Multimedia Tools and Applications, 15, 115–126, 2001

Multimedia Information Delivery Over Wireless Channels

B. PRABHAKARAN* prabha@comp.nus.edu.sg
School of Computing, National University of Singapore, Singapore 117543

Abstract. Wireless communication channels are generally characterized by availability of low bandwidth. Hence, exchange of multimedia information is normally done over a single or unicast channel. Multimedia information presentation can have certain constraints, such as precedence of one page of information with respect to another or the time of availability of information. Multimedia information delivery needs to be scheduled based on these constraints. In this paper, we model this scheduling problem using linear programming approach. We suggest the use of approximation algorithms proposed for this purpose (L.A. Hall, A.S. Schulz, D.R. Shmoys, and J. Wein, Mathematics of Operations Research, Vol. 22, pp. 513–544, 1997). We have also presented the use of on-line approximation algorithms for scheduling multimedia information delivery when their availability varies with time. We have presented our implementation experience also.

Keywords: multimedia delivery, wireless applications, job scheduling, approximation algorithms

1. Introduction

Multimedia information typically comprises text, image, audio, and video. Information can be organized or presented in the form of pages, as shown in figure 1. Presentation of multimedia information is typically done page by page. These presentations can have certain constraints. For example, one can have precedence constraints where a particular page of information needs to be presented ahead of another one. Alternatively, some of the pages may be available only after some point of time. In some cases, availability of these pages might vary with time.

Wireless networks normally have low availability of network bandwidth. Exchange of multimedia information is, hence, done using unicast communication channels. Here, delivery of multimedia information pages is done one after another, as shown in figure 2. We need to come up with a delivery schedule that identifies when a multimedia page can be delivered over the wireless network, based on the existing constraints. We can consider the following constraints.

1. *Available time and precedence*: Multimedia pages may become available for delivery at different instants of time and may have precedence relationships among them. For instance, a page i may become available at time $t1$ and another page j at time $t2$. In some cases, a page k may have precedence over another one l, for delivery.

*Supported in part by National University of Singapore Academic Research Fund Grant RP 981669.

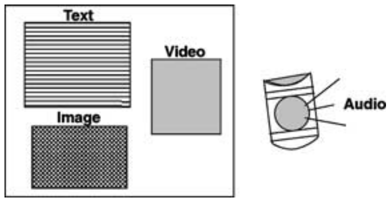

Figure 1. Multimedia pages composition.

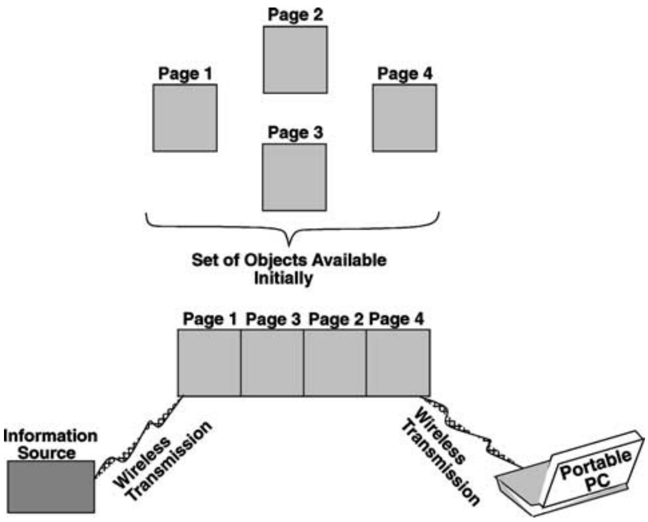

Figure 2. Delivering multimedia pages over unicast wireless channels.

2. *Pre-emption*: User going through the presentation of multimedia pages might want to go to the next page even before the delivery of the current page is complete, i.e., the delivery of a page may be pre-empted.
3. *Dynamic availability*: Multimedia pages' availability may not be known initially. For instance, a set of pages may become available after time t. This is different from the first case in the sense that in the first case, the set of pages that need to be scheduled are known initially itself. However, the page available times may be different. Whereas in the second case, the set of pages to be scheduled is not known initially.
4. *Broadcast mode considering popularity*: In a broadcast mode, where delivery is listened to by multiple wireless clients, we can consider scheduling the delivery of multimedia pages by considering popularity of different pages of information.

