International Journal of Engineering Research in Computer Science and Engineering Engineering (IJERCSE) Vol 2, Issue 3, March 2015

Data Broadcasting for Multiple Request In a Multiple Channel Mobile Environment

[1]Kayalvili.E,Ohmshree.R(Student),[2]Mrs.K.Rejini(Assistant Professor)
Department Of Computer Science & Engineering Anand Institute Of Higher Technology, Chennai,India
jillushravs@gmail.com , shreema93@gmail.com

Abstract — **Wireless Data broadcast has been a widely used technique of disseminating data to users. In this paper, we investigate the data retrieval problem in both push-based and pull-based broadcasts. When user only retrieve one data item per request, the retrieving process is straightforward. However, it is common that a user requests multiple data items at a time. In addition, the fast development of wireless communication technologies such as OFDM (Orthogonal Frequency Division Multiplexing) makes efficiently broadcasting through multiple channels possible .In the last decade, how to allocate data items onto multiple channels to minimize the expected response time has become a hot research topic which captured a great deal of attentions. It is clear that, Largest Number Data Retrieval given a deadline, when users want to download as many requested data items as possible. Minimum Cost Data Retrieval with the objective of minimizing the response time and energy consumption. We also propose a heuristic algorithm for it based on maximum independent set. For the case that all channels are synchronized, we propose a polynomial time optimal algorithm for LNDR.**

Index Terms — **Wireless dataBroadcast, Heuristic algorithm, Minimum Cost Data Retrieval, Largest Number Data Retrieval, Pull Based System, Multi Channels.**

I. INTRODUCTION

1.1 Data Retrieval for Wireless Data Broadcast

In mobile communication environments, wireless data broadcast has been a widely used technique of disseminating data to users. It is especially suitable for providing users public information, such as weather reports, traffic conditions, and stock quotes, because of its scalability and flexibility. Generally, there are two types of broadcast systems: push-based and pull-based. In broadcast push the server repeatedly sends information to the clients without explicit client requests. Anynumber of clients can monitor the broadcast channel and retrieve data as they arrive on the broadcast channel. Ifdata is properly organized to cater to the needs of the clients, such a scheme makes an effective use of the lowwireless bandwidth and is ideal to achieve maximal scalability.

1.2 Pull-Based System

In broadcast pull, the clients make explicit requests for data. If multiple clients request the same dataat approximately the same time, the server may aggregate these requests, and only broadcast the data once.Such a scheme also makes an effective use of the low wireless bandwidth and clearly improves user perceivedperformance. Several scheduling algorithms have been proposed that attempt to achieve maximum aggregation.

In both Push based and Pull based systems, clients have to listen to the channels, wait for the requested data items and download them when they arrive.

Mobile computing and wireless networks are fast-growing technologies that are making ubiquitous computing a reality. Mobile and wireless computing systems have found many applications, including Defense Messaging System (DMS), Digital Battlefield and Data Dissemination (BADD), and as a general-purpose computing tool. With the increasing popularity of portable wireless computers, mechanisms to efficiently transmit information to such clients are of significant interest. For instance, such mechanisms could be used by a satellite or a base station to communicate information of common interest to wireless hosts. In the environment under consideration, the downstream communication capacity, from server to clients, is relatively much greater than the upstream communication capacity, from clients to server.

For example in wireless mobile network, servers may have a relatively high bandwidthbroadcast capability while clients cannot transmit or can doso only over a lower bandwidth (e.g., cellular) link. Such systems have been proposed for many application domains,

including traffic information systems, hospital information systems, public safety applications, and wireless classrooms Such environments are, hence, called asymmetric communication environments.

Communications asymmetry can arise in two ways:
Thefirst is from the bandwidth limitations of the physical communications medium. An example of physical asymmetry
is the wireless environment as described above; stationary
servers have powerful broadcast transmitters while mobile
clients have little or no transmission capability.

Perhaps less obviously, communications asymmetry can also arise from the patterns of information flow in the application. For example, an information retrieval system in which the number of clients is far greater than the number of servers is asymmetric because there is insufficient capacity (either in the network or at the servers) to handle the simultaneous requests generated by the multiple clients.

I.ANALYSIS OF CHANNEL CONDITION

Orthogonal frequency division multiplexing is commonly implemented in many emerging communications protocols because it provides several advantages over the traditional FDM approach to communications channels. More specifically, OFDM systems allow for greater spectral efficiency reduced intersymbol interference (ISI), and resilience to multi-path distortion and makes efficient broadcasting. For the push-based programs, we adopt a dynamic programming approach for data scheduling at the server side. Besides down-link channels, clients will send requests to the server through an up-link channel, and the server will decide the broadcast schedule based on the requests received.

