# Data Dissemination in Mobile Computing Environment

**S Krishna Mohan Rao**[1] and **Dr. A Venugopal Reddy**[2]

**Abstract -** ***Data dissemination in asymmetrical communication environment, where the downlink communication capacity is much greater than the uplink communication capacity, is best suited for mobile environment. In this architecture there will be a stationary server continuously broadcasting different data items over the air. The mobile clients continuously listen to the channel and access the data of their interest whenever it appears on the channel and download the same. The typical applications of such architecture are stock market information, weather information, traffic information etc. The important issue that is to be addressed in this type of data dissemination is – how quickly the mobile clients access the data item of their interest i.e. minimum access time so that the mobile clients save the precious battery power while they are on mobile. This paper reviews the various techniques for achieving the minimum access time. The advantages and disadvantages are discussed and explored different research areas for achieving the minimum access time.***

## 1. INTRODUCTION

There are two fundamental information delivery methods for wireless data 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 the outstanding requests for that item simultaneously. As such, 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 [1, 2], the server disseminates information using a periodic/aperiodic broadcast program (generally without any intervention of clients). In on demand broadcast [3, 4], the server disseminates information based on the outstanding requests submitted by clients;

In hybrid broadcast [5, 6, 7], push based broadcast and on demand data deliveries are combined to complement each other. Consequently, there are three kinds of data scheduling methods (i.e., *push based scheduling*, *on demand scheduling*, and *hybrid scheduling*) corresponding to these three data broadcast models.

## 2. PUSH BASED DATA SCHEDULING

In push based data broadcast, the server broadcasts data proactively to all clients according to the broadcast program generated by the data scheduling algorithm. The broadcast program essentially determines the order and frequencies that the data items are broadcast in. The scheduling algorithm may make use of precompiled access profiles in determining the broadcast program. In the following, two typical methods for push based data scheduling are described, namely *flat broadcast* and *broadcast disks.*

***Flat Broadcast*** The simplest scheme for data scheduling is flat broadcast. With a flat broadcast program, all data items are broadcast in a round robin manner. The access time for every data item is the same, i.e., half of the broadcast cycle. This scheme is simple, but its performance is poor in terms of average access time when data access probabilities are skewed.

***Broadcast Disks*** Hierarchical dissemination architecture, called *Broadcast Disk* (Bdisk), was introduced in [1]. Data items are assigned 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 the relative broadcast frequencies assigned to the disks. This is achieved by further dividing each disk into smaller, equal size units called *chunks*, broadcasting a chunk from each disk each time, and cycling through all the chunks sequentially over all the disks. A *minor cycle* is defined as a sub cycle consisting of one chunk from each disk. Consequently, data items in a minor cycle are repeated only once. The number of minor cycles in a broadcast cycle equals the Least Common Multiple (LCM) of the relative broadcast frequencies of the disks. Conceptually, the disks can be conceived as real physical disks spinning at different speeds, with the faster disks placing more instances of their data items on the broadcast channel.

However, if the number of minor cycles in a broadcast cycle is not equal the Least Common Multiple (LCM) of the relative broadcast frequencies of the disks, dividing precisely the desired number of chunks, is a problem. [13] addressed this problem by suggesting to fill up the disk with other information and making it divisible so that the number of minor cycles is equal to the LCM of relative broadcast frequencies.

## 3. ON-DEMAND DATA SCHEDULING

Push based wireless data broadcasts are not tailored to a particular user's needs but rather satisfy the needs of the majority. Further, push-based broadcasts are not scalable to a large database size and react slowly to workload changes. To alleviate these problems, many recent research studies on wireless data dissemination have proposed using on-demand data broadcast (e.g., [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 (if necessary) at the server upon arrival. The server repeatedly chooses an item from among the

[1]*Head-MCA, Mahaveer Institute of Science & Technology, Bandlaguda, (Post): Kesavagir, Hyderabad – 500 005*
[2]*Professor & Dean – CSE, O U College of Engineering, Osmania University Hyderabad – 500 007*
*E-Mail: [1]skrishnamohanrao@yahoo.com*

outstanding requests, broadcasts it over the broadcast channel, and removes the associated request(s) from the queue. The clients monitor the broadcast channel and retrieve the item(s) 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.

## 4. HYBRID DATA SCHEDULING

Push-based data broadcast cannot adapt well to a large database and a dynamic environment. On-demand data broadcast can overcome these problems. However, it has two main disadvantages: i) more uplink messages are issued by mobile clients, thereby adding demand on the scarce uplink bandwidth and consuming more battery power on mobile clients; ii) if the uplink channel is congested, the access latency will become extremely high. A promising approach, called hybrid broadcast, is to combine push-based and on-demand techniques so that they can complement each other. In the design of a hybrid system, one of the main issues is the assignment of a data item to push-based broadcast, on-demand broadcast or both.

Concerning this issue, there are different proposals for hybrid broadcast in the literature. In the following, we introduce the techniques for balancing push and pull and adaptive hybrid broadcast.

