

# Hybrid Scheduling for Heterogeneous Asymmetric Mobile Communication

R.Kavitha

Siddhartha Institute of  
Engineering and Technology  
Ibrahimpattanam , Hyderabad

M.V.Ramana Murthy

Department of Mathematics  
Osmania University,Hyderabad

Mahaboob Sharief Shaik

Computer Science Dept  
Faculty of Computing &  
Information Technology  
King Abdulaziz University  
Jeddah, Saudi Arabia

## ABSTRACT

Rapid development in wireless communication technology has paved a path for the increase in mobile users to access the information from anywhere and at anytime. Data broadcasting plays a vital role in asymmetrical communication environment. Here server disseminates the information required by the mobile users at a faster rate as the bandwidth of the downlink channel is much more than the uplink. In data broadcasting, the mobile users put their request for the required data item and continuously listen for the response upon the channel from the server. In order to save the precious battery power of the mobile clients it should be that they have to be served quickly by the server. That is the access time of the requested data item should be minimized. This paper focus on the lengths of the data items which uses stretch as the metric to devise a server broadcasting algorithm in order to achieve optimal results. Simulation results are obtained to evaluate the performance of the proposed algorithm.

## Keywords

Access time, Hashing, Hybrid push-pull model, Multiple channels

## 1. INTRODUCTION

Broadly, all the data dissemination applications have three types of broadcast models; push, pull and hybrid push-pull data models. In push based data model, server disseminates the data irrespective of client's request [5]. In pull based data model, data is transferred by the server upon the client's demand [3, 6, 13]; where as in hybrid push-pull data model, most significant data is broadcasted in the form of a broadcast disk and the rest of the data is pulled from the server via explicit client requests [2, 12].

To retrieve individual records of interest, mobile users can either monitor the broadcast channel for the arrival of desired data or issue a pull request to the server.

In broadcast scheduling, majority of researchers focus on push-based broadcasting model [3, 7, 10, 11]. In push-based broadcasting, a pre-compiled access based periodic broadcasting program is used by the server exclusive of the user intervention to deliver the data. The limitation of such broadcasting schemes is that they can't adopt dynamic changes.

Pull-based (on-demand) broadcasting technique, broadcast dynamic and huge client request data from the server to the client's request. On-demand broadcast scheduling scheme is initiated by Aksoy and Franklin [4].

But all these algorithms were restricted for different sized data. The authors of [1] defined scheduling algorithms which use

stretch as the metric for variable-sized data. Based on stretch, various algorithms have been scrutinized [14,4]. All these algorithms considered are preemptive in the sense that the scheduling decision is reevaluated after broadcasting data item and more over they have used single channel for data dissemination.

The basic idea is to use the available channels so as to achieve better overall system performance. A dynamic and large database environment can't be well acclimatized by the push-based broadcasting. Such problems can be surmounting by on-demand data broadcasting. However, it has two main disadvantage: 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.

## 1.1 Data Allocation over Multiple Channels

Broadcasting techniques have less scalability since a single broadcast response is used to satisfy many clients. This paper utilizes multiple channel broadcast to satisfy the client's requests and investigates a method to efficiently broadcast data. Multiple physical channels have capabilities and applications that can-not be mapped on to single channel.

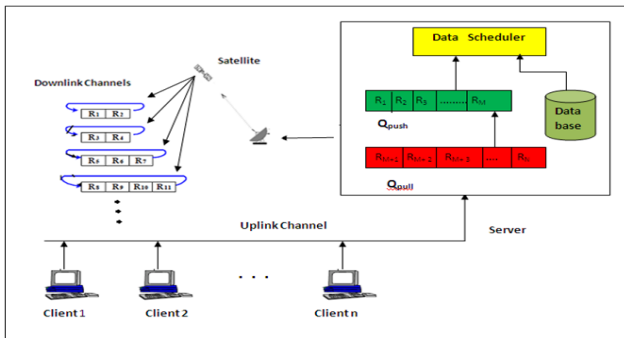
As stated in [9], the advantages of using multi channel dissemination are better fault tolerance, configurability and scalability. The main idea of organization of broadcast disks is to allocate data according to changing access patterns to achieve efficient data access and channel utilization. There are two major issues involved in organization of broadcast disks. The first issue is how to efficiently allocate channels in terms of different pull-request workloads. It is logical to have more channels for on-demand service when the workload of pull-requests is heavy and fewer channels when the workload is light [11]. The second issue is how to determine which data items should be delivered on the currently available broadcasting channels and how to organize these broadcast data items.

