Survey on wireless XML broadcasting system

Wireless XML broadcasting

P.Prabhavathy
Research Scholar, DCSE
CEG campus, Anna University
Chennai, India
pprabhavathy@gmail.com

S.Bose
Associate Professor, DCSE
CEG campus, Anna University
Chennai, India

***Abstract*— Wireless communication is blooming along with Internet access from mobile devices which is an essential criterion for information access from anywhere, any time and any device. In this survey, review on various technologies like pervasive computing, mobile communication systems, selective dissemination of information, with location and device independent information interchange specification like XML (Extensible Markup Language. Wireless XML based selective dissemination of information is done in two ways. They are push based and pull based systems. Push based system broadcast the XML information over the wireless channel in a periodic or aperiodic manner. Pull based system gathers request from mobile clients through uplink channel and broadcasts the corresponding XML data for request in the downlink channel. Major performance metrics for wireless XML broadcasting system are tune-in time and access latency. Since wireless system has restricted power and bandwidth, efficiently utilizing these resources is essential.**

Keywords- wireless broadcasting; XML dissemination; mobile computing

I. INTRODUCTION

There are several characteristics that are typical of wireless computing environments. They are: Limited Wireless Channel Bandwidth, Susceptibility of Wireless Communications to Interference, Asymmetric Communication Environments, Limited Effective Battery Lifespan, Threat to Security and High Cost.

Some successful examples that exploit mobile and wireless computing technologies includes the following:

1. **Financial Industry**: Stockbrokers at the New York Stock Exchange no longer need to use paper and rush to phones to collect order. They can carry handheld PDAs and access information from HP servers through a wireless network.
2. **Manufacturing Industry**: Manufacturing time cycles are cut using handheld computers and RF-based scanning devices on assembly lines.
3. **Health Care Services**: Devices that monitor patients conditions can be made to transmit critical information directly to nurses and doctors equipped with portable computers or advanced pagers
4. **Transportation Industry**: Courier companies such as Federal Express, UPS and DHL as well as trucking industries are also among the early adopters of wireless technology. Parcel tracking as well as vehicle tracking applications based on wireless networks are some of the successful applications that provide customers with superior service.
5. **Education**: Wireless technology has also been deployed in educational institutes to allow staff and students to learn, interact and access real world resources at any time and any place. The Global Campus project at the National University of Singapore is one such initiative Law Enforcement and Public Safety.

A. *Design issues in wireless data Dissemination*

- Efficient wireless bandwidth utilization
- Efficient and effective scheduling strategies at the server
- Energy-efficient data access for battery-powered portable devices
- Support for disconnection
- Support for secured and reliable transmission.

B. *Models for information dissemination*

There are two fundamental information delivery methods for wireless data dissemination applications: Point-to-Point access and Broadcast. Compared with Point-to-Point access, broadcast is a more attractive method. A single broadcast of a data item can satisfy all outstanding requests for that item simultaneously. Broadcast can scale up to an arbitrary number of users. There are three kinds of broadcast models, namely push-based broadcast, On-demand (or pull-based) broadcast, and hybrid broadcast.

In push based broadcast system [2,6], the server broadcasts information using a periodic/aperiodic broadcast program to generate the broadcast cycle. Whereas in on demand broadcast [3, 4], the server broadcasts information based on the outstanding requests submitted by clients. Hybrid broadcast [1, 5, 7] combines both the push based broadcast and on demand to complement each other. Time critical broadcast concentrates on reduced access latency metric at the optimal cost of tuning time. Energy conserving broadcast aims at reduced tune-in time by keeping access latency as optimal as possible.

II. SCHEDULING FOR INFORMATION DISSEMINATION

Time critical broadcast deals with scheduling and data organization over wireless broadcast channel for access efficiency. Therefore, three kinds of data scheduling methods such as push based scheduling, on demand scheduling and hybrid scheduling are available to reduce access latency in above mentioned three data broadcast models respectively.

A. Push based scheduling

Broadcast program computes the order and frequencies that data items are broadcast in the wireless channel. Scheduling algorithm makes use of precompiled access profiles in finding the broadcast program. Two typical methods for push based data scheduling are namely flat broadcast and broadcast disks. Simplest scheme for data scheduling is flat broadcast, which is flat broadcast program where all data items are broadcast in a round robin manner.

