



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Σύγκριση Αλγορίθμων Δημιουργίας Ευρετηρίων σε Πολλαπλά Ασύρματα Κανάλια
Εκπομπής

Comparison of Algorithms for Broadcast Indices in Wireless Multi-Channel
Environments

Διπλωματική Εργασία
Σταματίου Ορσαλία

Επιβλέποντες: Κατσαρός Δημήτριος
Επίκουρος Καθηγητής Π.Θ

Κοράκης Αθανάσιος
Επίκουρος Καθηγητής Π.Θ

Βόλος 2019

Στην οικογένεια μου και στους φίλους μου.

Ευχαριστίες

Με την περάτωση της παρούσας Διπλωματικής εργασίας θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Κατσαρό Δημήτριο για την εμπιστοσύνη που μου έδειξε και για τη συνεχή καθοδήγηση. Επίσης θα ήθελα να ευχαριστήσω τον συνεπιβλέποντα καθηγητή κ. Κοράκη Αθανάσιο. Τέλος, οφείλω ένα μεγάλο ευχαριστώ στην οικογένειά μου για την αμέριστη υποστήριξη και τη βοήθεια που μου παρείχαν καθ' όλη τη διάρκεια των σπουδών μου, καθώς και στους φίλους μου για τις αξέχαστες στιγμές.

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη τριών διαφορετικών αλγορίθμων ευρετηρίασης δεδομένων σε πολλαπλά ασύρματα κανάλια εκπομπής και η σύγκρισή τους σε ένα ενοποιημένο περιβάλλον. Πιο συγκεκριμένα, κατασκευάζονται τρεις αλγόριθμοι σε γλώσσα προγραμματισμού C και στη συνέχεια γίνεται ανάλυση της απόδοσής τους. Η καθυστέρηση πρόσβασης (Access Latency) και ο χρόνος συντονισμού (Tuning Time) είναι δύο βασικά κριτήρια για την αξιολόγηση της απόδοσης τέτοιων συστημάτων. Η καθυστέρηση πρόσβασης μετράται από τη στιγμή που ένας client θέτει ένα ερώτημα μέχρι τη στιγμή που αποκτά όλο το αποτέλεσμα. Ο χρόνος συντονισμού είναι ο χρόνος που ένας client είναι ενεργός στο κανάλι εκπομπής.

Abstract

The purpose of this diploma thesis is to develop three different index allocation algorithms for multiple broadcasts wireless channels and to compare them in a unified environment. More specifically, three algorithms are constructed in C programming language and then analyzing their performance. Access latency and tuning time are two main criteria to evaluate the performance of such systems. Access delay is measured from the moment a client asks for a query until it reaches the full result. Tuning time is the time a client is active on the broadcast channel.

Περιεχόμενα

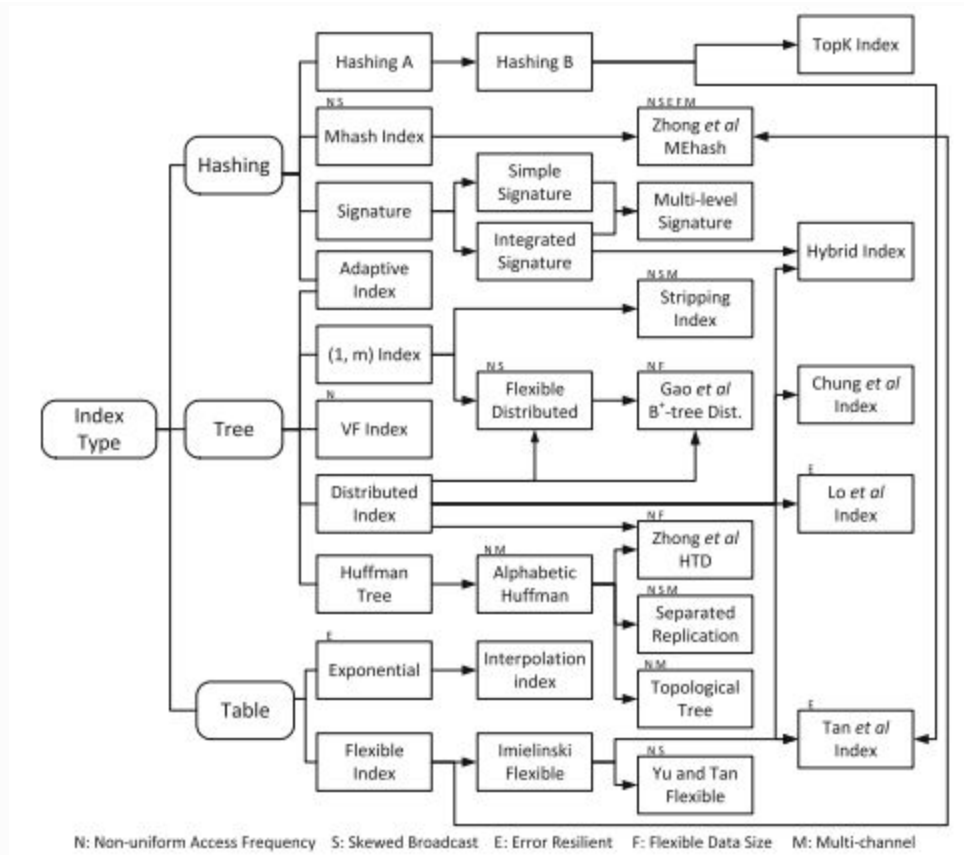
Ευχαριστίες	4
Περίληψη	6
Abstract	7
1.Εισαγωγή	10
2. Περιγραφή Προβλήματος	14
3. Αλγόριθμος 1	16
3.1 Αλγόριθμος Κατανομής Δεδομένων RDP	16
3.2 Αλγόριθμος TMBT	17
3.3 Αναζήτηση ενός δεδομένου	20
4. Αλγόριθμος 2	22
4.1 Κατανομή των δεδομένων στα κανάλια δεδομένων	23
4.2 Tree-Structure Index Allocation Method	23
4.3 Αναζήτηση ενός δεδομένου	25
5. Αλγόριθμος 3	26
5.1 Εξωτερικό Ευρετήριο (External Index)	26
5.2 Αναζήτηση ενός δεδομένου	27
6. Ανάλυση Απόδοσης	28
6.1 Αυξανόμενος αριθμός καναλιών μετάδοσης	29
6.2 Αυξανόμενος αριθμός δεδομένων	33
6.3 Συμπεράσματα	37
7. Σύνοψη	40
Βιβλιογραφία	42

1.Εισαγωγή

Οι πρόσφατες εξελίξεις στην ανάπτυξη φορητών υπολογιστών και ασύρματων δικτύων επικοινωνίας επιτρέπουν στους mobile clients να έχουν πρόσβαση σε δεδομένα από οπουδήποτε και ανά πάσα στιγμή. Η μετάδοση δεδομένων έχει γίνει μια ευρέως αποδεκτή προσέγγιση επικοινωνίας στο περιβάλλον του κινητού υπολογιστικού συστήματος. Παραδείγματα αυτών των εφαρμογών είναι οι προβλέψεις καιρού, το σύστημα διαχείρισης οδικής κυκλοφορίας και τα ηλεκτρονικά ενημερωτικά δελτία. Σε αυτές τις εφαρμογές, ένας server μεταδίδει περιοδικά ένα σύνολο δεδομένων σε ένα μεγάλο αριθμό από clients και οι clients συντονίζονται στο κανάλι εκπομπής για να ανακτήσουν τα δεδομένα που τους ενδιαφέρουν.

Η ευρετηρίαση είναι μια τεχνολογία που βοηθάει τους clients να μειώσουν το κόστος αναζήτησης και ανάκτησης δεδομένων. Ένα ευρετήριο είναι μια συγκεκριμένη δομή δεδομένων που περιέχει πληροφορίες σχετικά με τη χρονική μετατόπιση των δεδομένων σε ένα σύστημα μετάδοσης. Όταν ένας client ζητάει ένα συγκεκριμένο δεδομένο, αφού λάβει αυτή τη χρονική μετατόπιση, μπορεί να γνωρίζει πότε αλλά και σε ποιο κανάλι θα μεταδοθεί το δεδομένο που τον ενδιαφέρει.

Οι αλγόριθμοι ευρετηρίασης για συστήματα μετάδοσης δεδομένων μπορούν να ταξινομηθούν σε τρεις κατηγορίες: hashing, tree, και table-based schemes. Το “Σχήμα 1” απεικονίζει την ταξινόμηση ορισμένων από τα μεγαλύτερα υφιστάμενα συστήματα ευρετηρίασης. Οι Hashing-based αλγόριθμοι χρησιμοποιούν λειτουργίες κατακερματισμού και αποθηκεύουν πληροφορίες ευρετηρίου μέσα σε data buckets. Οι αλγόριθμοι που βασίζονται σε trees, όπως το alphabetic Huffman tree και το B+ tree, κατανέμουν τα δεδομένα στο αντίστοιχο δέντρο και το χωρίζουν σε αναπαραγόμενο και μη αναπαραγόμενο τμήμα ώστε το αναπαραγόμενο τμήμα να εκπέμπεται πιο συχνά. Οι table-based αλγόριθμοι διαιρούν τα δεδομένα σε τμήματα σύμφωνα με μία παράμετρο και αποθηκεύουν δείκτες σε αυτά τα δεδομένα σε έναν control table ώστε οι πελάτες να μπορούν να αναζητήσουν ένα δεδομένο σε έναν αυθαίρετο κόμβο. Ένα τέτοιο παράδειγμα είναι το Exponential Index.



Σχήμα 1: Συστήματα ευρετηρίασης

Η καθυστέρηση πρόσβαση (Access Latency) και ο χρόνος συντονισμού (Tuning Time) είναι δύο βασικά κριτήρια για την αξιολόγηση της απόδοσης ενός συστήματος μετάδοσης δεδομένων. Ως καθυστέρηση πρόσβασης υπολογίζεται ο χρόνος από τη στιγμή που ένας client θέτει ένα ερώτημα μέχρι τη στιγμή που έχει μεταδοθεί ολόκληρο το αποτέλεσμα. Ο client έχει δύο λειτουργίες: ενεργή λειτουργία(active mode) και αφύπνιση(doze mode). Μπορεί να βρίσκεται σε active mode όσο μεταδίδονται τα δεδομένα που τον ενδιαφέρουν και να επιστρέφει σε doze mode στην αντίθετη περίπτωση. Ως χρόνος συντονισμού υπολογίζεται ο χρόνος που ο client είναι σε active mode. Επομένως, η Access Latency αξιολογεί τον χρόνο απόκρισης ενός ερωτήματος σε ένα σύστημα και ο Tuning Time αξιολογεί την ενεργειακή απόδοση.

