

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ
ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ ΚΑΙ ΔΙΚΤΥΩΝ



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
Κατανεμημένες δομές δεδομένων σε ασύρματη επικοινωνία

Νάρου Αναστασία

Επιβλέποντες: Παναγιώτης Μποζάνης, Επίκουρος Καθηγητής
Δημήτριος Κατσαρός, Συμβασιούχος ΠΔ407/80

ΒΟΛΟΣ
ΜΑΡΤΙΟΣ 2009

Περιεχόμενα

ΜΕΡΟΣ I Κινητά και διάχυτα δεδομένα (Mobile and Pervasive Data)

ΚΕΦΑΛΑΙΟ 1

Τεχνικές υπογραφής και αποθήκευσης υπογραφών για φιλτράρισμα πληροφοριών σε ασύρματο και κινητό περιβάλλον

1.1 Εισαγωγή.....	10
1.2 Η Τεχνική της Υπογραφής	12
1.3 Μετάδοση πληροφορίας με την τεχνική της υπογραφής	13
1.4 Ανάλυση Απόδοσης	19
1.5 Αποθήκευση των Υπογραφών	31

ΚΕΦΑΛΑΙΟ 2

Τεχνικές εκπομπής πληροφορίας με πολλαπλά γνωρίσματα

2.1 Εισαγωγή.....	35
2.2 Παράμετροι για τις τεχνικές εκπομπής πληροφορίας με πολλά γνωρίσματα.....	36
2.3 Ανάλυση Απόδοσης	48

ΜΕΡΟΣ II Ομότιμα Δίκτυα (Peer-To-Peer (P2P) Networks)

ΚΕΦΑΛΑΙΟ 3

Υπογραφές γειτνίασης για αναζήτηση σε ομότιμα (P2P) δίκτυα

3.1 Εισαγωγή.....	56
3.2 Ομότιμο (P2P) σύστημα.....	57
3.3 Αναζήτηση σε (P2P) συστήματα.....	58
3.4 Υπογραφές Γειτόνων (Neighborhood signatures).....	59
3.5 Αλγόριθμοι Αναζήτησης.....	61
3.6 Κατασκευή και Διατήρηση Υπογραφών.....	63
3.7 Ανάλυση Απόδοσης.....	65

ΚΕΦΑΛΑΙΟ 4

Υποστήριξη πολύπλοκων πολυδιάστατων ερωτήσεων σε ομότιμα (P2P) δίκτυα

4.1 Εισαγωγή.....	70
4.2 Μέθοδος R*-tree για επεξεργασία ερωτήσεων	70
4.3 Μοντέλο Συστήματος	72
4.4 Διαμόρφωση και Διατήρηση Συστάδας.....	73
4.5 Κατασκευή και Διατήρηση Ευρετηρίου (Index).....	74
4.6 Αλγόριθμοι Επεξεργασίας Ερωτήσεων (Query Processing).....	77
4.7 Ανάλυση Απόδοσης	83

ΒΙΒΛΙΟΓΡΑΦΙΑ

86

Πρόλογος

Η παρούσα διπλωματική εργασία είναι μια μελέτη πάνω σε κατανεμημένες δομές δεδομένων σε ασύρματη επικοινωνία. Οι αλγόριθμοι και οι δομές παρουσιάστηκαν από θεωρητικής και πρακτικής απόψεως. Δόθηκε κυρίως έμφαση σε δομές δεδομένων σε κινητά και διάχυτα δεδομένα (mobile and pervasive data) και ομότιμα δίκτυα (p2p networks).

Οι αλγόριθμοι που παρουσιάζονται στη συνέχεια επεξηγούνται μέσω αναλυτικής περιγραφής, χαρακτηριστικών παραδειγμάτων, ψευτοκώδικα και γραφικών παραστάσεων. Σε κάποιες περιπτώσεις δε αναλύεται και η πολυπλοκότητα αυτών και διάφοροι ενδιαφέροντες χρόνοι της απόδοσης των αλγορίθμων.

Στο πρώτο κεφάλαιο παρουσιάζονται τρεις τεχνικές υπογραφών (Μέθοδος Απλής Υπογραφής (Simple Signature), Μέθοδος Ενοποιημένης Υπογραφής (Integrated Signature), Μέθοδος Πολυεπίπεδης Υπογραφής (Multi-level Signature) που εξυπηρετούν στο φιλτράρισμα της πληροφορίας και άρα στην καλύτερη επικοινωνία στο δίκτυο από άποψη χρόνου και χώρου. Επίσης, παρουσιάζονται και τέσσερις μέθοδοι αποθήκευσης των υπογραφών στην προσωρινή μνήμη.

Στο δεύτερο κεφάλαιο έχουν αναλυθεί τρεις αλγόριθμοι και οι αντίστοιχες αποδόσεις τους που χρησιμοποιούνται στην εκπομπή πληροφοριών με πολλά γνωρίσματα. Οι αλγόριθμοι είναι οι εξής : το Δέντρο Ευρετηρίου (Index Tree), η τεχνική της υπογραφής (Signature Scheme), η υβριδική τεχνική (Hybrid Method).

Στο τρίτο κεφάλαιο εξετάζονται τρεις τεχνικές υπογραφών γειτνίασης για αναζήτηση πληροφοριών σε ομότιμα δίκτυα και δύο αλγόριθμοι αναζήτησης πληροφοριών. Οι τρεις τεχνικές είναι: Ολοκληρωμένη –γειτονική υπογραφή (complete-neighborhood signature (CN), μερική-γειτονική υπογραφή (partial-neighborhood signature (PN), μερική-γειτονική υπογραφή με υπέρθεση (partial-neighborhood superimposed signature (PN-S), μερική-γειτονική υπογραφή με προσάρτηση (partial- neighborhood appended signature (PN-A). Οι αλγόριθμοι είναι: Αναζήτηση Πλημμύρας (Flooding Search), Αναζήτηση Μοναδικού Μονοπατιού (Single Path Search).

Στο τέταρτο και τελευταίο κεφάλαιο παρουσιάζονται οι δομές και οι αλγόριθμοι που μπορούν να υποστηρίξουν πολύπλοκες πολυδιάστατες ερωτήσεις (queries) σε ομότιμα δίκτυα. Αναλύεται η κατασκευή του NR-Tree ευρετηρίου και η σημασία του για ένα ομότιμο δίκτυο και οι αλγόριθμοι επεξεργασίας ερωτήσεων αλληλουχίας (Range query processing) και επεξεργασίας ερωτήσεων k-κοντινότερου γείτονα (k-nearest neighbor query processing).

ΜΕΡΟΣ Ι

Κινητά και διάχυτα δεδομένα (Mobile and Pervasive Data)

ΜΕΡΟΣ I

Κινητά και διάχυτα δεδομένα (Mobile and Pervasive Data)

Εισαγωγικά.....	9
-----------------	---

ΚΕΦΑΛΑΙΟ 1

Τεχνικές υπογραφής και αποθήκευσης υπογραφών για φιλτράρισμα πληροφοριών σε ασύρματο και κινητό περιβάλλον

Περιεχόμενα

1.1 Εισαγωγή.....	10
1.2 Η Τεχνική της Υπογραφής.....	12
1.3 Μετάδοση πληροφορίας με την τεχνική της υπογραφής.....	13
1.3.1 Μέθοδος Απλής Υπογραφής (Simple Signature).....	14
1.3.2 Μέθοδος Ενοποιημένης Υπογραφής (Integrated Signature).....	15
1.3.3 Μέθοδος Πολυεπίπεδης Υπογραφής (Multi-level Signature).....	17
1.4 Ανάλυση Απόδοσης.....	19
1.4.1 Σύμβολα και Παράμετροι.....	19
1.4.2 Πιθανότητα Λανθασμένου Ταιριάσματος (False Drop Probability).....	20
1.4.3 Μοντέλα Κόστους (Cost Models).....	21
1.4.3.1 Κόστος για την Μέθοδο Απλής Υπογραφής.....	22
1.4.3.2 Κόστος για την Μέθοδο Ενοποιημένης Υπογραφής.....	23
1.4.3.3 Κόστος για την Μέθοδο Πολυεπίπεδης Υπογραφής.....	24
1.4.3.4 Συγκρίσεις.....	25
1.4.4 Βελτιστοποίηση Υπογραφών.....	30
1.5 Αποθήκευση των Υπογραφών.....	31
1.5.1 Bit Tags and Active Refresh (BA).....	32
1.5.2 Version Numbers and Active Refresh (VA).....	32
1.5.3 Bit Tags and Passive Refresh (BP).....	32
1.5.4 Version Numbers and Passive Refresh (VP).....	33

ΜΕΡΟΣ I Κινητά και διάχυτα δεδομένα(Mobile and Pervasive Data)

Τα τελευταία χρόνια η ραγδαία εξέλιξη των ασύρματων δικτύων δεδομένων έχουν ανοίξει νέους ορίζοντες για τους χρήστες. Ποικίλα εμπορικά και πειραματικά **προσωπικού τύπου υπολογιστικά συστήματα (Personal Computing System PCSs)** έχουν κατακλύσει την αγορά και προβλέπεται πως θα γίνουν τόσο ευρέως διαδεδομένα, όπως παλαιότερα ήταν τα κινητά κασετόφωνα (walkmans) και οι κινητές τηλεοράσεις. Μέσω της ασύρματης επικοινωνίας οι κινητοί χρήστες με τα PCSs τους θα μπορούν να επικοινωνούν με άλλους χρήστες σε άλλες πόλεις, χώρες και ηπείρους χωρίς κανένα περιορισμό. Στην δική μας μελέτη θεωρούμε PCSs με μπαταρία.

Οι εφαρμογές για ασύρματη μετάδοση πληροφορίας είναι αναρίθμητες. Ένα πολύ χαρακτηριστικό παράδειγμα είναι οι αγορές που μπορούν να γίνουν μέσω διαδικτύου. Εκεί ο εκάστοτε χρήστης μπορεί να περιηγηθεί σε σχέδια ρούχων, συσκευών, βιβλίων και των αντίστοιχων τιμών και με το πάτημα μερικών κουμπιών στο PCS του να έχει το αγαθό που επιθυμεί στην πόρτα του σπιτιού του. Βέβαια αυτό μπορεί και να επεκταθεί και σε ολόκληρα εμπορικά κέντρα όπου πληροφορίες όπως κατηγορίες προϊόντων, τοποθεσίες καταστημάτων ακόμα και ειδικές εκπτώσεις μπορούν να γίνουν γνωστές στους χρήστες μέσω του αέρα(ασύρματη επικοινωνία).

Μπορούμε να θεωρήσουμε δύο καταστάσεις λειτουργίας στις εφαρμογές μετάδοσης πληροφορίας. Για τα PCSs με δυνατότητα μετάδοσης, τα PCSs στέλνουν στον κεντρικό **εξυπηρετητή (server)** τις ερωτήσεις τους, ο οποίος στη συνέχεια μαζεύει τις ερωτήσεις για ένα χρονικό διάστημα και έπειτα μεταδίδει την επιθυμητή πληροφορία στο κανάλι. Για τα PCSs που δεν έχουν δυνατότητα μετάδοσης, ο server μεταδίδει όλες τις διαθέσιμες πληροφορίες στο κανάλι με μια συγκεκριμένη σειρά, τα PCSs "ακούν" το κανάλι και επιλέγουν τις πληροφορίες που είναι χρήσιμες για τους χρήστες. Θεωρείτε δεδομένο δε ότι σε κάθε περίπτωση η πληροφορία από τον server μεταδίδεται μέσω ενός μόνο καναλιού. Σε άλλη περίπτωση αν έπρεπε να έχουμε ένα κανάλι για κάθε PCS χρήστη αντιλαμβανόμαστε πως το κόστος θα ήταν πολύ μεγάλο από τη μια και τα PCSs θα έπρεπε να φιλτράρουν τις πληροφορίες με αποτέλεσμα να τελειώνει γρήγορα η μπαταρία .

Η αποτελεσματική πρόσβαση σε δεδομένα στην ασύρματη επικοινωνία με κινητούς χρήστες είναι ένα ζήτημα που αποκτά ολοένα και περισσότερο ενδιαφέρον. Στο **Μέρος I** της μελέτης θα εξετάσουμε τεχνικές ευρετηρίου(indexing)και αποθήκευσης(caching) ώστε να μειωθεί η κατανάλωση ενέργειας από τα PCSs, βελτίωση της πρόσβασης σε δεδομένα, εντοπισμός δεδομένων για πολλαπλά ασύρματα κανάλια επικοινωνίας, ακύρωση μνήμης και πολιτική αντικατάστασης για ασύρματη διάδοση δεδομένων, απόδοση και αξιολόγηση για ασύρματη πρόσβαση σε δεδομένα.

ΚΕΦΑΛΑΙΟ 1 Τεχνικές υπογραφής και αποθήκευσης υπογραφών για φιλτράρισμα πληροφοριών σε ασύρματο και κινητό περιβάλλον

1.1 Εισαγωγή

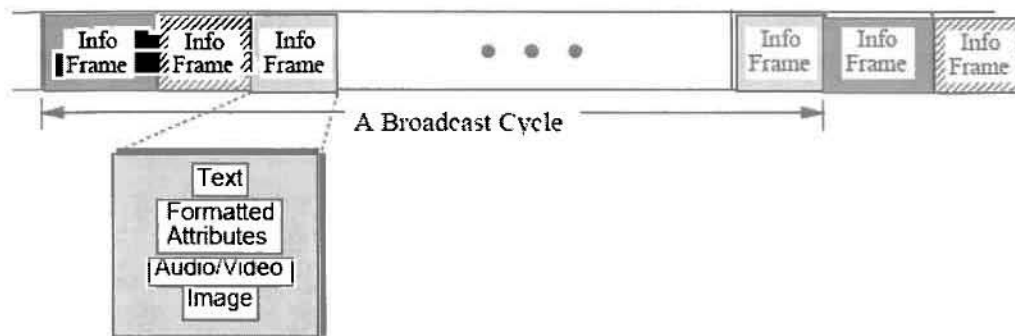
Ένα μεγάλο πρόβλημα που δημιουργείτε με τα PCSs είναι η παροχή ενέργειας τους, δηλαδή η μπαταρία τους. Εφόσον τα περισσότερα PCSs χρησιμοποιούν μπαταρίες, για να γίνουν φορητά μπορούν να χρησιμοποιηθούν μικρές μπαταρίες όπως AA ή AAA. Παρόλα αυτά αυτές οι μπαταρίες χρειάζονται επαναφόρτιση ή αντικατάσταση μετά από κάποιο χρονικό διάστημα.

Υπάρχουν δύο παράγοντες που μπορούν να επηρεάσουν την κατανάλωση ενέργειας στα PCSs : (1) τα PCSs μπορούν να αλλάξουν από **ενεργή κατάσταση (active mode)** σε **κατάσταση "ύπνου" (doze mode)** και (2) η παραλαβή μηνυμάτων καταναλώνει πολύ λιγότερη ενέργεια από την αποστολή τους. Υπάρχουν διάφορες μέθοδοι που έχουν προταθεί για την εξοικονόμηση ενέργειας και για την μετάδοση δεδομένων.

Στην μετάδοση δεδομένων, ο κεντρικός σταθμός στέλνει μια σειρά από **πλαίσια πληροφοριών (information frames)** (βλέπε Σχήμα 1). Ένα **πλαίσιο (frame) δεδομένων** είναι μια λογική μονάδα πληροφορίας που μεταδίδεται στον αέρα και μπορεί να είναι **οπτικοακουστικού τύπου (multimedia)**, όπως κείμενο, εικόνα, ήχος, βίντεο και άλλα. Τα **πλαίσια (frames)** ποικίλουν σε μέγεθος και αποτελούνται από **πακέτα (packets)** τα οποία είναι τα οι φυσικές μονάδες της μετάδοσης δεδομένων. Ένα πλαίσιο αποτελείται από την επικεφαλίδα (header), η οποία χρειάζεται για τον συγχρονισμό των δεδομένων καθώς επίσης περιέχει και μετα-πληροφορίες που δείχνουν τον τύπο και μήκος του πλαισίου. Στο τέλος κάθε πλαισίου ο χρήστης μπορεί να καθορίσει κάποιες συνθήκες, ώστε να γίνει η επιλογή από τα PCSs αυτά τα πλαίσια που ταιριάζουν καλύτερα με τις ανάγκες κάθε χρήστη. Από την στιγμή που τα **πλαίσια πληροφοριών (information frames)** εκπέμπονται με περιοδικό ρυθμό τότε μια ολοκληρωμένη μετάδοση των **πλαισίων πληροφοριών (information frames)** ονομάζεται **κύκλος μετάδοσης (broadcast cycle)**. Ένας **κύκλος μετάδοσης (broadcast cycle)** αρχίζει με ένα οποιοδήποτε πλαίσιο (frame) και ολοκληρώνεται όταν αυτό το πλαίσιο εμφανιστεί ξανά. Κάποια σημαντικά πλαίσια πληροφοριών μπορεί να επαναληφθούν και γενικότερα τα πλαίσια πληροφοριών μπορεί να εισαχθούν, να διαγραφούν ακόμα και να τροποποιηθούν. Όλες αυτές οι αλλαγές θα φανούν στον αμέσως επόμενο **κύκλος μετάδοσης (broadcast cycle)**.

Ο χρόνος που θα πρέπει τα PCSs να μείνουν σε **ενεργή κατάσταση (active mode)** για να απαντήσουν σε μια **σειρά ερωτήσεων (query)** ονομάζεται **tune-in time**, ο οποίος είναι σε αναλογία με την κατανάλωση ενέργειας. Ο χρόνος για την συλλογή όλων των επιθυμητών πλαισίων ονομάζεται **access time**. Αυτοί οι δύο χρόνοι **tune-in time** και **access time** είναι ίσοι με τον **κύκλο μετάδοσης (broadcast cycle)** ο καθένας. Αυτό συμβαίνει μιας και για να βρεθούν όλα τα επιθυμητά πλαίσια

πρέπει να γίνει αναζήτηση του σε όλο τον κύκλο μετάδοσης (**broadcast cycle**). Κάτι τέτοιο έρχεται σε σύγκρουση με την κατανάλωση ενέργειας μιας και είναι προφανές ότι μόνο μερικά πλαίσια σε έναν **κύκλο μετάδοσης (broadcast cycle)** είναι κάθε φορά τα επιθυμητά.



Σχήμα 1.1 : Ροή δεδομένων

Υπάρχει η δυνατότητα να αναπτυχθούν μέθοδοι πρόσβασης, έτσι ώστε τα PCSs να μπορούν να αλλάζουν από **κατάσταση (active mode)** σε **κατάσταση "ύπνου" (doze mode)** και συνεπώς να μειωθεί η κατανάλωση ενέργειας. Για να γνωρίζουμε εκ των προτέρων πια πλαίσια πληρούν τις προϋποθέσεις του κάθε χρήστη πρέπει να προστεθούν βοηθητικές πληροφορίες σχετικές με το περιεχόμενο των πλαισίων. Εξαιτίας του μικρού αριθμού καναλιών μετάδοσης, υποθέτουμε ότι μόνο ένα κανάλι χρησιμοποιείται για τις βασικές και για τις βοηθητικές πληροφορίες. Έχοντας μόνο ένα κανάλι μετάδοσης είναι λογικό να αυξάνεται το μήκος του **κύκλου μετάδοσης (broadcast cycle)** και συνεπώς θα αυξάνεται ο χρόνος για την συλλογή όλων των επιθυμητών πλαισίων (**access time**). Παρόλα αυτά θα μειωθεί ο χρόνος που θα πρέπει τα PCSs να μείνουν σε ενεργή κατάσταση (active mode) για να απαντήσουν σε μια σειρά ερωτήσεων (**query**) (**tune-in time**).

Έχουν προταθεί δύο μέθοδοι για την κωδικοποίηση βοηθητικών πληροφοριών πρόσβασης για ασύρματη μετάδοση δεδομένων, η **αναζήτηση (hashing)** και η **δεικτοδότηση (indexing)**. Στην δική μας μελέτη θα θεωρήσουμε δεικτοδότηση με ένα μόνο **γνώρισμα (attribute)**. Η μεγαλύτερη πρόκληση σε αυτή τη μέθοδο ευρετηρίου είναι ότι τα δεδομένα πρέπει να διαπεραστούν σειριακά. Κάτι τέτοιο αυτομάτως αλλάζει και τους παράγοντες κόστους των τεχνικών ευρετηρίου.

Στη συνέχεια θα επικεντρωθούμε στην δεικτοδότηση δεδομένων προς μετάδοση με την **τεχνική υπογραφής αρχείων (signature file technique)**. Οι τεχνικές υπογραφής θεωρούνται πιο αργές συγκριτικά με τις δομές δέντρων. Από την άλλη πλευρά βέβαια σε ένα κανάλι μετάδοσης οι τεχνικές υπογραφής προτιμούνται από τις δομές δέντρων μιας και οι δεύτερες χάνουν το πλεονέκτημα της ταχύτητας. Τελικώς ένα βασικό πλεονέκτημα της **τεχνικής υπογραφής αρχείων (signature file technique)** είναι η απλότητα της που την καθιστά πολύ καλή επιλογή για πληροφορίες πραγματικού χρόνου.

1.2 Η Τεχνική της Υπογραφής

Οι τεχνικές υπογραφής έχουν χρησιμοποιηθεί ευρέως για εξόρυξη γνώσης, βάση δεδομένων εικόνας, βάση οπτικοακουστικών δεδομένων και άλλων συμβατικών βάσεων δεδομένων. Η υπογραφή ουσιαστικά αποτελεί μία αφηρημένη σύνοψη των πληροφοριών που βρίσκονται αποθηκευμένες σε ένα αρχείο. Εξετάζοντας μόνο την υπογραφή μπορούμε εύκολα να αποφανθούμε κατά πόσο το αρχείο περιέχει τις επιθυμητές πληροφορίες σε ένα ασύρματο περιβάλλον μετάδοσης.

Η υπογραφή ενός αρχείου προκύπτει αρχικά από την μετατροπή κάθε χρήσιμης πληροφορίας του αρχείου σε ένα **αλφαριθμητικό από bits (0 ή 1) (bit strings)** και στη συνέχεια συνενώνονται όλα τα **(bit strings)** στην **υπογραφή** ολόκληρου του **αρχείου (record signature)**. Κατά τη διάρκεια του φιλτραρίσματος μια υπογραφή της **ερώτησης (query)** κατασκευάζεται με τον ίδιο τρόπο και μετέπειτα συγκρίνεται με την **υπογραφή του αρχείου(record signature)**. Υπάρχουν τρία πιθανά αποτελέσματα από την σύγκριση των δύο αυτών υπογραφών: **(1) το αρχείο ταιριάζει με την ερώτηση ($S_Q \wedge S_I = S_Q$)** **(2) το αρχείο δεν ταιριάζει με την ερώτηση ($S_Q \wedge S_I \neq S_Q$)** **(3) η σύγκριση των υπογραφών συνιστά κάποιο ταίριασμα αλλά στην ουσία το αρχείο δεν ταιριάζει με τα κριτήρια αναζήτησης.** Η τελευταία περίπτωση ονομάζεται **λανθασμένο ταίριασμα (false drop)**. Για να αποφύγουμε **λανθασμένα ταιριάσματα (false drops)** θα πρέπει το αρχείο να συγκρίνεται κατευθείαν με την **ερώτηση (query)** αφού πρώτα βρεθεί ότι οι δύο υπογραφές ταιριάζουν. Το Σχήμα 1.2 δείχνει την γέννηση των υπογραφών και τη σύγκριση διαδικασιών ενός αρχείου μιας εταιρίας με δύο σταθερές **name** και **type**.

Σε ένα ασύρματο περιβάλλον, **η τεχνική της υπογραφής (signature technique)** προσφέρει πολλά πλεονεκτήματα για φιλτράρισμα πληροφοριών :

- μπορεί να χρησιμοποιηθεί σε πολλά είδη πληροφοριών
- είναι ιδιαίτερα αποτελεσματική για περιπτώσεις εξόρυξης πληροφορίας με πολλά γνωρίσματα (attributes)
- οι υπογραφές είναι πολύ εύκολο να παραχθούν και να αναζητηθούν, κάτι που είναι πολύ χρήσιμο για τα PCSs όπου η αναζήτηση πραγματικού χρόνου με περιορισμένο αποθηκευτικό χώρο είναι απαραίτητη
- η υπογραφή είναι πολύ πιο μικρή από το **πλαίσιο πληροφοριών (information frame)**, οπότε δεν αυξάνεται ιδιαίτερα ο χρόνος για την συλλογή όλων των επιθυμητών πλαισίων (**access time**)
- ένα **αρχείο υπογραφών (signature file)** είναι ουσιαστικά μια σειριακή δομή αρχείου, οπότε γίνεται εύκολη η μετάδοση στον "αέρα" και η αναζήτηση (scanning) από τα PCSs .

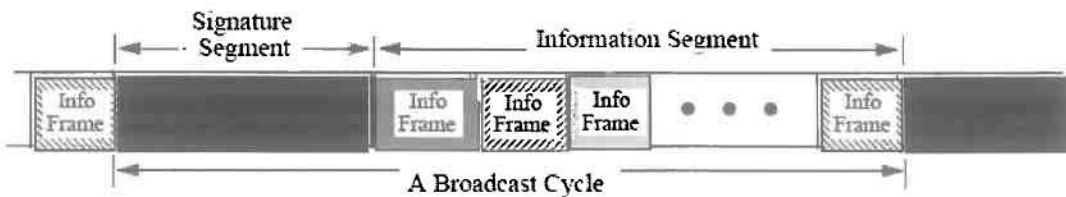
Στη συνέχεια θα δούμε τις τρεις μεθόδους υπογραφής για δεικτοδότηση και φιλτράρισμα πληροφορίας. Η επίδοση τους εκτιμάται βάση των χρόνων πρώτον που θα πρέπει τα PCSs να μείνουν σε ενεργή κατάσταση (active mode) για να απαντήσουν σε μια σειρά ερωτήσεων (query) που ονομάζεται **tune-in time** και του χρόνου για την συλλογή όλων των επιθυμητών πλαισίων που ονομάζεται **access time**. Έπειτα παρουσιάζονται τέσσερις πολιτικές για την μέθοδο την **πολυεπίπεδης υπογραφής (multi-level signature)** και αναλύεται η επίδοση βάση του **tune-in time**.

	attribute signatures:				
	IBM	001 000 110 010			
	Computer	000 010 101 001			
	record signature (V)	001 010 111 011			
record	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>IBM</td><td>Computer</td></tr></table>	IBM	Computer		
IBM	Computer				
	Queries	Query Signatures	Results		
	1) IBM	000 010 101 001	— match		
	2) GE	010 001 000 011	— no match		
	3) Ford	001 000 111 000	— false drop		

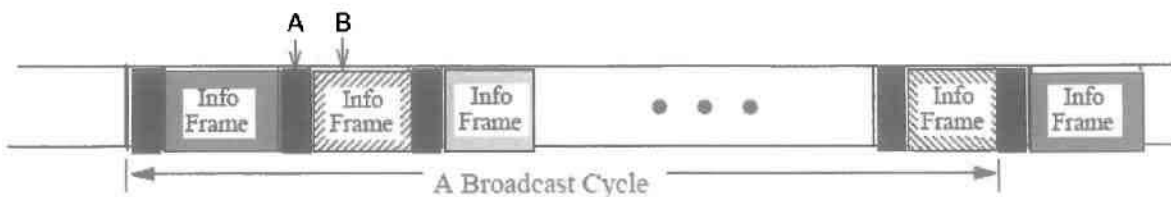
Σχήμα 1.2: Γέννηση υπογραφών και συγκρίσεις

1.3 Μετάδοση πληροφορίας με την τεχνική της υπογραφής

Οι υπογραφές προκύπτουν από τα πλαίσια πληροφοριών (**information frames**) και εκπέμπονται μαζί με τα αυτά. Οι υπογραφές μπορεί να μεταδοθούν σαν **μια ομάδα πριν από τα πλαίσια πληροφοριών (non-interleaved signatures)** ή να μεταδοθούν **οι υπογραφές μαζί με τα πλαίσια πληροφοριών (interleaved signatures)**. Τα Σχήματα 1.3 και 1.4 δείχνουν αυτούς τους δύο τρόπους.



Σχήμα 1.3 : Μετάδοση υπογραφών μαζικά πριν τα πλαίσια πληροφοριών



Σχήμα 1.4 : Μετάδοση υπογραφών μαζί με το αντίστοιχο πλαίσιο πληροφορίας η κάθε μια

Για την προσέγγιση με την **μετάδοση των υπογραφών μαζικά πριν τα πλαίσια πληροφοριών (non-interleaved signatures)**, αφού ο χρήστης μπορεί να αρχίσει να παρακολουθεί τον **κύκλο μετάδοσης (broadcast cycle)** οποιαδήποτε στιγμή, τότε μπορεί να τύχει να έχει περάσει η ομάδα των υπογραφών και έτσι να πρέπει να περιμένει μέχρι τον επόμενο **κύκλο μετάδοσης (broadcast cycle)**. Ο χρόνος από την στιγμή που ένας χρήστης ξεκινά να παρακολουθεί έναν **κύκλο μετάδοσης (broadcast cycle)** μέχρι να ληφθεί η πρώτη υπογραφή ονομάζεται **initial probe**

time. Είναι προφανές ότι η μέθοδος **non-interleaved** δεν είναι η επιθυμητή, επειδή οδηγεί σε μεγαλύτερο χρόνο μέχρι να ληφθεί η πρώτη υπογραφή (**initial probe time**). Συνεπώς, σε αυτή την μελέτη θεωρούμε μόνο **interleaving** προσεγγίσεις, όπου η υπογραφή κάθε πλαισίου μεταδίδεται μαζί με αυτό. Σε αυτήν την μελέτη θα θεωρήσουμε ότι τα PCSs μένουν σε **κατάσταση "ύπνου" (doze mode)** κατά το χρονικό διάστημα μέχρι να ληφθεί η πρώτη υπογραφή (**initial probe time**). Οι τρεις μέθοδοι που θα συζητήσουμε βασίζονται σε **interleaving** προσεγγίσεις.

1.3.1 Μέθοδος Απλής Υπογραφής (Simple Signature Scheme)

Στην μέθοδο απλής υπογραφής κατασκευάζεται μια υπογραφή για κάθε πλαίσιο και μεταδίδεται και μετά ακολουθεί το αντίστοιχο πλαίσιο. Όταν ένας κινητός χρήστης θέλει να αποσπάσει κάποιες πληροφορίες από το κανάλι μετάδοσης, καθορίζει μια ερώτηση (query) στο PCS του. Βάση της ερώτησης παράγεται και η αντίστοιχη υπογραφή της, S_Q . Στη συνέχεια το PCS «ακούει» το κανάλι και συγκρίνει την S_Q με τις υπογραφές των άλλων πλαισίων. Όταν βρεθεί κάποιο ταίριασμα τότε λαμβάνεται το αντίστοιχο πλαίσιο από το PCS για περαιτέρω έλεγχο μήπως υπάρχει **λανθασμένο ταίριασμα (false drop)**. Εάν τελικά δεν είναι **λανθασμένο ταίριασμα (false drop)**, το πλαίσιο κρατείται για το αποτέλεσμα. Εάν η υπογραφή της ερώτησης δεν ταιριάζει με την υπογραφή του πλαισίου, το PCS αλλάζει σε **κατάσταση "ύπνου" (doze mode)** και περιμένει να φτάσει η επόμενη υπογραφή πλαισίου. Ακόμα και αν περάσει αρκετή ώρα μέχρι το επόμενο ταίριασμα, το PCS θα παραμείνει σε **κατάσταση "ύπνου" (doze mode)** για το μεγαλύτερο μέρος του **κύκλου μετάδοσης (broadcast cycle)** και έτσι θα εξοικονομήσουμε ενέργεια.