The remaining part of the paper is organized as follows: Section 2 discusses the model used. Section 3 proposes the variable sized data item scheduling algorithm and generates a broadcast program for hybrid push-pull model with multiple channels. Performance analysis is conducted in section 4. Section 5 ends up with conclusion and future work.

## 2. HYBRID PUSH-PULL MODEL

In a hybrid push-pull environment we have a down link channels for supporting push- based data broadcast and a low bandwidth uplink channel for supporting pull based data items. Client sends a request through the uplink channel to the server when a data item is deserved. At the server, the client requests are queued up upon arrival if necessary. Server frequently selects an item among the outstanding requests, broadcasts it over the broadcast channel. The clients monitor the broadcast channel and retrieve the item(s) they require.

This model serves for the data items with the same size. But the same architecture can be extended to incorporate selection of heterogeneous data items (different item lengths) along with the requests for their broadcasting. Hence this motivates to use the stretch metric.



**Fig.1: Hybrid Push-Pull model for the proposed algorithm**

Fig.1 shows the proposed architecture which assumes a single server serving multiple clients. Server has a database which consists of N number of data items with variable lengths, and each item  $i$  has different access probability. Here the size of the item is considered as the service time of that item, as service time is dependent on the size.

Simple Hashing technique is adopted in this hybrid algorithm as the push mechanism which is applied on the push queue. On the other hand we select the data item which has maximum stretch value for pull mechanism. And we redefined stretch as

$$S_i = \frac{\text{number of requests for the } i^{\text{th}} \text{ data item}}{\text{size}_i}$$

From the Fig.1 it can be observed that the server maintains two queues: push queue ( $Q_{\text{push}}$ ) and pull queue ( $Q_{\text{pull}}$ ). The server's data scheduler picks the items based on the simple hash function from the push queue for broadcasting along the push channels and the items in the pull queue are selected for push queue based on the maximum stretch value. Periodically server monitors the access probabilities of the items with their arrival rate. Fig.2 provides the pseudo-code for the proposed Variable Sized Data Item Scheduling algorithm which is executed at the server while the push and pull queues do not change.

### Variable Sized Data Item Scheduling Algorithm:

```

While (true) do
Begin
Broadcast the selected data items from  $Q_{\text{push}}$ 
according to the Simple Hashing technique;
Handle the requests occurring during
broadcasting;
If  $Q_{\text{pull}}$  is not empty then
Select from the  $Q_{\text{pull}}$  the data item whose
stretch is maximum;
If tie
Select the item with the highest  $P_i$ ;
Clear the number of pending requests
for this data item
End
    
```

**Fig.2 Variable Sized Data Item Scheduling Algorithm**

## 3. ASSUMPTIONS AND MOTIVATIONS FOR MODELING THE SYSTEM

Before developing the Variable Sized Data Item Scheduling Algorithm (VSDISA) for hybrid push-pull environment, an analytical model has to be developed to evaluate the access efficiency of the data items for dissemination. The goal of the analysis is to estimate (i) the minimum expected delay of the data items in each channel from push queue and (ii) minimum waiting time of the items in the pull queue. After estimating the times we derive the overall waiting time of any data item in the hybrid push-pull system.