Στην παρούσα εργασία έγινε επιλογή τριών διαφορετικών αλγορίθμων ευρετηρίασης δεδομένων σε πολλαπλά ασύρματα κανάλια εκπομπής οι οποίοι θα συγκριθούν κάτω από τις ίδιες συνθήκες σε ένα ενοποιημένο περιβάλλον. Κάθε αλγόριθμος δέχεται ένα σύνολο δεδομένων καθένα από τα οποία περιγράφεται από τη συχνότητα μετάδοσής του. Στη συνέχεια, οι τρεις αλγόριθμοι χρησιμοποιούν ένα σχήμα ευρετηρίασης, B+ tree,

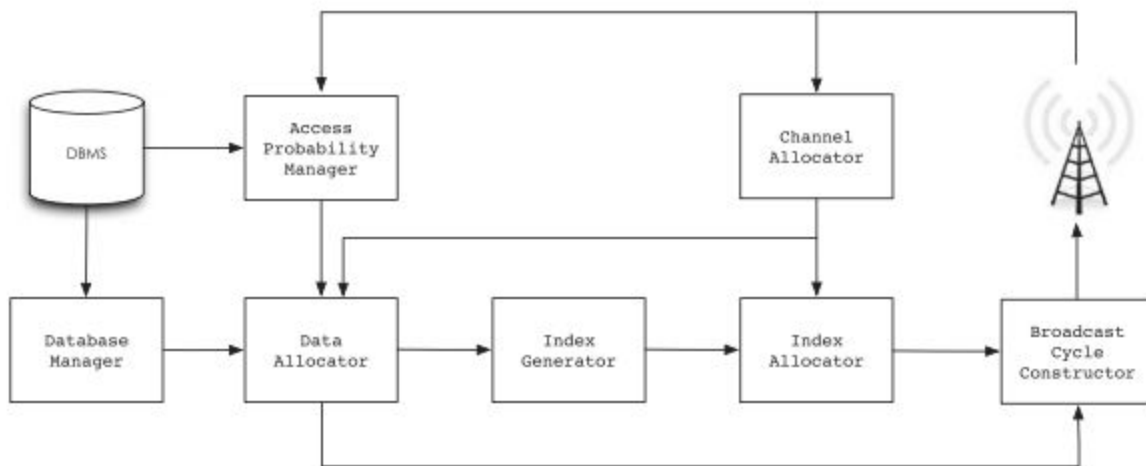
Huffman tree και exponential index αντιστοιχα, και κατανέμουν τα δεδομένα στα διαθέσιμα πολλαπλά κανάλια μετάδοσης για να μπορεί να γίνει γνωστό πότε και σε ποιο κανάλι μεταδίδονται τα δεδομένα. Ένας client μπορεί να έχει πρόσβαση στα δεδομένα που επιθυμεί αν συντονιστεί στο κατάλληλο κανάλι μετάδοσης. Τα επιστημονικά άρθρα που χρησιμοποιήθηκαν για την κατασκευή των αλγορίθμων αναφέρονται στη βιβλιογραφία [1,2,3]. Παρακάτω θα γίνει περιγραφή του κάθε αλγορίθμου.

2. Περιγραφή Προβλήματος

Η ασύρματη μετάδοση δεδομένων είναι μια αποτελεσματική τεχνολογία μετάδοσης δεδομένων σε έναν μαζικό αριθμό από mobile clients. Ένας server μεταδίδει περιοδικά πληροφορίες δημόσια σε κανάλια και κάθε mobile client μπορεί να έχει πρόσβαση στα κανάλια, να περιμένει τα απαιτούμενα δεδομένα και να τα λαμβάνει. Τα κριτήρια για την αξιολόγηση της απόδοσης ενός ασύρματου συστήματος μετάδοσης δεδομένων είναι ο χρόνος λήψης και η κατανάλωση ενέργειας των κινητών συσκευών. Αντίστοιχα, η access latency και ο tuning time είναι δύο ευρέως αποδεκτά πρότυπα αξιολόγησης του συστήματος όπως περιγράφηκε προηγουμένως. Η τεχνολογία ευρετηρίου και οι μέθοδοι κατανομής δεδομένων είναι οι πιο αποτελεσματικές μέθοδοι για τη μείωση της access latency και του tuning time. Με τα ευρετήρια οι clients μπορούν να ξέρουν πότε να βρίσκονται σε doze mode και πότε να επανέρχονται σε active mode όταν εμφανίζονται τα απαιτούμενα δεδομένα. Επιπλέον οι κατάλληλες μέθοδοι κατανομής μπορούν να κατανέμουν τα δεδομένα με τρόπο ώστε να μειώσουν τον χρόνο αναμονής των clients.

Στο άρθρο [4] της βιβλιογραφίας περιγράφεται ένα ενοποιημένο σύστημα μετάδοσης δεδομένων σε ένα κανάλι εκπομπής και γίνεται σύγκριση των πιο δημοφιλή αλγορίθμων ευρετηρίασης δεδομένων ώστε να βρεθεί ποιος είναι ο πιο αποδοτικός και υπό ποιες συνθήκες όσον αφορά την ελαχιστοποίηση της Access Latency και του Tuning Time. Στην εργασία αυτή θα δημιουργηθεί ένα παρόμοιο ενοποιημένο περιβάλλον μετάδοσης δεδομένων, αλλά για πολλαπλά ασύρματα κανάλια εκπομπής.

Η αρχιτεκτονική ενός συστήματος πολλαπλών καναλιών απεικονίζεται στο “Σχήμα 2”. Η ροή του συστήματος μετάδοσης δεδομένων μπορεί να περιγραφεί ως εξής: Πρώτον, ο Database Manager συλλέγει τα δεδομένα από το DBMS και κάθε στοιχείο έχει πιθανότητα πρόσβασης που παράγεται από το Access Probability Manager από ένα ιστορικό αρχείο. Στη συνέχεια, ο Allocator Channel εκχωρεί index και data channels. Στη συνέχεια, ο Index Generator θα δημιουργήσει ακολουθίες ευρετηρίων από ένα σύνολο δεδομένων και ο Index Allocator εκχωρεί αυτές τις ακολουθίες στα index channels, ενώ συγχρόνως, ο Data Allocator κατανέμει τα δεδομένα στα data channels. Τέλος, ο Broadcast Cycle Constructor συνδυάζει αυτά μαζί για να σχηματίσει ένα πλήρες πρόγραμμα εκπομπής.



Σχήμα 2: System Architecture

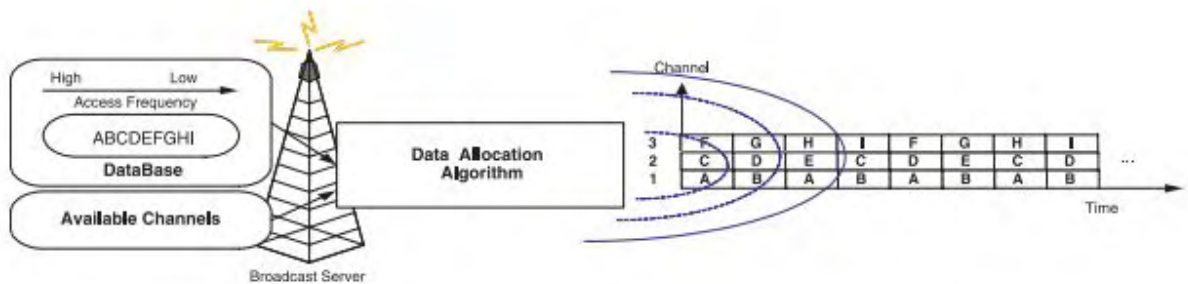
Οι μέθοδοι διάδοσης δεδομένων σε πολλαπλά κανάλια εκπομπής ποικίλουν. Η εργασία αυτή δημιουργεί ένα περιβάλλον προσομοίωσης και σύγκρισης αλγορίθμων με αριθμητικά πειράματα, ώστε να μπορεί να γίνει γνωστό ποιος και υπό ποιες συνθήκες είναι ο πιο αποδοτικός. Μετά από έρευνα επιλέχθηκαν τρεις αλγόριθμοι οι οποίοι χρησιμοποιούν B+ tree, Huffman tree και Exponential Index αντίστοιχα, για να ευρετηριάσουν τα δεδομένα προς μετάδοση. Αυτό το μοντέλο επικοινωνίας μπορεί εύκολα να τροποποιηθεί ώστε να προστεθούν αργότερα και άλλοι νέοι αλγόριθμοι. Τέλος, μπορεί να χρησιμοποιηθεί από τους παρόχους υπηρεσιών επικοινωνίας για να μπορούν να αξιολογήσουν και να επιλέξουν το καλύτερο για τα συστήματά τους.

3. Αλγόριθμος 1

Στον αλγόριθμο αυτό που ονομάζεται TMBT (time-multiplexed multiple binary trees), υποθέτουμε ότι ένα φυσικό κανάλι διατίθεται για το ευρετήριο (index channel) και τα υπόλοιπα φυσικά κανάλια έχουν εκχωρηθεί για δεδομένα (data channels). Αρχικά χρησιμοποιείται ένας αλγόριθμος που ονομάζεται RDP (restricted dynamic programming algorithm) και κατανέμει τα δεδομένα στα data channels ανάλογα με τη συχνότητα εμφάνισής τους. Στη συνέχεια, για κάθε data channel δημιουργείται ένα B+ tree και όλοι οι κόμβοι ευρετηρίου κάθε καναλιού εκχωρούνται σε ένα μοναδικό index channel. Όταν ένας client θέλει να ανακτήσει ένα δεδομένο πρέπει πρώτα να επισκεφτεί το index channels και να πάρει τις απαραίτητες πληροφορίες για να γνωρίζει σε ποιο data channel θα μεταδοθεί.

3.1 Αλγόριθμος Κατανομής Δεδομένων RDP

Ο RDP χρησιμοποιείται για τη διαίρεση N δεδομένων σε K data channels με στόχο την ελαχιστοποίηση της μέσης καθυστέρησης πρόσβασης των clients. Ένα παράδειγμα φαίνεται στο "Σχήμα 3". Ο αλγόριθμος δέχεται μια βάση δεδομένων με εννέα στοιχεία και τα κατανέμει σε τρία κανάλια εκπομπής ανάλογα με τη δημοτικότητα κάθε δεδομένου.