Στη μέθοδο της απλής υπογραφής ο **μέσος** χρόνος μέχρι να ληφθεί η πρώτη υπογραφή (**initial probe time**) είναι το μισό του μέσου μεγέθους ενός πλαισίου και της υπογραφής του. Ο χρόνος που θα πρέπει τα PCSs να μείνουν σε ενεργή κατάσταση (active mode) για να απαντήσουν σε μια σειρά ερωτήσεων (query) που ονομάζεται **tune-in time** και ο χρόνος για την συλλογή όλων των επιθυμητών πλαισίων που ονομάζεται **access time** είναι ανεξάρτητα από την θέση που θα ληφθεί η πρώτη υπογραφή (**initial probe time**).

Θέση A: εάν ξεκινώντας να "ακούμε" ένα κανάλι βρισκόμαστε στη μέση μιας υπογραφής (θέση A στο Σχήμα 1.4), τότε το PCS θα πρέπει να παραμείνει σε **ενεργή κατάσταση (active mode)** για το υπόλοιπο της συγκεκριμένης υπογραφής μέχρι να εντοπίσει την αρχή ενός άλλου πλαισίου. Έπειτα το PCS μπορεί να αλλάξει σε **κατάσταση "ύπνου" (doze mode)**. Έτσι οι χρόνοι διαμορφώνονται ως εξής :

Initial probe time = το μήκος της εναπομείνουσας υπογραφής + το μήκος του πρώτου πλαισίου πληροφοριών.

Access time = initial probe time + broadcast cycle

Tune-in time = μήκος της εναπομείνουσας υπογραφής + κάθε υπογραφή σε

έναν κύκλο εκπομπής (broadcast cycle) + λανθασμένο ταίριασμα (false drop) και **σωστό ταίριασμα(true drop)** σε έναν κύκλο εκπομπής (broadcast cycle).

Θέση B: εάν ξεκινώντας να "ακούμε" ένα κανάλι βρισκόμαστε στη μέση ενός πλαισίου πληροφοριών (θέση B στο Σχήμα 1.4), τότε το PCS θα πρέπει να παραμείνει σε **ενεργή κατάσταση(active mode)** για να εντοπίσει την αρχή της επόμενης υπογραφής ενός πλαισίου. Έτσι οι χρόνοι διαμορφώνονται ως εξής :

Initial probe time = το μήκος του εναπομείναντος πλαισίου πληροφοριών.

Access time = initial probe time + broadcast cycle

Tune-in time = initial probe time + κάθε υπογραφή σε έναν κύκλο εκπομπής (broadcast cycle) + λανθασμένο ταίριασμα (false drop) και **σωστό ταίριασμα(true drop)** σε έναν κύκλο εκπομπής (broadcast cycle).

1.3.2 Μέθοδος Ενοποιημένης Υπογραφής (Integrated Signature Scheme)

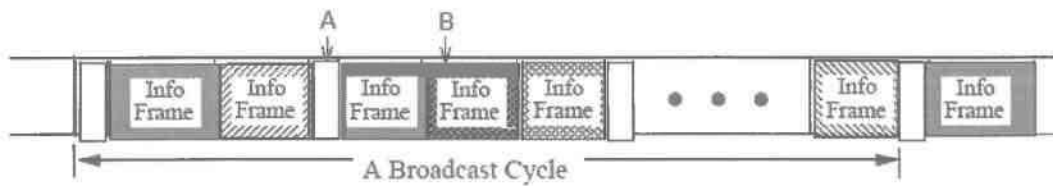
Η μέθοδος της Ενοποιημένης Υπογραφής αποτελεί μια γενίκευση της μεθόδου της Απλής Υπογραφής. Ουσιαστικά η Ενοποιημένη Υπογραφή αντιπροσωπεύει μια ομάδα από πλαίσια πληροφοριών (frame group). Η ενοποιημένη υπογραφή μεταδίδεται πριν από το πλαίσιο πληροφορίας. Το Σχήμα 1.5 δείχνει πως λειτουργεί η ενοποιημένη υπογραφή όπου για παράδειγμα η πρώτη υπογραφή αντιπροσωπεύει δύο πλαίσια πληροφορίας ενώ η δεύτερη υπογραφή αντιπροσωπεύει τρία πλαίσια πληροφορίας.

Στην διαδικασία του φιλτραρίσματος ουσιαστικά συμβαίνει ότι και στην μέθοδο της απλής υπογραφής. Δηλαδή αρχικά ελέγχεται αν ταιριάζουν οι υπογραφές της ερώτησης και της ομάδας πλαισίων, εάν υπάρχει ταίριασμα ελέγχεται περαιτέρω κάθε πλαίσιο από την ομάδα να ταιριάζει με την ερώτηση και επιλέγονται τα συμβατά πλαίσια, εάν δεν υπάρχει ταίριασμα υπογραφών συνεχίζεται ο έλεγχος με την επόμενη ομάδα πλαισίων. Στην πρώτη περίπτωση το PCS σπαταλάει αρκετή ενέργεια μένοντας σε **ενεργή κατάσταση(active mode)** μέχρι να βρεθούν τα κατάλληλα πλαίσια, ενώ στην δεύτερη περίπτωση το PCS μπορεί να μείνει σε **κατάσταση "ύπνου" (doze mode)** μέχρι να βρεθεί κάποιο ταίριασμα.

Είναι προφανές ότι από την στιγμή που έχουμε μία υπογραφή για παραπάνω από ένα πλαίσιο πληροφορίας υπάρχει μεγαλύτερη πιθανότητα για **λάθος ταίριασμα (false drop)**. Γι αυτό ακριβώς τον λόγο υπάρχει ανάγκη μιας πιο καλής οργάνωσης των πλαισίων, δηλαδή να ομαδοποιούνται τα πλαίσια με παρόμοιες πληροφορίες για να μειωθεί η πιθανότητα λανθασμένου ταιριάσματος (false drop). Η προηγούμενη ιδέα μπορεί πραγματικά να βοηθήσει δεδομένου ότι δεν μας ενδιαφέρει η σειρά των

πλαισίων πληροφορίας, οπότε μπορούμε να ομαδοποιούμε πλαίσια που ομοιάζουν μεν αλλά υπό άλλες συνθήκες θα ήταν σε μη γειτονικές θέσεις στον κύκλο εκπομπής.

Ο μέσος χρόνος μέχρι να ληφθεί η πρώτη υπογραφή (**initial probe time**) είναι το μισό του μέσου μεγέθους της ομάδας των πλαισίων και της υπογραφής τους. Ωστόσο ο μέσος χρόνος μέχρι να ληφθεί η πρώτη υπογραφή (**initial probe time**) είναι μεγαλύτερος από τον αντίστοιχο της μεθόδου της απλής υπογραφής. Θεωρούμε ότι αρχικά το PCS είναι σε κατάσταση "ύπνου" (**doze mode**) μέχρι να ληφθεί η πρώτη υπογραφή. Όπως και πριν οι **tune-in time** και **access time** είναι ανεξάρτητοι από την θέση που θα ληφθεί η πρώτη υπογραφή (**initial probe time**).



Σχήμα 1.5 : Μέθοδος Ενοποιημένης Υπογραφής

Θέση A: εάν ξεκινώντας να "ακούμε" ένα κανάλι βρισκόμαστε στη μέση μιας ενοποιημένης υπογραφής (θέση A στο Σχήμα 1.5), τότε το PCS θα πρέπει να παραμείνει σε **ενεργή κατάσταση (active mode)** για το υπόλοιπο της συγκεκριμένης υπογραφής μέχρι να εντοπίσει την επόμενη. Έπειτα το PCS μπορεί να αλλάξει σε **κατάσταση "ύπνου" (doze mode)**. Έτσι οι χρόνοι διαμορφώνονται ως εξής :

Initial probe time = το μήκος της εναπομείνουσας ενοποιημένης υπογραφής
+ το μήκος του πρώτου πλαισίου πληροφοριών.

Access time = initial probe time + broadcast cycle

Tune-in time = μήκος της εναπομείνουσας ενοποιημένης υπογραφής + κάθε υπογραφή σε έναν κύκλο εκπομπής (broadcast cycle) + λανθασμένο ταίριασμα (false drop) και **σωστό ταίριασμα (true drop)** σε έναν κύκλο εκπομπής (broadcast cycle).

Θέση B: εάν ξεκινώντας να "ακούμε" ένα κανάλι βρισκόμαστε στη μέση ενός πλαισίου πληροφοριών (θέση B στο Σχήμα 1.5), τότε το PCS θα πρέπει να παραμείνει σε **ενεργή κατάσταση (active mode)** για να εντοπίσει την αρχή της επόμενης υπογραφής μιας ομάδας πλαισίου. Έτσι οι χρόνοι διαμορφώνονται ως εξής :

Initial probe time = το μήκος της εναπομείνουσας ενοποιημένης υπογραφής
+ το μήκος του πρώτου πλαισίου πληροφοριών.

Access time = initial probe time + broadcast cycle

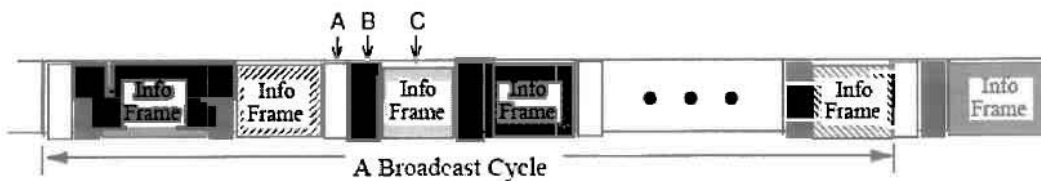
Tune-in time = μήκος της εναπομείνουσας ενοποιημένης υπογραφής + κάθε ενοποιημένη υπογραφή σε έναν κύκλο εκπομπής (broadcast cycle) + λανθασμένο ταίριασμα (false drop) και **σωστό ταίριασμα (true drop)** σε έναν κύκλο εκπομπής (broadcast cycle).

1.3.3 Μέθοδος Πολυεπίπεδης Υπογραφής (Multi-level Signature Scheme)

Η μέθοδος την πολυεπίπεδης υπογραφής είναι ένας συνδυασμός της μεθόδου της απλής υπογραφής και της μεθόδου της ενοποιημένης υπογραφής. Στα πιο υψηλά επίπεδα οι υπογραφές είναι ενοποιημένου τύπου και στα χαμηλότερα επίπεδα είναι απλού τύπου. Το Σχήμα 1.6 δείχνει μία υπογραφή 2^{ου} επιπέδου. Οι άσπρες υπογραφές του σχήματος είναι ενοποιημένου τύπου και για την συγκεκριμένη περίπτωση αντιπροσωπεύει δύο πλαίσια.. Οι μαύρες υπογραφές είναι απλού τύπου και αντιπροσωπεύουν κάθε ένα πλαίσιο πληροφορίας.

Για να απαντηθεί μία ερώτηση κατασκευάζονται δύο υπογραφές για την ερώτηση. Η μια υπογραφή S_Q είναι για το πιο πάνω επίπεδο των ενοποιημένων υπογραφών και η άλλη S_Q είναι για το πιο κάτω επίπεδο των απλών υπογραφών. Γίνεται πρώτα η σύγκριση με την S_Q και την ενοποιημένη υπογραφή των πλαισίων(εδώ δύο πλαίσια), αν βρεθεί ταίριασμα προχωρούμε σε σύγκριση της S_Q με την απλή υπογραφή κάθε πλαισίου και βρίσκουμε τα συμβατά πλαίσια ή έχουμε false drops. Αν δεν βρεθεί ταίριασμα με την πρώτη σύγκριση τότε το PCS αλλάζει σε **κατάσταση "ύπνου" (doze mode)** μέχρι να ληφθεί η επόμενη υπογραφή.

Σε σχέση με τις δύο προηγούμενες μεθόδους εδώ μπορεί να επιτευχθεί καλύτερος χρόνος που θα πρέπει τα PCSs να μείνουν σε ενεργή κατάσταση (active mode) για να απαντήσουν σε μια σειρά ερωτήσεων (query) (**tune-in time**). Όπως και πριν οι **tune-in time** και **access time** είναι ανεξάρτητοι από την θέση που θα ληφθεί η πρώτη υπογραφή (**initial probe time**).



Σχήμα 1.6 : Μέθοδος Πολυεπίπεδης Υπογραφής

Θέση A: εάν ξεκινώντας να "ακούμε" ένα κανάλι βρισκόμαστε στη μέση μιας ενοποιημένης υπογραφής (θέση A στο Σχήμα 1.6), τότε το PCS θα πρέπει να παραμείνει σε **ενεργή κατάσταση(active mode)** για το υπόλοιπο της συγκεκριμένης υπογραφής. Έτσι οι χρόνοι διαμορφώνονται ως εξής :

Initial probe time = το μήκος της εναπομείνουσας ενοποιημένης υπογραφής

Access time = initial probe time + broadcast cycle – την πρώτη ενοποιημένη υπογραφή (initial integrated signature)

Tune-in time = initial probe time + κάθε απλή υπογραφή στην πρώτη ομάδα πλαισίων + όλες οι ενοποιημένες υπογραφές πλην της αρχικής ενός κύκλου εκπομπής (broadcast cycle) πλην της αρχικής + απλές υπογραφές που ακολουθούν τις ενοποιημένες + false drop και true drop που σχετίζονται με τις αντίστοιχες απλές υπογραφές

Θέση B: εάν ξεκινώντας να "ακούμε" ένα κανάλι βρισκόμαστε στη μέση μιας απλής υπογραφής (θέση B στο Σχήμα 1.6), τότε το PCS θα πρέπει να παραμείνει σε **ενεργή κατάσταση (active mode)** για το υπόλοιπο της συγκεκριμένης υπογραφής και μετά αλλάζει σε **κατάσταση "ύπνου" (doze mode)**. Έτσι οι χρόνοι διαμορφώνονται ως εξής :

Initial probe time = το μήκος της απλής υπογραφής και του αντίστοιχου πλαισίου πληροφορίας .

Access time = initial probe time + ένας ολόκληρος broadcast cycle

Tune-in time = το μήκος της εναπομείνουσας ενοποιημένης υπογραφής + κάθε απλή υπογραφή πριν την επόμενη ενοποιημένη υπογραφή + όλες τις ενοποιημένες υπογραφές ενός κύκλου εκπομπής + τις απλές υπογραφές που σχετίζονται με τις ενοποιημένες υπογραφές που έχει βρεθεί ταίριασμα + (εάν η τελευταία ενοποιημένη υπογραφή βρεθεί ότι ταιριάζει) οι απλές υπογραφές πριν τον initial probe + false drop και true drop πλαίσια πληροφορίας που σχετίζονται με τις απλές υπογραφές για τις οποίες έχει βρεθεί ταίριασμα.

Θέση Γ: εάν ξεκινώντας να "ακούμε" ένα κανάλι βρισκόμαστε στη μέση ενός πλαισίου πληροφορίας (θέση C στο Σχήμα 1.6), τότε το PCS θα πρέπει να παραμείνει σε **ενεργή κατάσταση (active mode)** για το υπόλοιπο του συγκεκριμένου πλαισίου. Έτσι οι χρόνοι διαμορφώνονται ως εξής :

Initial probe time = το μήκος του εναπομείναντος πλαισίου πληροφοριών.

Access time = initial probe time + broadcast cycle

Tune-in time = το μήκος του εναπομείναντος πλαισίου πληροφοριών + κάθε απλή υπογραφή πριν την επόμενη ενοποιημένη υπογραφή + όλες τις ενοποιημένες υπογραφές ενός κύκλου εκπομπής + τις απλές υπογραφές που σχετίζονται

με τις ενοποιημένες υπογραφές που έχει βρεθεί ταιρίασμα + (εάν η τελευταία ενοποιημένη υπογραφή βρεθεί ότι ταιριάζει) οι απλές υπογραφές πριν τον initial probe + false drop και true drop πλαίσια πληροφορίας που σχετίζονται με τις απλές υπογραφές για τις οποίες έχει βρεθεί ταιρίασμα.

1.4 Ανάλυση Απόδοσης

Υπάρχουν διάφοροι παράγοντες που μπορεί να επηρεάσουν τους χρόνους tune-in time και access time των υπογραφών των πλαισίων. Σημαντικό ρόλο παίζουν το μέγεθος και ο αριθμός των υπογραφών, η ικανότητα φιλτραρίσματος των υπογραφών, το λανθασμένο ταιρίασμα των υπογραφών και ο χρόνος μέχρι να ληφθεί η πρώτη υπογραφή που ονομάζεται (initial probe time).

1.4.1 Σύμβολα και Παράμετροι

- A : αριθμός των πλαισίων πληροφορίας σε ένα κύκλο εκπομπής (broadcast cycle).
- A_f : αριθμός των πλαισίων πληροφορίας που έχουν ληφθεί εξαιτίας λανθασμένων ταιριασμάτων (false drops).
- A_t : αριθμός των πλαισίων πληροφορίας που έχουν ληφθεί εξαιτίας σωστών ταιριασμάτων (true drops).
- I : αριθμός ενοποιημένων υπογραφών σε έναν κύκλο εκπομπής (broadcast cycle).
- I_f : αριθμός ενοποιημένων υπογραφών που έχουν ταιριαστεί λανθασμένα (false drops)
- P_f : πιθανότητα λανθασμένου ταιριάσματος (false drop probability).
- P_f^s : πιθανότητα λανθασμένου ταιριάσματος (false drop probability) για απλές υπογραφές.
- P_f¹ : πιθανότητα λανθασμένου ταιριάσματος (false drop probability) για ενοποιημένες υπογραφές.
- P_S : επιλεξιμότητα μιας ερώτησης
- P(x, y) : πιθανότητα ένα συγκεκριμένο σετ από x bits να είναι στο 1 σε ένα πλαίσιο πληροφορίας το οποίο έχει επιβληθεί από y αλφαριθμητικά γνωρίσματα (attribute bit strings).
- k : αριθμός των πλαισίων πληροφορίας που έχουν χρησιμοποιηθεί για κάθε ενοποιημένη υπογραφή.
- l : μέσος όρος σωστών ταιριασμάτων σε μια ομάδα πλαισίων (frame group).
- m : μήκος της υπογραφής σε bits.
- n : μέσος αριθμός πακέτων σε ένα πλαίσιο πληροφορίας.
- p : αριθμός των bits σε ένα πακέτο.
- r : αριθμός πακέτων σε μια υπογραφή ($r = \lceil m/p \rceil$).
- s : αριθμός των αλφαριθμητικών από bit (bit string) που έχουν χρησιμοποιηθεί σε μια υπογραφή.
- w_b : αριθμός από 1 σε ένα αλφαριθμητικό από bits (bit string) που προκύπτουν από την αναζήτηση.
- w_f : μέσος όρος από 1 σε ένα πλαίσιο πληροφορίας ($w_f = m - w_t$)
- w_t : μέσος όρος από 0 σε ένα πλαίσιο πληροφορίας

1.4.2 Πιθανότητα Λανθασμένου Ταιριάσματος (False Drop Probability)

Η πιθανότητα λανθασμένου ταιριάσματος είναι ένας σημαντικός παράγοντας για την εκτίμηση του χρόνου για την συλλογή όλων των επιθυμητών πλαισίων που ονομάζεται **access time** και του χρόνου που θα πρέπει τα PCSs να μείνουν σε ενεργή κατάσταση (active mode) για να απαντήσουν σε μια σειρά ερωτήσεων (query) που ονομάζεται **tune-in time**. Η πιθανότητα λανθασμένου ταιριάσματος P_f είναι :

$$P_f = \frac{A_f}{A - A_f}$$

Στη συνέχεια βγάζουμε το P_f θεωρώντας ότι έχουμε μια ανεπιτυχή αναζήτηση απάντησης σε μια ερώτηση (query). Θεωρούμε ότι χρησιμοποιείται μια καλή συνάρτηση αναζήτησης ώστε κάθε πιθανό αλφαριθμητικό από bits (bit string) να έχει την ίδια πιθανότητα να συμμετάσχει στην παραγωγή υπογραφών για κάθε πλαίσιο.

Έτσι έχουμε :

$$P(x, y) = \sum_{i=0}^y (-1)^i \binom{y}{i} \binom{m-i}{w_b}^x \binom{m}{w_b}^{-x}$$

Εάν το x είναι επαρκώς μεγάλο και $w_b \ll m$,

$$P(x, y) = (1 - (1 - w_b/m)^x)^y$$

Η υπογραφή ενός πλαισίου φτιάχνεται συγχωνεύοντας αλφαριθμητικά από bits (bit strings) που αντιπροσωπεύουν τις σημαντικότερες πληροφορίες σε ένα πλαίσιο. Έτσι, $P(s, 1)$ αντιπροσωπεύει την πιθανότητα ένα bit θέσης β να είναι τιμής 1 στην υπογραφή του πλαισίου, όπου s είναι ο αριθμός των αξιών - κλειδιά (key values) σε ένα πλαίσιο πληροφορίας. Υπάρχουν m bits σε μια υπογραφή, οπότε και ο μέσος αριθμός από 1 σε ένα πλαίσιο είναι :

$$w_f = m P(s, 1) = m (1 - (1 - w_b/m)^s)$$

Έτσι ο μέσος αριθμός από 0 σε μια υπογραφή πλαισίου είναι :

$$w'_f = m - w_f = m (1 - w_b/m)^s \approx m e^{-(w_b s/m)}$$

Ένα λανθασμένο ταίριασμα (false drop) προκύπτει όταν κάθε ένα από τα bit στην υπογραφή του πλαισίου που αντιστοιχεί σε τιμή 1 στην υπογραφή της ερώτησης έχει τιμή 1. Δηλαδή όταν ισχύει η παρακάτω συνθήκη :

$$\text{Εάν } a_i = 1 \text{ τότε } \beta_i = 1, 1 \leq i \leq m$$

όπου a_i και β_i είναι το i -οστό bit της υπογραφής της ερώτησης και της υπογραφής του πλαισίου αντίστοιχα. Οπότε το λανθασμένο ταίριασμα (false drop) συμβαίνει όταν έχουμε την εξής συνθήκη :

$$\text{Εάν } \beta_i = 0 \text{ τότε } a_i = 0, 1 \leq i \leq m$$

Για μια ερώτηση με μια μόνο αξία –κλειδί (key value), η πιθανότητα λανθασμένου ταιριάσματος είναι :

$$\begin{aligned}
 P_f &= \text{Probability } [a_1 = 0 \wedge a_2 = 0 \wedge \dots \wedge a_{w'_f} = 0] \\
 &= \binom{m-w_f}{w_b} / \binom{m}{w_b} \\
 &= \frac{(m-w_b)! / (m-w'_f-w_b)!}{m! / (m-w'_f)!} \\
 &= \frac{m-w_b}{m} \frac{m-w_b-1}{m-1} \dots \frac{m-w_b-w'_f+1}{m-w'_f+1} \\
 &= \left(1 - \frac{w_b}{m}\right) \left(1 - \frac{w_b}{m-1}\right) \dots \left(1 - \frac{w_b}{m-w'_f+1}\right) \\
 &\approx \left(1 - \frac{1}{m}\right)^{w_b} \left(1 - \frac{1}{m-1}\right)^{w_b} \dots \left(1 - \frac{1}{m-w'_f+1}\right)^{w_b} \\
 &= \left(\frac{(m-1)(m-2)\dots(m-w'_f)}{m(m-1)\dots(m-w'_f+1)}\right)^{w_b} \\
 &= \left(\frac{m-w'_f}{m}\right)^{w_b} \\
 &= \left(1 - \frac{w'_f}{m}\right)^{w_b} \\
 &\approx \left(1 - e^{-w_b s/m}\right)^{w_b}.
 \end{aligned}$$

Ο παραπάνω τύπος δίνει βέλτιστα αποτελέσματα όταν

$$w_b = w_{opt} = m \ln 2 / s.$$

Συνεπώς η βέλτιστη πιθανότητα λανθασμένου ταιριάσματος είναι :

$$P_f \approx 0.5^{w_{opt}}.$$

1.4.3 Μοντέλα Κόστους (Cost Models)

Σε αυτήν την ενότητα θα δούμε τα μοντέλα κόστους για τους χρόνους initial probe time, access time και tune-in time για τις τρεις μεθόδους υπογραφών που έχουν ήδη αναφερθεί. Χρησιμοποιούμε τον αριθμό των πακέτων σαν την μονάδα εκτίμησης (μέτρησης) του χρόνου. Χάρην ευκολίας θεωρούμε ότι κάθε πλαίσιο πληροφορίας έχει τον ίδιο αριθμό πακέτων. Έτσι ο συνολικός αριθμός των πακέτων για τα δεδομένα είναι:

$$DATA = A * n$$

1.4.3.1 Κόστος για την Μέθοδο Απλής Υπογραφής

Ο συνολικός αριθμός των πακέτων για την απλή υπογραφή σε έναν κύκλο είναι:

$$SIG_S = A * [m/p] = A * r.$$

Χρησιμοποιούμε $CYCLE_S$ για να εκφράσουμε το μήκος ενός ολόκληρου κύκλου εκπομπής.

$$CYCLE_S = SIG_S + DATA.$$

Ο χρόνος initial probe time είναι το χρονικό διάστημα μέχρι να ληφθεί η επόμενη υπογραφή. Έτσι, ο μέσος initial probe time είναι:

$$\begin{aligned} ACCESS &= PROBE_S + CYCLE_S \\ &= (A + 0.5) (r + n). \end{aligned}$$

Το σύμβολο PT_S εκφράζει το χρονικό διάστημα που το PCS μένει σε active mode κατά τη διάρκεια του initial probe time.

$$\begin{aligned} PT_S &= \frac{r * 1/2 + r + n * 1/2 * n}{r + n} \\ &= \frac{r^2 + n^2}{2(r + n)} \end{aligned}$$

Για να υπολογίσουμε τον tune-in time πρέπει να υπολογίσουμε τον αριθμό των σωστών ταιριασμάτων (true drops). Θεωρούμε ότι P_S εκφράζει την επιλεκτικότητα (selectivity) της ερώτησης, ο αριθμός των σωστών ταιριασμάτων είναι:

$$A_t = A * P_S.$$

Στην διαδικασία φιλτραρίσματος, το PCS παραμένει σε ενεργή κατάσταση για κάθε υπογραφή πλαισίου, αληθινό ταίριασμα ή λανθασμένο ταίριασμα οπότε ο tune in time είναι:

$$TUNE_S = PT_S + SIG_S + A_t * n + A_f * n$$

$$\begin{aligned}
 &= PT_S + SIG_S + A_r * n + P_f^s * A * n - P_f^s * A * n \\
 &= PT_S + SIG_S + A * n * P_s + A * n * P_f^s - A * n * P_s * P_f^s \\
 &= PT_S + SIG_S + DATA * P_s + DATA * P_f^s - DATA * P_s * P_f^s
 \end{aligned}$$

1.4.3.2 Κόστος για την Μέθοδο Ενοποιημένης Υπογραφής

Θεωρούμε ότι k πλαίσια πληροφορίας έχουν ομαδοποιηθεί για να προκύψει η ενοποιημένη υπογραφή. Ο συνολικός αριθμός ενοποιημένων υπογραφών είναι :

$$I = \lfloor A/k \rfloor.$$

Ο συνολικός αριθμός των πακέτων για τις ενοποιημένες υπογραφές σε έναν κύκλο εκπομπής είναι:

$$SIG_i = I * r = \lfloor A/k \rfloor * r.$$

Οπότε το μήκος ενός ολόκληρου κύκλου εκπομπής για την ενοποιημένη υπογραφή είναι:

$$CYCLE_i = SIG_i + DATA.$$

Ο μέσος initial probe time είναι:

$$PROBE_i = (r + k * n) / 2.$$

Παρομοίως με την απλή υπογραφή, ο access time είναι :

$$ACCESS_i = PROBE_i + CYCLE_i .$$

Το σύμβολο PT_i εκφράζει το χρονικό διάστημα που το PCS μένει σε active mode κατά τη διάρκεια του initial probe time.

$$\begin{aligned}
 PT_S &= \frac{r * 1/2 * r + n * k * 1/2 * n}{r + n * k} \\
 &= \frac{r^2 + n^2 * k}{2(r + n * k)}
 \end{aligned}$$

Θεωρούμε το l , το μέσο αριθμό των πλαισίων που αντιστοιχούν σε ένα σωστό ταίριασμα ενοποιημένης υπογραφής. Έτσι ο tune-in time είναι:

$$\begin{aligned}
 TUNE_i &= PT_i + SIG_i + I_r * n * k + I_f * n * k \\
 &= PT_i + SIG_i + \lfloor \frac{A * P}{l} \rfloor * n * k + P_f^l * l * n * k \\
 &\approx PT_i + SIG_i + \lfloor \frac{P * l}{l} \rfloor * DATA + P_f^l * DATA.
 \end{aligned}$$

Η P_f^l είναι μια συνάρτηση του αριθμού των bit strings που υπερθέτονται και αναμένουμε να είναι μικρότερος από $s * k$.

1.4.3.3 Κόστος για την Μέθοδο Πολυεπίπεδης Υπογραφής

Εδώ θεωρούμε ένα σχήμα υπογραφής 2^{00} επιπέδου, το οποίο περιέχει από ενοποιημένες υπογραφές στο πάνω επίπεδο και απλές υπογραφές στο κάτω επίπεδο. Συνεπώς, ο συνολικός αριθμός από πακέτα που χρησιμοποιούνται για την κατασκευή της υπογραφής είναι :

$$SIG_m = SIG_i + SIG_s.$$

Το μήκος ενός ολόκληρου κύκλου είναι:

$$CYCLE_m = SIG_m + DATA.$$

Ο μέσος initial probe time προκύπτει βάση της πιθανότητας της θέσης που θα βρεθούμε ξεκινώντας να "ακούμε" ένα κανάλι εκπομπής καθώς και του αντίστοιχου probe time οπότε:

$$PROBE_m = \frac{k \cdot n}{k \cdot n + k \cdot r + r} \cdot \frac{n}{2} + \frac{k \cdot r}{k \cdot n + k \cdot r + r} \left(\frac{r}{2} + n \right) + \frac{r}{k \cdot n + k \cdot r + r} \cdot \frac{r}{2}$$

$$= \frac{k \cdot n^2 + (k+1)r^2 + 2 \cdot k \cdot nr}{2(k \cdot n + (k+1)r)}.$$

Ο access time μπορεί να εκτιμηθεί ως εξής:

$$ACCESS_m = PROBE_m + CYCLE_m.$$

Ο tune-in time στην αρχική αναζήτηση είναι:

$$PT_m = \frac{k \cdot n}{k \cdot n + k \cdot r + r} \cdot \frac{n}{2} + \frac{k \cdot r}{k \cdot n + k \cdot r + r} \cdot \frac{r}{2} + \frac{r}{k \cdot n + k \cdot r + r} \cdot \frac{r}{2}$$

$$= \frac{k \cdot n^2 + (k+1)r^2}{2(k \cdot n + (k+1)r)}.$$

Για να απλοποιήσουμε την φόρμουλα του tune-in time θεωρούμε ότι η ενοποιημένη υπογραφή για την αρχική αναζήτηση είναι ένα σωστό ταίριασμα (true drop). Οπότε ο tune-in time γίνεται:

$$TUNE_m = PT_m + SIG_i + k \cdot r + k \cdot P_s \cdot n$$

$$+ k \cdot P_f^s \cdot n + \left(1 - \left[P_s \cdot \frac{A}{l} \right] - 1 \right) \cdot P_f^i (k \cdot r + k \cdot P_f^s \cdot n)$$

$$+ \left[P_s \cdot \frac{A}{l} \right] \cdot (k \cdot r + l \cdot n + (k-1) \cdot P_f^s \cdot n).$$

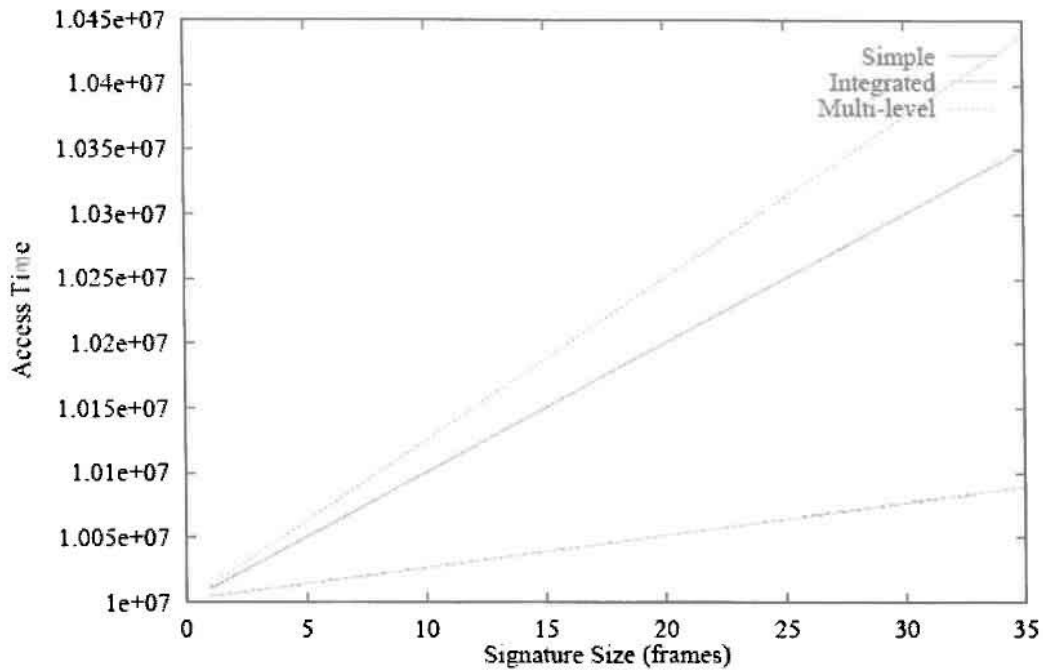
1.4.3.4 Συγκρίσεις

Συγκρίνουμε τους tune-in time και access time των τριών μεθόδων υπογραφής. Στον Πίνακα 1.1 έχουν δοθεί τιμές σε όλες τις παραμέτρους που χρησιμοποιούνται στην σύγκριση. Θεωρούμε δε στη σύγκριση πως η επιλεκτικότητα (selectivity) μιας ερώτησης είναι 1%. Για την ενοποιημένη και πολυεπίπεδη υπογραφή τέσσερα πλαίσια πληροφορίας έχουν ομαδοποιηθεί για να προκύψει η ενοποιημένη υπογραφή.

