Scheduling Real-Time Information in a Broadcast System with Non-Real-Time Information

Hsin-Wen Wei, Pei-Chi Huang, Hsung-Pin Chang[*] and Wei-Kuan Shih
Department of Computer Science National Tsing Hua University, Hsinchu, Taiwan, ROC.
[*]*Department of Computer Science National Chung Hsing University, Taichung, Taiwan, ROC.*
{bertha, peggy}@rtlab.cs.nthu.edu.tw; hpchang@nchu.cs.nthu.edu.tw; wshih@cs.nthu.edu.tw

Abstract

Data broadcast is an efficient information delivery model that can deliver information to a large population simultaneously. In this paper, we propose two efficient algorithms to broadcast real-time and non-real-time data together. The goal of our algorithms is to reduce the average response time of non-real-time data under the constraint that all real-time data must meet their deadlines. The experimental results show that our proposed algorithms can reduce the average response time while guarantee the timing constraints.

1. Introduction

Due to the high market competition of the commerce, the wireless networks grow very fast. The direct impact is the growth of wireless device, high-speed wireless network, portable computing device and the exploitation of software technology. An interesting application for the mobile computing and the wireless network is to transfer the information to mobile users efficiently.

However, the properties of the wireless network are different from that of the wired network. The differences include: (1) Asymmetric communication bandwidth: the upstream bandwidth from clients to servers is relatively small comparing to the downstream bandwidth from servers to clients. The servers need to serve a large number of clients concurrently [13]. (2) Low-power issues: portable devices are operated with batteries, thus, power saving is an important research issue. (3) The bandwidth is limited: comparing to the bandwidth of wired network, the wireless network has fairly small bandwidth. For the data broadcast scheduling algorithms in the asymmetric communication environment, it is important to decide what and when the data should be broadcast, and several scheduling algorithms have been developed [2-6] [9] [14].

If a data to be broadcast is not timing sensitive, it should be scheduled with minimum average response time. In the previous studies [8] [11], the authors investigate the problem of finding the proper broadcast schedule with minimum average response time. Their works show that the data should be transferred in equal space to get the lowest average response time for data. Moreover, the delivered data may have real-time constraints such as in the multimedia systems, the stock trading system, and the control systems.

To address the real-time requirement in a data broadcast system, the authors in [3] [10] propose scheduling algorithms based on the concept of push based broadcast disks. Their methods focus on the organization issues of broadcast disks, i.e. broadcast program or broadcast schedule. For example, in [3], the authors present the organizations of real-time fault tolerant broadcast disks and proposed the AIDA-based (Adaptive Information Dispersal Algorithm) broadcast algorithm to reduce the impact coming from the failures of the sporadic disk. In addition, [15] proposed an Earliest Deadline First (EDF) broadcasting algorithm that schedules broadcast data items according to their deadlines.

However, most previous works do not consider the case that the real-time data and the non-real-time data can be scheduled together [12]. That is, real-time data has to be transferred to users within a deadline, and non-real-time data should be sent in a way that the average response time is minimized. Thus, in this paper, we propose a new approach for this problem. Our main idea is that we try to schedule non-real-time data on some specific and fixed position initially. That is, we schedule non-real-time data in equal space. If it is not possible to do so, we will relax and schedule some selected non-real-time data by EDF. The simulation results show that our approach can achieve good performance.

The rest of this paper is organized as follows: Our system model is defined in section 2. In Section 3, we present our broadcasting algorithms in detail. Section 4 evaluates the performance of our algorithms. Section 5 is the conclusion of this paper.

2. System Model

In this paper, the data broadcast system is based on the Push-based architecture. We assume that the server has a real-time database to access/store real-time data. Also, the server provides an index channel that describes the information and broadcast sequence of real-time and non-real-time data to clients. Consequently, clients can enter the sleep mode to save energy and wakeup when their required data become available. Furthermore, transmission error is ignored and the data items are preemptive.