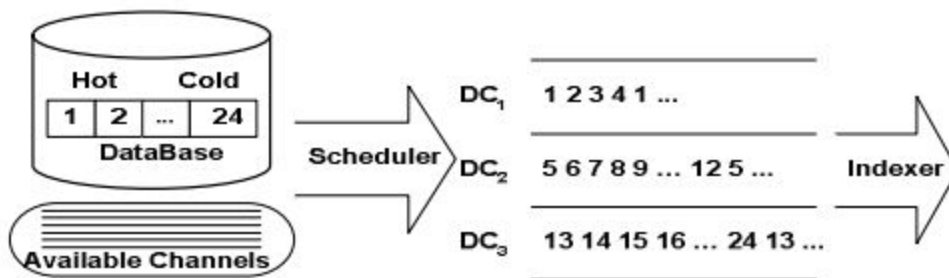


Σχήμα 3: Data allocation problem on multiple channels

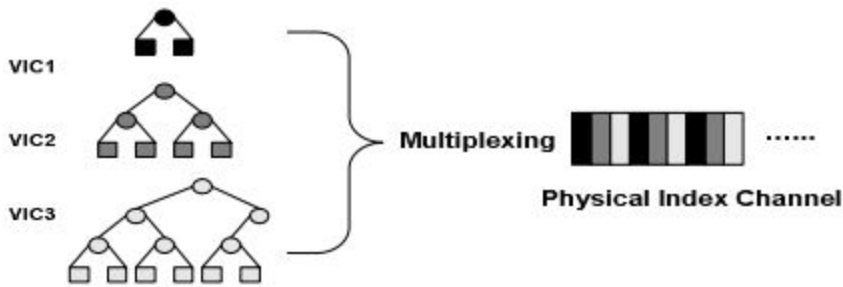
Αρχικά τα δεδομένα κατατάσσονται σε φθίνουσα σειρά με βάση τη συχνότητα εμφάνισής τους. Ο λόγος που γίνεται αυτή η κατάταξη είναι επειδή θέλουμε τα δεδομένα με υψηλότερες συχνότητες πρόσβασης να κατανεμηθούν σε κανάλια που περιέχουν λιγότερα στοιχεία. Ο αλγόριθμος υπολογίζει $K-1$ βέλτιστα σημεία κοπής της βάσης δεδομένων, όπως περιγράφεται στο [4], και έτσι χωρίζει τα δεδομένα στα K κανάλια.

3.2 Αλγόριθμος TMBT

Ο TMBT αρχικά δέχεται τα data channels που δημιουργήθηκαν από τον προηγούμενο αλγόριθμο. Για κάθε data channel DC_i δημιουργεί ένα δέντρο αναζήτησης ευρετηρίου που βασίζεται σε B+ tree και τα δεδομένα διατηρούνται στα φύλλα του δέντρου. Για κάθε DC_i , αναθέτουμε σε αυτό ένα VIC_i (εικονικό κανάλι ευρετηρίου) και αποθηκεύουμε σε αυτό τους κόμβους του B+ tree σε σειρά εμφάνισης “preorder” και έτσι δημιουργείται ένας εικονικός κύκλος εκπομπής ευρετηρίου (virtual index broadcast cycle) για το VIC_i . Στη συνέχεια, όλα τα εικονικά κανάλια VIC εκχωρούνται σε ένα φυσικό index channel. Η αρχιτεκτονική του συστήματος αυτού παρουσιάζεται στο “Σχήμα 4 (α) & (β)”.



(α)



(β)

Σχήμα 4 (α) & (β): System Architecture.

Όπως αναφέρθηκε παραπάνω, για την ευρετηρίαση των δεδομένων κάθε καναλιού χρησιμοποιείται ένα B+ tree με αριθμό κλάδων 2. Ένα B+ tree αποτελείται από τρεις τύπους κόμβων μετάδοσης: τη ρίζα (Root), το φύλλο(Leaf) και τον εσωτερικό κόμβο(Internal). Οι κόμβοι των φύλλων βρίσκονται στο χαμηλότερο επίπεδο του δέντρου, το καθένα από τα οποία περιέχει το πολύ δύο κόμβους δεδομένων. Στο “Σχήμα 5” παρουσιάζεται η δομή ενός κόμβου φύλλων και στο “Σχήμα 6” η δομή της ρίζας και του εσωτερικού κόμβου.

Leaf node

<i>NodeId</i>	<i>Type</i>	<i>Ncnum</i>
<i>key1</i>	<i>cnun</i>	<i>key2</i>
<i>seq1</i>	<i>Rootptr</i>	<i>seq2</i>

Σχήμα 5: Δομή κόμβου φύλλου

Root / Internal node

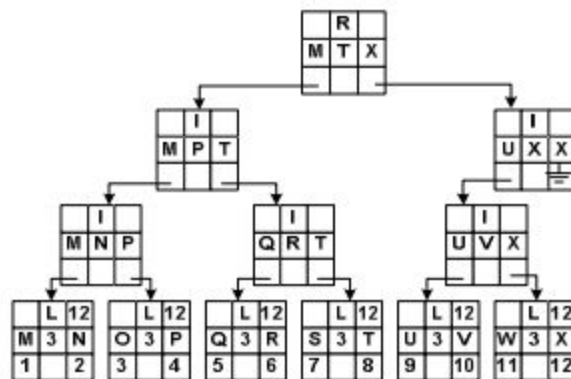
<i>NodeId</i>	<i>Type</i>	<i>unused</i>
<i>minkey</i>	<i>midkey</i>	<i>maxkey</i>
<i>Lptr</i>	<i>Rootptr</i>	<i>Rptr</i>

Σχήμα 6: Δομή κόμβου ρίζας και εσωτερικού κόμβου

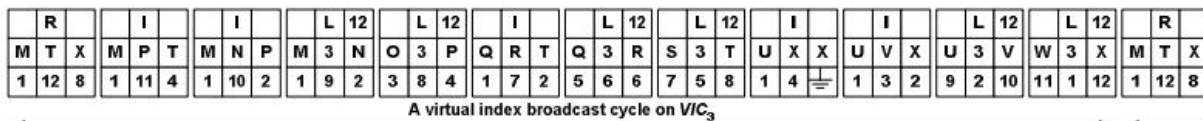
Παρακάτω περιγράφονται τα στοιχεία των δύο κόμβων:

- Type: ο τύπος του κόμβου (L: leaf, I: internal, R: root)
- Nodeld: η μετατόπιση από την αρχή του global broadcast cycle στον τρέχοντα κόμβο.
- Cnum: ο αριθμός του καναλιού δεδομένων
- Ncnum: το μέγεθος του data broadcast cycle στο DCnum
- key1 και key2: τα δύο δεδομένα του κόμβου
- Seq1 και seq2: οι θέσεις των key1 και key2 στον data broadcast cycle στο DCnum
- [minkey, maxkey]: το key range του subtree που έχει ρίζες σε αυτόν τον κόμβο
- Lptr: η μετατόπιση από τον τρέχοντα κόμβο στο αριστερό του παιδί
- Rptr: η μετατόπιση από τον τρέχοντα κόμβο στο δεξιό του παιδί
- Rootptr: η μετατόπιση από τον τρέχοντα κόμβο στην αρχή του επόμενου virtual index broadcast cycle

Στο “Σχήμα 7” παρουσιάζεται ένα παράδειγμα B+ tree του καναλιού 3 (DC3) του σχήματος 2 και στο “Σχήμα 8” ο virtual index broadcast cycle για το ίδιο κανάλι.



Σχήμα 7: B+ tree για το κανάλι DC3



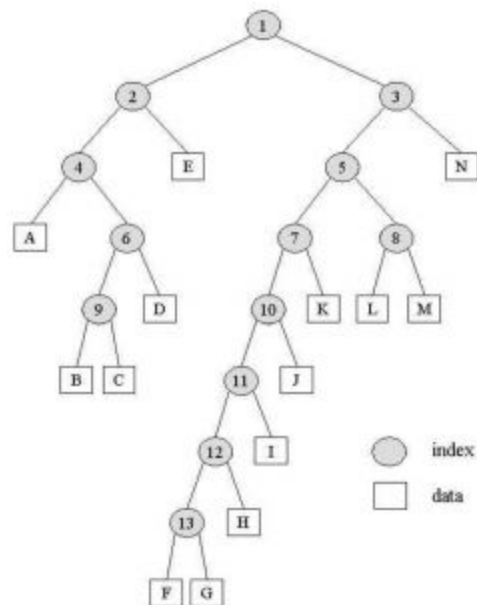
Σχήμα 8: virtual index broadcast cycle για το κανάλι DC3

3.3 Αναζήτηση ενός δεδομένου

Αρχικά το πρόγραμμα δέχεται ένα σύνολο δεδομένων με τις συχνότητες εμφάνισής τους. Στη συνέχεια ζητάει από το χρήστη να δώσει τον αριθμό $K-1$ των data channels, αφού στο συγκεκριμένο αλγόριθμο $K-1$ κανάλια θα είναι data channels και 1 κανάλι αποδίδεται για τον index. Στη συνέχεια θα κατατάξει τα δεδομένα σε φθίνουσα σειρά και θα δημιουργήσει τα κανάλια και τον τελικό broadcast cycle όπως περιγράφεται παραπάνω. Όταν ένας client ζητήσει ένα δεδομένο συντονίζεται στο index channel και ελέγχει τους $K-1$ κόμβους ρίζας για να βρει σε ποιο κανάλι πρέπει να συντονιστεί. Στη συνέχεια περιμένει να έρθει το επιθυμητό κανάλι και μέσα από τους εσωτερικούς κόμβους βρίσκει αν πρέπει να ψάξει αριστερά η δεξιά αυτού του κόμβου. Ακολουθεί αυτή την τακτική μέχρι να βρει τον επιθυμητό κόμβο φύλλου που είναι αποθηκευμένο το δεδομένο που αναζητεί.

4. Αλγόριθμος 2

Ο αλγόριθμος αυτός είναι μια tree-structured index allocation method που χωρίζει τα διαθέσιμα κανάλια μετάδοσης σε data και index channels. Δημιουργεί αρχικά ένα αλφαβητικό δέντρο Huffman όπου τα φύλλα του αναφέρονται ως κόμβοι δεδομένων και η ρίζα με τους εσωτερικούς κόμβους ως κόμβοι ευρετηρίου. Οι κόμβοι ευρετηρίου δημιουργούνται αναδρομικά συνδυάζοντας τους πρώτους δύο κόμβους δεδομένων με τη μικρότερη συχνότητα μετάδοσης. Ο νέος κόμβος θα έχει συχνότητα ίση με το άθροισμα των συχνοτήτων αυτών των δύο κόμβων. Τα δεδομένα με μεγαλύτερη συχνότητα μετάδοσης βρίσκονται στα πιο υψηλά επίπεδα του δέντρου, ενώ αυτά με μικρότερη συχνότητα μετάδοσης στα χαμηλότερα. Αυτή η ιδιότητα καθιστά τα δεδομένα στα υψηλά επίπεδα να έχουν λιγότερους tuning και access times από εκείνα στα χαμηλότερα. Ένα παράδειγμα αλφαβητικού δέντρου Huffman παρουσιάζεται στο “Σχήμα 9”. Το ‘Ε’ και το ‘Ν’ έχουν μεγαλύτερη συχνότητα μετάδοσης από τα υπόλοιπα δεδομένα. Οι κόμβοι δεδομένων και ευρετηρίου μεταδίδονται στα data και index channels αντίστοιχα.



