Adaptive Information Dissemination: An Extended Wireless Data Broadcasting Scheme with Loan-Based Feedback Control

Chih-Lin Hu and Ming-Syan Chen, *Senior Member*, *IEEE*

Abstract—The dissemination of numerous information broadcast services gives rise to the scalability issue in wireless networks. Previous researchers have shown that the push-based data broadcast mechanism is efficient in reducing message traffic. However, most research efforts are dedicated to the dissemination of static information contents. In practice, information broadcast services can produce and deliver dynamic information contents. To efficiently convey the dynamic data, we devise, in this paper, an adaptive information dissemination mechanism by exploiting the functionality of data broadcasting, to support the dissemination of static and dynamic information services simultaneously. In our design, both static and dynamic information services are subsumed as *service groups*, i.e., the building blocks with the uniform representation of structure and group popularity and, thus, the conventional scenario becomes a special case of our framework. Furthermore, in order to tolerate the broadcast traffic dynamics, we design an online loan-based slot allocation and feedback control technique to deal with the adaptation of the service group classification, bandwidth allocation, and broadcast schedule so as to avoid performance degradation. It is shown by the experimental study that the proposed adaptive information dissemination mechanism associated with the online loan-based feedback control is able to achieve a substantial reduction of message traffic for dynamic information dissemination in wireless networks.

Index Terms—Push, adaptation, feedback control, data broadcast, information dissemination, wireless network.

◆

1 INTRODUCTION

In the aspects of bandwidth capacity and information flow, the *asymmetric communication* essentially encounters many challenges in the developments of information broadcast applications and services. Particularly, the capacity of downward bandwidth is larger than the opposite one, for example, in satellite networks, radio networks, wireless links in WLAN, and mobile cellular networks. In addition, information flow can be asymmetric in several examples, such as electronic auction and tender, instant messaging, personalized news distribution, and Web surfing systems, to name a few. Thus, with the rapid proliferation of information broadcast services and mobile recipients in wireless networks, the traditional pull-based/client-server delivery model suffers from the scalability problem and performance degradation. Substantial researchers have shown that the push-based/broadcast delivery model is a viable solution to resolve the scalability issue, especially in asymmetric communication environments. In the paradigm of push-based data broadcasting, data are *pushed* in a round-robin manner over a shared broadcast medium and accessed by the clients passively without *pull* requests [10], [23], [24], [25], [40]. More supplements to the data broadcast model will be reviewed in Section 2.

Notice that, although many previous research works have explored the data broadcast methodologies, most of them are manifested in the traditional data management systems, where a data item, specifically mapped to a pair of state and value in the database, is *persistent* and *static*. However, many modern application domains encounter different scenarios where data contents are produced continually and dynamically. In these scenarios, dynamic data streams need to be online processed rather than being stored and later retrieved to answer queries. Indeed, the data broadcast traffic has the nature of dynamic changes. As a result, most prior studies in the data broadcast community have to resort to either the assumption that the traffic is static, or that the prior knowledge of traffic patterns is available. Hence, these methods are mainly designed for the static optimization of data broadcasting, but not to address the efficient dissemination of *dynamic* data and information contents.

In this paper, we examine the issue of adaptively disseminating dynamic data contents on broadcast channels. We consider two important phenomena: 1) the data broadcast contents are produced continually and dynamically. 2) The data broadcast traffic has the nature of dynamic changes. Accordingly, we have designed a novel *group-based information dissemination* (GID) mechanism with the *loan-based slot allocation and feedback control* (LSAFC) technique by exploiting the functionality of data broadcasting, to cope with the impacts of information and traffic dynamics. For brevity, the integration of the GID mechanism and the LSAFC technique is abbreviated as GID+LSAFC. The GID mechanism subsumes static and dynamic information services as service groups, i.e., the

---

- *The authors are with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China, E-mail: mschen@cc.ee.ntu.edu.tw and clhu@arbor.ee.ntu.edu.tw.*

Manuscript received 23 Sept. 2002; revised 16 Apr. 2003; accepted 10 July 2003.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number 20-092002.

Published by the IEEE CS, CASS, ComSoc, IES, & SPS

building blocks, representing access commonality. Further, the online LSAFC technique deals with the adaptation of group classification, broadcast contents, and bandwidth allocation to tolerate dynamic traffic changes. Note that the GID+LSAFC mechanism is under the paradigm of a clients-providers-servers system [9], [15]: particularly, items are produced dynamically by the information providers, disseminated efficiently by the broadcasting servers, and accessed in the client sides. In addition, both of the client-oriented and the server-oriented traffic factors are considered. The former includes the request workload, skew access pattern, client population, etc., and the latter indicates the *dynamic item production rate*, specifically a variable number of dynamic items from an information provider within a cycle time. Note that a dynamic item corresponds to an item produced dynamically, rather than an item whose length varies in time. The GID+LSAFC mechanism analyzes the impact of information and traffic dynamics, while its performance is evaluated online. These features distinguish our work from prior ones, which consider static data (and certain traffic factors) in the traditional data broadcast model [10], [38].