$p = 128$
$n = 1000$
$s_s = 100$
$s_l = 280$
$k = 4$
$l = 3$
$A = 10000$
$P_s = 0.01$

Πίνακας 1.1: Παράμετροι των μοντέλων κόστους

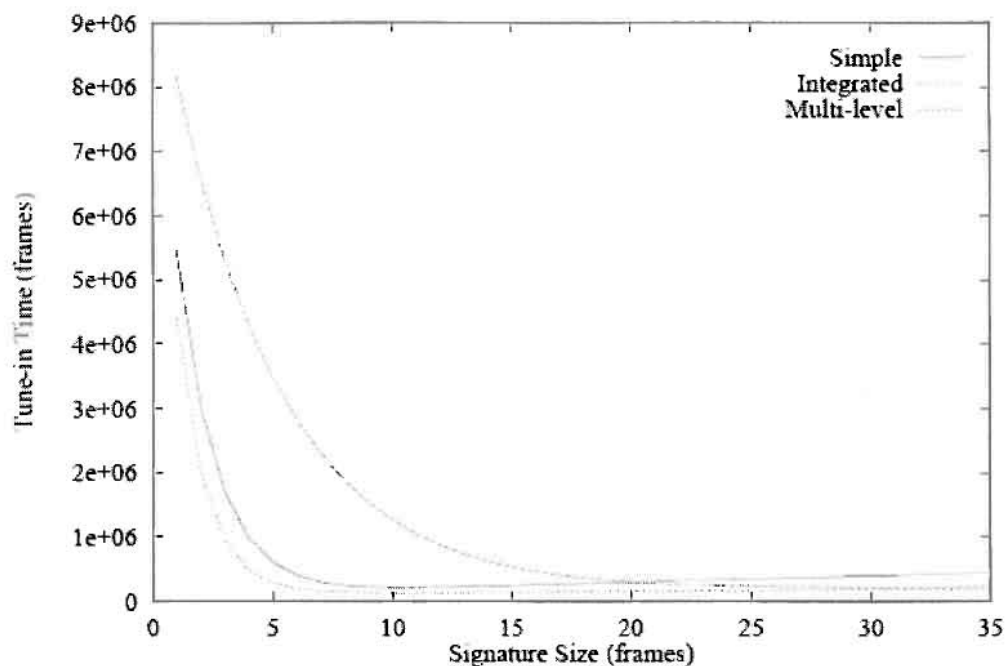
Η πιθανότητα λανθασμένου ταιριάσματος (false drop probability) για την απλή και ενοποιημένη υπογραφή μπορεί να υπολογιστεί βάση του m και του αριθμού των bit αλφαριθμητικών που έχουν υπερτεθεί, s_s και s_l . Από την στιγμή που η ενοποιημένη υπογραφή είναι καλύτερη για εκπομπή δεδομένων με παρόμοια πλαίσια πληροφορίας, θεωρούμε ότι το 30% των bit αλφαριθμητικών (bit strings) συμπίπτουν. Οπότε, $s_l = 70\% * s_s * 4$. Επίσης θεωρούμε ότι η τοπικότητα των πλαισίων είναι 3 (δηλαδή από τις ταιριαστές ενοποιημένες υπογραφές, τρία από τα αντίστοιχα τους πλαίσια πληροφορίας παρουσιάζουν σωστό ταιρίασμα) για ενοποιημένες και πολυεπίπεδες υπογραφές. Το μήκος της υπογραφής πλαισίου m κυμαίνεται από 1 έως 35 φυσικά πλαίσια για να παρατηρήσουμε τις διακυμάνσεις σε τους tune-in time και access time.



Σχήμα 1.7 : Access time σε σχέση με το μέγεθος της υπογραφής (signature frame)

Το Σχήμα 1.7 δείχνει ότι ο access time και των τριών μεθόδων υπογραφών είναι γραμμικώς ανάλογος με το μέγεθος της υπογραφής. Εάν παρατηρήσουμε στο Σχήμα 1.7 θα δούμε πως αν αθροίσουμε τις τιμές της απλής και ενοποιημένης υπογραφής ουσιαστικά προκύπτει η πολυεπίπεδη.

Το Σχήμα 1.8 δείχνει τον tune-in time για τις τρεις υπογραφές. Τα μεγέθη των υπογραφών και ο access time που αντιστοιχεί στον ελάχιστο tune-in time των τριών μεθόδων βρίσκονται στον Πίνακα 1.2.

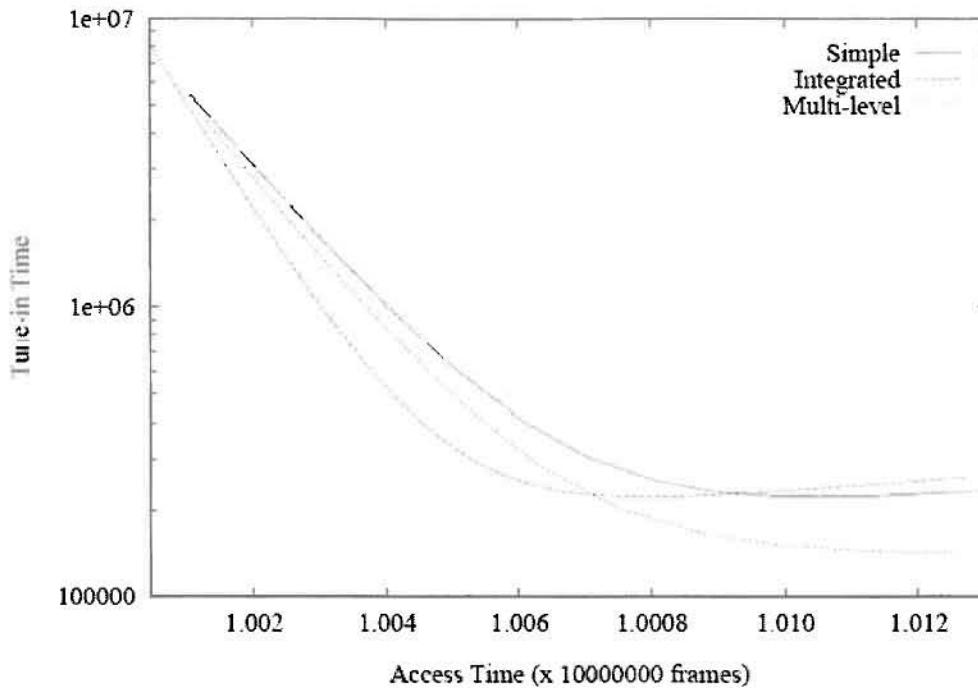


Σχήμα 1.8 : Tune-in time σε σχέση με το μέγεθος της υπογραφής

	Minimal Tune Time	Access Time	Sig. Size
Simple	221620.82	10100505.00	10
Integrated	222370.82	10077015.00	31
Multi-level	142319.26	10138007.58	11

Πίνακας 1.2 : Μέγεθος Υπογραφής, Access time, Ελάχιστος Tune-in time

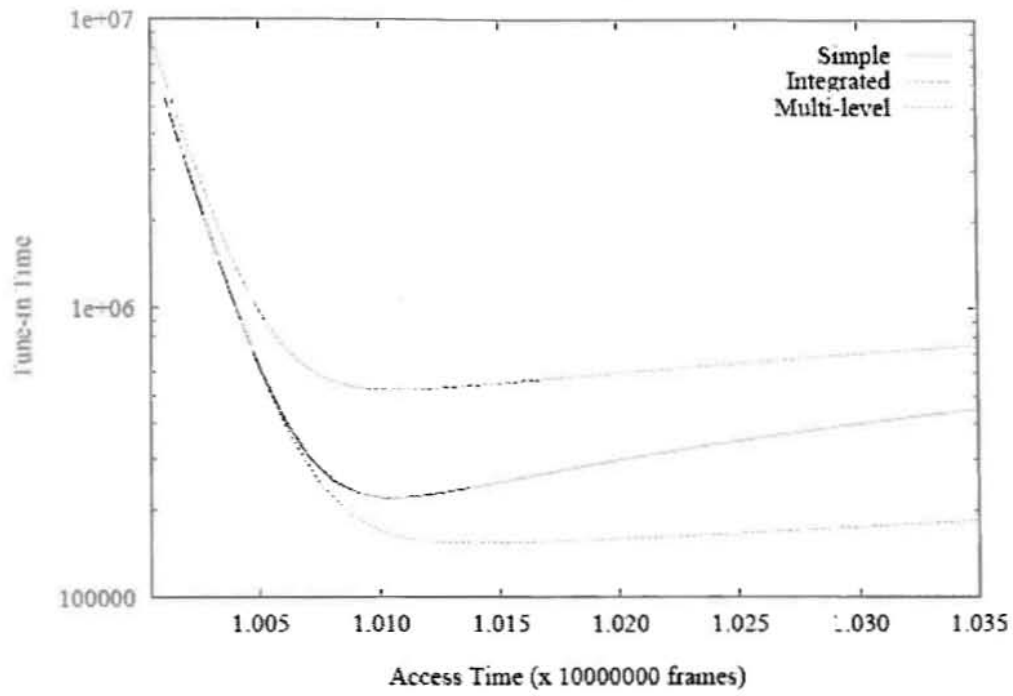
Το Σχήμα 1.9 συγκρίνει τον tune-in time στα πλαίσια του access time. Δεδομένου του access time η γραφική παράσταση μας δίνει την καλύτερη επιλογή ανάμεσα στις τρεις μεθόδους υπογραφών. Αναλύοντας ακόμα περισσότερο βλέπουμε ότι τελικά η ενοποιημένη και πολυεπίπεδη υπογραφή είναι πολύ καλύτερες μέθοδοι από την απλή. Η ενοποιημένη υπογραφή είναι η καλύτερη επιλογή όταν ο access time είναι μικρός και η πολυεπίπεδη υπογραφή ούτως η άλλως υπερσχύει και των δύο μεθόδων. Βέβαια η απόδοση της ενοποιημένης υπογραφής εξαρτάται από την τοπικότητα των πλαισίων πληροφορίας και τα προσπεράσματα που θα γίνουν λόγω μη ταιριάσματος.



Σχήμα 1.9 : Tune-in time σε σχέση με τον Access time ($l = 3$, $s_i = 270$)

Για την πολυεπίπεδη υπογραφή, οι υπογραφές στο χαμηλότερο επίπεδο είναι ουσιαστικά απλές υπογραφές. Λογικά περιμένουμε σε αυτήν την μέθοδο ότι ο tune-in time να είναι τουλάχιστον τόσο καλός όσο είναι της απλής υπογραφής με κάποια καθυστέρηση για τον access time για τα χαμηλότερα επίπεδα. Έτσι λοιπόν συγκρίνουμε την χειρότερη περίπτωση της ενοποιημένης (integrated) και της πολυεπίπεδης (multi-level) υπογραφής (η τοπικότητα είναι 1 και δεν υπάρχει προσπέρασμα bit strings σε ένα πλαίσιο δεδομένων).

Το Σχήμα 1.10 δείχνει την καθυστέρηση του access time (λιγότερο από 6% του βέλτιστου access time) και ο tune-in time της απλής υπογραφής και της πολυεπίπεδης είναι ουσιαστικά ο ίδιος. Για καθυστερήσεις στον access time μεγαλύτερες του 0.6% του βέλτιστου access time, η πολυεπίπεδη υπογραφή έχει πολύ καλύτερο tune-in time από την απλή υπογραφή.



Σχήμα 1.10 : Tune-in time σε σχέση με τον Access time ($l = 1, s_i = 400$)

1.4.4 Βελτιστοποίηση Υπογραφών

Έχοντας ως δεδομένο ότι η ενοποιημένη υπογραφή συνοψίζει την πληροφορία περισσότερων πλαισίων πληροφορίας από την απλή υπογραφή, η πιθανότητα λανθασμένου ταιριάσματος για τις ενοποιημένες υπογραφές είναι μεγαλύτερη από αυτήν των απλών υπογραφών. Από την άλλη πλευρά όμως αφού χρησιμοποιείται περισσότερο η μέθοδος της ενοποιημένης υπογραφής από την απλή, είναι λογικό να θέλουμε να μειώσουμε την πιθανότητα λανθασμένου ταιριάσματος (false drop probability) αυξάνοντας το μέγεθος της υπογραφής για να είναι πιο πιθανό το σωστό ταιρίασμα. Αυξάνοντας το μήκος της υπογραφής αυξάνεται και ο tune-in time.

Στη συνέχεια θα ρυθμίσουμε το συνολικό φόρτο των υπογραφών και τον χώρο που καταλαμβάνεται από τις απλές και ενοποιημένες υπογραφές για να παρατηρήσουμε τον βέλτιστο tune-in time για τις πολυεπίπεδες υπογραφές. Πάλι θεωρούμε $2^{\text{ου}}$ επιπέδου υπογραφή. Τα i και s χρησιμοποιούνται για τις ενοποιημένες (integrated) και απλές (simple) υπογραφές αντίστοιχα. Ας υποθέσουμε ότι το συνολικό μέγεθος της υπογραφής είναι q . Έτσι,

$$SIG_i = q * SIG_m$$

και

$$SIG_s = (1-q) * SIG_m .$$

Έτσι ο αριθμός των πακέτων για κάθε ενοποιημένη και απλή υπογραφή είναι :

$$r_i = [SIG_i / I] \quad \text{και} \quad r_s = [SIG_s / I] .$$

Το κόστος για την πολυεπίπεδη υπογραφή έχει ως εξής :

$$PROBE_m = \frac{k*n}{k*n+k*r_s+r_i} \frac{n}{2} + \frac{k*r_s}{k*n+k*r_s+r_i} \frac{r_s}{2} + \frac{r_i}{k*n+k*r_s+r_i} \frac{r_i}{2}$$

$$= \frac{k*n^2+(n+r_s)^2+r_i^2}{2(k*n+k*r_s+r_i)}$$

Ο tune-in time στην αρχική αναζήτηση είναι:

$$PT_m = \frac{n}{2} + \frac{k*r_s}{k*n+k*r_s+r_i} \frac{r_s}{2} + \frac{r_i}{k*n+k*r_s+r_i} \frac{r_i}{2}$$

Οπότε ο tune-in time γίνεται:

$$TUNE_m = PT_m + SIG_i + k * r_s + k * P_s * n$$

$$+ k * P_f^s * n + \left(1 - \left[P_s * \frac{A}{I} \right] - 1 \right) * P_f^i (k*r_s + k*P_f^s * n)$$

$$+ [P_s * \frac{A}{I}] * (k*r_s + l*n + (k-l) * P_f^s * n).$$

1.5 Αποθήκευση των Υπογραφών

Η αποθήκευση πληροφοριών που χρησιμοποιούνται πολύ συχνά στα PCSs μπορεί να μειώσουν τον tune-in time και την κατανάλωση ενέργειας, επειδή οι πληροφορίες που θα ζητούνται κάθε φορά θα έρχονται από την προσωρινή μνήμη (cache) χωρίς να πρέπει να συνδεθούν με το κανάλι εκπομπής.

Θεωρούμε το σχήμα της 2^{ου} επιπέδου υπογραφής . Υπάρχουν δύο παράγοντες που επηρεάζουν την διατήρηση των υπογραφών στην cache :

- **Ειδοποίηση ακύρωσης (Invalidation Notice)** : είναι μια ειδοποίηση για τις αποθηκευμένες υπογραφές στην cache που είναι πολύ παλιές. Επειδή οι υπογραφές στην cache πρέπει να είναι ακριβείς, ο εξυπηρετητής (server) πληροφοριών πρέπει να παρέχει ενημερώσεις για τις παλιές υπογραφές. Έτσι στην εκπομπή ο server αν υπάρχουν παλιές και συνεπώς υπογραφές που δεν αντιπροσωπεύουν τα πλαίσια πληροφορίας στέλνει ειδοποιήσεις ακύρωσης με τις αλλαγές που πρέπει να γίνουν. Υπάρχουν δύο ειδών ειδοποιήσεις που χρησιμοποιούνται σε διαφορετικές περιπτώσεις :
 - **Bit tags**: για κάθε πλαίσιο πληροφορίας ανατίθεται 1-bit tag που υποδηλώνει αν έχει αλλάξει το περιεχόμενο του πλαισίου από τον τελευταίο κύκλο εκπομπής.
 - **Αριθμοί Έκδοσης (Version Numbers)** : για να μειωθεί το κόστος επανάληψης της αποθήκευσης των υπογραφών στην cache κάθε φορά που το PCS συνδέεται με ένα κανάλι επικοινωνίας έχουμε τους **αριθμούς έκδοσης (version numbers)** για να ξέρουμε πότε αποθηκεύτηκε τελευταία φορά η κάθε υπογραφή. Κάθε φορά που αλλάζει το πλαίσιο πληροφορίας αυξάνεται κατά 1 και ο αντίστοιχος αριθμός έκδοσης (παίρνει τιμή 0 όταν φτάσει τον μέγιστο αντιπροσωπευτικό αριθμό).
- **Στρατηγική Ανανέωσης (Refresh Strategy)**: καθορίζει ποιες υπογραφές θα παραμείνουν στην cache.
 - **Ενεργητική Ανανέωση (Active Refresh)** : διατηρεί όλες τις υπογραφές στη cache και απλά τις ανανεώνει με τις ειδοποιήσεις ακύρωσης (invalidation notice).
 - **Παθητική Ανανέωση (Passive Refresh)** : διατηρεί στην cache τις υπογραφές που προσφάτως έχουν χρησιμοποιηθεί. Βάση των ειδοποιήσεων ακύρωσης, αντί να ανανεωθούν οι υπογραφές που έχουν αλλάξει όπως πριν, απλά διαγράφονται από την cache.

1.5.1 Bit Tags και Ενεργητική Ανανέωση (Active Refresh) (BA)

Η BA πολιτική διατηρεί όλες τις ενοποιημένες (integrated) υπογραφές και απλές υπογραφές στην προσωρινή μνήμη (cache memory). Για κάθε πλαίσιο πληροφορίας, χρησιμοποιούμε bit tags μπροστά από την ενοποιημένη υπογραφή για να υποδηλώσουν την ενημέρωση των πληροφοριών του πλαισίου στην ομάδα. Όταν το bit tag σηματοδοτεί ότι οι αντίστοιχες πληροφορίες πλαισίου έχουν αλλάξει, οι ενοποιημένες και απλές υπογραφές για το πλαίσιο πρέπει να αποθηκευτούν ξανά στην προσωρινή μνήμη. Οι υπογραφές των εκάστοτε υπογραφών στη συνέχεια ταιριάζονται με τις υπογραφές που διατηρούνται στην προσωρινή μνήμη για να αποφασιστεί ποια πλαίσια πληροφορίας θα προσπεραστούν και ποια θα πρέπει να μεταφερθούν στα PCs για να αποκλειστούν τα λανθασμένα ταιριάσματα.

1.5.2 Αριθμοί Έκδοσης (Version Numbers) και Ενεργητική Ανανέωση (Active Refresh) (VA)

Το VA σχήμα χρησιμοποιεί τους αριθμούς έκδοσης των πλαισίων πληροφορίας για να ενημερώσει τους κινητούς χρήστες για αλλαγές στα πλαίσια. Παρομοίως με τα bit tags, οι αριθμοί έκδοσης για τα πλαίσια πληροφορίας εκπέμπονται πριν από τις ενοποιημένες υπογραφές. Εάν ο αριθμός έκδοσης ενός πλαισίου στην συγκεκριμένη εκπομπή είναι διαφορετικός από αυτόν που έχει το πλαίσιο στην προσωρινή μνήμη, καταλαβαίνουμε ότι οι υπογραφές στην προσωρινή μνήμη δεν είναι έγκυρες. Έτσι λοιπόν πρέπει να ενημερωθούν οι υπογραφές που βρίσκονται στην προσωρινή μνήμη.

1.5.3 Bit Tags και Παθητική Ανανέωση (Passive Refresh) (BP)

Στην BP πολιτική αντί να αποθηκεύουμε όλες τις υπογραφές στο PCS, μόνο οι υπογραφές που έχουν ληφθεί από προηγούμενες ερωτήσεις κρατούνται στην μνήμη. Οι παλιές υπογραφές που δεν ισχύουν πλέον διαγράφονται από την προσωρινή μνήμη ανάλογα με τα bit tags. Νέες υπογραφές εισάγονται στην προσωρινή μνήμη μόνο εάν χρειάζονται στη καινούρια διαδικασία φιλτραρίσματος. Όπως στην BA πολιτική, τα bit tags που αντιστοιχούν σε ένα πλαίσιο πληροφορίας εκπέμπονται πριν από τις ενοποιημένες υπογραφές. Εάν τα bit tags δεν παρουσιάσουν καμία διαφορά στο πλαίσιο πληροφορίας, θα χρησιμοποιηθεί η ενοποιημένη υπογραφή στην προσωρινή μνήμη για την επεξεργασία της ερώτησης. Εάν τα bit tags παρουσιάσουν κάποια διαφορά σε κάποια πλαίσια πληροφορίας, οι αντίστοιχες ενοποιημένες και απλές υπογραφές στην προσωρινή μνήμη διαγράφονται και οι ενημερωμένες ενοποιημένες υπογραφές φορτώνονται στην προσωρινή μνήμη για την σύγκριση με την υπογραφή της ερώτησης. Εάν η σύγκριση μεταξύ της υπογραφής της ερώτησης και της ενοποιημένης υπογραφής καταλήξει σε ταίριασμα, οι απλές υπογραφές που δεν

βρίσκονται στην προσωρινή μνήμη φορτώνονται σε αυτήν και θα συγκριθούν με την απλή υπογραφή της ερώτησης

1.5.4 Αριθμοί Έκδοσης (Version Numbers) και Παθητική Ανανέωση (Passive Refresh) (VP)

Το σχήμα VP χρησιμοποιεί τους αριθμούς έκδοσης των πλαισίων πληροφορίας για να ενημερώσει τα PCs για αλλαγές στα πλαίσια και να ενημερώσει την προσωρινή μνήμη με έναν παθητικό τρόπο. Νέες υπογραφές εισάγονται στην προσωρινή μνήμη μόνο εάν χρειάζονται στη καινούρια διαδικασία φιλτραρίσματος.

ΚΕΦΑΛΑΙΟ 2

Τεχνικές εκπομπής πληροφορίας με πολλαπλά γνωρίσματα

Περιεχόμενα

2.1 Εισαγωγή.....	35
2.2 Παράμετροι για τις τεχνικές εκπομπής πληροφορίας με πολλά γνωρίσματα.....	36
2.2.1 Το Δέντρο Ευρετηρίου (Index Tree).....	38
2.2.1.1 Σύζευξη (Λ) Γνωρισμάτων για Index Tree.....	40
2.2.1.2 Διάζευξη (V) Γνωρισμάτων για Index.....	41
2.2.2 Η τεχνική της υπογραφής (Signature Scheme).....	42
2.2.2.1 Σύζευξη (Λ) Γνωρισμάτων για Signature Scheme.....	44
2.2.2.2 Διάζευξη (V) Γνωρισμάτων για Signature Scheme.....	45
2.2.3 Η υβριδική τεχνική (Hybrid Method).....	46
2.2.3.1 Σύζευξη (Λ) Γνωρισμάτων για Hybrid Method.....	46
2.2.3.2 Διάζευξη (V) Γνωρισμάτων για Hybrid Method.....	47
2.3 Ανάλυση Απόδοσης.....	48
2.3.1 Απόδοση Ερωτήσεων Σύζευξης (Λ).....	48
2.3.2 Απόδοση Ερωτήσεων Διάζευξης (V).....	50

ΚΕΦΑΛΑΙΟ 2 Τεχνικές εκπομπής πληροφορίας με πολλαπλά γνωρίσματα

2.1 Εισαγωγή

Στην πραγματικότητα οι εφαρμογές περιέχουν κομμάτια δεδομένων με πολλαπλά γνωρίσματα (multiple attributes). Επιπλέον οι ερωτήσεις που βασίζονται σε πολλαπλά γνωρίσματα παρέχουν πιο ακριβείς πληροφορίες στους χρήστες. Σε αυτό το κεφάλαιο θα δούμε τρεις **τεχνικές ευρετηρίου (indexing techniques)** με συντηρητική κατανάλωση ενέργειας, το **δέντρο ευρετηρίου (index tree)**, η **υπογραφή (signature)** και η **υβριδική (hybrid)**.

Όπως και στο κεφάλαιο 1 θεωρούμε ότι έχουμε ένα κοινό κανάλι επικοινωνίας μεταξύ του χρήστη και του σταθμού μέσω του οποίου γίνεται η εκπομπή δεδομένων. Για να λάβουν τα πλαίσια πληροφορίας οι κινητές εφαρμογές (mobile applications) ή οι χρήστες από τον αέρα, εκπέμπουν ερωτήσεις (queries) που καθορίζουν τις επιθυμητές τιμές για ένα ή περισσότερα γνωρίσματα. Σαν αποτέλεσμα έχουμε οι κινητοί υπολογιστές να "ακούν" το κανάλι επικοινωνίας και να επιστρέφουν στους χρήστες τα πλαίσια πληροφορίας που ικανοποιούν τις ερωτήσεις τους.

Οι **τεχνικές ευρετηρίου (indexing techniques)** βασίζονται στην ιδέα ότι με την εκπομπή βοηθητικών πληροφοριών μαζί με τα πλαίσια πληροφορίας (Information frames) σε ένα κανάλι εκπομπής μπορεί να προβλεφθεί από τους κινητούς υπολογιστές ο χρόνος άφιξης των πλαισίων. Αν λοιπόν είναι γνωστός ο χρόνος άφιξης των πλαισίων τότε το PCS μπορεί να μείνει σε κατάσταση "ύπνου" (doze mode) και έτσι να εξοικονομήσουμε ενέργεια. Αφού λοιπόν θα λαμβάνονται μόνο τα πλαίσια πληροφορίας που ικανοποιούν τις ερωτήσεις (queries) των κινητών πελατών (mobile clients) λογικό είναι ότι η CPU (Κεντρική Μονάδα Επεξεργασίας) κάθε PCS θα γλιτώνει χρόνο και χώρο στην προσωρινή μνήμη (cache). Εκτός από την δεικτοδότηση υπάρχουν και άλλοι δύο παράγοντες που καθορίζουν τον access time και tune-in time, η οργάνωση των πλαισίων πληροφορίας προς εκπομπή και οι μέθοδοι πρόσβασης στα δεδομένα (data access methods).

Ο **χρονοπρογραμματισμός δεδομένων (data scheduling)** καθορίζει το περιεχόμενο ενός κύκλου εκπομπής και την συχνότητα εκπομπής των πλαισίων πληροφορίας. Ένας απλός τρόπος χρονοπρογραμματισμού, ονομάζεται **επίπεδη εκπομπή (flat broadcast)**, είναι να εκπέμπεται κάθε πλαίσιο μια φορά σε κάθε κύκλο εκπομπής. Μια άλλη μέθοδος, η **εκπομπή δίσκων (broadcast disks)** εκπέμπει τα σημαντικά δεδομένα πιο συχνά από τα άλλα για να μειωθεί ο μέσος χρόνος για την συλλογή όλων των επιθυμητών πλαισίων (**average access time**) για ερωτήσεις με το συγκεκριμένο γνώρισμα. Βέβαια με την μέθοδο της **εκπομπής δίσκων (broadcast disks)** αυξάνεται ο κύκλος εκπομπής και ο πελάτης καθυστερεί να συλλέξει αυτές τις πληροφορίες που δεν εκπέμπονται τόσο συχνά. Σε αυτό το κεφάλαιο θεωρούμε ότι χρησιμοποιείται η μέθοδος **flat broadcast**.

Η **ομαδοποίηση δεδομένων (data clustering)** αναφέρεται στην οργάνωση των δεδομένων με ίδιες τιμές για ένα συγκεκριμένο γνώρισμα για να εμφανιστούν σειριακά σε ένα κύκλο εκπομπής. Αν ο πελάτης εντοπίσει μια πρώτη πληροφορία που τον ενδιαφέρει τότε μπορεί να πάρει όλες τις επόμενες πληροφορίες με την ίδια τιμή στο γνώρισμα που τον ενδιαφέρει. Υπάρχει η δυνατότητα μέσα στην κάθε ομαδοποίηση να γίνει και άλλη ομαδοποίηση της πληροφορίας βάση ενός άλλου γνωρίσματος. Για τα **γνωρίσματα που δεν έχει γίνει ομαδοποίηση (non-clustered attribute)** ο κύκλος εκπομπής μπορεί να χωριστεί σε ένα πλήθος από τεμάχια, τα **μετά-τεμάχια (meta segments)**. Ορίζουμε τον αριθμό των **meta segments** σε έναν κύκλο εκπομπής για ένα γνώρισμα ως τον **παράγοντα διασκορπισμού (scattering factor)** του γνωρίσματος.