Σχήμα 9: An example of Alphabetic Huffman tree.

4.1 Κατανομή των δεδομένων στα κανάλια δεδομένων

Τα δεδομένα αρχικά ταξινομούνται σε φθίνουσα σειρά σύμφωνα με τη συχνότητα μετάδοσής τους. Στη συνέχεια, υπολογίζεται η συνολική συχνότητα πρόσβασης όλων των δεδομένων και διαιρείται με τον αριθμό των διαθέσιμων data channels. Έτσι υπολογίζεται μία μέση συχνότητα πρόσβασης κάθε καναλιού που αντιπροσωπεύει την χωρητικότητά του. Ο αλγόριθμος κατανέμει ένα δεδομένο σε ένα κανάλι εάν το άθροισμα των συχνοτήτων του καναλιού είναι μικρότερο από τη χωρητικότητα του καναλιού. Με τον τρόπο αυτό τα πιο υψηλά κανάλια θα περιέχουν λιγότερα δεδομένα σε σχέση με τα χαμηλότερα. Τα δεδομένα στο κανάλι i θα είναι δεδομένα με μεγαλύτερη συχνότητα από αυτά στο κανάλι $i+1$.

4.2 Tree-Structure Index Allocation Method

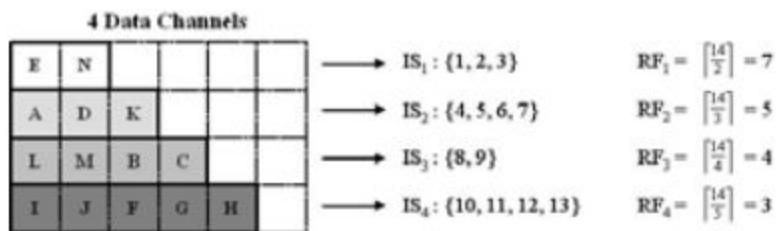
Έστω το DC i να είναι το i -data channel. Το IS 1 δηλώνει το πρώτο σύνολο ευρετηρίου που περιέχει την ακολουθία των κόμβων ευρετηρίου κατά μήκος της διαδρομής από τη ρίζα ενός Huffman tree στα δεδομένα του DC 1 . Στη συνέχεια, το IS i ($2 \leq i \leq$ ο αριθμός των data channels) αντιπροσωπεύει το i -σύνολο ευρετηρίου που περιέχει την ακολουθία των κόμβων ευρετηρίου κατά μήκος της διαδρομής από τη ρίζα ενός Huffman tree στα δεδομένα του DC i , εξαιρουμένων όλων των κόμβων ευρετηρίου στο IS j για $j = 1, i-1$. Οι κόμβοι ευρετηρίου στο IS i δείχνουν σε κόμβους δεδομένων που περιέχουν δεδομένα με μεγαλύτερη συχνότητα από εκείνους στο IS $i + 1$. Επιπλέον, οι κόμβοι ευρετηρίου στο IS i πρέπει να μεταδίδονται τόσο συχνά όσο τα αντίστοιχα δεδομένα στο DC i . Για αυτό, υπολογίζεται η συχνότητα επανάληψης RFi, ώστε το IS i να μεταδοθεί σε ένα μόνο index broadcast cycle, που είναι ίση με " $RF_i = \frac{DB}{DX_i}$ ", όπου DB είναι το σύνολο όλων των δεδομένων και DX i το σύνολο των δεδομένων στο i data channel. Από το IS i , μια ακολουθία των υποσυνόλων S $i1, S_{i2}, \dots, S_{in}$ ορίζεται ως το index block IB i για IS i . Και το S ip ονομάζεται ως το p μέλος του index block IB i όπου $1 \leq p \leq n$. Το index block IB i πρέπει να πληροί τις ακόλουθες απαιτήσεις:

- $IS_i = \bigcup_{j=1}^n S_{ij}$
- $S_{ip} \cap S_{iq} = 0$, για $1 \leq p, q \leq n$ και $p \neq q$
- Όλοι οι κόμβοι ευρετηρίου στο S ip δεν θα πρέπει να σχετίζονται μεταξύ τους με σχέση προγόνου - απογόνου

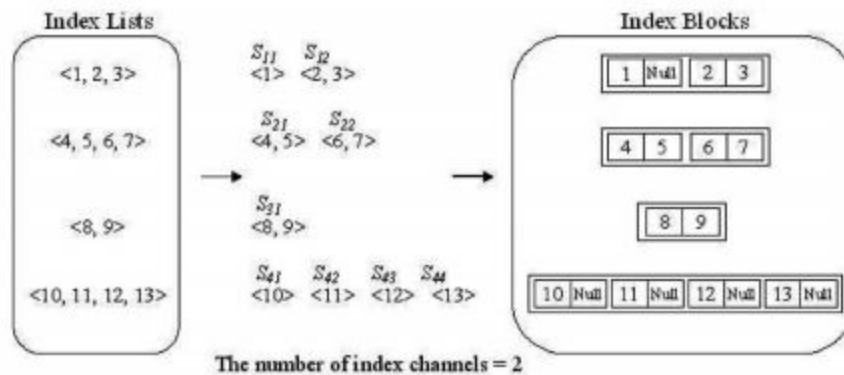
- Για $1 \leq p \leq n$, κάνουμε το $|S_{ip}|$ να είναι ίσο με k (k =number of index channels), όπου $|S_{ip}|$ αντιπροσωπεύει το μέγεθος του S_{ip} . Αν $|S_{ip}| < k$, τοποθετήστε ένα null κόμβο ευρετηρίου στο S_{ip} έως ότου το μέγεθός του να γίνει k .

Στη συνέχεια για κάθε Index Block και για κάθε $S_{ip} = \{i_1, i_2, \dots, i_k\}$ κατανέμεται ο κόμβος ευρετηρίου in ($1 \leq n \leq k$) στο index channel n .

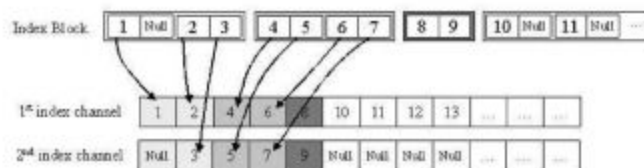
Τα Σχήματα 10, 11 και 12 δείχνουν ένα παράδειγμα αυτής της διαδικασίας που βασίζεται στο Huffman tree του Σχήματος 7 για τέσσερα data channels και δύο index channels.



Σχήμα 10: Δημιουργία συνόλων IS



Σχήμα 11: Δημιουργία index blocks



Σχήμα 12: Δημιουργία index channels

4.3 Αναζήτηση ενός δεδομένου

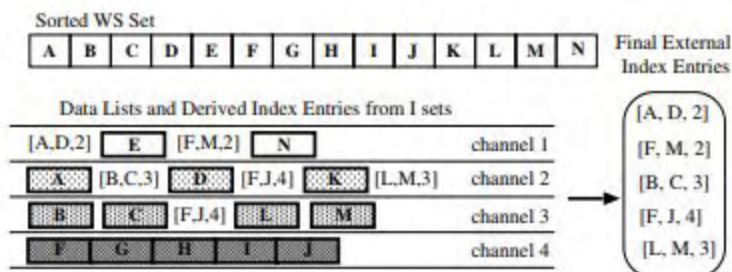
Αρχικά το πρόγραμμα δέχεται ένα σύνολο δεδομένων με τις συχνότητες εμφάνισής τους και τα κατανέμει σε ένα αλφαβητικό Huffman tree. Στη συνέχεια ζητάει από το χρήστη να δώσει τον αριθμό M των data channels και τον αριθμό N των index channels. Αξίζει να σημειωθεί ότι ο αριθμός N των index channels πρέπει να είναι μικρότερος από το βάθος του δέντρου. Στη συνέχεια θα κατατάξει τα δεδομένα σε φθίνουσα σειρά και θα δημιουργήσει τα κανάλια και τον τελικό broadcast cycle όπως περιγράφεται παραπάνω. Όταν ένας client ζητάει ένα δεδομένο, αρχικά συντονίζεται στα index channels και αναζητεί τον κόμβο ευρετηρίου που είναι “γονικός κόμβος” του κόμβου δεδομένων που αναζητεί. Στη συνέχεια αφού λάβει τις απαραίτητες πληροφορίες μπαίνει σε doze mode και περιμένει να μεταδοθεί το data channel που περιέχει αυτό το δεδομένο.

5. Αλγόριθμος 3

Ο αλγόριθμος αυτός δεν διαχωρίζει τα διαθέσιμα κανάλια σε data και index channels όπως οι προηγούμενοι δύο αλγόριθμοι. Δέχεται ένα σύνολο δεδομένων και τα κατατάσσει σε φθίνουσα σειρά με βάση τη συχνότητα μετάδοσής τους και στη συνέχεια τα κατανέμει σε data channels με την ίδια διαδικασία που χρησιμοποιεί ο Αλγόριθμος 2. Για να μπορεί ο client να αναζητήσει ένα δεδομένο, δημιουργούνται κόμβοι εξωτερικού ευρετηρίου που κατανέμονται και αυτοί στα data channels. Ο client πρώτα ελέγχει αυτούς τους κόμβους για να βρει σε ποιο κανάλι μεταδίδεται το δεδομένο.