The GID+LSAFC mechanism is described as follows: The GID mechanism aims to mitigate the heavy traffic penalty in response to dynamic information dissemination. The access commonality of an information service is conceptualized as a group with the uniform structure. Two group types, an *actual* group and a *virtual* group, are used to represent access commonality. An actual group includes a number of clients who retrieve dynamic data items from the same information broadcast service. In contrast, a virtual group is defined to backward support the original data broadcast scenario where clients have an interest in a certain item. In other words, a virtual service always generates the same item accessed by a virtual group as a special case of the GID model. Each group is allocated with a slot quota to deliver dynamic data items. For example, it is desirable to assign some slots to a news broadcast service for distributing updated news at any time. In contrast, the manner of predetermining the broadcast contents and schedules by the classical broadcast scheduling schemes is not suitable in dynamic environments.

When data items are generated dynamically and continually, in order to attain the system robustness, the broadcasting server needs a runtime technique to monitor dynamic traffic changes and to react to them adaptively. Note that the design of the LSAFC technique provides the GID mechanism the ability to adjust the broadcast contents on the broadcast channels. Explicitly, during a broadcast cycle, the LSAFC technique is able to analyze the temporary traffic patterns and manipulate the slot quota dynamically. When a group has not enough slots in response to additional items, a group can loan slots dynamically from other groups with the specific loaning policy. In the end of the broadcast cycle, the broadcasting server will have the feedback from the loan information. Accordingly, the GID mechanism unifies the quantitative measure of message traffic generated by different groups and, therefore, controls the adaptation of group classification, bandwidth allocation, and broadcast schedule periodically. Herein, the above procedure can be better understood by an illustrative example below.

Example 1. The design of the GID+LSAFC mechanism is an adaptive, hybrid data dissemination framework in dynamic environments. Suppose there are five clients who receive dynamic data from four information broadcast services $\{I_1, I_2, I_3, I_4\}$ in a service area. Given 10 broadcast slots, according to the traffic patterns in the last broadcast cycle, the broadcasting server initially allocates slot quotas $\{s_1, s_2, s_3, s_4\} = \{1, 2, 7, 0\}$ to the information services in the beginning of the broadcast cycle. Thus, the broadcast program contains $\{I_1, I_2, I_3\}$. As $S_4 = 0$, dynamic data from I_4 will be delivered in the pull manner. Suppose that I_1 is accessed by three, I_2 by two, I_3 by two, and I_4 by four clients; $I_1, I_2, I_3,$ and I_4 dynamically generate/update information with variable rates $\{1^*, 5, 4, 1\}$ within the time interval of the current cycle. "1^*" indicates that I_1 delivers a single/static item corresponding to the traditional data broadcast case. In this context, I_1 is regarded as a virtual group, but the others are actual groups. During the broadcast cycle, I_2 loans three slots from I_3 to broadcast excess items by using the loan-based slot allocation. In the end of this cycle, the server recalculates the respective traffic loads of service groups as $\{3, 10, 8, 4\}$ by multiplying the number of clients and the number of dynamic items in a specific group. Accordingly, the broadcast program in the next cycle will contain $\{I_2, I_3, I_4\}$ with slot quotas $\{s_1, s_2, s_3, s_4\} = \{0, 5, 4, 1\}$. Therefore, the GID+LSAFC mechanism has a total traffic of 14 messages (10 messages for broadcasting $\{I_1, I_2, I_3\}$ and four messages for delivering I_4) as compared to 25 messages by using the exclusive pull mode.

In the data broadcast arena, most research works did not consider disseminating dynamic data contents with respect to information and traffic dynamics. In contrast, the study of this paper involves both client-oriented and server-oriented traffic dynamics in the clients-servers-providers environments. Accordingly, we have designed the GID+LSAFC mechanism, in essence, an adaptive information dissemination framework, which not only preserves the original functionality of data broadcasting, but also offers the dissemination of dynamic information services. Extensive simulations have been conducted to investigate the scalability and robustness of the GID+LSAFC mechanism. Also, we have derived new server-centric performance metrics, which provide much insight into the problem of dynamic information dissemination, instead of the client-centric metrics used in the traditional data broadcast model where static data are concerned.

The rest of the paper is organized as follows: Section 2 gives the problem description and related work. Section 3 describes the design of the GID mechanism, and the LSAFC technique is presented in Section 4. Section 5 shows the performance evaluation. This paper concludes with Section 6.

