IJCSMA
V.Mathimalar *et al*, International Journal of Computer Science and Mobile Applications,
Vol.3 Issue. 6, June- 2015, pg. 45-50 **ISSN: 2321-8363**

INFORMATION RECLAMATION PROGRESS FOR MULTI-ITEM REQUIREMENTS IN MULTI-CHANNEL TRANSMIT ENVIRONMENTS

V.Mathimalar[1], K.Kiruthika[2]
[1]Assistant Professor , [2]M.Phil Research Scholar, [1]P.G. Department of Computer Applications,
[1, 2] Shrimathi Indira Gandhi College (Women), Trichy, India – 620002.

Abstract: **With the emergence of various multimedia applications, service and devices, multimedia delivery is expected to become the major traffic of Internet which will keep increasing rapidly. In order to serve such large scale multimedia applications, more and more service providers store their video assets in the cloud and delivery streaming to their consumers cross cloud, for example, YouTube. 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 retrieves 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.**

Keyword: **content delivery system, multimedia application, data retrieval scheduling, NP-hard.**

I. INTRODUCTION

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 a push-based system, the server will broadcast a set of data items to the clients periodically according to a fixed schedule; while in a pull-based system, the clients will first send requests to the server and the server will provide timely broadcast according to the requests received. In both systems, clients have to listen to the channels, wait for the requested data items and download them when they arrive. In the literature, many index techniques have been proposed for wireless data broadcast to provide clients the broadcasting time and channel locations of data items.In this paper, we investigate the data retrieval problem in both push-based and pull-based broadcasts. When users 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 (e.g., a user may submit a query for 10 stocks 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. Therefore, different retrieving order may result in different response time and different energy consumption. Assuming the arriving time and channel locations of requested data items are already known from the index information, our work will mainly focus on the data retrieval scheduling from the client's point of view, in which multiple data items have to be downloaded from different channels.

With the increasing popularity of portable wireless computers, mechanisms to efficiently transmit information to such clients are of significant interest. The environment under consideration is asymmetric in that the information server has much more bandwidth available, as compared to the clients. It has been proposed that in such systems the server should broadcast the information periodically. A broadcast schedule determines what is broadcast by the server and when. This paper makes the simple, yet useful, observation that the problem of broadcast scheduling is closely related to the problem of fair queueing. Based on this observation, we present a log-time algorithm for scheduling broadcast, based on an existing fair queueing algorithm. This algorithm significantly improves the time-complexity over previously proposed broadcast scheduling algorithms. Also, for environments where different users may be listening to different number of broadcast channels, we present an algorithm to coordinate broadcasts over different channels. Simulation results are presented for proposed algorithms.

This paper makes three contributions:

- We observe that the problem of broadcast scheduling is closely related to packet fair queuing. While obvious in the hindsight, this observation has not been exploited before to design efficient broadcasting algorithms.
- Based on the above observation, we present a O(log M) broadcast scheduling algorithm, where M is the number of information items. Simulations show that this algorithm achieves near-optimal performance.
- In environments where different clients may listen to different number of broadcast channels (depending on how many they can afford), the schedules on different broadcast channels should be coordinated so as to minimize the access time for most clients. We extend the above algorithm to such an environment.

II. PROBLEM DEFINITION

We first describe the system model used in this paper for push-based broadcast. Suppose a server has multiple down-link channels and a set of data items are broadcasted repeatedly according to some fixed schedule. The model descriptions are as follows:
1) The channels have uniform bandwidth ba and the data items have uniform size. All channels are partitioned into distributed time slots, each of size. Thus one time slot is the smallest unit to download data. In the rest of this paper, "time t" means the time slot with sequence number t, and a triple.
2) The index information is allocated on separate channels from data items based on some index technique. Clients will retrieve all the necessary indices before downloading data, which is not related to the data retrieval scheduling.
3) The mobile clients can only access a single channel at any particular time. So a client cannot download data from two or more channels at time t for any t.

