολέα δόθηκαν οι βασικές πληροφορίες για το στόχο,

δηλαδή το όνομά του, η διεύθυνση, το επάγγελμα και μερικές άλλες προσωπικές λεπτομέρειες, και έπρεπε να επιλέξει τον αμέσως επόμενο άνθρωπο μεταξύ των γνωστών του, στον οποίο θα προωθούσε την επιστολή, ώστε η επιστολή να φτάσει στον τελικό παραλήπτη όσο το δυνατόν γρηγορότερα. Οι επόμενοι παραλήπτες ακολούθησαν την ίδια διαδικασία, και η αλυσίδα επικοινωνίας ολοκληρώθηκε όταν η επιστολή έφτασε στον προορισμό. Ο μέσος αριθμός βημάτων μιας αλυσίδας που καταλήγει στον προορισμό βρέθηκε ότι είναι έξι, από το οποίο προέκυψε και η αρχή των «έξι βαθμών συγγένειας».

Από τότε, διάφορα μοντέλα δικτύου έχουν προταθεί για την αναλυτική μελέτη του προβλήματος. Το πιο γνωστό από αυτά διατυπώθηκε από τους Watts και Strogatz, οι οποίοι αποδείξανε ότι το φαινόμενο του μικρού-κόσμου ισχύει σε ένα σύνολο φυσικών και τεχνολογικών δικτύων και αποτελεί θεμελιώδες συστατικό στην εξέλιξη του παγκόσμιου ιστού [45, 46, 47]. Σύμφωνα με αυτούς, η δομή ενός δικτύου που προκαλεί συμπεριφορά μικρού κόσμου απαιτεί κάθε κόμβος στο δίκτυο να γνωρίζει τους φυσικούς του γείτονες, καθώς επίσης και έναν μικρό αριθμό τυχαία επιλεγμένων απομακρυσμένων κόμβων. Ειδικότερα, αρχίζοντας από ένα p -διάστατο δικτυωτό πλέγμα, στο οποίο οι κόμβοι συνδέονται μόνο με τους κοντινότερους γείτονές τους, προστίθενται στη συνέχεια k κατευθυνόμενες συνδέσεις μεγάλης ακτίνας από κάθε κόμβο v , για k σταθερό και το άλλο άκρο κάθε σύνδεσης να επιλέγεται τυχαία. Σύμφωνα με τη θεωρία τυχαίων γράφων, αποδεικνύεται ότι με μεγάλη πιθανότητα θα υπάρχουν μικρά μονοπάτια που συνδέουν όλα τα ζευγάρια των κόμβων [48], των οποίων τα μήκη θα περιορίζονται από ένα πολυώνυμο του $\log n$ (πολυλογαριθμικό), όπου n ο αριθμός κόμβων του δικτύου. Ο Kleinberg απέδειξε όμως ότι γι' αυτό το μοντέλο δικτύου δεν υπάρχει κατανεμημένος αλγόριθμος που να μπορεί να υπολογίζει μονοπάτια μικρού μέσου μήκους [41]. Προσδιόρισε στη συνέχεια ένα σύνολο μοντέλων δικτύου που αποτελούν γενίκευση του προηγούμενου και απέδειξε ότι υπάρχει ακριβώς ένα μοντέλο μεταξύ αυτών, για το οποίο ένας κατανεμημένος αλγόριθμος μπορεί να βρίσκει μικρά μονοπάτια με μεγάλη πιθανότητα. Σύμφωνα με αυτό το μοντέλο, στη σύνδεση μεγάλης ακτίνας (v, w) από τον κόμβο v , ο κόμβος w δεν επιλέγεται τυχαία αλλά με πιθανότητα ανάλογη προς d^{-a} , όπου d η απόσταση στο δικτυωτό πλέγμα από τον v στον w . Με αυτό τον τρόπο, οι συνδέσεις μεγάλης ακτίνας συσχετίζονται με τη γεωμετρία του δικτυωτού πλέγματος. Αποδεικνύεται πως όταν το a γίνει ίσο με p , όπου p η διάσταση του δικτυωτού πλέγματος, ένας κατανεμημένος *άπληστος αλγόριθμος* μπορεί να κατασκευάσει μονοπάτια μήκους πολυωνυμικού του $\log n$, επιλέγοντας σε κάθε βήμα τον κόμβο που βρίσκεται όσο το δυνατόν πιο κοντά στο στόχο, από πλευράς απόστασης δικτυωτού πλέγματος, ενώ για κάθε άλλη τιμή του a δεν υπάρχει κατανεμημένος αλγόριθμος με πολυλογαριθμικό χρόνο αναζήτησης. Στην περίπτωση του διδιάστατου δικτυακού πλέγματος, εάν κάθε κόμβος συνδέεται με όλους του κοντινότερους γείτονές του και με έναν μακρινό του κόμβο με

πιθανότητα ανάλογη προς d^2 , τότε ο μέσος χρόνος αναζήτησης του αλγορίθμου είναι $a_2 \log^2 n$, όπου a_2 σταθερά ανεξάρτητη του n (Σχήμα 5.12).



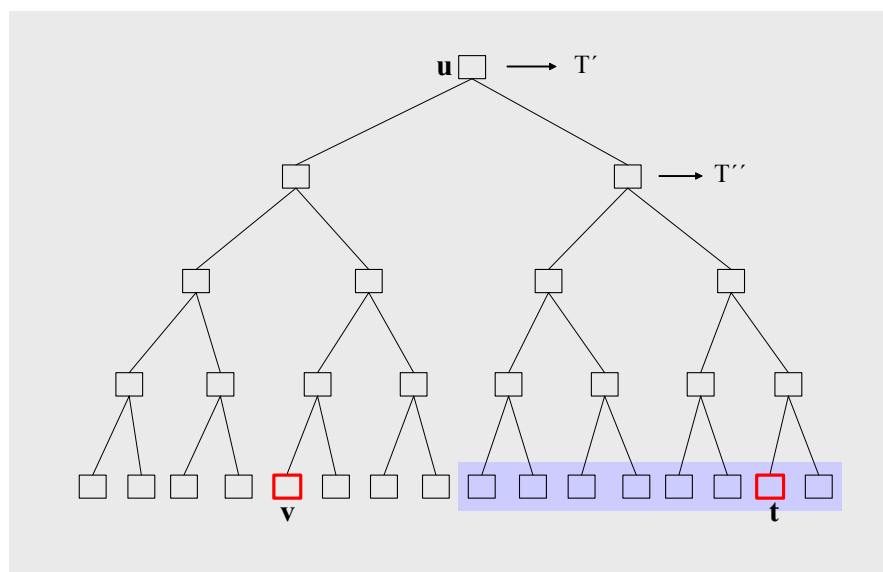
Σχήμα 5.12: Γειτονικές και μεγάλης ακτίνας συνδέσεις του κόμβου v στο διδιάστατο πλέγμα

Ο Kleinberg διερεύνησε κατά πόσο οι αρχές της κατανεμημένης αναζήτησης θα μπορούσαν να εφαρμοστούν και σε δίκτυα που δεν έχουν τη δομή πλέγματος. Για το λόγο αυτό μελέτησε δίκτυα τα οποία μπορούν να παραχθούν από μια ιεραρχική δομή σύμφωνα με την θεωρία που ακολουθεί [49]:

Σε αρκετές περιπτώσεις οι κόμβοι αντιπροσωπεύουν αντικείμενα, τα οποία μπορούν να ταξινομηθούν σύμφωνα με μια ιεραρχία. Οι κόμβοι είναι πιθανότερο να συνδέονται μεταξύ τους εάν ανήκουν στο ίδιο μικρό υποδένδρο της ιεραρχίας, υποδεικνύοντας ότι συσχετίζονται περισσότερο μεταξύ τους. Για να κατασκευάσουμε ένα μοντέλο δικτύου από αυτή την ιδέα, αναπαριστάμε την ιεραρχία αντικειμένων με ένα πλήρες b -ary δέντρο T , όπου το b είναι μια σταθερά. Έστω V το σύνολο φύλλων του T , και n το μέγεθος του V , και για δύο φύλλα v και w , έστω $h(v,w)$ το ύψος του πιο κοντινού κοινού προγόνου τους στο T . Δίνεται επίσης μια μονότονη, μη-αυξανόμενη συνάρτηση $f(\cdot)$, η οποία θα καθορίσει τις πιθανότητες συνδέσεων. Για κάθε κόμβο $v \in V$, δημιουργούμε μια τυχαία σύνδεση με το w με πιθανότητα ανάλογη προς $f(h(v,w))$. Με άλλα λόγια, η πιθανότητα επιλογής του w είναι ίση προς $f(h(v,w)) / \sum_{x \neq w} f(h(v,x))$. Με αυτό τον τρόπο, δημιουργούμε k συνδέσεις από κάθε κόμβο v , επιλέγοντας το άλλο άκρο w κάθε φορά ανεξάρτητα και επιτρέποντας την επανάληψη. Έτσι προκύπτει ο γράφος G από το σύνολο V . Αξίζει να σημειωθεί ότι το δένδρο T χρησιμοποιείται μόνο κατά τη διαδικασία παραγωγής του G . Ούτε οι συνδέσεις, αλλά ούτε οι εσωτερικοί κόμβοι του T εμφανίζονται στον γράφο. Αναφερόμαστε στη διαδικασία παραγωγής του G ως *ιεραρχικό μοντέλο με εκθέτη α* (*hierarchical model with exponent α*) εάν η $f(h)$ εξελίσσεται ασυμπτωτικά ως $b^{-\alpha h}$:

$$\lim_{h \rightarrow \infty} \frac{f(h)}{b^{-\alpha h}} = 0 \quad \text{για } \alpha' < \alpha \quad \text{και} \quad \lim_{h \rightarrow \infty} \frac{b^{-\alpha'' h}}{f(h)} = 0 \quad \text{για } \alpha'' > \alpha \quad (9)$$

Θεωρούμε τώρα το πρόβλημα αναζήτησης σε ένα γράφο G που παράγεται από ένα ιεραρχικό μοντέλο: Ένας καταναμημένος αλγόριθμος έχει επίγνωση του δένδρου T , και γνωρίζει επίσης τη θέση του φύλλου που αντιστοιχεί στο αντικείμενο που αναζητά. Ωστόσο, μαθαίνει τη δομή του G καθώς επισκέπτεται τους κόμβους. Ο εκθέτης a καθορίζει τον τρόπο που συσχετίζεται η δομή του G με αυτή του T . Αποδεικνύεται ότι στο ιεραρχικό μοντέλο με εκθέτη $a=1$ και $k=c \cdot \log_b^2 n$, ένας καταναμημένος αλγόριθμος αναζήτησης μπορεί να πετύχει χρόνο $O(\log_b n)$. Ο αλγόριθμος αυτός λειτουργεί ως εξής: Θεωρούμε έναν κόμβο v και τον κόμβο-στόχο t (Σχήμα 5.13). Έστω ότι ο πιο κοντινός κοινός τους πρόγονος στο T είναι ο $u \in T$ στο ύψος i και T' το υποδένδρο του T με αρχή το u . Έστω επίσης T'' το υποδένδρο του T' με ύψος $i-1$ που περικλείει το t (ενώ $v \notin T''$). Λέμε ότι ο v είναι *χρήσιμος* όσον αφορά τον t εάν υπάρχει μια απευθείας σύνδεση (στον G) από τον v σε κάποιον κόμβο του T'' . Εάν όλοι οι κόμβοι του G είναι χρήσιμοι, τότε από οποιοδήποτε αρχικό κόμβο s , μπορούμε να κατασκευάσουμε μονοπάτι μέγιστου μήκους $\log_b n$ προς τον t ως εξής: από τον τρέχοντα κόμβο v , επιλέγεται μια σύνδεση στο T'' όπως περιγράφηκε νωρίτερα και η διαδικασία επαναλαμβάνεται με τον ίδιο τρόπο, μέχρι να φτάσουμε στον t .



Σχήμα 5.13: Πλήρες b -ary δένδρο ($b=2$) για τους σκοπούς του ιεραρχικού μοντέλου δικτύου

5.8.2 ΜΗΧΑΝΙΣΜΟΙ ΑΝΑΖΗΤΗΣΗΣ

Η περιορισμένη κατάσταση που απαιτούμε να διατηρείται σε κάθε ομότιμο κόμβο δημιουργεί την ανάγκη εφαρμογής ενός σχεδίου αντικατάστασης εγγραφών στον RSD μόλις αυτός γεμίσει. Ένα απλό σχέδιο υπαγορεύει τυχαία αντικατάσταση, αλλά μια τέτοια απλότητα αναμένεται να υποβαθμίσει την επίδοση του συστήματος από πλευράς χρόνου αναζήτησης οποιασδήποτε πληροφορίας. Υποκινούμενοι από το φαινόμενο του μικρού-κόσμου και ειδικότερα από το ιεραρχικό μοντέλο δικτύου προτείνουμε ένα μηχανισμό

διανομής διαφημίσεων, ο οποίος εξελίσσεται με τρόπο ώστε η πληροφορία που διατηρείται στους RSDs να απεικονίζει τις συνδέσεις του γράφου που παράγεται από το ιεραρχικό μοντέλο, με πλήρως καταναμημένο τρόπο.

Υποθέτουμε ότι οι διαθέσιμες πηγές πληροφορίας ταξινομούνται σύμφωνα με μια ιεραρχία, βάσει των ονομάτων των αντίστοιχων αντικειμένων πληροφορίας [50]. Κατά συνέπεια, κάθε πηγή αντιστοιχεί σε ένα συγκεκριμένο φύλλο ενός πλήρους b -ary δένδρου πληροφορίας (*context tree*). Αξίζει να σημειωθεί ότι περισσότερες από μια πηγές είναι δυνατόν να αντιστοιχούν στο ίδιο φύλλο του δένδρου, δεδομένου ότι πολλές πηγές μπορούν να παράγουν αντικείμενα πληροφορίας του ίδιου ονόματος αλλά διαφορετικών ιδιοτήτων. Με βάση το γράφο που παράγεται από το ιεραρχικό μοντέλο, η ιδέα που ακολουθείται είναι η εξής: για κάθε τοπική πηγή που είναι καταχωρημένη σε έναν LSD και αντιστοιχεί σε ένα συγκεκριμένο κόμβο του παραγόμενου γράφου, οι εγγραφές των πηγών που αντιστοιχούν στο άλλο άκρο των ακμών που ξεκινούν από αυτόν τον κόμβο θα πρέπει να διατηρούνται στον αντίστοιχο RSD. Με αυτόν τον τρόπο, οι RSDs περιέχουν πληροφορία για εκείνες τις πηγές, οι οποίες διαμορφώνουν μια σχετική συγκέντρωση με ελαφρά τυχαιότητα γύρω από κάθε τοπική πηγή και συνεπώς κάθε αντικείμενο που αναζητείται αναμένεται να εντοπιστεί σε περιορισμένο αριθμό βημάτων. Η λεπτομερής περιγραφή της *διανομής διαφημίσεων* και του *αλγορίθμου αναζήτησης* δίνονται στις ενότητες που ακολουθούν.

5.8.2.1 Προεπεξεργασία: Διανομή Διαφημίσεων

Μόλις εισέλθει μια νέα πηγή στο υπερκείμενο δίκτυο εισάγοντας μια καταχώρηση στον LSD του κόμβου φιλοξενίας της, μεταδίδεται μια διαφήμιση της πηγής σε όλους τους άλλους κόμβους, σύμφωνα με έναν απλό αλγόριθμο πλημμύρας. Ένας κόμβος αποθηκεύει μια λαμβανόμενη διαφήμιση εάν υπάρχει χώρος στον RSD του και επιπλέον δεν υπάρχει σε αυτόν ήδη εγγραφή κάποιας πηγής που ανήκει στο ίδιο φύλλο του δένδρου με τη νέα πηγή. Μόλις ο RSD γεμίσει πραγματοποιείται μια *στοιχειώδης κατανομή* των εγγραφών του, αντιστοιχίζοντας έναν ισόποσο αριθμό απομακρυσμένων πηγών σε κάθε τοπική πηγή ως εξής: θεωρώντας με τη σειρά κάθε τοπική πηγή S_i , $i \in \{1, 2, \dots, LSD_SIZE\}$, επιλέγονται εκείνες οι απομακρυσμένες πηγές από τον RSD, οι οποίες βρίσκονται πιο κοντά της από πλευράς δενδρικής απόστασης, δημιουργώντας με αυτόν τον τρόπο μια λίστα συγκεντρωμένων πηγών L_{S_i} . Σημειώνεται ότι η απόσταση μεταξύ δυο οποιονδήποτε πηγών X_i, X_j , η οποία συμβολίζεται ως $dist(X_i, X_j)$, μετράται από το ύψος του πιο κοντινού κοινού προγόνου των αντίστοιχων φύλλων του δένδρου. Στη συνέχεια, η ίδια διαδικασία επαναλαμβάνεται κάθε φορά που μια νέα πηγή γίνεται διαθέσιμη μέσω του συγκεκριμένου κόμβου.

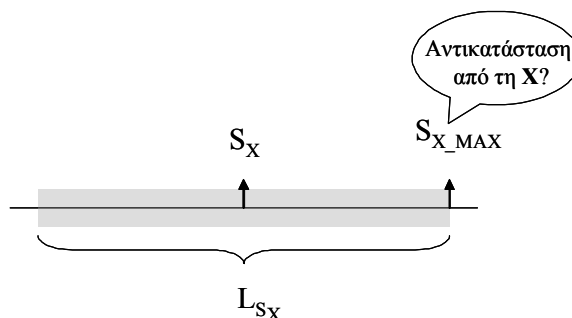
Όταν ληφθεί από έναν κόμβο η διαφήμιση μιας νέας πηγής, η αποθήκευση της οποίας θα προκαλούσε υπέρβαση της χωρητικότητας του RSD, το σχέδιο αντικατάστασης λαμβάνει

χώρα. Εάν υποθέσουμε ότι η διαφήμιση αυτή αναφέρεται σε μια πηγή X , η αντικατάσταση είναι δυνατή μόνο στην περίπτωση που ο RSD δεν περιέχει εγγραφή για πηγή που ανήκει στο ίδιο φύλλο με την X . Ειδικότερα, τα βήματα που ακολουθούνται είναι τα εξής:

1. Μεταξύ των τοπικών πηγών, οι οποίες είναι καταχωρημένες στον LSD, επιλέγεται εκείνη, έστω S_X , που είναι πιο κοντά στην X από πλευράς δενδρικής απόστασης, δηλαδή $dist(S_X, X) = \min_{S_i \in LSD} dist(S_i, X)$.

2. Από την αντίστοιχη λίστα συγκεντρωμένων πηγών L_{S_X} , επιλέγεται η πηγή, έστω S_{X_MAX} , που βρίσκεται πιο μακριά από την S_X από πλευράς δενδρικής απόστασης, δηλαδή $dist(S_{X_MAX}, S_X) = \max_{X_i \in L_{S_X}} dist(X_i, S_X)$. Η S_{X_MAX} θα «συναγωνιστεί» με

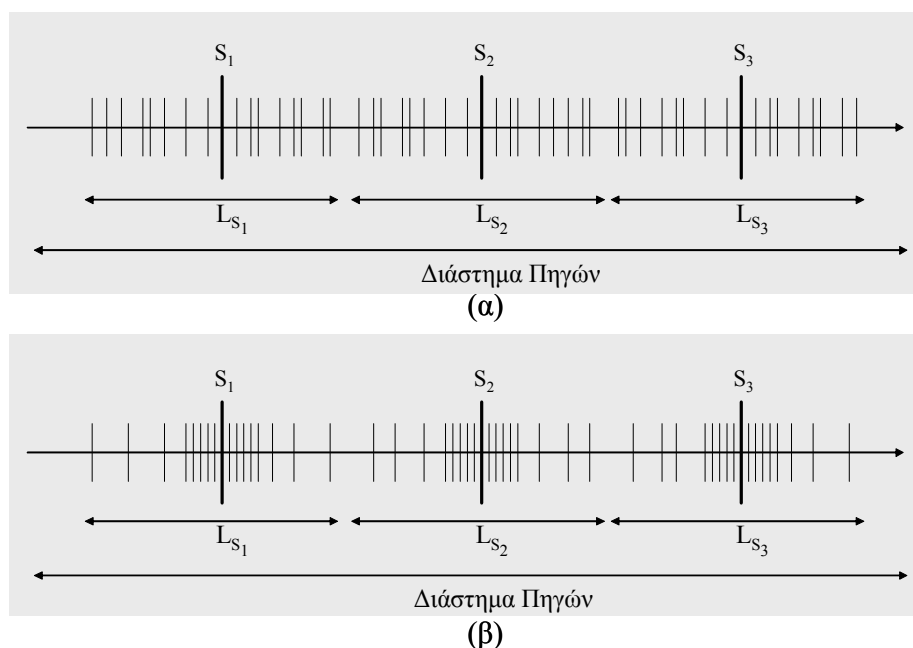
την X για το ποια θα παραμείνει στη λίστα. Βάσει του μοντέλου, η S_X θα πρέπει να διατηρεί σύνδεση προς την S_{X_MAX} με πιθανότητα ανάλογη προς $p_1 = b^{-dist(S_X, S_{X_MAX})}$, και σύνδεση προς την X με πιθανότητα ανάλογη προς $p_2 = b^{-dist(S_X, X)}$. Σύμφωνα με τις κανονικοποιημένες πιθανότητες, η X θα αντικαταστήσει την S_{X_MAX} με πιθανότητα $\frac{p_2}{p_1 + p_2}$ (Σχήμα 5.14).