5.1 Εξωτερικό Ευρετήριο (External Index)

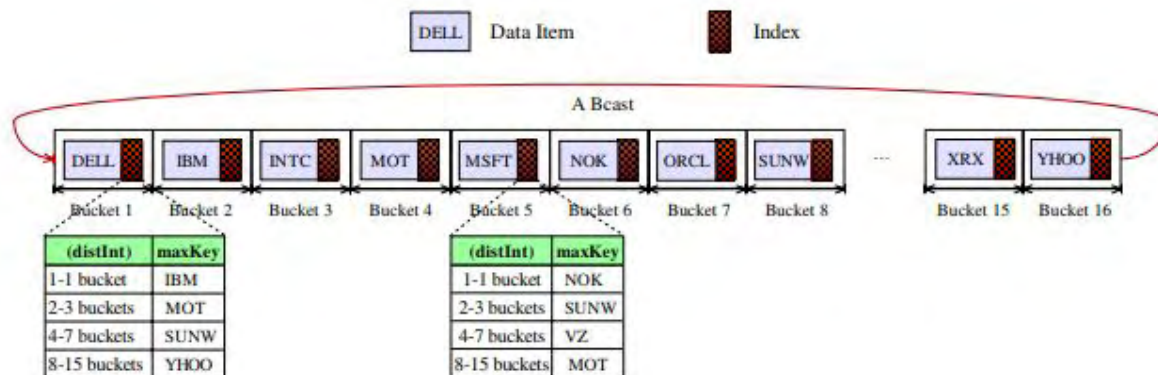
Ο σκοπός του External Index είναι να επιτρέπει στους clients να ανακαλύπτουν γρήγορα σε ποιο κανάλι εκπέμπονται τα δεδομένα που τους ενδιαφέρουν. Αρχικά όλα τα δεδομένα κατατάσσονται σε αύξουσα σειρά με βάση τα index keys τους και το ίδιο γίνεται και με τα δεδομένα σε κάθε data channel. Στη συνέχεια για κάθε data channel, εκτός του τελευταίου, δημιουργούνται οι external index οι οποίοι περιέχουν τα σύνολα των δεδομένων που παρεμβάλλονται ανάμεσα στα δεδομένα του. Για να ελέγξει ένας client αν το δεδομένο που αναζητά ανήκει σε ένα κανάλι, ελέγχει πρώτα τα διαστήματα που έχουν δημιουργηθεί στους external index αυτού του καναλιού. Αν το δεδομένο δεν ανήκει σε αυτά τα διαστήματα τότε μεταδίδεται σε αυτό το κανάλι. Ένα παράδειγμα αυτής της διαδικασίας απεικονίζεται στο “Σχήμα 13” για το σύνολο δεδομένων: $WS = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N\}$.



Σχήμα 13: Δημιουργία External Index

5.2 Αναζήτηση ενός δεδομένου

Όταν ο client θέτει ένα ερώτημα ελέγχει αρχικά τους External Index για να εντοπίσει το επιθυμητό data channel. Στη συνέχεια για την αναζήτηση ενός δεδομένου σε ένα data channel ο αλγόριθμος χρησιμοποιεί τη μέθοδο του εκθετικού ευρετηρίου (Exponential Index). Για κάθε δεδομένο δημιουργείται ένα “bucket” που περιέχει το “κλειδί” του δεδομένου αυτού και έναν πίνακα ελέγχου (control table). Ο πίνακας ελέγχου περιέχει τα δεδομένα που βρίσκονται σε απόσταση: $1, r, r^2, \dots, r^n$, όπου το r ονομάζεται index base και σε αυτή την περίπτωση θα θεωρήσουμε ότι είναι ίσο με 2. Όταν μεταδίδεται ένα δεδομένο, ο αλγόριθμος μπορεί να ελέγξει τα δεδομένα που βρίσκονται σε αποστάσεις $1, r, r^2, \dots, r^n$ από αυτό το δεδομένο ώστε να μπορεί ο client να γνωρίζει πότε θα μεταδοθεί το επιθυμητό δεδομένο και να αφυπνίζεται όσο μεταδίδονται τα υπόλοιπα δεδομένα. Στον αλγόριθμό μας θεωρούμε ότι το n είναι ίσο με τον αριθμό των data channels. Ένα παράδειγμα αναζήτησης exponential index για ένα κανάλι παρουσιάζεται στο “Σχήμα 14” όπου ο client έχει ζητήσει το δεδομένο με κλειδί “NOK” ακριβώς πριν από την εκπομπή του “DELL”. Ο πελάτης συντονίζεται στο data channel και αρχικά ανακτά τον control table στο bucket 1. Επειδή το “NOK” πέφτει μεταξύ του “MOT” και του “SUNW”, πρέπει να βρίσκεται στους κάδους που είναι 4 έως 7 θέσεις μακριά. Ο client στη συνέχεια παραμένει στη doze mode μέχρι να μεταδοθεί το bucket 5 και εξετάζει το δεδομένο στο bucket 5. Καθώς το δεδομένο που θέλει δεν μπορεί να βρεθεί στο bucket 5, ο client ελέγχει περαιτέρω τον control table στο bucket 5. Εφόσον το “NOK” ταιριάζει με το πρώτο δεδομένο του control table, πρέπει να βρίσκεται στο επόμενο bucket, δηλαδή στο bucket 6.



Σχήμα 14: Παράδειγμα αναζήτησης Exponential Index

6. Ανάλυση Απόδοσης

Σε αυτή την ενότητα, θα γίνει ανάλυση και σύγκριση της απόδοσης των τριών αλγορίθμων σε ένα ενοποιημένο περιβάλλον με το ίδιο σύνολο δεδομένων και τον ίδιο αριθμό ασύρματων καναλιών. Οι αλγόριθμοι υλοποιήθηκαν σε γλώσσα προγραμματισμού C σε λειτουργικό Linux. Ο προσομοιωτής λειτουργεί ως εξής: Για κάθε αλγόριθμο ευρετηρίασης, ο προσομοιωτής μας δημιουργεί πρώτα την ακολουθία εκπομπής και τις εκχωρεί στα κανάλια εκπομπής. Στη συνέχεια, θα δημιουργήσει μια σειρά τυχαίων αιτήσεων. Για κάθε αλγόριθμο γίνεται αναζήτηση του κάθε δεδομένου και μετριέται η μέση καθυστέρηση πρόσβασης (AAL) και ο μέσος χρόνος συντονισμού (ATT) για όλα τα αιτήματα που δημιουργήθηκαν σε “time units”.

Θεωρούμε ότι ο client συντονίζεται τη χρονική στιγμή 0. Όλοι οι κόμβοι έχουν μέγεθος 1KB. Τα κανάλια που εξετάζονται υποτίθεται ότι είναι όλα συμμετρικά και φυσικά κανάλια. Ο χρόνος για τη μετάβαση από ένα κανάλι στο άλλο θεωρείται αμελητέος. Η Average Access Latency είναι ο χρόνος από τη στιγμή που ο client θέτει ένα ερώτημα μέχρι τη στιγμή που έχει λάβει ολόκληρο το αποτέλεσμα. Ο Average Tuning Time είναι ο χρόνος που ο client βρίσκεται σε active mode.

Για απλότητα θα χρησιμοποιηθούν για τους αλγόριθμους οι ακόλουθες συντομογραφίες: “TMBT” για τον Αλγόριθμο 1, “Huffman” για τον Αλγόριθμο 2 και “Exponential” για τον Αλγόριθμο 3. Στον “Πίνακα 1” παρουσιάζονται οι διάφορες παράμετροι που χρησιμοποιούνται στις προσομοιώσεις.

System Parameters	Range
Database Size	50-300
Total number of Simulations	30
Number of Data Channels	2-10

Πίνακας 1

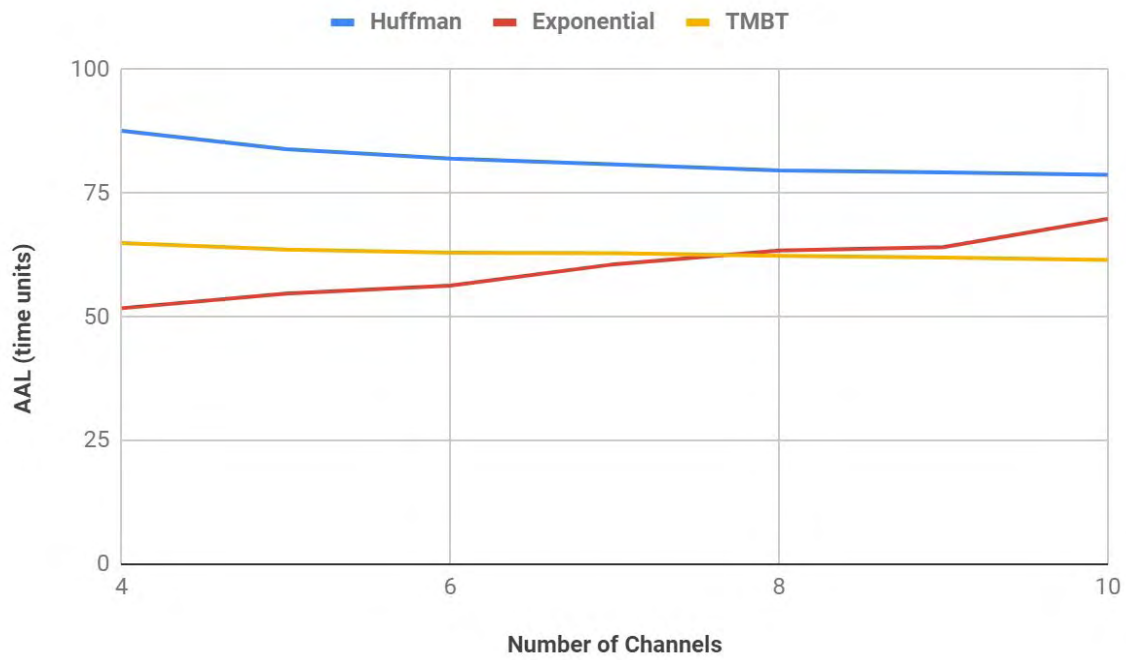
6.1 Αυξανόμενος αριθμός καναλιών μετάδοσης

Θα γίνει ανάλυση της απόδοσης των αλγορίθμων για αυξανόμενο αριθμό των καναλιών εκπομπής για 50 δεδομένα. Στον “Πίνακα 2” φαίνονται αναλυτικά οι διάφορες παράμετροι. Αρχικά θεωρούμε ότι ο αλγόριθμος “Huffman” και ο αλγόριθμος “TMBT” χωρίζουν τα διαθέσιμα κανάλια σε index και data channels. Ο “Huffman” θα έχει δύο index channels και ο “TMBT” από τον ορισμό του έχει ένα index channel, αλλά συνολικά θα έχουν και οι τρεις αλγόριθμοι ίσο πλήθος καναλιών μετάδοσης.