Access latency for each data item is the same equal to half of the broadcast cycle. Its performance is poor in terms of average access time when data access probabilities are skewed in nature. Broadcast Disk (Bdisk) is hierarchical dissemination architecture introduced in [6], where data items are designated to different logical disks so that data items in the same range of access probabilities are grouped on the same disk. Data items are then selected from the disks for broadcast according to their relative broadcast frequencies. Each disk is divided into equal sized small units called chunks. Disks can be logically conceived as real physical disks spinning at different speeds, with the faster disks placing more instances of their data items on the broadcast channel. Figure 1 illustrates the analysis on various broadcast programs.

Access Probability			Expected Delay		
A	B	C	Flat	Skewed	Multi-Disk
0.333	0.333	0.333	1.50	1.75	1.67
0.5	0.25	0.25	1.50	1.63	1.50
0.7	0.125	0.125	1.50	1.44	1.25
0.9	0.05	0.05	1.50	1.33	1.10
1.0	0.0	0.0	1.50	1.25	1.00

Figure 1. Broadcast program analysis for bandwidth allocation

Access probabilities of data item A, B and C is given along with their access latency in various broadcast programs. Flat broadcast is best for Uniform access probabilities of all data items. Skewed broadcast is better for non-flat broadcast. Higher access probabilities of data item leads to more bandwidth requirement in allocation.

Data item A is accessed with probability 0.5, so its bandwidth allocation is $\sqrt{0.5} / (\sqrt{0.5} + \sqrt{0.25} + \sqrt{0.25}) = 41\%$

For flat broadcast program, 33% (8% less than skewed)

For broadcast disk, 50% (9% more than skewed)

With dynamic broadcast there are three different changes that have to be handled. They are: 1) Changes to the value of the objects being broadcast.2) Reorganization of the broadcast. 3) Changes to the contents of the broadcast.

B. On-demand data scheduling

Many recent research studies on wireless data dissemination have proposed using on-demand data broadcast [3, 4, 8, 9]. A wireless on demand broadcast system supports both broadcast and on demand services through a broadcast channel and a low bandwidth uplink channel. The uplink channel can be a wired or a wireless link. When a client needs a data item, it sends to the server an on demand request for the item through the uplink. Client requests are queued up at server upon arrival. The server repeatedly chooses an item from among the outstanding requests, broadcasts it over the broadcast channel, and removes the associated request from the queue. The clients monitor the broadcast channel and retrieve the item they require. The data-scheduling algorithm in on demand broadcast determines which request to service from its queue of waiting requests at every broadcast instance.

C. Balancing Push and Pull

Hybrid architecture was first investigated in [10, 11]. In that model, items are classified as either frequently requested or infrequently requested. It is assumed that clients know which items are frequent and which are infrequent. Frequent data or hot data is broadcasted using broadcast cycle and infrequent data using on-demand broadcasting. In the downlink scheduling, the server makes consecutive transmissions of frequented items according to a broadcast program, followed by the transmission of the first item in the infrequent queue. Analytical results for the average access time were derived in [11]. In [5], the push based Bdisk model was extended to integrate with a pull based approach. The proposed hybrid solution, called Interleaved Push and Pull (IPP), consists of an uplink for clients to send to the server pull requests for the items that are not on the push-based broadcast.

The server interleaves the Bdisk broadcast with the responses to pull requests on the broadcast channel. The disadvantage of this approach is that if there is not enough bandwidth for pulls, the performance might degrade severely since the pull latencies for non-broadcast items will be extremely high. In [12], an attempt was made to compare various broadcast scheduling algorithms. For this, a simulation model, Sketch-it, is developed and compared various algorithms. This is very useful for conducting various experiments by changing the critical parameters. In [13], the multicast server offering the data items at a variety of transmission speeds to the clients' varied requests is discussed. The paper proposes to slice a server's available outgoing network capacity in to data channels, assign server's data to those channels, and assign clients to the channels given client' varied requests and download speeds.

