International Journal of Latest Trends in Engineering and Technology
Vol.(9)Issue(4), pp.257-262
DOI: http://dx.doi.org/10.21172/1.94.40
e-ISSN:2278-621X

COMPARATIVE STUDY OF VARIOUS AIR INDEXING TECHNIQUES USED ON DIFFERENT METHODS OF INFORMATION DISSEMINATION ACCESS

Thirupurasundari D R[1], Amit Kumar Gupta[2], Vikas Goel[3]

Abstract- In wireless data broadcasting, the data is promulgated widely to a tremendous number of users (clients) or mobile subscribers in which two major parameters are considered i.e., Access Time and Tuning Time to evaluate the performance of an indexing scheme of data broadcasting system. In this paper, different existing and popular indexing schemes has been summarized w.r.t energy efficiency depending majorly upon data structure, data type, broadcast channels and channel hopping. All indexing schemes are somewhat different from each other and therefore it is hard to compare and classify these indexing schemes w.r.t efficiency. This paper unveils the comparison between these indexing schemes in order to achieve the tradeoff between tuning time and access time while broadcasting the data.
Keywords – data broadcasting, character based data, alphabetic Huffman tree, full text search, XML data broadcast, on demand broadcast, inverted list, air indexing, and channel hopping

1. INTRODUCTION

Boosted by the maturating of high-speed wireless network and personal mobile device technology, mobile computing has become a research area of rapidly emerging interest. Recently, 4G networks together with smart phones make mobile computing in a wide range possible. Consequently, more mobile users will constantly need to access public information (for instance, stock price, local points of interest, real-time traffic and weather information) through the wireless connection. However, due to the technical limitations of mobile computing (such as insufficient bandwidth and power supply), how to disseminate data efficiently in the mobile computing environment has become a challenging problem. Data broadcast is an attractive solution for this problem because of its scalability and flexibility [1][2].

A tradeoff between Tuning time and access time is essential to deal with this problem. Access time is the time elapsed from the moment an initial probe is made into the broadcast channel to the moment the desired data are acquired. This is the total time the clients must spend and is often used to evaluate the performance of the broadcast programs. Tuning time is the time spent by the clients listening to the broadcast channel. Indexing can significantly reduce tuning time by switching clients to turn into two modes i.e., doze mode and active mode so that the clients can operate well. When the clients are listening to the data items in the broadcast channel, the CPU must operate in the active mode, which is costly for power consumption. However, the clients can operate in the doze mode to save power consumption, when the requested data items have not arrived. By switching between active and doze modes, the power consumption of a mobile device is reduced significantly [1]. Indexing technologies and data/index allocation methods are the most effective methods to reduce the tuning time and access latency. In the former method, indices help to reduce the active time of a mobile device significantly and clients can follow the direction of indices, turn off during the waiting period and turn on again right before the desired datum appears. The later deals with the allocation of data and indices, the proper allocation method assigns relevant data items and indices as closely as possible and thus reduces the clients waiting time [2]. Indexing technologies are developing very fast during recent years and it is fair and square to choose the state of the art design for comparison. To overcome the aforesaid shortcomings, we are aiming at comparing the performance of various indexing techniques under all possible situations. To summarize, the communication environment varies from four aspects: broadcast environment, data structure, data type and channel hopping. Many works discussed efficient indexing schemes based on data structure which can be classified into three categories: hashing based indexing [3, 4] tree based indexing [5, 6] and table based indexing [7, 8]. Tree based indexing is a method that undergoes in different indexing techniques such as (1,m) Indexing, distributed indexing which are based on B+ tree and Alphabetic Huffman tree.