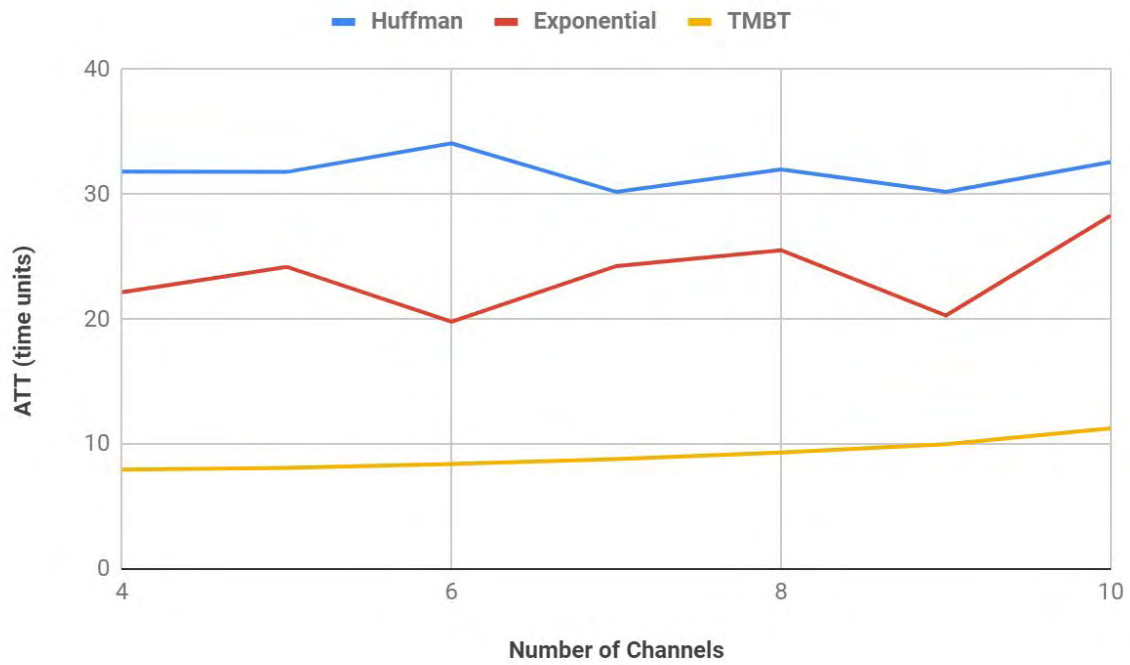
System Parameters	Range
Database Size	50
Total number of Simulations	30
Number of Data Channels	2-10

Πίνακας 2

Στα διαγράμματα 1 και 2 παρουσιάζεται η Average Access Latency και ο Average Tuning Time για 4 έως 10 κανάλια εκπομπής:



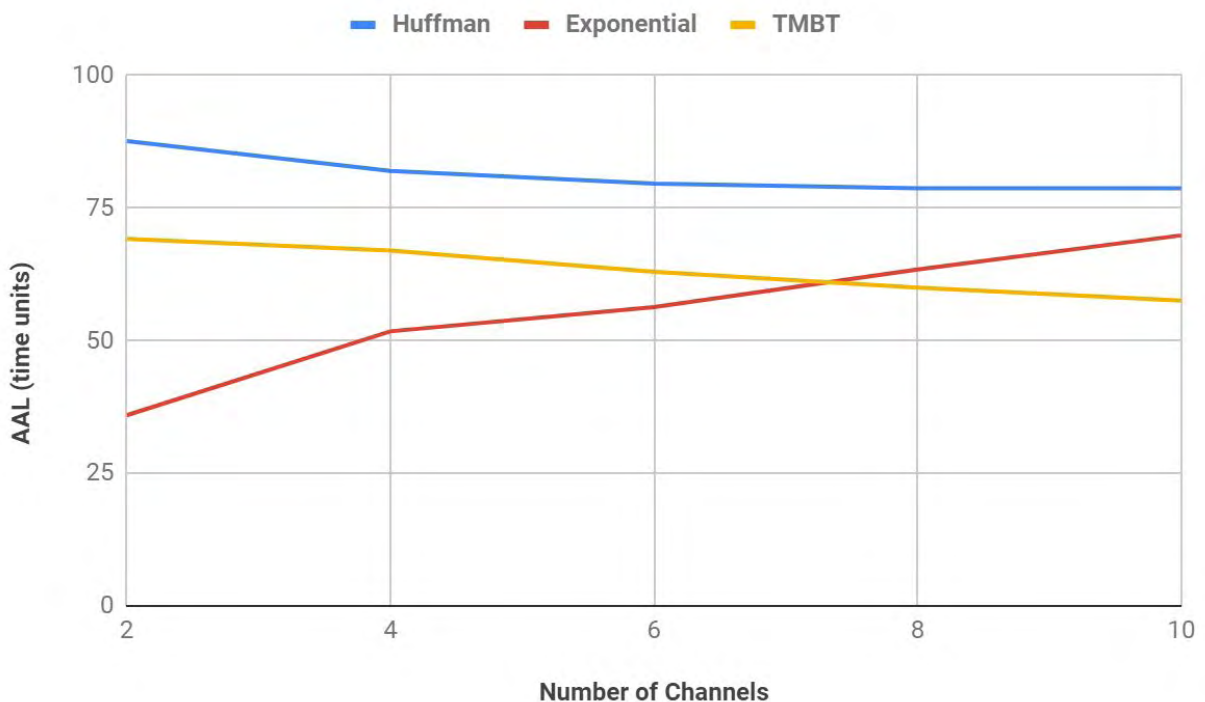
Διάγραμμα 1: Average Access Latency



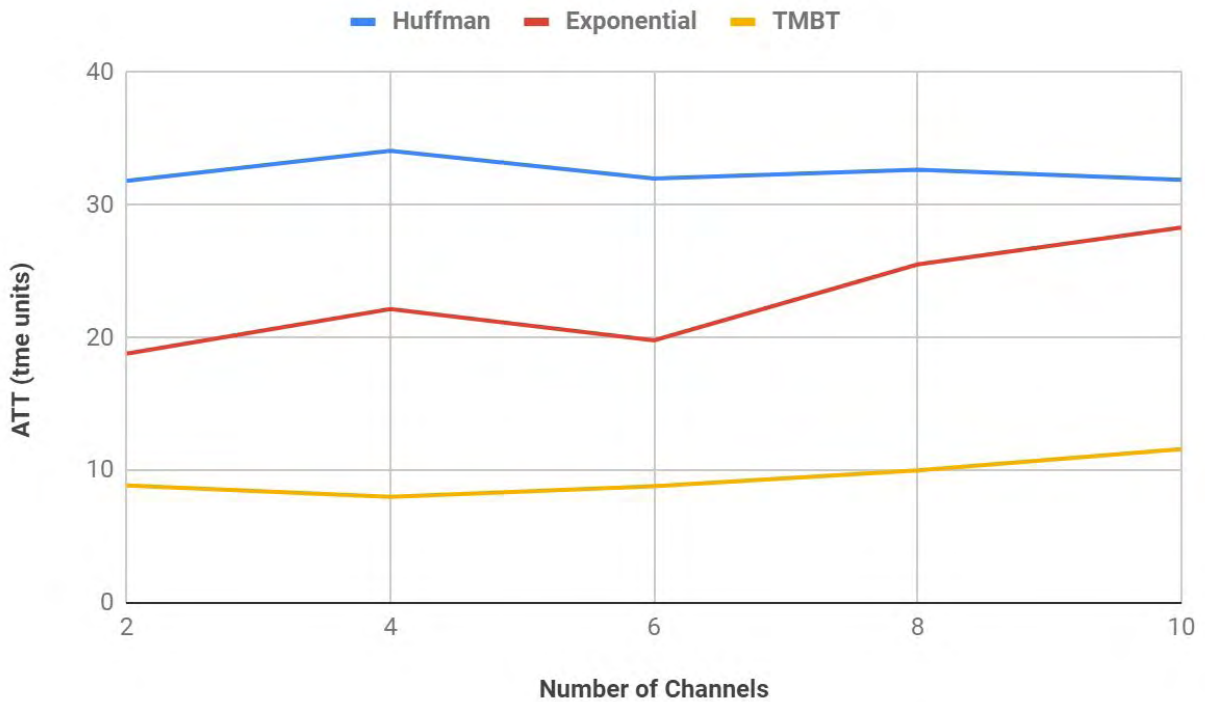
Διάγραμμα 2: Average Tuning Time

Καθώς αυξάνονται τα κανάλια εκπομπής παρατηρείται μείωση στη Average Access Latency για τον “TMBT” και τον “Huffman”. Αντίθετα ο “Exponential” παρουσιάζει αύξηση. Όσον αφορά τον Average Tuning Time καλύτερη επίδοση παρουσιάζει ο “TMBT”.

Στα διαγράμματα 3 και 4 παρουσιάζεται η απόδοση των αλγορίθμων στην περίπτωση που ο “TMBT” και ο “Huffman” δεν διαχωρίζουν τα κανάλια εκπομπής σε index και data αλλά έχουν επιπλέον 1 και 2 index channels αντίστοιχα. Οι μετρήσεις έγιναν για 2 έως 10 κανάλια εκπομπής.



Διάγραμμα 3: Average Access Latency



Διάγραμμα 4: Average Tuning Time

Στην περίπτωση αυτή, ενώ αρχικά ο “Exponential” παρουσιάζει καλύτερη επίδοση όσον αφορά την Average Access Latency, από τα 7 κανάλια και μετά καλύτερη επίδοση έχει ο “TMBT” όπου παρουσιάζει μείωση. Ο “Huffman” είναι σχετικά σταθερός. Και σε αυτή την περίπτωση αποδοτικότερο Average Tuning Time παρουσιάζει ο “TMBT”.

6.2 Αυξανόμενος αριθμός δεδομένων

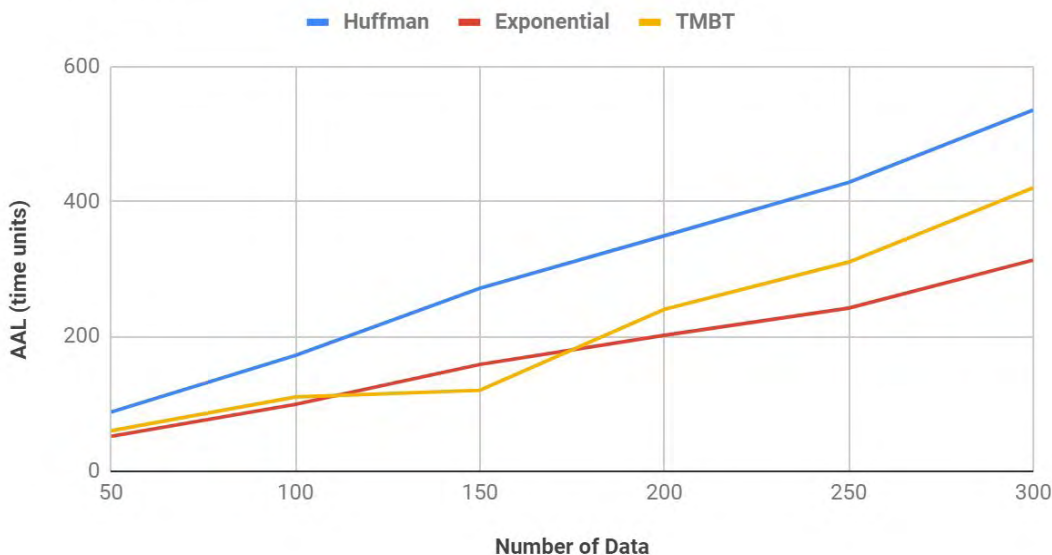
Στην ενότητα αυτή θα γίνει ανάλυση της απόδοσης των τριών αλγορίθμων όταν το πλήθος των δεδομένων προς μετάδοση αυξάνεται. Πραγματοποιούνται 3 διαφορετικά πειράματα. Τα δεδομένα που θα χρησιμοποιηθούν παρουσιάζονται στον “Πίνακα 3”.