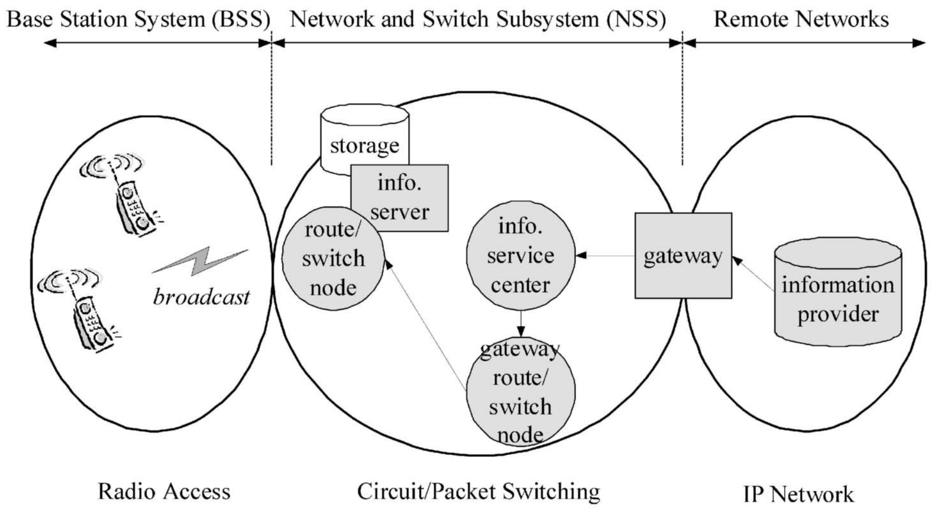


Fig. 1. A generalized mobile data access framework.

2 PRELIMINARY

This section first describes the perspective of this paper and then reviews prior works in the data broadcast model.

2.1 Perspective in Dynamic Information Dissemination

Fig. 1 illustrates a generalized mobile data access framework [1]. Due to the lack of an efficient multicast mechanism in the existing mobile data systems, a broadcast/multicast message incurs a number of message relays and transmission cost within the Network and Switch Subsystem (NSS) and the Base Station System (BSS) simultaneously [1], [26], [27], [33]. With the rapid growth of the information broadcast services and mobile recipients, reducing the enormous communication overhead is a crucial challenge in wireless networks.

In the BSSs, there exist commonly shared broadcast channels. Many prior works show that the use of data broadcasting is scalable in the traditional data management systems [2], [10], [11] where data are persistent and static. However, as mentioned before, in many modern application domains, data items on the broadcast channels are generated continually and dynamically. Since previous efforts did not explore the dissemination of dynamic data contents, they are not applicable to the dynamic environments. Comparatively, we exploit, in this paper, the potential of data broadcasting for the dissemination of dynamic data contents so as to mitigate the heavy traffic penalty. Therefore, our proposal is able to support not only the dynamic information dissemination, but also the traditional data broadcast scenarios.

2.2 Related Work

2.2.1 Data Broadcast and Periodicity

The basic *data broadcast* model assumes that an information server maintains a broadcast database and applies a periodic schedule to disseminate data items to clients through a shared medium [2], [23], [40]. In teletext systems, [7] proves that there exists an optimal schedule which is periodic, and also presents the lower bound of mean service time. Deductively, the greedy heuristics are empirically tested to formulate the suboptimal policy for the deterministic optimization problem [37]. However, the data broadcast problem is NP-Hard [8] as a special case of the *generalized maintenance scheduling problem*.

2.2.2 Transaction-Based Data Broadcast

The transaction-based data broadcast is first discussed in the Datacycle project [11]. The maintenance of semantic and temporal coherence is the primary issue [32]. The consistency in [11] is ensured by the expensive serialization. Shanmugasundaram et al. [35] introduces new correctness criterion. In addition, the invalidation [30] and the multiversion [31] techniques are presented to increase the concurrency of read-only transaction.

2.2.3 Broadcast Disks

The “broadcast disks” introduces a simplified data broadcast model with restrictive assumptions [2] and, thus, renders the data broadcast problem to be not NP-Hard. Data are static in this model. Each partition of static data set is viewed as a rotating disk, and the relative access popularity determines the rotation speed. The broadcast schedule is the output from these rotating disks. Note that many research efforts of scheduling [29], indexing [14], [22], [25], prefetching, and caching techniques [3], [41] are elaborated upon this context [10].

2.2.4 Broadcast Scheduling Strategy

In the traditional data management systems, whatever the length of an item may be, an item is static and its length is ”constant.” Accordingly, this proposition renders the data broadcast problem tractable. Otherwise, if the length of an item varies at any time, the classical broadcast schedule schemes are not sustainable without guarantees of performance, reliability, and robustness. Regarding the generation of a push-based broadcast program, Wong’s [40], Su et al.’s [37], and Broadcast Disks [2], [13] consider the static and uniform-length item, the item access frequency (or request rate), and the elapsed span (from the latest broadcast of the item), except Hameed and Vaidya’s [18], [39], where items