2. RELATED WORKS

A series of indexing schemes have been developed for wireless data broadcasting systems in order to reduce average searching time and tuning time. Wireless data broadcasting focuses on the design and analysis of various wireless networking

[1] Research Scholar, Shri Venketeshwara University, Uttar Pradesh
[2] Associate Professor, Dept. of MCA, KIET Group of Institutions, Ghaziabad, Uttar Pradesh
[3] Associate Professor, Dept. of CS&E, AKGEC, Ghaziabad, Uttar Pradesh

environments that seek distributed, autonomous, robust and scalable algorithms for real world applications including optimization strategies, system architectures, indexing technologies in order to reduce access latency and tuning time.
Imielinski et al. proposed (1, m) index [1] which broadcasts the index part m times in front of each fraction of the data file. They also customized distributed index [1] which divides the index tree such as B+ tree into replicated part and non – replicated part. B+ tree distributed index (BTD) was extended by many other researchers to satisfy different system requirements. Gao et al. [14] redesigned BTD and built a complete multi- channel broadcasting system with non- uniform data access probabilities an unequal data sizes. Huffman tree is a skewed index tree which takes into account the data access probabilities, where more popular data have a shorter path from the root of the tree, thus the average tuning time is minimized. [15]. A construction of Huffman tree [16] is similar to Huffman code construction, but it has a problem that the clients may fail to find desired data by traversing that Huffman tree. The other algorithm for constructing skewed Huffman tree has the same problem. There is another type of Huffman tree, Alphabetic Huffman tree [17] which serves as a binary search tree that is further extended to k-ary search tree [16] so that a tree node will fit in any size wireless packet by adjusting the fan out of the tree. Later , Zhong et al. [18] proposed an Alphabetic Huffman tree distributed indexing scheme which minimizes both average access latency and average tuning time and outperforms the B+ tree distributed indexing scheme. Yang et al. [6] uses Alphabetic Huffman tree data structure for indexing the data items that are associated with skewed access probabilities. Gao et al. [2] presented a global optimization for multi channel wireless data broadcast system with AH – Tree indexing scheme by providing three designs to improve the performance of the system that involves constructing an arbitrary k –ary AH Tree in O(t2) by dynamic programming and improving the control table design which eliminates 50% redundant entries while searching. New index and data allocation algorithms w.r.t hopping cost has been introduced to further reduce the average access time and tuning time. Shi et al. [19] have thoroughly discussed the optimization problems in wireless data broadcasting. XML character based data is accessed and disseminated in On demand environment to improve the usability of wireless channel. In an On demand data broadcast environment, the server responds to requests (XPath queries) made by clients[22][23][24]. In (1,α(1,β)) indexing scheme , the inverted list of documents is used for indexing as an index tree which is replicated 1/β times of the size of inverted list because inverted list is replicated 1/α times of the database size, the index tree is replicated αβ times constituting an index tree, a fraction of the inverted list and a fraction of the data [26][27[28].

3. INDEXING BASED ON DATA STRUCTURE

3.1 B+ Tree Indexing

The first popular indexing scheme extended from disk based indexing is the B+ Tree. In case of single channel, the indices are organized in B+ Tree structure which is called as an Index Tree [29],[30]. Each index node has a number of pointers point to its child nodes. The pointers of the bottom level indices point at the actual data nodes. To find a specific broadcast data item, the search starts from top to bottom. The top level index node is searched first to determine which child node contains the data item. Then the same process will be performed on that node. This procedure continues and finally reaches the data item at the bottom. The sequence of the index nodes traversed is called the index path of the data item [1], [31].