Σχήμα 5.14: Διάστημα πηγών κατανεμημένων γύρω από την τοπική πηγή S_X βάσει απόστασης

Σημειώνεται επιπλέον ότι η διαφήμιση μιας νέας πηγής που ανήκει στο ίδιο φύλλο με μια τοπική πηγή αποθηκεύεται οπωσδήποτε, υπό την προϋπόθεση ότι δεν υπερβαίνει τη χωρητικότητα του RSD. Αυτό επιτρέπει σε κάθε ομότιμο κόμβο να «ειδικεύεται» στο να απαντάει σε ερωτήσεις για πηγές παρόμοιες με τις τοπικές του. Οι αντίστοιχες εγγραφές στον RSD σχηματίζουν τις λεγόμενες *τοπικές συνδέσεις* ενώ οι υπόλοιπες τις *μακρινές συνδέσεις*.

Η επίδραση της διανομής διαφημίσεων παριστάνεται διαισθητικά στο Σχήμα 5.15, το οποίο δείχνει δυο στιγμιότυπα της κατανομής των διαφημίσεων στον RSD ενός κόμβου. Το πρώτο αναφέρεται στη στοιχειώδη κατανομή των εγγραφών που πραγματοποιείται μόλις γεμίσει ο RSD, ενώ το δεύτερο δείχνει την απώτερη επίδραση του σχεδίου αντικατάστασης στον RSD: συγκέντρωση με ελαφρά τυχαιότητα γύρω από τις τοπικές πηγές.



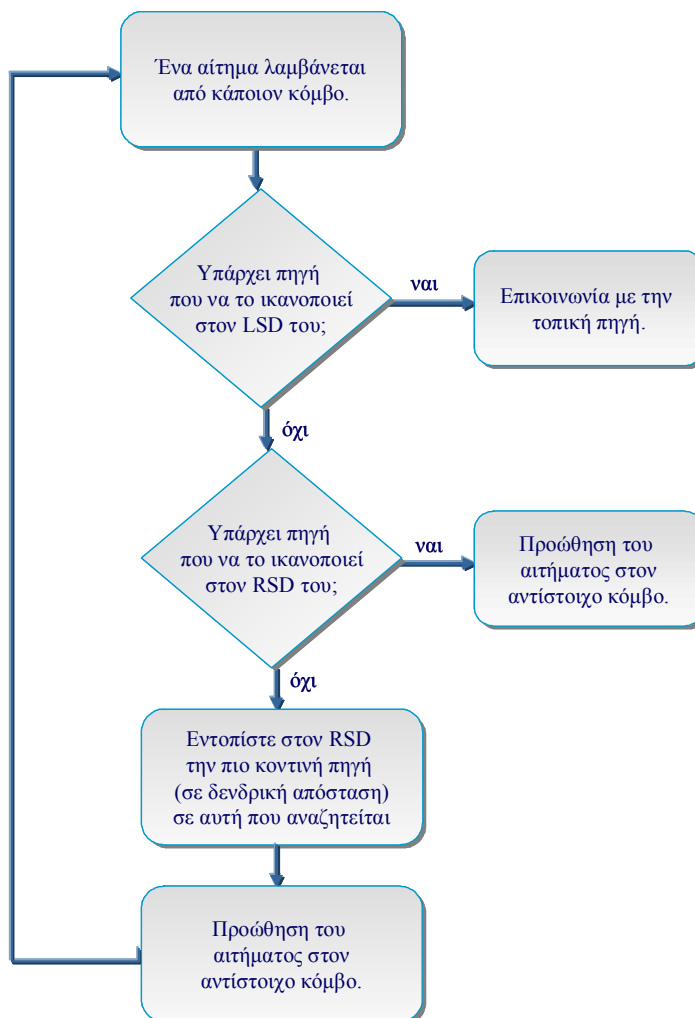
Σχήμα 5.15: Κατανομή πηγών στον RSD ενός κόμβου με 3 τοπικές πηγές (S_1, S_2, S_3) (α) Στοιχειώδης κατανομή των εγγραφών γύρω από τις τοπικές πηγές (β) Απώτερη κατανομή μετά την εφαρμογή του σχεδίου αντικατάστασης

5.8.2.2 Αλγόριθμος Αναζήτησης: Επεξεργασία και Δρομολόγηση Αιτημάτων

Δεδομένου ότι οι RSDs αποθηκεύουν διαφημίσεις με τρόπο ώστε να παρουσιάζουν ιδιότητες μικρού-κόσμου, ένας άπληστος αλγόριθμος εφαρμόζεται για την αναζήτηση της κατάλληλης πηγής: Μόλις ένας κόμβος λάβει ένα αίτημα, ψάχνει πρώτα στις τοπικές πηγές του, δηλαδή στον LSD του. Εάν δεν βρει την κατάλληλη πηγή, ψάχνει στον RSD του. Εάν εντοπίσει την εγγραφή μιας κατάλληλης πηγής, προωθεί το αίτημα κατευθείαν στον κόμβο που υποδεικνύει η εγγραφή. Διαφορετικά, ψάχνει στον RSD για την πιο κοντινή πηγή (από πλευράς δενδρικής απόστασης) σε αυτή που αναζητείται, της οποίας επιπλέον ο κόμβος φιλοξενίας δεν έχει λάβει το συγκεκριμένο αίτημα από τον ίδιο κόμβο στο παρελθόν, μια πρόβλεψη που έχει προστεθεί ώστε να αποφεύγονται οι κύκλοι. Στη συνέχεια προωθεί το αίτημα στον αντίστοιχο κόμβο. Η διαδικασία επαναλαμβάνεται μέχρι να εντοπιστεί η κατάλληλη πηγή ή να μηδενιστεί το TTL, προκειμένου να αποφευχθεί η εξαντλητική αναζήτηση (Σχήμα 5.16).

Σημειώνεται ότι είναι πιθανό ένα βήμα προς τα εμπρός να επιφέρει απόκλιση από την πηγή-στόχο, προκαλώντας απομάκρυνση από αυτή. Αυτό θα συμβεί εάν δεν βρεθεί στον RSD πηγή που να απέχει στη χειρότερη περίπτωση την ίδια απόσταση από το στόχο με αυτή που επιλέχθηκε κατά το προηγούμενο βήμα (περίπτωση μη-χρήσιμου κόμβου όσον αφορά την πηγή που αναζητείται, σύμφωνα με την ορολογία του μοντέλου). Για το λόγο αυτό, εφαρμόζουμε δυο εκδόσεις του αλγορίθμου αναζήτησης: η πρώτη θα εκτελεί μόνο βήματα προς τα εμπρός, επιτρέποντας «παραπλανητικά» βήματα, τα οποία είναι δυνατόν να

προκαλέσουν απομάκρυνση από την πηγή που αναζητείται. Η δεύτερη επιτρέπει μια κίνηση προς τα πίσω, επαναλαμβάνοντας την επίσκεψη του προηγούμενου κατά σειρά επίσκεψης κόμβου, εάν δεν θέλει να συνεχίσει από τον τρέχοντα κόμβο. Οι σχετικοί μηχανισμοί αποκαλούνται στη συνέχεια *Advertisement Flooding-Forward (AF-F)* και *Advertisement Flooding-Backwards (AF-B)*, αντίστοιχα.



Σχήμα 5.16: Κατανεμημένος αλγόριθμος αναζήτησης

5.8.2.3 Παρατηρήσεις

Ο προτεινόμενος μηχανισμός οδηγεί διαισθητικά στις ακόλουθες παρατηρήσεις:

- Για κάθε τοπική πηγή S_i δημιουργούνται στον RSD με μεγάλη πιθανότητα καταχωρήσεις που αφορούν σε πηγές, οι οποίες ανήκουν στο ίδιο φύλλο με την S_i . Αυτές οι τοπικές συνδέσεις επιτρέπουν σε έναν ομότιμο κόμβο να είναι ενήμερος σχετικά με τις πηγές του συστήματος που έχουν ίδιο όνομα με τις τοπικές πηγές του.

- Ο μηχανισμός εξελίσσεται με τέτοιο τρόπο ώστε κάθε RSD να διατηρεί τελικά πληροφορία για το πολύ μια πηγή ενός φύλλου (εξαιρώντας τις πηγές που ανήκουν στο ίδιο φύλλο με τις τοπικές πηγές, όπως αναφέρθηκε νωρίτερα).
- Η διανομή των διαφημίσεων τείνει να προκαλέσει συγκέντρωση με ελαφρά τυχαιότητα των απομακρυσμένων πηγών γύρω από τις τοπικές, διαμορφώνοντας ιδιότητες μικρού-κόσμου.

5.8.3 ΠΡΟΕΤΟΙΜΑΣΙΑ ΠΡΟΣΟΜΟΙΩΣΗΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ ΠΕΙΡΑΜΑΤΩΝ

Στην τρέχουσα ενότητα αξιολογούμε την επίδοση των προτεινόμενων μηχανισμών μέσω συγκεκριμένων πειραματικών σεναρίων. Ο σκοπός είναι να μελετηθεί και να ποσοτικοποιηθεί το πιθανό κέρδος από την εφαρμογή του μοντέλου μικρού-κόσμου στη διανομή των διαφημίσεων. Σαν αναφορά για σύγκριση, χρησιμοποιούμε ένα απλό πρωτόκολλο, το οποίο θα αποκαλούμε *Random* και το οποίο στο στάδιο της προεπεξεργασίας εφαρμόζει ένα τυχαίο σχέδιο αντικατάστασης στους RSDs, ενώ στο στάδιο της αναζήτησης τον ίδιο άπληστο αλγόριθμο.

Για να διευκολύνουμε τη διεξαγωγή του πειράματος θεωρούμε ότι ένας αριθμός πηγών εισέρχονται διαδοχικά στο σύστημα. Έχοντας όλοι οι κόμβοι διαθέσει τον ίδιο αριθμό πηγών, ένα σύνολο αιτημάτων ξεκινούν από διαφορετικούς κόμβους του δικτύου. Σημειώνεται ότι κάθε πηγή αντιστοιχεί σε ένα συγκεκριμένο φύλλο του δένδρου έτσι ώστε το σύνολο των διαθέσιμων πηγών να κατανέμονται ομοιόμορφα μεταξύ των φύλλων του δένδρου. Υποθέτουμε επίσης ότι η δημοτικότητα των αιτημάτων ακολουθεί κατανομή Zipf (με $a=0.6$). Η επίδοση των μηχανισμών θα αξιολογηθεί με βάση το *ποσοστό επιτυχίας (request hit ratio)*, το *μέσο μήκος μονοπατιού ανά αίτημα (average path-length per request)* και το *μέσο μήκος μονοπατιού ανά επιτυχές αίτημα (average path-length per successful request)*. Ο *μέσος αριθμός μηνυμάτων ανά αίτημα (average number of messages per request)* δεν είναι ενδεικτικός σε αυτήν την περίπτωση, δεδομένου ότι ένα μόνο μήνυμα μεταδίδεται σε κάθε βήμα. Έχοντας θέσει διαισθητικά το TTL σε μια λογική τιμή, υπενθυμίζουμε ότι στην περίπτωση που το αίτημα αναζήτησης αποτύχει, θεωρούμε ότι έχει διανύσει μήκος μονοπατιού ίσο με TTL.

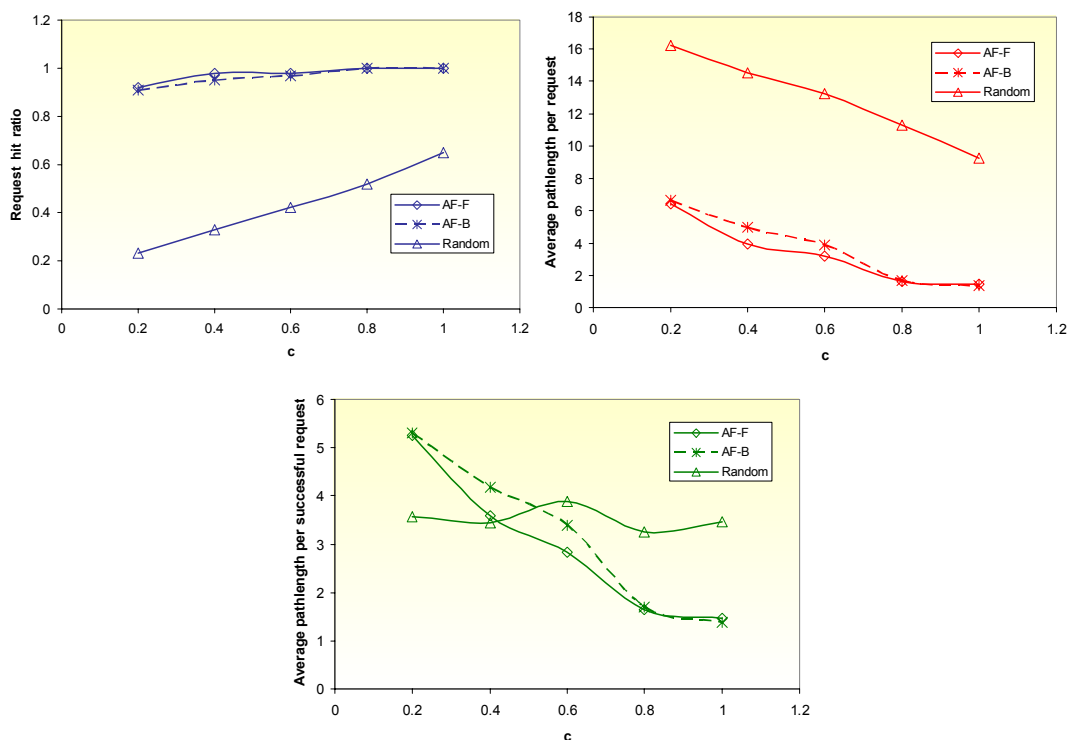
Μεταβάλλοντας το μέγεθος του καταλόγου RSD, το οποίο συμβολίζουμε ως RSD_SIZE, προσπαθούμε να εντοπίσουμε την επιθυμητή ισορροπία μεταξύ αποθηκευτικής ικανότητας κάθε ομότιμου κόμβου και μήκους μονοπατιού για τον εντοπισμό της κατάλληλης πηγής. Βάσει του ιεραρχικού μοντέλου μικρού-κόσμου, το RSD_SIZE, μη-συμπεριλαμβανομένου των καταχωρήσεων που αντιστοιχούν στις τοπικές συνδέσεις, υπολογίζεται ως εξής:

$$RSD_SIZE = LSD_SIZE \cdot (c \cdot \log_b^2 n) \quad (10)$$

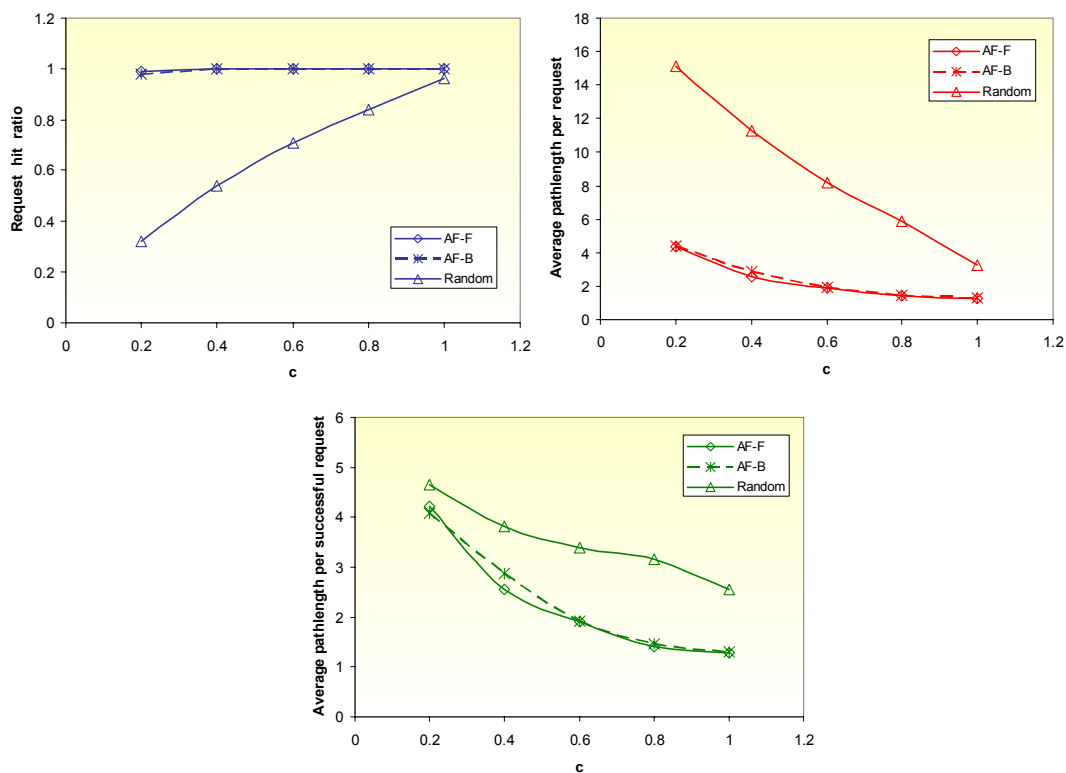
όπου n ο συνολικός αριθμός φύλλων του πλήρους b -ary δένδρου πληροφορίας και c μια σταθερά. Στην ουσία, για διαφορετικές τιμές του LSD_SIZE , δηλαδή του μεγέθους του LSD , θα μεταβάλλουμε το RSD_SIZE , μεταβάλλοντας την τιμή της παραμέτρου c .

5.8.3.1 Περίπτωση 1^η: Κάθε κόμβος ελέγχει πηγές αυθαίρετου περιεχομένου

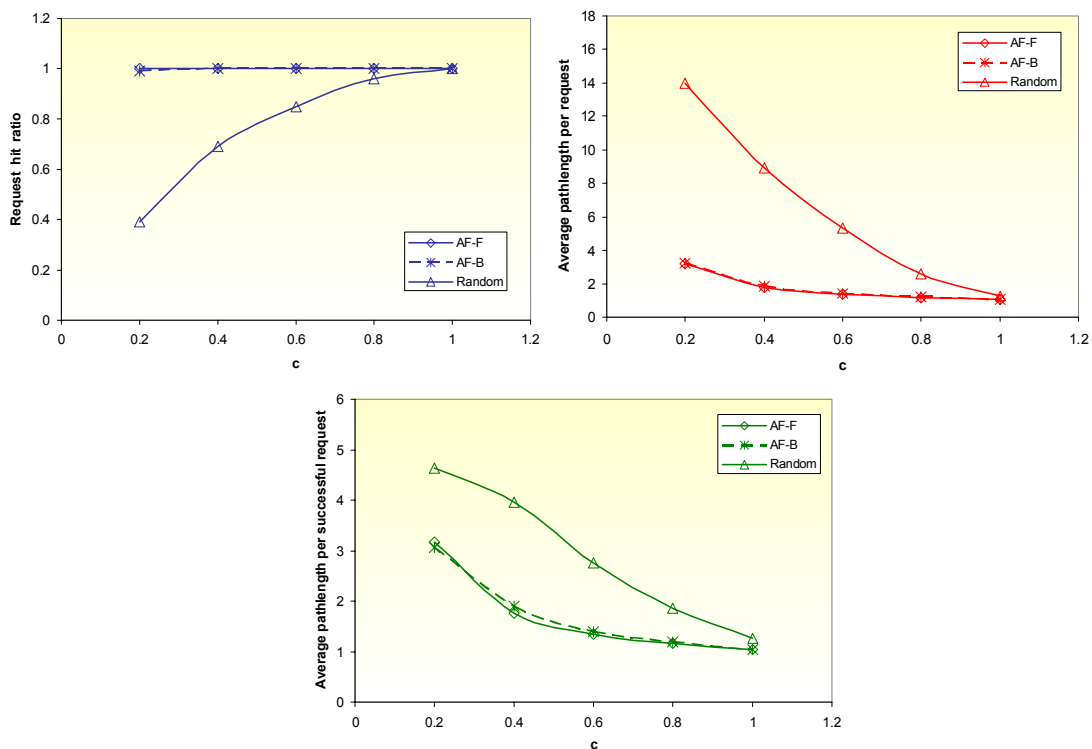
Για το συγκεκριμένο πείραμα θεωρούμε ένα b -ary δένδρο με $b=4$ και ύψος ίσο με 5, οπότε ο συνολικός αριθμός των φύλλων του δένδρου είναι ίσος με 1024. Σε αυτήν την περίπτωση θεωρούμε ότι το ύψος του πιο κοντινού κοινού προγόνου των πηγών που ανήκουν στον ίδιο ομότιμο κόμβο είναι ίσο με 5, δηλαδή οι πηγές κάθε κόμβου κατανέμονται ομοιόμορφα μεταξύ όλων των φύλλων του δένδρου. Επίσης θεωρούμε έναν αριθμό ομότιμων κόμβων ίσο με 200. Θέτοντας το LSD_SIZE σε διαφορετικές τιμές, εξερευνούμε τον τρόπο με τον οποίο μεταβάλλονται τα κριτήρια επίδοσης σε σχέση με το RSD_SIZE , το οποίο καθορίζεται από την τιμή της παραμέτρου c . Πιο συγκεκριμένα, το c μεταβάλλεται από το 0.2 στο 1 με βήμα ίσο προς 0.2, μεταβάλλοντας το λόγο του RSD_SIZE προς το συνολικό αριθμό των πηγών σε 2.5%, 5%, 7.5%, 10%, και 12.5%, αντίστοιχα. Τα κριτήρια επίδοσης υπολογίζονται κατά μέσο όρο για αριθμό αιτημάτων ίσο με 100.



