Information Systems 34 (2009) 164–177

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/infosys

On optimal scheduling for time-constrained services in multi-channel data dissemination systems

Chao-Chun Chen [a], Chiang Lee [b,*], Shih-Chia Wang [b]

[a] *Department of Computer Science and Information Engineering, Southern Taiwan University, Tainan, Taiwan, ROC*
[b] *Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, ROC*

ARTICLE INFO

Article history:
Received 11 December 2006
Received in revised form
14 August 2007
Accepted 1 May 2008
Recommended by: K.A. Ross

Keywords:
Mobile computing
Data dissemination
Wireless communication
Time-constrained service
Multi-channel broadcast
Schedule
Transformation
Algorithm analysis

ABSTRACT

We study the problem of disseminating data of time-constrained services through multiple broadcast channels. By time-constrained services, we mean those services whose data must reach clients before a certain constrained time. Otherwise, the data would become useless or substantially less valuable to the clients. We first explore the difficulties of solving the problem and derive the theoretical minimum number of channels required for the task. Then, we propose a *transformation-based data allocation* (TDA) algorithm that guarantees to fulfill the task (i.e., all requested data reach the clients within the constrained time) by using the minimum number of channels. Finally, we analyze the computation complexity and prove the validity and optimality of the TDA algorithm.

1. Introduction

The advantages of perfect scalability and high utilization of precious communication bandwidth make *data broadcasting* an attractive solution for wireless communication applications [1–5]. Clients in such a system listen to the broadcast, wait until the required information comes, and then download the data. The broadcast information is sifted from a data bank, by some mechanism, to ensure that it is needed by most clients. It is then scheduled in a particular order, so that it is disseminated in that order. This schedule is referred to as a *broadcast program*.

In the past, one of the major focuses of research on the scheduling of broadcast programs was to reduce the average access time, the amount of time it takes from when a client begins to listen (which could be any time) until the client receives the required information [2,4,6–9]. Although reducing this average access time shortens the waiting time of clients, it does not necessarily suffice to meet the individual needs of the clients. With the increasing involvement of services of mobile applications in our daily lives, the need of acquiring these broadcast data *under a certain constrained time* is becoming paramount. For example, the timing of buying/selling stocks is crucial for a stock holder. If the stock information cannot reach a stockholder in time, the information might become useless. Another useful application is that information about a traffic jam caused by a car accident should reach mobile clients in a timely manner. If a client receives such information early enough, he or she can react accordingly to avoid the traffic jam. The value of the

* Corresponding author. Tel.: +886 6 2757575; fax: +886 6 2747076.
E-mail addresses: chencc@mail.stut.edu.tw (C.-C. Chen), leec@ads.csie.ncku.edu.tw (C. Lee), ripken95@ads.csie.ncku.edu.tw (S.-C. Wang).

0306-4379/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.is.2008.05.004

information would degrade significantly, as the client gets closer to the spot of the accident.

Another scenario that may cause problems is that a client may not always be patient in listening to the broadcast information, particularly if the needed information is not urgent [10,11]. When the waiting time is longer than the *expected time* of a client (an actual *time constraint*), the client could switch the access from a broadcast channel to an on-demand channel and actively send a pull request through an uplink channel to ask for the desired data, instead of passively waiting on the broadcast channel. Too many such actions could seriously congest the on-demand channels. Therefore, the less a broadcast program meets a client's expectations, the more serious congestion is wrought on the on-demand channels [11]. If the waiting time of clients in the broadcast channels can be controlled within the constrained time, the quality of service of the on-demand channels can also remain unaffected. These practical reasons and real-time requirements motivate the need for new research on the scheduling of broadcast data so as to fulfill the needs of the clients [10,12,13].