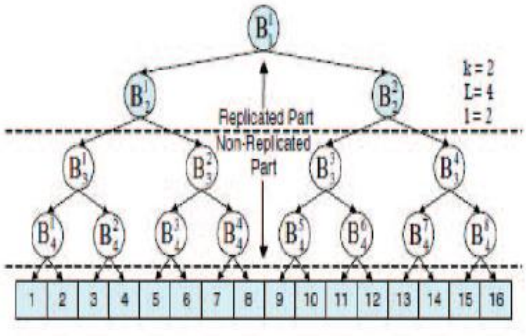

Figure 1. An example of a B+ Tree Indexing scheme

In figure 1, B+ Tree is made for indexing the data. The tree is divided into two parts: the upper part is called replicated part and the lower part is called non replicated part. The internal nodes are the index nodes and the outer leaves are the data nodes. For each, two types of buckets are used that is data bucket and index bucket. A node in the index tree represents an index bucket in the broadcast channel. A broadcast data item is represented by a data bucket. It consists of partially replicated index wherein if a leaf node has keys belonging to different index channel then this leaf node is also replicated in channels that own the index keys. Root nodes are replicated in all the index channels, some of the non leaf nodes are replicated while others are not. It is advantageous in a way that the overhead of sending the index directory is reduced as well as the client can tune into any index channel and enable to find the right pointer even if the pointer is present in different index channel. Index replications are cut down as it is sufficient to have only one index which is relevant to the data and immediately follows it in

the broadcast. But this result in space consumption due to index information which could have been used for the transmission of the data items [1],[29],[32].

3.2 Alphabetic Huffman Tree

Alphabetic Huffman Tree (AH – Tree) is an appropriate data structure to index data set with skewed access frequencies, which fits the feature of web based wireless data broadcast service to a massive number of mobile clients. AH – Tree has been considered a good choice among other tree based indexing (e.g., B+ Tree, Huffman Tree) because it overcome the shortcoming of bringing the longer tuning time on average. In typical AH – Tree, the higher the frequency of a data item, the shorter the path from root to the corresponding leaf index. Many literatures have studied AH – Tree index during the past two decades and a sample AH tree is depicted in figure 2.

In [17], Hu and Tucker first proposed a binary AH –Tree algorithm with time complexity O(n log n), where n is the size of data items which was based on a complex queue technology as well as cannot be directly extended to construct k –ary AH – Tree with arbitrary k. Later, Shivkumar et al. [33] extended to Hu – Tucker algorithm into k- ary AH – Tree and was the first to implement it as indices to broadcast environment. Nevertheless, they didn't describe the algorithm clearly, only mentioned a skeleton of how to build a k – ary AH – Tree with lack of specification of internal node construction with k branches. This design gave time complexity up to O(n3).

Similarly [31],[34],[35] discussed AH – Tree index for data broadcast problem respectively , none of them provided complete tree – construction process with time complexity analysis instead they gave simple explanation according to the description in [4]. Moreover, Researchers tended to modify the index tree into a distributed index sequence to improve the performance [12],[14],[35] which depends on the use of control table. Although, control table results in space overhead due to containing redundant information X.Gao et al. [6] proposed an efficient AH –tree construction with the help of dynamic programming. They gave an algorithm that can build arbitrary k – ary AH –Tree index with bounded tree height in O (kn2) time which is modified into a distributed index sequence with control table design to further reduce the searching steps. A general and effective scheme was proposed by them to eliminate redundant entries in control tables to reduce the overall index packet length, which saved almost 50% of the storage.

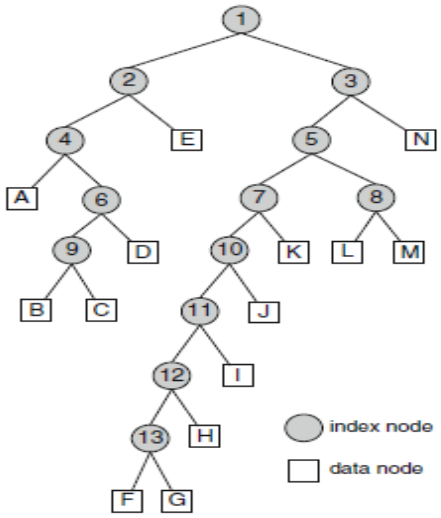

Figure 2. An example of Alphabetic Huffman Tree

3.3 Hashing

Hashing techniques store hashing parameters in data buckets without requiring separate index buckets or segments. Each data bucket consists of two parts: data part and control part. The former contains the actual data records and the letter contains control information used to guide clients to the right data bucket. The control part also consists of a hashing function and a shift value. The hashing function is used to map the key value of the broadcast data into a hashing value. Each bucket has a hashing value H assigned to it. Hashing function eliminates the need to broadcast index structure and hence shortens the broadcast cycle to much an extent which in turn results in reducing access latency. Although, it leads to the mechanism of collision [11][26].