Σχήμα 5.17: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=10$ (2000 πηγές συνολικά) και το πολύ 2 πηγές σε κάθε φύλλο



Σχήμα 5.18: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα κα ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=20$ (4000 πηγές συνολικά) και το πολύ 4 πηγές σε κάθε φύλλο

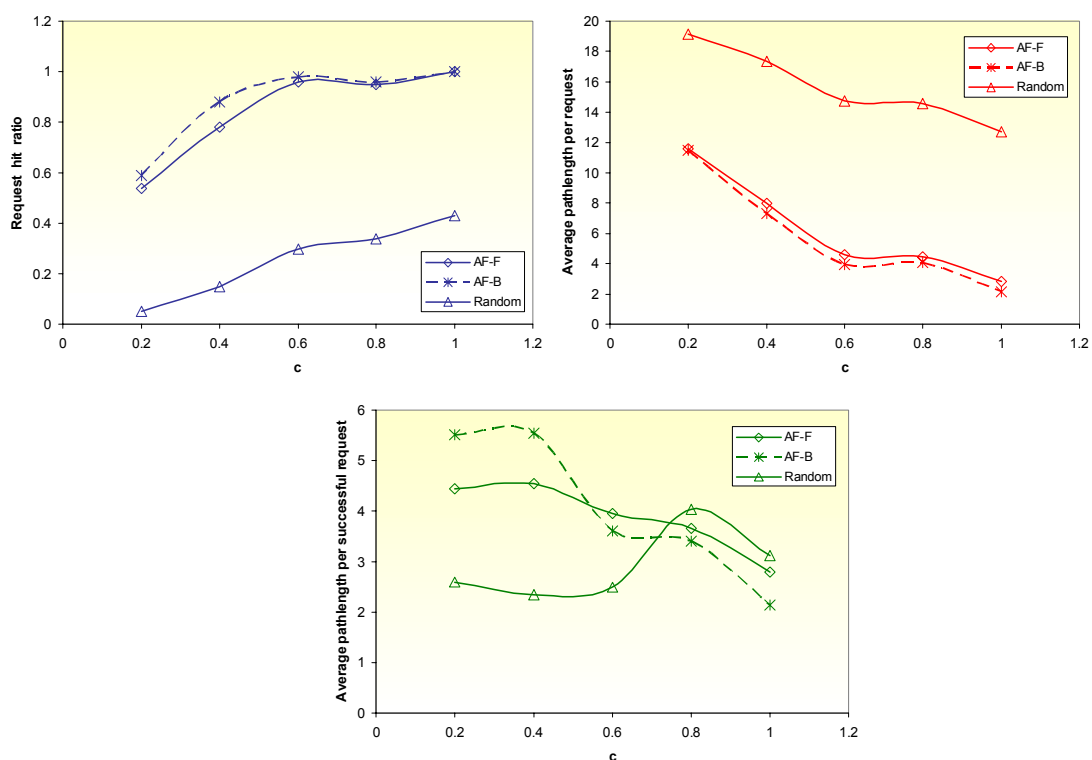


Σχήμα 5.19: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=30$ (6000 πηγές συνολικά) και το πολύ 6 πηγές σε κάθε φύλλο

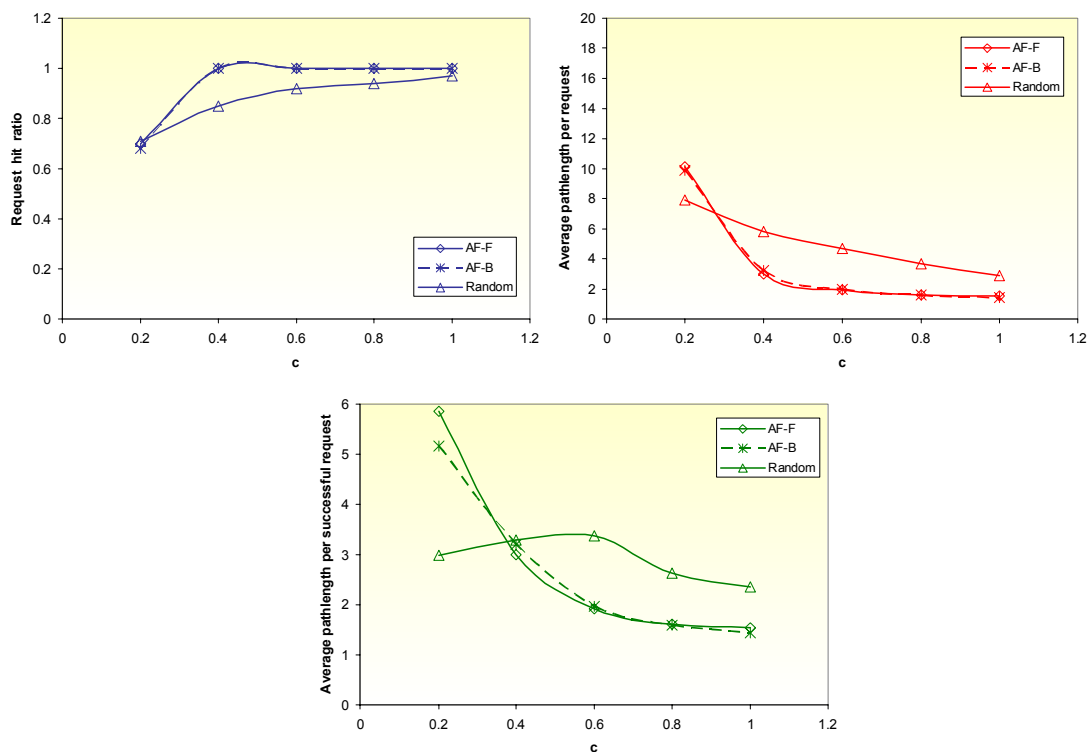
Τα παραπάνω σχήματα δείχνουν ότι οι προτεινόμενοι μηχανισμοί υπερτερούν σαφέστερα του Random, σημειώνοντας υψηλά ποσοστά επιτυχίας και επιβεβαιώνοντας πειραματικά το μέσο μήκος μονοπατιού $O(\log_b n)$ που εγγυάται το θεωρητικό μοντέλο. Είναι επίσης φανερό ότι καθώς η τιμή της παραμέτρου c , η οποία καθορίζει το αριθμό των μακρινών συνδέσεων που αντιστοιχούν σε κάθε τοπική πηγή αυξάνει, η βελτίωση της επίδοσης που μπορούν να πετύχουν οι προτεινόμενοι μηχανισμοί σε σχέση με τον Random, όσον αφορά το ποσοστό επιτυχίας και το μέσο μήκος μονοπατιού ανά αίτημα, γίνεται λιγότερο αισθητή. Επίσης, καθώς το LSD_SIZE αυξάνει, η απόκλιση από τον Random γίνεται όλο και λιγότερο αισθητή καθώς αυξάνει το c . Συνεπώς, η αξία των νέων μηχανισμών είναι ιδιαίτερα σημαντική κυρίως στην περίπτωση περιορισμένης αποθηκευτικής ικανότητας των RSDs (μικρή τιμή για το c), αφού σε αυτή την περίπτωση σχηματίζονται σαφώς συνδέσεις μικρού-κόσμου. Επιπλέον, αυξάνοντας σταδιακά το LSD_SIZE και διατηρώντας σταθερή την τιμή του c ώστε ο λόγος του RSD_SIZE προς το συνολικό αριθμό πηγών να παραμένει σταθερός, τα νέα πρωτόκολλα πετυχαίνουν όλο και καλύτερα αποτελέσματα. Συνεπώς, καθώς ο αριθμός των τοπικών πηγών ενός κόμβου αυξάνει, μπορούμε να επιλέγουμε όλο και μικρότερη τιμή του c και να διατηρούμε την ίδια ποιότητα αποτελεσμάτων, αφού οι μακρινές συνδέσεις που αντιστοιχούν σε κάθε τοπική πηγή συμπίπτουν όλο και περισσότερο με συνδέσεις που αντιστοιχούν σε διαφορετικές τοπικές πηγές. Όσον αφορά το μέσο μήκος μονοπατιού ανά επιτυχές αίτημα, οι παρατηρήσεις είναι παρόμοιες. Σημειώνεται ωστόσο, ότι σε αυτήν την περίπτωση, τα αποτελέσματα δεν είναι ενδεικτικά για μικρή τιμή του LSD_SIZE, αφού το ποσοστό επιτυχίας του Random είναι πολύ μικρό και είναι πιθανό η ικανοποίηση των αιτημάτων να οφείλεται κυρίως σε τοπικές συνδέσεις. Για το συγκεκριμένο κριτήριο, τα αποτελέσματα είναι πιο ενδεικτικά καθώς το LSD_SIZE αυξάνει. Τέλος, ο AF-F παρέχει ελαφρώς καλύτερα αποτελέσματα από τον AF-B.

5.8.3.2 Περίπτωση 2^η: Κάθε κόμβος ελέγχει πηγές με σχετική εγγύτητα περιεχομένου

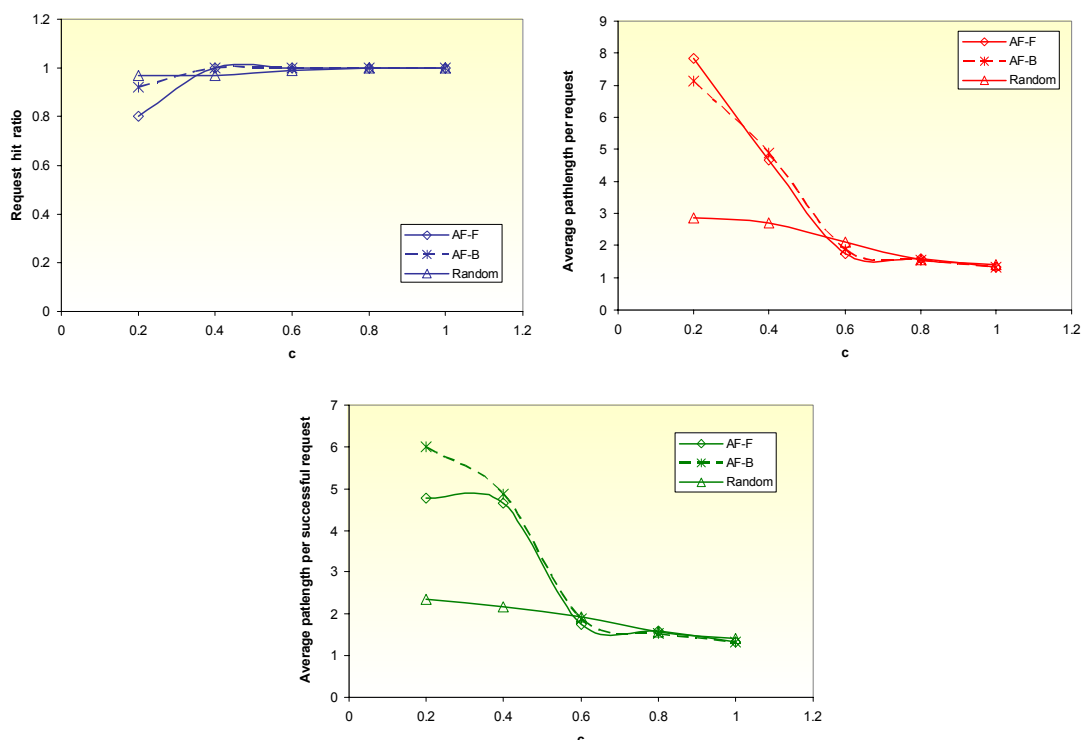
Το προηγούμενο πείραμα επαναλαμβάνεται, με τη διαφορά ότι σε αυτήν την περίπτωση κάθε ομότιμος κόμβος εξειδικεύεται περισσότερο ως προς το περιεχόμενο των πηγών που διαθέτει τοπικά, φέροντας πηγές παρόμοιου περιεχομένου. Ειδικότερα, υποθέτουμε ότι το ύψος του πιο κοντινού κοινού προγόνου των πηγών που ανήκουν στον ίδιο ομότιμο κόμβο είναι ίσο με 2, δηλαδή οι πηγές κάθε κόμβου κατανέμονται ομοιόμορφα μεταξύ όλων των φύλλων ενός συγκεκριμένου υπο-δένδρου ύψους 2. Θέτοντας και πάλι το LSD_SIZE σε διαφορετικές τιμές, μεταβάλλουμε την τιμή του c από το 0.2 στο 1 με βήμα ίσο με 0.2, διαμορφώνοντας το λόγο του RSD_SIZE προς το συνολικό αριθμό πηγών στην τιμή 2.5%, 5%, 7.5%, 10%, και 12.5%, αντίστοιχα.



Σχήμα 5.20: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=5$ (1000 πηγές συνολικά) και το πολύ 1 πηγή σε κάθε φύλλο



Σχήμα 5.21: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=10$ (2000 πηγές συνολικά) και το πολύ 2 πηγές σε κάθε φύλλο



Σχήμα 5.22: Ποσοστό επιτυχίας, μέσο μήκος μονοπατιού ανά αίτημα και ανά επιτυχές αίτημα σε σχέση με την παράμετρο c , για $LSD_SIZE=20$ (4000 πηγές συνολικά) και το πολύ 4 πηγές σε κάθε φύλλο

Σύμφωνα με τα παραπάνω σχήματα και συγκρίνοντας με τα αποτελέσματα της προηγούμενης περίπτωσης είναι προφανές ότι οι προτεινόμενοι μηχανισμοί δεν ευνοούνται από τον αυξημένο βαθμό ομοιότητας μεταξύ των τοπικών πηγών, δεδομένου ότι η απόκλιση τους από τον Random είναι λιγότερο αισθητή. Συνεπώς, καταλήγουμε στο συμπέρασμα ότι μια τέτοια συγκέντρωση πηγών δυσκολεύει τη διαμόρφωση συνδέσεων μικρού-κόσμου. Επιπλέον, είναι φανερό ότι καθώς αυξάνει το LSD_SIZE οι προτεινόμενοι μηχανισμοί απαιτούν όλο και μεγαλύτερη τιμή για το c , ώστε να παράγουν καλύτερα αποτελέσματα από τον Random. Αυτό εξηγείται ως εξής: Καθώς ο αριθμός των τοπικών πηγών, οι οποίες παρουσιάζουν αυξημένη ομοιότητα ως προς το περιεχόμενό τους σε σχέση με την προηγούμενη περίπτωση αυξάνει, ο αριθμός των μακρινών συνδέσεων θα πρέπει να αυξάνει ανάλογα για να σχηματιστούν συνδέσεις μικρού-κόσμου, και για το λόγο αυτό απαιτείται αυξημένη τιμή για το c . Σημειώνεται ότι τα αποτελέσματα που αφορούν στο μέσο μήκος μονοπατιού ανά επιτυχές αίτημα του Random δεν είναι ενδεικτικά για μικρή τιμή του LSD_SIZE , όπως και στην προηγούμενη περίπτωση, εξαιτίας του μικρού ποσοστού επιτυχίας που σημειώνει. Τέλος, σε αντίθεση με την προηγούμενη περίπτωση, ο AF-B μερικές φορές παράγει ελαφρώς καλύτερα αποτελέσματα από τον AF-F, ειδικότερα όταν το LSD_SIZE είναι μικρό.

5.9 ΠΑΡΑΤΗΡΗΣΕΙΣ – ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο παρόν κεφάλαιο μελετήσαμε διαφορετικές λύσεις σε ένα πρόβλημα εντοπισμού πηγών πληροφορίας που ορίστηκε στα πλαίσια της παροχής χρονικά κρίσιμων υπηρεσιών με επίγνωση του περιβάλλοντος πάνω από P2P υπόδειγμα επικοινωνίας, με σκοπό την ενίσχυση της αποδοτικότητας και επεκτασιμότητας του συστήματος. Οι λύσεις αυτές εφαρμόζονται είτε με *αντιδραστικό (reactive)*, είτε με *προδραστικό (proactive)* τρόπο, αναλόγως εάν λαμβάνουν μέτρα προκειμένου να βελτιωθεί ο τρόπος αναζήτησης των πληροφοριών κατά τη διάρκεια ή πριν την εκτέλεση των αιτημάτων, αντίστοιχα.

Οι αντιδραστικοί μηχανισμοί αποφεύγουν την εξαντλητική σπατάλη πόρων των συνήθων αλγορίθμων πλημμύρας, παρακολουθώντας το προφίλ ζήτησης των διαθέσιμων πηγών πληροφορίας με αποδοτικά μέσα που βασίζονται στο κατανεμημένο περιβάλλον υπολογισμού που παρέχεται από τα Ενεργά Δίκτυα. Τα αποτελέσματα των πειραμάτων ποσοτικοποιούν τα κέρδη των προτεινόμενων μηχανισμών, θεωρώντας ως αναφορά για τη σύγκριση τον αλγόριθμο της πλημμύρας, ο οποίος πετυχαίνει πολύ μικρό χρόνο αναζήτησης με εξαντλητική κατανάλωση εύρους ζώνης, και τον τυχαίο περίπατο με τοπική πλημμύρα, ο οποίος πετυχαίνει πολύ χαμηλή κατανάλωση εύρους ζώνης με επιβάρυνση του χρόνου αναζήτησης. Συγκεκριμένα, αποδεικνύουν ότι οι προτεινόμενοι μηχανισμοί εγγυώνται πολύ χαμηλό χρόνο αναζήτησης, περιορίζοντας ταυτόχρονα την κατανάλωση εύρους ζώνης. Από την άλλη, οι προδραστικοί μηχανισμοί χρησιμοποιούν το μοντέλο μικρού-κόσμου στη διανομή διαφημίσεων για τις νέες πηγές που εισέρχονται στο σύστημα. Πιο συγκεκριμένα, μελετήθηκαν δυο παραλλαγές ενός βασικού μηχανισμού, με την καθεμία να υπερέχει ελαφρώς της άλλης σε συγκεκριμένες περιπτώσεις. Γενικά, η αξία του προτεινόμενου μηχανισμού αναγνωρίζεται κυρίως όταν ο αριθμός των φύλλων που αντιστοιχούν στις τοπικές πηγές κάθε κόμβου διατηρείται μικρός συγκριτικά με το συνολικό αριθμό φύλλων του δένδρου πληροφορίας. Τα αποτελέσματα της προσομοίωσης έδειξαν ότι στην περίπτωση που οι κόμβοι φέρουν πηγές αυθαίρετου περιεχομένου, η εφαρμογή του μηχανισμού ευνοείται όταν οι αποθηκευτικές ικανότητες των κόμβων είναι περιορισμένες. Αντίθετα, στην περίπτωση συγκέντρωσης πηγών με παραπλήσιο περιεχόμενο σε κάθε κόμβο, οι προτεινόμενοι μηχανισμοί ξεπερνούν τον Random, ο οποίος χρησιμοποιείται για τη σύγκριση, όταν ο αριθμός των μακρινών συνδέσεων που αντιστοιχούν σε κάθε τοπική πηγή υπερβαίνει ένα όριο, το οποίο εξαρτάται από τον αριθμό τοπικών των πηγών.

Τέλος, αξίζει να σημειωθεί ότι, αν και οι εν λόγω μηχανισμοί προτείνονται στα πλαίσια του συγκεκριμένου προβλήματος, θα μπορούσαν να χρησιμοποιηθούν και σε πιο γενικές περιπτώσεις ή περιβάλλοντα που παρουσιάζουν παρόμοια συμπεριφορά. Απώτερος σκοπός αυτής της μελέτης είναι να συνεισφέρει προς την αποτελεσματική και οικονομική δραστηριότητα των κατανεμημένων μηχανισμών αναζήτησης.

5.10 ΒΙΒΛΙΟΓΡΑΦΙΑ – ΑΝΑΦΟΡΕΣ

- [1] M. Parameswaran, A. Susarla, A.B. Whinston, “P2P Networking: An Information-sharing Alternative”, *Computer*, Vol.34, No.7, pp.31-38, July 2001.
- [2] D.S. Milojevic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, Z. Xu, “Peer-to-Peer Computing”, Technical Report HPL-2002-57, HP Laboratories Palo Alto, March 8, 2002.
- [3] <http://www.napster.com>
- [4] G. Kan, “Gnutella”, chapter 8 in the book “Peer-to-Peer: Harnessing the Power of Disruptive Technologies”, Publisher: O’Reilly, ISBN: 0-596-00110-X, February 2001.
- [5] M. Jovanovic, F.S. Annexstein, and K.A. Berman, “Scalability Issues in Large Peer-to-Peer Networks - A Case Study of Gnutella”, Technical Report, University of Cincinnati, 2001.
- [6] L.A. Adamic, R. Lukose, A. Puniyani, and B. Huberman, “Search in Power-law Networks”, *Physical Review E* 64, pp.46135-46143, 2001.
- [7] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, “Search and Replication in Unstructured Peer-to-Peer Networks”, *Proceedings of the ACM Conference on Supercomputing*, New York, USA, pp.84-95, 2002.
- [8] B. Yang and H. Garcia-Molina, “Efficient Search in Peer-to-Peer Networks”, *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS)*, Vienna, Austria, 2002.
- [9] C. Gkantsidis, M. Mihail, and A. Saberi, “Random Walks in Peer-to-Peer Networks”, *Proceedings of the IEEE Infocom 2004*, March 2004.
- [10] C. Gkantsidis, M. Mihail, and A. Saberi, “Hybrid Search Schemes for Unstructured Peer-to-Peer Networks”, *Proceedings of the IEEE Infocom 2005*, March 2005.
- [11] D.A. Menascé and L. Kanchanapalli, “Probabilistic Scalable P2P Resource Location Services”, *ACM Sigmetrics Performance Evaluation Rev.*, Vol.30, No.2, pp.48-58, 2002.
- [12] F. Pedone, N. Duarte and M. Goulart, “Probabilistic Queries in Large-scale Networks”, *Proceedings of 4th European Dependable Computing Conference (EDCC-4)*, Toulouse, France, pp.209-226, 2002.
- [13] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, “Freenet: A Distributed Anonymous Information Storage and Retrieval System”, *Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, Berkeley, CA, pp. 46-66, 2000.