In this paper, we model this problem of scheduling the delivery of multimedia pages based on specified constraints, using linear programming approach. We suggest the use of approximation algorithms described in [6] for this purpose. We also outline the use of on-line approximation strategies for the case where the availability of multimedia pages vary with time. We then present our implementation experience in a simple wireless environment.

2. Related work

Issues in wireless mobile computing environments are discussed in [4]. Management of data and querying aspects of databases are presented in [7, 8]. Issues and solutions for presentation of video objects in a mobile environment is described in [10]. Our work discussed in this paper is more general in that it deals with diverse media objects and also it deals with scheduling delivery of objects over a unicast wireless channel. Support for collaborative applications in mobile environments is outlined in [3]. Here, group co-ordinator services are provided in the International Standards Organization (ISO) Open Distributed Processing (ODP) environment and collaboration aware tools are designed using these services. We do not address the issue of collaborative nature of a presentation in this paper.

Techniques for indexing broadcast data have been investigated in [5, 9]. Some of these techniques involve interleaving of index information with data. However, the paper does not address the organization of data objects based on precedence constraints or release dates. Use of repetitive broadcast as a way of augmenting the memory hierarchy of clients in an asymmetric, mobile communication environments is proposed in [1]. The technique, termed *Broadcast Disks*, helps in structuring broadcast data such that performance for non-uniformly accessed data is improved. As shown in figure 3, the broadcast disk methodology repeats delivery of certain database pages depending on their frequency of access. Multimedia object organization in a wireless channel was studied in [2]. Heuristics were proposed to map the objects in polynomial time, by modeling the objects and their relationships as a directed graph. However, their work do not consider several constraints such as varying availability of multimedia pages and the lack of initial knowledge about the pages to be scheduled. Since they consider only a broadcast mode, they do no not consider the issue of pre-emption of a page delivery.

3. Delivery schedule based on release dates

In this section, we examine the applicability of a set of techniques for scheduling the delivery of multimedia pages based on precedence constraints. These techniques are based

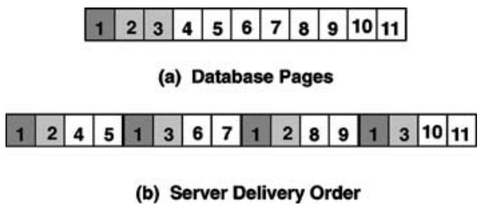

Figure 3. Broadcast scheduling of disk pages.

on a set of algorithms that have been designed to minimize average completion time of jobs based on certain constraints such as precedence and release dates [6]. These techniques are based on linear programming approach and help in scheduling jobs on single as well as parallel machines. The idea in using linear programming is as follows. There are n jobs that need to be scheduled. Each job j has a weight w_j and a non-negative processing time p_j, $j = 1, \ldots, n$. It is assumed that jobs will be processed without interruption, and a machine can process at most one job at a time. There might be precedence constraints among the jobs that need to be scheduled, i.e., there might be a partial order $j \prec k$, among any two jobs j and k. Each job j can also have a specified *release date*, r_j. This release date specifies the time at which a job becomes available. Assume C_j to denote the completion time for a job, with C_j dependent both on release date and processing time ($C_j = r_j + p_j$). Linear programming algorithm for scheduling the jobs with the above constraints try to minimize $\sum_j w_j C_j$, or equivalently $(\sum_j w_j C_j)/n$, i.e., the average completion cost for a job).