Before proceeding further, let us enumerate the parameters used in the model:

1. The server has a database which consists of N number of data items say  $R_1, R_2, R_3, \dots, R_N$  and are sorted in descending order of their access probabilities  $P_1 \geq P_2 \geq P_3 \dots \geq P_N$ . Access probabilities give the item's popularity among clients.  $P_i$  is the access probability of  $i^{\text{th}}$  data item, which follow the ZipF's distribution with access skew-coefficient  $\theta$  [16]:  

$$P_i = \frac{(1/i)^\theta}{\sum_{j=1}^N (1/j)^\theta}$$
 Every data item has different size and is randomly selected between 1 and Size where Size is the maximum size.
2. The server pushes M data items and N-M data items are pulled by the clients. Therefore the total access probabilities of push and pull queues are  $\sum_{i=1}^M P_i$  and  $\sum_{i=M+1}^N P_i$
3. Let  $\lambda$  be the arrival rate of the system and arrival rate of the clients requesting for the items for the pull model is defined as  $\lambda_{\text{pull}} = (1 - \sum_{i=M+1}^N P_i) * \lambda$ .
4. Server supports C wireless channels among which  $C_{\text{push}}$  be the number of push channels for broadcasting the items and  $C_{\text{pull}}$  are the number of pull channels to collect the requests from the clients.

### Minimum Expected Delay for Push Model:

In order to allocate data items on to the broadcast channels we use Simple Hashing technique which is defined as

$$F(S_i) = S_i \bmod C_{\text{Push}} \quad (\text{Eq. 1})$$

From the given number of push (broadcast) channels and the access probabilities of the data items we have determined the number of data items that are allocated in each broadcast

channel using Eq.1 for calculating minimum expected delay of all the data items. The mean expected delay of all the data items in the push channels is

$$MED_{Q_{push}} = \frac{W_{time\ in\ Q_{push}}}{M} \quad (Eq. 2)$$

Where

$$W_{time\ in\ Q_{push}} = \sum_{j=1}^{C_{push}} \left( \sum_{i=1}^{N_i} \frac{(N_i-i)}{N_i} \right) \times \sum_{q=1}^{N_i} p_q * size_q$$

### Minimum Waiting Time for Pull Model:

We have assumed  $C_{pull}$  number of channels we use M/M/1 Queue model for calculating the waiting time for the items in the pull queue. In this model the time between successive pull requests and the service time of the server is exponentially distributed. From Steady State System of queuing theory, the mean waiting time of the data items in the pull queue ( $Q_{pull}$ ) obtained by the Eq.3

$$MWT_{Q_{pull}} = \frac{\lambda_{pull}}{\mu_{pull}(\mu_{pull} - \lambda_{pull})} \quad (Eq. 3)$$

Where

$$\mu_{pull} \text{ is the service time } \mu_{pull} = \sum_{i=M+1}^N size_i$$

Thus the mean access time,  $MAT_{hybrid}$ , of our hybrid model is given by :

$$MAT_{hybrid} = MED_{Q_{push}} + MWT_{Q_{pull}} \quad (Eq. 4)$$

## 4. EXPERIMENTAL RESULTS

In this section the simulation model is used to explore the performance of VSDI Scheduling Algorithm. The primary performance metric employed in this study is the mean access time, measured in *broadcast units* and goal is to minimize mean access time. The effectiveness of VSDI Scheduling Algorithm is studied in terms of mean access time obtained by varying the number of channels, the database size and the ZipF ( $\theta$ ) parameter. The results obtained from the studies are analyzed.

### 4.1 Simulation Model

The simulation model is implemented by converting the VSDIS Algorithm in CPP programming. The simulation is run on Pentium – IV computer with 512 MB RAM with 80 GB hard disk.

### 4.2 Comparative Analysis

The following assumption are made for the simulation

- Total number of data items  $N=120$  and  $M=100$
- Arrival rate,  $\lambda$ , varied between 5-20.
- Sizes of the data items selected randomly between 1-5.
- Access probabilities of the data item are changed dynamically from 0.2 to 0.8
- To compare the performance of our hybrid system, we have chosen 3 different hybrid scheduling algorithms [8, 17, and 18] as performance benchmark.