- [14] I. Stoica, R. Morris, D. Karger, M.E Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications”, Proceedings of ACM SIGCOMM 2001, San Deigo, CA, August 2001.
- [15] A. Rowstron and P. Druschel, “Pastry: Scalable, Distributed, Object Location and Routing for Large-scale Peer-to-Peer Systems”, Proceedings of IFIP/ACM International Conference on Distributed System Platforms (Middleware), Heidelberg, Germany, pp.329-350, November 2001.
- [16] Y. Zhao, J.D. Kubiawicz, and A. Joseph, “Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing”, Technical Report UCB/CSD-01-1141, Berkeley, April 2000.
- [17] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A Scalable Content Addressable Network”, Proceedings of ACM Sigcomm 2001, pp.161-172, August 2001.
- [18] M. Newman, D. Watts, S. Strogatz, “Random Graph Models of Social Networks”, Proceedings of the National Academy of Science of the USA, Vol.99, pp.2566-2572, February 2002.
- [19] The Network Simulator NS-2, <http://www.isi.edu/nsnam/ns>
- [20] OPNET Modeler, <http://www.opnet.com/products/modeler/home.html>
- [21] D. Cavin, Y. Sasson, and A. Shiper, “On the Accuracy of MANET Simulators”, Proceedings of 2nd ACM International Workshop on Principles of Mobile Computing (POMC'02), pp.38-43, October 2002.
- [22] I. Sygkouna, M. Anagnostou and E. Sykas, “Efficient Search Mechanisms in a Context Distribution System”, accepted for publication in the Proceedings of the 20th IEEE International Conference on Advanced Information Networking and Applications (AINA 2006), Vienna, Austria, April 2006.
- [23] A. Bestavros, “Speculative Data Dissemination and Service to Reduce Server Load, Network Traffic and Service Time for Distributed Information Systems”, Proceedings of International Conference on Data Engineering, New Orleans, Louisiana, pp.180-187, March 1996.
- [24] F. Chinchilla, M. Lindsey, and M. Papadopouli, “Analysis of Wireless Information Locality and Association Patterns in a Campus, Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Hong Kong, China, , pp.906–917, March 2004.
- [25] S. Jin and A. Bestavros, “Sources and Characteristics of Web Temporal Locality”, Proceedings of IEEE/ACM International Symposium on Modeling, Analysis and

- Simulation of Computer and Telecommunication Systems, San Fransisco, CA, pp.28-35, August 2000.
- [26] CONTEXT, “Active Creation, Delivery and Management of Efficient Context-Aware Services”, IST-2001-38142, Homepage URL: <http://context.upc.es>
- [27] V. Almeida, A. Bestavros, M. Crovella, and A. Oliveira, “Characterizing Reference Locality in the WWW”, Proceedings of the 4th International Conference on Parallel and Distributed Information Systems (PDIS), IEEE Comp. Soc. Press, pp.92-106, December 1996.
- [28] W. Caripe, G. Cybenko, K. Moizumi, R. Gray, “Network Awareness and Mobile Agent Systems”, IEEE Communications Magazine, Vol.36, No.7, pp.44-49, July 1998.
- [29] B.M. Waxman, “Routing of Multipoint Connections”, IEEE Journal on Selected Areas in Communications, Vol.6, No.9, pp.1617-1622, December 1988.
- [30] P. Barford and M. Crovella, “Generating Representative Web Workloads for Network and Server Performance Evaluation”, Proceedings of the ACM SIGMETRICS’98 Conference, Madison, WI, pp.151-160, 1998.
- [31] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web Caching and Zipf-like Distributions: Evidence and Implications”, Proceedings of the IEEE Infocom 1999 (1), pp.126-134, March 1999.
- [32] I. Sygkouna, M. Anagnostou and E. Sykas, “Advertisement-aided Search in a P2P Context Distribution System”, accepted for publication by Springer-Verlag in the Proceedings of the IEEE International Workshop on P2P for High Performance Computational Sciences (P2P HPCS’06), May 2006.
- [33] I. Sygkouna, M. Anagnostou and E. Sykas, “A Proactive Mechanism for Efficient Information Discovery”, accepted for publication in the Proceedings of the 2nd IEEE International Conference on Information & Communication Technologies: From Theory to Applications (ICTTA’06), April 2006.
- [34] K. Arnold, B. Osullivan, R.W. Scheifler, J. Waldo and A. Wollrath, “The Jini Specification”, Publisher: Addison-Wesley, ISBN: 0-201-61634-3, June 1999.
- [35] The Salutation Consortium Inc., “Salutation Architecture Specification Part 1, Version 2.1 Edition”, <http://www.salutation.org>, 1999.
- [36] R. John, “UPnP, Jini and Salutaion - A Look at Some Popular Coordination Frameworks for Future Network Devices”, Technical Report, California Software Labs, <http://www.cswl.com/whiteppr/tech/upnp.html>, 1999.
- [37] E. Guttman, C. Perkins and J. Veizades, “RFC 2165: Service Location Protocol”, June 1997.

- [38] A. Ranganathan and R.H. Campbell, “Advertising in a Pervasive Environment”, Proceedings of the 2nd ACM International Workshop on Mobile Commerce, Atlanta, Georgia, pp.10-14, September 2002.
- [39] T. Finin, O. Ratsimor, A. Joshi and Y. Yesha, “eNcentive: A Framework for Intelligent Marketing in Mobile Peer-To-Peer Environments”, Proceedings of the 5th International Conference on Electronic Commerce (ICEC’03), New York, NY, USA, pp.87-94, October 2003.
- [40] H. Zhang, A. Goel, R. Govindan, “Using the Small-world Model to Improve Freenet Performance”, Proceedings of the IEEE Infocom 2002, Vol.21, No.1, pp.1228-1237, June 2002.
- [41] J. Kleinberg, “The Small-world Phenomenon: An Algorithmic Perspective”, Proceedings of the 32nd ACM Symposium on Theory of Computing, pp.163-170, 2000.
- [42] S. Milgram, “The Small World Problem”, Psychology Today, Vol.1, No.61, pp.60-67, May 1967.
- [43] J. Travers and S. Milgram, “An Experimental Study of the Small-world Problem”, Sociometry, Vol.32, No.4, pp.425-443, 1969.
- [44] C. Korte and S. Milgram, “Acquaintance Networks between Racial Groups: Application of the Small World Method”, J. Personality and Social Psych., Vol.15, pp.101-108, 1978.
- [45] H. Kautz, B. Selman, M. Shah, “ReferralWeb: Combining Social Networks and Collaborative Filtering”, Communications of the ACM, Vol.40, No.3, pp.63-65, March 1997.
- [46] D. Watts and S. Strogatz, “Collective Dynamics of Small-world Networks”, Nature, Vol.393, pp.440-442, 1998.
- [47] R. Albert, H. Jeong, A.L. Barabasi, “The Diameter of the World Wide Web”, Nature, Vol.401, pp.130-131, 1999.
- [48] B. Bollobas, F. Chung, “The Diameter of a Cycle plus a Random Matching”, SIAM Journal on Discrete Mathematics, Vol.1, No.3, pp.328-333, 1988.
- [49] J. Kleinberg, “Small-world Phenomena and the Dynamics of Information”, Advances in Neural Information Processing Systems (NIPS), Vol.14, 2001.
- [50] S. Xynogalas, M. Chantzara, I. Sygkouna, S. Vrontis, I. Roussaki, M. Anagnostou, “Context Management for the Provision of Adaptive Services to Roaming Users”, IEEE Wireless Communications, Vol.11, No.2, pp.40-47, April 2004.

ΚΕΦΑΛΑΙΟ 6

Επίλογος

ΚΕΦΑΛΑΙΟ 6: ΕΠΙΛΟΓΟΣ

6.1 ΑΝΑΣΚΟΠΗΣΗ

Οι γρήγορες εξελίξεις στην τεχνολογία των υπολογιστών και των επικοινωνιών και η σύγκλιση τους οδηγούν ταχύτατα σε μια παγκόσμια αγορά πληροφοριών, η οποία θα αποτελείται από ομοσπονδίες πληροφοριακών συστημάτων που θα συνεργάζονται για να καλύπτουν τις ανάγκες των χρηστών για πληροφορίες. Προς αυτήν την κατεύθυνση υιοθετούνται λύσεις μεσισμικού, οι οποίες βασίζονται στη χρησιμοποίηση μεσιτών για την ενοποίηση των συστημάτων και την εξασφάλιση διαλειτουργικότητας, αντιμετωπίζοντας αποτελεσματικά ζητήματα ετερογένειας, αυτονομίας και κατανομής. Εν τω μεταξύ, ο όγκος και η ποικιλομορφία της πληροφορίας που γίνεται διαθέσιμη ηλεκτρονικά θέτει ενδιαφέροντα ζητήματα για τα πληροφοριακά συστήματα μεγάλης κλίμακας. Μια από τις μεγαλύτερες ερευνητικές προκλήσεις που τίθενται είναι η αποτελεσματική και αποδοτική διανομή της πληροφορίας. Προκειμένου να χρησιμοποιηθούν αποτελεσματικά οι σχετικές υπηρεσίες, είναι απαραίτητο να υπάρχουν αποδοτικοί αλγόριθμοι που να ισορροπούν τη μεμονωμένη με τη γενική απόδοση, παρέχοντας ταυτόχρονα δυνατότητες επέκτασης από πλευράς όγκου δεδομένων, αριθμού πελατών και εύρους ζώνης.

Στην παρούσα διατριβή, η μελέτη επικεντρώθηκε στην ανάπτυξη αλγορίθμων για την αντιμετώπιση ορισμένων προβλημάτων που θεωρούνται κρίσιμα για την καλύτερη επίδοση ενός πληροφοριακού συστήματος. Ειδικότερα:

- Σε ένα σύστημα κεντρικής μεσιτείας, η επικοινωνία του κεντρικού μεσίτη με τους εξωτερικούς παρόχους για ανάκτηση πληροφορίας αποτελεί μια δαπανηρή φάση της επεξεργασίας πληροφορίας. Η ταχύτητα στη μεταφορά των δεδομένων από τη περιοχή των παρόχων στη περιοχή του μεσίτη παίζει σημαντικό ρόλο στην απόκριση του συστήματος. Για το λόγο μελετήθηκε η εφαρμογή αποδοτικών τεχνικών για την πραγματοποίηση της απευθείας και σε πραγματικό χρόνο επικοινωνίας του μεσίτη με τους παρόχους του συστήματος για ανάκτηση πληροφορίας.
- Σε ένα σύστημα ομότιμων μεσιτών, η εξασφάλιση αποδοτικότητας συνίσταται στη μείωση του χρόνου απόκρισης του συστήματος στα επικείμενα αιτήματα των καταναλωτών, καθώς επίσης και στον περιορισμό των δαπανών που αφορούν σε επικοινωνιακούς ή αποθηκευτικούς πόρους του συστήματος. Σε ένα τέτοιο σύστημα, η αναζήτηση της πληροφορίας μεταξύ των μεσιτών αποτελεί συχνά το πιο χρονοβόρο στάδιο της διανομής πληροφορίας, και για το λόγο αυτό μελετήθηκε η εφαρμογή συγκεκριμένων μηχανισμών βελτιστοποίησης της αναζήτησης.

Τα παραπάνω προβλήματα μελετήθηκαν σε ένα ολοκληρωμένο πλαίσιο, το οποίο ξεκινάει από ένα θεωρητικό υπόβαθρο με αναφορά σε τεχνολογίες πληροφοριακών συστημάτων και υπολογιστικά μοντέλα και καταλήγει στα ειδικότερα προβλήματα μελέτης της διατριβής. Στο κεφάλαιο 1, το οποίο αποτελεί την εισαγωγή της διατριβής, έγινε μια σύντομη αναφορά στην εξέλιξη των πληροφοριακών συστημάτων και συζητήθηκαν οι προκλήσεις που παρουσιάζονται στον τομέα της συλλογής και διανομής πληροφορίας. Επίσης έγινε μια αναφορά στην εξέλιξη των υπολογιστικών μοντέλων, από την κατανεμημένη υπολογιστική προς την επόμενη γενιά υπολογιστικής, η οποία είναι γνωστή ως διεισδυτική υπολογιστική, και συζητήθηκε ο ρόλος του υποδείγματος κινητού κώδικα σε αυτήν την εξέλιξη.

Στο κεφάλαιο 2 δόθηκε η περιγραφή των βασικών αρχιτεκτονικών των κατανεμημένων πληροφοριακών συστημάτων και παρουσιάστηκαν οι τεχνολογίες διανομής δεδομένων. Στη συνέχεια, αναλύθηκε το πρόβλημα της ενοποίησης δεδομένων και περιγράφηκε το υπόδειγμα μεσιτείας ως η επικρατέστερη προσέγγιση. Τέλος, παρουσιάστηκαν δυο χαρακτηριστικά μεσιτικά συστήματα, τα οποία αποτέλεσαν τη βάση για τη μελέτη των ειδικότερων προβλημάτων στα οποία επικεντρώθηκε η διατριβή.

Αντικείμενο του κεφαλαίου 3 είναι το υπόδειγμα κινητού κώδικα και ο ρόλος του στα συστήματα διανομής πληροφορίας. Αφού αναλύθηκαν τα κίνητρα που οδηγούν στη χρήση κινητού κώδικα, παρουσιάστηκαν εναλλακτικά υποδείγματα κινητού κώδικα, τα οποία συγκρίθηκαν με το παραδοσιακό μοντέλο απομακρυσμένης επικοινωνίας. Στα πλαίσια υπολογιστικών μοντέλων, μελετήθηκαν οι τεχνολογίες των Κινητών Πρακτόρων και των Ενεργών Δικτύων. Τέλος, αναλύθηκε ο τρόπος αξιοποίησης του υποδείγματος κινητού κώδικα (στη μορφή Κινητών Πρακτόρων και Ενεργών Δικτύων) στα ειδικότερα προβλήματα μελέτης της διατριβής.

Στο κεφάλαιο 4 διατυπώθηκε και επιλύθηκε προσεγγιστικά το πρόβλημα της συλλογής κατανεμημένης πληροφορίας σε σύστημα κεντρικής μεσιτείας με παράλληλη αποστολή κινητών πρακτόρων. Οι δύο από τους προτεινόμενους αλγορίθμους προέρχονται από την περιοχή δρομολόγησης οχημάτων, ενώ ο τρίτος αναπτύχθηκε ειδικά για το πρόβλημα των πρακτόρων. Αναλύσεις χρονικής πολυπλοκότητας έδειξαν ότι οι αλγόριθμοι είναι πολυωνυμικοί, ενώ δοκιμές με χρήση κατάλληλου λογισμικού προσομοίωσης παρείχαν πολύτιμες ενδείξεις για την ποιότητα των λύσεων που παράγουν οι αλγόριθμοι, υποδεικνύοντας τον πιο αποτελεσματικό από αυτούς σε διαφορετικές περιοχές του παραμετρικού χώρου. Τέλος, μελετήθηκε ο τρόπος που επηρεάζουν συγκεκριμένες παράμετροι του προβλήματος τη λύση του.

Στο κεφάλαιο 5 μελετήθηκε το πρόβλημα της αναζήτησης πληροφορίας σε σύστημα ομότιμων μεσιτών. Στα πλαίσια των αντιδραστικών μηχανισμών, οι οποίοι βασίζονται στην

ευρεία διάδοση των αιτημάτων αναζήτησης μέσω του δικτύου, προτάθηκε μηχανισμός που αξιοποιεί τις ιδιότητες εντοπιότητας αναφοράς που είναι πιθανό να επιδεικνύουν τα μοντέλα ζήτησης της πληροφορίας. Τα αποτελέσματα των δοκιμών έδειξαν ότι ο προτεινόμενος μηχανισμός διατηρεί τον ίδιο καλό χρόνο εντοπισμού με τον αλγόριθμο της πλημμύρας, ενώ επιτυγχάνει αποταμίευση εύρους ζώνης. Στα πλαίσια προδραστικών μηχανισμών προτάθηκε ένας μηχανισμός διανομής διαφημίσεων πηγών, ο οποίος αξιοποιεί το μοντέλο μικρού-κόσμου, με σκοπό να εξασφαλιστεί μικρός χρόνος εντοπισμού οποιασδήποτε πληροφορίας, πραγματοποιώντας επιλεκτική αποθήκευση των διαφημίσεων από κάθε μεσίτη. Το κεφάλαιο ολοκληρώθηκε με την παρουσίαση πειραματικών αποτελεσμάτων σχετικών με τη δυνατότητα επίτευξης της επιθυμητής ισορροπίας μεταξύ χρόνου εντοπισμού και δαπάνης αποθηκευτικών πόρων.

6.2 ΓΕΝΙΚΑ ΣΥΜΠΕΡΑΣΜΑΤΑ

Στα πλαίσια του προβλήματος συλλογής κατανεμημένης πληροφορίας από κεντρικό μεσίτη, η επικοινωνία του μεσίτη με τους εξωτερικούς παρόχους μπορεί να πραγματοποιηθεί είτε απομακρυσμένα, είτε με αποστολή κινητών πρακτόρων, και επιπλέον με παράλληλη ή σειριακή προσπέλαση των παρόχων. Δεδομένου ότι το σύνολο των παρόχων από τους οποίους θα ανακτηθεί η πληροφορία είναι γνωστό εκ των προτέρων και επιπλέον δεν υπάρχουν περιορισμοί ως προς τη σειρά προσπέλασης αυτών, η παράλληλη προσπέλαση με αποστολή κινητών πρακτόρων κρίθηκε πιο αποτελεσματική για τους εξής λόγους: ο κινητός πράκτορας μετακινεί τη συνιστώσα υπολογισμού κοντά στα δεδομένα προς επεξεργασία, με αποτέλεσμα να εκτελεί τις απαιτούμενες διαδικασίες αναζήτησης τοπικά, εξαλείφοντας την ανάγκη μεταφοράς περιττών δεδομένων μέσω του δικτύου. Εάν επιπλέον η εφαρμογή απαιτεί επεξεργασία μεγάλου όγκου πληροφορίας στον απομακρυσμένο κόμβο, η σημαντική μείωση της πληροφορίας που θα επιστραφεί μέσω του κατάλληλου φιλτραρίσματος που εκτελεί ο πράκτορας υποσκελίζει την επιβάρυνση που προκαλείται από την αποστολή του πράκτορα στον απομακρυσμένο κόμβο. Με τη μετανάστευση στην άλλη πλευρά μιας σύνδεσης, ο πράκτορας μπορεί να συνεχίσει να λειτουργεί ακόμη και αν η σύνδεση διακοπεί, γεγονός το οποίο εκτιμάται ιδιαίτερα σε περιβάλλον κινητής υπολογιστικής, με αναξιόπιστες συνδέσεις χαμηλού εύρους ζώνης και υψηλής καθυστέρησης. Συνεπώς, με την παράλληλη αποστολή πολλών κινητών πρακτόρων, μπορούν να επιλέγονται διαφορετικές στρατηγικές μετακίνησης των πρακτόρων ανάλογα με το έργο και την τρέχουσα κατάσταση του δικτύου, με σκοπό την ταχύτερη εκπλήρωση του έργου.

Δεδομένου ότι ένας αυξημένος αριθμός πρακτόρων συνεπάγεται αυξημένη κατανάλωση πόρων του συστήματος, ο καθορισμός του ελάχιστου αριθμού πρακτόρων και των δρομολογίων τους που εξασφαλίζουν συλλογή της κατανεμημένης πληροφορίας στον

ελάχιστο χρόνο αποτελεί ένα σύνθετο πρόβλημα βελτιστοποίησης. Χαλαρώνοντας ένα βαθμό δυσκολίας του προβλήματος με τη θεώρηση σταθερού αριθμού πρακτόρων, το πρόβλημα αποδείχτηκε ότι είναι *NP-hard*, και για το λόγο αυτό η λύση του προσεγγίστηκε με τη βοήθεια ευριστικών αλγορίθμων πολυωνυμικής πολυπλοκότητας. Η εφαρμογή των Savings και Insertion, οι οποίοι προέρχονται από την ερευνητική περιοχή που ασχολείται με τη δρομολόγηση οχημάτων, εξετάστηκε συγκριτικά με την εφαρμογή του SR-CR, ο οποίος σχεδιάστηκε ειδικά για το συγκεκριμένο πρόβλημα. Τα αποτελέσματα της προσομοίωσης έδειξαν ότι ο SR-CR σε συνδυασμό με τον Savings παράγουν λύσεις που πλησιάζουν τη βέλτιστη σε λογικό χρόνο. Πιο συγκεκριμένα, ο SR-CR αποδεικνύεται ως ο πιο κατάλληλος (από πλευράς ποιότητας λύσεων και χρόνου υπολογισμού) από τους τρεις, για αριθμό πρακτόρων σχετικά μικρό συγκριτικά με τον αριθμό των κόμβων που φέρουν τη ζητούμενη πληροφορία, ενώ ο Savings έρχεται πρώτος στην αντίθετη περίπτωση.