2.2 Παράμετροι για τις τεχνικές εκπομπής πληροφορίας με πολλά γνωρίσματα

Πριν αναλύσουμε τις τρεις τεχνικές εκπομπής με πολλά γνωρίσματα πρέπει πρώτα να καθορίσουμε κάποιες παραμέτρους. Θεωρούμε ότι υπάρχουν m κοινά γνωρίσματα σε κάθε πλαίσιο πληροφορίας και τα γνωρίσματα ταξινομούνται βάση της συχνότητας πρόσβασης (access frequency). Υποθέτουμε ότι τα επιθυμητά γνωρίσματα είναι a_1, a_2, \dots, a_m . Και οι πιθανότητες τους να περιέχονται σε μια ερώτηση p_1, p_2, \dots, p_m όπου $p_i \geq p_{i+1}$, ($1 \leq i < m$). Το a_1 ονομάζεται το **σημαντικότερο γνώρισμα (major attribute)** και είναι αυτό το οποίο προσπελαύνεται τις περισσότερες φορές. Όλα τα άλλα γνωρίσματα ονομάζονται **ασήμαντα γνωρίσματα (minor attributes)**. Το a_m είναι το γνώρισμα που προσπελαύνεται το λιγότερο. Η δεικτοδότηση στηρίζεται στα πιο σημαντικά και πιο συχνά προσπελάσιμα γνωρίσματα. Οι υπόλοιπες παράμετροι καθορίζονται στον Πίνακα 2.1 παρακάτω.

F	Αριθμός πλαισίου πληροφορίας σε έναν κύκλο εκπομπής
P	Μέσος αριθμός πακέτων σε ένα πλαίσιο πληροφορίας
m	Αριθμός γνωρισμάτων σε ένα πλαίσιο πληροφορίας
r	Αριθμός δεικτοδοτημένων γνωρισμάτων σε ένα πλαίσιο πληροφορίας
q	Αριθμός γνωρισμάτων σε μια ερώτηση ($1 < q < m$)
S_i	Ποσοστό των πλαισίων με τις απαιτούμενες τιμές γνωρισμάτων
p_i	Πιθανότητα των ερωτήσεων να βασίζονται στο a_i

Πίνακας 2.1 : Παράμετροι Συστήματος

Επειδή σε μια ερώτηση με πολλά γνωρίσματα μπορούν να υπάρχουν πολλοί Boolean συνδυασμοί (σύζευξη (\wedge), διάζευξη (\vee)) σε αυτήν την μελέτη θα μελετήσουμε μόνο δύο περιπτώσεις, τη σύζευξη (\wedge) όλων των γνωρισμάτων και την διάζευξη (\vee) αυτών. Έτσι $Q\{a'_1, \dots, a'_q\}$ είναι μια ερώτηση με q γνωρίσματα a'_i ($1 \leq i \leq q \leq m$), τα οποία έχουν ταξινομηθεί ανάλογα με την σειρά που δεικτοδοτήθηκαν και είναι και τα q πρώτα γνωρίσματα που έχουν προσπελαστεί πιο συχνά. Οπότε οι δύο εκφράσεις που θα μελετήσουμε είναι:

$$a_1, a_2, \dots, a_q * Q\{a_1 \wedge \dots \wedge a_q\}$$

Και

$$Q\{a_1 \vee \dots \vee a_q\}.$$

Ξεκινώντας έχουμε την προσέγγιση μη ευρητηρίου των πλαισίων. Σε αυτή την προσέγγιση ο tune-in time είναι πολύ μεγάλος και ίσος με τον access time. Ο εκτιμώμενος tune-in time και access time για $Q\{a_1 \wedge \dots \wedge a_q\}$ είναι:

$$A(a_1 \wedge \dots \wedge a_q) = T(a_1 \wedge \dots \wedge a_q)$$

$$= P * F * (1/2 + S_i)$$

Όπου το a_i είναι ομαδοποιημένο. Τα A και T είναι για τον access time και τον tune-in time αντίστοιχα. Για $Q\{a_1 \vee \dots \vee a_q\}$, ο πελάτης πρέπει να σαρώσει ολόκληρο τον κύκλο εκπομπής για να αποσπάσει τα επιθυμητά πλαίσια πληροφορίας :

$$A\{a_1 \vee \dots \vee a_q\} = T\{a_1 \vee \dots \vee a_q\}.$$

$$= P * F$$

2.2.1 Δέντρο Ευρετηρίου (Index Tree)

Για δεδομένα με πολλά γνωρίσματα μπορούμε να κατασκευάσουμε ένα δέντρο ευρετηρίου για κάθε γνώρισμα. Για να μειωθεί το κόστος και για να μην υπάρχουν χρονικές καθυστερήσεις λογικό είναι ότι πρώτα πρέπει να κατασκευαστεί το δέντρο ευρετηρίου αυτού του γνωρίσματος με την μεγαλύτερη επιλεξιμότητα (αυτό που επιλέγεται πιο συχνά) και να ακολουθήσουν με φθίνουσα επιλεξιμότητα και τα άλλα γνωρίσματα. Από την άλλη πλευρά για να πετύχουμε τη βέλτιστη μέση απόδοση του συστήματος, η δεικτοδότηση πρέπει να γίνει βάση της **συχνότητας πρόσβασης (access frequency)**, έτσι ώστε όσο υψηλότερη είναι η συχνότητα πρόσβασης τόσο μεγαλύτερη και η προτεραιότητα στην επιλογή ενός γνωρίσματος για δεικτοδότηση. Όμως αν κατασκευάσουμε ένα δέντρο βάση της μικρότερης επιλεξιμότητας θα έχουμε υψηλότερο διαμερισμό και υψηλότερη ικανότητα φιλτραρίσματος. Βλέπουμε ότι οι παράγοντες της επιλεξιμότητας και της συχνότητας πρόσβασης έχουν συγκρουόμενα αποτελέσματα, οπότε πρέπει να βρεθεί μια ισορροπία για να αποφασιστεί η σειρά ευρετηρίου των γνωρισμάτων.

Η πρόσβαση είναι πιο εύκολη για τα **ομαδοποιημένα δεδομένα (clustered data)**, όπως και το κόστος σε αυτήν την περίπτωση είναι πολύ μικρότερο από ότι να είχαμε **μη-ομαδοποιημένα δεδομένα (non-clustered data)**. Επιλέγουμε λοιπόν να ομαδοποιήσουμε τα δεδομένα βάση του **σημαντικότερου γνωρίσματος (major attribute) a_1** . Οπότε τα **ασήμαντα γνωρίσματα (minor attributes)** δεν είναι ομαδοποιημένα. Ας θεωρήσουμε $I = \{a_1, a_2, \dots, a_m\}$ τα δεικτοδοτημένα γνωρίσματα. Ο κύκλος εκπομπής (broadcast cycle) χωρίζεται για κάθε γνώρισμα σε I ακολουθιακά μέρη από το a_1 μέχρι το a_r . Οπότε ο **παράγοντας διασκορπισμού (scattering factor)** του γνωρίσματος a_i , που συμβολίζεται M_i , αυξάνεται μαζί με το i . Το κόστος του **δέντρου ευρετηρίου (index tree)** του a_i εξαρτάται από το M_i . Συνεπώς, όσο μικρότερη η συνεισφορά του γνωρίσματος στο I τόσο μικρότερο και το κόστος.

Από την στιγμή που το κανάλι εκπομπής είναι ένα γραμμικό μέσο και η πληροφορία ευρετηρίου αυξάνει το μήκος του κύκλου εκπομπής, η δεικτοδότηση για ένα γνώρισμα επηρεάζει την απόδοση **ερωτήσεων (queries)** που βασίζονται όχι μόνο σε αυτό το γνώρισμα αλλά και σε άλλα. Όταν το M_i (όπου $i > r$), είναι πολύ μεγάλο, η κατασκευή ενός κατανεμημένου δέντρου ευρετηρίου για το a_1 δεν βοηθάει πολύ στην βελτίωση του **tune-in time** αλλά έχει ως αποτέλεσμα υψηλό κόστος ευρετηρίου.

Η τιμή του M_j εξαρτάται από το S_i και τη σχέση του με το a_i (όπου $1 < i < j$). Η εξάρτηση μεταξύ των γνωρισμάτων είναι δύσκολο να καθοριστεί οπότε για να απλοποιήσουμε το μοντέλο κόστους θεωρούμε ότι τα γνωρίσματα είναι τυχαία και ανεξάρτητα. Βασιζόμενοι σε αυτό το συμπέρασμα, μια απλή εκτίμηση του M_i είναι :

$$M_i = 1 / \prod_{j=1}^{(i-1)} S_j$$

h_i	Ύψος του δέντρου ευρετηρίου για το a_i
t_i	Ύψος του αντιγραμμένου μέρους του δέντρου ευρετηρίου για το a_i
N	Αριθμός των πακέτων σε έναν κόμβο του δέντρου ευρετηρίου
n	Κλειδί αναζήτησης και αριθμός δείκτη που κρατάει ένας κόμβος

Πίνακας 2.2 : Παράμετροι για το δέντρο ευρετηρίου (index tree)

Ο Πίνακας 2.2 περιγράφει τις παραμέτρους του μοντέλου κόστους του δέντρου ευρετηρίου. Επίσης, θεωρούμε τα $X[h]$ και $X[t]$ είναι ο αριθμός των κόμβων του δέντρου ευρετηρίου και του αντιγραμμένου μέρους του δέντρου ευρετηρίου αντίστοιχα. Επίσης, ο αριθμός των κόμβων στο i -οστό επίπεδο του δέντρου ευρετηρίου συμβολίζεται $L[i]$. Στη δεικτοδότηση για πολλά γνωρίσματα η επιπλέον πληροφορία για την δεικτοδότηση αυξάνει το μήκος του κύκλου σε $P \cdot F + N \cdot \sum_{j=1}^r E_j$ πακέτα, όπου E_j είναι ο συνολικός αριθμός κόμβων για το a_j . Για το a_j ($j \neq 1$), κατασκευάζεται ένα δέντρο ευρετηρίου για κάθε **μετά-τεμάχιο (meta segment)**. Οπότε υπάρχουν M_j δέντρα ευρετηρίου που αντιστοιχούν στο a_j . Αφού κάθε δέντρο ευρετηρίου έχει $X[h_j] + L[t_j + 1] - 1$ κόμβους ευρετηρίου για το a_j σε ένα μετά-τεμάχιο (meta segment) το συνολικό κόστος για την κατασκευή των δέντρων για το a_j σε έναν κύκλο εκπομπής είναι :

$$E_j = M_j \cdot (X[h_j] + L[t_j + 1] - 1) \text{ κόμβους.}$$

Θεωρούμε για ευκολία στους υπολογισμούς ότι όλα τα πλαίσια έχουν το ίδιο μέσο κόστος για δέντρο ευρετηρίου το οποίο είναι :

$$TREE = N \cdot \sum_{j=1}^r E_j / F.$$

Το αντιγραμμένο μέρος του δέντρου ευρετηρίου για το a_i εκπέμπεται κάθε $1/L[t_i + 1]$ για κάθε μετά-τεμάχιο (meta-segment). Τα πλαίσια πληροφορίας χωρίζονται σε $M_i \cdot L[t_i + 1]$ μετά-τεμάχια (meta-segments) με αντιγραμμένους κόμβους ευρετηρίου στην αρχή κάθε μετά-τεμαχίου (meta-segment). Το μήκος κάθε μετά-τεμαχίου (meta-segment) είναι $(P \cdot F + N \cdot \sum_{j=1}^r E_j) / (M_i \cdot L[t_i + 1])$.

Ο initial probe time για το δέντρο ευρετηρίου για το a_i είναι:

$$PROBE^{idx} = \frac{P \cdot F + N \cdot \sum_{j=1}^r E_j}{2 \cdot M_i \cdot L[t_i + 1]}.$$

2.2.1.1 Σύζευξη (\wedge) Γνωρισμάτων για Index Tree

Η γενική μέθοδος πρόσβασης για $Q \{a_1 \wedge \dots \wedge a_q\}$ είναι:

Αλγόριθμος Σύζευξης(\wedge) Γνωρισμάτων για Index Tree

Search: FOR each query attribute a_i , DO

- Initial probe for the index tree built based on a_i , within the data segment qualified for a_{i-1} .
- Search the index tree built based on a_i : the client follows a list of pointers to find out the arrival time of the desired data frame.

Retrieval: Scan the current meta segment for the desired data frames. At the end of the meta segment a jump is made to the other meta segments where the client should examine whether that meta segment is qualified for the attributes a_1, \dots, a_q contained in the query.

Ο access time για $Q \{a_1 \wedge \dots \wedge a_q\}$ έχει άνω όριο:

$$A^{idx}(a_1 \wedge \dots \wedge a_q) = \text{initial probe time για το major attribute index} \\ + \text{χρόνος αναμονής για το πρώτο επιθυμητό segment δεδομένων} \\ + \text{access time για τα επιθυμητά δεδομένα μέσα σε ένα segment} \\ = PROBE^{idx} + \frac{P \cdot F \cdot N \cdot \sum_{j=1}^r E_j}{2} + P \cdot S_1 \cdot F.$$

Ο tune-in time για $Q \{a_1 \wedge \dots \wedge a_q\}$ εξαρτάται από τον αριθμό των επιπέδων στο δέντρο ευρετηρίου για το a_i , $1 \leq i \leq r$ και την επιλεξιμότητα της ερώτησης σύζευξης. Ξεκινάει μια αναζήτηση για κόμβους για το a_1 στο δέντρο ευρετηρίου. Όταν βρεθεί το επιθυμητό τεμάχιο δεδομένων (data segment) ερευνάται και αυτό για να βρεθεί το δέντρο ευρετηρίου του a_2 . Αφού βρεθεί το δέντρο ευρετηρίου του a_2 στο αρχικό δέντρο ευρετηρίου (index tree) για να βρεθούν τα segment που ταιριάζουν με το a_2 . Η διαδικασία φιλτραρίσματος συνεχίζεται με όλα τα γνωρίσματα (attributes) που υπάρχουν μέσα στην ερώτηση. Στη συνέχεια λαμβάνονται τα επιθυμητά πλαίσια πληροφορίας. Ο tune-in time προκύπτει ως εξής :

$$T^{idx}(a_1 \wedge \dots \wedge a_q) = (I + \sum_{i=1}^l h_i) \cdot N + (I + F \cdot \prod_{i=1}^l S_i) \cdot P$$

όπου $l = \min(r, q)$ και $\prod_{i=1}^l S_i$ είναι ο αριθμός των αντικειμένων για τα οποία βρέθηκε ταίριασμα με τα γνωρίσματα της ερώτησης για $Q \{a_1 \wedge \dots \wedge a_q\}$.

2.2.1.2 Διάζευξη (V) Γνωρισμάτων για Index Tree

Η γενική μέθοδος πρόσβασης για $Q \{a_1 \vee \dots \vee a_q\}$ είναι:

Αλγόριθμος Διάζευξης(V) Γνωρισμάτων για Index Tree

Initial probe: For any index trees built on a_1, \dots, a_q , the client tunes into the broadcast channel and determines when the next index tree nodes for the data frames are broadcast.

Search: For each index tree built on a_1, \dots, a_q , the client follows a list of pointers to find out the arrival time of the desired data frame.

Retrieval: The client tunes into the broadcast channel to download all the qualified frames.

Ο access time για $Q \{a_1 \vee \dots \vee a_q\}$ είναι:

$$A^{idx}(a_1 \vee \dots \vee a_q) = \text{initial probe time} \\ + \text{χρόνος αναμονής για όλα τα επιθυμητά segments δεδομένων} \\ + \text{access time για τα επιθυμητά δεδομένα μέσα σε ένα segment} \\ = PROBE^{idx} + P * F * N * \sum_{j=1}^r E_j$$

Ο tune-in time για $Q \{a_1 \vee \dots \vee a_q\}$ είναι:

$$\text{Εάν } r < q \quad : \quad T^{idx}(a_1 \vee \dots \vee a_q) = r * N + P * F$$

Αλλιώς ($q < r$) :

$$T^{idx}(a_1 \vee \dots \vee a_q) < (1 + \sum_{i=1}^q h_i) * N + P * \sum_{i=1}^q M_i + F * P * \sum_{i=1}^q (-1)^{i+1} S_i^{(q)}$$

όπου ο αριθμός των σωστών ταιριασμάτων (true matches) δεν ξεπερνάει το $P * F * \sum_{i=1}^q (-1)^{i+1} S_i^{(q)}$ και $S = \max \{S_i : 1 < i < q\}$.

2.2.2 Η Τεχνική της Υπογραφής (Signature Scheme)

Στην μέθοδο του δέντρου ευρετηρίου (index tree) υπάρχει δέντρο ευρετηρίου για κάθε γνώρισμα (attribute) που έχει δεικτοδοτηθεί. Έτσι λοιπόν το κόστος αυτής της μεθόδου είναι ανάλογο του αριθμού των δεικτοδοτημένων γνωρισμάτων. Όμως για την τεχνική της υπογραφής, μια υπογραφή μπορεί να παρέχει πληροφορίες ευρετηρίου για όλα τα γνωρίσματα. Προφανώς λοιπόν το κόστος της τεχνικής της υπογραφής δεν εξαρτάται από τον αριθμό των γνωρισμάτων και η μέθοδος πρόσβασης (access method) είναι πολύ πιο απλή από αυτήν του index tree.

P_i	Η πιθανότητα λανθασμένου ταιριάσματος ενός πλαισίου για το α_i
P_f^S	Η πιθανότητα λανθασμένου ταιριάσματος μιας απλής υπογραφής
P_f^I	Η πιθανότητα λανθασμένου ταιριάσματος μιας ενοποιημένης υπογραφής
K	Αριθμός των πλαισίων που έχουν δεικτοδοτηθεί με την ενοποιημένη υπογραφή
l_i	Μέσος αριθμός σωστών ταιριασμάτων σε ένα πλαίσιο πληροφορίας
P	Αριθμός bits σε ένα πακέτο
R_S	Αριθμός πακέτων που παίρνει μια απλή υπογραφή
R_I	Αριθμός πακέτων που παίρνει μια ενοποιημένη υπογραφή
S	Αριθμός bit strings που υπερθέτονται σε μια υπογραφή

Πίνακας 2.3 : Παράμετροι για την τεχνική της Υπογραφής

Στον Πίνακα 2.3 καθορίζονται οι παράμετροι για τα μοντέλα κόστους της τεχνικής της υπογραφής. Για δεικτοδότηση με **πολλαπλά γνωρίσματα (multi-attribute)**, η τεχνική της **πολυεπίπεδης υπογραφής (multi-level signature)**, που έχουμε αναφέρει στο Κεφάλαιο 1, είναι η καλύτερη προσέγγιση και αυτήν επιλέγουμε να μελετήσουμε εδώ. Υποθέτουμε ότι κάθε γνώρισμα δεικτοδοτείται στην υπογραφή ($r=m$). Η πιθανότητα λανθασμένου ταιριάσματος για τις απλές υπογραφές καθώς και για τις ενοποιημένες μπορεί να υπολογιστεί ως εξής :

ΛΗΜΜΑ 2.1 : (Βέλτιστη πιθανότητα λανθασμένου ταιριάσματος (Optimal false drop probability))

Δεδομένου του μεγέθους μιας υπογραφής, R (R_I ή R_S :για ενοποιημένη και απλή αντίστοιχα), και ο αριθμό των bit strings που υπερθέτονται στην υπογραφή s , η πιθανότητα λανθασμένου ταιριάσματος είναι : $P_f^S = P_f^I = 2^{-R(p \cdot \ln 2)/s}$.

Σε γενικά πλαίσια το μέγεθος μιας υπογραφής (R_I ή R_S) είναι πολύ μικρό σε σύγκριση με ένα πλαίσιο δεδομένων. Διαφορετικά ο tune-in time και ο access time θα ήταν πολύ μεγάλοι. Συνεπώς, ο χρόνος για να φιλτράρει ο πελάτης την εναπομείνασα υπογραφή και το εναπομείναν πλαίσιο πληροφορίας, που ονομάζεται initial probe time μπορεί να υπολογιστεί ως μισό του μεγέθους του πλαισίου, $PROBE = P/2$.

Ο tune-in time στην initial probe περίοδο είναι το αναμενόμενο άθροισμα του χρόνου όταν ο πελάτης είναι ενεργός για να φιλτράρει μερικώς μια απλή υπογραφή, μια ενοποιημένη υπογραφή, μια πολυεπίπεδη υπογραφή. Για παρόμοιους λόγους, ο tune-in time για φιλτράρισμα μιας μερικής υπογραφής είναι αμελητέος και ο initial tune-in time μπορεί να εκτιμηθεί στα $P/2$.

Η μεγαλύτερη διαφορά μεταξύ του δέντρου ευρετηρίου (Index Tree) και της τεχνικής της υπογραφής (Signature Scheme) είναι ότι ένας κόμβος ενός index tree είναι χρήσιμος μόνο για ένα δεικτοδοτημένο γνώρισμα (Indexed attribute), ενώ μια υπογραφή περιέχει τις πληροφορίες για όλα τα δεικτοδοτημένα γνωρίσματα.

2.2.2.1 Σύζευξη (\wedge) Γνωρισμάτων για Signature Scheme

Η γενική μέθοδος πρόσβασης για $Q \{a_1 \wedge \dots \wedge a_q\}$ είναι:

Αλγόριθμος Σύζευξης(\wedge) Γνωρισμάτων για Signature Scheme

Initial probe: The client tunes into the broadcast channel for the first received signature

Filtering: For each signature received in a broadcast cycle do the following:

for each query attribute a_i ,

IF the signature does not match the query signature based on attribute a_i ,

THEN repeat the filtering step

Retrieval and checking: The client retrieves frames corresponding to the matched signature for further checking to eliminate false drops. Repeat filtering step.

Ο access time για $Q \{a_1 \wedge \dots \wedge a_q\}$ είναι:

$$\begin{aligned}
 A^{sig}(a_1 \wedge \dots \wedge a_q) &= \text{initial probe time} \\
 &+ \text{χρόνος φιλτραρίσματος για το πρώτο επιθυμητό πλαίσιο} \\
 &+ \text{χρόνος για να βρεθούν όλα τα επιθυμητά πλαίσια} \\
 &= PROBE + CYCLE/2 + S_1 * F \\
 &= P/2 + F * (SIG_I + SIG_S + P) / 2 + P * S_1 * F
 \end{aligned}$$

Ο tune-in time για $Q \{a_1 \wedge \dots \wedge a_q\}$ είναι:

$$\begin{aligned}
 T^{sig}(a_1 \wedge \dots \wedge a_q) &= \text{ο tune-in time στην initial probe περίοδο} \\
 &+ \text{σωστό ταιριασμα πλαισίων στην εκπομπή} \\
 &+ \text{κάθε ενοποιημένη υπογραφή σε μισό κύκλο εκπομπής} \\
 &+ \text{απλές υπογραφές σχετιζόμενες με συμβατές ενοποιημένες} \\
 &\text{υπογραφές} + \text{πιθανότητα λανθασμένου ταιριασματος (false} \\
 &\text{drop) σε μισό κύκλο εκπομπής} \\
 &= \frac{P}{2} + P * F * \prod_{i=1}^q S_i + \frac{F}{2} * (R_I/k + [S_1/l_1] * k * R_S) \\
 &+ \text{πιθανότητα λανθασμένου ταιριασματος (false drop) σε μισό} \\
 &\text{κύκλο εκπομπής} \\
 &\leq \frac{P}{2} + P * F * \prod_{i=1}^q S_i + \frac{F}{2} * (R_I/k + [S_1/l_1] * k * R_S) \\
 &+ P * F * \sum_{i=1}^q (q * S^{i-1} * D^{q+1-i}) / 2
 \end{aligned}$$

όπου ο αριθμός των σωστών ταιριασμάτων εκτιμάται στο $F * \prod_{i=1}^q S_i$.
 $S = \max \{S_i; 1 < i < q\}$

2.2.2.2 Διάζευξη (V) Γνωρισμάτων για Signature Scheme

Η γενική μέθοδος πρόσβασης για $Q \{a_1 \vee \dots \vee a_q\}$ είναι:

Αλγόριθμος Διάζευξης(V) Γνωρισμάτων για Signature Scheme

Initial probe: The client tunes into the broadcast channel for the frame signature in a broadcast cycle.

Filtering: The client tunes into retrieve the signatures arriving and matches the frame signatures with the signatures for any attribute values in a_1, \dots, a_q .

Retrieval and checking: For the matched signatures, the client tunes into get the corresponding data frames from the channel for further checking to eliminate false drops.

Ο access time για $Q \{a_1 \vee \dots \vee a_q\}$ είναι:

$$\begin{aligned}
 A^{sig}(a_1 \vee \dots \vee a_q) &= \text{initial probe time} \\
 &\quad + \text{έννας κύκλος εκπομπής} \\
 &\quad + \text{access time για τα επιθυμητά δεδομένα μέσα σε ένα segment} \\
 &= P/2 + F * SIG_I + SIG_S + P
 \end{aligned}$$

Ο tune-in time για $Q \{a_1 \vee \dots \vee a_q\}$ έχει άνω όριο:

$$\begin{aligned}
 T^{sig}(a_1 \vee \dots \vee a_q) &= \text{o tune-in time στην initial probe περίοδο} \\
 &\quad + \text{σωστό ταιρίασμα πλαισίων σε κάθε κύκλο εκπομπής} \\
 &\quad + \text{κάθε ενοποιημένη υπογραφή σε ένα κύκλο εκπομπής} \\
 &\quad + \text{απλές υπογραφές σχετιζόμενες με συμβατές ενοποιημένες} \\
 &\quad \text{υπογραφές + πιθανότητα λανθασμένου ταιριάσματος (false} \\
 &\quad \text{drop) σε ένα κύκλο εκπομπής} \\
 &\leq \frac{P}{2} + F * P * \sum_{i=1}^q (-1)^{i+1} S^i \binom{q}{i} \\
 &\quad + F * (R_I/k + [S_i/l_i] * k * R_S) \\
 &\quad + \text{πιθανότητα λανθασμένου ταιριάσματος (false drop) σε ένα} \\
 &\quad \text{κύκλο εκπομπής} \\
 &\leq \frac{P}{2} + F * P * \sum_{i=1}^q (-1)^{i+1} S^i \binom{q}{i} \\
 &\quad + F * (R_I/k + [S_i/l_i] * k * R_S) \\
 &\quad + P/2 + F * P * \sum_{i=1}^q ((1 - S_i) * P_i)
 \end{aligned}$$

Όπου ο αριθμός των σωστών ταιριασμάτων δεν ξεπερνά $P * F * \sum_{i=1}^q (-1)^{i+1} S^i \binom{q}{i}$ και $S = \max \{S_i; 1 < i < q\}$.

2.2.3 Η υβριδική τεχνική (Hybrid Method)

Η υβριδική τεχνική αποτελείται από δύο μέρη : το αραιό δέντρο ευρετηρίου και την υπογραφή. Το αραιό δέντρο ευρετηρίου στηρίζεται στο **σημαντικότερο γνώρισμα (major attribute)** και χρησιμοποιείται για γενικό φιλτράρισμα. Η πολυεπίπεδη υπογραφή χρησιμοποιείται για να μεταφέρει το τοπικό φιλτράρισμα που βασίζεται σε τιμές γνωρισμάτων, καθορισμένες στις ερωτήσεις. Με αυτό τον τρόπο το δέντρο ευρετηρίου βοηθάει στην βελτίωση του tune-in time του πελάτη. Ο μέσος χρόνος αναμονής για την παραλαβή ενός πλαισίου πληροφορίας από τον κύκλο εκπομπής μπορεί να εκφραστεί ως εξής :

$$TREE+SIG+P=N*(X/t_1 + 1)/F+R(1+1/k)+P$$

Ο μέσος initial probe time για την υπογραφή και το μήκος ενός κύκλου δίνεται από :

$$PROBE^{hyb} = (TREE + SIG + P)/2$$

$$CYCLE^{hyb} = (TREE + SIG + P) * F.$$

2.2.3.1 Σύζευξη (\wedge) Γνωρισμάτων για Hybrid Method

Η γενική μέθοδος πρόσβασης για $Q \{a_1 \wedge \dots \wedge a_q\}$ είναι:

Αλγόριθμος Σύζευξης(\wedge) Γνωρισμάτων για Hybrid Method

If a_1 is the major attribute

- based on the single index tree access protocol to retrieve the frame block which contains all the frames satisfying the value for a_1
- successive signature matches based on $\{a_2, \dots, a_q\}$ are conducted to filter out the desired frames

Else based on the signature access method to retrieve the qualified frames based on $\{a_1, \dots, a_q\}$. The sparse index tree is skipped.

Ο access time για $Q \{a_1 \wedge \dots \wedge a_q\}$ είναι:

$$\begin{aligned} A^{hyb}(a_1 \wedge \dots \wedge a_q) &= PROBE^{hyb} + \frac{CYCLE^{hyb}}{2} + F * S_1 * P \\ &= (TREE+SIG+P)*(1+F)/2+F*S_1 * P \end{aligned}$$

Ο tune-in time για $Q \{a_1 \wedge \dots \wedge a_q\}$ έχει άνω όριο:

$$T^{hyb}(a_1 \wedge \dots \wedge a_q) = (I+t_1) * N + T(B) + (I+F * \prod_{j=1}^q S_j) * P$$

όπου $T(B)$ είναι ο tune-in time για φιλτράρισμα του μπλοκ δεδομένων B με την τεχνική της υπογραφής, η οποία υπολογίζεται ως εξής :

$$T(B) = \text{κάθε υπογραφή στο μπλοκ δεδομένων } B \\ + \text{πιθανότητα λανθασμένου ταιριάσματος στο μπλοκ} \\ \text{δεδομένων } B$$

2.2.3.2 Διάζευξη (\vee) Γνωρισμάτων για Hybrid Method

Η γενική μέθοδος πρόσβασης για $Q \{a_1 \vee \dots \vee a_q\}$ είναι ίδια με αυτήν της πολυεπίπεδης υπογραφής:

$$A^{hyb}(a_1 \vee \dots \vee a_q) = PROBE^{hyb} + CYCLE^{hyb} + F * S_1 * P \\ = (TREE+SIG+P) * (1/2+F)$$

Ο tune-in time για $Q \{a_1 \vee \dots \vee a_q\}$ είναι:

$$T^{hyb}(a_1 \vee \dots \vee a_q) = N + T^{sig}(a_1 \vee \dots \vee a_q)$$

όπου $T^{sig}(a_1 \vee \dots \vee a_q)$ αντιπροσωπεύει τον tune-in time για την αντίστοιχη τεχνική της υπογραφής.

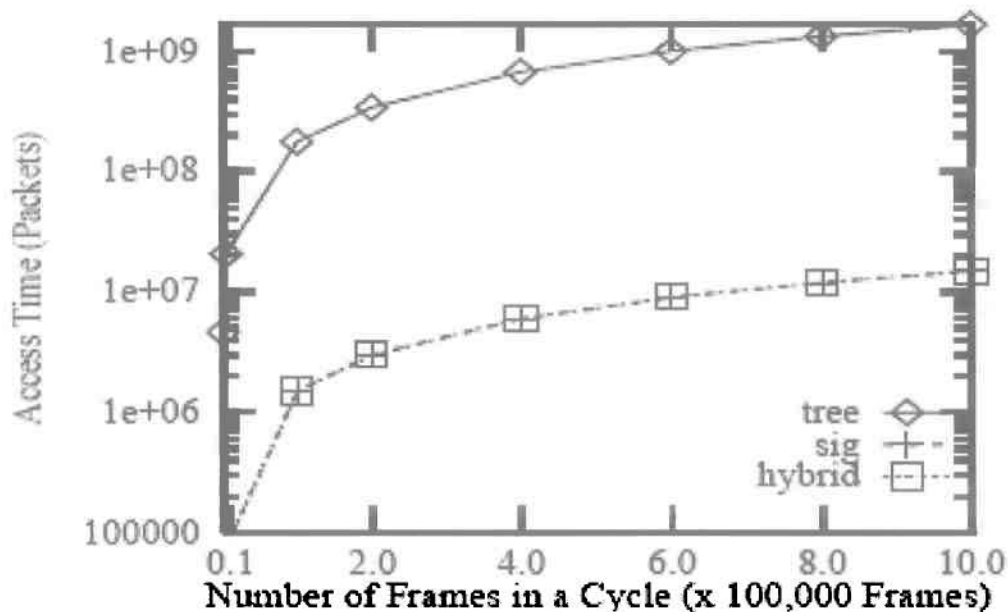
2.3 Ανάλυση Απόδοσης

Στη συνέχεια παραθέτονται μερικές γραφικές παραστάσεις που απεικονίζουν τη σχέση κάθε προαναφερθείσας μεθόδου με τον tune-in time, access time, την επιλεξιμότητα της ερώτησης (query selectivity) και τον αριθμό των γνωρισμάτων της ερώτησης για την σύζευξη γνωρισμάτων αλλά και την διάζευξη αυτών. Οι παρακάτω γραφικές παραστάσεις θα μας δώσουν τις ομοιότητες και διαφορές στην συμπεριφορά των τριών μεθόδων.