In the event of collision, the colliding data is inserted just after the bucket which has the same hashing value because of which the rest of the buckets will have to be shift. By shifting the data bucket is being out of place. The shift value in each bucket is used to find the right position of the corresponding data. It points to the first bucket containing the data with the right hashing value [11][26][29][36].

Broadcast sequence of hashing technique is depicted in figure 3.

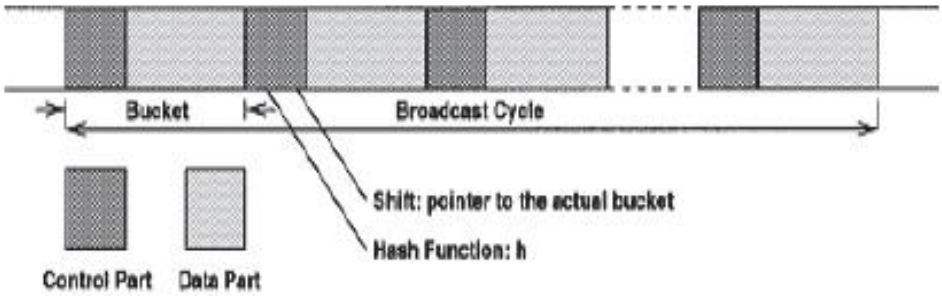

Figure 3. Broadcast sequence in hashing scheme

Over the period of time a couple of variations in the hashing techniques are evolved, each of them having an improvement over the basic hashing scheme; simple hashing scheme uses the primarily developed simple access method and a single hash function [37]. Imielinski et al. [11] presented two hashing protocols i.e., Hashing A and Hashing B. The former protocol calculates h(K) and then follows the shift value to find data, the latter one applied a minor modification of the hashing function to improve performance. It has been extended to Mhash by Yao et al.[9], which considers a two argument hash function H(k,l) to map each data to a number of slots, thus facilitates skewed access probabilities and reduces access latency i.e., it aims at reducing the tuning time by mapping the data items to the slots in the broadcasting by suing a hole free hash function.

3.4 Signature Indexing

In signature indexing an index frame generated by a signature and data frames are interleaved on a wireless broadcast channel. The formation of signature is done by hashing each field of a record into a random bit string and then superimposing together all the bit strings into a record signature. The signature can be generated based on all attributes of data records in database. The number of collisions in a signature depends on how perfect the hashing function is and how many attributes a data record has. In signature indexing, collisions occur when two or more data records have the same signature value. Usually the more attributes each data record has, the more likely collisions will occur. Such collisions will translate into false drops. In false drop , clients download wrong data records but with matching signatures [38]. There are three signature algorithms based on their costs for access time and tuning time.

4. CLASSIFICATION OF VARIOUS INDEXING SCHEMES

Different indexing schemes have been compared in literature to achieve a better tradeoff between tuning time and access latency with respect to several factors and parameters. Every indexing scheme has its own advantages and disadvantages that lead to enhance performance but it is hard to say which indexing scheme is better. Also, it is not easy to achieve a truly fair comparison among different types of indexing schemes under various allocation scenarios and different broadcasting environment. Hybrid technique was proposed by M. Narvekar et al. [32] involving sparse index tree, hash table and signature indexing wherein sparse index tree has a hashing table at every node consisting of the addresses of the next nodes which facilitates quick retrieval of the address of the next node till it reaches the leaf node by reducing the tuning time.