Αξίζει να σημειωθεί ότι η αποδοτική εφαρμογή των προτεινόμενων αλγορίθμων σε ένα σύστημα ανάκτησης κατανεμημένης πληροφορίας ευνοείται ιδιαίτερα από τις εξής συνθήκες: όταν απαιτείται να χρησιμοποιηθεί περιορισμένος αριθμός πρακτόρων, όταν οι πράκτορες έχουν σχετικά μεγάλο αρχικό μέγεθος (όπως για παράδειγμα όταν περιλαμβάνουν σύνθετο κώδικα για την εκτέλεση εξειδικευμένων λειτουργιών φιλτραρίσματος πληροφορίας), όταν μεγάλος όγκος πληροφορίας ανακτάται από κάθε κόμβο ή η κατάσταση του δικτύου είναι κακή. Ο λόγος είναι ότι σε αυτές τις περιπτώσεις ο χρόνος υπολογισμού των δρομολογίων θα είναι αμελητέος συγκριτικά με το χρόνο ολοκλήρωσης της αποστολής ανάκτησης της πληροφορίας.

Ένας αριθμός πειραμάτων εκτελέστηκαν επίσης προκειμένου να διερευνηθεί η σχέση μεταξύ αριθμού πρακτόρων και χρόνου ολοκλήρωσης της συλλογής πληροφορίας σε σχέση με διάφορες παραμέτρους. Τα συμπεράσματα που προέκυψαν συνοψίζονται στα εξής: Ενώ οι εφαρμογές που συνεπάγονται μεταφορά μεγάλου όγκου πληροφοριών δείχνουν να επωφελούνται περισσότερο από μεγάλους πληθυσμούς πρακτόρων, η αποταμίευση χρόνου που επιτυγχάνεται από τη χρησιμοποίηση περισσότερων πρακτόρων γίνεται πιο εμφανής καθώς το αρχικό μέγεθος των πρακτόρων μειώνεται. Σε σχέση με το γράφο που αντιστοιχεί στο δίκτυο, προκύπτει ότι ένας αραιός γράφος επωφελείται σε μικρότερο βαθμό από μια αύξηση του πληθυσμού πρακτόρων σε σχέση με έναν πυκνότερο, ενώ αν επιπλέον υπάρχουν συνδέσεις συμφόρησης στον κόμβο του κεντρικού αποστολέα, το όφελος γίνεται ακόμη μικρότερο.

Η εφαρμογή της προτεινόμενης μεθόδου επεκτείνεται και σε έναν μεταβλητό αριθμό πρακτόρων, παρέχοντας μεγαλύτερη ευελιξία στο σύστημα. Σε αυτήν την περίπτωση, η επιθυμητή ισορροπία μεταξύ του χρόνου ολοκλήρωσης της αποστολής και της ποσότητας των πόρων του συστήματος που καταναλώνονται αποτελεί κρίσιμη απαίτηση για τον

καθορισμό του βέλτιστου αριθμού πρακτόρων. Αξίζει να σημειωθεί ότι η αποτίμηση ενός συνόλου παραμέτρων, όπως αυτές που εξετάστηκαν ως προς τον τρόπο που επηρεάζουν τη λύση, μέσω μιας σύνθετης συνάρτησης με βάρη, θα μπορούσε να αποδειχτεί ιδιαίτερα ενδιαφέρουσα για τον ακριβέστερο προσδιορισμό του βέλτιστου αριθμού πρακτόρων.

Η λύση που δόθηκε στο πρόβλημα των πρακτόρων βασίστηκε στην υπόθεση ότι η πληροφορία για την κατάσταση του δικτύου είναι γνωστή από στατιστικά δεδομένα. Στα πραγματικά δίκτυα όμως, αυτή η πληροφορία υπόκεινται σε διακυμάνσεις και συνεπώς οι πράκτορες θα πρέπει να είναι σε θέση να αντιδρούν σε μη-αναμενόμενες συνθήκες ώστε τελικά να πραγματοποιούν τη λύση που είχε αρχικά υπολογιστεί. Η ενσωμάτωση μιας συνιστώσας που θα λαμβάνει υπόψη την αστάθεια του δικτύου στο προτεινόμενο μοντέλο παρουσιάζει ιδιαίτερο ενδιαφέρον, και για το λόγο αυτό θα μπορούσε να αποτελέσει αντικείμενο μελέτης σε μελλοντική εργασία.

Σχετικά με το πρόβλημα της αναζήτησης πληροφορίας σε σύστημα ομοτίμων μεσιτών και με δεδομένο ότι ο χρόνος αναζήτησης αποτελεί την πιο κρίσιμη παράμετρο, μελετήθηκαν αντιδραστικοί και προδραστικοί μηχανισμοί. Οι μηχανισμοί που εφαρμόζονται με αντιδραστικό τρόπο, βασίζονται στην εξάπλωση των αιτημάτων αναζήτησης μέσω του δικτύου μέχρι να εντοπιστεί η ζητούμενη πληροφορία, ενώ οι προδραστικοί μηχανισμοί συνεπάγονται την εκτέλεση διεργασιών πριν από την εκτέλεση των αιτημάτων αναζήτησης με σκοπό τη βελτίωση της απόδοσης της αναζήτησης.

Στα πλαίσια των αντιδραστικών μηχανισμών, ο πολύ καλός χρόνος απόκρισης σε ένα αίτημα αναζήτησης με την εφαρμογή του αλγορίθμου της πλημμύρας επιτυγχάνεται με αντίτιμο την εξαντλητική χρήση του δικτύου, από πλευράς κατανάλωσης πόρων επικοινωνίας και επεξεργασίας. Ο προτεινόμενος μηχανισμός βασίστηκε σε κατάλληλα περιορισμένες εκδόσεις του αλγορίθμου της πλημμύρας, αξιοποιώντας ιδιότητες εντοπιότητας αναφοράς που είναι πιθανό να παρουσιάζουν τα μοντέλα ζήτησης της πληροφορίας. Τα αποτελέσματα των δοκιμών έδειξαν ότι ο χρόνος εντοπισμού της ζητούμενης πληροφορίας με εφαρμογή του προτεινόμενου μηχανισμού είναι συγκρίσιμος με το χρόνο που απαιτεί ο αλγόριθμος της πλημμύρας, ενώ ταυτόχρονα επιτυγχάνεται αποταμίευση εύρους ζώνης, το μέγεθος της οποίας είναι ανάλογο με το βαθμό εμφάνισης των ιδιοτήτων.

Η αποδοτικότητα του προτεινόμενου μηχανισμού επιτυγχάνεται με αμελητέα επιβάρυνση, αφού οι πίνακες που αποθηκεύουν πληροφορία σχετική με τη ζήτηση της πληροφορίας είναι περιορισμένου μεγέθους, και επίσης δεν απαιτείται επιπλέον κίνηση στο δίκτυο για την ανανέωσή τους. Οι συνθήκες αυτές εξασφαλίζονται δεδομένου ότι ένα μικρό ποσοστό όλων των διαθέσιμων αντικειμένων είναι πιθανό να παρουσιάζουν κάποια ιδιότητα εντοπιότητας, ενώ οι πίνακες ανανεώνονται με αποδοτικούς μηχανισμούς επωμισμού (piggybacking) που παρέχονται από τα Ενεργά Δίκτυα. Τα Ενεργά Δίκτυα επιτρέπουν

ευέλικτη και δυναμική αλλαγή του κώδικα που μεταφέρεται από ένα ενεργό πακέτο, ώστε εκτός από τον κώδικα που υλοποιεί την ερώτηση αναζήτησης να «φορτώνεται» δυναμικά και κώδικας που υλοποιεί λειτουργίες σχετικές με την επίγνωση της ζήτησης της πληροφορίας με προσαρμοστικό τρόπο. Επιπλέον, σε περιβάλλον που χαρακτηρίζεται από ιδιότητες εντοπιότητας αναφοράς, αναμένεται ότι η πλειοψηφία των αιτημάτων θα αναφέρεται σε αντικείμενα που επιδεικνύουν κάποια ιδιότητα, ενώ ο προτεινόμενος μηχανισμός θα προσαρμόζεται ανάλογα με το είδος και το βαθμό των ιδιοτήτων ώστε να επιτυγχάνει μέγιστη αποταμίευση εύρους ζώνης χωρίς να χειροτερεύει το χρόνο εντοπισμού του αλγορίθμου της πλημμύρας.

Αξίζει να σημειωθεί ότι, ενώ οι μηχανισμοί που έχουν προταθεί στη βιβλιογραφία βασίζονται συνήθως σε συγκεκριμένες ιδιότητες της τοπολογίας (π.χ. λίγοι κόμβοι με υψηλό βαθμό συνδετικότητας και πολλοί με χαμηλό) ώστε να περιορίσουν τις δαπάνες του αλγορίθμου της πλημμύρας, ο προτεινόμενος μηχανισμός εκμεταλλεύεται ιδιότητες της ζήτησης πληροφορίας, και εφαρμόστηκε σε τυχαίους γράφους. Το γεγονός αυτό θα μπορούσε να αποτελέσει τη βάση για περαιτέρω μελέτη, ώστε ο προτεινόμενος μηχανισμός να προσαρμόζεται ανάλογα και με τον τύπο του δικτύου για ακόμη καλύτερη απόδοση.

Στα πλαίσια των προδραστικών μηχανισμών προτάθηκε ένας μηχανισμός διανομής διαφημίσεων πηγών, ο οποίος βασίστηκε στην εφαρμογή του μοντέλου μικρού-κόσμου όσον αφορά την αποθήκευση των διαφημίσεων από κάθε μεσίτη, έτσι ώστε με κατάλληλη αναζήτηση, βάσει ενός άπληστου αλγορίθμου, να εντοπίζεται γρήγορα οποιαδήποτε πηγή. Τα πειραματικά αποτελέσματα επιβεβαίωσαν τον περιορισμένο χρόνο εντοπισμού οποιασδήποτε πληροφορίας που εγγυάται το θεωρητικό μοντέλο, αναδεικνύοντας την αξία του μηχανισμού σε περιπτώσεις περιορισμένης αποθηκευτικής ικανότητας των μεσιτών. Επιπλέον, ο προτεινόμενος μηχανισμός αποδείχτηκε περισσότερο αποτελεσματικός καθώς μειώνεται ο βαθμός εγγύτητας περιεχομένου μεταξύ των τοπικών πηγών ενός μεσίτη.

Οι παραπάνω μηχανισμοί είναι πλήρως αποκεντρωμένοι, και η αναζήτηση δεν αποτυγχάνει εξαιτίας της πιθανής αποτυχίας μεμονωμένων κόμβων ή συνδέσεων. Αντίθετα, στην περίπτωση του αντιδραστικού μηχανισμού, κάθε αίτημα αναζήτησης προωθείται σε πολλαπλούς κόμβους σε κάθε βήμα, ενώ στην περίπτωση του προδραστικού μηχανισμού κάθε διαφήμιση αποθηκεύεται από πολλαπλούς κόμβους, με αποτέλεσμα να αποφεύγονται τα μεμονωμένα σημεία αποτυχίας. Επίσης, οι μηχανισμοί προσαρμόζονται δυναμικά στις αλλαγές που συμβαίνουν στο περιβάλλον (ως προς τη ζήτηση της πληροφορίας, την εισαγωγή νέων πηγών, κ.λπ.), με σκοπό την εξασφάλιση αποτελεσματικότητας και αποδοτικότητας στην αναζήτηση. Εξαιτίας της ανθεκτικότητας και προσαρμοστικότητάς τους, οι προτεινόμενοι μηχανισμοί θεωρούνται κατάλληλοι και για περιβάλλοντα διεισδυτικής υπολογιστικής. Σημειώνεται ωστόσο, ότι η μελέτη των μηχανισμών αυτών στην παρούσα

διατριβή πραγματοποιήθηκε από αλγοριθμική σκοπιά, θεωρώντας σταθερή την υποκείμενη δικτυακή υποδομή. Η εξέταση των μηχανισμών υπό την ενσωμάτωση επιπλέον βαθμών δυναμικότητας, όπως δυναμικά μεταβαλλόμενης τοπολογίας, θα μπορούσε να αποτελέσει αντικείμενο μελλοντικής εργασίας.

ΠΑΡΑΡΤΗΜΑ

1. ΠΡΟΒΛΗΜΑ ΔΡΟΜΟΛΟΓΗΣΗΣ ΟΧΗΜΑΤΩΝ

1.1 ΠΕΡΙΓΡΑΦΗ ΚΑΙ ΔΙΑΤΥΠΩΣΗ ΠΡΟΒΛΗΜΑΤΟΣ

Το Πρόβλημα Δρομολόγησης Οχημάτων (Vehicle Routing Problem - VRP) αποτελεί γενίκευση μιας κατηγορίας προβλημάτων που έχει μελετηθεί τόσο ως θεωρητικό πρόβλημα όσο και για την πρακτική αξία του σε εφαρμογές. Υπάρχουν διάφορες παραλλαγές του VRP, εντούτοις το βασικό πρόβλημα αναφέρεται σε ένα σύνολο πελατών που εξυπηρετούνται από έναν στόλο οχημάτων, τα οποία αποστέλλονται από κεντρικό σταθμό με σκοπό τη διανομή αγαθών σε αυτούς. Κάθε πελάτης έχει συγκεκριμένη απαίτηση για ποσότητα αγαθών και υπάρχουν όρια τόσο στη χωρητικότητα των οχημάτων όσο και στο μέγιστο χρόνο ή την απόσταση που οποιοδήποτε όχημα μπορεί να ταξιδέψει. Σκοπός είναι να βρεθεί ένα σύνολο διαδρομών, οι οποίες ελαχιστοποιούν το συνολικό κόστος διανομής αγαθών στους πελάτες. Αυτή η διατύπωση καλύπτει ένα ευρύ φάσμα εφαρμογών στον πραγματικό κόσμο, όχι μόνο παράδοση αγαθών από το όχημα αλλά και συλλογή, καθώς και άλλους τύπους παραδόσεων (π.χ. ταχυδρομικές διαδρομές). Διάφορες επεκτάσεις στο βασικό VRP έχουν μελετηθεί, συμπεριλαμβανομένης της τοποθέτησης χρονικών περιορισμών στις παραδόσεις, της ύπαρξης πολλών κεντρικών σταθμών, και της περίπτωσης πραγματοποίησης παράδοσης και συλλογής ταυτόχρονα [1]. Αξίζει επίσης να σημειωθεί ότι ο αριθμός οχημάτων που χρησιμοποιείται, είτε καθορίζεται εκ των προτέρων, είτε αποτελεί μεταβλητή απόφασης. Στη δεύτερη περίπτωση, σκοπός είναι η ελαχιστοποίηση της συνολικής απόστασης που διανύεται (ή του χρόνου που ξοδεύεται) με ταυτόχρονη ελαχιστοποίηση του αριθμού οχημάτων.

Το VRP θεωρείται από τα δυσκολότερα προβλήματα γραμμικού προγραμματισμού, τόσο εξαιτίας του μεγάλου αριθμού μεταβλητών που περιλαμβάνει, όσο και επειδή ασχολείται ταυτόχρονα με κατανομή πόρων και καθορισμό δρομολογίων. Ειδικότερα, συνδυάζει δύο προβλήματα: το *Πρόβλημα Γενικευμένης Ανάθεσης (Generalized Assignment problem - GAP)* και το *Πρόβλημα Περιοδεύοντος Πωλητή (Traveling Salesman Problem - TSP)* [2]. Ακολουθεί η διατύπωση γραμμικού προγραμματισμού των Fisher-Jaikumar [3]:

Μεταβλητές

K = αριθμός οχημάτων.

n = αριθμός πελατών στους οποίους πρέπει να γίνει παράδοση. Οι πελάτες αριθμούνται από 1 έως n , και ο δείκτης 0 αναφέρεται στον κεντρικό σταθμό.

b_k = χωρητικότητα οχήματος k .

a_i = μέγεθος παράδοσης στον πελάτη i .

c_{ij} = κόστος της απευθείας μετάβασης από τον πελάτη i στον πελάτη j .

Μεταβλητές Απόφασης

$$y_{ik} = \begin{cases} 1, & \text{αν ο πελάτης } i \text{ εξυπηρετείται από το όχημα } k \\ 0, & \text{διαφορετικά} \end{cases}$$

$$x_{ijk} = \begin{cases} 1, & \text{αν το όχημα } k \text{ οδεύει κατευθείαν από τον πελάτη } i \text{ στον πελάτη } j \\ 0, & \text{διαφορετικά} \end{cases}$$

Η διατύπωση γραμμικού προγραμματισμού του προβλήματος δρομολόγησης που αποσκοπεί σε ελαχιστοποίηση του κόστους, δεδομένων των περιορισμών χωρητικότητας των οχημάτων, δίνεται στη συνέχεια:

$$\min \sum_{ijk} c_{ij} x_{ijk} \tag{1}$$

Περιορισμοί:

$$\sum_i a_i y_{ik} \leq b_k, \quad k = 1, \dots, K \tag{2}$$

$$\sum_k y_{ik} = \begin{cases} K, & i = 0 \\ 1, & i = 1, \dots, n \end{cases} \tag{3}$$

$$y_{ik} = 0 \text{ ή } 1, \quad \begin{matrix} i = 0, \dots, n \\ k = 1, \dots, K \end{matrix} \tag{4}$$

$$\sum_i x_{ijk} = y_{jk}, \quad j = 0, \dots, n \tag{5}$$

$$\sum_j x_{ijk} = y_{ik}, \quad i = 0, \dots, n \tag{6}$$

$$\sum_{ij \in S \times S} x_{ijk} \leq |S| - 1, \quad \begin{matrix} S \subseteq \{1, \dots, n\} \\ 2 \leq |S| \leq n - 1 \end{matrix} \tag{7}$$

$$x_{ijk} = 0 \text{ ή } 1, \quad \begin{matrix} i = 0, \dots, n \\ j = 0, \dots, n \end{matrix} \tag{8}$$

} $k = 1, \dots, K$

Δύο γνωστά συνδυαστικά προβλήματα βελτιστοποίησης συνυπάρχουν μέσα σε αυτή τη διατύπωση. Οι περιορισμοί (2)-(4) είναι οι περιορισμοί ενός Προβλήματος Γενικευμένης Ανάθεσης και εξασφαλίζουν ότι κάθε διαδρομή αρχίζει και τελειώνει στον κεντρικό σταθμό (πελάτης 0), ότι κάθε πελάτης εξυπηρετείται από κάποιο όχημα, και ότι το φορτίο που μεταφέρεται από οποιοδήποτε όχημα δεν πρέπει να ξεπερνά την χωρητικότητά του. Για κάθε k με y_{ik} να ικανοποιεί τις εξισώσεις (2)-(4), οι περιορισμοί (5)-(8) καθορίζουν ένα Πρόβλημα Περιοδούντος Πωλητή στους πελάτες που είχαν προηγουμένως ανατεθεί στο όχημα k .

1.2 ΛΥΣΗ

Ως γενίκευση του Προβλήματος Περιοδεύοντος Πωλητή, το VRP είναι *NP-hard* [4], που σημαίνει ότι δεν υπάρχει καμία γνωστή μέθοδος εύρεσης της βέλτιστης λύσης μέσα σε πολυωνυμικό χρόνο. Συνεπώς, το ενδιαφέρον εστιάζεται σε ευριστικούς αλγορίθμους.

Οι οικογένειες ευριστικών που έχουν προταθεί για το VRP ταξινομούνται σε δυο κατηγορίες: στους *Κλασικούς Ευριστικούς*, οι οποίοι αναπτύχθηκαν μεταξύ 1960 και 1990, και στους *Μετα-ευριστικούς*, η διάδοση των οποίων πραγματοποιήθηκε κυρίως τη τελευταία δεκαετία. Οι περισσότερες τυποποιημένες διαδικασίες δόμησης και βελτίωσης λύσεων που χρησιμοποιούνται σήμερα ανήκουν στην πρώτη κατηγορία. Οι αλγόριθμοι αυτοί εκτελούν μια σχετικά περιορισμένη εξερεύνηση του διαστήματος αναζήτησης λύσεων και γενικά παράγουν λύσεις καλής ποιότητας, σε μέτριους χρόνους. Επιπλέον, οι περισσότεροι από αυτούς μπορούν να προσαρμοστούν σχετικά εύκολα για να αντιμετωπίσουν την ποικιλομορφία των περιορισμών που παρουσιάζονται στα πλαίσια πραγματικών εφαρμογών και επομένως χρησιμοποιούνται ευρέως σε εμπορικές εφαρμογές. Στη περίπτωση των *Μετα-ευριστικών αλγορίθμων*, πραγματοποιείται σχολαστική εξερεύνηση των υποσχόμενων περιοχών του διαστήματος λύσεων. Οι αλγόριθμοι αυτοί συνδυάζουν τα χαρακτηριστικά των περίπλοκων κανόνων αναζήτησης γειτονικών λύσεων, των δομών μνήμης, και του επανασυνδυασμού των λύσεων. Η ποιότητα των λύσεων που παράγουν είναι συνήθως πολύ καλύτερη από αυτή των *Κλασικών Ευριστικών*, αλλά το τίμημα είναι ο αυξημένος χρόνος υπολογισμού. Επιπλέον, η προσαρμογή των εν λόγω αλγορίθμων στις απαιτήσεις του εκάστοτε προβλήματος απαιτεί ιδιαίτερα προσεκτική επιλογή των τιμών των παραμέτρων, καθιστώντας την επέκτασή τους σε διαφορετικές καταστάσεις δύσκολη. Από μία άποψη, οι *Μετα-ευριστικοί* αποτελούν περίπλοκες διαδικασίες βελτίωσης λύσεων και μπορούν απλά να αντιμετωπισθούν ως φυσικές προεκτάσεις των *Κλασικών Ευριστικών* [5].