III. INDEXING FOR ENERGY EFFICIENT DATA BROADCAST

Portable device operates in wireless environments are typically powered by batteries. As a result, power conservation is a key issue [19, 72, 21]. Unfortunately, the lifetime of a battery is very short and is not expected to increase much [14, 96]. Moreover, when a client is listening to the channel, the CPU must be in active mode to examine the data packets. This operation is very expensive from an energy point of view, because often only few data packets are of interest to a particular client. It is therefore important for a mobile client to be able to operate under two different modes: doze mode where it is still connected to the network but it is not active; and active mode where it performs its usual activities. Clearly, an important topic is to provide mechanisms that allow clients to switch from doze mode to active mode alternatively when listening to the broadcast data. By selectively tuning to the desired portion of the broadcast, energy consumption can be minimized and the effective battery life can be prolonged.

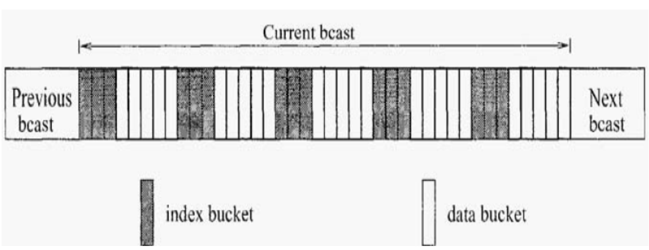

Figure 2. Wireless broadcasting stream structure

To allow clients to selectively tune to the broadcast, the clients need to be equipped with a directory that indicates the point of time in the broadcast channel when particular objects will be broadcast. There are two solutions to this:

1. The mobile clients cache a copy of the directory. This solution, however, has limited applicability. Since mobile clients may move from cell to cell, it means that the clients will have to cache all the directories of all the cells that they may possibly move to. The storage requirement on the portable clients may become too high. In addition, the clients must frequently refresh their copies of the directories to ensure consistency. This may also generate excessive traffic between clients and the server.

2. The directory is broadcast in the form of an index on the broadcast channel. In this case, the client must first access the directory once it moves into a cell. This approach incurs the overhead of additional channel bandwidth requirement for broadcasting the directory, but it is scalable and avoids the problems that are associated with the caching approach. Thus, this is the most widely used and preferred solution.

A. General protocol for retrieving broadcast

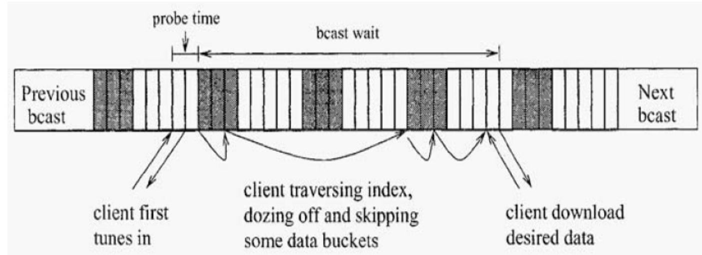

Figure 3. Protocol for data retrieving from broadcast channel

1. The mobile client tunes to the channel and looks for the offset pointing to the next index bucket. During this operation, it must be in active mode. Most works assume that each bucket contains the offset to the next index bucket. So this step requires only one bucket access. Let n be the determined offset.

2. The client switches to the doze mode until time (n –1) x T. At that time, it tunes into the channel (thus, it is again in active mode) and, following a chain of pointers, determines the offset m, corresponding to the first bucket containing the data of interest. We note that during this process, the client may switch between the doze mode and active mode multiple times.

3. The client switches to doze mode until time (m – 1) x T. At that time, it tunes into the channel and retrieves the data of interest.

Access time: It is the average duration from the instant in which a client wants to access objects with a specific key value to the instant when all required objects are downloaded by the client. The access time is based on following two parameters, which is clearly depicted in Figure 3.

Probe time: Time duration between accessing objects with a specific key value and nearest index information related to the relevant data is obtained by the client.

Bcast wait: The duration from the point the index information related to the relevant data is encountered to the point when all required objects are downloaded.

Tuning time: It is the time spent by a client listening to the channel. Thus, it measures the time during which the client is in active mode and therefore determines the power consumed by the client to retrieve the relevant data.