System Parameters	Range
Database Size	50 - 300
Total number of Simulations	30
Number of Data Channels	4, 10

Πίνακας 3

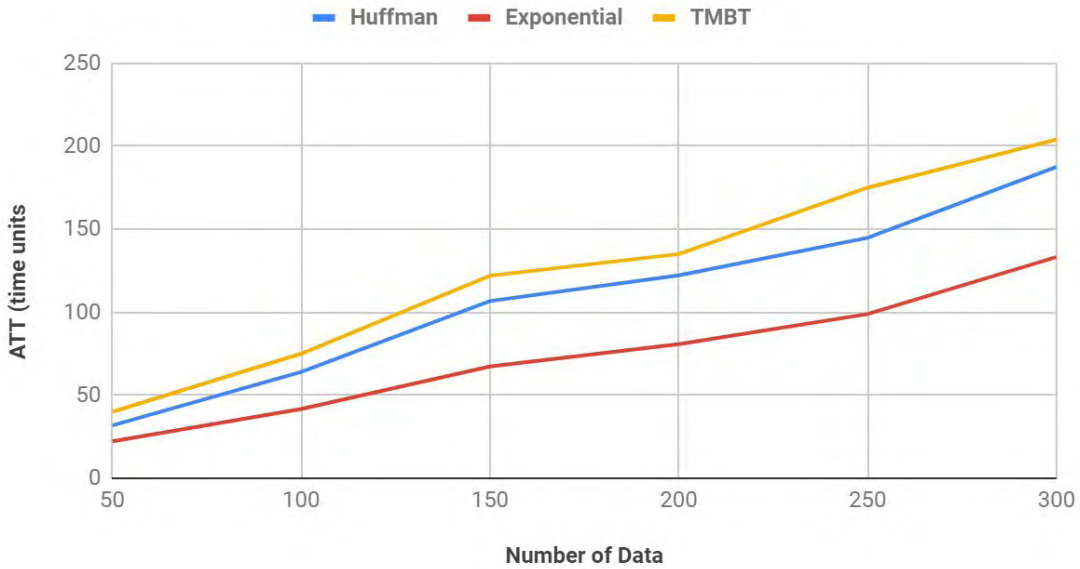
Στα Διαγράμματα 5 και 6 παρουσιάζεται η απόδοση των αλγορίθμων για “4 κανάλια εκπομπής”.

4 Channels



Διάγραμμα 5: Average Access Latency: 4 data channels

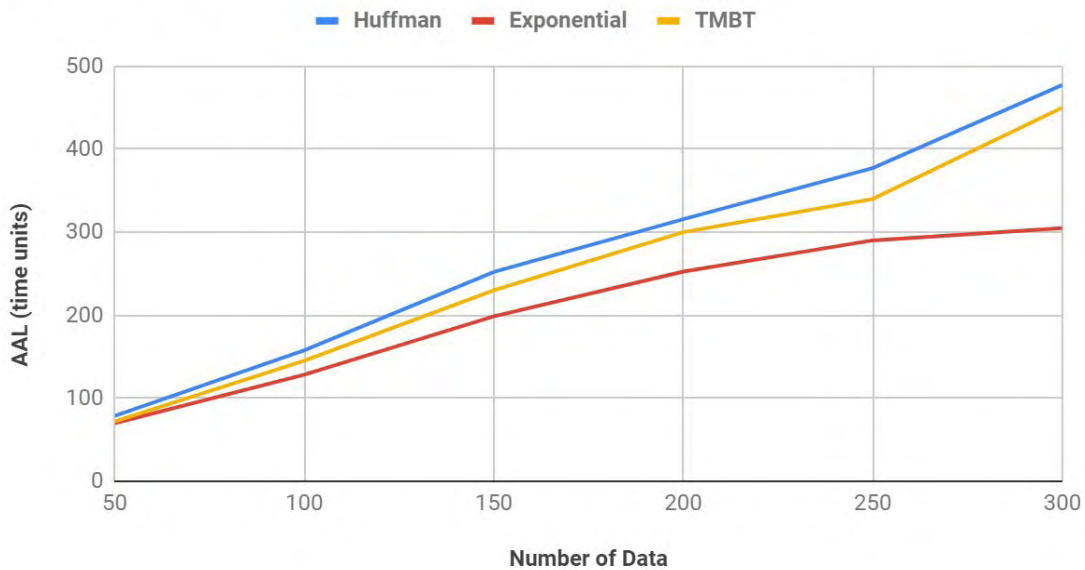
4 Channels



Διάγραμμα 6: Average Tuning Time: 4 data channels

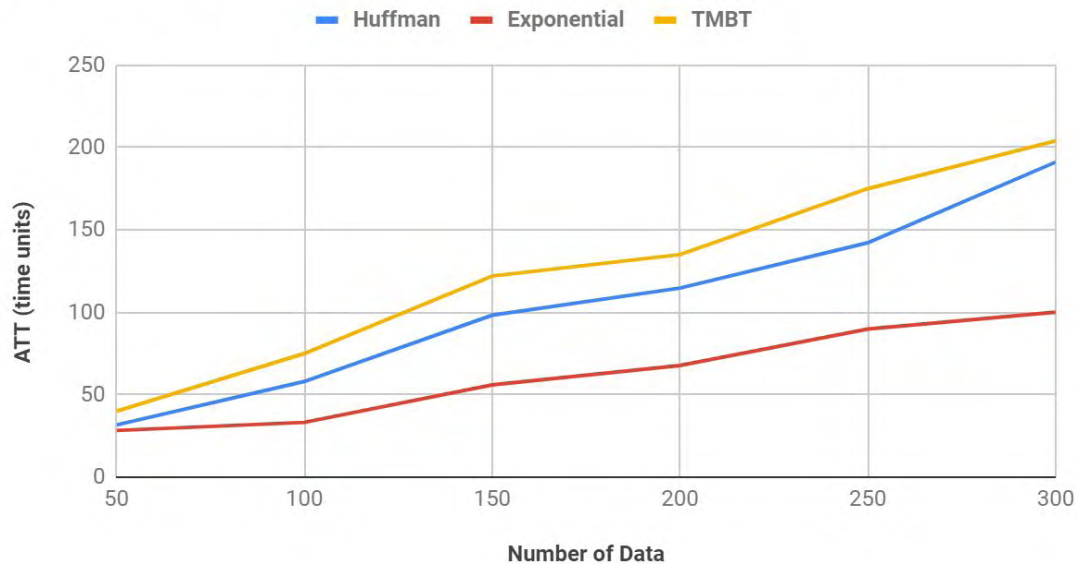
Στα Διαγράμματα 7 και 8 παρουσιάζεται η απόδοση των αλγορίθμων για “10 κανάλια εκπομπής”.

10 Channels



Διάγραμμα 7: Average Access Latency: 10 data channels

10 Channels

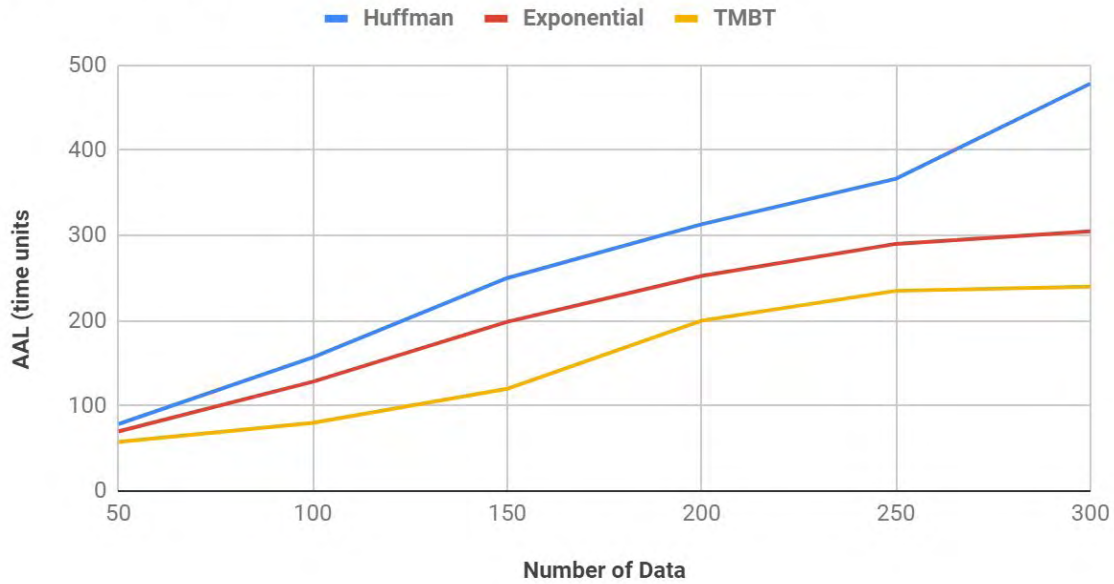


Διάγραμμα 8: Average Tuning Time: 10 data channels

Καθώς τα δεδομένα μετάδοσης αυξάνονται καλύτερη επίδοση παρουσιάζει ο “Exponential” διότι και στις δύο περιπτώσεις έχει χαμηλότερη Average Access Latency και χαμηλότερο Average Tuning Time. Όσον αφορά τους άλλους δύο αλγορίθμους, ο “TMBT” έχει χαμηλότερη Average Access Latency από τον “Huffman”. Αντίθετα ο “Huffman” έχει χαμηλότερο Average Tuning Time.