1.2.1 ΚΛΑΣΙΚΟΙ ΕΥΡΙΣΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Οι *Κλασικοί Ευριστικοί* για το VRP ταξινομούνται σε τρεις κατηγορίες: Στους *ευριστικούς δόμησης λύσης (constructive heuristics)*, στους *ευριστικούς δύο-φάσεων (two-phase heuristics)* και στους *ευριστικούς βελτίωσης (improvement heuristics)*. Οι *ευριστικοί δόμησης λύσης* καταστρώνουν βαθμιαία μια εφικτή λύση παρακολουθώντας το κόστος της. Με τους *ευριστικούς δύο-φάσεων*, το πρόβλημα αποσυντίθεται σε δύο στάδια: την ανάθεση των κόμβων σε εφικτές διαδρομές και την πραγματική κατασκευή των διαδρομών, με πιθανά συστήματα ανατροφοδότησης πληροφοριών μεταξύ των δύο αυτών σταδίων. Κατά συνέπεια, οι *ευριστικοί δύο-φάσεων* διαιρούνται σε δυο κατηγορίες: στην κατηγορία «*πρώτον-συστάδες, δεύτερον-διαδρομές*» και στην κατηγορία «*πρώτον-διαδρομή, δεύτερον-συστάδες*». Στην πρώτη κατηγορία, οι κόμβοι οργανώνονται αρχικά σε εφικτές συστάδες, και

στη συνέχεια μια διαδρομή κατασκευάζεται για κάθε μια από αυτές. Στη δεύτερη κατηγορία, αρχικά κατασκευάζεται μια διαδρομή γύρω από όλους τους κόμβους, και στη συνέχεια αυτή διαιρείται σε εφικτές διαδρομές οχημάτων. Τέλος, οι ευριστικοί βελτιώσης προσπαθούν να βελτιώσουν οποιαδήποτε εφικτή λύση με την εκτέλεση μιας ακολουθίας ανταλλαγών συνδέσεων ή κόμβων εντός ή μεταξύ των διαδρομών των οχημάτων. Η διάκριση μεταξύ των ευριστικών δόμησης λύσης και αυτών της βελτιώσης είναι συχνά δυσδιάκριτη δεδομένου ότι οι περισσότεροι αλγόριθμοι δόμησης λύσης περιλαμβάνουν και βήματα βελτιώσεων (όπως 3-opt [6]) στα διάφορα στάδια [5]. Στη συνέχεια, θα περιγραφούν οι πιο γνωστοί αλγόριθμοι των δυο πρώτων κατηγοριών.

1.2.1.1 Ευριστικοί Αλγόριθμοι Δόμησης Λύσης

❖ Clarke & Wright [7]

Ο αλγόριθμος αποταμίευσης Clarke και Wright υλοποιείται και ως παράλληλος και ως σειριακός. Στη παράλληλη έκδοση, τα δρομολόγια διαμορφώνονται ταυτόχρονα. Αρχικά, κάθε πελάτης εξυπηρετείται από ένα ξεχωριστό όχημα. Η ικανοποίηση των αιτημάτων των πελατών i και j από το ίδιο όχημα θα οδηγήσει σε αποταμίευση $s_{ij} = c_{1i} + c_{1j} - c_{ij}$. Οι αποταμιεύσεις που υπολογίζονται κατά το τρόπο αυτό αποθηκεύονται με φθίνουσα σειρά. Ο αλγόριθμος συγχωνεύει τους πελάτες i και j που αντιστοιχούν στην υψηλότερη αποταμίευση s_{ij} χωρίς παραβίαση του περιορισμού χωρητικότητας του οχήματος και συνεχίζει με τον ίδιο τρόπο έως ότου δεν είναι δυνατές άλλες συγχωνεύσεις. Αν και τα δρομολόγια αναπτύσσονται ταυτόχρονα, μόνο δύο δρομολόγια συγχωνεύονται σε κάθε βήμα του αλγορίθμου. Στη σειριακή έκδοση, τα δρομολόγια διαμορφώνονται σειριακά. Αντί λοιπόν να υπολογίζεται όλος ο πίνακας αποταμίευσης, ένας αυθαίρετος πελάτης επιλέγεται για το πρώτο στοιχειώδες δρομολόγιο. Σε κάθε βήμα του αλγορίθμου, η σύνδεση στον τελευταίο προστιθέμενο πελάτη που αντιστοιχεί στην υψηλότερη αποταμίευση επιλέγεται, εφ' όσον δεν παραβιάζει αυτή η προσθήκη τον περιορισμό χωρητικότητας. Μόλις δεν μπορεί να επεκταθεί το παρόν δρομολόγιο περισσότερο, ένα νέο δρομολόγιο διαμορφώνεται και ούτω καθεξής μέχρις ότου εξυπηρετηθούν όλοι οι πελάτες.

❖ Altinkemer [8]

Ο συγγραφέας περιγράφει έναν παράλληλο αλγόριθμο αποταμίευσης για την παραγωγή εφικτών λύσεων στο πρόβλημα παράδοσης. Ο αλγόριθμος βασίζεται στην ιδέα της αποταμίευσης που εισήχθη αρχικά από τους Clarke και Wright. Η κύρια διαφορά εντοπίζεται στην αντικατάσταση της διαδοχικής διαδικασίας συγχώνευσης που επιτρέπει μόνο δύο δρομολόγια να συγχωνεύονται σε κάθε βήμα του αλγορίθμου, από μια διαδικασία συνταιριάσματος που συγχωνεύει πολλά δρομολόγια σε κάθε επανάληψη. Ο αριθμός των συγχωνεύσεων σε κάθε επανάληψη καθορίζεται με την επίλυση ενός προβλήματος

συνταιριάσματος (*matching problem*), που μεγιστοποιεί τη συνολική αποταμίευση που λαμβάνεται σε κάθε επανάληψη.

Πιο συγκεκριμένα, στην πρώτη επανάληψη, κάθε συστάδα αποτελείται από έναν μοναδικό κόμβο του αρχικού γράφου, με V το σύνολο των κόμβων του και E το σύνολο των ακμών του. Όταν δύο συστάδες συγχωνευτούν, οι κόμβοι σε εκείνες τις συστάδες εξυπηρετούνται από το ίδιο όχημα. Η αποταμίευση που προκύπτει από τη συγχώνευση των συστάδων p και q , δίνεται από $S_{pq} = C(R_p) + C(R_q) - C(R_p \cup R_q)$, όπου R_k είναι το σύνολο κόμβων στη συστάδα k , και $C(R_k)$ είναι το μήκος της διαδρομής που προκύπτει από τη βέλτιστη λύση του προβλήματος περιοδευόντος πωλητή στο R_k . Η συγχώνευση των συστάδων p και q είναι μη-αποδεκτή, εάν παραβιάζει τον περιορισμό χωρητικότητας. Το πρόβλημα συνταιριάσματος μέγιστου κόστους με βάρη (*maximum cost weighted matching problem*) για το γράφο $G(V, E, S)$, όπου το S αντιπροσωπεύει τις καταχωρήσεις αποταμιεύσεων, τίθεται προς επίλυση. Νέες συστάδες διαμορφώνονται με τη συγχώνευση των συστάδων προς συνταίριασμα. Η διαδικασία επαναλαμβάνεται έως ότου καμιά περαιτέρω συγχώνευση δεν είναι αποδεκτή. Διάφορες παραλλαγές του βασικού αλγορίθμου είναι δυνατές, μια από τις οποίες καταφεύγει σε προσέγγιση των $C(S_k)$ τιμών αντί για τον ακριβή υπολογισμό τους.

❖ Wark [9]

Οι συγγραφείς περιγράφουν έναν ευριστικό που εφαρμόζει επαναλαμβανόμενο συνταίριασμα για να συγχωνεύει διαδοχικά συστάδες, που ορίζονται ως τα διαταγμένα σύνολα κόμβων, στα άκρα τους. Το κόστος συνταιριάσματος μπορεί να οριστεί ως η συνηθισμένη αποταμίευση, ή αυτό μπορεί να τροποποιηθεί για να ευνοήσει τις συγχωνεύσεις συστάδων που πετυχαίνουν συνολικό βάρος αρκετά μικρότερο από την χωρητικότητα των οχημάτων, ή χρόνο διαδρομής αρκετά μικρότερο από το μέγιστο χρόνο διαδρομής. Η διαδικασία αρχίζει με τη θεώρηση του συνόλου συστάδων για το οποίο όλες οι συστάδες είναι μεγέθους 1, δηλαδή κάθε πελάτης αποτελεί ξεχωριστή συστάδα, και συγχωνεύει διαδοχικά τις συστάδες. Αν κάποια στιγμή όλες οι συστάδες συνταιριάζουν με τους εαυτούς τους μόνο, τότε μερικές από αυτές διασπώνται με δεδομένη πιθανότητα. Η διαδικασία αναπτύσσει έτσι ένα δέντρο που αποτελείται από τα σύνολα συστάδων, από τα οποία η καλύτερη λύση-σύνολο συστάδων επιλέγεται. Για περαιτέρω βελτιώσεις, ένα νέο σύνολο συστάδων διαμορφώνεται λαμβάνοντας το καλύτερο σύνολο συστάδων που βρίσκεται στο προηγούμενο δέντρο και διασπώντας κάθε συστάδα μεγέθους μεγαλύτερης από ένα σε δύο συστάδες. Αυτό διαμορφώνει το «σπόρο» για ένα νέο δέντρο, και ολόκληρη η διαδικασία επαναλαμβάνεται. Ο κύριος περιορισμός του αλγορίθμου είναι ότι απαιτεί ένα μεγάλο ποσό χρόνου υπολογισμού. Για το λόγο αυτό, οι συγγραφείς ερευνούν τη χρήση του παράλληλου υπολογισμού για να μειώσουν το χρόνο που απαιτείται.

❖ Mole [10]

Με βάση το γενικευμένο κριτήριο αποταμίευσης των Clarke και Wright, μια νέα μέθοδος δρομολόγησης οχημάτων προτείνεται, η οποία βασίζεται στη διαδοχική παραγωγή των διαδρομών. Ο διαδοχικός αλγόριθμος παραγωγής διαδρομών που χρησιμοποιεί ένα παραμετροποιημένο κριτήριο αποταμίευσης εξελίσσεται σε μια σειρά επανάληψης τριών βημάτων, υπό τον όρο ότι μια αρχική διαδρομή με έναν πελάτη κατασκευάζεται αρχικά: Στο πρώτο βήμα, καθορίζεται η πιο συμφέρουσα και εφικτή τοποθέτηση κάθε πελάτη που δεν έχει τοποθετηθεί σε διαδρομή, στην υπό κατασκευή διαδρομή, ως αυτήν που προκαλεί την ελάχιστη επιβάρυνση. Στο δεύτερο βήμα, επιλέγεται ο πελάτης που θα τοποθετηθεί στη διαδρομή, ως αυτός που προκαλεί τη μέγιστη αποταμίευση από όλους οι πελάτες που θα μπορούσαν να παρεμβληθούν. Ο επιλεγμένος τοποθετείται λοιπόν στη θέση που προσδιορίστηκε κατά τη διάρκεια του πρώτου βήματος. Στο τρίτο βήμα, η τρέχουσα διαδρομή βελτιστοποιείται με τη βοήθεια μιας 3-opt διαδικασίας. Εάν καμία εισαγωγή δεν είναι εφικτή στο πρώτο βήμα, μια νέα διαδρομή αρχικοποιείται και η διαδικασία εξελίσσεται με παρόμοιο τρόπο. Όταν όλοι οι πελάτες τοποθετηθούν σε διαδρομές, μια διαδικασία βελτιστοποίησης ενεργοποιείται. Ένας πελάτης ενδέχεται να μεταφερθεί από μια διαδρομή σε μια άλλη εάν η επιβάρυνση σε αυτή τη περίπτωση είναι μικρότερη, και οι περιορισμοί οχημάτων και διαδρομών δεν παραβιάζονται.

1.2.1.2 Ευριστικοί Αλγόριθμοι Δυο-Φάσεων

❖ Fisher [11]

Οι Fisher και Jaikumar παρουσιάζουν έναν ευριστικό σύμφωνα με τον οποίο ο τρόπος ανάθεσης των πελατών στα οχήματα καθορίζεται από την επίλυση ενός προβλήματος γενικευμένης ανάθεσης με αντικειμενική συνάρτηση που προσεγγίζει το κόστος παράδοσης. Συγκεκριμένα, τα σημεία αρχικοποίησης j_k καθορίζονται αρχικά, προκειμένου να αρχικοποιηθεί κάθε συστάδα. Ο αριθμός των σημείων αρχικοποίησης είναι ίσος με τον αριθμό οχημάτων K . Στη συνέχεια, το κόστος τοποθέτησης κάθε πελάτη σε κάθε συστάδα d_{ij_k} υπολογίζεται ως το κόστος τοποθέτησης του πελάτη i στη διαδρομή (κεντρικός σταθμός, j_k , κεντρικός σταθμός). Η λύση του γενικευμένου προβλήματος ανάθεσης με τιμές κόστους d_{ij_k} , βάρη πελατών q_i και χωρητικότητα οχημάτων Q , καθορίζει μια εφικτή ανάθεση των πελατών στα οχήματα. Η σειρά προσπέλασης των πελατών από το όχημα καθορίζεται με την εφαρμογή οποιουδήποτε ευριστικού αλγορίθμου του προβλήματος περιοδεύοντος πωλητή. Τέλος, αξίζει να σημειωθεί ότι οι συγγραφείς προτείνουν επιπλέον μια γεωμετρική μέθοδο για την επιλογή των σημείων αρχικοποίησης. Η μέθοδος βασίζεται σε διχοτόμηση του χώρου σε K κώνους σύμφωνα με τα βάρη των πελατών. Τα σημεία αρχικοποίησης είναι εικονικοί πελάτες που βρίσκονται κατά μήκος των ακρινών που διχοτομούν τους κώνους.

❖ Baker [12]

Ο προτεινόμενος ευριστικός είναι βασισμένος στη γενικευμένη προσέγγιση ανάθεσης των Fisher και Jaikumar [3]. Αυτό απαιτεί την επιλογή ενός σημείου αρχικοποίησης για κάθε συστάδα, και προσέγγιση του κόστους τοποθέτησης κάθε πελάτη σε μια συστάδα. Κατόπιν, ένα πρόβλημα ανάθεσης ελαχίστου κόστους, υπό τον όρο ότι δεν παραβιάζονται οι χωρητικότητες των οχημάτων, καταλήγει σε ένα γενικευμένο πρόβλημα ανάθεσης (GAP), και στη συνέχεια ένα πρόβλημα περιοδεύοντος πωλητή (TSP) πρέπει να λυθεί για κάθε όχημα. Ο ευριστικός που λύνει το GAP χρησιμοποιεί μια χαλάρωση των δευτερευόντων περιορισμών, τύπου lagrangean. Εντούτοις, τα αποτελέσματα δείχνουν ότι, κατά την εφαρμογή ευριστικού του GAP, η βέλτιστη λύση για το GAP δεν αντιστοιχεί απαραίτητως στη βέλτιστη λύση για το πρόβλημα δρομολόγησης οχημάτων. Μια καλύτερη λύση είναι πιθανό να βρεθεί με την εξέταση διάφορων γειτονικών λύσεων της βέλτιστης του GAP. Από αυτή την άποψη, ως επόμενο βήμα, οι συγγραφείς προσπάθησαν να βρουν τα σημεία αρχικοποίησης, που ελαχιστοποιούν τη τιμή της βέλτιστης λύσης του GAP, με την ελπίδα ότι αυτό μπορεί να δώσει καλύτερες λύσεις στο πρόβλημα δρομολόγησης οχημάτων. Η προσέγγισή τους είναι να αρχίσουν από τα σημεία αρχικοποίησης που παράγονται με την αυτόματη μέθοδο των Fisher και Jaikumar, και να χρησιμοποιήσουν μια επαναληπτική διαδικασία για να βρίσκουν διαδοχικά νέα σημεία αρχικοποίησης που δίνουν διαδοχικά μικρότερες τιμές στο GAP. Δύο μέθοδοι εφαρμόστηκαν επιτυχώς για το σκοπό αυτό.

❖ Bramel [13]

Ένας ευριστικός δυο-φάσεων εισήχθη από τους Bramel και Simchi-Levi. Κατ' αρχάς, τα σημεία αρχικοποίησης καθορίζονται με την επίλυση ενός προβλήματος θέσης με βάση τις χωρητικότητες (*Capacitated Location Problem - CLP*) και στη συνέχεια, οι πελάτες εισάγονται στις διαδρομές που κατασκευάζονται διαδοχικά. Στην πρώτη φάση, τα K σημεία αρχικοποίησης, αποκαλούμενα ως συμπυκνωτές, επιλέγονται μεταξύ των θέσεων των n πελατών, έτσι ώστε να ελαχιστοποιηθεί η συνολική απόσταση των πελατών από το πιο κοντινό στον καθένα σημείο αρχικοποίησης, εξασφαλίζοντας τους περιορισμούς χωρητικότητας. Στη δεύτερη φάση, οι διαδρομές οχημάτων κατασκευάζονται, εισάγοντας σε κάθε βήμα τον πελάτη στη διαδρομή με το ελάχιστο κόστος εισαγωγής. Εξετάζοντας μια μερική διαδρομή j , $(x_0 = x_{j_0}, x_{j_1}, x_{j_2}, \dots, x_{j_p}, x_{j_{p+1}})$, οι συγγραφείς προτείνουν δύο προσεγγίσεις για το κόστος εισαγωγής c_{ij} ενός πελάτη i στη διαδρομή j : το κόστος άμεσης εισαγωγής, δηλαδή $c_{ij} = \min_{l=0, \dots, p} \{2d_{ij_l}\}$, και το κόστος κοντινότερης εισαγωγής, δηλαδή $c_{ij} = \min_{l=0, \dots, p} \{d_{j_l i} + d_{ij_{l+1}} - d_{j_l j_{l+1}}\}$. Αποδεικνύεται ότι ο αλγόριθμος που καθορίζεται από τον πρώτο κανόνα είναι ασυμπτωτικά βέλτιστος.

❖ Gillet [14]

Οι Gillet και Miller πρότειναν τον αλγόριθμο «σάρωσης», κατά τον οποίο η θέση κάθε πελάτη αντιπροσωπεύεται σε ένα πολικό σύστημα συντεταγμένων που έχει αρχή τη θέση του κεντρικού σταθμού, ως (θ_i, ρ_i) , όπου θ_i είναι η γωνία και ρ_i είναι το μήκος της ακτίνας που αντιστοιχεί στη θέση του πελάτη i . Αντιστοιχίζοντας $\theta_i^* = 0$ σε έναν αυθαίρετο πελάτη, οι γωνίες των υπόλοιπων πελατών υπολογίζονται ανάλογα και οι πελάτες ταξινομούνται κατά αυξανόμενη σειρά του θ_i . Επιλέγοντας ένα αχρησιμοποίητο όχημα k και ξεκινώντας από τον πελάτη με τη μικρότερη γωνία που δεν έχει εκχωρηθεί ακόμα σε όχημα, οι πελάτες εκχωρούνται διαδοχικά στο όχημα k , όσο η χωρητικότητα του οχήματος ή το μέγιστο μήκος διαδρομής δεν παραβιάζονται. Διαφορετικά, ένα νέο όχημα επιλέγεται και η σάρωση συνεχίζεται, με τις αναθέσεις να γίνονται τώρα στο νέο όχημα. Όταν όλοι οι πελάτες εκχωρηθούν σε οχήματα, κάθε διαδρομή βελτιστοποιείται χωριστά, με την επίλυση του αντίστοιχου προβλήματος περιοδεύοντος πωλητή.

1.2.2 ΜΕΤΑ-ΕΥΡΙΣΤΙΚΟΙ ΑΛΓΟΡΙΘΜΟΙ

Τα τελευταία χρόνια διάφοροι μεταευριστικοί έχουν προταθεί για το VRP. Αυτοί αποτελούν γενικές διαδικασίες εύρεσης λύσης που εξερευνούν το διάστημα λύσεων για να εντοπίσουν τις καλές λύσεις και συχνά εμπεριέχουν μερικούς από τους τυποποιημένους ευριστικούς κατασκευής διαδρομών και βελτίωσης λύσεων που περιγράφονται παραπάνω. Με σημαντική διαφοροποίηση από τις κλασσικές προσεγγίσεις, οι μεταευριστικοί επιτρέπουν και χειροτέρευση ακόμη και μη-εφικτών ενδιάμεσων λύσεων κατά τη διάρκεια της διαδικασίας αναζήτησης. Οι πιο γνωστοί μεταευριστικοί που αναπτύχθηκαν για το VRP τυπικά εντοπίζουν καλύτερα τοπικά βέλτιστα από τους κλασσικούς ευριστικούς, αλλά καταναλώνοντας περισσότερο χρόνο [15].