In this paper, we have classified different indexing schemes based on four aspects and providing the comparisons of indexing based on data structures in all the possible situations by presenting an intuitive comparisons in the form of table, for the measurement criteria as follows (Table 1): access latency, tuning time, power conservation, indexing efficiency, space, flexibility, error resilience, correct response, ease of construction and searching, broadcast cycle and broadcast environment.

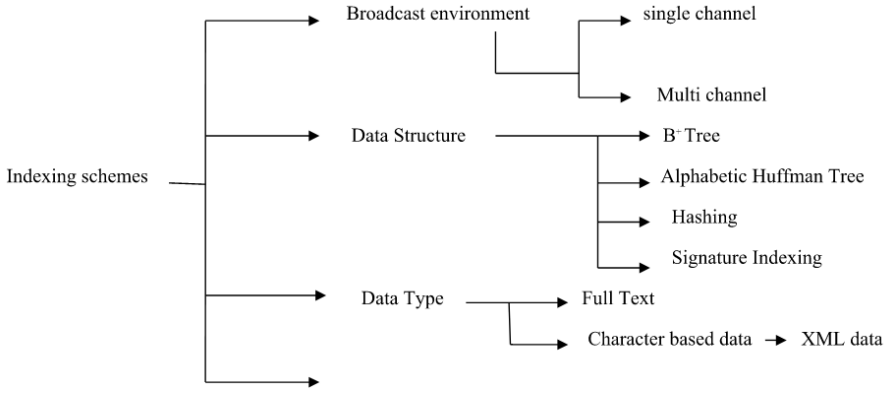

Table 1. The comparison of different indexing schemes based on Data Structure

	Data Structures			
Features	B+ Tree	Alphabetic Huffman Tree	Hashing	Signature Indexing
Access latency	Minimal access time as compared to signature indexing	Excellent access time as compared to three of them	Use of non flat data broadcast reduces access time	Has high access time as it checks consecutive signatures
Tuning time	Clients can do a quick search in the index tree to find out the arrival time of the desired data hence the tuning time is very minimal	Better in retrieving the desired data more efficiently than B+ Tree	Indexing broadcast Information using hashing functions reduces tuning time	Filtering efficiency heavily depends on the false drop probability leading to high tuning time
Power consumption	Less power usage as compared to signature indexing	Less power consumption as compared to B+ Tree	Caching mechanism reduces power consumption	High access time is directly proportional to the high power consumption
Indexing efficiency	Not as powerful as hashing in order to provide a balance between access latency and tuning time	Highly efficient with skewed data	Efficient to balance between access latency and tuning time	Usually proves to be efficient in hybrid architecture along with hashing or indexing techniques
Space	Needs to hold a lot of index information in turn consuming large amount of space	Can be reduced by using control tables that eliminates redundant entries	Caching proves to be efficient in space handling	Efficient in space consumption as compared to B+ Tree
Flexibility	Flexible even where data sizes are different	Flexible enough to handle various data set features without any impractical assumptions	Not flexible as B+ Tree as it needs further modifications for different data sets	Not flexible as compared to three of them
Error resilience	Sometimes resilient to link errors	Better resilient than B+ Tree to link Errors	Good error Resilient	Poor error resilient
Correct response	Yes	Dynamic programming is needed for data selection in order to achieve correct response	Yes	Sometimes
Ease of construction and searching	Easy to construct and performs well in Searching	Needs two stage construction process	May take some time for constructing the broadcast sequences	-
Broadcast cycle	Has short bcast with smaller index buckets	Has shorter bcast cycle while allocating index due to channel hopping	Has short bcast	-
Broadcast environment	Clustered	Non – clustered	Non – clustered	Clustered

5. CONCLUSION

In this paper, we summarized indexing schemes based on different categories such as Broadcast environment; Single Channel and Multi channel, Data Structures that includes four popular indices i.e., B+ Tree, Alphabetic Huffman Tree, Hashing and Signature indexing, Data types mainly Character based data (XML Data) and Full Text Search and Channel hopping. We have compared channel hopping techniques also that helps in searching of the indices over multiple broadcast channels with index allocation techniques that consider hopping cost by using a technique Stripe that sequentially stripe index levels across

all the broadcast channels. We strive to study the performance of all channel hopping techniques in all possible situations but here we aimed at tuning time, hop count and aed on which hopping techniques depends.