Assume that the broadcast system has ***n*** real-time data items and one non-real-time data item, the notations used in this paper are described as follows：

- $\boldsymbol{T}$ ：denotes the set of all real-time data items.
- T_i ：the i th real-time data item.
- P_i ：the period of data item T_i.
- d_i ：the deadline of real-time data item T_i.
- c_i ：the execution time of real-time data item T_i.
- T_{nr} ：the non-real-time data set.
- c_{nr} ：the execution time of non-real-time data.

In the previous studies [8] [11], the authors show that when the instances of data items are equally spaced, the average response time is minimized. Therefore, the goal of this paper is to schedule the non-real-time data in equally spaced manner to minimize the average response time.

3. Algorithm

Scheduling the instances of non-real-time data in equal space is not always possible, since the timing constraints of real-time data items must be met. If we insist that the non-real-time data should be equally spaced in the schedule, some real-time data items may miss their deadlines. Therefore, in this section, we propose several algorithms to schedule the non-real-time data in equal space as much as possible. In this paper, we consider two different kind of non-real-time data: (1) the simple case, c_{nr}=2 and (2) the general case.

3.1 The simple Case

If the execution time of non-real-time data is 2, we will try to schedule the non-real-time data in the middle of its period. The period of non-real-time data is defined by equation (1). Our conjecture is based on the following observation: the non-real-time data can be divided into two parts with one unit in each part. To schedule these two parts consecutively, we should put them in the middle of the period.

$$p_{nr} = c_{nr} / (1 - \sum_{i=1}^{n} \frac{c_i}{p_i}) \qquad (1)$$

Our algorithm is designed based on this conjecture. Unfortunately, fixing non-real-time data at specific position may result in the deadline missing of some real-time data. To handle this situation, we design a rollback process. If some real-time data miss their deadlines at current time, the scheduler would mark the current non-real-time data period as “EDF” and roll back to the beginning of current non-real-time data period to reschedule all unscheduled data. The pseudo code of our algorithm is listed as follows.

Centralize_NR_Data (CNR)

```
CNR (T, T_nr){
//Initialize:
1.  Calculate the period of non-real-time data.
2.  All non-real-time data periods are marked as "FIXED".
3.  Set the ready time and the deadline of non-real-time data as ceiling((i-0.5)*p_nr )-1 and ceiling((i-0.5)*p_nr ) +1 , respectively.
//Loop begin
4.  while( the final non-real-time data has not been scheduled){
5.      Schedule data by the EDF scheme.
6.      If some real-time data miss its deadline {
7.          Rollback until the first non-real-time
              data marked as "FIXED" is found.
8.          Mark this period as "EDF" and restore
              original ready time and deadline of
              non-real-time data.
9.          Rollback the scheduling point to the
              beginning of this period.
10.     }
11. }
```

Although this rollback process will continue in the worst case, with the help of a good data structure, the overhead of the rollback process is relatively small.

3.2 The general case

When the execution time of the non-real-time data is larger than 2, in the worst case, the non-real-time data will be preempted and distributed into several non-contiguous time instants. The rollback mechanism introduced in previous section is useful for this case. In our second algorithm, the non-real-time data is fixed at the end of its period initially. The ready time and the deadline of non-real-time data will be set properly to guarantee this. That is, for the *i*th non-real-time data, the ready time and the deadline of this data are set to be $i*p_{nr} - c_{nr}$ and $i*p_{nr}$, respectively. The pseudo code of our second algorithm is listed as follows.

Multiple_Length_NR_Data (MLNR)

```
MLNR (T, T_nr){
//Initialize:
1.   Calculate the period of non-real-time data.
2.   All non-real-time data periods are marked as "FIXED".
3.   Set the ready time and the deadline of non-real-time data as $i*p_{nr} - c_{nr}$ and $i*p_{nr}$ , respectively.
//Loop begin
4.   while( the final non-real-time data has not been scheduled){
5.        Schedule data by the EDF scheme.
6.        If some real-time data miss its deadline {
7.             Rollback until the first non-real-time data marked as "FIXED" is found.
8.             Mark this period as "EDF" and restore original ready time and deadline of non-real-time data.
9.             Rollback the scheduling point to the beginning of this period.
10.       }
11.  }
```

4. Simulation Results

4.1 Simulation Environment

In this section, we present some simulation results on the average response time of non-real-time data under our proposed algorithms. The average response time for transmitting a data item is calculated by [1]:

$$R_{nr} = \sum_{j=1}^{k} \frac{S_j}{H}\left(\frac{S_j}{2} + L_j\right) \quad (2)$$

The spacing between two consecutive instances of an item is denoted as S_j. In our simulation, the number of instances of non-real-time data, denoted by k in (2), is equal to H/P_{nr}. In equation (2), H represents the broadcast cycle of all data and L_j is the time that the client start receiving the required data until the transmission of non-real-time data is finished.