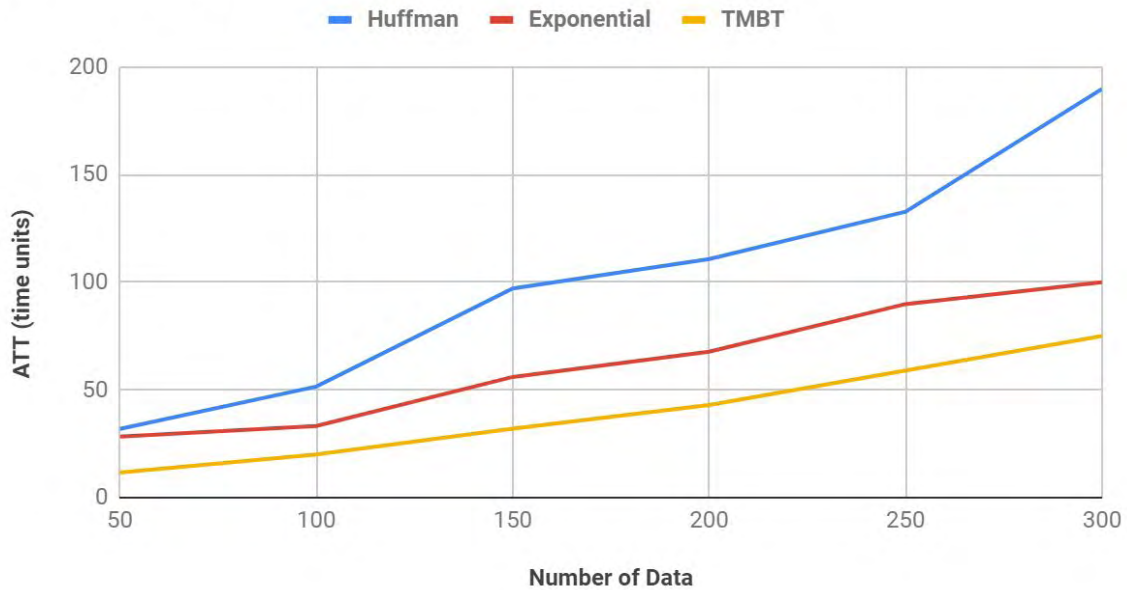
Στα Διαγράμματα 9 και 10 παρουσιάζεται η απόδοση των αλγορίθμων για “10 κανάλια εκπομπής”, αλλά αυτή τη φορά ο “TMBT” και ο “Huffman” δεν διαχωρίζουν τα κανάλια σε index και data channels αλλά έχουν 1 και 2 επιπλέον index channels αντίστοιχα.

10 Channels



Διάγραμμα 9: Average Access Latency: 10 data channels + index channels

10 Channels



Διάγραμμα 10: Average Tuning Time: 10 data channels + index channels

Παρατηρείται ότι καλύτερη επίδοση έχει ο “TMBT”, ενώ χειρότερη επίδοση ο “Huffman”.

6.3 Συμπεράσματα

Με βάση τις συγκρίσεις που έγιναν παραπάνω καταλήγουμε στα ακόλουθα συμπεράσματα για τον κάθε αλγόριθμο.

Αλγόριθμος 1: Ο “TMBT” βασίζεται στο δέντρο αναζήτησης B+. Είναι εύκολος στην κατασκευή του αλλά και στην αναζήτηση ενός δεδομένου. Παρατηρείται ότι για σταθερό αριθμό δεδομένων, όσο αυξάνεται ο αριθμός των data channels, μειώνεται η Average Access Latency ενώ ο Average Tuning Time διατηρείται σχετικά σταθερός. Ανάμεσα στους τρεις αλγορίθμους, όταν και οι τρεις έχουν ίσο αριθμό διαθέσιμων καναλιών μετάδοσης, παρουσιάζει την καλύτερη επίδοση όσον αφορά τον Average Tuning Time, αλλά και την καλύτερη επίδοση στην Average Access Latency όσο τα διαθέσιμα κανάλια μετάδοσης αυξάνονται. Αντίθετα όταν ο αριθμός των διαθέσιμων καναλιών μετάδοσης διατηρείται σταθερός και αυξάνεται το μέγεθος της βάσης δεδομένων παρουσιάζει τη χειρότερη επίδοση με βάση τον Average Tuning Time.

Αλγόριθμος 2: Ο “Huffman” βασίζεται στο αλφαβητικό δέντρο Huffman. Παρατηρείται ότι για σταθερό αριθμό δεδομένων, όσο αυξάνεται ο αριθμός των data channels, η Average Access Latency μειώνεται ελαφρά, ενώ ο Average Tuning Time ενδέχεται να έχει αυξομειώσεις ή να διατηρείται σχετικά σταθερός. Ανάμεσα στους τρεις αλγορίθμους, όταν και οι τρεις έχουν ίσο αυξανόμενο αριθμό διαθέσιμων καναλιών μετάδοσης, παρουσιάζει τη χειρότερη επίδοση. Όταν ο αριθμός των διαθέσιμων καναλιών μετάδοσης διατηρείται σταθερός και αυξάνεται το μέγεθος της βάσης δεδομένων παρατηρείται αύξηση στην Average Access Latency και στον Average Tuning Time. Στην περίπτωση αυτή παρουσιάζει τη χειρότερη επίδοση όσον αφορά την Average Access Latency, αλλά μια μέση επίδοση στον Average Tuning Time.

Αλγόριθμος 3: Ο “Exponential” βασίζει την αναζήτηση στο “Exponential Index” και είναι ο ευκολότερος στην κατασκευή. Παρατηρείται ότι για σταθερό αριθμό δεδομένων, όσο αυξάνεται ο αριθμός των data channels, η Average Access Latency αυξάνεται ενώ ο Average Tuning Time παρουσιάζει αυξομειώσεις και έχει μια μέση επίδοση σε σχέση με τους άλλους δυο αλγορίθμους. Όταν ο συνολικός αριθμός των channels είναι ίδιος και για τους τρεις αλγορίθμους και το πλήθος των δεδομένων προς μετάδοση αυξάνεται, αυξάνεται και η Average Access Latency και ο Average Tuning Time, ενώ παρουσιάζει

την καλύτερη επίδοση και στις δύο μετρήσεις. Αντίθετα , όταν μόνο ο αριθμός των data channels είναι ίσος και για τους τρεις αλγορίθμους, παρουσιάζει μέση επίδοση.

Στους πίνακες 4 και 5 παρουσιάζονται τα παραπάνω συμπεράσματα για την Average Access Latency και τον Average Tuning Time αντίστοιχα.

Πείραμα	Number of Data	Number of Channels	Καλύτερη επίδοση
1	50	4-10 Broadcast Channels	“TMBT”
2	50	2-10 Data Channels	“TMBT”
3	50-300	4 Broadcast Channels	“Exponential” + “TMBT”
4	50-300	10 Broadcast Channels	“Exponential”
5	50-300	10 Data Channels	“TMBT”

Πίνακας 4: Average Access Latency

Πείραμα	Number of Data	Number of Channels	Καλύτερη επίδοση
1	50	4-10 Broadcast Channels	“TMBT”
2	50	2-10 Data Channels	“TMBT”
3	50-300	4 Broadcast Channels	“Exponential”
4	50-300	10 Broadcast Channels	“Exponential”
5	50-300	10 Data Channels	“TMBT”

Πίνακας 5: Average Tuning Time

7. Σύνοψη

Σε αυτή την εργασία, κατασκευάζεται μια στρατηγική αξιολόγηση τριών διαφορετικών αλγορίθμων ευρετηρίασης δεδομένων σε πολλαπλά ασύρματα κανάλια εκπομπής σε ένα ενοποιημένο περιβάλλον. Μεταξύ πολλών αλγορίθμων έγινε επιλογή τριών με βάση το σχήμα ευρετηρίασης που χρησιμοποιούν, δηλαδή τον αλφαριθμητικό Huffman tree, το B+ tree και το Exponential Index. Οι αλγόριθμοι κατασκευάστηκαν σε γλώσσα προγραμματισμού C σε λειτουργικό Linux, και πραγματοποιήθηκε η μέτρηση της Average Access Latency και του Average Tuning Time στο ενοποιημένο περιβάλλον με τον ίδιο αριθμό δεδομένων μετάδοσης και τον ίδιο αριθμό καναλιών εκπομπής. Στη συνέχεια πραγματοποιήθηκε η αξιολόγηση των τριών αλγορίθμων με βάση ποιος από τους τρεις έχει την ελάχιστη Average Access Latency και τον ελάχιστο Average Tuning Time. Συμπερασματικά για σταθερό πλήθος δεδομένων και αυξανόμενα κανάλια εκπομπής καλύτερη επίδοση είχε ο “TMBT” που βασίζεται σε B+ tree, ενώ για σταθερό αριθμό καναλιών μετάδοσης και αυξανόμενο πλήθος δεδομένων καλύτερη επίδοση παρουσίασε ο “Exponential” που βασίζεται στο Exponential Index.

Σα μελλοντική έρευνα θα μπορούσαν να γίνουν συγκρίσεις σε μια πολύ μεγαλύτερη βάση δεδομένων και ανάμεσα σε περισσότερους πιο σύγχρονους αλγόριθμους. Επιπλέον, στην εργασία αυτή θεωρούμε ότι όλοι οι κόμβοι προς μετάδοσης (index και data) έχουν το ίδιο μέγεθος. Μελλοντικά μπορούν να πραγματοποιηθούν συγκρίσεις για άνισο μέγεθος κόμβων.

Βιβλιογραφία

- [1] B. Lee and S. Jung, “An efficient tree-structure index allocation method over multiple broadcast channels in mobile environments,” in Proc. 14th Int. Conf. Database Expert Syst. Appl., 2003, pp. 433–443.
- [2] S. Wang and H. Chen, “TMBT: An efficient index allocation method for multi-channel data broadcast,” in Proc. 21st Int. Conf. Adv. Inf. Netw. Appl. Workshops, 2007, vol. 2, pp. 236–242.
- [3] Damdinsuren Amarmend, Masayoshi Aritsugi, and Yoshinari Kanamori, “An Air Index for Data Access over Multiple Wireless Broadcast Channels,” Gunma University, Department of Computer Science
- [4] Jiaofei Zhong, Weili Wu, Xiaofeng Gao, Yan Shi and Xiaodong Yue, “Evaluation and Comparison of various indexing schemes in single-channel broadcast communication environment”, Springer Verlag Knowledge and Information Systems, Volume 40, Issue 2, Pages 375-409, 22 August 2014.
- [5] S. Wang and H.L. Chen, “Near-optimal data allocation over multiple broadcast channels”, Computer Communications, vol. 29, Issue 9, pp. 1341-1348, May 2006.
- [6] J. Xu, W. Lee, and X. Tang. “Exponential index: A parameterized distributed indexing scheme for data on air”. In Proc. 2nd Int’l Conf. on Mobile Systems, Applications, and Services (ACM MobiSYS’04), pages 153–164, 2004
- [7] Xiaofeng Gao, Yongtian Yang, Guihai Chen, Xin Lu, and Jiaofei Zhong. “Global Optimization for Multi-Channel Wireless Data Broadcast with AH-Tree Indexing Scheme”. July 2016
- [8] S. Jung, B. Lee, and S. Pramanik, “A tree-structured index allocation method with replication over multiple broadcast channels in wireless environments,” IEEE Trans. Knowl. Data Eng., vol. 17, no. 3, pp. 311–325, Mar. 2005
- [9] Huffman tree: https://en.wikipedia.org/wiki/Huffman_coding