II.DATA RETRIEVAL SCHEDULING

SCHEDULING ALGORITHMS

In this section, in order to set the stage for the presentation of our new scheduling algorithm, we will discuss in some more detail four of the major scheduling policies previously proposed in the literature: First-Come First-Served, Shortest Service Time First, Most Requests First, and RxW.

First Come First Served (FCFS)*:* In this simple scheduling policy, requests are served and broadcast in their arrival order.

Shortest Service Time First (SSTF): The scheduler serves the data item which has the minimum resource requirements first. In our case, the downlink channel is the shared resource. Hence, the requesteddata item with the smallest size is broadcast first.

Most Requests First (MRF):The scheduler selects the most popular item to broadcast, i.e., the data item with the maximum number of pending requests.

RxW:This scheme combines the benefits of MRF and FCFS. At each broadcast cycle, the server selects item, while the value is the longest wait time for a request to that data item the data item with the highest R W value to broadcast. The R value is the number of requests for that item while the W value is the longest waiting time for a request to that data item.

It is much more likely a client query a set of data instead of only one data at a time. After obtaining the locations of requested data items, we need to make a schedule to download the data one by one in some order. The arriving time of each requested data item is already known from the index, then the energy consumption depends purely on the number of channel switching happens during the retrieval process. It can be used in almost any data broadcast programs, in which the data access frequencies, data sizes, and channel bandwidths can all be non-uniform.

2.2 Problem Definition:

Definition 1 (VDRS).*Given a request set D and a broadcast*
schedule, a Valid Data Retrieval Schedule is a set T of triples
Tr1; Tr2; . . . ; Trk, where each triple Tri corresponds to a distinct data item dTri 2 D, and there is no pairwise conflict
between any two triples.

According to Definition 1, if the channels are synchronized,
8 two distinct triples Tri; Trj 2 T , we have dTri 6¼ dTrj and tTri 6¼ tTrj . If the channels are unsynchronized, 8
two distinct triples Tri; Trj 2 T , we have dTri 6¼ dTrj and

tTri 6¼ tTrj; cTri ¼ cTrj
j tTri _ tTrj j >1; otherwise:

Definition 2 (LNDR).*Given a request set D, a broadcast schedule and a time duration ½t1; t2_, the largest number data retrieval problem is to find a valid data retrieval schedule that can download the largest number of data items in D during time ½t1; t2.*

In some time critical situations, the requests have deadlines.
When a request cannot complete, the user will prefer a data retrieval schedule to download as many requested
data items as possible. The LNDR problem is defined accordingly.

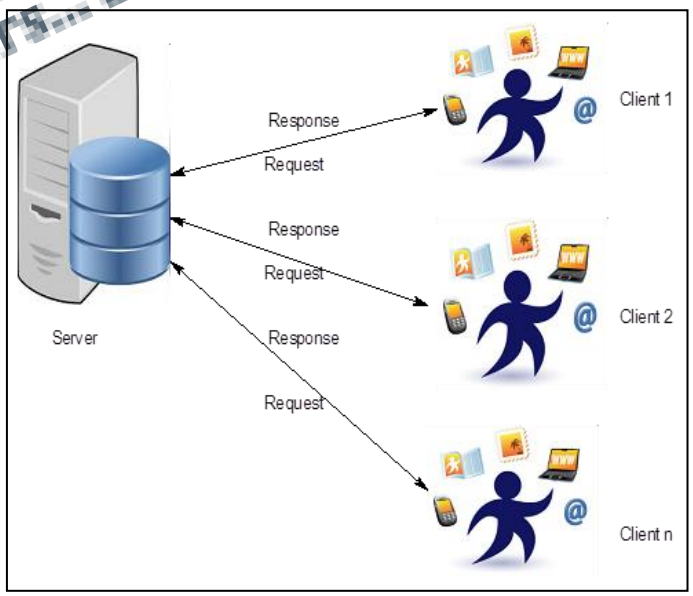
Definition 3 (MCDR). *Given a request set D, a broadcast schedule and a parameter 0 _ a _ 1, the minimum cost data*
retrieval problem is to find a valid data retrieval schedule to
download all the data items in D such that aT þ ð1 _ aÞC is
minimized, where T is the response time and C is the energy
consumption.

MCDR aims at reducing the cost of a request, not the energy consumption only, but also the response time. In
Definition 3, the total cost of a data retrieval is defined as
aT þ ð1 _ aÞC, where a is an adjustable parameter reflecting the users' preference.

In this paper, we mainly investigate the MCDR problem
in push-based broadcast. In pull-based broadcast, downloading
all the requested data items may result in a long
response time. Sometimes, the server may never broadcast a
requested data item for a query. Therefore, in pull-based
broadcast, the response time and energy consumption of a
request highly depend on the broadcast schedule at the
server side and has little to do with the data retrieval schedulingat the client side.