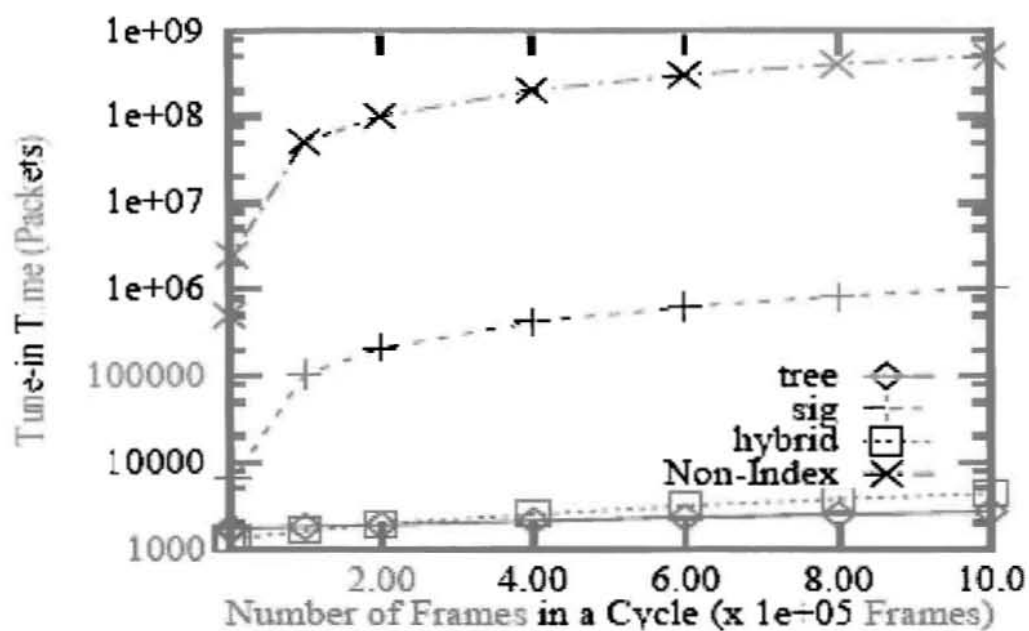
2.3.1 Απόδοση Ερωτήσεων Σύζευξης (\wedge)

$F = 10^4 \sim 10^6$	$P = 10^3$	$S = 0.01$	$p = 64$	$m = 10$
$n = 128$	$N = 100$	$k = 4$	$r = 6$	$q = 3$

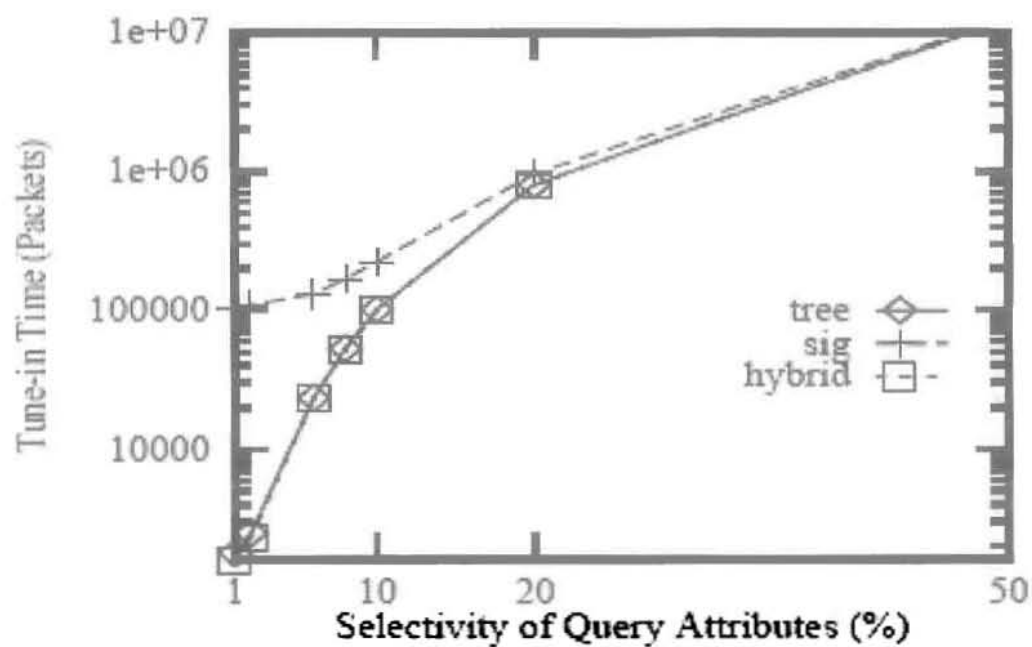
Πίνακας 2.4 : Παράμετροι για τα μοντέλα κόστους



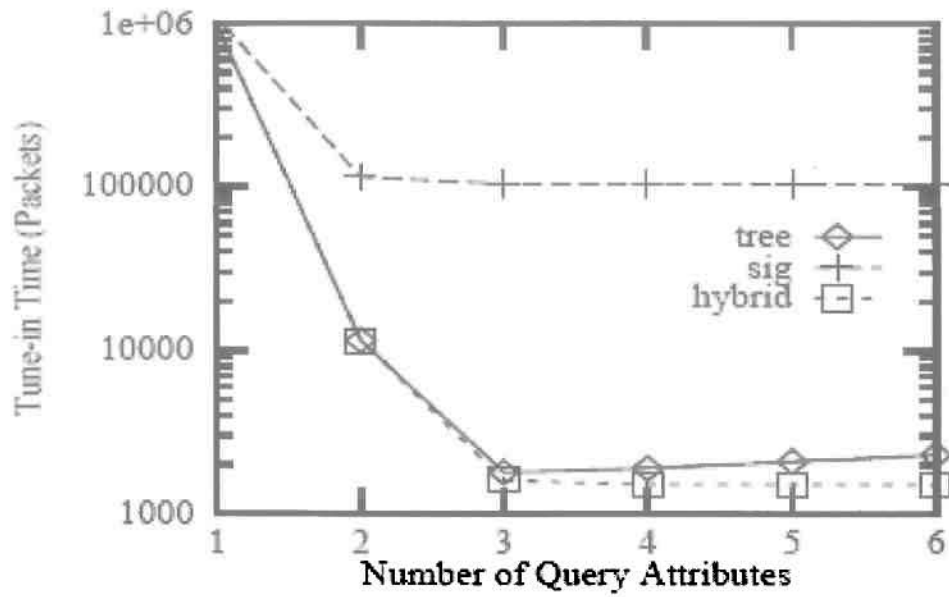
Σχήμα 2.1 : Access time για ερωτήσεις σύζευξης (\wedge)



Σχήμα 2.2 : Tune-in time για ερωτήσεις σύζευξης (Λ)

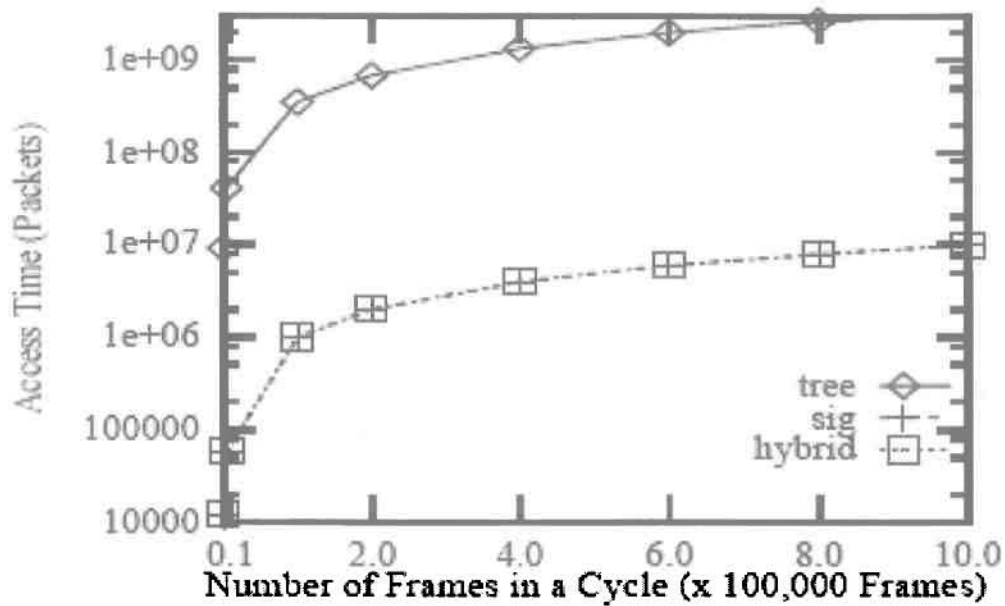


Σχήμα 2.3 : Tune-in time σε σχέση με την επιλεξιμότητα της ερώτησης

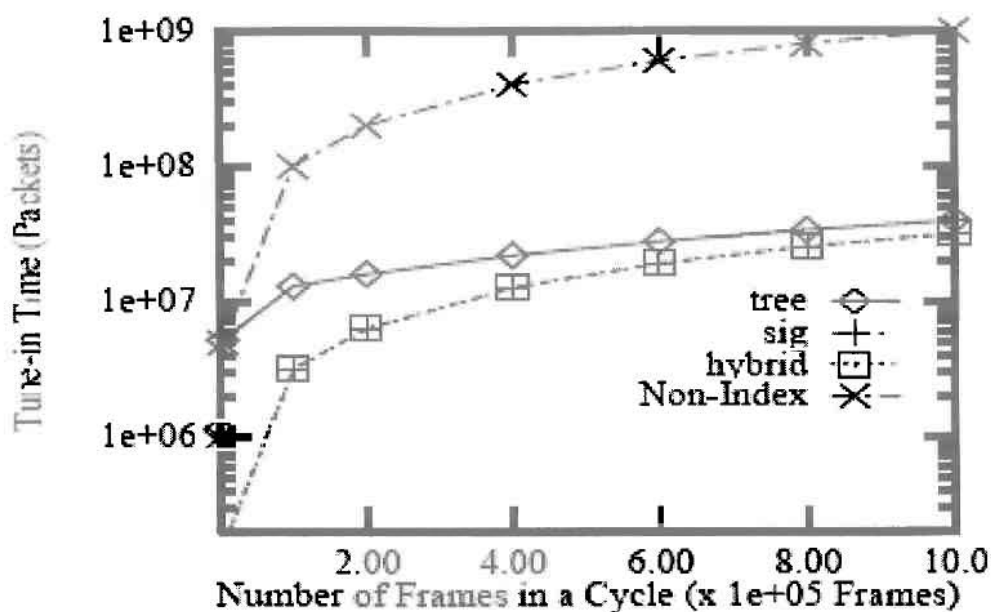


Σχήμα 2.4 : Tune-in time σε σχέση με τον αριθμό των γνωρισμάτων στην ερώτηση

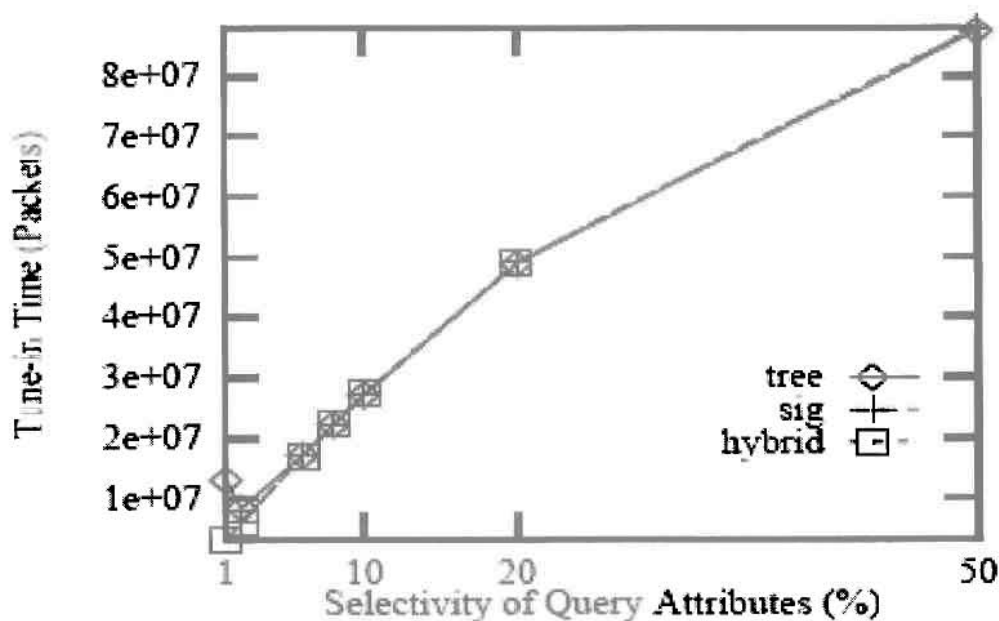
2.3.2 Απόδοση Ερωτήσεων Διάζευξης (V)



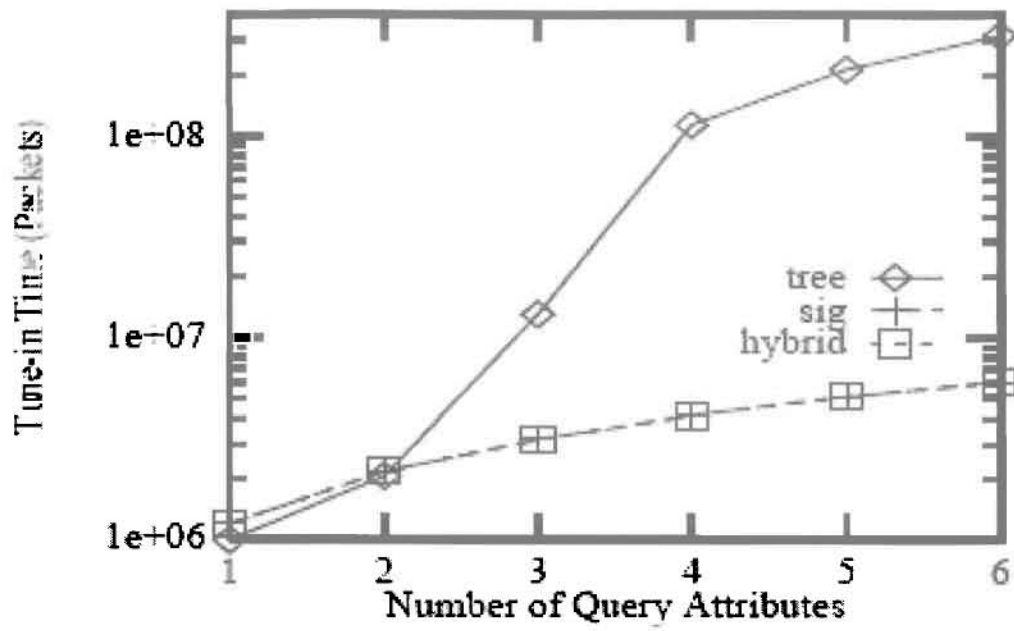
Σχήμα 2.5 : Access time για ερωτήσεις διάζευξης (V)



Σχήμα 2.6 : Tune-in time για ερωτήσεις διάζευξης (V)



Σχήμα 2.7 : Tune-in time σε σχέση την επιλεκτικότητα της ερώτησης



Σχήμα 2.8 : Tune-in time σε σχέση με τον αριθμό των γνωρισμάτων στην ερώτηση

ΜΕΡΟΣ II

Ομότιμα Δίκτυα (Peer-To-Peer (P2P) Networks)

ΜΕΡΟΣ II Ομότιμα Δίκτυα (Peer-To-Peer (P2P) Networks)

Εισαγωγικά.....	55
-----------------	----

ΚΕΦΑΛΑΙΟ 3

Υπογραφές γειτνίασης για αναζήτηση σε ομότιμα (P2P) δίκτυα

Περιεχόμενα

3.1 Εισαγωγή.....	55
3.2 Ομότιμο (P2P) σύστημα.....	57
3.3 Αναζήτηση σε (P2P) συστήματα.....	58
3.4 Υπογραφές Γειτόνων (Neighborhood signatures).....	59
3.4.1 Ολοκληρωμένη –γειτονική υπογραφή (complete-neighborhood signature (CN)).....	59
3.4.2 Μερική-γειτονική υπογραφή (partial-neighborhood signature (PN)).....	60
3.4.2.1 Μερική-γειτονική υπογραφή με υπέρθεση (partial-neighborhood superimposed signature (PN-S)).....	60
3.4.2.2 Μερική-γειτονική υπογραφή με προσάρτηση (partial-neighborhood appended signature (PN-A)).....	60
3.5 Αλγόριθμοι Αναζήτησης.....	61
3.5.1 Αναζήτηση Πλημμύρας (Flooding Search).....	61
3.5.2 Αναζήτηση Μοναδικού Μονοπατιού (Single Path Search).....	63
3.6 Κατασκευή και Διατήρηση Υπογραφών.....	63
3.7 Ανάλυση Απόδοσης.....	65

ΜΕΡΟΣ II Ομότιμα Δίκτυα (Peer-To-Peer (P2P) Networks)

Τα ομότιμα δίκτυα έχουν προσελκύσει τελευταία έντονο ενδιαφέρον εξαιτίας της ευρείας χρήσης ομότιμων (peer-to-peer) εφαρμογών όπως SETI, Napster, Gnutella και Morpheus. Λόγω των πολλών αντιγράφων στους αναρίθμητους ομότιμους χρήστες, τα ψηφιακά δεδομένα μπορούν να διαμοιραστούν ή ανταλλαχθούν με μεγάλη ανεκτικότητα και διαθεσιμότητα.

Παρόλα αυτά όμως οι υπάρχουσες peer-to peer εφαρμογές παρουσιάζουν υπερβολικό φόρτο στην κίνηση του δικτύου. Έτσι λοιπόν στο **Κεφάλαιο 1** θα αναλύσουμε θέματα διαχείρισης του αποθηκευτικού χώρου στους ομότιμους σταθμούς για να μειωθεί η υπερφόρτωση του δικτύου. Θα προτείνουμε την χρήση υπογραφών για κατευθυνόμενη αναζήτηση κατά μήκος επιλεγμένων δικτυακών μονοπατιών και συγκεκριμένα θα αναλυθούν τρεις μέθοδοι : **ολοκληρωμένη – γειτονική υπογραφή (complete-neighborhood signature(CN))**, **μερική-γειτονική υπογραφή με υπέρθεση (partial-neighborhood superimposed signature (PN-S))** και **μερική-γειτονική υπογραφή με προσάρτηση (partial-neighborhood appended signature (PN-A))** για να γίνει πιο αποτελεσματική αναζήτηση των κοινής χρήσης δεδομένων στα ομότιμα (P2P) δίκτυα.

Όλο και περισσότερες εφαρμογές χρειάζονται P2P συστήματα για να υποστηρίξουν πολύπλοκες ερωτήσεις (queries) σε πολυδιάστατα δεδομένα. Ένα χαρακτηριστικό παράδειγμα είναι οι ομότιμες διαδικτυακές δημοπρασίες ακίνητης περιουσίας όπου πολύ συχνά χρειάζεται να απαντηθούν ερωτήσεις του τύπου "βρες πέντε διαθέσιμα κτήρια που βρίσκονται κοντά στο κέντρο της πόλης". Τέτοιου είδους ερωτήσεις όμως δεν υποστηρίζονται εύκολα από τα ήδη υπάρχοντα ομότιμα συστήματα. Στο **Κεφάλαιο 2** προτείνεται ένα λειτουργικό σχήμα διαμοιρασμού, ευρετηρίου-δεικτοδότησης, και απάντησης ερωτήσεων σε πολυδιάστατα δεδομένα όπου (i) οι ομότιμοι σταθμοί με μεγαλύτερη υπολογιστική δύναμη συντονίζουν την δεικτοδότηση και την επεξεργασία των ερωτήσεων και (ii) οι υπόλοιποι ομότιμοι σταθμοί συμμετέχουν ως ένα βαθμό στον υπολογισμό για να επιτευχθεί μεγαλύτερη κλιμάκωση και να υπάρξει μια ισορροπία στο φόρτο επεξεργασίας. Έτσι προτείνονται οι εξής μέθοδοι : **Network-R-tree (NR-tree)**, μια **P2P προσαρμογή του υπάρχοντος spatial index-R*-tree, NR-tree**. Η δεικτοδότηση δεδομένων σε ομαδοποιημένους ομότιμους σταθμούς κάνει πιο εύκολη την επεξεργασία πολύπλοκων ερωτήσεων. Στη συνέχεια προτείνονται και αλγόριθμοι επεξεργασίας ερωτήσεων για ερωτήσεις αλληλουχίας (range queries) και k-κοντινότερου γείτονα (k-NN neighbor) και αποδεικνύεται η λειτουργικότητα τους με πραγματικά δεδομένα.

ΚΕΦΑΛΑΙΟ 3 Υπογραφές γειτνίασης για αναζήτηση σε ομότιμα (P2P) δίκτυα

3.1 Εισαγωγή

Εφαρμογές όπως οι Napster και Gnutella έκαναν το διαδίκτυο ένα πολύ δημοφιλές μέσο για την εξάπλωση της διάδοσης των πόρων και των δεδομένων μεταξύ χιλιάδων χρηστών. Σε αντίθεση με το παραδοσιακό μοντέλο πελάτη-εξυπηρετητή (client-server), αυτά τα ομότιμα (P2P) συστήματα μπορούν να χρησιμοποιήσουν τους δικούς τους κόμβους ως εξυπηρετητές (servers) για τους άλλους κόμβους. Παρά το ότι θέλουμε να αποφεύγουμε την περίπτωση ενός κεντρικού εξυπηρετητή, ο οποίος επιβαρύνεται από όλο τον φόρτο (bottleneck) και αποτελεί **μοναδικό σημείο βλάβης (single point of failure)**, τα P2P συστήματα παρουσιάζουν ενδιαφέρουσες προοπτικές στον εντοπισμό των δεδομένων ανάμεσα στους αναρίθμητους κόμβους.

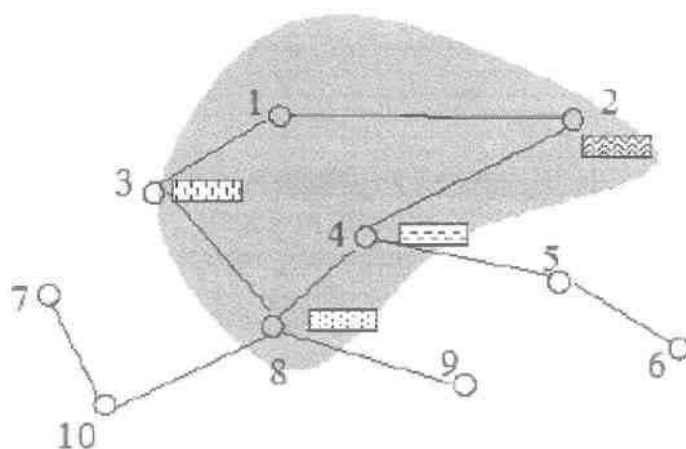
Έχουν προταθεί δύο βασικές στρατηγικές για την αναζήτηση σε αποκεντρωμένα και δυναμικά περιβάλλοντα των P2P συστημάτων χωρίς να εξαρτώνται από την τοπολογία του δικτύου και την τοποθέτηση των δεδομένων.

- **1^η Στρατηγική** : με αυτή τη στρατηγική τα μηνύματα διαδίδονται σε κάθε κόμβο χωρίς να υπάρχει καμία γνώση για το που βρίσκονται τα δεδομένα μέχρι αυτά να βρεθούν. Οι Gnutella και Random walk εφαρμογές χρησιμοποιούν αυτή τη στρατηγική. Το αρνητικό αυτής της στρατηγικής είναι η πιθανότητα της υπερφόρτωσης του δικτύου εξαιτίας του μεγάλου αριθμού των μηνυμάτων αναζήτησης (όπως στη Gnutella) ή της μεγάλης καθυστέρησης (latency) να ικανοποιηθεί μια αίτηση (όπως στο Random walk).
- **2^η Στρατηγική** : αυτή η στρατηγική διατηρεί επιπλέον πληροφορίες στους δικτυακούς κόμβους (κάτι που δεν κάνει η 1^η στρατηγική), έτσι ώστε να μειώσει την δικτυακή συμφόρηση και/ή των αριθμό των επισκέψεων στους κόμβους. Συνεπώς τα μηνύματα κατευθύνονται σε συγκεκριμένα μονοπάτια που θεωρούνται ότι θα είναι πιο παραγωγικά. Οι επιπλέον πληροφορίες περιέχονται στα δεδομένα που έχουν ομαδοποιηθεί ιεραρχικά ή σε κοντινούς γείτονες. Αυτή η προσέγγιση απαιτεί να καθοριστεί ποια αρχικά γνωρίσματα (attributes) θα χρησιμοποιηθούν ώστε να γίνει πιο αποτελεσματική η αναζήτηση και παράλληλα να είναι πιο μικρό το κόστος της αποθήκευσης αυτών.

Με βάση την 2^η στρατηγική υπάρχουν προσεγγίσεις που χρησιμοποιούν **αρχεία υπογραφών (signature files)** και μπορούν να παρέχουν καλύτερη κίνηση των μηνυμάτων με χαμηλότερο κόστος αποθήκευσης. Οι μέθοδοι υπογραφών έχουν χρησιμοποιηθεί ευρέως στην εξόρυξη γνώσης, σε βάσεις δεδομένων με εικόνες, σε βάσεις δεδομένων με οπτικοακουστικό περιεχόμενο και άλλες βάσεις δεδομένων. Όπως εξηγήθηκε και στο προηγούμενο κεφάλαιο ουσιαστικά η υπογραφή αποτελεί μια **αφαίρεση (abstract)** των πληροφοριών που είναι αποθηκευμένες σε ένα αρχείο. Έτσι απλά εξετάζοντας μια υπογραφή μπορούμε να εκτιμήσουμε αν το αρχείο περιέχει την επιθυμητή για εμάς πληροφορία. Συνεπώς η τεχνική της υπογραφής είναι ιδιαίτερως χρήσιμη για φιλτράρισμα πληροφοριών αποθηκευμένων σε κόμβους ομότιμων (P2P) συστημάτων.

3.2 Ομότιμο (P2P) σύστημα

Ένα ομότιμο (P2P) σύστημα αποτελείται από αναρίθμητους κόμβους (ονομάζονται **peers**), οι οποίοι συνδέονται μεταξύ τους άμεσα ή έμμεσα. Κάθε **κόμβος (peer)** παρέχει πληροφορίες που διαμοιράζεται με τους άλλους κόμβους. αυτές οι πληροφορίες μπορεί να είναι ψηφιακά αρχεία όπως κλιπάκια μουσικής, εικόνες, pdf αρχεία ή οποιαδήποτε άλλη μορφή ψηφιακής πληροφορίας. Οι κόμβοι έχουν μια λογική σύνδεση μεταξύ τους, κάτι τι οποίο σημαίνει ότι η τοπολογία αλλάζει δυναμικά κάθε φορά που **προστίθεται ένας κόμβος (peer join)** ή **αφαιρείται (peer leave)** από το δίκτυο. Επιπλέον, ακόμα και οι πληροφορίες για κοινή χρήση αλλάζει δυναμικά μιας και οι κόμβοι μπορεί να **ενημερώσουν (peer update)** το ψηφιακό περιεχόμενό τους.



Σχήμα 3.1 : Μέρος ενός ομότιμου (P2P) δικτύου

Στο Σχήμα 3.1 βλέπουμε ένα μερικό στιγμιότυπο ενός P2P δικτύου. Εδώ χρησιμοποιείται ένα διάγραμμα που αντιπροσωπεύει τον **κόμβο (peer)** και μια κορυφή που δηλώνει την σύνδεση μεταξύ δύο κόμβων (peers). Όταν ο κόμβος (peer) A συνδέεται κατευθείαν με τον κόμβο (peer) B, ονομάζουμε αυτούς τους δύο γείτονες (neighbors). Όμως σε ένα δίκτυο δύο κόμβοι (peers) μπορούν να επικοινωνούν μέσω ενός ή περισσότερων συνδέσεων που ονομάζονται **μονοπάτια (paths)**. Το **μήκος ενός μονοπατιού (path length)** μπορεί να υπολογιστεί μετρώντας τον **αριθμό των συνδέσεων (hops of connections)**. Η απόσταση μεταξύ δύο κόμβων (peers) είναι το μικρότερο μήκος μονοπατιού (path length). Για παράδειγμα για τους κόμβους (peers) 1 και 8 υπάρχουν δύο μονοπάτια που τους ενώνουν, το ένα με κόστος 2 μέσω του κόμβου 3 και το άλλο με κόστος 3 μέσω του κόμβου 2 και 4. Συνεπώς, η απόσταση των κόμβων (peers) 1 και 8 είναι 2. Στην συνέχεια της μελέτης έκτος από την έννοια των γειτόνων θα συναντήσουμε και την έννοια της γειτονιάς. Γειτονιά θα θεωρούμε όλους εκείνους τους κόμβους που θα βρίσκονται εντός μιας συγκεκριμένης ακτίνας. Παράδειγμα στο Σχήμα 3.1 η σκιασμένη περιοχή αντιπροσωπεύει μια γειτονιά ακτίνας 2 (περιλαμβάνει τους κόμβους 2,3,4,8) για τον κόμβο 1.

3.3 Αναζήτηση σε (P2P) συστήματα

Όπως αναφέρθηκε και προηγουμένως τα P2P δίκτυα ευρέως χρησιμοποιούνται για τον διαμοιρασμό και κοινή χρήση ψηφιακού υλικού. Συνεπώς είναι πολύ σημαντικό να γίνεται αποτελεσματική αναζήτηση του ψηφιακού υλικού για την καλύτερη εξυπηρέτηση των κόμβων (peers). Ένας χρήστης μπορεί να ξεκινήσει μια αναζήτηση από οποιονδήποτε κόμβο στο δίκτυο. Το **μήνυμα αναζήτησης (search message)** προωθείται σε όλα ή σε ένα μέρος των γειτόνων. Για να αποφεύγεται μια ατέρμονη αναζήτηση συνήθως ορίζεται μια συνθήκη τερματισμού σε κάθε ερώτηση (query). Στη συνέχεια θα δούμε δύο συνθήκες που χρησιμοποιούνται συχνά για να μην έχουμε τέτοια φαινόμενα.