***Balancing Push and Pull:*** Hybrid architecture was first investigated in [10, 11]. In that model, items are classified as either frequently requested (*frequest*) or infrequently requested (*irequest*). It is assumed that clients know which items are *frequests* and which are *irequests*.

The model services *frequests* using a broadcast cycle, and *irequests* using on-demand. 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 *irequest* queue (if at least one such request is waiting). 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 [14], 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 [15], 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.

## 5. DATA ALLOCATION OVER MULTIPLE BROADCAST CHANNELS

Multiple physical channels have capabilities and applications that can not be mapped on to single channels. As stated in [12] some example advantages include better fault tolerance, configurability and scalability. By having access to multiple physical channels fault tolerance is improved. For example if a server broadcasting on a certain frequency crashes, its work must be migrated to another server. If this server is already broadcasting on another frequency it can only accept the additional work if it has the ability to access multiple channels. More flexibility is allowed in configuring broadcast servers. Assume that there are two contiguous cells, which contain broadcast servers that transmit at different channels. A single server that wishes to take over the responsibility of transmitting in both cells can only do so if it can transmit over multiple channels. Finally, being able to transmit over multiple channels has scalability benefits. A broadcasting system must be able to handle both high powered and low powered clients. In order to do so, multiple channels can be used and clients can monitor a number of channels commensurate to their capacities and data needs.

This calls for a data-scheduling algorithm, which works dynamically, and allocates data according to changing access patterns to achieve efficient data access and channel utilization so that the access time is minimum. However, the area of interest is hoe to adjust the broadcast program when the data items are changing dynamically. This calls for a research on incremental algorithms to change the programs dynamically.

[16] explores the problem of adjusting broadcasting programs to effectively respond to the changes of data access frequencies, and develop an efficient algorithm DL to address this aspect. The DL algorithm showed the high quality of results and close to the optimal ones.

[17] explains the effects of dynamicity on broadcast program with respect to *item placement, Disk structure, Disk content* and *Disk Values. Item placement or Disk Structure* changes the relative frequencies and/ or order of appearance of data items already being broadcast. The *value of Data item* changes only when it is updated. Dynamicity due to *Disk contents* does not influence the items that appear on broadcast.

[18] explores the problem of dynamic data and channel allocation with the number of communication channels and the number of data items are given. Algorithm SOM is a composite algorithm which will cooperate with 1) a search strategy and 2) a broadcast program generation. However the algorithm is not easy for implementation.

[10] proposes optimal allocation algorithm which searches exhaustively to find the optimal solution for channel allocation and data page organisation. However, the execution of algorithm is very slow

## 6. CONCLUSION & FUTURE STUDY

This paper discusses various techniques for data dissemination in mobile communication environments. Data scheduling methods are investigated with respect to their performance in minimum access time. For data scheduling push based, on-demand, hybrid, and multi channel broadcast were discussed. Push based broadcast is attractive when access patterns are known before hand, while on-demand broadcast is desirable for dynamic access patterns. Hybrid data broadcast offers more flexibility by combining push-based and on-demand broadcasts. Broadcasting in multi channel does have advantages in terms of high fault tolerance. The research areas like scheduling data items dynamically by employing incremental algorithms are identified. Another research area of interest is how the server gets the feedback from the mobile clients regarding their access patterns so that it adjusts the scheduling program accordingly. The authors are working on developing incremental algorithm for the multi channel data scheduling so that the server program adjusts itself to the dynamically changing data access patterns of the mobile clients.

## REFERENCES


[1] 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.

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

[4] 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.

[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] 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

[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] Wai Gen Yee, Shanmukhanath, B Navathe, Edward Omaian Ciniski and Christopher Jermaine. Efficient Data Allocation over multiple channel at Broadcast Servers. IEEE Transactions on Computers, Volume 51, No 10, October 2002.

[13] Karl Aberer, Data Broadcasting in Mobile Networks, EPFL-SSC, laboratore de systemes d'informations repartis, pages 1-38, 2004

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

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

[16] Een-Chih Peng, Jiun-Long Huang and Ming Syan Chen, Dynamic Levelling: Adaptive Data Broadcasting in a Mobile Computing Environment. Mobile Networks and Applications 8,355-364, 2003

[17] R K Ghosh, Data Dissemination, Mobile Computing, IIT – Kanpur, pages 1-70, 2006

[18] jiun-Long Huang, Wen-Chih Peng and Ming-Syan Chen. SOM: Dynamic Push-Pull Channel Allocation Framework for Mobile Data Broadcasting. IEEE Transactions on Mobile Computing Vol 5, No. 8 August 2006

[19] S Krishna Mohan Rao and Dr A Venugopal Reddy: Optimal allocation algorithm for data broadcasting programs in a mobile computing environment. Proceedings of National Conference INDIACom-2007.


***Continued from page no. 56***

## REFERENCES


[1] A.K. Pujari 2002. "Data Mining Techniques", University press 2002.