Largest Number Data Retrieval
If the channels are synchronized, a client can download many data item from any channel at time even though it downloads a data item from a different channel at time. In push based broadcast, LNDR can be applied directly for the cases where the data items have different sizes and the channels have different bandwidths and its performance ratio remains unchanged. That is, a data item may require multiple time slots to download. A valid retrieval schedule for an LNDR instance is a set of triples without conflicts. Thus, finding a valid schedule with the largest number of requested data items is equivalent to finding a maximum independent set. The LNDR problem takes the "deadline" into consideration and therefore also describes the time-critical scenario.

Minimum Cost Data Retrieval
We develop a greedy heuristic for MCDR. It combines the benefits of both channel scheduling to reduce the energyconsumption in channel switching and data item scheduling to reduce the response time and the energy consumption in the doze mode. The objective of MCDR is to reduce the response time and energy consumption. The index information is assumed to be obtained before data retrieving , hence the tuning time for index retrieval is not considered when calculating the energy consumption for data retrieving.



1.System Architecture

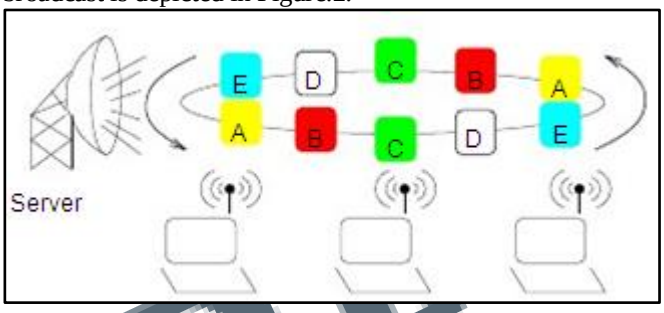
In traditional client-server information systems, clients initiate data transfers by sending requests to a server. Such systems are pull-based; the clients "pull" data from the server in order to provide data to locally running applications.

Pull-based systems are a poor match for asymmetric communications environments, as they require substantial upstream communications capabilities. To address this incompatibility, we have proposed a new information system architecture that exploits the relative abundance of downstream communication capacity in asymmetric environments. This new architecture is called Broadcast Disks. The central idea is that the servers exploit their advantage in bandwidth by broadcasting data to multiple clients. We refer to this arrangement as a push-based architecture; data is pushed from the server out to the clients. In this approach, a server continuously and repeatedly broadcasts data to the clients. In effect, the broadcast channel becomes a "disk" from which clients can retrieve data as it goes by. Broadcasting data has been proposed previously. Our technique differs, however, in two ways. First, we superimpose multiple disks of different sizes and speeds on the broadcast medium, in effect, creating an arbitrarily fine-grained memory hierarchy. Second, we exploit client storage resources as an integral part of this extended memory hierarchy.

The broadcast is created by assigning data items to different "disks" of varying sizes and speeds, and then multiplexing the disks on the broadcast channel. Items stored on faster disks are broadcast more often than items on slower disks. This approach creates a memory hierarchy in which data on the fast disks are "closer" to the clients than data on slower disks. The number of disks, their sizes, and relative speeds can be adjusted, in order to more closely match the broadcast with the desired access probabilities at the clients. If the server has an indication of the client access patterns (e.g., by watching their previous activity or from a description of intended future use from each client), then hot pages (i.e., those that are more likely to be of interest to a larger part of the client community) can be brought closer while cold pages can be pushed further away.

Properties of Broadcast Programs:

In a push-based information system, the server must construct a broadcast "program" to meet the needs of the client population. In the simplest scenario, given an indication of the data items that are desired by each client listening to the broadcast, the server would simply take the union of the requests and broadcast the resulting set of data items cyclically. Such a broadcast is depicted in Figure.2.