- **Μέγιστο βάθος αναζήτησης (Maximum search depth)** : χρησιμοποιείται στην Gnutella. Έχει οριστεί εξαρχής η **TTL** τιμή (αριθμός hops-κόμβων που επισκέπτεται το μήνυμα) και περιέχεται στο μήνυμα αναζήτησης (search message) ώστε να γνωρίζει πότε να σταματήσει. Κάθε φορά που ένα μήνυμα προωθείται σε ένα γείτονα η τιμή TTL μειώνεται κατά 1. Όταν η τιμή TTL φτάσει το 0 το μήνυμα αναζήτησης σταματάει να μεταδίδεται.
- **Ελάχιστος αριθμός αποτελεσμάτων (Minimum number of results)** : χρησιμοποιείται στην εφαρμογή random walk. Σε αυτήν την συνθήκη χρησιμοποιείται ο **TNR** (συνολικός αριθμός των αποτελεσμάτων που έχουν βρεθεί μέχρι στιγμής), ο οποίος περιέχεται στο **μήνυμα αναζήτησης (search message)**. Κάθε φορά που βρίσκεται ένα αποτέλεσμα ο **TNR** αυξάνεται κατά 1. Η αναζήτηση σταματάει όταν ο **TNR** φτάσει την τιμή που έχει οριστεί εκ των προτέρων.

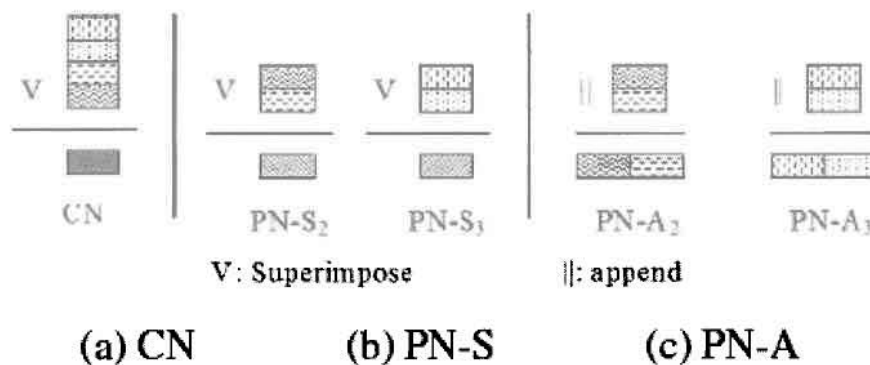
Επειδή στην μελέτη μας πολύ σημαντικό ρόλο παίζει να μειωθεί η δικτυακή συμφόρηση μέσω της πιο αποτελεσματικής αποθήκευσης των δεδομένων χρησιμοποιούμε τον **συνολικό όγκο του μηνύματος (total message volume)**. Για τις μεθόδους υπογραφών όπως και ευρετηρίου, εκτός από την συμφόρηση λόγω αναζήτησης, επιπλέον συμφόρηση προκύπτει λόγω του κόστους κατασκευής και διατήρησης των βοηθητικών πληροφοριών στους κόμβους. Για να μπορούμε λοιπόν να έχουμε δίκαιη σύγκριση μεταξύ των διαφόρων προσεγγίσεων χρησιμοποιούμε **συνολικό όγκο του μηνύματος (total message volume)** (συνολικός αριθμός μηνυμάτων και το μέγεθος διαφορετικών μηνυμάτων συμπεριλαμβανομένων μηνυμάτων αναζήτησης και αυτών που χρησιμοποιούνται όταν έχουμε **έναν κόμβο να προστίθεται/να φεύγει/να ενημερώνεται(peer join/leave/update)**).

3.4 Υπογραφές Γειτόνων (Neighborhood signatures)

Πριν ακόμα παρουσιάσουμε τις μεθόδους υπογραφών για ομότιμα (p2p) συστήματα, θα θεωρήσουμε ότι μια **τοπική υπογραφή (local signature)** δημιουργείται για κάθε κόμβο (peer) στο δίκτυο για να δεικτοδοτήσει (index) την τοπική πληροφορία που είναι διαθέσιμη σε κάθε κόμβο (peer). Με αυτόν τον τρόπο γίνεται πιο αποτελεσματική η αναζήτηση. Επιπλέον κάθε κόμβος (peer) μπορεί να συλλέξει και να διατηρήσει βοηθητικές πληροφορίες που αφορούν ψηφιακό περιεχόμενο διαθέσιμο στους γειτονικούς κόμβους. Έτσι ένας κόμβος (peer) μπορεί να φιλτράρει μηνύματα αναζήτησης τα οποία δεν μπορούν να ικανοποιηθούν πριν τα προωθήσει στους γείτονες του. Βασιζόμενοι σε αυτήν την ιδέα προτείνονται παρακάτω τρεις μέθοδοι υπογραφής.

3.4.1 Ολοκληρωμένη –γειτονική υπογραφή (complete-neighborhood signature (CN))

Η μέθοδος της ολοκληρωμένης-γειτονικής υπογραφής (complete-neighborhood signature (CN)) ουσιαστικά βασίζεται στο ότι μια CN υπογραφή προκύπτει υπερθέτοντας όλες τις τοπικές (local) υπογραφές που βρίσκονται στην γειτονιά ενός κόμβου (peer). Στο Σχήμα 3.1 βλέπουμε τις τοπικές υπογραφές των κόμβων 2,3,4 και 8 που αντιπροσωπεύονται από τα παραλληλόγραμμα με διαφορετικά μοτίβα.



Σχήμα 3.2 : Απεικόνιση της κατασκευής των γειτονικών υπογραφών :
(a) CN (b) PN-S (c) PN-A
(b)

Το Σχήμα 3.2 δείχνει ένα παράδειγμα μιας ολοκληρωμένης-γειτονικής υπογραφής (complete-neighborhood signature (CN)) για τον κόμβο (peer) 1, ο οποίος συμπεριλαμβάνει όλο το διαθέσιμο περιεχόμενο στους κόμβους 2, 3, 4 και 8. Έχοντας μια ολοκληρωμένη-γειτονική υπογραφή (complete-neighborhood signature (CN)), ένας κόμβος μπορεί να αποφασίσει αν η αναζήτηση θα πρέπει να επεκταθεί στη γειτονιά του ή θα πρέπει να προωθηθεί σε κόμβους εκτός της γειτονιάς του που μπορεί να περιέχουν την επιθυμητή πληροφορία.

3.4.2 Μερική-γειτονική υπογραφή (partial-neighborhood signature (PN))

Η CN μέθοδος υπογραφής που εξηγήθηκε προηγουμένως έχει το πλεονέκτημα να βγαίνει εκτός των ορίων μιας γειτονιάς αν δεν υπάρχει ταίριασμα της υπογραφής αναζήτησης και των υπογραφών της γειτονιάς και στην αντίθετη περίπτωση αν βρεθεί ταίριασμα της υπογραφής αναζήτησης και των υπογραφών της γειτονιάς πρέπει να προωθήσει το μήνυμα αναζήτησης σε όλους τους κόμβους μιας γειτονιάς. Στην μερική-γειτονική υπογραφή η ιδέα είναι ότι περιλαμβάνει την πληροφορία ενός από τους άμεσα γειτονικούς κόμβους κάθε κόμβου και όχι όλων των γειτόνων ενός κόμβου. Κάθε κόμβος έχει μια μερική-γειτονική υπογραφή (partial-neighborhood signature). Η αναζήτηση θα επεκταθεί μόνο στους γείτονες, των οποίων οι μερικές-γειτονικές υπογραφές ταιριάζουν με την υπογραφή της αναζήτησης. Παρακάτω θα δούμε τις δύο εναλλακτικές που έχουμε για την μερική-γειτονική υπογραφή (partial-neighborhood signature (PN)).

3.4.2.1 Μερική-γειτονική υπογραφή με υπέρθθεση (partial-neighborhood superimposed signature (PN-S))

Σε αυτήν την προσέγγιση όλες οι τοπικές (local) υπογραφές που βρίσκονται σε μια γειτονιά συμπύσσονται σε μια υπογραφή, που ονομάζεται PN-S υπογραφή. Στο Σχήμα 3.2 (b) βλέπουμε ότι ο κόμβος 1 έχει 2 PN-S υπογραφές, όπου PN-S₂, η γειτονική υπογραφή για το κλαδί 2 περιλαμβάνει όλο το περιεχόμενο των κόμβων 2 και 4 και PN-S₃, η γειτονική υπογραφή για το κλαδί 3 που περιλαμβάνει όλο το περιεχόμενο των κόμβων 3 και 8.

3.4.2.2 Μερική-γειτονική υπογραφή με προσάρτηση (partial-neighborhood appended signature (PN-A))

Στην προηγούμενη μέθοδο PN-S μπορεί να τύχει να χάσουμε κάποια δεδομένα. Όταν η PN-S υπογραφή ταιριάζει σε έναν κόμβο, δεν μας δίνονται επαρκή στοιχεία για το ποιοι είναι οι κόμβοι που πρέπει να επισκεφτούν, οπότε γίνεται η αναζήτηση σε όλους τους κόμβους-γείτονες. Μια εναλλακτική που προτείνεται ονομάζεται PN-A υπογραφή και ουσιαστικά προσαρτά όλες τις τοπικές υπογραφές σε ένα κλαδί της γειτονιάς σε μια μερική υπογραφή. Όταν μια υπογραφή αναζήτησης ταιριάζει με κάποιες υπό-υπογραφές σε μια PN-A υπογραφή, το μήνυμα αναζήτησης θα προωθηθεί μόνο σε αυτούς τους κόμβους που σχετίζονται με τις ταιριαστές υπό-υπογραφές. Στο Σχήμα 3.2(c) βλέπουμε ότι οι κόμβοι 1 και 2 έχουν PN-A υπογραφές, όπου PN-A₂ περιλαμβάνει όλο το περιεχόμενο διαθέσιμο στους κόμβους 2 και 4 και PN-A₃ περιλαμβάνει όλο το περιεχόμενο διαθέσιμο στους κόμβους 3 και 8.

3.5 Αλγόριθμοι Αναζήτησης

Για να μπορέσουμε να συγκρίνουμε τις μεθόδους υπογραφών θα δούμε παρακάτω δύο μεθόδους αναζήτησης : **πλημμύρα/μέγιστο βάθος αναζήτησης (flooding/maximum depth)** και **μοναδικό μονοπάτι/ελάχιστος αριθμός αποτελεσμάτων (single-path/minimum-result)**. Για διευκόλυνση χρησιμοποιούμε το x ως την ακτίνα μιας γειτονιάς.

3.5.1 Αναζήτηση Πλημμύρας (Flooding Search)

Σε αυτόν τον αλγόριθμο περιγράφουμε πως ένας κόμβος (peer) χρησιμοποιεί τις υπογραφές των γειτόνων του για να πραγματοποιήσει αναζητήσεις βασισμένες στο **μέγιστο βάθος αναζήτησης (maximum search depth)** ως συνθήκη τερματισμού. Στη συνέχεια θα δούμε τον αλγόριθμο πλημμύρας βασισμένο στην ολοκληρωμένη – γειτονική υπογραφή (complete-neighborhood signature (CN)) και θα τονίσουμε τις διαφορές που υπάρχουν με τις PN υπογραφές.

Algorithm : Flooding search based on complete neighborhood signatures.

Incoming Message: Search_Msg(*TTL*)

Local Variables: Local_Sig, Neighborhood_Sig, Search_Sig

System Parameters: x {the neighborhood radius}

Procedure:

- 1: compute Search_Sig based on Search_Msg.
 - 2: {check local content}
 - 3: if match(Search_Sig, Local_Sig) then
 - 4: examine local content to verify whether this is a true match or not.
 - 5: if true match then
 - 6: return a pointer to the result back to the peer from which this node got the search message.
 - 7: end if
 - 8: end if
 - 9: {check whether reach an edge of the search area}
 - 10: if $TTL = 0$ then
 - 11: stop
 - 12: end if
 - 13: {continue to search the neighborhood}
 - 14: if match(Search_Sig, Neighborhood_Sig) then
 - 15: forward the message Search_Msg($TTL - 1$) to all the neighbors.
 - 16: else
 - 17: if $TTL > x$ then
 - 18: forward the message Search_Msg($TTL - x - 1$) to all the neighbors located $x + 1$ hops away.
 - 19: end if
 - 20: end if
-

Ο αλγόριθμος εφαρμόζεται κάθε φορά που ένα μήνυμα αναζήτησης στέλνεται ή παραλαμβάνεται από έναν κόμβο (peer) σε ένα P2P δίκτυο. Κάθε μήνυμα αναζήτησης έχει και έναν TTL μετρητή ο οποίος έχει οριστεί εξ αρχής με μια τιμή (το μέγιστο βάθος αναζήτησης) που μπορεί αυτό το μήνυμα να μεταδοθεί. Αρχικά συγκρίνεται η υπογραφή αναζήτησης με την υπογραφή της τοπικής υπογραφής. Εάν βρεθεί κάποιο ταίριασμα, το περιεχόμενο αυτού του κόμβου εξετάζεται για να διαπιστωθεί αν πρόκειται για σωστό ταίριασμα (true match) ή για λανθασμένο ταίριασμα (false positive). Εάν πρόκειται για σωστό ταίριασμα (true match), επιστρέφεται στον κόμβο από τον οποίο προήλθε η αίτηση ένας δείκτης του αποτελέσματος. Στη συνέχεια ο κόμβος εξετάζει την τιμή του TTL να δει αν έχει φτάσει την μηδενική τιμή και αν ναι τότε σταματάει η συγκεκριμένη αναζήτηση. Σε άλλη περίπτωση συνεχίζει η σύγκριση της υπογραφής αναζήτησης με την γειτονική υπογραφή. Εάν βρεθεί κάποιο ταίριασμα τότε η αναζήτηση επεκτείνεται σε όλους τους γείτονες προωθώντας το μήνυμα και ταυτόχρονα μειώνοντας την TTL κατά 1. Εάν πάλι η υπογραφή αναζήτησης δεν ταιριάζει με την γειτονική υπογραφή, οι γείτονες που βρίσκονται εντός x **συνδέσεων (hops)** μακριά (την γειτονιά) δεν θα εξεταστούν. Σαν αποτέλεσμα, το μήνυμα αναζήτησης απορρίπτεται όταν $TTL \leq x$. Στην προηγούμενη περίπτωση η αναζήτηση θα πρέπει να συνεχίσει στους κόμβους που είναι $x+1$ συνδέσεις (hops) μακριά.

Οι αλγόριθμοι πλημμύρας για τις PN μεθόδους υπογραφής είναι ελάχιστα διαφορετικοί από αυτόν που προηγουμένως αναλύσαμε. Όταν μια υπογραφή αναζήτησης ταιριάζει με μια **PN-S** υπογραφή, το μήνυμα αναζήτησης προωθείται στον σχετικό γείτονα. Διαφορετικά, το μήνυμα αναζήτησης προωθείται σε κόμβους (peers) $x+1$ συνδέσεις (hops) μακριά, έξω από την γειτονιά από την οποία προήλθε η προηγούμενη μερική υπογραφή. Η σύγκριση μιας υπογραφής αναζήτησης με μια **PN-A** υπογραφή πραγματοποιείται ταιριάζοντας όλες τις τοπικές υπογραφές. Για κάθε ταιριαστή τοπική υπογραφή, ένα μήνυμα αναζήτησης προωθείται άμεσα στον αντίστοιχο κόμβο (peer). Εάν η υπογραφή αναζήτησης δεν ταιριάζει με μια **PN-A** υπογραφή (όπως και στην **PN-S**), το μήνυμα αναζήτησης προωθείται σε κόμβους (peers) $x+1$ συνδέσεις (hops) μακριά, έξω από την γειτονιά από την οποία προήλθε η προηγούμενη μερική υπογραφή.

3.5.2 Αναζήτηση Μοναδικού Μονοπατιού (Single Path Search)

Σε αυτήν τον αλγόριθμο βλέπουμε πως ένας κόμβος (peer) χρησιμοποιεί τις υπογραφές των γειτόνων του για να πραγματοποιήσει μια αναζήτηση μοναδικού μονοπατιού βασισμένος στον ελάχιστο αριθμό αποτελεσμάτων ως συνθήκη τερματισμού. Η βασική διαφορά μεταξύ του μοναδικού μονοπατιού αναζήτησης και της αναζήτησης πλημμύρας είναι ότι αν οι γειτονικές υπογραφές δεν ταιριάζουν με την υπογραφή αναζήτησης, ένας κόμβος (peer) τοποθετημένος $x+1$ συνδέσεις (hops) μακριά επιλέγεται τυχαία να επεκτείνει την αναζήτηση. Μια παράμετρος του συστήματος TNR (συνολικός αριθμός αποτελεσμάτων που έχουν βρεθεί μέχρι στιγμής) έχει τον ίδιο ρόλο με την παράμετρο TTL στην αναζήτηση πλημμύρας.

Για την CN υπογραφή, εάν η γειτονικές υπογραφές ταιριάζουν με την υπογραφή αναζήτησης, όλοι οι γείτονες είναι πιθανοί υποψήφιοι για σωστά ταιριάσματα (true matches). Για να καθοριστεί αν το ταιριασμα είναι σωστό ή όχι και πόσα αποτελέσματα υπάρχουν στην γειτονιά, η αναζήτηση θα πρέπει να επεκταθεί στην γειτονιά για να γίνει έλεγχος. Η διαφορά ανάμεσα στον αλγόριθμο μοναδικού μονοπατιού για CN, PN-S και PN-A είναι παρόμοια με αυτή που παρατηρήσαμε για τον αλγόριθμο πλημμύρας. Εάν υπάρχει ταιριασμα μεταξύ υπογραφών, τα μηνύματα ελέγχου των γειτονιών προωθούνται μόνο στους γείτονες με ταιριαστές γειτονικές υπογραφές σε PN-S ή κατευθείαν στους κόμβους (peers) με ταιριαστές τοπικές υπογραφές σε PN-A.

3.6 Κατασκευή και Διατήρηση Υπογραφών

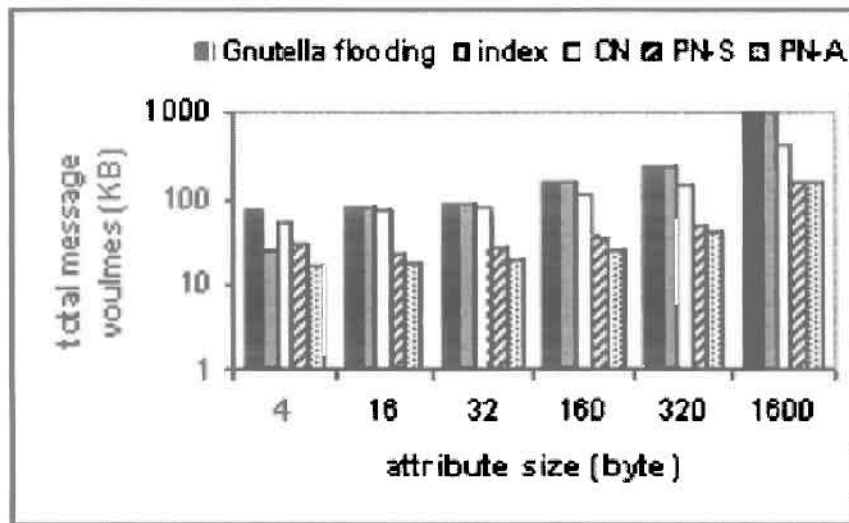
Βασικά οι γειτονικές υπογραφές κατασκευάζονται σε έναν κόμβο (peer) όταν αυτός προστίθεται για πρώτη φορά στο δίκτυο. Οι υπογραφές των γειτόνων χρειάζεται να αναδομηθούν ή να ενημερωθούν όταν κάποιοι κόμβοι (peers) προστίθενται/ φεύγουν από την γειτονιά τους ή και αν ενημερώνουν το περιεχόμενό τους. Παρακάτω περιγράφονται οι ενέργειες που πρέπει να λάβουν μέρος όταν ένας κόμβος προστίθεται, φεύγει ή ενημερώνεται.

- **Peer join:** ένας κόμβος ενημερώνει για την άφιξη του στέλνοντας ένα **μήνυμα άφιξης (join message)**, που περιλαμβάνει την τοπική του υπογραφή, στους κόμβους της γειτονιάς του. Όταν ένας κόμβος λάβει για πρώτη φορά ένα τέτοιο μήνυμα, πρώτα **προσθέτει (υπερθέτει (superimposes) ή προσαρτεί (appends))** την τοπική υπογραφή στο μήνυμα άφιξης στην αντίστοιχη γειτονική υπογραφή. Έπειτα στέλνει πίσω την δική του υπογραφή στον καινούριο κόμβο (peer) ώστε αυτός να μπορέσει να κατασκευάσει τις γειτονικές υπογραφές. Επιπλέον κάποιοι κόμβοι (peers) που δεν ήταν πριν στην γειτονιά, μπορεί να προστεθούν σε αυτήν εξαιτίας των ανακατατάξεων που θα γίνουν με τους νέους κόμβους. Οπότε και σε αυτήν την περίπτωση θα πρέπει να ανταλλαχθούν υπογραφές για να διατηρηθεί η ακρίβεια των γειτονικών υπογραφών.

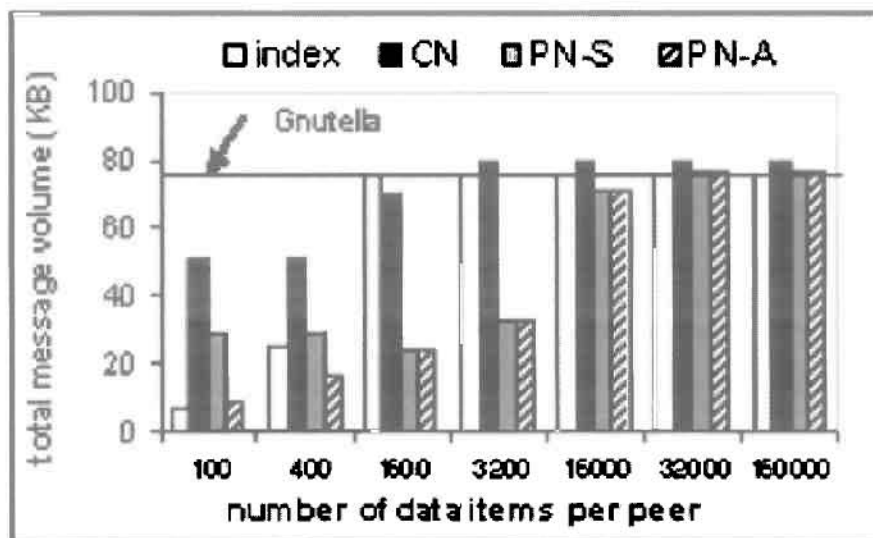
- **Peer leave:** όταν ένας κόμβος (peer) φύγει από το δίκτυο, ενημερώνει τους γείτονες στέλνοντας ένα **μήνυμα αποχώρησης (leave message)**. Για PN-A, το μήνυμα αποχώρησης περιέχει το αναγνωριστικό του κόμβου που αποχωρεί. Η ενημέρωση των γειτονικών υπογραφών για PN-A γίνεται απλά με την απομάκρυνση της υπογραφής του αποχωρούντος κόμβου από τις γειτονικές υπογραφές. Για CN ή PN-S αυτό το κομμάτι είναι πιο πολύπλοκο. Σε αυτήν την περίπτωση οι κόμβοι της γειτονιάς θα πρέπει να αναδομήσουν τις υπογραφές τους εξ αρχής μιας και δεν μπορούν απλά να τις αφαιρέσουν. Για CN οι ενδιαφερόμενοι κόμβοι ζητούν τις ατομικές γειτονικές υπογραφές από τους άλλους κόμβους της γειτονιάς. Για PN-S οι ενδιαφερόμενοι κόμβοι ζητούν τις ατομικές τοπικές υπογραφές από τους κόμβους στο επηρεασμένο κλαδί.
- **Peer update:** όταν ένας κόμβος ενημερώνει το περιεχόμενό του, η τοπική υπογραφή ενημερώνεται αντιστοίχως. Η διαδικασία για ενημέρωση των γειτονικών υπογραφών για CN και PN-S είναι η ίδια όπως στην περίπτωση **peer leave**. Για PN-A οι ενδιαφερόμενοι κόμβοι απλά πρέπει να ενημερώσουν τις σχετικές υποπογραφές στις γειτονικές υπογραφές.

3.7 Ανάλυση Απόδοσης

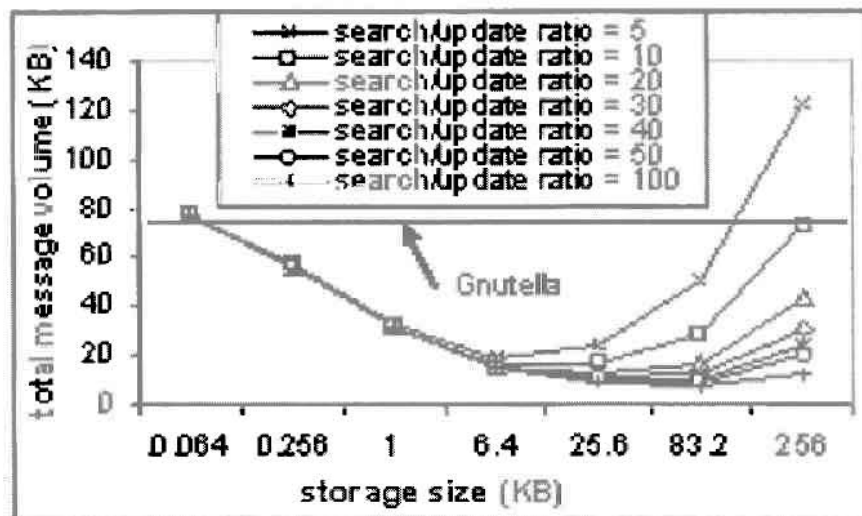
Στη συνέχεια θα δούμε μερικές γραφικές παραστάσεις που συνοψίζουν όλα όσα έχουν προαναφερθεί. Επιπλέον γίνονται πιο σαφείς οι διαφορές, οι ομοιότητες και οι ευρύτερες σχέσεις των διαφόρων μεθόδων υπογραφής (CN, PN-S, PN-A) με τους δύο αλγορίθμους (Flooding και Single Path Search).



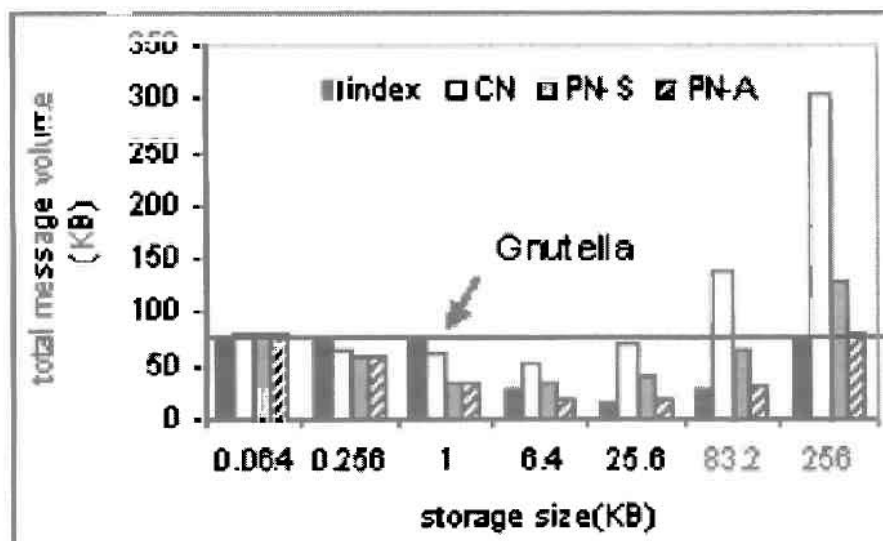
Σχήμα 3.3 : Πλημμύρα: επίδραση του μεγέθους των γνωρισμάτων στις μεθόδους των υπογραφών. Ο y άξονας είναι σε λογαριθμική σκάλα για ευκολία.



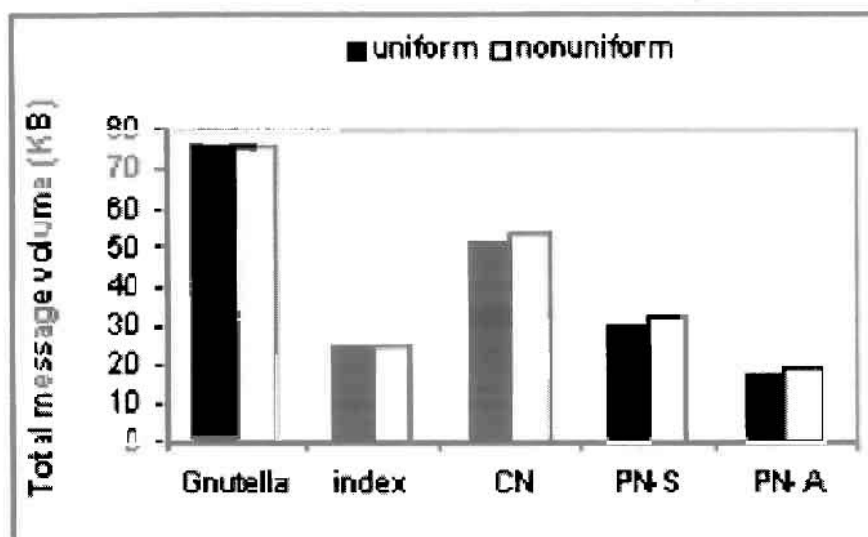
Σχήμα 3.4 : Πλημμύρα: επίδραση του αριθμού των δεδομένων ανά κόμβο (peer) στις τρεις μεθόδους υπογραφής. Ο συνολικός όγκος μηνυμάτων στη Gnutella φαίνεται από την οριζόντια γραμμή.



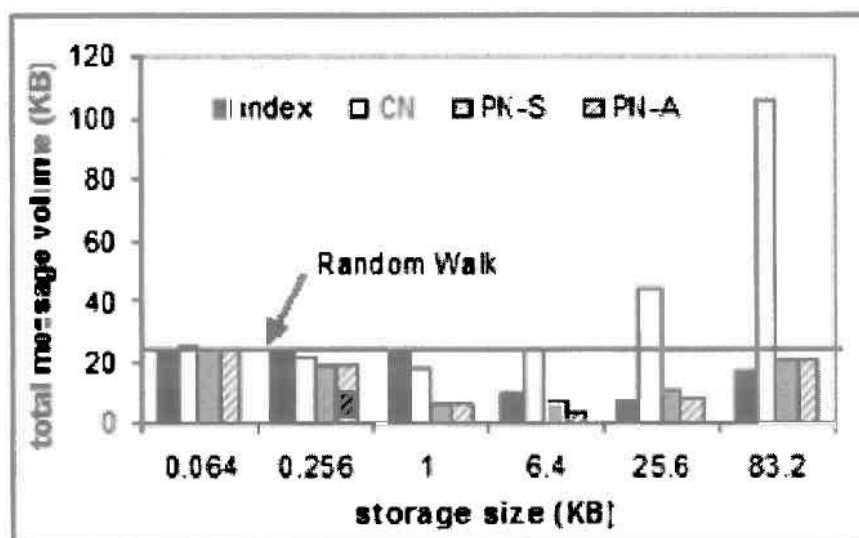
Σχήμα 3.5 : Πλημμύρα: επίδραση του ρυθμού αναζήτησης/ ενημέρωσης στις τρεις μεθόδους υπογραφής.