Ειδικότερα, με δεδομένη μια λύση χ , η μέθοδος επιλέγει μεταξύ των λύσεων στη γειτονιά της $N(\chi)$, την επόμενη λύση $\bar{\chi}$, σύμφωνα με κάποιο κανόνα. Η διαδικασία επαναλαμβάνεται με τη $\bar{\chi}$ να αντικαθιστά τη χ (ή σταματάει όταν ικανοποιείται κάποιο κριτήριο λήξης) κ.ο.κ. Κατά συνέπεια ένας μεταευριστικός χαρακτηρίζεται από:

- Τη μέθοδο επιλογής μιας αρχικής λύσης.
- Το καθορισμό της γειτονιάς $N(\chi)$ μιας λύσης χ .
- Το κανόνα για την επιλογή της επόμενης λύσης από τη γειτονιά $N(\chi)$.
- Το κριτήριο λήξης.

Αν και ο καθορισμός της γειτονιάς εξαρτάται συχνά από το εκάστοτε πρόβλημα, μερικές γενικές κατηγορίες διαδικασιών για την παραγωγή γειτονικών λύσεων έχουν αναπτυχθεί. Μια τέτοια κατηγορία αποτελούν οι *γενετικοί αλγόριθμοι* (*genetic algorithms*). Σε μερικές περιπτώσεις, οι γειτονιές αλλάζουν δυναμικά, και μπορούν να εξαρτώνται όχι μόνο

από την τρέχουσα λύση, αλλά και από διάφορες προηγούμενες λύσεις. Η μέθοδος της αναζήτησης ταμπού (*tabu search method*) εμπίπτει σε αυτήν την κατηγορία διαδικασιών.

Το κριτήριο για την επιλογή μιας λύσης μέσα από μια γειτονιά είναι συνήθως το κόστος της λύσης, αλλά μερικές φορές ένα πιο σύνθετο κριτήριο υιοθετείται, το οποίο βασίζεται στα διάφορα χαρακτηριστικά του προβλήματος ή/και στις εκτιμήσεις παραβίασης περιορισμών. Αρκετά συχνά, όπως στη περίπτωση της μεθόδου προσομοιωμένης απόπτησης (*simulated annealing method*), χρησιμοποιείται και ο τυχαίος μηχανισμός.

Σχετικά με το κριτήριο λήξης, πολλές μέθοδοι αναζήτησης βασίζονται στη βελτίωση του κόστους (*cost improvement*), και σταματούν όταν δεν μπορεί να βρεθεί καλύτερη λύση μέσα στην τρέχουσα γειτονιά. Αυτό σημαίνει ότι αυτές οι μέθοδοι σταματούν σε ένα τοπικό ελάχιστο, δηλαδή σε μια λύση που δεν είναι χειρότερη από όλες τις άλλες λύσεις μέσα στη γειτονιά της. Δυστυχώς όμως, για πολλά προβλήματα, ένα τοπικό ελάχιστο μπορεί να απέχει αρκετά από τη βέλτιστη λύση, ιδιαίτερα εάν η γειτονιά που χρησιμοποιείται είναι σχετικά μικρή. Κατά συνέπεια, για τις μεθόδους που βασίζονται στη βελτίωση κόστους, η χρησιμοποίηση μιας μεγάλης γειτονιάς μειώνει την πιθανότητα για τοπικό ελάχιστο από τη μία, αλλά από την άλλη απαιτεί αυξημένο κόστος υπολογισμού ανά επανάληψη. Ωστόσο, υπάρχει ένα σημαντικό πλεονέκτημα στις μεθόδους αυτές: ποτέ δεν επαναλαμβάνουν την ίδια λύση, έτσι ώστε λαμβάνοντας υπόψη το πεπερασμένο του συνόλου εφικτών λύσεων F , τερματίζουν πάντα με ένα τοπικό ελάχιστο.

Ένα εναλλακτικό είδος επιλογής γειτονικής λύσης και κριτηρίου λήξης που χρησιμοποιείται από τους *simulated annealing* και *tabu search*, είναι να επιτρέπεται η επόμενη λύση να έχει χειρότερο κόστος από την προηγούμενη, αλλά να παρέχονται μηχανισμοί που να εξασφαλίζουν με σημαντική πιθανότητα ότι στο μέλλον θα προκύψουν καλύτερες λύσεις. Το πλεονέκτημα της διατήρησης λύσεων με χειρότερο κόστος είναι ότι αποφεύγονται τοπικά ελάχιστα. Για παράδειγμα, αποδεικνύεται ότι η μέθοδος *simulated annealing* δεν είναι δυνατόν να καταλήξει σε τοπικό ελάχιστο. Δυστυχώς όμως, οι μέθοδοι αυτές διατρέχουν τον κίνδυνο να επαναλαμβάνουν ίδιες λύσεις. Είναι επομένως ουσιαστική σε αυτές τις μεθόδους η παροχή μηχανισμού που να αποκλείει, ή να καθιστά σχεδόν απίθανο το ενδεχόμενο επανάληψης ίδιων λύσεων.

1.2.2.1 Tabu Search

Ο Tabu Search στοχεύει στην αποφυγή τοπικών ελαχίστων, με την αποδοχή κατά περίπτωση μιας χειρότερης ή ακόμα και μη-εφικτής λύσης μέσα από την τρέχουσα γειτονιά. Δεδομένου ότι η βελτίωση κόστους δεν είναι δεδομένη, ο προδιαγεγραμμένος τρόπος αναζήτησης διατρέχει τον κίνδυνο επαναλήψεων, δηλαδή να επαναλαμβάνει την ίδια ακολουθία λύσεων κατά τρόπο αόριστο. Για να ξεπεράσει αυτό το πρόβλημα, ο Tabu Search παρακολουθεί τις

πρόσφατα αποκτηθείσες λύσεις καταχωρώντας αυτές σε έναν «απαγορευμένο» κατάλογο (ταμπού). Οι λύσεις στον κατάλογο ταμπού δεν επιτρέπεται να αναπαραχθούν, οπότε με αυτόν τον τρόπο αποφεύγεται η αναπαραγωγή ίδιων λύσεων, τουλάχιστον βραχυπρόθεσμα. Σε μια περίπλοκη παραλλαγή αυτής της στρατηγικής, ο κατάλογος ταμπού περιέχει ιδιότητες των πρόσφατων λύσεων, και όχι τις ίδιες τις λύσεις. Οι λύσεις με ιδιότητες αυτές του καταλόγου ταμπού θεωρούνται απαγορευμένες (εκτός από ιδιαίτερα ευνοϊκές περιπτώσεις, κάτω από τις οποίες ο κατάλογος ταμπού αγνοείται).

Διαφορετικές υλοποιήσεις του Tabu Search έχουν αναπτυχθεί μέσω της εμπειρίας από ένα μεγάλο αριθμό πρακτικών προβλημάτων. Ανάλογα με την υλοποίηση, ρυθμίζεται το μέγεθος της τρέχουσας γειτονιάς, το κριτήριο επιλογής μιας νέας λύσης από την τρέχουσα γειτονιά ή το κριτήριο λήξης. Επίσης, είναι δυνατό να χρησιμοποιηθούν επιλεκτική αποθήκευση των προηγούμενων λύσεων ή των ιδιοτήτων τους, ποινικοποίηση των περιορισμών με παραμέτρους ποινικής ρήτρας που μεταβάλλονται με το χρόνο, καθώς και πολλαπλοί κατάλογοι ταμπού [2].

❖ **Rochat [16]**

Οι Rochat και Taillard εισήγαγαν την έννοια της προσαρμοστικής μνήμης, μέσω της παρουσίασης μιας τεχνικής που βασίζεται σε πιθανότητες για να διαφοροποιήσουν, να ενισχύσουν και να χρησιμοποιήσουν την παράλληλη επεξεργασία σε μια μέθοδο τοπικής αναζήτησης που χρησιμοποιείται για την επίλυση των προβλημάτων δρομολόγησης οχημάτων. Η εν λόγω τεχνική μπορεί να εφαρμοστεί σε ένα ευρύ φάσμα μεθόδων τοπικής αναζήτησης αλλά χρησιμοποιείται συνήθως στην αναζήτηση ταμπού. Η προσαρμοστική μνήμη είναι ένα σύνολο καλών λύσεων που ενημερώνεται δυναμικά σε όλη τη διαδικασία αναζήτησης. Περιοδικά, μερικές διαδρομές αυτών των λύσεων εξάγονται από το εν λόγω σύνολο, με προτίμηση σε εκείνες τις διαδρομές που ανήκουν σε «καλές» λύσεις και στη συνέχεια συνδυάζονται με διάφορους τρόπους για να παραγάγουν μια νέα καλή λύση. Καθώς επιλέγονται οι διαδρομές που θα εξαχθούν, οι διαδρομές που περιλαμβάνουν πελάτες που ανήκουν σε διαδρομές που έχουν ήδη εξαχθεί, αγνοούνται. Δεδομένου ότι η λύση που κατασκευάζεται μέσω αυτής της επαναλαμβανόμενης διαδικασίας είναι πιθανό να μην περιέχει όλους τους πελάτες του προβλήματος, η μερική αυτή λύση πρέπει να ολοκληρωθεί χρησιμοποιώντας έναν ευριστικό δόμησης. Η προκύπτουσα εφικτή λύση θεωρείται ως αρχική λύση μιας μεθόδου τοπικής αναζήτησης που θα προσπαθήσει να βελτιώσει την ποιότητά της. Η νέα λύση που παράγεται εισάγεται στην προσαρμοστική μνήμη και ολοκληρωθεί η διαδικασία επαναλαμβάνεται έως ότου ικανοποιηθεί ένα κριτήριο λήξης.

❖ **Gendreau1 [17]**

Ο Gendreau περιγράφει τον αλγόριθμο Taburoute, μια νέα ευριστική μέθοδο αναζήτησης ταμπού, ειδικά για το πρόβλημα δρομολόγησης οχημάτων. Η διαδικασία της αναζήτησης

(procedure), κεντρική στον Taburoute, προσπαθεί να βελτιώσει μια δεδομένη λύση, βασιζόμενη στις αρχές της αναζήτησης ταμπού. Συγκεκριμένα, γειτονική μιας λύσης θεωρείται οποιαδήποτε λύση προκύπτει από την τρέχουσα με την αφαίρεση ενός κόμβου από μια διαδρομή και την παρεμβολή του σε μια άλλη που περιέχει έναν από τους πιο κοντινούς του κόμβους, χρησιμοποιώντας τη διαδικασία Geni, μια γενικευμένη διαδικασία εισαγωγής. Η Search εξετάζει επίσης και τις λύσεις που θεωρούνται μη-εφικτές με βάση του περιορισμούς μέγιστης χωρητικότητας ή μέγιστου μήκους διαδρομών, θεωρώντας μια αντικειμενική συνάρτηση με όρους ποινικής ρήτρας. Μερικές φορές, μια διαδικασία επαναβελτιστοποίησης, αποκαλούμενη US, χρησιμοποιείται για περαιτέρω βελτίωση της διαδρομής στην οποία παρεμβάλλεται ο κόμβος. Επιπλέον, όταν ένας κόμβος μετακινείται από τη διαδρομή r στη διαδρομή s , η επιστροφή του στη διαδρομή r θεωρείται απαγορευμένη για ορισμένο αριθμό επαναλήψεων. Τελικά, η Search προβλέπει και μια στρατηγική διαφοροποίησης, που επιβάλλει «ποινή» σε κόμβους που μετακινούνται συχνά, με σκοπό να εξερευνεί συνεχώς καινούριες λύσεις.

Μια γενική περιγραφή του Taburoute είναι η ακόλουθη: Αρχικά, διάφορες δοκιμαστικές αρχικές λύσεις παράγονται. Η Search εφαρμόζεται σε κάθε μια λύση για έναν περιορισμένο αριθμό επαναλήψεων, και η πιο πολλά υποσχόμενη λύση επιλέγεται ως αρχική λύση του Taburoute. Η Search καλείται έπειτα δύο φορές, με διαφορετικές παραμέτρους. Η πρώτη κλήση αποφέρει συνήθως τη σημαντικότερη βελτίωση στην αρχική λύση, ενώ η δεύτερη κλήση ενισχύει την τοπική αναζήτηση ώστε να επικεντρωθεί σε συγκεκριμένα υποσύνολα πόλεων που αντιστοιχούν στη καλύτερη γνωστή εφικτή λύση αν υπάρχει, ή στη καλύτερη γνωστή μη-εφικτή λύση ειδάλλως.

❖ Gendreau2 [18]

Το πρόβλημα δρομολόγησης ετερογενών οχημάτων (*Heterogeneous Vehicle Routing Problem - HVRP*) είναι μια παραλλαγή του κλασσικού προβλήματος δρομολόγησης οχημάτων, στο οποίο οι πελάτες εξυπηρετούνται από έναν ετερογενή στόλο οχημάτων με διαφορετικές χωρητικότητες, σταθερά και μεταβλητά κόστη. Οι συγγραφείς περιγράφουν έναν ευριστικό αλγόριθμο αναζήτησης ταμπού για το HVRP, ως συνδυασμό δύο ευριστικών: της διαδικασίας προσαρμοστικής μνήμης των Rochat και Taillard [16] και του Taburoute [17], οι οποίοι περιγράφηκαν παραπάνω. Πιο συγκεκριμένα, αυτός ο ευριστικός βασίζεται στη διαδικασία προσαρμοστικής μνήμης, με μέθοδο τοπικής αναζήτησης τον Taburoute.

❖ Barbarosoglu [19]

Ο ευριστικός αλγόριθμος που περιγράφεται αποσκοπεί στη λύση ενός προβλήματος δρομολόγησης οχημάτων που προέκυψε με αφορμή τις ανάγκες μιας συγκεκριμένης επιχείρησης διανομής. Αν και χρησιμοποιεί τις βασικές αρχές που χρησιμοποιούνται και στους προηγούμενους αλγορίθμους αναζήτησης ταμπού, αξίζει να αναφέρουμε τις διαφορές

που εισάγει. Καταρχήν, περιγράφει τη διαδικασία κατασκευής γειτονικών λύσεων που ονομάζεται *Sene*. Σύμφωνα με τη *Sene*, ένα σύνολο γειτονικών λύσεων παράγεται εκτελώντας αρχικά μια διαδικασία που αγνοεί τη σχετική συγκέντρωση των κόμβων και στη συνέχεια μια δεύτερη διαδικασία που λαμβάνει υπόψη τις σχετικές θέσεις των κόμβων των πελατών. Η διαδικασία 2-opt υιοθετείται για τοπική βελτίωση κάθε διαδρομής στην υπό εξέταση λύση. Κατόπιν επιλέγεται η εφικτή και μη-απαγορευμένη λύση με την καλύτερη τιμή αντικειμενικής συνάρτησης. Δεύτερον, η στρατηγική ενδυνάμωσης της λύσης (*intensification strategy*) υιοθετεί το τρόπο αναζήτησης *hill-climbing* και τρίτον, η ιδέα της διαφοροποίησης (*diversification*), η οποία θεωρείται τυποποιημένη σε όλες τις προηγούμενες προσεγγίσεις αναζήτησης ταμπού δεν συμπεριλαμβάνεται σε αυτόν τον ευριστικό, για λόγους υπολογιστικής αποταμίευσης.

❖ Xu [20]

Οι Xu και Kelly αναπτύσσουν μια νέα προσέγγιση τοπικής αναζήτησης βασισμένη σε ένα μοντέλο ροής δικτύων. Αυτό το μοντέλο χρησιμοποιείται για να καθορίσει τη δομή γειτονικών λύσεων με την παροχή ενός μηχανισμού βέλτιστης τοποθέτησης κόμβων σε διαδρομές. Τα κόστη εισαγωγής και αφαίρεσης κόμβων από δεδομένο δρομολόγιο υπολογίζονται προσεγγιστικά και λαμβάνοντας υπόψη τους περιορισμούς χωρητικότητας των οχημάτων. Επιπλέον, μια διαδικασία ανταλλαγής κόμβων χρησιμοποιείται που εξετάζει τις ανταλλαγές κόμβων μεταξύ δύο διαδρομών. Οι περιορισμοί χωρητικότητας χαλαρώνουν χρησιμοποιώντας όρους ποινικής ρήτρας. Η αναζήτηση ταμπού ενσωματώνεται στη διαδικασία αναζήτησης για να αποφύγει τα τοπικά ελάχιστα. Μια στρατηγική διαφοροποίησης (*diversification strategy*) εφαρμόζεται που καταγράφει τη συχνότητα με την οποία γίνεται ανάθεση των πελατών σε μια συγκεκριμένη διαδρομή και δίνει προτεραιότητα σε εκείνους τους πελάτες που εμφανίζονται λιγότερο συχνά σε αυτή. Αφ' ετέρου, μια στρατηγική ενδυνάμωσης (*intensification strategy*) εξετάζεται μέσω της οποίας μια ομάδα των καλύτερων λύσεων απομνημονεύεται και χρησιμοποιείται περιοδικά κάθε φορά που αρχικοποιείται η αναζήτηση.

❖ Kelly [21]

Οι Kelly και Xu πρότειναν έναν αλγόριθμο δυο-φάσεων για το πρόβλημα δρομολόγησης οχημάτων. Η πρώτη φάση του αλγορίθμου, η *φάση παραγωγής*, περιλαμβάνει την παραγωγή ενός ικανοποιητικού αριθμού ανεξάρτητων διαδρομών, χρησιμοποιώντας έναν απλό ευριστικό δόμησης. Η δεύτερη φάση, η *φάση ενοποίησης*, περιλαμβάνει τον εντοπισμό των «καλών» διαδρομών μέσα από το σύνολο των φυλαχθέντων, ώστε συνδυαστούν αυτές σε μια αρκετά βελτιωμένη λύση, χρησιμοποιώντας έναν ευριστικό αλγόριθμο χωρισμού συνόλων (*set-partitioning heuristic*). Για το πρόβλημα χωρισμού των συνόλων, χρησιμοποιείται αλγόριθμος αναζήτησης ταμπού. Η αρχική λύση που χρησιμοποιείται στην αναζήτηση

ταμπού είναι η καλύτερη εφικτή λύση που βρίσκεται στη φάση παραγωγής ενώ μια γειτονική λύση προσδιορίζεται με μια λειτουργία ανταλλαγής, που αντικαθιστά μια διαδρομή στην τρέχουσα λύση από μια νέα διαδρομή από το σύνολο των φυλαχθέντων. Αυτή η ανταλλαγή μπορεί να έχει ως αποτέλεσμα μερικοί πελάτες να εμφανίζονται στη λύση περισσότερες από μία φορές, και άλλοι πελάτες να μην εμφανίζονται καθόλου. Εντούτοις, αυτά τα δύο προβλήματα διορθώνονται με τη χρησιμοποίηση δύο επιδιορθωτικών διαδικασιών, οι οποίες είναι σε θέση να παραγάγουν νέες διαδρομές, οι οποίες συνδυάζουν τις καλές ιδιότητες των υπάρχουσών διαδρομών. Αυτές οι νέες διαδρομές προστίθενται στο σύνολο των φυλαχθέντων για την επόμενη επανάληψη. Ο αλγόριθμος ενσωματώνει τα στοιχειώδη χαρακτηριστικά γνωρίσματα της αναζήτησης ταμπού, όπως τη *βραχυπρόθεσμη μνήμη*, για να αποτρέψει τη διαδικασία αναζήτησης από την επανεμφάνιση λύσεων, καθώς επίσης και πολλές προηγμένες στρατηγικές, όπως τη *στρατηγική ταλάντωσης*, η οποία παρέχει τα μέσα για ρύθμιση της αλληλεπίδρασης μεταξύ ενδυνάμωσης (intensification) και διαφοροποίησης (diversification).

❖ Taillard [22]

Ο Taillard παρουσιάζει δύο μεθόδους διαμελισμού (partition methods), οι οποίες επιταχύνουν τις επαναληπτικές μεθόδους αναζήτησης που εφαρμόζονται στα προβλήματα δρομολόγησης οχημάτων, ακόμη και στη περίπτωση μεγάλου αριθμού οχημάτων. Η αναζήτηση ταμπού που χρησιμοποιείται ως η επαναληπτική μέθοδος αναζήτησης αρχίζει με μια αρχική λύση που αναθέτει ένα όχημα σε κάθε πελάτη. Η γειτονιά λύσεων καθορίζεται χρησιμοποιώντας το μηχανισμό παραγωγής που βασίζεται στη λ-ανταλλαγή. Οι εισαγωγές κόμβων σε δρομολόγια εκτελούνται χρησιμοποιώντας απλές ευριστικές μεθόδους εισαγωγής. Επιπλέον, τα ανεξάρτητα δρομολόγια βελτιστοποιούνται με τη χρησιμοποίηση του κώδικα των Volgenant και Jonker [23] που προσφέρεται για την ακριβή επίλυση του προβλήματος περιοδούντος πωλητή. Μια στρατηγική διαφοροποίησης χρησιμοποιείται που «τιμωρεί» τις κινήσεις μεταξύ λύσεων που εκτελούνται συχνά.

Η πρώτη μέθοδος διαμελισμού είναι κατάλληλη για τα ευκλείδεια προβλήματα, στα οποία οι κόμβοι-πελάτες διανέμονται ομοιόμορφα γύρω από ένα κεντρικό σταθμό. Σε τέτοια προβλήματα, υπο-προβλήματα λαμβάνονται διαμοιράζοντας αρχικά τους κόμβους σε τομείς που έχουν επίκεντρο τον κεντρικό σταθμό, και σε ομόκεντρες περιοχές μέσα σε κάθε τομέα. Κάθε υπο-πρόβλημα μπορεί να λυθεί ανεξάρτητα, με την εκτέλεση ενός δεδομένου αριθμού επαναλήψεων αναζήτησης ταμπού, αλλά οι περιοδικές κινήσεις κόμβων στους παρακείμενους τομείς είναι απαραίτητες. Η δεύτερη μέθοδος διαμελισμού είναι κατάλληλη για οποιοδήποτε πρόβλημα και είναι βασισμένη στη δένδροφυία που χτίζεται από τα συντομότερα μονοπάτια μεταξύ των κόμβων-πελατών και του κεντρικού σταθμού.