B. Flat broadcast programs with indexes

Many indexing techniques are available for flat broadcast programs. They are as follows:

1. *(1,m) Indexing technique*

The (1, m) indexing scheme [20] is an index allocation method where a complete index is broadcast m times during a bcast. All buckets have an offset to the beginning of the next index segment. The first bucket of each index segment has a tuple containing two fields. The first field contains the key value of the object that was broadcast last and the second field is an offset pointing to the beginning of the next bcast. This tuple guides clients who missed the required object in the current bcast so that they can tune to the next bcast.

The main problem with the (1, m) index organization is that the entirety of the index has to be replicated m times. This increases the length of the broadcast cycle and hence the average access time. The optimal m value that gives minimal average access time is shown to be The scheme has, however, good tuning time.

2. *Tree-based Index*

In the tree-based indexing scheme, which is also referred to as the distributed indexing scheme in [20], a data file is associated with a B+-tree index structure [17]. This is shown in figure 4. Since the broadcast medium is a sequential medium, the data file and index must be flattened so that the data and index are broadcast following a preorder traversal3 of the tree. Furthermore, the index comprises two portions: the first k levels of the index will be partially replicated in the broadcast, and the remaining levels will not to be replicated. The index nodes at the (k + 1)th level are called the non-replicated roots.

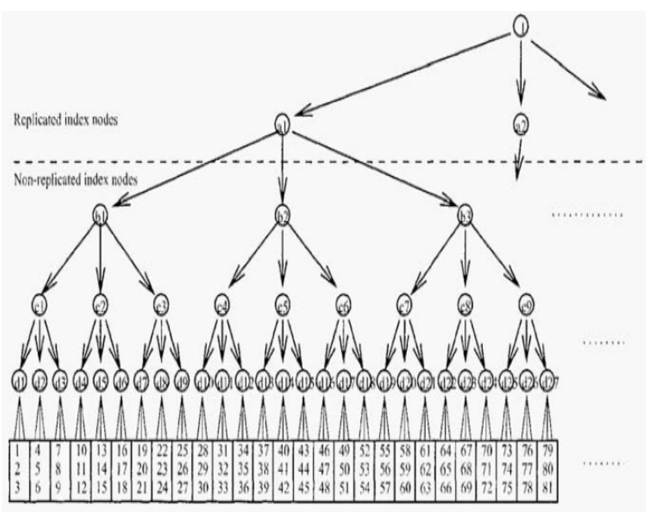

Figure 4. Tree based indexing technique

The access protocol of this scheme for a client searching for an object with key K is as follows:

1. Tune to the current bucket of the bcast. Get the offset to the next index bucket, and doze off.
2. Tune to the beginning of the designated bucket and examine the meta-data.

If the desired object has been missed, doze off till the beginning of the next bcast. Goto 2.

If the desired object is not within the data segment covered by the index bucket, doze off to next higher level index bucket. Goto 3.

If the desired object is within the data segment covered by the index bucket, goto 3.

3. Probe the designated index bucket and follow a sequence of pointers to determine when the data bucket containing the target object will be broadcast. The client may doze off in between two probes.
4. Tune in again when the bucket containing objects with key K is broadcast, and download the bucket (and all subsequent buckets as long as they contain objects with key K).

3. *Hash-based Index*

In a hash-based scheme, data are hashed into a set of partitions. Because of the non-uniform distribution of data, partitions may differ in size. Partitions are further organized into fixed size buckets. Each partition consists of one or more buckets with the last bucket being possibly partially full. The data are then broadcast in partitions as a sequence of buckets. Figure 5 shows an example of 4 partitions. In the figure, each box represents a bucket. An unfilled box represents the beginning of a partition. As shown, partitions 1, 2, 3 and 4 have 4, 6, 3 and 3 buckets respectively.