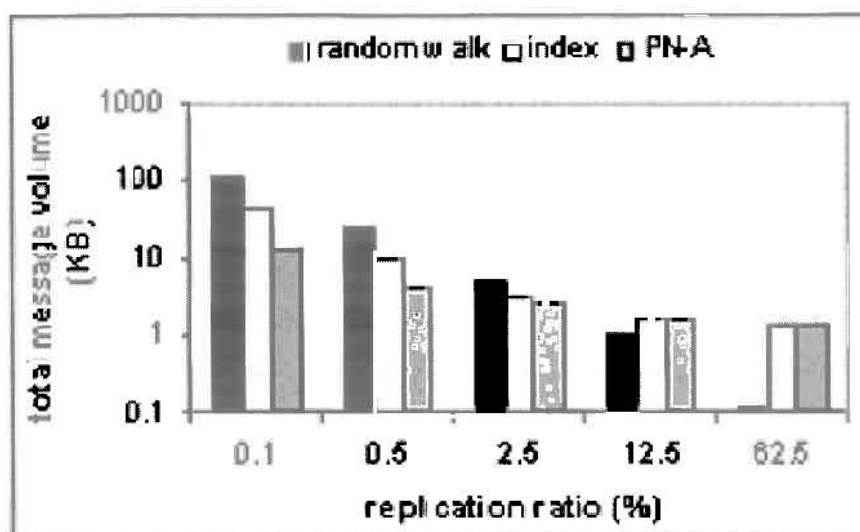
Σχήμα 3.6 : Πλημμύρα: σύγκριση του συνολικού όγκου μηνυμάτων μεταξύ του τοπικού ευρετηρίου (local index), CN, PN-S και PN-A. Ο συνολικός όγκος μηνυμάτων στη Gnutella φαίνεται από την οριζόντια γραμμή.



Σχήμα 3.7 : Πλημμύρα: επίδραση του διαμορισμού δεδομένων στην Gnutella, Ευρετήριο (Index) και στις τρεις μεθόδους υπογραφής.



Σχήμα 3.8 : Μοναδικό Μονοπάτι: σύγκριση του συνολικού όγκου μηνυμάτων μεταξύ του τοπικού ευρετηρίου (local index), CN, PN-S και PN-A. Ο συνολικός όγκος μηνυμάτων στη random walk φαίνεται από την οριζόντια γραμμή.



Σχήμα 3.8 : Μοναδικό Μονοπάτι: επίδραση του ρυθμού αντιγραφής (πολλά αντίγραφα στο δίκτυο) στη random walk, στο τοπικό ευρετήριο (local index) και στις τρεις μεθόδους υπογραφής.

ΚΕΦΑΛΑΙΟ 4

Υποστήριξη πολύπλοκων πολυδιάστατων ερωτήσεων σε ομότιμα (P2P) δίκτυα

Περιεχόμενα

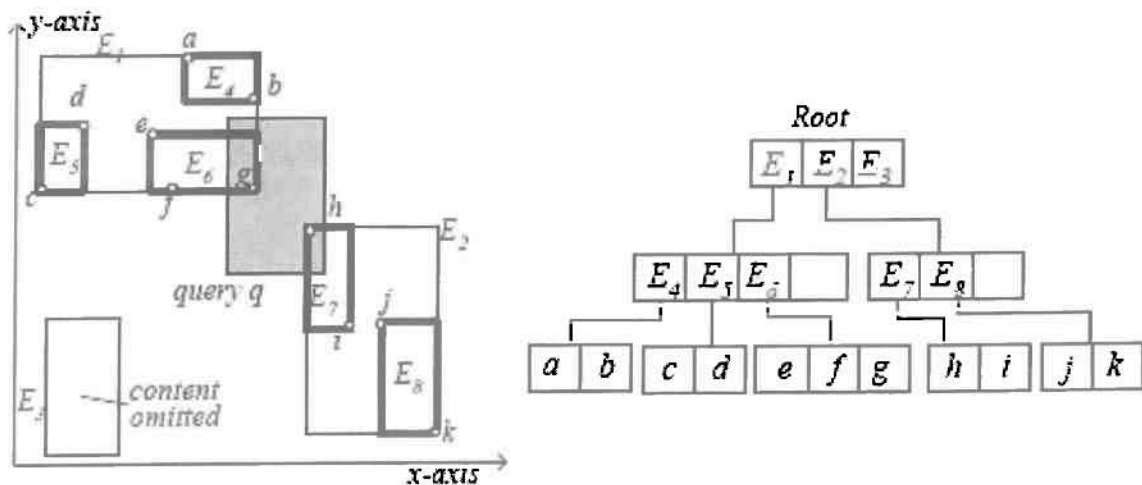
4.1 Εισαγωγή.....	70
4.2 Μέθοδος R*-tree για επεξεργασία ερωτήσεων	70
4.3 Μοντέλο Συστήματος	72
4.4 Διαμόρφωση και Διατήρηση Συστάδας.....	73
4.5 Κατασκευή και Διατήρηση Ευρετηρίου (Index).....	74
4.5.1 Κατασκευή NR-tree.....	74
4.5.2 Διατήρηση του NR-tree.....	76
4.6 Αλγόριθμοι Επεξεργασίας Ερωτήσεων (Query Processing).....	77
4.6.1 Αλγόριθμος επεξεργασίας ερωτήσεων αλληλουχίας (Range query processing).....	78
4.6.2 Αλγόριθμος επεξεργασίας ερωτήσεων k-κοντινότερου γείτονα (k-nearest neighbor query processing).....	80
4.7 Ανάλυση Απόδοσης	83
4.7.1 Κόστος επεξεργασίας μιας ερώτησης και Μέγεθος ερώτησης	83
4.7.2 Μέγεθος συστάδας και Πλεονασμός υπερ-κόμβων.....	84

4.1 Εισαγωγή

Τα τελευταία χρόνια βλέπουμε ότι υπάρχει μεγάλο ενδιαφέρον για την κοινή χρήση και διαμοιρασμό των δεδομένων ανάμεσα στους χρήστες του διαδικτύου. Για αυτό άλλωστε υπάρχουν και πολλές εφαρμογές που χρησιμοποιούν το παράδειγμα των ομότιμων (P2P) συστημάτων. Ακριβώς για τους παραπάνω λόγους είναι απαραίτητο οι ερωτήσεις αναζήτησης να γίνουν πιο σαφείς για να μπορούν να απαντηθούν όλες με τον καλύτερο τρόπο. Κάτι τέτοιο προφανώς κάνει τις ερωτήσεις αναζήτησης πιο πολύπλοκες. Ένα παράδειγμα είναι οι διαδικτυακές δημοπρασίες ακίνητης περιουσίας όπου μπορεί να χρειαστεί να απαντηθεί μια ερώτηση όπως "βρες πέντε διαθέσιμα διαμερίσματα με 4 υπνοδωμάτια που βρίσκονται κοντά σε σχολείο". Για να απαντηθούν τέτοιες ερωτήσεις επαρκώς στα ομότιμα (P2P) συστήματα πρέπει αν λάβουμε υπόψη τα χαρακτηριστικά τους. Τελευταίως έχει γίνει πρόοδος για την υποστήριξη πολυδιάστατων ερωτήσεων και στη συνέχεια θα αναλύσουμε σχετικές μεθόδους.

4.2 Μέθοδος R*-tree για επεξεργασία ερωτήσεων

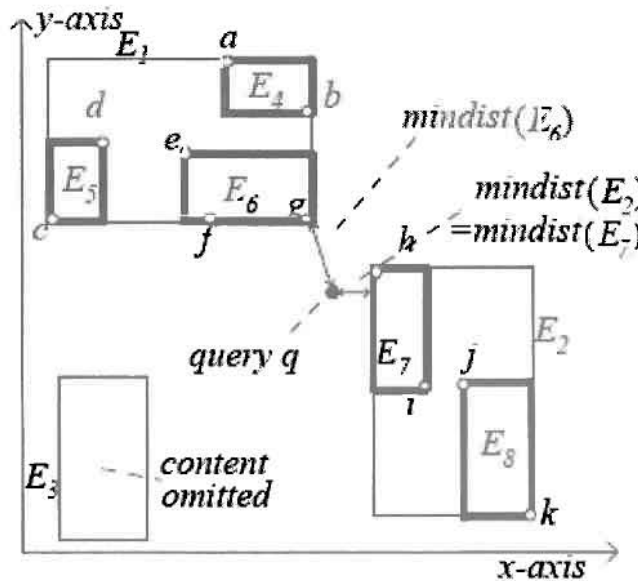
Η μέθοδος R*-tree ανάμεσα σε πολλές άλλες πολυδιάστατες μεθόδους πρόσβασης έχει γίνει ευρέως αποδεκτή από την βιομηχανία αλλά και από την κοινωνία της έρευνας. Η μέθοδος R*-tree είναι μία πολυδιάστατη επέκταση του B+-tree. Το Σχήμα 4.1 απεικονίζει ένα R*-tree που δεικτοδοτεί κάποια σημεία {a, b, c ...} υποθέτοντας χωρητικότητα τριών εισόδων ανά κόμβο (peer). Τα σημεία που είναι κοντά στο χώρο (όπως τα e, f, g) ομαδοποιούνται στον ίδιο κόμβο-φύλλο (peer-leaf) (E_6) που αντιπροσωπεύεται από ένα **ελάχιστο ορθογώνιο παραλληλόγραμμο οριοθέτησης (minimum bounding rectangle (MBR))**. Με τον ίδιο τρόπο ομαδοποιούνται και οι υπόλοιποι κόμβοι (peers) μέχρι το ανώτερο επίπεδο όπου αποτελείται από την **ρίζα (root)**.



Σχήμα 4.1 : Παράδειγμα R*-tree

Τα R*-trees μπορούν αποτελεσματικά να απαντήσουν σε πολλούς τύπους πολυδιάστατων ερωτήσεων (queries), ειδικά δε σε **ερωτήσεις αλληλουχίας (range queries)**. Δεδομένου ενός **παράθυρου ερώτησης (query window)** q (σκιασμένη περιοχή στο Σχήμα 4.1), μια ερώτηση αλληλουχίας επανακτά όλα τα αντικείμενα που βρίσκονται εντός ή τέμνουν το q .

Η επεξεργασία ξεκινάει από τη ρίζα (root) του R*-tree. Για οποιαδήποτε είσοδο της οποίας το MBR τέμνει το παράθυρο ερώτησης (query window), το **υπόδεντρο (sub-tree)** της εξερευνάται αναδρομικά. Εάν μια είσοδος-φύλλο (entry leaf) απαντηθεί τότε εξετάζονται όλα τα αντικείμενα των οποίων το ελάχιστο ορθογώνιο παραλληλόγραμμο οριοθέτησης (minimum bounding rectangle (MBR)) τέμνει το παράθυρο ερώτησης (query window). Οι εισόδους που δεν τέμνουν το παράθυρο ερώτησης (query window) δεν εξετάζονται.



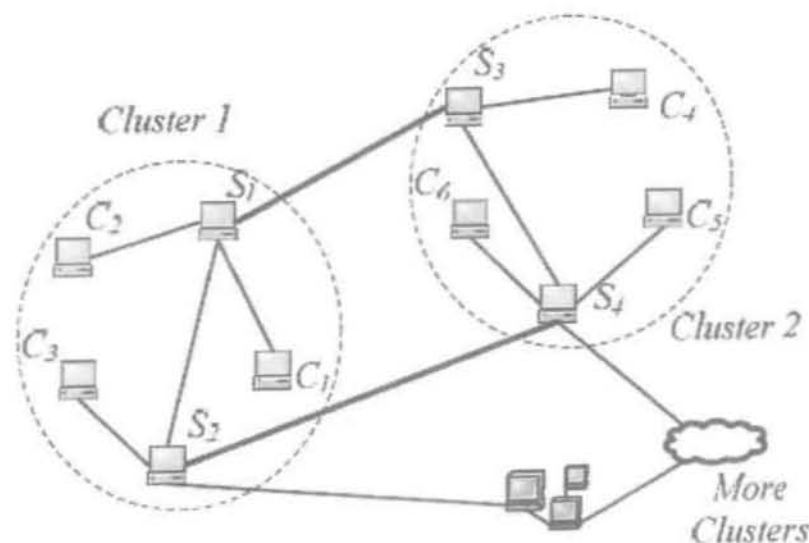
Σχήμα 4.2: Παράδειγμα kNN

Τα R*-trees επίσης υποστηρίζουν πολύ καλά και ένα άλλο είδος ερωτήσεων, τις **ερωτήσεις του κ-κοντινότερου γείτονα (k-nearest neighbor (kNN))**. Δεδομένου ενός **σημείου ερώτησης (query point)** q και της παραμέτρου $k (> 1)$, μια kNN ερώτηση ανακτά τα πρώτα k αντικείμενα (εάν αυτό είναι δυνατό) με την μικρότερη απόσταση από το q . Πιο συγκεκριμένα ξεκινώντας από την ρίζα, οι εισόδους ταξινομούνται ανάλογα με την **ελάχιστη απόσταση (mindist)** από το σημείο της ερώτησης q και τα υποδέντρα των εισόδων με την μικρότερη mindist εξερευνούνται πρώτα. Αυτή η διαδικασία επαναλαμβάνεται αναδρομικά μέχρι να φτάσουμε στο επίπεδο του φύλλου και να έχουν βρεθεί οι πρώτοι k υποψήφιοι.

Για να καταλάβουμε καλύτερα πως λειτουργεί ο αλγόριθμος θα δούμε ένα παράδειγμα. Στο Σχήμα 4.2 υποθέτουμε πως θέλουμε να βρούμε τους 2-κοντινότερους γείτονες του σημείου q . Ξεκινώντας από την ρίζα του R^* -tree υπολογίζεται η *mindist* κάθε εισόδου από το q στο ίδιο επίπεδο. Η είσοδος E_2 ερευνάται πρώτη μιας και έχει την μικρότερη *mindist* και διατρέχεται αναδρομικά. Το φύλλο εισόδου E_7 είναι το επόμενο που ερευνάται και τα σημεία h και i είναι υποψήφιοι για τους κοντινότερους γείτονες. Η *nn-mindist* (η **μικρότερη ελάχιστη απόσταση**) είναι από το h στο q . Έπειτα πάμε στην είσοδο E_8 αλλά αφού έχει μεγαλύτερη *mindist* από *nn-mindist* δεν εξετάζεται. Ο επόμενος κόμβος που εξετάζεται είναι ο E_1 και όπως προηγουμένως βρίσκεται το g και αντικαθιστά το i μιας και είναι πιο κοντά από το q . Οπότε η νέα *nn-mindist* είναι η απόσταση του g από το q .

4.3 Μοντέλο Συστήματος

Στη συνέχεια θα χρησιμοποιήσουμε ένα **υπερ-ομότιμο μοντέλο δικτύου (super-peer network model)** όπως φαίνεται στο Σχήμα 4.3. σε αυτό το μοντέλο μεταξύ μιας συστάδας από κοντινούς πελάτες κάποιοι **υπερ-κόμβοι (super-peers)** όπως S_1 και S_2 συμπεριφέρονται σαν **τοπικοί εξυπηρετητές (local servers)**, στους οποίους άλλοι **παθητικοί κόμβοι (passive peers)** συνδέονται. Ένας υπερ-κόμβος είναι υπολογιστικά πιο δυνατός και έχει μεγαλύτερη **ταχύτητα σύνδεσης στο διαδίκτυο (network bandwidth)** σε σχέση με τους παθητικούς κόμβους (passive peers). Έτσι οι παθητικοί κόμβοι (passive peers) μπορούν να υποβάλουν τις ερωτήσεις τους στους τοπικοί εξυπηρετητές (local servers) με τους οποίους συνδέονται και να λάβουν ικανοποιητικές απαντήσεις.



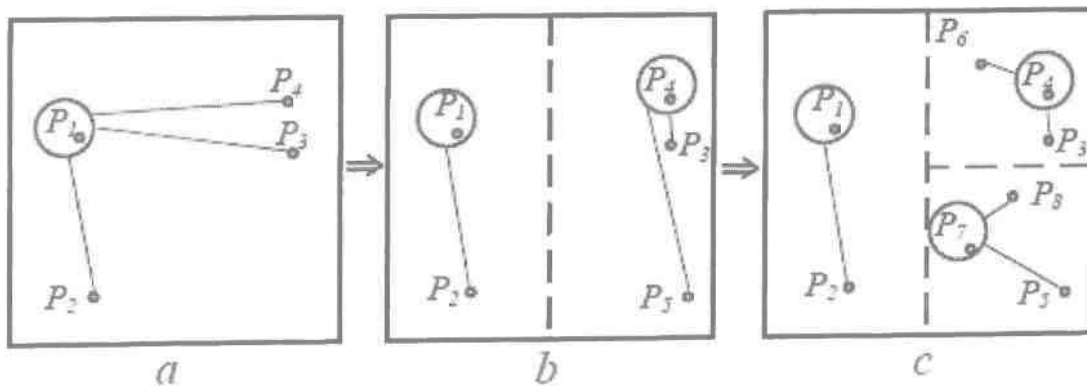
Σχήμα 4.3 : Ένα δίκτυο με υπερ-κόμβους (super-peer network)

Επιπλέον είναι αναγκαίο να υπάρχει αρκετός αριθμός τοπικών εξυπηρετητών (local servers) για δύο σημαντικούς λόγους : (i) εάν κάποιος υπερ-κόμβος (super-peer) αποτύχει να μπορεί ακόμα να εξυπηρετηθεί η συστάδα από κάποιον άλλο υπερ-κόμβο (super-peer) και (ii) οι υπερ-κόμβοι (super-peers) μοιράζονται τον φόρτο μεταξύ τους ώστε να γίνεται πιο αποτελεσματικά και γρήγορα η διακίνηση της πληροφορίας. Έτσι ενώ μπορούμε να έχουμε παραπάνω του ενός υπερ-κόμβους σε μια συστάδα, ωστόσο ένας παθητικός κόμβος μπορεί να συνδέεται μόνο σε έναν υπερ-κόμβο κάθε φορά.

4.4 Διαμόρφωση και Διατήρηση Συστάδας

Η συστάδα είναι μια ομάδα από υπερ-κόμβους (super-peers) και παθητικούς κόμβους (passive peers), όπου οι υπερ-κόμβοι συμπεριφέρονται ως τοπικοί εξυπηρετητές (local servers). Μπορούμε να υποθέσουμε ότι για οποιαδήποτε πολυδιάστατα δεδομένα υπάρχει μια **παγκόσμια αλληλουχία (universe range)** σε κάθε διάσταση. Βασιζόμενοι στο **κεντροειδές (centroid)** των δεδομένων κάθε κόμβου μπορούμε να τοποθετήσουμε αυτούς μοναδικά στο σύμπαν. Σε αυτήν την μελέτη θεωρούμε ότι η θέση ενός κόμβου σημαίνει το κεντροειδές των δεδομένων του και χρησιμοποιούμε ακριβώς αυτό το κεντροειδές για να εκπροσωπήσει τον κόμβο στο σύμπαν.

Όταν ένα σύστημα ξεκινάει, υπάρχει μόνο μία **συστάδα (cluster)**. Όταν ο συνολικός αριθμός κόμβων (peers) ξεπεράσει την χωρητικότητα (capacity) της συστάδας, το σύμπαν χωρίζεται σε δύο ίσες **ζώνες (zones)** και κάθε μια περιέχει μια καινούρια συστάδα. Σε κάθε ζώνη νέοι υπερ-κόμβοι (super-peers) εκλέγονται ανάλογα με την υπολογιστική τους δύναμη. Για να γίνει ακόμα πιο κατανοητό θα δούμε ένα παράδειγμα. Στο Σχήμα 4.4 υποθέτουμε πως η χωρητικότητα μιας συστάδας είναι 4. Στην αρχή υπάρχουν 4 κόμβοι (peers) και ο P_1 είναι υπερ-κόμβος (Σχήμα 4.4 (a)). Όταν προστίθεται ο κόμβος P_5 ο χώρος χωρίζεται σε 2 ζώνες και στην δεξιά ζώνη ο P_4 γίνεται ο νέος υπερ-κόμβος (Σχήμα 4.4 (b)). Όταν εισέρχονται νέοι κόμβοι τότε αναθέτονται στη ζώνη, στην οποία πέφτει το κεντροειδές των δεδομένων τους. Μια ζώνη μπορεί να διαχωριστεί ακόμη περισσότερο αν προστεθούν και άλλοι κόμβοι (Σχήμα 4.4 (c)).



Σχήμα 4.4 : Διαχωρισμός χώρου

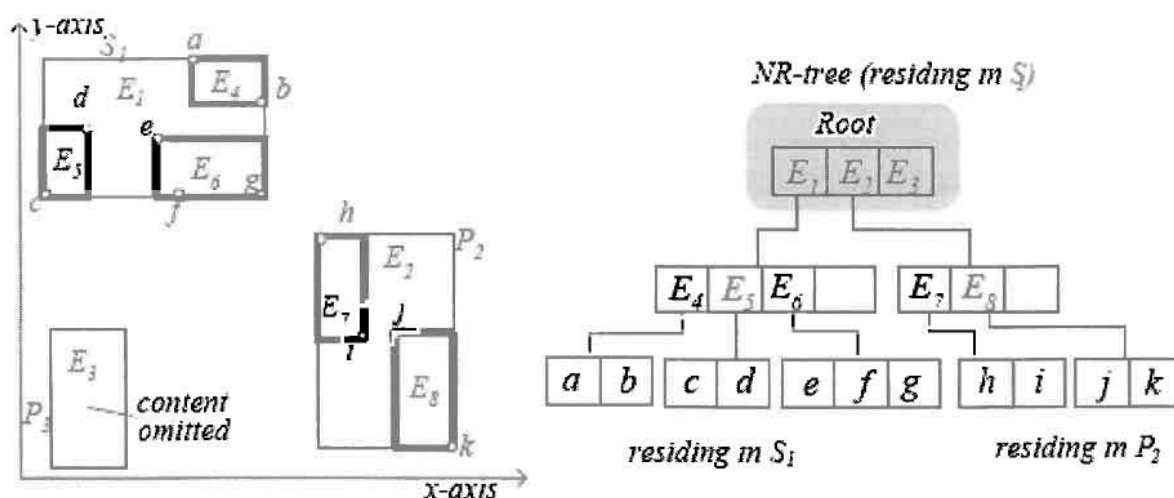
4.5 Κατασκευή και Διατήρηση Ευρετηρίου (Index)

Μέσα σε κάθε συστάδα, οι υπερ-κόμβοι (super-peers) οργανώνουν το ευρετήριο (index) και την επεξεργασία των ερωτήσεων (query processing) με την συνεργασία των παθητικών κόμβων. Για να μην υπερφορτώνονται οι υπερ-κόμβοι (super-peers), μέρος της δεικτοδότησης (indexing) και της επεξεργασίας ερωτήσεων (query processing) αναλαμβάνουν να κάνουν και οι παθητικοί κόμβοι. Στην συνέχεια θα δούμε την κατασκευή του ευρετηρίου του δέντρου NR (NR-tree index) μέσα σε μια συστάδα καθώς και την διατήρηση αυτής της δομής όταν ένας κόμβος εισέρχεται ή φεύγει.

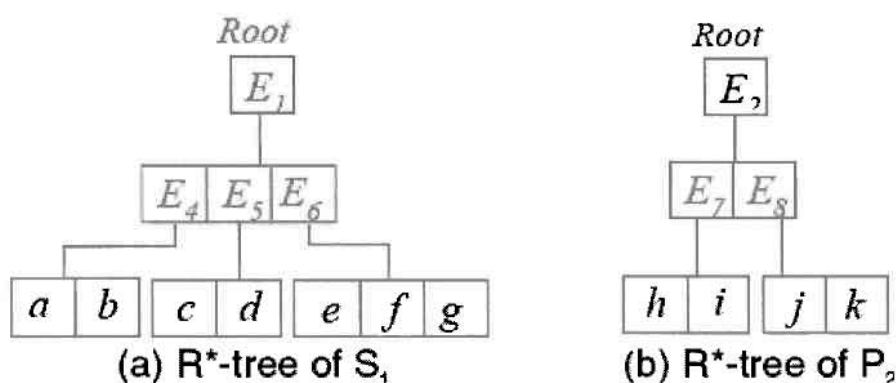
4.5.1 Κατασκευή NR-tree

Εκτός από το R*-tree που δεικτοδοτεί (indexing) ειδικά τα δικά του δεδομένα, κάθε υπερ-κόμβος (super-peer) διατηρεί ένα NR-tree που δεικτοδοτεί δεδομένα κόμβου καθώς και δικά του δεδομένα. Το R*-tree ενός υπερ-κόμβου (super-peer) παίζει τον ίδιο ρόλο με αυτόν στους παθητικούς κόμβους (passive peers). Παρόλα αυτά το NR-tree ακολουθεί έναν διαφορετικό ρόλο δεικτοδοτώντας ειδικά MBRs που συνοψίζουν δεδομένα σε κάθε παθητικό κόμβο και υπερ-κόμβο. Αυτά τα ειδικά MBRs έχουν προκύψει από περαιτέρω ομαδοποίηση των υπάρχοντων MBRs στα R*-trees των κόμβων.

Θα δούμε στη συνέχεια ένα παράδειγμα που θα βοηθήσει να κατανοήσουμε την κατασκευή ενός NR-tree. Στο Σχήμα 4.5 θεωρούμε τρεις κόμβους (S_1 , P_2 και P_3) που είναι στην ίδια ομάδα και ο S_1 είναι ο υπερ-κόμβος. Για χάρην ευκολίας θεωρούμε έναν υπερ-κόμβο αλλά το παράδειγμα μπορεί να επεκταθεί και σε περίπτωση με δύο υπερ-κόμβους. Ο S_1 περιέχει τα σημεία $\{a, \dots, g\}$, ο P_2 έχει τα $\{h, i, j, k\}$ και οι δύο κόμβοι δεικτοδοτούν τα δεδομένα τους χρησιμοποιώντας ένα R*-tree (Σχήμα 4.6). θεωρούμε ότι αρχικά μόνο ο S_1 είναι στο δίκτυο και ο P_2 επικοινωνεί με τον S_1 γιατί θέλει να εισέλθει στο δίκτυο. Τότε ο S_1 οφείλει να ενημερώσει τον P_2 τον αριθμό των MBRs που μπορούν να εξυπηρετηθούν αν πούμε m . Ένα μεγάλο m , για παράδειγμα 2, σε αυτήν την περίπτωση επιτρέπει στους παθητικούς κόμβους να στείλουν πιο πολλά MBRs στον υπερ-κόμβο αλλά κάτι εισάγει το πρόβλημα του χώρου. Υποθέτουμε ότι το $m=1$ και το P_2 πρέπει να ομαδοποιήσει ξανά τα υπάρχοντα MBRs στο R*-tree του για να φτιάξει ένα μεγαλύτερο MBR. Μιας και υπάρχει μόνο μια είσοδος στο επίπεδο ρίζας (E_2), στέλνεται στον S_1 . Εντωμεταξύ ο S_1 πρέπει να πραγματοποιήσει την ίδια εργασία με το δικό του R*-tree. Ο S_1 επιλέγει MBR της εισόδου E_1 . Παρομοίως υποθέτουμε ότι υπάρχει και ένα άλλο MBR από τον κόμβο P_3 . Έτσι λοιπόν με τρία MBRs, ο S_3 μπορεί να ξεκινήσει να φτιάχνει το δικό του NR-tree όπως φαίνεται στα δεξιά του Σχήματος 4.5 (σκιασμένη περιοχή). Η σκιασμένη περιοχή του επιπέδου ρίζας σώζεται στο NR-tree και ουσιαστικά τα δεδομένα διαμένουν σε όλους τους κόμβους.



Σχήμα 4.5 : Παράδειγμα NR-tree



Σχήμα 4.5 : R*-trees στον S_1 και P_2

Κάθε είσοδος στο NR-tree έχει ένα επιπλέον μαρκάρισμα ενός bit (bit marking) είτε τα δεδομένα είναι τοπικά αποθηκευμένα είτε σε άλλους κόμβους. Εάν υπάρχουν πολλοί υπερ-κόμβοι (super peers) στην συστάδα, πραγματοποιούνται παρόμοιες διαδικασίες σε όλους τους υπερ-κόμβους όπως στον S_1 για να επιτευχθεί η αντιγραφή του NR-tree. Το συνολικό κόστος της εισόδου ενός νέου κόμβου είναι το ίδιο με αυτό της εισαγωγής m ορθογώνιων αντικειμένων και είναι πολύ μικρό για μικρές τιμές του m .

4.5.2 Διατήρηση του NR-tree

Η διατήρηση ενός NR-tree ενεργοποιείται όταν νέοι κόμβοι (peers) εισάγονται. Όταν ένας παθητικός κόμβος φεύγει απλά πρέπει να μαρκάρουμε αυτό το γεγονός στα NR-trees των υπερ-κόμβων (super-peers). Όταν ένας υπερ-κόμβος (super-peer) φύγει, πρέπει να εκλεγεί ένα νέος υπερ-κόμβος στην θέση του.

Όταν ένας υπερ-κόμβος φεύγει, το NR-tree πρέπει να ενημερωθεί μιας και τα κοινά δεδομένα του δεν είναι πλέον διαθέσιμα. Όλα τα MBRs του αποσυνδεδεμένου κόμβου πρέπει να διαγραφούν από το NR-tree. Όταν ένας υπερ-κόμβος (super-peer) φύγει, ένας νέος κόμβος από τους παθητικούς εκλέγεται από τους εναπομείναντες κόμβους ανάλογα με την υπολογιστική του ικανότητα και την ταχύτητα (bandwidth) του. Ο νέος υπερ-κόμβος παίρνει ένα αντίγραφο του NR-tree από κάθε εναπομείναν υπερ-κόμβο, μετατρέπει όλα τα υπάρχοντα τοπικά bits σε λανθασμένα (false), διαγράφει εισόδους από τον αποσυνδεδεμένο υπερ-κόμβο και μαρκάρει τις εισόδους που αντιστοιχούν στα δικά του δεδομένα ως **τοπικές (local)**. Εάν την δεδομένη στιγμή υπάρχει μόνο ένας υπερ-κόμβος και αποτύχει (fail), η συστάδα καταστρέφεται και οι κόμβοι επανεισάγονται στο δίκτυο. Αυτή η περίπτωση είναι αρκετά σπάνια μιας και ενισχύουμε με παραπάνω του ενός υπερ-κόμβου. Τέλος, μπορεί να προκύψει αλλαγή στους υπερ-κόμβους αν εισαχθεί στη συστάδα ένας κόμβος με μεγαλύτερη υπολογιστική δύναμη, τότε θα αντικαταστήσει τον πιο αδύναμο υπερ-κόμβο.

4.6 Αλγόριθμοι Επεξεργασίας Ερωτήσεων (Query Processing)

Η επεξεργασία ερωτήσεων (query processing) συντονίζεται από τους υπερ-κόμβους (super-peers) και βοηθούν εν μέρει και οι παθητικοί κόμβοι (passive peers). Όταν ένας παθητικός κόμβος θέλει να θέσει μια ερώτηση, στέλνει την ερώτηση στον υπερ-κόμβο του. Για παράδειγμα, στην περίπτωση μιας ερώτησης αλληλουχίας (range query), ο παθητικός κόμβος στέλνει το ορθογώνιο παραλληλόγραμμο της ερώτησης (query rectangle) στους υπερ-κόμβους. Στην περίπτωση της kNN ερώτησης (kNN query), στέλνονται το **σημείο της ερώτησης (query point)** και ο επιθυμητός αριθμός των κοντινότερων γειτόνων (k). Όταν ένας υπερ-κόμβος λάβει μια ερώτηση από έναν παθητικό κόμβο, προωθεί την ερώτηση στην αντίστοιχη συστάδα. Πιο συγκεκριμένα, η ερώτηση (query) προωθείται στη συστάδα που είναι υπεύθυνη για την ζώνη που βρίσκεται το κεντροειδές (centroid) της ερώτησης αλληλουχίας (range query) ή το σημείο της kNN ερώτησης (kNN query). Επισημαίνουμε τον υπερ-κόμβο σε αυτήν την συστάδα ως τον **πρωτεύοντα (primary)** υπερ-κόμβο για αυτήν την ερώτηση. Αυτός ο πρωτεύοντας (primary) υπερ-κόμβος ξεκινάει την επεξεργασία της ερώτησης εκ μέρους του κόμβου που την έθεσε. Πρώτα ελέγχει το δικό του NR-tree για αποτελέσματα και επίσης προωθεί την ερώτηση στους γείτονες του στη ζώνη με την TTL τιμή να υποδηλώνει το μήκος του μονοπατιού αναζήτησης. Οι άλλες συστάδες επεξεργάζονται την ερώτηση με παρόμοιο τρόπο και μειώνουν την τιμή TTL κατά ένα πριν προωθήσουν την ερώτηση παραπέρα. Όταν η τιμή TTL φτάσει το 0, η ερώτηση δεν προωθείται περαιτέρω.