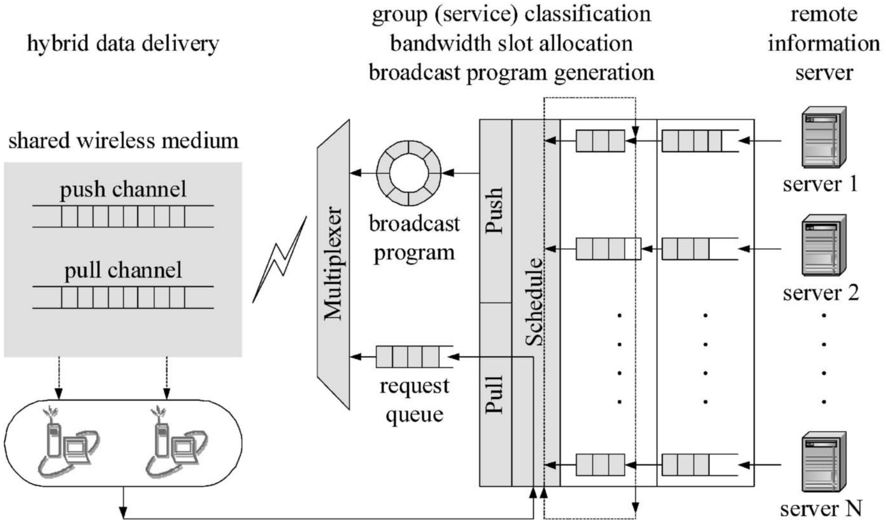


Fig. 2. The GID mechanism with an extended hybrid data broadcast model.

have the nonuniform lengths. As for scheduling the pull-based broadcast program, some basic schemes, e.g., Earliest First Request, Most Requests First, and Longest Wait First, are designed to schedule static and uniform-length items, as reviewed in [16], [40]. In [6], the $R \times W$ algorithm combines the benefits of MRF and EFR with a lower overhead. On the other hand, Shortest Service Time First (SSTF) scheme schedules static items of nonuniform lengths, but favors items of short lengths. Recently, a preemptive scheduling scheme [5], [34] is explored. Particularly, if a lengthy item is dividable, it is profitable to preempt the broadcast of a lengthy item to broadcast another item with a large number of outstanding requests.

2.2.5 Hybrid Data Broadcast Model

In general, an information server can disseminate data in either the pull or the push data delivery model. However, the push delivery mode, which broadcasts all data periodically, can result in an unacceptable access time when the number of data items in the database is huge. In contrast, the pull delivery mode responding to a client's individual request inevitably incurs a scalability bottleneck with a large number of mobile recipients. Therefore, the hybrid delivery strikes a compromise between the trade offs [4], [12], [19], [21], [36]. Explicitly, data items are classified as hot (or popular) and cold (or unpopular) items according to their access frequencies, and item slots in a broadcast cycle are partitioned into the push and the pull sets. Push slots carry hot items and cold items are delivered on pull slots by the request-response manner. With the access commonality, the hybrid data broadcast model provides an opportunity to make the access time malleable. Consequently, in this paper, we deliberate the design of dynamic information dissemination based on the hybrid data broadcast model.

3 DESIGN OF GID MECHANISM

Section 3.1 describes the system model and notation. Section 3.2 abstracts the design of the GID mechanism. The parts in the GID mechanism are presented in subsequent sections: group association, group popularity, group classification, broadcast program generation, and group schedule priority.

3.1 System Model and Notation

As illustrated in Fig. 2, the GID mechanism is an extended hybrid data broadcast model for dynamic information dissemination. Table 1 lists the notation used in the system model. A broadcasting server interacts with clients over the wireless medium. The downward bandwidth is partitioned into a series of interleaved data slots of equal sizes. Suppose that the slotted time model is employed in this model. It takes one time slot to disseminate each data item. Thus, the terms of data slot and time slot are interchangeable where there is no ambiguity. Data slots are further classified as the push or pull mode. Logically, the push slots are viewed as a push channel and those pull ones are as a pull channel.

The broadcasting server disseminates two sorts of data: static data and dynamic data. The former is mapped to the original data broadcast scenario, where the server maintains a broadcast database D and applies a broadcast program P to cyclically deliver static data items, as particularly addressed in the transaction-based and the broadcast disk models. In contrast, the latter is generated dynamically from information broadcast services, denoted as Is. Each I_i has a dynamic item production rate, denoted as γ_i, within a time interval of a broadcast cycle, denoted as L. Contrarily, data are statically kept in the broadcast database in the traditional data broadcast models.

As for the respective access commonality of dynamic and static data, we devise a uniform building block, called as a (service) group, for the structural representation in this model. Two types of groups are specified in the GID mechanism: virtual group and actual group, denoted as VG_i and AG_i, respectively. Clients, who have the same interest in a static item, virtually form a group; the server broadcasts a static item to clients in a VG_i at the same time. Note that we use the word "virtual" because there is no membership among clients in reality. Hence, a VG_i is