It is observed from Fig. 3 that our algorithm results in accessing the items with less time compared to the existing algorithm [18] with different arrival rates. It can be inferred from Fig. 4 that our algorithm consistently gains over existing algorithm [17, 8] with different arrival rates.

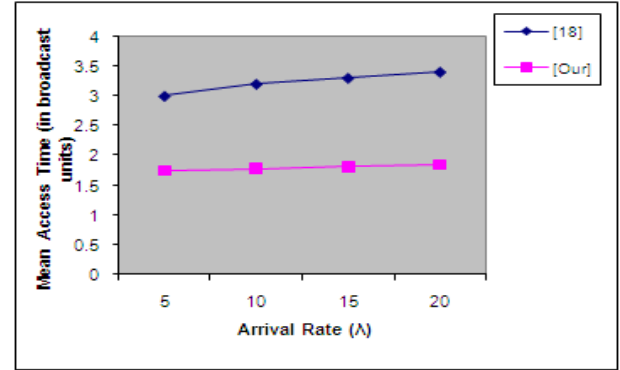


Fig. 3 Performance comparison with different arrival rates

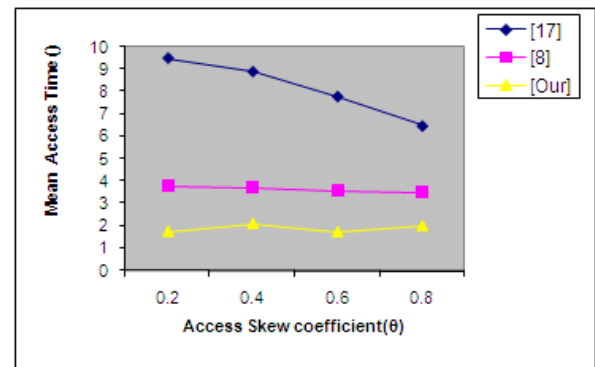


Fig. 4 Performance comparison with different Skew coefficients

We have also taken look into distribution of data items among different channels. With the help of Simple Hashing Technique used, the items are distributed among the channels with varying Skew Coefficients as shown in Fig. 5.