A. Problem Analysis

Although there have been many works done on data scheduling at the server side, there have been few works done on data retrieval scheduling from the client's point of view. They proposed several heuristic algorithms such as first come first served (FCFS), most requests first (MRF) and longest wait first (LWF) for downloading multiple data items from parallel channels. But they did not provide any theoretical analysis on either the data retrieval scheduling or their proposed algorithms. They also assumed the data set is partitioned over multiple channels without replications, which further restricts the applications of their works. An optimal data retrieval scheduling was proposed to minimize the response time for multi-item requests in push-based broadcast, in which a data item can appear multiple times in one cycle.

- In a push-based system, the broadcast schedule is fixed, which may result in long response time for some frequent request.
- The tuning time with regards to index/data retrieval is not counted in the energy consumption.

B. Problem Solution

In order to maximize the number of downloads before a deadline, we define a problem called largest number data retrieval (LNDR). In some situations, users will lose patience if the data retrieval takes a long time, and they may drop a request if the response time exceeds a certain threshold. The LNDR problem takes the "deadline" into consideration and therefore also describes the time-critical scenario. When all the requested data items have to be downloaded, we formulate another problem, namely minimum cost data retrieval (MCDR), with the objective of minimizing the response time and energy consumption. Response time is defined as the time interval between the moments that a request is submitted to the moment that all the requested data items are downloaded. It is obvious that shorter response time is more desirable.

- Minimizing the response time and energy consumption using heuristic algorithm.
- It is highly depend on the broadcast schedule at the server side and has little to do with the data retrieval scheduling at the client side.

Although there have been many works done on data scheduling at the server side, there have been few works done on data retrieval scheduling from the client's point of view. Juran and Hurson proposed several heuristic algorithms for downloading multiple data items from parallel channels. But they did not provide any theoretical analysis on either the data retrieval scheduling or their proposed algorithms. They also assumed the data set is partitioned over multiple channels without replications, which further restricts the applications of their works. As pointed out in popular data items should be broadcasted more frequently than unpopular ones. In our previous work , an optimal data retrieval scheduling was proposed to minimize the response time for multi-item requests in push-based broadcast, in which a data item can appear multiple times in one cycle.

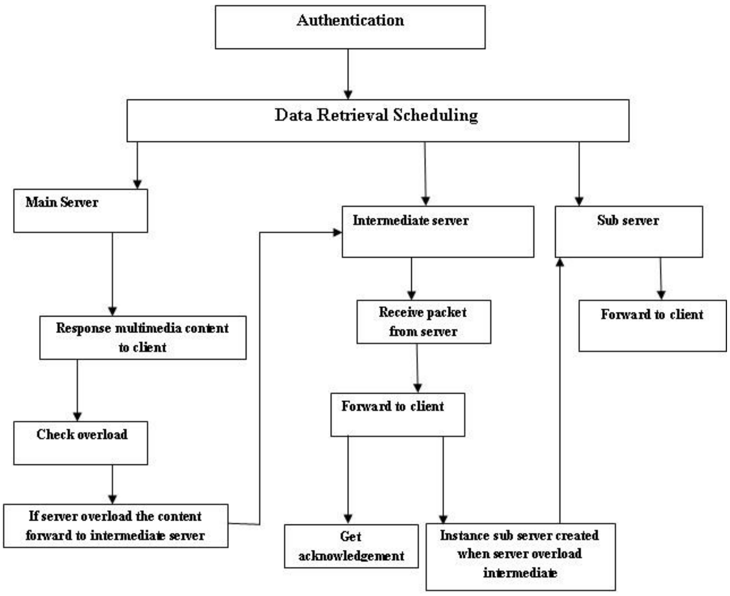

Figure 1

III.PROCESS FLOW

A. 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 inter symbol 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.

B. Data Retrieval Scheduling

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.

C. 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.

D. Minimum Cost Data Retrieval

We develop a greedy heuristic for MCDR. It combines the benefits of both channel scheduling to reduce the energy consumption 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.