1.2.2.2 *Simulated Annealing*

Ο Simulated Annealing είναι παρόμοιος με την αναζήτηση ταμπού δεδομένου ότι επιτρέπει περιστασιακά να παράγονται λύσεις χειρότερου κόστους. Διαφέρει όμως στον τρόπο με τον οποίο αποφεύγει την επανεμφάνιση των ίδιων λύσεων. Αντί να ελέγχει ντετερμινιστικά τις προηγούμενες λύσεις για την αποφυγή ανακύκλωσης, επιλέγει τυχαία την επόμενη λύση. Με αυτό τον τρόπο, όχι μόνο αποφεύγει την ανακύκλωση λύσεων, αλλά και παρέχει κάποια θεωρητική εγγύηση για διαφυγή από τα τοπικά ελάχιστα και εύρεση καθολικού ελαχίστου. Ο Simulated Annealing τυχαιοποιεί την επιλογή της επόμενης λύσης μέσα από την τρέχουσα γειτονιά, με έναν τρόπο που δίνει προτεραιότητα στις λύσεις μικρότερου κόστους, και με αυτό τον τρόπο στοχεύει να βρει καθολικό ελάχιστο ταχύτερα από τις κοινές μεθόδους τυχαίας αναζήτησης.

Ειδικότερα, δεδομένης μιας λύσης χ , επιλέγουμε με τυχαία δειγματοληψία την επόμενη λύση $\bar{\chi}$ από τη γειτονιά $N(\chi)$. Οι πιθανότητες δειγματοληψίας είναι θετικές για όλα τα μέλη της $N(\chi)$, ειδικά είναι απροσδιόριστες. Η λύση $\bar{\chi}$ γίνεται αποδεκτή, εάν έχει μικρότερο κόστος, δηλαδή εάν $f(\bar{\chi}) < f(\chi)$. Διαφορετικά, γίνεται αποδεκτή με πιθανότητα $e^{-(f(\bar{\chi})-f(\chi))/T}$, όπου T είναι κάποια θετική σταθερά, και απορρίπτεται με τη συμπληρωματική πιθανότητα.

Η σταθερά T ρυθμίζει την πιθανότητα αποδοχής λύσεων χειρότερου κόστους, και αποκαλείται *θερμοκρασία* της διαδικασίας. Η πιθανότητα αποδοχής μιας λύσης $\bar{\chi}$ χειρότερου κόστους από τη χ μειώνεται καθώς το κόστος της αυξάνεται. Επιπλέον, όταν η σταθερά T είναι μεγάλη (ή μικρή), η πιθανότητα αποδοχής μιας χειρότερης λύσης είναι κοντά στο 1 (ή κοντά στο 0, αντίστοιχα). Στην πράξη, είναι χαρακτηριστικό στην αρχή να χρησιμοποιείται μεγάλο T , επιτρέποντας με καλύτερη πιθανότητα την αποφυγή τοπικών ελαχίστων, και στη συνέχεια μικρότερο, ώστε να ενισχύεται βαθμιαία η επιλεκτικότητα της μεθόδου προς τις βελτιωμένες λύσεις.

Σε αντίθεση με την αναζήτηση ταμπού, η οποία δεν προσφέρει καμία γενική θεωρητική εγγύηση καλής επίδοσης, ο Simulated Annealing βασίζεται σε μια ισχυρή θεωρητική βάση. Ειδικότερα, κάτω από αρκετά γενικούς όρους, μπορεί να αποδειχθεί ότι ένα καθολικό ελάχιστο τελικά θα εντοπιστεί, και ότι με βαθμιαία μείωση της θερμοκρασίας T , η διαδικασία αναζήτησης θα επικεντρωθεί με μεγάλη πιθανότητα σε λύσεις που είναι καθολικά βέλτιστες.

Η ταχύτητα σύγκλισης είναι τόσο σημαντική όσο και το ενδεχόμενο σύγκλισης στο βέλτιστο, και η επίλυση ενός δεδομένου προβλήματος με τον Simulated Annealing μπορεί να αποδειχθεί πολύ αργή. Μια αξιολογή όμως πτυχή της μεθόδου είναι ότι εξαρτάται ελάχιστα από τη δομή του προβλήματος προς επίλυση, και αυτό ενισχύει την αξία του για τα σχετικά μη δομημένα προβλήματα που δεν γίνονται αρκετά κατανοητά. Για άλλα προβλήματα, όπου

υπάρχει συσσωρευμένη γνώση και εμπειρία, ο Simulated Annealing θεωρείται συνήθως κατώτερος από τις άλλες προσεγγίσεις τοπικής αναζήτησης [2].

❖ **Breedam [24]**

Ο Breedam μελετά τη χρήση μεθόδων βελτίωσης που βασίζονται στον *Simulated Annealing* για το πρόβλημα δρομολόγησης οχημάτων. Οι μέθοδοι βελτίωσης που εξετάζονται είναι οι ακόλουθες: κατ' αρχάς, η μέθοδος επανατοποθέτησης σειράς, η οποία προσπαθεί να παρεμβάλει ένα κόμβο ή μια σειρά κόμβων από μια διαδρομή σε μια άλλη. Δεύτερον, η μέθοδος ανταλλαγής σειράς, η οποία πραγματοποιεί ανταλλαγή κόμβων ή σειράς κόμβων μεταξύ δύο διαδρομών. Τρίτον, η μικτή μέθοδος, η οποία αποτελεί σύνθεση των προηγούμενων δύο μεθόδων. Η συνήθης υλοποίηση των παραπάνω μεθόδων βελτίωσης έχει το τυπικό χαρακτηριστικό ότι μόνο οι κινήσεις που οδηγούν σε καλύτερη λύση από τη τρέχουσα γίνονται αποδεκτές. Μια κίνηση είναι μια μετάβαση από μια εφικτή λύση σε μια άλλη, με τη βοήθεια μιας επανατοποθέτησης ή ανταλλαγής κόμβων. Αντιθέτως, η υλοποίηση των μεθόδων βελτίωσης με βάση τον Simulated Annealing κάνουν αποδεκτές με ορισμένη πιθανότητα και εφικτές λύσεις που προκαλούν χειροτέρευση της αντικειμενικής συνάρτησης, αποτρέποντας κατά συνέπεια την παγίδευση σε ένα τοπικό ελάχιστο. Τα πειραματικά αποτελέσματα έδειξαν ότι οι μέθοδοι βελτίωσης που υλοποιούνται με βάση τον Simulated Annealing δίνουν αρκετά καλύτερα αποτελέσματα. Εντούτοις, αξίζει να σημειωθεί ότι γενικά, ο υπερβολικά υψηλός χρόνος επεξεργασίας παραμένει ένα σημαντικό μειονέκτημα των μεθόδων Simulated Annealing.

1.2.2.3 Γενετικοί Αλγόριθμοι

Οι Γενετικοί Αλγόριθμοι είναι μέθοδοι τοπικής αναζήτησης, όπου ο μηχανισμός παραγωγής γειτονικών λύσεων είναι εμπνευσμένος από τις πραγματικές διαδικασίες της γενετικής και της εξέλιξης. Ειδικότερα, η τρέχουσα λύση τροποποιείται με τις μεθόδους «συνένωση» και «μεταλλαγή», ώστε να εξασφαλιστούν οι γειτονικές λύσεις. Ένα χαρακτηριστικό παράδειγμα προέρχεται από τα προβλήματα καθορισμού μιας χρονικής διάταξης, όπως το πρόβλημα περιοδεύοντος πωλητή. Η γειτονιά μιας χρονικής διάταξης T μπορεί να είναι μια συλλογή άλλων διατάξεων που λαμβάνονται κατόπιν τροποποίησης κάποιου παρακείμενου τμήματος του T με κάποιο τρόπο, κρατώντας το υπόλοιπό του άθικτο. Εναλλακτικά, η γειτονιά μιας διάταξης μπορεί να ληφθεί με την ανταλλαγή της θέσης μερικών διεργασιών, όπως συμβαίνει στον K -OPT ευριστικό περιοδεύοντος πωλητή.

Σε μια παραλλαγή αυτής της προσέγγισης, μια ομάδα λύσεων μπορεί να διατηρείται, μερικές από τις οποίες είναι δυνατόν να τροποποιηθούν, ενώ μερικά ζευγάρια αυτών των λύσεων μπορούν να συνδυαστούν για να δημιουργήσουν νέες λύσεις. Αυτές οι λύσεις προστίθενται στην ομάδα εάν ικανοποιείται κάποιο κριτήριο, το οποίο συνήθως βασίζεται

στη βελτίωση κόστους, ενώ μερικές από τις λύσεις της υπάρχουσας ομάδας μπορούν να απομακρυνθούν. Κατ' αυτό τον τρόπο, θεωρείται ότι η ομάδα εξελίσσεται με έναν τρόπο *Darwinian* μέσω της «επιβίωσης της καταλληλότερης διαδικασίας».

Η εφαρμογή αυτής της προσέγγισης λειτουργεί σε φάσεις. Στην αρχή μιας φάσης, έχουμε έναν πληθυσμό X που αποτελείται από n εφικτές λύσεις x_1, \dots, x_n . Η φάση εξελίσσεται ως εξής:

Βήμα 1: (Τοπική αναζήτηση) Για κάθε λύση x_i του τρέχοντος πληθυσμού X , εφαρμόζεται ένας τοπικός αλγόριθμος αναζήτησης μέχρι να ληφθεί ένα τοπικό ελάχιστο \bar{x}_i . Έστω ότι προκύπτει ο νέος πληθυσμός $\bar{X} = \{\bar{x}_1, \dots, \bar{x}_n\}$.

Βήμα 2: (Μεταλλαγή) Επιλέγεται τυχαία ένα υποσύνολο των στοιχείων του \bar{X} , και κάθε στοιχείο του τροποποιείται σύμφωνα με κάποιο μηχανισμό (που εξαρτάται από το πρόβλημα), ώστε να ληφθεί μια άλλη εφικτή λύση.

Βήμα 3: (Επανασυνδυασμός) Επιλέγεται τυχαία ένα υποσύνολο ζευγαριών στοιχείων του \bar{X} , και από κάθε ζευγάρι παράγεται μια εφικτή λύση σύμφωνα με κάποιο μηχανισμό (που εξαρτάται από το πρόβλημα).

Βήμα 4: (Επιλογή επιζώντων) Έστω \tilde{X} το σύνολο εφικτών λύσεων που προέκυψαν από τα βήματα 3 και 4. Από τον πληθυσμό $\bar{X} \cup \tilde{X}$, επιλέγεται ένα υποσύνολο n στοιχείων σύμφωνα με κάποιο κριτήριο. Αυτό το υποσύνολο χρησιμοποιείται για την έναρξη της επόμενης φάσης.

Η μεταλλαγή επιτρέπει προσιτές θεωρητικές παραλλαγές των τοπικών ελάχιστων, ενώ ο επανασυνδυασμός (επίσης γνωστός ως *διασταύρωση*) στοχεύει να συνδυάσει τις ιδιότητες ενός ζευγαριού τοπικών ελάχιστων. Οι διαδικασίες της μεταλλαγής και του επανασυνδυασμού πραγματοποιούνται συνήθως με τη βοήθεια κάποιας δομής δεδομένων που χρησιμοποιείται για την αναπαράσταση μιας λύσης. Υπάρχει ένας μεγάλος αριθμός παραλλαγών των γενετικών αλγορίθμων. Στην ουσία, η κάθε παραλλαγή εξαρτάται από το εκάστοτε πρόβλημα και απαιτεί εμπειρικές μεθόδους δοκιμής και σφάλματος. Εντούτοις, οι γενετικοί αλγόριθμοι είναι αρκετά εύκολο να υλοποιηθούν, και για το λόγο αυτό είναι ιδιαίτερα δημοφιλείς [2].

❖ Ombuki [25]

Οι συγγραφείς παρουσιάζουν μια μέθοδο υβριδικής αναζήτησης βασισμένη σε μεταευσριστικούς για την κατά προσέγγιση επίλυση του *προβλήματος δρομολόγησης οχημάτων με χρονικούς περιορισμούς (Vehicle Routing Problem Time Windows - VRPTW)*. Η σημαντικότερη συμβολή τους είναι η πρόταση μιας υβριδικής τεχνικής κατάλληλης για βελτιστοποίηση πολλαπλών στόχων σε προβλήματα VRPTW. Σκοπός του προβλήματος που μελετάται είναι η ελαχιστοποίηση του αριθμού οχημάτων που χρησιμοποιούνται και της απόστασης που διανύεται ώστε να ικανοποιηθούν τα αιτήματα όλων των πελατών χωρίς να

υπερβαίνεται η μέγιστη χωρητικότητα των οχημάτων και ο μέγιστος χρόνος για την εξυπηρέτηση κάθε πελάτη. Από αυτή την άποψη, το πρόβλημα μπορεί να αντιμετωπιστεί ως πρόβλημα βελτιστοποίησης πολλαπλών και αντικρουόμενων στόχων. Προκειμένου να μειωθεί η απόσταση που διανύεται, θα πρέπει να αυξηθεί ο αριθμός των οχημάτων. Αφ' ετέρου, για να μειωθεί το κόστος από τη χρήση πολλών οχημάτων, πρέπει να αυξηθεί η απόσταση που διανύεται από κάθε όχημα. Τα αποτελέσματα υπολογισμών έδειξαν ότι ο γενετικός αλγόριθμος (GA) είναι αποτελεσματικός στον καθορισμό του αριθμού οχημάτων που χρησιμοποιούνται, ενώ η αναζήτηση ταμπού (TS) είναι αποτελεσματικότερη στη μείωση της συνολικής απόστασης που διανύεται από τα οχήματα. Κατά συνέπεια, η προτεινόμενη προσέγγιση εξελίσσεται σε δυο φάσεις: στη φάση ομαδοποίησης πελατών, όπου η κάθε ομάδα εξυπηρετείται από διαφορετικό όχημα, που βασίζεται σε GA και στη φάση όπου πραγματοποιείται μια τοπική τεχνική αναζήτησης για περαιτέρω βελτιστοποίηση που βασίζεται στον TS.

❖ **Pedroso [26]**

Το πρόβλημα που αντιμετωπίζεται είναι γνωστό ως το *Πρόβλημα Εφημεριδοπώλη του Μανχάταν*: υποθέτουμε ότι υπάρχουν μια κεντρική αποθήκη εφημερίδων σε μια συγκεκριμένη θέση σε μια πόλη, ένα σύνολο διανομέων, και ένα σύνολο κόμβων που αντιστοιχούν στους συνδρομητές όπου οι εφημερίδες πρέπει να παραδοθούν. Ο στόχος είναι να διανεμηθεί μια εφημερίδα σε κάθε έναν από τους συνδρομητές, και να ελαχιστοποιηθεί ο χρόνος παράδοσης στον συνδρομητή που εξυπηρετείται τελευταίος. Οι συγγραφείς προτείνουν την εφαρμογή μιας υβριδικής στρατηγικής γι' αυτό το συνδυαστικό πρόβλημα βελτιστοποίησης που προκύπτει από το συνδυασμό μιας μεθόδου τοπικής αναζήτησης με έναν εξελικτικό αλγόριθμο. Η μέθοδος τοπικής αναζήτησης περιλαμβάνει την αναζήτηση σε διάφορους τύπους γειτονιών, η οποία εκτελείται επαναληπτικά έως ότου δεν υπάρχει καμία περαιτέρω βελτίωση. Αφ' ετέρου, η μέθοδος αναζήτησης Niche είναι ένας εξελικτικός αλγόριθμος όπου ο συνολικός πληθυσμός λύσεων ομαδοποιείται σε υποπληθυσμούς (niches), οι οποίοι εξελίσσονται ανεξάρτητα. Η αξίωση είναι ότι με αυτό τον τρόπο, πολλές τοπικές αναζητήσεις γίνονται συγχρόνως, μέσα σε κάθε υποπληθυσμό, ενώ οι υποπληθυσμοί υπόκεινται επίσης σε μεταξύ τους ανταγωνισμό. Από αυτή την άποψη, επιτυγχάνεται ένας καλός συμβιβασμός μεταξύ της ενδυνάμωσης (intensification) της τοπικής αναζήτησης και διαφοροποίησης (diversification) του πληθυσμού.

1.3 ΒΙΒΛΙΟΓΡΑΦΙΑ-ΑΝΑΦΟΡΕΣ

- [1] T. Dumcan, “Experiments in the Use of Neighbourhood Search Techniques for Vehicle Routing”, Report AIAI-TR176, University of Edinburgh, June 1995.
- [2] D.P. Bertsekas, “Network Optimization: Continuous and Discrete Models”, Publisher: Athena Scientific, ISBN: 1-886529-02-7, 1998.
- [3] M.L. Fisher and R. Jaikumur, “A Generalized Assignment Heuristic for Vehicle Routing”, Networks Vol.11, No.2, pp.109-124, 1981.
- [4] J. Lenstra and A. Kan, “Complexity of Vehicle Routing and Scheduling Problems”, Networks, Vol.11, No.2, pp.221-228, 1981.
- [5] G. Laporte, M. Gendreau, J-Y. Potvin, F. Semet, “Classical and Modern Heuristics for the Vehicle Routing Problem”, International Transaction in Operational Research, Vol.7, pp.285-300, 2000.
- [6] S. Lin, “Computer Solutions of the Traveling Salesman Problem”, Bell System Technical Journal, Vol.44, pp.2245-2269, 1965.
- [7] G. Clarke, J. Wright, “Scheduling of Vehicles from a Central Depot to a Number of Delivery Points”, Operations Research, Vol.12, pp.568-581, 1964.
- [8] K. Altinkemer and B. Gavish, “Parallel Savings Based Heuristics for the Delivery Problem”, Operations Research, Vol.39, No.3, pp.456-469, 1991.
- [9] P. Wark and J. Holt, “A Repeated Matching Heuristic for the Vehicle Routing Problem”, Journal of the Operational Research Society, Vol.45, No.10, pp.1156-1167, 1994.
- [10] R.H. Mole and S.R. Jameson, “A Sequential Route-building Algorithm Employing a Generalized Savings Criterion”, Operational Research Quarterly, Vol.27, No.2, pp.503-511, 1976.
- [11] M.L. Fisher and R. Jaikumur, “A Generalized Assignment Heuristic for Vehicle Routing”, Networks 11, pp.109-124, 1981.
- [12] B.M. Baker, and J. Sheasby, “Extensions to the Generalized Assignment Heuristic for Vehicle Routing”, European Journal of Operational Research 119, pp.147-157, 1999.
- [13] J. Bramel and D. Simchi-Levi, “A Location based Heuristic for General Routing Problems”, Operations Research, Vol.43, pp.649-660, 1995.
- [14] B. Gillett and L. Miller, “A Heuristic Algorithm for the Vehicle Dispatch Problem”, Operations Research, Vol.22, pp.340-349, 1974.
- [15] M. Gendreau, G. Laporte, J. Potvin, “Metaheuristics for the Vehicle Routing Problem”, Technical Report CRT-963, University of Montreal, September 1998.

- [16] Y. Rochat and E. Taillard, "Probabilistic diversification and intensification in local search for vehicle routing", *Journal of Heuristics*, Vol.1, No.1, pp.147-167, 1995.
- [17] M. Gendreau, A. Hertz, G. Laporte, "A Tabu Search Heuristic for the Vehicle Routing Problem", *Management Science*, Vol.40, No.10, pp.1276-1290, 1994.
- [18] M. Gendreau, G. Laporte, C. Musaraganyi, I.D. Taillard, "A Tabu Search Heuristic for the Heterogeneous Fleet Vehicle Routing Problem", *Computers & Operations Research* 26, pp.1153-1173, 1999.
- [19] G. Barbarosoglu and D. Ozgur, "A Tabu Search Algorithm for the Vehicle Routing Problem", *Computers & Operations Research* 26, pp.255-270, 1999.
- [20] J. Xu and J. Kelly, "A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem", *Transportation Science* 30, pp.379-393, 1996.
- [21] J.P. Kelly and J. Xu, "A Set-Partitioning-Based Heuristic for the Vehicle Routing Problem", *INFORMS Journal on Computing*, Vol.11, No.2, pp.161-172, 1999.
- [22] I.D. Taillard, "Parallel Iterative Search Methods for Vehicle Routing Problems", *Networks* 23, pp.661- 673, 1993.
- [23] A. Volgenant and R. Jonker, "The Symmetric Traveling Salesman Problem and Edge Exchanges in Minimal 1-tree", *European Journal of Operational Research*, Vol.12, pp.394-403, 1983.
- [24] A.V. Breedam, "Improvement Heuristics for the Vehicle Routing Problem based on Simulated Annealing", *European Journal of Operational Research* 86, pp.480-490, 1995.
- [25] B.M. Ombuki, M. Nakamura and M. Osamu, "A Hybrid Search Based on Genetic Algorithms and Tabu Search for Vehicle Routing", Technical Report # CS-02-07, May 2002.
- [26] J.P. Pedroso, "Niche Search: An Application in Vehicle Routing", *Proceedings of the IEEE International Conference on Evolutionary Computation*, Vol.1, Anchorage, Alaska. 1998.