Based on this approach, we model delivery of multimedia pages as follows. Each multimedia page, as discussed earlier, is composed of various media objects such as text, image, audio, and video. One multimedia page is considered as a job that needs to be scheduled for completion. Processing time p_j for a multimedia page denotes the transfer time that might be needed for delivering all the objects composing the page. Processing time will be related to the sizes of the multimedia objects that compose a page and the network bandwidth. The weight w_j of a multimedia page denotes the importance of the page. In addition, pages j and k might be constrained by precedence relations, $j \prec k$. Now, a simple schedule for delivery multimedia pages sequentially can be derived using Algorithm A1 below. This algorithm schedules multimedia page delivery based on both the importance and the transfer time for the pages, however without considering the precedence relationships among the pages.

Algorithm A1: Simple Schedule

1. Find the ratio of processing time to the weight of a multimedia page (p_j / w_j).
2. Sequence the jobs in the non-decreasing order of p_j / w_j ratio.

This algorithm is proved to be optimal in [11]. However, it does not take into consideration other constraints such as precedence and release dates. Optimization of the average job completion time is a *NP*-hard problem when these (precedence, release) constraints are considered [6]. Hence, approximation algorithms can be used to generate near-optimal schedules.

4. Constrained schedule

In this section, we consider the following additional constraints to the simple schedule described in Algorithm A1:

1. Precedence relationships among multimedia pages to be delivered. In some instances, all the pages for delivery may not be available at the same time. Different pages may

become available at different time instants. Hence, a release date constraint (i.e., the time at which multimedia pages become available) can be added as a constraint for deriving a delivery schedule.
2. When a reverse communication channel is available for feedback, user can request for a subsequent page even before the delivery of the current page is completed. This scenario can be viewed as delivery with pre-emption, and hence constraints can be added to facilitate pre-emptions.
3. Availability of multimedia pages can be dynamic with respect to time. For instance, a new set of pages may become available after a time interval. Here, the schedule for delivery has to be determined in real-time (or, in other words, *on-line*).

As observed earlier, all the above three scenarios (i.e., delivery schedule with the specified constraints) are proved to be *NP*-hard, and hence we apply the approximation algorithms in [6] for deriving delivery schedules with the added constraints.

4.1. Delivery with precedence constraints

Here, we consider the problem of scheduling the delivery of multimedia pages taking into consideration precedence constraints. We consider an approximation algorithm proposed in [6] for scheduling multimedia pages with precedence constraints. Precedence relationship $j \prec k$ denotes that multimedia page j needs to be delivered (or presented) before page k. As an example, consider the presentation of multimedia pages shown in figure 4. Here, page 1 needs to be presented before pages 2 and 3. Similarly, pages 2 and 3 are to be presented before page 4. Based on these precedence constraints, figure 4 shows a possible schedule for delivery. In this schedule, it is assumed that w_3 is more than w_2, and hence page 3 is scheduled ahead of page 2. We also consider the case where multimedia pages become

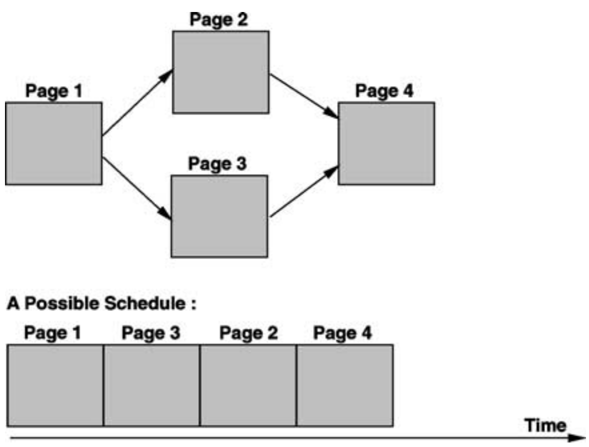

Figure 4. Scheduling of multimedia pages delivery with precedence constraints.

available for delivery at different time instants. This factor is incorporated into the schedule as release date (r_j) for a particular multimedia page j.