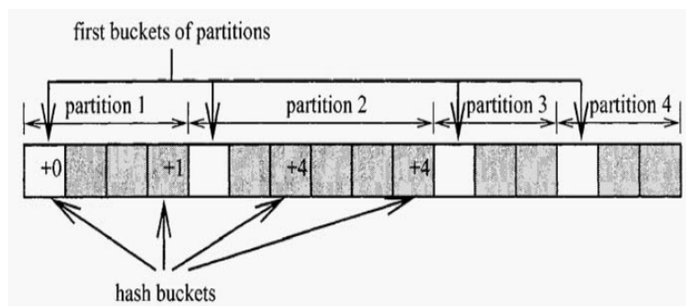

Figure 5. Hash based indexing technique

4. *Signature-based Index*

Signature techniques have been successfully employed to facilitate speedy retrieval in conventional databases [15], text databases [18], image databases [16] and multimedia databases [22]. A signature is basically an abstraction of the information stored in an object or a file. It is typically formed by hashing each value in the object into a bit string and then superimposing all bit strings generated from the object together to obtain the object signature, say Si. During filtering, a query signature, Sq, is constructed in the same manner and then compared to the signatures of all objects. An object signature matches the query signature if for every bit that is set in the query signature, the corresponding bit is set in the object signature; otherwise, the object does not match the query. Clearly, it is possible for false drops to occur. An object signature and a query signature match but the object does not satisfy the query.

Signature-based approach shown in figure 6 offers several advantages that make it a promising index structure for wireless environments [77].

- signature techniques can be applied to all kinds of information media.
- signatures are easy to generate and search with small buffer requirement.
- signatures are compact encoding of the data they represent.
- length of the signature can be controlled by varying the number of values hashed into a signature with the corresponding effect on the number of false drops.
- a signature file containing all signatures of the objects is a sequential file structure. This makes it easy to "linearize" the signature file for broadcasting on air and scanning by a mobile client, as compared to a tree- based approach, which lose its speed advantage since random access cannot be done on a broadcast channel.

There are several signature-based techniques. On the one hand, each object can be used to construct a signature called simple signature. On the other hand, we can group several objects to produce an integrated signature.

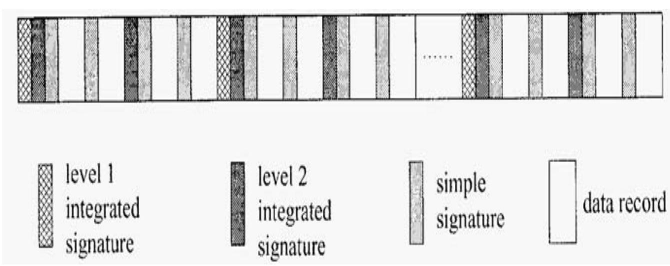

Figure 6. Signature based indexing technique

To answer a query, a query signature is generated for each level of the signatures. Let the query signatures at level i be Qi. The protocol to access objects of interest is as follows:

1. Tune to the current bucket and listen for a signature.
2. Upon receipt of the signature, compare it with the corresponding query signature at that level.
3. If it is an integrated signature

(a) If the query and integrated signatures do not match, determine when the next integrated signature at the same or higher level will be broadcast; doze off for that amount of time and goto Step 2.

(b) If there is a match, we need to look out for signatures at the next lower level. If it is not the lowest level, goto Step 2; otherwise, goto Step 4;

4. If it is a simple signature

(a) If the query and simple signatures do not match, determine when the next simple signature or integrated signature will be broadcast; doze off for that amount of time and goto Step 2.

(b) If there is a match, the corresponding object is received, and further check to see if it satisfies the query. Read the next signature. Goto Step 2.

Because signatures are compact, the tuning time is expected to be small. However, since there is no way to know whether an object exists or not unless the entirety of the signature file has been "scanned" access time is at least one broadcast cycle.

IV. WIRELESS XML DATA BROADCAST SYSTEM

In wireless XML broadcasting, the broadcast server retrieves XML data to be disseminated from the XML repository. Then, it parses and generates a wireless XML stream. The XML stream is continuously disseminated through broadcast channel. If query issued by mobile client, then mobile client tunes in to the broadcast channel and selectively downloads the XML stream for query processing.

To measure the energy-efficiency and latency-efficiency in wireless broadcasting, the tuning time and access time are used respectively [23], [20], [36]. The tuning time is the sum of the time duration spent by a mobile client to download the required data. When a mobile client downloads the data, it consumes more energy than when it waits for data. So the tuning time is used as a performance measure for energy-efficiency. The access time is the time duration from when a mobile client tunes in to the broadcast channel to when the desired data is completely retrieved from the stream. It is used as a performance measure for access-efficiency.