TABLE 1
Parameters Used in the GID+LSAFC Mechanism

notation	meaning
M	available broadcast bandwidth (M data slots)
d_i	a data item belonging to either an actual group or a virtual group
I_i	an information broadcast service which generates dynamic items
γ_i	dynamic item production rate of I_i
S_i	slot quota (S_i data slots) of a group
G_i	an actual or a virtual group
AG_i	an actual group including a number of clients with the same access interest in I_i
VG_i	a virtual group including a number of clients who access the same static item
$\|AG_i\|$	the number of clients in AG_i.
$\|VG_i\|$	the number of clients in VG_i.
GP_i	the strength of a group popularity
Λ_i	the positive difference between $\|AG_i\|$ and γ_i
$\mathcal{P}$	the flat broadcast program
L	the length of a broadcast cycle
U^H	the hot group set which includes all groups in P
U^C	the cold group set

backward compatible with the traditional data broadcast scenario. In contrast, an AG_i includes a number of clients who have subscribed the same information broadcast service I_i. In such a setting, the group classification policy is the relative group popularity, denoted as GP_i, as will be defined later. Accordingly, a group is classified into a hot/cold group; correspondingly, data items in a group are hot/cold. Without loss of generality, given a specific broadcast bandwidth (M data slots), the server maintains a *flat* broadcast program to disseminate hot group-specific items periodically. Items in cold groups are delivered by the request-response way in response to their members' pull requests from the uplink channels. Several primitives in this model are as follows:

1. Each group item is self-identified and read-only.
2. An item slot can alternatively be switched in the pull or push mode.
3. A client must at least belong to an actual/virtual group, and its access behavior should be regulated by its group membership.
4. The broadcasting server allocates a slot quota, i.e., a number of data slots, to a group if its group popularity is higher relatively.

This model is similar to those in [4], [12], [20], [36], while having a graceful extension for the design of the GID framework.

3.2 Design Abstraction

Fig. 3 illustrates the cyclic flowchart of the GID mechanism with the periodicity of data broadcasting. In the beginning of each broadcast cycle, the server initially groups the access commonality of static and dynamic data. Each group is assigned with a broadcast slot quota for disseminating the group-specific data items. During each cycle, the server analyzes the dynamic traffic patterns and the use of slot quotas of all groups. Accordingly, in the end of the broadcast cycle, the server manipulates the group association and the group popularity of each group. Moreover, the hot and the cold group sets are updated by the group classification. In light of the group schedule priority, the server executes the group replacement, i.e., a series of group demotion and promotion operations, to adjust the broadcast program for the next broadcast cycle. Due to the variety of item production rate, we devise the loan-based feedback control for the dynamic slot allocation as will be described in the next section.

3.3 Group Association

In the GID mechanism, the actual group and the virtual group are defined to represent the dynamic and the static information broadcast services with a uniform representation structure.

- *Actual Group*: In the subscription-based dynamic information scenarios, the broadcasting server groups the clients who have subscribed the same information service I_i as an AG_i. Let $\|AG_i\|$ denote the number of clients in AG_i. The value of $\|AG_i\|$ also indicates the amount of message duplications in response to an item d_i generated dynamically by I_i. Provided that I_i has a dynamic item production rate γ_i, the message traffic for AG_i is $\|AG_i\| \times \gamma_i$ within the time interval of a broadcast cycle L.
- *Virtual Group*: To backward support the original functionality of data broadcasting, each static item in the database is mapped to a virtual information service which always generates the same item. The server assumes that a client with access interest in a static item d_i joins VG_i. Thus, VG_i is a special case in the framework with $\gamma_i = 1$, and the number of clients in VG_i, denoted as $\|VG_i\|$, is the access frequency for item d_i.

It is assumed that the initial value of γ_i in AG_i and the initial access frequency of a static item in VG_i are obtained from the prior knowledge of traffic patterns. Nevertheless, in light of the LSAFC technique, as will be mentioned in Section 4, the server is able to adapt the group association responding to dynamic changes of these two factors.

3.4 Group Popularity

This section formulates the quantification of the group popularity and then presents the ways to consolidate the quantification due to dynamic traffic changes.

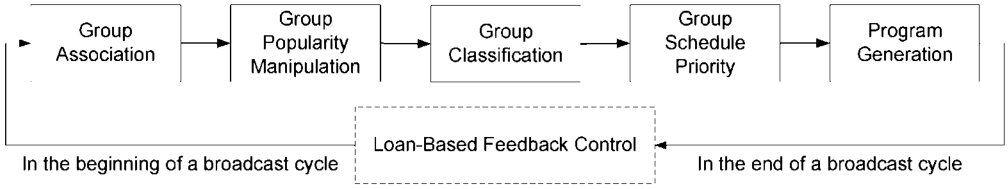


Fig. 3. The flowchart of the GID mechanism.

3.4.1 Quantification of Group Popularity

Whereas the use of data broadcasting has the significant provision of scalability, the message traffic should be minimized when the server disseminates static and dynamic data. In light of Definitions 1 and 2, the GID mechanism quantitates the group popularity, denoted as GP_i, in the period of a broadcast cycle.

Definition 1. *The* GP_i *of an* AG_i *is the message traffic by* I_i, *i.e.,* $\|AG_i\| \times \gamma_i$.

Definition 2. *The* GP_i *of a* VG_i *is the number of clients in* VG_i, *i.e.,* $\|VG_i\|$.

Note that the item access frequency is a preemptive factor in the calculation of message traffic. In the conventional data broadcast model, the access frequency of a static item is directly proportional to the number of interested clients during the time interval of a broadcast cycle, i.e., $\|VG_i\|$ in a VG_i. In contrast, the message traffic of an AG_i is the multiple of the number of clients in an AG_i and dynamic item production rate.