For this problem of deriving delivery schedule with release dates and precedence relationships among different multimedia pages, the linear programming approach is again to minimize the average completion time:

$$\text{minimize} \quad \sum_{j=1}^{N} w_j C_j. \tag{1}$$

This minimization is done subject to the precedence constraints among multimedia pages, expressed as below.

$$C_k \geq C_j + p_k, \quad \text{for each pair } j, k \text{ such that } j \prec k, \quad \text{where} \quad C_j \geq r_j + p_j \tag{2}$$

$$C_k \geq C_j + p_k \quad \text{or} \quad C_j \geq C_k + p_j, \quad \text{for each pair} \quad j,\ k. \tag{3}$$

Equation (2) describes the completion of delivery of pages j and k, when they have a precedence relationship among them. Equation (3) describes the delivery completion time when there is no such precedence relationship. It should be observed here that the delivery completion time of a page depends both on the release date r_j (i.e., the time at which page j is available for delivery) and the processing time p_j (i.e., the processing or data transfer time). The last constraint (3) is *disjunctive* in nature, and so it is not possible to model using linear inequalities. The last constraint can be rewritten as inequalities as follows [6].

We can consider the weight or the importance of the page to be the same as its data transfer time, as a simple estimate. Hence, if we set $w_j = p_j$ for all jobs j, then the sum $\sum_j w_j C_j = \sum_j p_j C_j$. For any schedule, the following is a valid constraint

$$\sum_{j=1}^{n} p_j C_j \geq \sum_{j=1}^{n} p_j \left(\sum_{k=1}^{j} p_k \right) = \sum_{j=1}^{n} \sum_{k=1}^{j} p_k p_j = \frac{(p^2(N) + p(N)^2)}{2}, \tag{4}$$

where N denotes the entire set of jobs $1, \ldots, n$. Here, $p(N)^2$ denotes the square of the data transfer time whereas $p^2(N)$ denotes the set multiplication of the data transfer times (to take care of the two summations with different indices, j from 1 to n and k from 1 to j). For each subset $S \subseteq N$, we can consider completion times C_j, $j \in S$. We can apply the previous inequality (4) to each subset and derive the following valid inequality:

$$\sum_{j \in S} p_j C_j \geq \frac{(p^2(S) + p(S)^2)}{2}, \quad \text{for each } S \subseteq N. \tag{5}$$

Algorithm A3.1 summarizes the steps involved in determining a delivery schedule for multimedia pages with release dates and precedence constraints. The minimization step now involves constraints described in (2) and (5). Precedence relationship among multimedia pages is taken care of by (2), while (5) takes care of the case where there is no such precedence relationship.

Algorithm A3.1: Constrained Schedule

1. Minimize $\sum_{j=1}^{n} w_j C_j$ subject to constraints (2) and (5).
2. Find optimal solution to the above linear program and obtain $C_1, \ldots, C_n$.
3. Schedule the jobs in order of non-decreasing C_j. (In case of ties, precedence relations are used to sequence the jobs).

4.2. *Delivery with pre-emption*

Presentation of multimedia pages can be pre-empted by a client. As an example, a client may after viewing one of the objects composing a page may pre-empt the presentation by requesting the delivery of subsequent page. Here, the transfer time of the pre-empted page, p_j (i.e., the job processing time), is reduced based on the time at which pre-emption is made. Even when a page is pre-empted, there is a minimum waiting time for a page to become available (i.e., its release date). This factor is represented by a parameter $r_{\min}$. Now, the constraint to be added for handling scheduling with pre-emption is:

$$\sum_{j \in S} p_j C_j \geq l(S), \text{ for each } S \subseteq N, \text{ where, } l(S) = r_{\min}(S) p(S) + \frac{p^2(S) + p(S)^2}{2}. \quad (6)$$

This constraint specifies that the completion times of a set of jobs, with pre-emption being allowed, depend on their minimum release dates apart from the inequality derived in (5). Algorithm A3.2 summarizes the pre-emptable delivery schedule derivation for multimedia pages with release dates and precedence constraints.

Algorithm A3.2: Constrained Schedule With Pre-emption

1. Minimize $\sum_{j=1}^{n} w_j C_j$ subject to constraints (2), (5), and (6).
2. Find optimal solution to the above linear program and obtain $C_1, \ldots, C_n$.
3. Schedule a job j at time t such that t is the maximum of the following two: (a) all of j's predecessors have been scheduled, (b) time r_j, the release time for job j.
4. In case, a multimedia page's presentation is pre-empted by user. The next page is scheduled for delivery.