IV.EXPERIMENTAL RESULTS

A. Implementation

Implementation is the most crucial stage in achieving a successful system and giving the user's confidence that the new system is workable and effective. This type of conversation is relatively easy to handle, provide there are no major changes in the system. Each program is tested individually at the time of development using the data and has verified that this program linked together in the way specified in the programs specification, the computer system and its environment is tested to the satisfaction of the user. In graph theory, an independent set or stable set for a graph G is a subset of vertices that are pair wise non-adjacent. A maximum independent set is an independent set with the maximum cardinality. As we mentioned in Section 2, 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, considering conflicts as edges and triples as vertices. Although finding a maximum independent set is NP-hard, we still can devise heuristics that provide solutions not necessarily provable, but usually efficient for practice. We next present a sequential greedy heuristic that guarantees a maximal valid retrieval schedule (i.e., a valid set of triples that is not a subset of others).

Input: an LNDR instance which is represented by a set of triples.
construct a graph G of triples and add edges between conflicted triples;
let $P \leftarrow \emptyset$ (P denotes the set of triples selected);
while G is not empty **do**
    select a triple in G with the minimum degree; put it in P and delete its neighbors;
end while
output P;

Algorithm 1 Sequential Greedy Heuristic

Generally, when a subset of elements needs to be selected, a greedy based algorithm will construct a solution by adding elements sequentially. Decisions on which element is to be added is based on certain rule. In SGH (Algorithm 1), each time we add a triple with the minimum degree. It can be shown that choosing a vertex and removing its neighbors repeatedly will achieve a maximal independent set. Thus, the solution resulted by SGH is maximal. Moreover, based on our observation, SGH is very efficient in practice, e.g., in Fig. 3, data item d_1 appears twice and SGH will select the one at time 5, because of its relatively low degree. As a result, data item d_2 can also be downloaded (the number below a data item indicates its vertex degree).

V.Mathimalar *et al*, International Journal of Computer Science and Mobile Applications,
Vol.3 Issue. 6, June- 2015, pg. 45-50 **ISSN: 2321-8363**

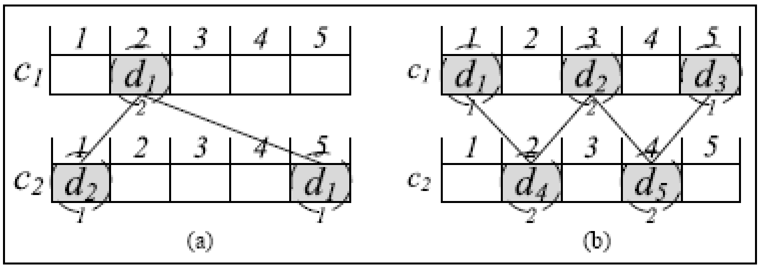

Figure 2

In Fig.3, SGH will select data items in channel c1. As a result, three data items can be downloaded. If selecting data items in channel c2, at most two data items can be downloaded. Since we convert LNDR into MIS only based on the conflicts, it is clear that SGH can be applied for non-uniform size data items and non-uniform bandwidth channels. 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 scheduling at the client side.

Running Time Performance

Finally, we would also like to evaluate the running time of the proposed algorithms: SGH and MGH so that we can validate their scalability. Since AS is a theoretical algorithm and cannot be applied when k is large, it is not considered in this experiment. The simulation is run on a PC with a 2.6 Ghz processer and 6 GB memory. We simulate a pushbased system with a large number of data items (N = 5;000, k = 500, u = 1:0, n = 8). The average running time of SGH is about 22 ms when T = 200, and that of MGH is about 240 ms. Therefore, both of them are scalable.