3.4.2 Consolidation of Group Popularity

In practice, the measure of dynamic GP_i of a VG_i is difficult. The work in [20] has proposed the Selective Deferment and Reflection (SDR) technique to estimate dynamic item access frequency by reflectively exploiting the notion of client impatience. Briefly, provided that the clients have a mean patience (ω time slots)\ in waiting for a broadcast item d_i in VG_i, the estimated mean access frequency of item d_i in L can be obtained with a reflective base d_j as

$$\overline{GP}_i = \|VG_i\| = \overline{\lambda_i} = \overline{\gamma_i} \cdot \left(L - e^{-\frac{L^2}{2\omega}}\right) \cdot \frac{\overline{\lambda_j}}{r_j}, \tag{1}$$

where $\overline{\lambda_i}$ is the estimated mean access frequency of item d_i, and $\overline{r_i}$ is the mean number of impatient requests for item d_i.

On the other hand, the consolidation of GP_i of an AG_i uses the mean value of $\|AG_i\| \times \gamma_i$. Since $\|AG_i\|$ and γ_i are various during a broadcast cycle, the server calculates the mean GP_i as

$$\overline{GP}_i = \frac{\sum_{1 \le k \le L} \|AG_i(k)\| \cdot \gamma_i}{L}, \quad G_i \in U^A, \tag{2}$$

where k indicates the kth slot within a broadcast cycle L. Note that the number of subscribed clients in an AG_i is known, transparently, by the broadcasting server on behalf of the information broadcast service I_i. In addition, the LSAFC technique performs dynamic slot allocation during each broadcast cycle and functionally provides the calculation of dynamic item production γ_i. Therefore, the server is able to consolidate GP_i of each AG_i.

3.5 Group Classification and Broadcast Program Generation

The relative group popularity is the basic policy for group classification and broadcast program generation. Without loss of generality, the higher the group popularity, the hotter a group will be. Given with M broadcast bandwidth slots, the server iteratively selects the group of the largest group popularity into the broadcast program P. Each group AG_i/VG_i in P is assigned a number of data slots, i.e., the slot quota, denoted as S_i. Note that, for an AG_i, the value of S_i is equal to dynamic item production rate γ_i as measured in the last cycle; in contrast, the slot quota of a VG_i owns one slot. Accordingly, the iterative selection will continue if and only if the aggregate of allocated slots to all groups in P is less than or equal to M. Thus, we have the length of a broadcast cycle L as

$$L = \sum\nolimits_{\forall AG_i \text{ in } P} S_i + \sum\nolimits_{\forall VG_i \text{ in } P} 1 \le M. \tag{3}$$

In this context, we define that a group is a hot group if it is scheduled in P; otherwise, that group is a cold group. As such, all groups in P form the hot group set U^H, and others form a cold group U^C. Note that the server will not continue to look for a group of lower group popularity to digest the remanent bandwidth slots, i.e., $M - L$ slots, so as to ensure that all groups in P are relatively hotter than others in U^C. In addition, the schedule positions of the group-specific items in P cannot be predetermined as a result of dynamic data contents. For that reason, the study in this paper arranges a flat broadcast program, where the scheduled position of any group-specific item is flexible and dependent on the input sequence of dynamic data, in comparison with the static schedule optimization in other previous works.

3.6 Group Schedule Priority

Notwithstanding the server schedules the broadcast program according to the relative group popularity, it is a critical situation that multiple groups have the same group popularity. For the sake of efficacy and robustness of the broadcast program, two auxiliary definitions, Definitions 3 and 4, are given for the server to determine the relative schedule priority.

Definition 3. *Given two groups* G_a *and* G_b *with* $GP_a = GP_b$, G_a *has a higher priority than* G_b *if* G_a *is a virtual group and* G_b *is an actual group.*

Definition 4. *Given two actual groups* AG_a *and* AG_b *with*

$$GP_a = \|AG_a\| \times \gamma_a = GP_b = \|AG_b\| \times \gamma_b,$$

AG_a *has a higher priority than* AG_b *if the factor difference* Λ_a *is smaller than* Λ_b, *where the factor difference is expressed as* $\Lambda_i = | \parallel AG_i \parallel - \gamma_i |$.

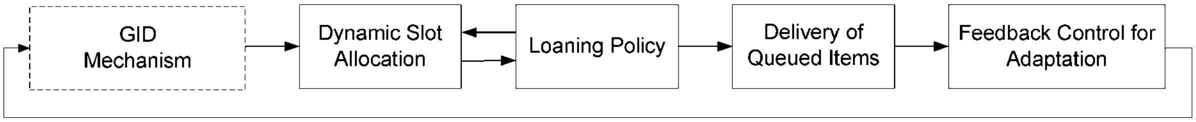


Fig. 4. The flowchart of the LSAFC technique.