4.3. *Constrained on-line schedule*

Here, we consider the problem of scheduling multimedia pages delivery in the following scenario, as in figure 5. A set of pages are available at time 0 for scheduling. These pages are to be scheduled for delivery to the user till a time D. At time D, the pages delivery schedule is re-constructed based on the available pages, i.e., new pages might become available in the time interval [0, D]. This process is repeated till all pages are scheduled.

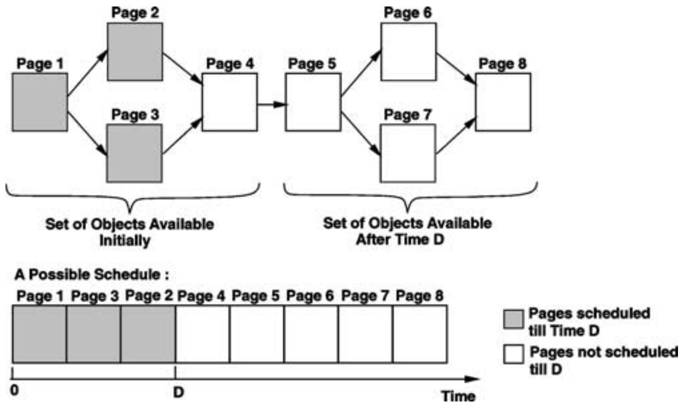

Figure 5. Online scheduling of multimedia pages delivery with precedence constraints.

Hence, apart from release dates, precedence relations, and pre-emptions, we also consider the dynamic availability of multimedia pages with respect to time. Dynamic availability or on-line scheduling is different from the first case considering the release dates alone in Section 4.1. The reason is that in the on-line case, the set of pages that need to be delivered are not known initially. Where as in the first case involving release dates only, the set of pages that need to be delivered is known initially, only the available times are at different instants of time.

We use a *greedy-interval* scheduling algorithm proposed in [6] for this purpose. This algorithm, summarized in A3, tries to schedule all the jobs that are available initially subject to constraints described in (2), (5), and (6). The algorithm is greedy in the sense that it tries to schedule the delivery of all the pages that are available initially. Delivery of multimedia pages is then carried out till time D. At time D, if new multimedia pages have become available, then the algorithm considers this new set of pages as well as undelivered pages, and derives a new delivery schedule. It should be observed here that all the pages whose schedule were derived initially might not have been delivered at D. Algorithm A3.3 works out a dynamic, on-line schedule from time to time, considering existing undelivered pages as well as the new ones that have become available. The value of D depends on the arrival rate of new pages.

Algorithm A3.3: Constrained On-line Schedule

1. Consider the set of pages available at time τ_l. (Initially, $\tau_l = 0$).
2. Minimize $\sum_{j=1}^{n} w_j C_j$ subject to constraints (2), (5) and (6).
3. Find optimal solution to the above linear program and obtain $C_1, \ldots, C_n$.
4. Schedule the jobs in order of non-decreasing C_j till time D. (In case of ties, precedence relations are used to sequence the jobs).
5. At time D, consider the new set of jobs that are available for delivering. Repeat steps 1 through 5 after incrementing τ_l by D, till all pages are scheduled.

4.4. Broadcast schedule

So far, we were considering scheduling the delivery of multimedia pages to one wireless client based on release dates, precedence constraints, and dynamic availability. We also consider the case where a client can pre-empt the delivery of a page, when a reverse channel is available for feedback. Now, we consider the scenario where multiple wireless clients listen to the delivery of multimedia pages. In this case, pre-emption is not possible (since allowing pre-emption for multiple clients is not possible). So, clients are passive viewers of the delivered information. We explore techniques for structuring the delivery schedule to multiple clients based on the probability of page access. As an example, some of the multimedia pages may be popular and hence, performance might be improved by repeating the delivery of *popular* pages. This is especially interesting in a multiple clients environment, since all clients may not tune into the delivery at the same time. Hence, when popular pages are repeated during scheduling, waiting time for these pages can be reduced.