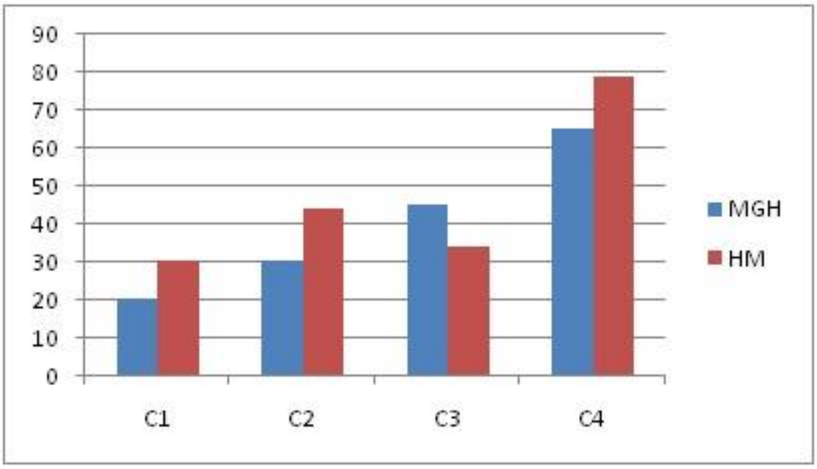

Figure 3

V.CONCLUSION

In this paper, the data retrieval scheduling for multi-item requests over multiple channels is studied. Two optimization problems, LNDR and MCDR, are defined and some approximation and heuristic algorithms are proposed. The algorithms are analyzed both theoretically and practically. Their efficiencies are also demonstrated through simulation. For LNDR in push-based broadcast, MM can download the maximum number of data items when the channels are synchronized. When the channels are unsynchronized, SGH always achieves a better solution with respect to GL, NO, MM and RS, and it scales well. AS is slightly better than SGH but it cannot be applied to download a large number of data items. For LNDR in pull-based broadcast, GL is better than NO, and other algorithms cannot be applied. For MCDR, MGH always outperforms MH, GL, NO and RS. RS is also an efficient scheduling when a large percentage of data items have to be downloaded. To the best of our knowledge, we do not find any algorithms in the literature which are designed for pull-based data scheduling at the server side over multiple unsynchronized channels. As a direction for further research, one can study the data scheduling problem for unsynchronized channels from the server's point of view.

REFERENCES

[1] J.E. Hopcroft and R.M. Karp, "An n5=2 Algorithm for Maximum Matchings in Bipartite Graphs," SIAM J. Computing, vol. 2, no. 4, pp. 225-231, 1973.
[2] H.D. Dykeman, M. Ammar, and J.W. Wong, "Scheduling Algorithms for Videotex Systems under Broadcast Delivery," Proc. IEEE Int'l Conf. Comm., pp. 1847-1851, 1986.
[3] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communication Environments," Proc. ACM SIGMOD Int'l Conf. Management of Data, pp. 199-210, 1995.
[4] N. Vaidya and S. Hameed, "Log Time Algorithms for Scheduling Single and Multiple Channel Data Broadcast," Proc. Ann. Int'l Conf. Mobile Computing and Networking, pp. 90-99, 1997.
[5] U. Feige, "A Threshold of lnn for Approximating Set Cover," J. ACM, vol. 45, no. 4, pp. 314-318, 1998.
[6] T. Imielinski, S. Viswanathan, and B.R. Badrinath, "Data on Air: Organization and Access," IEEE Trans. Knowledge and Data Eng., vol. 9, no. 3, pp. 353-372, May/June 1997.
[7] D. Aksoy and M. Franklin, "Scheduling for Large-Scale On- Demand Data Broadcasting," Proc. IEEE Int'l Conf. Computer Comm., pp. 651-659, 1998.
[8] D. Aksoy and M. Franklin, "R _W: A Scheduling Approach for Large-Scale On-Demand Data Broadcasting," IEEE/ACM Trans. Networking, vol. 7, no. 6, pp. 846-860, Dec. 1999.
[9] C. Kenyon and N. Schabanel, "The Data Broadcast Problem with Non-Uniform Transmission Time," Proc. ACM-SIAM Symp. Discrete Algorithms, pp. 547-556, 1999.
[10] C.D. Manning and H. Schutze, Foundations of Statistical Natural Language Processing. MIT Press, 1999.