We apply Definition 3 because the group popularity of a virtual group is more stable than that of an actual group. Explicitly, referring to Definition 2, the GP_i of a VG_i is mainly dominated by the various $\|VG_i\|$ with a constant production rate $\gamma_i = 1$. In contrast, the GP_i of an AG_i is more sensitive to the dynamic changes of $\|AG_i\|$ and γ_i, as specified in Definition 1. In addition, Definition 4 states that when two actual groups have the same group popularity, the one with a smaller Λ_i is less sensitive and is selected earlier. In practice, it is believed that, for an AG_i with a larger Λ_i, the increase or decrease in the value of γ_i or $\|AG_i\|$ can lead to a larger variation of GP_i. For example, suppose that AG_1 and AG_2 have $GP_1 = GP_2$, where $\|AG_1\| = 6$, $\gamma_1 = 4$, $\|AG_2\| = 12$, and $\gamma_2 = 2$. AG_1 has a higher schedule priority than AG_2 because $\Lambda_1 = 2 < \Lambda_2 = 10$ and the change of GP_1 is smaller than that of GP_2 in the case that both of γ_1 and γ_2 are increased by one item.

In the GID mechanism, the mean factor difference is used to consolidate the determination of the group schedule priority. After each broadcast cycle, the server has the values of

$$\overline{\|AG_i\|} = \frac{\sum_{1\le k\le L}\|AG_i(k)\|}{L},$$

γ_i, and $\overline{\Lambda}_i = \left|\overline{\|AG_i\|} - \gamma_i\right|$ of each AG_i. Consequently, according to (1) and (2), and the schedule priority, the server is able to cyclically adjust the group classification and the broadcast program by (3).

4 DESIGN OF LSAFC TECHNIQUE

This section devises an online loan-based slot allocation and feedback control (LSAFC) technique to enhance the performance of the GID mechanism. Section 4.1 presents the procedure of the LSAFC technique. Section 4.2 presents three loaning policies for dynamic slot allocation. Section 4.3 describes the feedback control for the GID mechanism. The algorithmic procedure of the GID+LSAFC mechanism is given in Section 4.4.

4.1 Procedure of the LSAFC Technique

Even though the GID mechanism adjusts the slot allocation periodically in the beginning of each broadcast cycle, the slot allocation is still vulnerable during data broadcasting, while the traffic changes dynamically. Particularly, the slot quota assigned to a group can be excess or scarce. Hence, the design of the LSAFC technique complements the GID mechanism functionally to improve the use of broadcast bandwidth and the performance.

Fig. 4 illustrates the procedure of the LSAFC technique. In the GID framework, as mentioned in Section 3, each group in the broadcast program is assigned with a slot quota in the start of each broadcast cycle. After that, the server performs dynamic slot allocation by allowing the loan of slots among groups during the broadcast cycle. The loan-based slot allocation is processed whenever some slots are available before the end of the broadcast cycle. Particularly, a group can loan slots from other groups to disseminate additional data items, in light of the specific loaning policy, once a group has exhausted its slot quota. However, it is possible that a group cannot get a loan of slots because the aggregate of dynamic data items exceeds the total slot quota reserved for the broadcast program. To resolve it, the server will deliver the excess items either by the pull way or by the push way with slot preemption in the next broadcast cycle. In addition, in the end of a broadcast cycle, the server has the loan-based feedback information for the GID mechanism to control accordingly the adaptation of slot allocation and broadcast program generation. Therefore, the LSAFC technique enables the GID mechanism to use the broadcast bandwidth efficiently and avoid the performance degradation for dynamic information dissemination.

4.2 Dynamic Slot Allocation

An efficient dynamic slot allocation is significant in the design of a robust LSAFC technique. Since the broadcast traffic is unpredictable, some heuristics are applied to devise the loaning policies. Accordingly, we present three loaning policies: the sensitive loan, the insensitive loan, and the greedy loan, for the server to determine the slot loaner. Note that dynamic slot allocation enables the server to tolerate the dynamic data generation.

Let a variable t present the time moment during a broadcast cycle with $0 \le t \le L$. The time interval $[0, t]$ means the elapsed time from the beginning of the broadcast cycle to the current moment. Suppose that the server receives a dynamic item d_i from the information service I_i at t. The server will take one slot out of the slot quota S_i of the group G_i to broadcast item d_i. In case the slot quota is used up, i.e., $S_i = 0$, the server will attempt to loan a slot from another group decided by one of the following policies:

- **Sensitive Loan (SL)**. Let S_i^t denote the number of used slots of each group G_i within $[0, t]$. The server estimates the total number of slots that will be used by each group within $[0, L]$ as $S_i' = S_i^t \times \frac{L}{t}$. Thus, the estimated number of remained slots of each group at the moment $t = L$ is given as $\Delta S_i = S_i - S_i' = S_i - S_i^t \times \frac{L}{t}$. Accordingly, the group with the maximal ΔS_i is selected as the slot loaner.
- **Insensitive Loan (IL)**. The server considers the slot quota S_i of each group as the base and the number of used slots S_i^t without regard to the future traffic within $[t, L]$. The server decides the group with the maximal value of $\frac{S_i - S_i^t}{S_i} = 1 - \frac{S_i^t}{S_i}$ as the slot loaner.