The system parameters used in the simulations are shown in Table 1. In the experiments, one hundred experiments are conducted with different seeds for generating random variable.

Table 1. Simulation parameters

Parameter	Description	Value
N	The number of real-time data items.	5
c_{nr}	Execution time of non-real-time data.	1 to 4
Slack	The slack time available in system after scheduling real-time data.	1% to 35%

4.2 Performance of proposed algorithms

In our simulation, we compare our results with EDF schedule and the ideal schedule. Figure 1 shows the performance of the simple case with c_{nr}=2, the average response time of our algorithm is significantly reduced, comparing to the EDF schedule. When the percentage of slack is increasing, the performance of the EDF is improved.

Figure 2 and Figure 3 show the simulation results of general case with c_{nr}=3 and c_{nr}=4, respectively. In Figure 2, our algorithm still has good performance closing to the ideal schedule. However, the growing of the execution time of non-real-time data has great impact on the average response time. Since, it is much harder to avoid the preemption made by real-time data.

Finally, Figure 4 shows the overall improvement of our algorithms in reducing average response time comparing with EDF algorithm. Our simulations show that our algorithms can reduce the average response time of non-real-time data, especially when the available slack in system is lower.

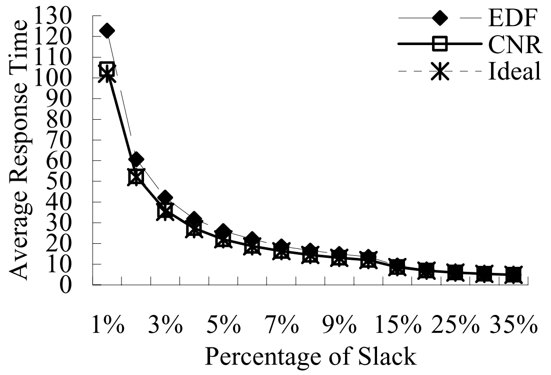

Figure 1. Average response time for simple case.

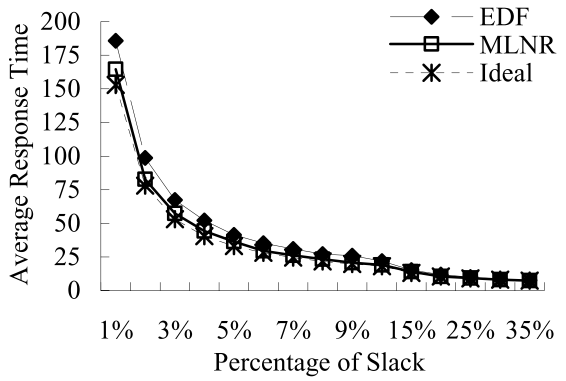

Figure 2. Average response time for general case, c_{nr}=3.

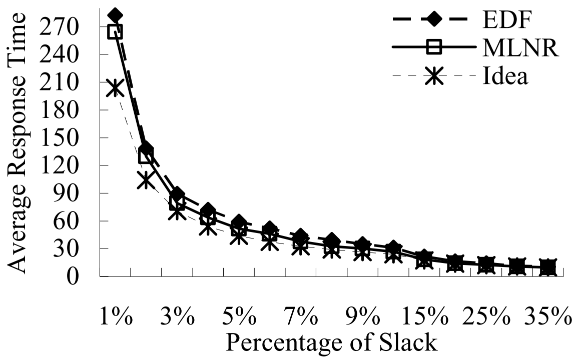

Figure 3. Average response time for general case, c_{nr}=4.