Much work has been conducted on efficient query processing on streamed XML data [24], [25], [26], [27], [28], [31], [34], [35], [37], [38], [39]. For providing energy-efficient query processing over XML data in wireless and mobile environments, several approaches exploiting the benefits of wireless broadcasting have been proposed to reduce structural overheads of the original XML document and attach indices containing time information to the XML data stream [29], [30], [32], [33]. These work enable mobile clients to selectively download the data of their interests by using indices. They do not contain branching information or parent-child relationship that made them not suitable for complex XPath query processing.

Several researches have been proposed to efficiently process complex XML twig pattern queries [25], [26], [27], [28], [35], [37]. MPMGJN (multi predicate merge join) algorithm [27] uses a merge join algorithm to provide higher cache utilization and superior performance than a standard RDBMS algorithm. Al-Khalifa et al. [28] investigated descendant nodes located at higher positions than their ancestor nodes in a stack. Based on this observation, they proposed efficient stack-based join algorithms. TwigStack [25] reduces the amount of the intermediate results and computational cost for merging the intermediate results using a chain of linked stacks that represent partial results to root-to-leaf query path. XR Twig [26] demonstrates superior performance because it skips elements that do not tally with given twig patterns, using an index-based algorithm.

In [35], the authors use two algorithms, a merge-based algorithm for sorted XML data and an index-based algorithm for indexed XML data, to enhance performance for matching twig queries with the OR-predicate. Kaushik et al. [37] proposed an XML path query processing algorithm integrating inverted lists and structure indices. This technique reduces computation costs omitting join operations. A number of researches have been proposed to efficiently evaluate a large number of XML queries on stream of XML documents [24], [31], [34], [38], [39]. XFilter [24] defines a Finite State Machine and converts each XPath query into a Finite State Machine representation. YFilter [31] proposes a filtering approach based on Nondeterministic Finite Automata. It improves matching performance by processing common query paths only once. Lazy DFA [34] constructs a Deterministic Finite Automaton in a lazy manner, thus, the number of states are reduced. XPush

[38] focuses on efficient evaluation of predicates. It avoids evaluation of redundant predicates by constructing a single deterministic pushdown automaton in a lazy manner. GFilter [39] addresses processing of the more complex Generalized-Tree-Pattern queries. It achieves polynomial time and space complexity by avoiding redundant predicate evaluation. Conventional XML query processing methods mainly address the problem of efficiency and scalability. None of these approaches focuses on the energy-efficiency issue.

Several approaches have been proposed for energy and access latency efficient XML query processing in the wireless mobile environment [29], [30], [32], [33]. S-node [29] generates a XML data stream based on the unit for XML broadcasting, called a S-node. This approach constructs indices based on structural characteristics of an XML document. S-node enables mobile clients to skip irrelevant nodes using these indices, providing energy-efficiency. DIX [33] proposes a fully distributed index structure and a clustering strategy for streaming XML data. DIX preserves the location path of an element as a form of bit string, to start query processing without unnecessary waiting time. The performance of query processing is further improved by clustering DIX nodes of the same depth. Path Summary [30] and Attribute Summarization [32] exploit the benefits of the structure indexing. They improve the performance of XML query processing in terms of both access and tuning times. PSGL nodes are proposed in [40] which exploit DOM structural characteristics of XML document along with level and sibling indices for selective tuning and path stream grouping of nodes for access efficient query processing.

V. SUMMARY

XML has become the de-facto standard of data representation for information exchange over the distributed environment. Hence the blooming wireless data dissemination applications make use of XML data representation for their information dissemination. The structural characteristic of XML is exploited for efficient wireless streaming techniques.

It is therefore important to determine good bucket interleaving strategy in order to obtain a good trade-off between access time and tuning time. Thus reducing the time for which a client has to wait for relevant data and reducing battery consumption during query processing.

Coinparing with the (1, m) index, the tree-based scheme has lower access time since it has a shorter broadcast cycle as not the entirety of the index is replicated. Its tuning time is also comparable to that of the (1, m) scheme. A signature file containing all signatures of the objects is a sequential file structure. This makes it easy to “linearize” the signature file for broadcasting on air and scanning by a mobile client, as compared to a tree- based approach, which lowers its speed advantage since random access cannot be done on a broadcast channel.