In [14,15], we studied a simple version of this problem and presented solutions for it under the assumption that every request of a client requires information of only a single broadcast data page. In this paper, we extend the work by relaxing the assumption and allowing the data required by a request to be dispersed in multiple broadcast pages. Each request of a client is associated with a time constraint. All the required data pages of a request must be received within the client's expected time constraint. The importance of the broadcast data remains the same if a client receives them within the expected time; otherwise, their value diminishes, or they even become useless. Therefore, the broadcast program should be designed so that no matter when a client starts to listen, the program always allows the client to receive the required data pages within the constrained time. We propose scheduling mechanisms for scheduling data in broadcast channels to ensure that this goal can be achieved. Notice that we assume the time-constraint program is scheduled in the multi-channel environment. Some works on broadcast scheduling that addresses time-constraints in a single-channel environment includes [16,17].

Concerning how to derive the time constraint of a request, several previous research studies can be applied to obtaining this time constraint. The *piggyback* and the *probing* techniques are a few of them suitable for this purpose [18–20]. They have been used to learn, in general, what data are of interest to the clients and should be broadcast to the public and what data are interesting only to specific groups of clients and should wait for on-demand requests.

In the piggyback technique, an uplink channel is reserved for the clients to actively report to the broadcast server the type of data they require. In the probing technique, the server may intentionally suspend the broadcast service for a certain period of time, in order to force clients to ask for data through the on-demand mode, so that their needs are revealed. The time constraint is attainable by adjusting the time duration of suspension in the probing technique or by urging the clients to indicate their tolerable waiting time in the piggyback technique. Later, this problem was also formally studied in [10], and a mathematical foundation was built for modeling the "patience" of clients. Hu et al. [11] further enhanced the work by proposing a selective deferment and reflection (SDR) technique to exploit the natural behavior of client patience and estimate the distribution of dynamic access frequency. All these studies lay a sound foundation for measuring the time constraint of a client request.

The difficulties involved in scheduling the broadcast data arise mainly from the fact that a client may start to listen at any time. The system has to ensure that each client, no matter when he or she starts to listen, will always be able to receive all the desired data pages within the time constraint. For instance, let page x be required by a client request with a time constraint of 4 time units. Fig. 1 shows the assignment of this page to the available channel. Assume that only one channel is available. If x is allocated to the channel in the way shown in Fig. 1(a), entering time a allows the client to receive x within the time constraint, but not entering time b, with which the client has to wait for a time much longer than the time constraint until in the next round the program is repeatedly broadcast. If, however, x is allocated, as shown in Fig. 1(b), the client is always able to receive x in time, regardless of its entering time, such as a, b, c, or d. That is, the broadcast data pages have to be allocated multiple times in the broadcast program, in order to satisfy a client who may enter the broadcast system at any time. However, this is only a scenario of a single request asking for a single page in a single-channel environment. The problem becomes highly complicated when we are considering multiple requests, each requesting multiple broadcast pages that are assigned multiple times to a multiple-broadcast-channel environment.

Several other issues also complicate the task. The virtually arbitrary constrained time of the requests adds another dimension of complexity to this problem. If the available channels are abundant, we are sure that the broadcast data can be arranged within the channels to meet clients' time requirements. But does there exist a minimal number of channels to satisfy the needs of all clients under such severe restrictions (entering the system at a random time and with an arbitrary distribution of the constrained time)? Furthermore, theoretically, if there

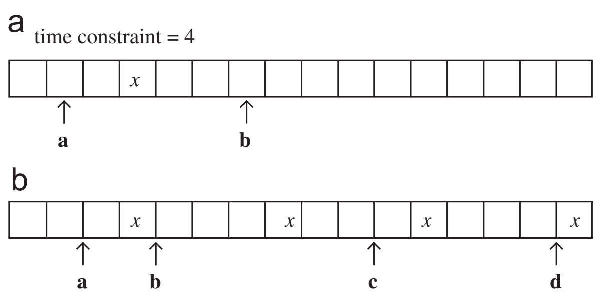

Fig. 1. A difficulty caused by an arbitrary time of entering the broadcast system.