[2] Ana Cardoso-Cachopo Arlindo L. Oliveira 2006, "Empirical Evaluation of Centroid-based Models for Single-label Text Categorization",INESC-ID Technical Report 2006

[3] Ken Williams 2006, "A Framework for Text Categorization", Web Engineering Group The University of Sydney Bldg J03, Sydney NSW 2006

[4] Makoto 2006, "Hierarchical Bayesian Clustering for Automatic Text Classification", Proceedings of the Fifth

International Conference on Machine Learning and Cybernetics, Dalian, 13-16 August 2006
[5] Andreas Hotho 2005. "A Brief Survey of Text Mining" KDE Group University of Kassel May 13, 2005
[6] Fafal Rak, Wojciech Stach 2005."Considering Re-occurring Features in Associative Classifiers", Advances in Knowledge Discovery and Data Mining: 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005. Proceedings
[7] Goutte, C. and Gaussier, E. 2005, "A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation." In Pro-ceedings of the 27th European Conference on Information Retrieval, pages 345-359.
[8] Huan Liu, and Lei Yu 2005. "Toward Integrating Feature Selection Algorithms for Classification and Clustering", Knowledge and Data Engineering, IEEE Transactions on April 2005 Vol. 17, pages 491- 502
[9] Huang, J. and Ling, C. 2005, "Using AUC and Accuracy in Evaluating Learning Algorithms". IEEE Trans. on Data and Knowledge Engineering,17 3 3 :299-310.
[10] Kathrin Eichler 2005, "Automatic Classification of Swedish Email Messages",17th August 2005
[11] Kiritchenko, S., Matwin, S., Nock, R.,and Famili, A. 2005, "Learning and Evaluation in the Presence of Class Hierarchies": Application to Text Categorization. Submitted.
[12] Mohammed J. Zaki 2005, "Efficient Algorithms for Mining Closed Item sets and Their Lattice Structure" IEEE transactions on knowledge and data engineering, vol. 17, no. 4, april 2005

***Continued from page no. 54***

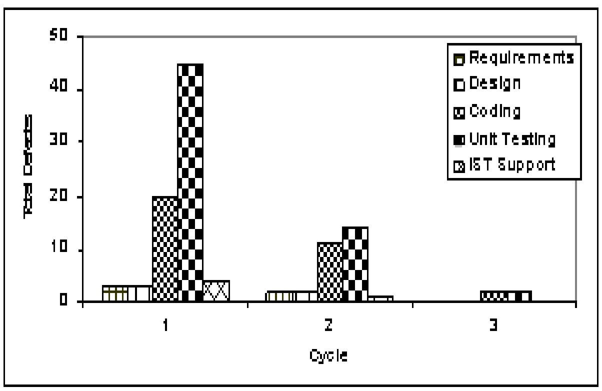

Figure 23: Total defects analysis for SD module

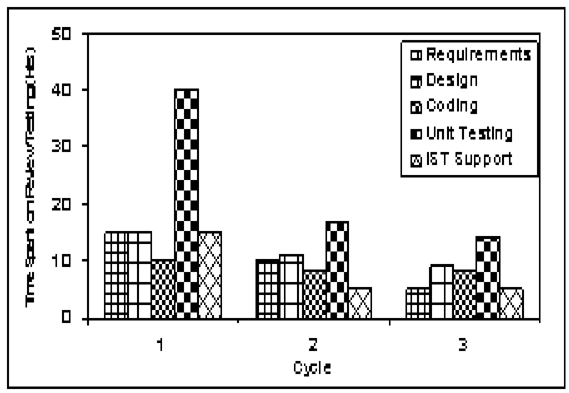

Figure24: Time spent on Review/testing (Hrs) analysis for SD module

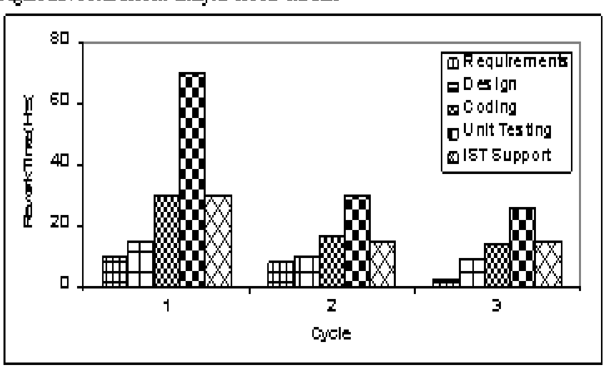

Figure 25: Rework Time (Hrs) analysis for SD module

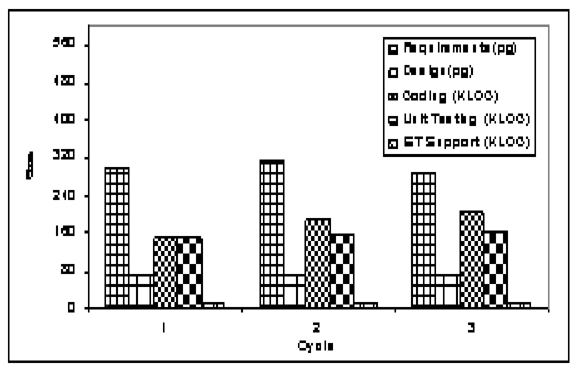

Figure 26: Size (pg/KLOCS) analysis for SD module