In this paper, a brief review on wireless data broadcasting along with XML broadcasting is provided. This opens up many research issues in wireless XML broadcasting, which can be resolved in the near future.

ACKNOWLEDGMENT

I would like to thank the authors namely T. Imielinski, S. Viswanathan, and B.R. Badrinath for their significant contribution in this research area.

REFERENCES

[1] T. Limielinski and S Viswanathan. Adaptive wireless information systems. In proceedings of the Special Interest Group in Database Systems (SIGDBS) Conference, Pages 19-41, Tokyo, Japan, October 1994

[2] S. Hameed and N. H. Vaidya. Efficient algorithms for scheduling data broadcast. ACM/Baltzer Journal of Wireless Networks (WINET), 5(3):183–193, 1999.

[3] D. Aksoy and M. Franklin. R x W: A scheduling approach for largescale ondemand data broadcast. IEEE/ACM Transactions on Networking, 7(6):846–860, December 1999.

[4] S. Acharya and S. Muthukrishnan. Scheduling ondemand broadcasts: New metrics and algorithms. October 1998.

[5] S. Acharya, M. Franklin, and S. Zdonik. Balancing push and pull for data broadcast. In Proceedings of ACM SIGMOD Conference on Management of Data, pages 183–194, Tucson, AZ, USA, May 1997.

[6] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik. Broadcast disks: Data management for asymmetric communications environments. In Proceedings of ACM SIGMOD Conference on Management of Data, pages 199–210, San Jose,CA, USA, May 1995.

[7] W.-C.Lee, Q. L. Hu, and D. L. Lee. A study of channel allocation methods for data dissemination in mobile computing environments. ACM/Baltzer Journal of Mobile Networks and Applications (MONET), 4(2):117–129, 1999.

[8] Q L Hue, D L Lee and W C Lee. Performance evaluation of a wireless hierarchial data dissemination system. Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom’99), pages 163-173, Seatle, WA,USA, August 1999.

[9] C J Su, L. Tassiulas and V J Tsotras. Broadcast scheduling for information distribution. ACM/Baltzer Journal of Wireless Networks (WINET), 5(2):137-147, 1999

[10] J W Wong, Broadcast delivery. Proceedings of the IEEE,76(12):1566-1577, December 1998

[11] J W Wong and H D Dykeman, Architechtecture and Performance of large scale information delivery networks. In Proceedings of the 12th International Teletraffic Congress, pages 440-446, Torino, Italy, June 1988.

[12] Alexander Hall and Hanjo Taubig, Comparing Push- and Pull- Based Broadcasting. www14.in.tum.de/publicationen/2003

[13] Wang Lam and Hector Garcia-Molina, Slicing Broadcast Disks, 2003

[14] R. Alonso and S. Ganguly. Query optimization for energy efficiency in mobile environment. In Proceedings of the 1993 Workshop on Optimization in Databases, Aigen, Austria, September 1993.

[15] W.W. Chang and H.J. Schek. A signature access method for the starburst database system. In Proceedings of the 15th VLDB Conference, Amsterdam, The Netherlands, August 1989.

[16] T. S. Chua, K. L. Tan, and B. C. Ooi. Fast signature-based color-spatial image retrieval. In Proc. of the International Conference on MutIimedia Computing and Systems '97, pages 362-269, June 1997.

[17] D. Comer. The ubiquitous b-tree. ACM Computing Surveys, 11(2):121-137, June 1979.

[18] C. Faloutsos and S. Christodoulakis. Signature files: An access method for documents and its analytical performance evaluation. ACM Transactions on Office Information Systems, 2(4):267-288, 1984.

[19] J. Flinn and M. Satyanarayanan. Powerscope: A tool for profiling the energy usage of mobile applications. In Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, February 1999.

[20] T. Imielinski, S. Viswanathan, and B.R. Badrinath. Energy efficient indexing on air. In Proceedings of the I994 ACM-SIGMOD International Conference on Management of Data, pages 25-36, June 1994.

[21] J.R. Lorch and A.J. Smith. Software strategies for portable computer energy management. IEEE Personal Communications, 5(3), June 1998.

[22] M.O. Rabin. Efficient dispersal of information for security, load balancing and fault tolerance. Journal of the ACM, 36(2):335-348, April 1989.