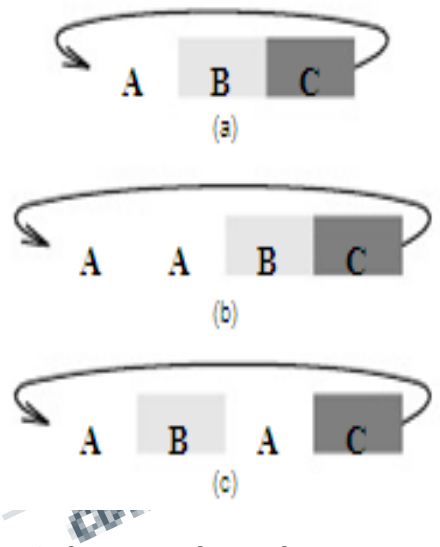


2. A Flat Broadcast Program

When an application running on a client needs a dataitem, it first attempts to retrieve that item from the localmemory or disk. If the desired item is not found, thenthe client monitors the broadcast and waits for the item toarrive.1 With the flat broadcast, the expected wait for anitem on the broadcast is the same for all items (namely, halfa broadcast period) regardless of their relative importanceto the clients. This "flat" approach has been adopted inearlier work on broadcast-based database systems such as Datacycle. Alternatively, the server can broadcast different itemswith differing frequency.Such a broadcast programcan emphasize the most popular items and de-emphasizethe less popular ones. Theoretically, the generation of such non-flat broadcast programs can be addressed as a bandwidth allocation problem; given all of the client access probabilities, the server determines the optimal percentage of the broadcast bandwidth that should be allocated to each item. The broadcast program can then be generated randomly according to those bandwidth allocations, such that the average inter-arrival time between two instances of the same item matches the needs of the client population. However, such a random broadcast will not be optimal in terms of minimizing expected delay due to the variance in the inter-arrival times.

The figure shows three different broadcast system of a

Programsfor a data set containing three equal-length items
(e.g., pages). Program (a) is a flat broadcast, while (b) and
(c) Both broadcast page A twice as often as pages B and
C. Program (b) is a skewed broadcast, in which subsequent
Broadcasts of page A are clustered together. In contrast,
Program (c) is regular; there is no variance in the inter-
Arrival time for each page. The performance characteristics
of program (c) are the same as if page A was stored on a disk
that is spinning twice as fast as the disk containing pages B
and C. Thus, we refer to program (c) as a Multi-disk broadcast.



3.Three Example Broadcast Programs

On-Demand Broadcast:

New Challenges and Scheduling Algorithms" proposed that, With the rapid growth in wireless technologies and the cost effectiveness in deploying wireless networks, wireless computers are quickly becoming the normal front-end devices for accessing enterprise data. In this paper, we are addressing the issue of efficient delivery of business-critical data in the form of summary tables to wireless clients equipped with OLAP front-end tools. Towards this, we propose a new heuristic, on-demand scheduling algorithm, called STOBS, that aggregates requests and broadcasts the results only once to all clients. STOBS exploits the structural dependencies among summary tables to maximize data sharing based on aggregation and does not assume fixed length or uniform broadcasts. The effectiveness of our proposed heuristic was evaluated experimentally using simulation with respect to both access time and Fairness, as well as power consumption in the case of mobile clients.

Conclusion:

In this paper, we have described our design of a multilevel
broadcast disk and cache management policies for this style of memory. We believe that this approach to data management is highly applicable to asymmetric network environments such as those that will naturally occur in the NII as well as many other modern data delivery systems. We have demonstrated that in designing such disks, the broadcast program and the caching policy must be considered together. It has been shown that there are cases in which the performance of both two and three level disks can outperform a flat broadcast even when there is no caching. We have argued that our scheme for interleaving the data is desirable because it provides a uniform expected latency.

Future Enhancement:

We have further shown that introducing a cache can provide an advantage by smoothing out disagreement between the broadcast and the client access patterns. The cache gives the clients a way to hoard their hottest pages regardless of how frequently they are broadcast. However, doing page replacement solely on probability of access can actually increase a client's sensitivity to the server's broadcast.

We believe that this study while interesting and useful in its own right, is just the tip of the iceberg. There are many other opportunities that can be exploited in future work. Here, we have only considered the static read-only case. How would our results have to change if we allowed the broadcast data to change from cycle to cycle? What kinds of changes would be allowed in order to keep the scheme manageable, and whatkinds of indexing would be needed to allow the client to make intelligent decisions about the cost of retrieving a data item from the broadcast.

References:

[1].S. Acharya, R. Alonso, M. Franklin, S. Zdonik, "Broadcast
Disks: Data Management for Asymmetric Communications
Environments",Tech. Report CS-94-43, Brown Univ.Tech. Report
CS-TR-3369, Univ. of Maryland, Oct. 1994.

[2]. Ping Yu, Weiwei Sun__, Yongrui Qin, Zhuoyao Zhang, and Bole Shi,"A Data Partition Based Near Optimal
Scheduling Algorithm for Wireless
Multi-channel Data Broadcast." Fudan University, China.

[3].T.Imielinski, S.Vishwanathan, B.R.Badrinath, "Data on Air: Organization and Access".IEEE Member.

[4].Jiun-Long Huang and Ming-Syan Chen, "Dependent Data Broadcasting for Unordered Queries in a Multiple Channel Mobile Environment" National Taiwan University
Taipei, Taiwan, ROC.

[5].Sohail Hameed Nitin, H. Vaidya, "Log-time Algorithms for Scheduling Single and Multiple Channel Data Broadcast",Texas A&M University.

[6].Mohamed A. Sharaf and Panos K. Chrysanthis, "On-Demand Broadcast: New Challenges and Scheduling Algorithms",Pittsburgh, PA 15260, USA.