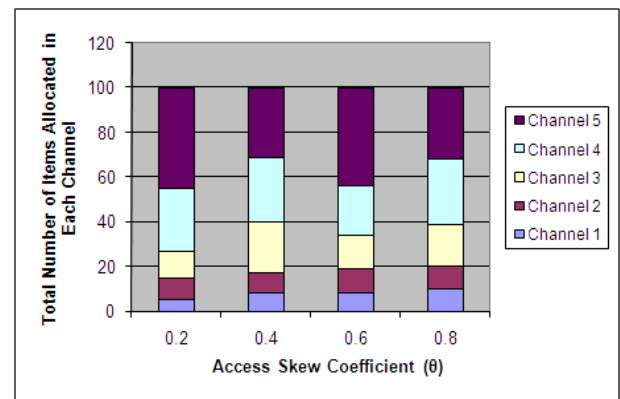


Fig.5 of data items allocated to each Channel with varying Skew coefficient

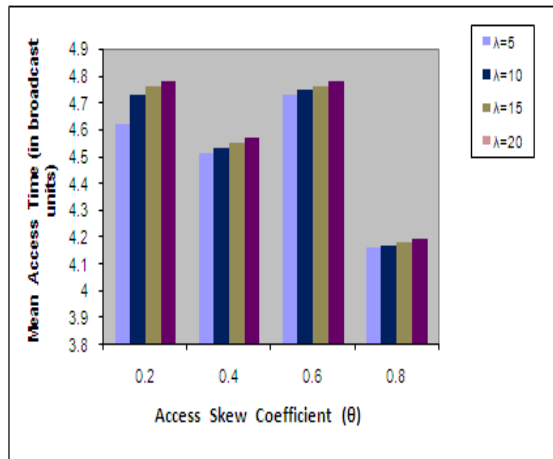


Fig.6 Varying Mean Access Time with Skew Coefficient

## 5. CONCLUSIONS AND FUTURE WORK

This paper has described the issue of generating a broadcast program for hybrid push-pull environment given the number of channels and the access probabilities of the data items. For the multiple channels, VSDIS algorithm is developed to minimize the access time. Performance of VSDIS algorithm was analyzed and sensitivity analysis on several parameters, including the number of data items, the number channels, access skew coefficient and arrival rate are conducted. It was shown by the simulation results that the proposed algorithm is of very high quality with good scalability which is very important for heterogeneous mobile computing environment. Here in the proposed algorithm it is assumed that the system has a single server but the research can be extended to have multiple servers.

## 6. ACKNOWLEDGEMENTS

The authors extend their profoundly sincere thanks to all the authors and publishers mentioned in the references for their most valuable contribution in this research used directly or indirectly since without their contribution this paper could not have been completed for further use by the beginners in the field.

## 7. 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. 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.
- [3] S. Acharya and S. Muthukrishnan. Scheduling ondemand broadcasts: New metrics and algorithms. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'98)*, pages 43–54, Dallas, TX, USA, 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. Hameed and N. H. Vaidya. Efficient algorithms for scheduling data broadcast. *ACM/Baltzer Journal of Wireless Networks (WINET)*, 5(3):183–193, 1999.
- [6] Q. L. Hu, D. L. Lee, and W.-C. Lee. Performance evaluation of a wireless hierarchical data dissemination system. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'99)*, pages 163–173, Seattle, WA, USA, August 1999.
- [7] T. Imielinski and S. Viswanathan. Adaptive wireless information systems. In *Proceedings of the Special Interest Group in DataBase Systems (SIGDBS) Conference*, pages 1941, Tokyo, Japan, October 1994.
- [8] Navrati Saxena, Cristina M, Kalyan Basu, Sajal K. Das. A Dynamic Hybrid Scheduling Algorithm for heterogeneous asymmetric environment., Vol:20 April-2005 , IJPDES.
- [9] Wai Gen Yee, Shamakanth, B Navathe, Edward Omian Ciniski and Christopher Jermaine. Efficient Data Allocation over multiple channel at Broadcast Servers. *IEEE Transactions on Computers*, Volume 51, No.10, October 2002.
- [10] 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.
- [11] C. W. Lin and D. L. Lee. Adaptive data delivery in wireless communication environments. In *Proceedings of the 20th IEEE International Conference on Distributed Computing Systems (ICDCS'2000)*, pages 444–452, Taipei, Taiwan, April 2000.
- [12] 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.
- [13] N. H. Vaidya and S. Hameed. Scheduling data broadcast in asymmetric communication environments. *ACM/Baltzer Journal of Wireless Networks (WINET)*, 5(3):171–182, 1999.
- [14] J. W. Wong. Broadcast delivery. *Proceedings of the IEEE*, 76(12):1566–1577, December 1988.
- [15] Yiqiong Wu, Jing Zhao, Min Shao, and Guohong Cao, "Stretch-Optimal Scheduling for On-Demand Data Broadcasts", © 2005 Springer Science
- [16] G. Zipf, Human Behaviour and the Principle of Least Effort, Addison-Wesley.
- [17] Y. Guo, S. K. Das and M. C. Pinotti. A New Hybrid Broadcast scheduling Algorithm for Asymmetric Communication Systems: Push and Pull Data based on Optimal Cut-Off Point. In *Mobile Computing and Communications Review (MC2R) Vol.5, No.4*, 2001.
- [18] W. Ni, Q. Fang and S. V. Vrbsky, "A Lazy Data Approach for On-demand Data Broadcasting", *Proc. of 23rd Intl. Conf. on Distributed Computing Systems Workshops (ICDCSW03)*