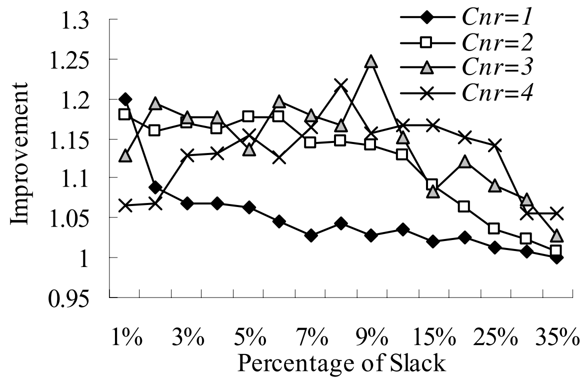

Figure 4. The improvement of our proposed algorithms with c_{nr}=1,2,3,4.

5. Conclusion

In this paper, we propose two algorithms in push-based data broadcast system, which improve the average response time of non-real-time data while guarantee the timing constraints of real-time data. The simulation results show that our proposed broadcast algorithms can reduce the average response time by comparing to EDF schedule.

Reference

[1] M.H. Ammar and J. W. Wong. "The design of teletex broadcast cycles". Performance Evaluation, Volume 5, Nov. 1985.

[2] Acharya, S., Alonso, R., Franklin, M. and Zdonik, S. "Broadcast disks: data management for asymmetric communication environments". In Proceeding of Fourth ACM SIGMOD Conference, San Jose. 1995.

[3] Azer Bestavros. "AIDA-based real-time fault-tolerant broadcast disks". In Proceedings of 1996 IEEE Real-Time Technology and Applications Symposium, May 1996, Boston, Massachusetts.

[4] Demet Aksoy and Michael Franklin. "R X W: A Scheduling Approach for Large-Scale On-Demand Data Broadcast" IEEE/ACM TRANSACTIONS ON NETWORKING, Volume 7, No. 6, December 1999.

[5] Jian-Hao Hu, K.L. Yeung, Gang Feng and K.F. Leung. "A Novel Push-and-Pull Hybrid Data Broadcast Scheme for Wireless Information Networks". In ICC 2000. 2000 IEEE International Conference on Communications, Volume: 3, 18-22 June 2000.

[6] Kalyanasundaram B., Velauthapillai M., "Scheduling broadcasts in wireless networks". In European Symposium on Algorithms. 2000.

[7] Lam, K., Chan, E., Leung, H., Au, M., "Data broadcast for time constrained read-only transactions in mobile computing systems". In Proceedings of International Workshop on Advance Issues of E-Commerce and Web-based Information Systems, Santa Clara, California. 1999.

[8] Nitin H. Vaidya and Sohail Hameed. "Scheduling data broadcast in asymmetric communication environments". In Wireless Networks May 1999 Volume 5 Issue 2.

[9] Pitoura, E., Samaras, G.," Data Management for Mobile Computing". Kluwer Academic Publishers. 1997.

[10] Sanjoy Baruah, Azer Bestavros. "Pinwheel Scheduling for Fault-tolerant Broadcast Disks in Real-time Database Systems". In Proceedings of 13th International Conference on Data Engineering, 7-11 April 1997.

[11] Sohail Hameed and Nitin H. Vaidya. "Efficient algorithms for scheduling data broadcast". In Wireless Networks May 1999 Volume 5 Issue 3.

[12] Sung-Hwa Lim, J.-H. Kim, "Real-time broadcast algorithm for mobile computing" .In the journal of systems and software, Volume 26 page173-181, 2000.

[13] Swarup Acharya, Michael Franklin and Stanley Zonik. "Dissemination-based Data Delivery using Broadcast Disks". In IEEE Personal Communications, December 1995.

[14] Swarup Acharya , Michael Franklin and Stanley Zdonik. "Balancing Push and Pull for Data Broadcast". In Proceedings of the 1997 ACM SIGMOD international conference on Management of data, Volume 26 Issue 2.

[15] Xuan, P., Gonzalez, J., Fernandez, Ramamritham, K., "Broadcast on demand: efficient and timely dissemination of data in mobile environment". In Proceeding of Third IEEE Real-Time Technology Application Symposium. 1997.