6. REFERENCES

[1] T.Imielinski S.Vishwanathan and B.R. Badrinath, "Energy efficient indexing on air". Vol 546, june 1994.
[2] Gao, X., Yang, Y., Chen, G., et al., Global optimization for multi-channel wireless data broadcast with AH-tree indexing scheme. IEEE Transactions on Computers, 65(7), 2104-2117,2016.
[3] Zhong, J., Gao, Z., Wu, W., Chen, W., Gao, X., Yue, X.: High Performance Energy Efficient Multi-Channel Wireless Data Broadcasting System, 1–6, 2013.
[4] Zhong, J., Wu, W., Gao, X., Shi, Y., Yue, X.: Evaluation and Comparison of Various Indexing Schemes in Single-Channel Broadcast Communication Environment. Knowl. Info. System 40(2), 375–409, 2014.
[5] Lu X, Gao X, Yang Y, Zhong J.: SETMES: a scalable and efficient tree-based mechanical scheme for multi-channel wireless data broadcast. In: Proceedings of the ACM international conference on ubiquitous information management and communication, Kota Kinabalu, Malaysia, 2013.
[6] Y.yang, x.gao, j.zhong, " distributed ah – tree based index technology for multichannel wireless data broadcast", 2013.
[7] Vishnoi, S., Goel, V.: Novel table based air indexing technique for full text search. In: IEEE International Conference CICT 2015, pp. 410–415, 2015.
[8] Xu, J., Lee, W.-C., Tang, X., Gao, Q., Li, S.: An error-resilient and tunable distributed indexing scheme for wireless data broadcast. TKDE 18(3), 392–404, 2006.
[9] Yao, Y., Tang, X., Lim, E.P., Sun, A.: Energy-efficient and access latency optimized indexing scheme for wireless data broadcast. IEEE TKDE 18(8), 1111–1124, 2006.
[10] Yang, K., Shi, Y., Wu, W., Gao, X., Zhong, J.: A novel hash-based streaming scheme for energy efficient full-text search in swireless data broadcast. In: Yu, J.X., Kim, M.H., Unland, R. (eds.) DASFAA 2011, Part I. LNCS, vol. 6587, pp. 372–388. Springer, Heidelberg, 2011.
[11] T. Imielinski, S. Viswanathan, and B.R. Badrinath. Power efficient filtering of data on air. Proceedings of EDBT Conference, pp. 245–258, March 1994.
[12] Imielinski, T., Viswanathan, S., Badrinath, B.R.: Data on Air: Organization and Access. IEEE Trans. Knowledge and Data Eng. 9(3), 353–372, 1997.
[13] V Goel, A k Ahlawat and M. N. Gupta: Partial index replicated and distributed scheme for full-text search on wireless broadcast", Springer Sadhana-Academy Proceedings in Engineering Sciences, vol-40, no.-7, pp. 2129-2142, Oct 2015.
[14] Gao X, Shi Y, Zhong J.: SAMBox: a smart asynchronous Multi-channel black box for wireless data broadcast. In: Proceedings of the 21st international conference on software engineering and data engineering, Los Angeles, CA, 2012.
[15] Chen, M., Wu, K., Yu, P.: Optimizing index allocation for sequential data broadcast in wireless mobile computing. TKDE 15(1), 161–173, 2003.
[16] Shivakumar, N., Venkatasubramanian, S.: Efficient Indexing for Broadcast Based Wireless Systems. Mobile Networks and Applications, 433–446, 1996.
[17] Hu, T., Tucker, A.: Optimal computer search trees and variable-length alphabetical codes. SIAM Journal on Applied Mathematics 21(4), 514–532,1971.
[18] Zhong J, Wu W, Shi Y.: Energy-efficient tree-based indexing scheme for efficient retrieval under mobile wireless data broadcasting environment. In: Yu J, Kim M, Unland R (eds) Proceedings of the 16th international conference on database systems for advanced applications, Hong Kong, China, April 2011, LNCS 6588:335–351, 2011.