[23] T. Imielinski, S. Viswanathan, and B. Badrinath, “Data on Air: Organizationand Access,” IEEE Trans. Knowledge and Data Eng., vol. 9, no.3, pp. 353-372, June 1997.

[24] M. Altinel and M. Franklin, “Efficient Filtering of XML Documents for Selective Dissemination of Information,” Proc. Int’l Conf. Very Large Data Bases (VLDB), pp. 53-64, 2000.

[25] N. Bruno, D. Srivastava, and N. Koudas, “Holistic twig joins: Optimal XML pattern matching,” Proc. ACM SIGMOD Conf., pp. 310-321, 2002.

[26] H. Jiang, W. Wang, H. Lu, and JX Yu, “Holistic twig joins on indexed XML documents,” Proc. VLDB Conf., pp. 273-284, 2003.

[27] C. Zhang, J. F. Naughton, D. J. DeWitt, Q. Luo, and G. M. Lohman,“On supporting containment queries in relational database management systems,” Proc. ACM SIGMOD Conf., pp. 425-436, 2001.

[28] D. Srivastava, S. Al-Khalifa, H. V. Jagadish, N. Koudas, J. M. Patel, and Y. Wu, “Structural joins: A primitive for efficient XML query pattern matching,” Proc. Int’l Conf. Data Eng. (ICDE), pp. 141-152, Feb. 2002.

[29] C. -S. Park, C. S. Kim, and Y. D. Chung, “Efficient stream organization for wireless broadcasting of xml data,” Proc. Int’l Conf. Asian Computing Science Conference, pp. 223-235, 2005.

[30] S. H. Park, J. H. Choi, and S. Lee, “An Effective, Efficient XML Data Broadcasting Method in Mobile Wireless Network,” Proc. DEXA Conf., pp. 358-367, 2006.

[31] Y. Diao, M. Altinel, M. Franklin, H. Zhang, and P.M. Fischer, “Path Sharing and Predicate Evaluation for High-Performance XML Filtering,” ACM Trans. Database Systems, vol. 28, no. 4, pp. 467-516, 2003.

[32] J. P. Park, C. -S. Park, and Y. D. Chung, “Attribute Summarization: A technique for wireless XML streaming,” Proc.Int’l Conf. Interaction Sciences: Information Technology, Culture and Human, pp. 492-496, Dec. 2009.

[33] J. P. Park, C. -S. Park, and Y. D. Chung, “Energy and Latency Efficient Access of Wireless XML Stream,” Journal of Database Management, vol. 21, no. 1, pp. 58-79, 2010.

[34] T.J. Green, A. Gupta, G. Miklau, M. Onizuka, and D. Suciu, “Processing XML Streams with Deterministic Automata and Stream Index,” ACM Trans. Database Systems, vol. 29, no. 4, pp. 752-788, 2004.

[35] H. Jiang, H. Lu, and W. Wang, “Efficient Processing of XML Twig Queries with OR-Predicates,” Proc. Of ACM SIGMOD Conf., pp. 59-70, June 2004.

[36] Y. D. Chung, S. Yoo, and M. H. Kim, “Energy- and Latency-Efficient Proessing of Full- Text Searches on a Wireless Broadcast Stream,” IEEE Trans. Knowledge and Data Eng., vol. 22, no. 2, 2010.

[37] R. Kaushik, R. Krishnamurthy, J. F. Naughton, and R. Ramakrishnan, “On the Integration of Structure Indexes and Inverted Lists,” Proc. ACM SIGMOD Conf., June, 2004.

[38] A. Gupta and D. Suciu, “Stream Processing of XPath Queries with Predicates,” Proc. ACM SIGMOD Conf., pp. 419-430, 2003.

[39] S. Chen, H. Li, J. Tatemura, W. Hsiung, D. Agrawal, and K. S. Candan, “Scalable Filtering of Multiple Generalizaed-Tree-Pattern Queries over XML Streams,” IEEE Trans. Knowledge and Data Eng., vol. 20, no. 12, 2008.

[40] Prabhavathy, P., and Bose.S "Path stream group level encoding: Efficient wireless XML streaming." Recent Trends in Information Technology (ICRTIT), 2013 International Conference on. IEEE, 2013.