The main step is to consider the page to be repeated (due to its popularity) as a new page (i.e., with a new page identifier) and determine its release date based on the following factors:

- Previous delivery time of the same page
- Popularity of the page

We can consider this as an on-line scheduling case with the modification that the new set of objects are the repeated ones due to their popularity. Algorithms A3.4 summarizes the modifications to A3.3 for handling broadcast delivery involving popularity of multimedia pages. The main difference is in step 5 that identifies the pages to be repeated and their release dates.

Algorithm A3.4: Broadcast Schedule

1. Consider the set of pages available at time τ_l. (Initially, $\tau_l = 0$).
2. Minimize $\sum_{j=1}^{n} w_j C_j$ subject to constraints (2) and (5).
3. Find optimal solution to the above linear program and obtain $C_1, \ldots, C_n$.
4. Schedule the jobs in order of non-decreasing C_j till time D. (In case of ties, precedence relations are used to sequence the jobs).
5. At time D, consider the popularity of the pages and identify the pages to be repeated. The release dates of these pages depend on their previous delivery time instant.
6. Now, repeat steps 1 through 5 after incrementing τ_l by D. (In a broadcast mode, the delivery might start all over again, in a cyclic fashion).

---

5. Implementation experience

We implemented some of the techniques discussed above in an implementation environment as shown in figure 6. Multimedia pages are stored in an Sun Ultra Sparc server that operates

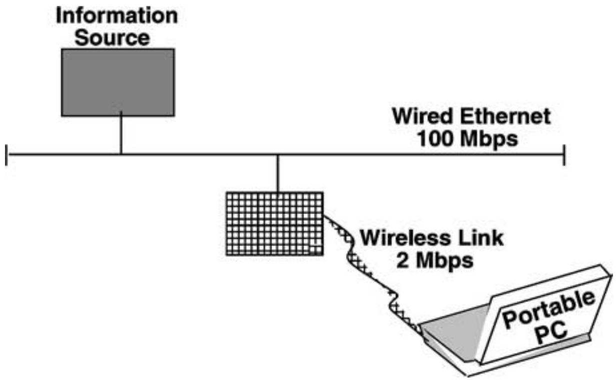

Figure 6. Implementation environment.

over a wired Ethernet at 100 Mbps. These pages, totalling around thirty in number, carries information about wild-life and the corresponding audio presents the sound made by each wild animal. Delivery of information was carried over a wireless network through a WavePoint wireless bridge operating at 2 Mbps. Client was a Pentium Laptop with WaveLan PCMCIA wireless network card. Each multimedia page is composed of objects such as text, image, and audio. We implemented Algorithm A2.1, i.e., constrained schedule of multimedia pages with pre-emption. Precedence constraints and weights for each page were assigned in an arbitrary manner among the thirty pages. We assigned about thirty constraints for the pages presentation. After identifying the delivery schedule, multimedia pages were delivered one after another over the wireless bridge to the laptop. Objects composing in one page (i.e., text, image, and audio) assigned three types of priorities: low, medium, and high. Presentation of a multimedia page can be pre-empted by clicking on a *link* object. Link object is one of the object (text or image) in a page. Links were assigned either based on object priorities or in a random manner. Time taken for identifying the delivery schedule was very negligible. Average sizes of text, image, and audio used in the multimedia pages were: 100 KBytes, 1 MByte, and 64 KBytes. We measured the time taken for presentation with and without pre-emption. Without pre-emption, delivery of all the thirty pages were completed in seconds. For presentation with pre-emption, link objects were assigned based on priorities as well as randomly. Following Table 1 summarizes the presentation

Table 1. Presentation times with pre-emption.

Links to object type	Average page time (ms)	Entire presentation (ms)
High priority	26	800
Medium priority	74	2200
Low priority	126	3800
Randomly assigned	90	3200

times with pre-emption. Presentation of each page was pre-empted by clicking on a link object.