[19] Y. shi, w. wu, j.zhong, x.gao, "Optimization problems in data broadcasting, handbook of combinatorial optimization", 2013.
[20] J.p park, c.s.parkn y.d chung, "lineage encoding: an efficient wireless xml streaming supporting twig pattern queries", 2013.
[21] W. hu, c.fan, j. luo and bo du, "on demand data broadcasting scheduling algorithm based on dynamic index strategy", 2013.
[22] W.sun, y.qin, j.wu, and j.zhang, " air indexing for on demand xml data broadcast", 2013.
[23] A. K. Mahapatra and S. Biswas, "Inverted Indexes: Types and Techniques", International Journal of Computer Science Issues (IJCSI), 8 (4), pp. 384-292, 2011.
[24] Chung, Y.D., Yoo, S., Kim, M.H.: Energy- and latency-efficient processing of fulltext searches on a wireless broadcast stream. IEEE TKDE 22(2), 207–218, 2010.
[25] Vishnoi S. and Goel V. :Performance evaluation and comparison of air indexing techniques for Full Text Searches under a unified communication environment. Proceedings of the IEEE 3rd International Conference on Reliability, Infocom technologies and Optimization (ICRITO), pp. 1–6, October 2014.
[26] M. wisely, s.s. sarvestani and a. r hurson, " public data dessimination via broadcasting" , 2015.
[27] On optimal scheduling for time constrained services in multichannel
[28] V. goel, G.panwar, A.K Ahlawat, "energy efficient aire indexing schemesfor single and multilevel wireless channels", proceeding of IEEE International Advanced Computing Conference (IACC-2013) on IEEE Xplore, pp. 525-530, Feb, 2013.
[29] X. Yang and A. Bouguettaya, "Broadcast-Based Data Access in Wireless Environments", Springer-Verlag Berlin Heidelberg, pp. 553-571, 2002.
[30] S. jung, b.lee, s.pramanik, "a tree structured index allocation method with replication over multiple broadcast channels in wireless environments", 2005.
[31] S.s.mantha,m narvekar, s.lopes, " comparative study of indexing techniques for data dessimination in wireless environment", 2015.
[32] N. Shivakumar and Venkatasubramanian, "Efficient Indexing for Broadcast Based Wireless Systems", Mobile Networks and Applications, pp. 433-446, 1996.
[33] J. Zhong, W. Wu, Y. Shi and X. Gao, "Energy-Efficient Tree-Based Indexing Schemes for Information Retrieval in Wireless Data Broadcast", DASFAA, Part-II, LNCS-6588, Springer-Verlag, pp. 335-351, 2011.
[34] Jiaofei Zhong, Zheng Gao, Weili Wu, Weidong Chen and Li Wang, "Multi-channel Energy-efficient Hash Scheme Broadcasting", proceedings of The 21st International Conference On Software Engineering And Data Engineering SEDE 2012, June 27-29, 2012.
[35] M. Vijayalakshmi and A. Kannan, "A Hashing Scheme for Multi-channel Wireless Broadcast", Journal of Computing and Information Technology - CIT 16, 3, pp. 197–207, 2008.
[36] D. Aksoy and M. Leung. Pull vs Push: a Quantitative Comparison for Data Broadcast. Global Telecommunications Conference, GLOBECOM'04 IEEE, 2004.
[37] W.-C. Lee and D.L. Lee, "Using signature techniques for information filtering in wireless and mobile environments", Distributed and Parallel Databases, 4, pp. 205–227, 1996.