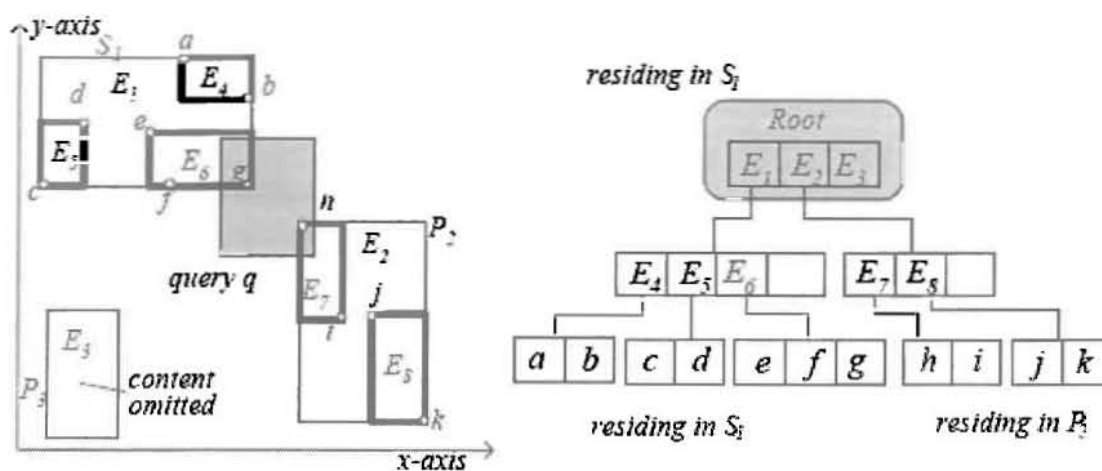
Η αποτυχία (fail) των υπερ-κόμβων είναι αρκετά συχνή στα ομότιμα δίκτυα. Για να είναι το σύστημα μας πιο ανθεκτικό, υπάρχει ένας μηχανισμός στους υπερ-κόμβους ώστε αν ένας αποτύχει (fails) τότε οι υπόλοιποι αναλαμβάνουν να συνεχίσουν να επεξεργάζονται ερωτήσεις για τις οποίες ήταν υπεύθυνος ο υπερ-κόμβος που απέτυχε. Συγκεκριμένα με το που λαμβάνει μια ερώτηση ένας υπερ-κόμβος, τυχαία επιλέγει έναν άλλο υπερ-κόμβο μέσα στην ίδια συστάδα ως **εφεδρικό (backup)** και του προωθεί την ερώτηση μαζί με την διεύθυνση **αυτού που έθεσε την ερώτηση (query issuer)**. Μετά από μια ορισμένη περίοδο, εάν ο εφεδρικός υπερ-κόμβος δεν λάβει από τον αρχικό κόμβο μια **αναγνώριση (acknowledgement)**, μπορεί να επεξεργαστεί την ερώτηση. Μόνο αν και οι δύο υπερ-κόμβοι αποτύχουν ταυτόχρονα δεν θα γίνει η επεξεργασία της ερώτησης.

4.6.1 Αλγόριθμος επεξεργασίας ερωτήσεων αλληλουχίας (Range query processing)

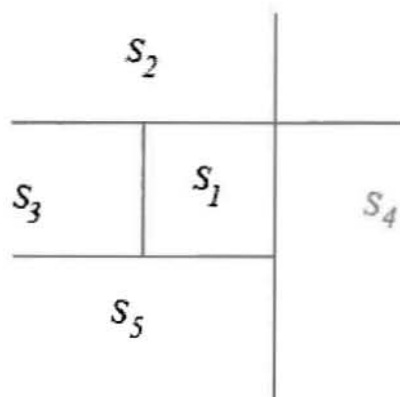
Οι ερωτήσεις αλληλουχίας απαντιούνται από συνεργαζόμενους υπερ-κόμβους και παθητικούς κόμβους. Όταν ένας υπερ-κόμβος (super-peer) λάβει μια ερώτηση αλληλουχίας από έναν παθητικό κόμβο, δρομολογεί την ερώτηση στη συστάδα (cluster) που είναι υπεύθυνη για την ζώνη μέσα στην οποία βρίσκεται το κεντροειδές (centroid) της ερώτησης. Με αυτόν τον τρόπο η ερώτηση δρομολογείται στον πρωτεύοντα (primary) υπερ-κόμβο, ο οποίος ελέγχει το NR-tree δέντρο του και βρίσκει παθητικούς κόμβους με δεδομένα που τέμνουν την περιοχή της ερώτησης. Αυτοί οι κόμβοι λαμβάνουν την ερώτηση και την διεύθυνση του κόμβου που έθεσε αυτήν, και επεξεργάζονται την ερώτηση με τα δικά τους R*-trees και επιστρέφουν τα αποτελέσματα ατομικά. Εντωμεταξύ, ο υπερ-κόμβος προωθεί την ερώτηση μαζί με την τιμή TTL, σημειώνοντας πόσο περισσότερο μπορεί να προωθηθεί. Με τον ίδιο τρόπο η ερώτηση προωθείται και στις άλλες συστάδες.

Για να καταλάβουμε ακόμα περισσότερο την διαδικασία επεξεργασίας μιας ερώτησης αλληλουχίας (range query) θα δούμε ένα παράδειγμα. Θεωρούμε το παράδειγμα στα Σχήματα 4.6 και 4.7. Στο Σχήμα 4.6 βλέπουμε κόμβους (peers) στην συστάδα S_1 (χρησιμοποιούμε την ονομασία του υπερ-κόμβου για να δείξουμε μια συστάδα) και το Σχήμα 4.7 δείχνει τις συστάδες γύρω από το S_1 . Θεωρούμε ότι ένας παθητικός κόμβος (passive peer) P_1 σε κάποια συστάδα καταθέτει μια ερώτηση αλληλουχίας (range query) q (το σκιασμένο ορθογώνιο παραλληλόγραμμο) και αρχικά δρομολογείται στον υπερ-κόμβο S_1 . Επιπλέον ο S_1 είναι ο **πρωτεύοντας (primary)** υπερ-κόμβος για αυτήν την ερώτηση. Από την πλευρά του υπερ-κόμβου, ο S_1 ελέγχει το NR-tree του και υποδεικνύει ότι q τέμνεται με το MBR E_2 του παθητικού κόμβου P_2 όπως επίσης και με το τοπικό MBR E_1 . Ο S_1 κατευθείαν προωθεί την ερώτηση q καθώς και την διεύθυνση του P_1 στον P_2 , ο οποίος θα συνεχίσει την επεξεργασία της ερώτησης. Να σημειωθεί δε ότι αυτή η προώθηση γίνεται αφού έχει διαπεραστεί το NR-tree. Έτσι με αυτόν τον τρόπο δεν ελέγχει ένας παθητικός κόμβος την ίδια ερώτηση πολλές φορές. Ο S_1 επίσης προωθεί τις ίδιες πληροφορίες στις γειτονικές του συστάδες (S_2, \dots, S_5) μαζί με τον μετρητή TTL. Ο S_2 είναι ο **επόμενος (subsequent)** υπερ-κόμβος για αυτήν την ερώτηση. Από την πλευρά του παθητικού κόμβου (passive peer), ο P_2 επεξεργάζεται την q χρησιμοποιώντας έναν συγκεκριμένο αλγόριθμο επεξεργασίας ερωτήσεων αλληλουχίας, ανακτώντας σημαντικά αποτελέσματα (όπως το σημείο h) και περνώντας το στον κόμβο P_1 που θα θέσει την ερώτηση. Ένας άλλος υπερ-κόμβος, ο S_2 , επεξεργάζεται το q με παρόμοιο τρόπο όπως το S_1 με μόνη διαφορά ότι το S_2 θα μειώσει τον μετρητή TTL κατά 1 εάν προωθήσει το q σε άλλες συστάδες.

Η διαδικασία επεξεργασίας ερωτήσεων αλληλουχίας στον **πρωτεύοντα υπερ-κόμβο (primary super-peer)** φαίνεται παρακάτω και μπορεί να τροποποιηθεί για τους **επόμενους (subsequent)** υπερ-κόμβους προσθέτοντας γραμμές για τον έλεγχο της λήξης ή της μείωσης της τιμής του μετρητή TTL. Μεγάλες τιμές του μετρητή TTL αυξάνουν τον αριθμό των αποτελεσμάτων, αλλά αυξάνουν και το φόρτο του συστήματος.



Σχήμα 4.6 : Παράδειγμα μιας ερώτησης αλληλουχίας (range query)



Σχήμα 4.7 : Συστάδες γύρω από το S_1

Ο αλγόριθμος είναι :

Algorithm range_query

Input q : range query rectangle, $addr$: address of the peer that issues q

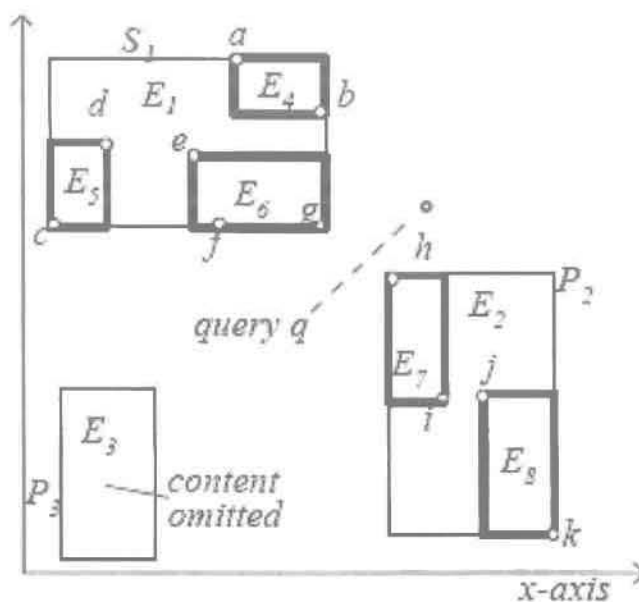
1. check NR-tree with q and retrieve all MBRs in leaf level intersecting q
2. for each MBR mbr retrieved in step 1
3. if mbr is local
4. check local R*-tree and return results to $addr$
5. else // mbr is non-local
6. forward q and $addr$ to the peer containing mbr
7. forward q and $addr$ to neighboring clusters

End range_query

4.6.2 Αλγόριθμος επεξεργασίας ερωτήσεων k-κοντινότερου γείτονα (k- nearest neighbor query processing)

Η επεξεργασία ερωτήσεων k-κοντινότερου γείτονα γίνεται από συνεργαζόμενους υπερ-κόμβους και παθητικούς κόμβους. Όταν ένας υπερ-κόμβος λάβει μια k-NN ερώτηση από τους παθητικούς του κόμβους δρομολογεί την ερώτηση στη συστάδα (cluster) που είναι υπεύθυνη για την ζώνη μέσα στην οποία βρίσκεται το σημείο (point) της ερώτησης. Στον πρωτεύοντα υπερ-κόμβο χρησιμοποιούμε τον αλγόριθμο **πρώτου βάθους (depth first)** έναντι του αλγορίθμου **πρώτου καλύτερου (best first)** αφού απαιτεί λιγότερο χώρο. Να επισημανθεί δε ότι το επίπεδο φύλλου στο NR-tree δεν περιέχει πραγματικά δεδομένα, μιας και τα δεδομένα βρίσκονται είτε σε παθητικούς κόμβους είτε έχουν ευρετηριαστεί (indexed) από ένα άλλο R*-tree στον υπερ-κόμβο (super-peer).

Ο αλγόριθμος ξεκινάει από το επίπεδο ρίζας του NR-tree, ταξινομώντας εισόδους ανάλογα με την mindist τους από το σημείο της ερώτησης (query point) και αναδρομικά διασχίζοντας τα υποδέντρα των εισόδων με μικρότερες mindist. Όταν επισκέπτεται μια είσοδος φύλλου στο NR-tree (που ουσιαστικά δεν περιέχει πραγματικά δεδομένα), ο αντίστοιχος κόμβος που περιέχει τα δεδομένα λαμβάνει την ερώτηση και την μέχρι τότε k-οστή μικρότερη mindist. Αυτό επιτρέπει στον κόμβο να συνεχίσει την επεξεργασία της ερώτησης μόνος του. Όταν ένας παθητικός κόμβος λάβει μια k-NN ερώτηση, διασχίζει το δικό του R*-tree με την mindist. Οι κόμβοι-είσοδοι με μεγάλη mindist δεν εξετάζονται. Ο παθητικός κόμβος επιστρέφει το πολύ k αντικείμενα στον υπερ-κόμβο μαζί με την απόσταση τους από το σημείο της ερώτησης (query point). Τα αποτελέσματα από τους παθητικούς κόμβους συνδυάζονται από τον αρχικό υπερ-κόμβο για να παράγουν τους πρώτους υποψήφιους k- κοντινότερους γείτονες. Σημειώνουμε την μεγαλύτερη mindist ως mindist_k. Εντωμεταξύ, ο πρωτεύοντας υπερ-κόμβος προωθεί την ερώτηση στις γειτονικές συστάδες με την μέχρι στιγμής mindist_k. Παρομοίως με τις ερωτήσεις αλληλουχίας (range queries), μια τιμή TTL έχει οριστεί για κάθε k-NN ερώτηση και τα τελικά αποτελέσματα συνδυάζονται στον αρχικό υπερ-κόμβο (super-peer) πριν αποσταλούν στον κόμβο που έθεσε (issuing node) την ερώτηση. Η τιμή TTL εξασφαλίζει ότι η ερώτηση δεν θα προωθείται ατέρμονα στο δίκτυο, αλλά θα επιστραφούν αποτελέσματα εντός πεπερασμένου χρόνου.



Σχήμα 4.8 : Παράδειγμα k-NN ερώτησης

Στο Σχήμα 4.8 βλέπουμε ένα παράδειγμα k-NN ερώτησης. Θεωρούμε ότι ο παθητικός κόμβος P_1 σε κάποια συστάδα θέτει μια 2-NN ερώτηση και δρομολογείται στον υπερ-κόμβο S_1 και το σημείο της ερώτησης (query point) είναι το q . Ο S_1 είναι ο πρώτος υπερ-κόμβος για αυτήν την ερώτηση. Ο S_1 ελέγχει το NR-tree του και αποφασίζει να επισκεφτεί το E_2 πρώτα μιας και έχει την μικρότερη mindist. Παρόλα αυτά το E_2 αντιστοιχεί στα δεδομένα που βρίσκονται στον P_2 . Συνεπώς η ερώτηση και η μέχρι στιγμής mindist_k (άπειρο μιας και δεν έχει βρεθεί κανείς υποψήφιος) στέλνονται στον P_2 . Ο P_2 επεξεργάζεται την k-NN ερώτηση χρησιμοποιώντας τον αλγόριθμο που περιγράφηκε στο 4.2 και αναφέρει τα αποτελέσματα στα σημεία h και i μαζί με την mindist τους από το q . Ο S_1 προσαρμόζει την mindist_k ώστε να είναι η 2^n μικρότερη απόσταση από τους μέχρι στιγμής υποψήφιους πριν αποφασίσουν να επισκεφθούν το E_2 . Αφού η mindist του E_2 είναι μικρότερη από αυτήν του σημείου i , το υπόδεντρο του E_2 πρέπει να ελεγχθεί. Η ερώτηση περνάει στο R*-tree του S_1 και το σημείο g αντικαθιστά το i . Ταυτόχρονα ο S_1 προωθεί την ερώτηση και την μέχρι στιγμής mindist_k στον S_2 , ο οποίος τρέχει τον αλγόριθμο με εξαίρεση : (i) η αρχική mindist θέτεται ίση με mindist_k και (ii) η τιμή TTL θα μειωθεί κι άλλο πριν η ερώτηση προωθηθεί. Ο S_2 θα επιστρέψει τα αποτελέσματα στον S_1 αργότερα. Παρακάτω θα δούμε τον αλγόριθμο για τον πρωτεύοντα υπερ-κόμβο αλλά μπορεί να τροποποιηθεί για του υπόλοιπους κόμβους αρχικοποιώντας την mindist_k με την τιμή που περνάει με την ερώτηση.

Ο αλγόριθμος είναι :

Algorithm *kNN_query*

Input *q*: query point, *k*: number of nearest neighbors, *addr*: address of the query-issuing peer

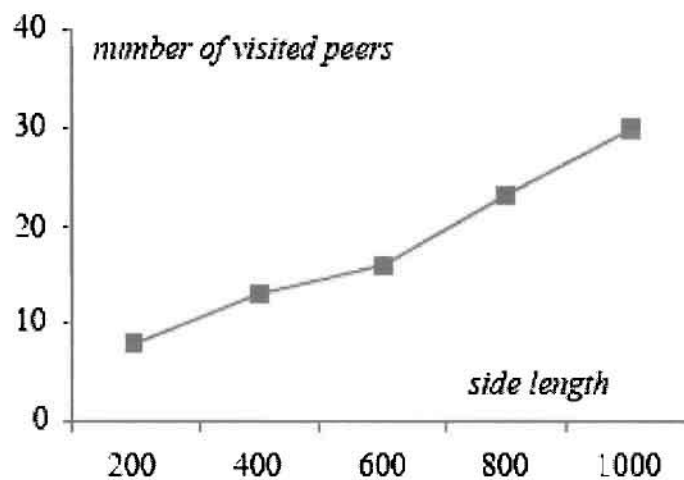
Variables: *mindist_k*, the largest mindist among nearest neighbor candidates, initially set to infinity. *DIST_N*, mindist of node *N* to *q*

1. for each node *N* in the NR-tree
 2. if the current node *N* is the leaf level of NR-tree
 3. if *DIST_N* < *mindist_k*
 pass *q*, *mindist_k*, and *addr* to the peer hosting data in *N*
 5. else
 6. Sort entries in *N* and recursively ascending their sub-tree
 7. forward *q*, *mindist_k*, and *addr* to neighboring clusters
- End *kNN_query***
-

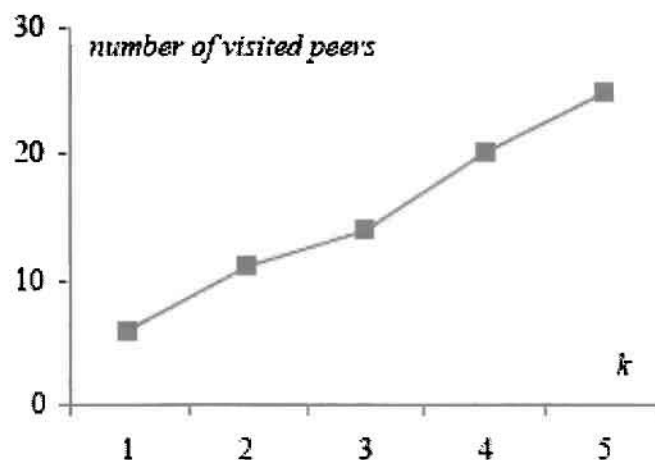
4.7 Ανάλυση Απόδοσης

4.7.1 Κόστος επεξεργασίας μιας ερώτησης και μέγεθος ερώτησης

Οι ερωτήσεις επηρεάζουν το κόστος επεξεργασίας και τον χρόνο. Πιο συγκεκριμένα, όσο πιο μεγάλο είναι το μήκος μιας ερώτησης αλληλουχίας (q_L) τόσο πιο πολλά αποτελέσματα θα ανακτηθούν και συνεπώς πιο πολλοί κόμβοι θα επισκεφτούν. Παρόμοια συμβαίνει και για k -NN ερωτήσεις. Το Σχήμα 4.9(a) δείχνει ότι ο αριθμός των κόμβων που έχουν επισκεφτεί μεγαλώνει όσο μεγαλώνει το μέγεθος της ερώτησης αλληλουχίας και το ίδιο συμβαίνει στο Σχήμα 4.9(b) για την k -NN ερώτηση.



(a) Ερώτησης αλληλουχίας (Range query)

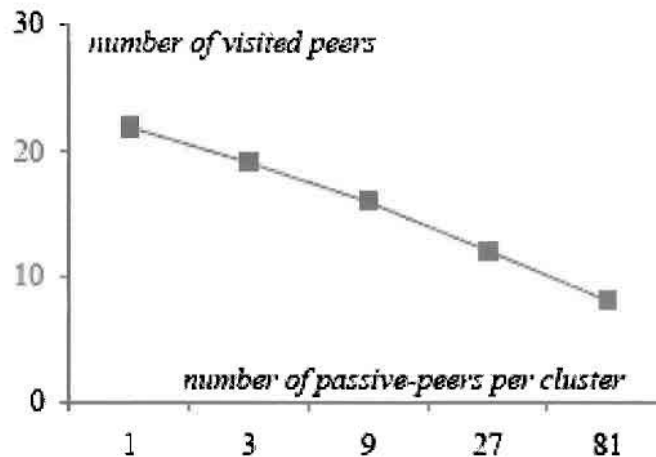


(b) k -NN ερώτηση (k -NN query)

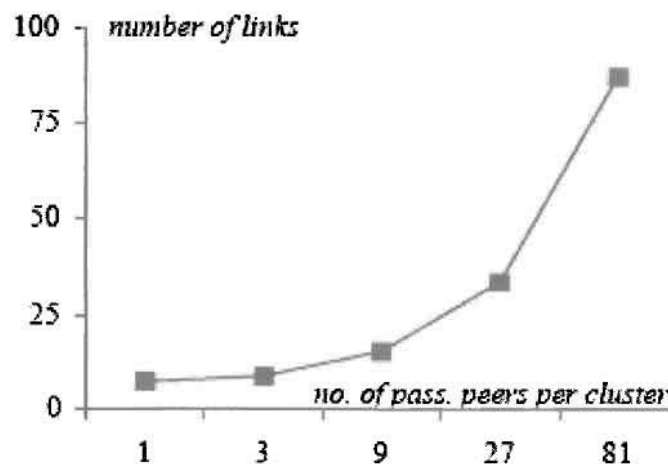
Σχήμα 4.9 : Κόστος vs. μέγεθος ερώτησης

4.7.2 Μέγεθος συστάδας και Πλεονασμός υπερ-κόμβων

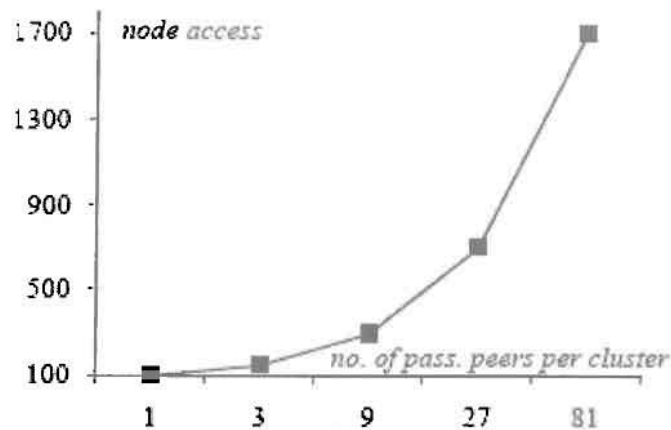
Το μέγεθος της συστάδας (cluster size) είναι σημαντικό αφού σχετίζεται με τον αριθμό των συστάδων και τον φόρτο των υπερ-κόμβων. Ένας μεγάλος αριθμός από κόμβους σε μια συστάδα σημαίνει μικρότερος αριθμός συστάδων και συνεπώς πιο φτηνό κόστος δρομολόγησης μέσα στην συστάδα. Βέβαια αυτό επίσης σημαίνει ότι οι υπερ-κόμβοι πρέπει να φέρουν εις πέρας πιο πολλές λειτουργίες όπως προσθήκη/αποχώρηση κόμβου, επεξεργασία ερώτησης και δρομολόγηση (peer join/ departure, query, and routing). Αυτές οι λειτουργίες εάν διαμοιράζονται από περισσότερους υπερ-κόμβους σε μια συστάδα, θα υπάρχει λιγότερος ατομικός φόρτος. Σε αυτήν την ενότητα μετράμε το κόστος διατήρησης από δύο πλευρές, **πρόσβαση στον κόμβο (node access)** από την προσθήκη/ αποχώρηση κόμβου και τον **συνολικό αριθμό των συνδέσεων (total number of links)** που διατηρούνται από έναν υπερ-κόμβο. Θεωρούμε το μέγεθος του δικτύου στα 1024, 100k προσθήκες/ αποχωρήσεις κόμβων και παράλληλα ερωτήσεις (queries) που θέτονται.



(a) Κόστος αναζήτησης (Search cost)



(b) Αριθμός Συνδέσεων (Number of links)



(c) Πρόσβαση στον κόμβο (Node Access)

Σχήμα 4.9 : Κόστος vs. Μέγεθος συστάδας

Τα αποτελέσματα φαίνονται στο Σχήμα 4.9 όπου σε κάθε κόμβο υπάρχουν τρεις υπερ-κόμβοι. Το (a) μέρος δείχνει ότι η αναζήτηση γίνεται πιο λειτουργική καθώς οι συστάδες μεγαλώνουν ακόμα περισσότερο, το οποίο συμβαίνει λόγω μικρότερων μονοπατιών δρομολόγησης μέσα στη συστάδα και μεγαλύτερα NR-trees που ευρετηριάζουν περισσότερα δεδομένα. Η αρνητική πλευρά μιας μεγάλης συστάδας φαίνεται στο (b) και (c) μέρος, όπου κάθε υπερ-κόμβος πρέπει να διατηρήσει περισσότερες συνδέσεις και να πραγματοποιήσει περισσότερες λειτουργίες διατήρησης. Συμπερασματικά, όσο οι υπερ-κόμβοι δεν υπερφορτώνονται από εργασίες, είναι επιθυμητό να υπάρχουν περισσότεροι κόμβοι σε μια συστάδα, εάν πάλι οι υπερ-κόμβοι είναι υπερφορτωμένοι, είναι καλύτερο να υπάρχουν περισσότεροι κόμβοι για να μοιραστούν τον φόρτο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] C.K. Lee, W.-C. Lee, and S. Madria, Pervasive Data Access in Wireless and Mobile Computing Environments, *Wireless Communication and Mobile Computing Journal*, to appear.
- [2] J. Xu, X. Tang, and W.-C. Lee, On Scheduling Time-Critical On-Demand Broadcast, *IEEE Transaction on Distributed and Parallel Systems*, Volume 17, Issue 1, Jan. 2006, pp. 3-14.
- [3] J. Xu, W.-C. Lee, and J. Liu, Scheduling Web Requests in Broadcast Environments, *the thirteenth International World Wide Web Conference (WWW'04)*, New York, NY, May 17-22, 2004, pp. 280-281. (Poster)
- [4] J. Xu, Q. Hu, W.-C. Lee, and D.L. Lee, Performance Evaluation of an Optimal Cache Replacement Policy for Wireless Data Dissemination under Cache Consistency, *IEEE Transaction on Knowledge and Data Engineering (TKDE)*, Volume 16, No. 1, January, 2004, pp. 125-139.
- [5] J. Xu, W.-C. Lee, D.L. Lee, and Q. Hu, Data Broadcast, *Handbook of Wireless Networks and Mobile Computing*, edited by I. Stojmenovic, John Wiley & Sons, 2002, ISBN 0-471-41902-8, pp. 243-265.
- [6] Q. Hu, W.-C. Lee, and D.L. Lee, Indexing Techniques for Power Management in Multi-Attribute Data Broadcast, *ACM Mobile Networks and Applications Journal (MONET): Special Issue on Management of Mobility in Distributed Systems*, Volume 6, No. 2, March 2001, pp. 185-197. [pdf]
- [7] Q. Hu, W.-C. Lee, and D.L. Lee, A Hybrid Index Techniques for Power Efficient Data Broadcast, *Distributed and Parallel Databases*, Volume 9, No. 2, March 2001, pp. 151-177.
- [8] Q. Hu, W.-C. Lee, and D.L. Lee, Power Conserving and Access Efficient Indexes for Wireless Computing, *Information Organization and Databases*, edited by K. Tanaka, S. Ghandeharizadeh, and Y. Kambayashi, Kluwer Academic Publishers, Boston, MA, 2000, pp. 249-264.
- [9] Q. Hu, W.-C. Lee, and D.L. Lee, Power Conservative Multi-Attribute Queries on Data Broadcast, *IEEE International Conference on Data Engineering (ICDE'00)*, San Diego. Feb. 2000, pp. 157-166.

- [10] Q. Hu, W.-C. Lee, and D.L. Lee, Indexing Techniques for Wireless Data Broadcast Under Data Clustering and Scheduling, *ACM International Conference on Information and Knowledge Management (CIKM'99)*, Kansas City, Missouri, Nov. 1999, pp. 351-358.
- [11] W.-C. Lee, Q. Hu, and D.L. Lee, A Study of Channel Allocation Methods for Data Dissemination in Mobile Computing Environments, *ACM Mobile Networks and Applications Journal (MONET): Special Issue on Resource Management in Wireless Network*, Volume 4, No. 2, 1999, pp. 117-129.
- [12] W.-C. Lee and D.L. Lee, Signature Caching Techniques for Information Filtering in Mobile Environments, *ACM Wireless Networks Journal (WINET)*, Volume 5, No. 1, January 1999, pp. 57-67.
- [13] D.L. Lee, Q. Hu, and W.-C. Lee, Indexing Techniques for Data Broadcast on Wireless Channel, *the Fifth International Conference of Foundations of Data Organization (FODO'98)*, Kobe, Japan, Nov. 1998, pp. 175-182.
- [14] W.-C. Lee, and D.L. Lee, Using Signature Techniques for Information Filtering in Wireless and Mobile Environments, *Distributed and Parallel Databases: Special Issue on Database and Mobile Computing*, Vol. 4, No. 3, July 1996, pp. 205-227. (Also appears as a chapter in the book, *Databases and Mobile Computing*, edited by D. Barbara, R. Jain and N. Krishnakumar, Kluwer Academic Publishers, 1996, ISBN: 0792397495.)
- [15] W.-C. Lee, and D.L. Lee, On Signature Caching of Wireless Broadcast and Filtering Services, *the Second International Mobile Computing Conference*, Hsinchu, Taiwan, March 25-27, 1996, pp.15-24.
- [16] W.-C. Lee, and D.L. Lee, Information Filtering in Wireless and Mobile Environments, *IEEE International Phoenix Conference on Computers and Communications (IPCCC'96)*, Phoenix, Arizona, March 27-29,
- [17] B. Liu, W.-C. Lee, and D.L. Lee, Supporting Complex Multi-dimensional Queries in P2P Systems, *IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, Columbus, OH, June, 2005, pp. 155-164.

[18] M. Li, W.-C. Lee, and A. Sivasubramaniam, Semantic Small World: An Overlay Network for Peer-to-Peer Search, *IEEE International Conference on Network Protocols (ICNP 04)*, Berlin, Germany, October, 2004, pp. 228-238.

[19] M. Li, W.-C. Lee, and A. Sivasubramaniam, Efficient peer to peer information sharing over mobile ad hoc networks, *the Second WWW Workshop on Emerging Applications for Wireless and Mobile Access (MobEA 04)*, New York City, NY, May 2004.

[20] M. Li, W.-C. Lee, and A. Sivasubramaniam, Neighborhood Signatures for Searching P2P Networks, *International Database Engineering and Application Symposium (IDEAS 03)*, Hong Kong, July 16- 18, 2003, pp. 142-151.

[21] www.wikipedia.org

[22] www.google.com