6. Summary

In this paper, we consider the problem of scheduling the delivery of multimedia pages over wireless networks. Here, these multimedia pages might have different constraints such as:

- Precedence relationships among pages.
- Their time of availability.
- Pre-emption of a multimedia page delivery.
- Lack of complete knowledge about the availability of information to be delivered.
- Delivering to multiple clients keeping in mind the popularity of different pages.

We have modeled this problem using linear programming approach. We have suggested the use of approximation algorithms suggested in [6] for scheduling the delivery of multimedia pages based on the existing constraints (such as precedence relationships). We have also outlined the use of on-line approximation algorithms for the case where the initial availability of multimedia pages is not known and varies with time. We carried out an implementation over a wireless bridge connected to a fast Ethernet.

References

1. S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, "Broadcast disks: Data management for asymmetric communication environments," ACM SIGMOD'95, San Jose, CA, USA, 1995, pp. 199–210.
2. Y.C. Chehadeh, A.R. Hurson, and M. Kavehrad, "Object organization on a single broadcast channel in the mobile computing environment," Multimedia Tools and Applications, pp. 69–94, July 1999.
3. N. Davies, G.S. Blair, K. Cheverst, and A. Friday, "Supporting collaborative applications in a heterogeneous mobile environment," Special Issue of Computer Communications on Mobile Computing, 1995.
4. D. Duchamp, "Issues in wireless mobile computing," in Proc. 3rd Workshop on Workstation Operating Systems (WWOS), Key Biscayne, Florida, USA, IEEE Computer Society Press; Los Alamitos, CA, 1992.
5. A. Fiat, R. Karp, M. Luby, L. McGeoch, D. Sleator, and N. Young, "Competitive paging algorithms," Journal of Algorithms, Vol. 12, pp. 685–699, 1991.
6. L.A. Hall, A.S. Schulz, D.R. Shmoys, and J. Wein, "Scheduling to minimize average completion time: Off-line and on-line approximation algorithms," Mathematics of Operations Research, Vol. 22, pp. 513–544, 1997.
7. T. Imielinski and B.R. Badrinath, "Querying in highly mobile distributed environment," in Proc. of 18th VLDB Conference, Vancouver, Canada, 1992.
8. T. Imielinski and B.R. Badrinath, "Data management for mobile computing," SIGMOD Record, Vol. 22, No. 1, pp. 34–39, 1993.
9. T. Imielinski, S. Viswanatha, and B.R. Badrinath, "Energy efficient indexing on air," in Proc. ACM SIGMOD'94, May 1994.
10. B. Mah, S. Seshan, K. Keeton, R. Katz, and D. Ferrari, "Providing network video services to mobile clients," in Proc. 4th Workshop on Workstation Operating Systems (WWOS), Napa, USA, Oct. 1993, pp. 48–54.
11. W. Smith, "Various optimizers for single-stage production problem," Naval Research Logist. Quart., Vol. 3, pp. 59–66.

B. Prabhakaran is currently with the faculty of Computer Science Department, National University of Singapore. Before that, he was a visiting research faculty with the Department of Computer Science, University of Maryland, College Park. He also served as a faculty in the Department of Computer Science, Indian Institute of Technology, Madras, India. B. Prabhakaran has been working in the area of multimedia systems: multimedia databases, authoring & presentation, resource management, and scalable web-based multimedia presentation servers. He has published several research papers in prestigious conferences and journals in this area.

He has been serving as an Associate Chair of the ACM Multimedia'2000 (to be held in November 2000, Los Angeles) and ACM Multimedia'99 conference (Florida, November 1999). He has served as guest-editor (special issue on Multimedia Authoring and Presentation) for ACM Multimedia Systems journal. He is also serving on the editorial board of Multimedia Tools and Applications journal, Kluwer Academic Publishers. He has also served as program committee member on several multimedia conferences and workshops. He has presented tutorials in ACM Multimedia and other multimedia conferences. He has filed for three patents in the field of multimedia Internet servers and multimedia authoring.