- **Greedy Loan (GL)**. The group which has the maximal number of remained slots, i.e., $S_i - S_i^t$, at the moment, t is served as the slot loaner.

In the SL policy, for each group, the server analyzes the temporary pattern of slot utilization, and then accordingly estimates the overall slot utilization progressively to the end of the broadcast cycle. Note that the SL policy is responsive to the decrease or increase of dynamic item production rate. Oppositely, the IL policy adopts a conservative premise that the slot utilization of each group will be the same as that in last broadcast cycle. Hence, the IL policy is less susceptible to the bursty traffic change. In addition, we investigate the feasibility of the GL policy without concern of any traffic patterns. These three loaning policies will be examined comparatively in Section 5.

4.3 Feedback Control

According to the loan-based feedback from the dynamic slot allocation, the server is able to control the adaptation in the GID mechanism, thereby maintaining the performance. This section describes the process of the feedback control below.

4.3.1 Direct Feedback

In the end of each broadcast cycle, the server acquires the information of slot utilization during the time period of a broadcast cycle. For each group G_i with an initial slot quota S_i, the server knows the number of slots lent to other groups S_i^-, the number of slots borrowed from other groups S_i^+, the number of remained slots $S_i^\times$, and the number of items kept in the queue Q_i. Thus, the server can learn the amount of slots taken to deliver all group-specific items of G_i by the calculation of the direct feedback (DF) method as

$$\mathrm{DF}: \; S_i' = S_i + S_i^+ - S_i^- - S_i^\times + Q_i, \quad (4)$$

where S_i' is the required slot quota to meet the actual need, that is, the dynamic item production rate γ_i' in this broadcast cycle. We notice that either S_i^+ or $S_i^\times$ is equal to zero in the end of every broadcast cycle, hereof depending on the various γ_i'. Particularly, a group will try to borrow slots from other groups only if it has used up the slot quota. In case γ_i' is larger than S_i, we have $S_i^+ \geq 0$ and $S_i^\times = 0$; otherwise, we have $S_i^+ = 0$ and $S_i^\times \geq 0$. In contrast, the values of S_i^+ and S_i^- are decided by the specific loaning policy and the data sequence received by the server. On the other hand, when a broadcast cycle finishes, there may exist Q_i items of group G_i in the queue because the aggregate of dynamic items of all groups possibly exceeds the bandwidth accommodation for the broadcast program.

Consequently, the DF method is used to measure dynamic item production rate reflectively from the feedback information. Along with the measure of group popularity in Section 3.4, accordingly, the server is able to further perform the adaptation, as specified in Section 3, for the next broadcast cycle.

4.3.2 Dissemination of Queued Items

Note that subject to the real-time restriction, the server must digest the queued items in some "real-time" fashions in advance of the next broadcast cycle. Based on the hybrid data broadcast model, two ways are considered below:

- **Pull**: The server delivers the queued items by the pull way in the end of the current broadcast cycle.
- **Push with slot preemption**: The server preempts slots in the next broadcast cycle and broadcasts the queued items by the push way.

However, disseminating queued items gives rise to an uncertain increase or decrease of message traffic. In practice, because the server has no knowledge of the future traffic, it is problematic in the selection of the pull way or the slot preemption. To provide more insight, several experiments are conducted to inspect the performance comparison as will be given in Section 5.

4.4 GID+LSAFC Procedure

In light of Sections 3 and 4, we integrate the procedures of the GID mechanism and the LSAFC technique as follows:

Step 1. Input: A group set.

a. The broadcasting server computes the group popularity GP_i of each AG_i VG_i.
b. According to the relative group popularity, the server performs group classification.
c. The server generates the broadcast program P by iteratively selecting a group G_i of the maximal schedule priority into P. In addition, each group in P is assigned with a slot quota S_i with (3).

Output: a flat broadcast program P, the length of a broadcast cycle L, the hot group set U^H, and the cold group set U^C.

Step 2. Input: P, L, and a queue Q which contains the received data items subsequently.

a. In the time period of L, the server measures the variations of $\|AG_i\|$, γ_i, and $\|VG_i\|$.
b. In the time period of L, the server iteratively fetches an item in Q and broadcasts it.
    - The server sequentially broadcasts the item of group G_i of the maximal group schedule priority.
    - The server decreases S_i by one. In case G_i has exhausted its slot quota, i.e., $S_i = 0$, G_i will loan a slot from another group in accordance with the GL/SL/IL loaning policy except that all groups have used up their slot quotas.

Step 3. In the end of L,

a. The server checks the queue Q and selects either the pull way or the push way with slot preemption for disseminating the queued items.
b. In accordance with the loan-based feedback information, the server uses the direct feedback method to calculate the mean group popularity $\overline{GP}_i$ of each G_i.
c. The server repeats Step 1 to adapt P, L, U^H, U^C, and S_i of each G_i in P.

After each broadcast cycle, the server repeats this procedure to adapt group popularity, group classification, and broadcast program. Note that each G_i in P has S_i slots corresponding to its γ_i. When the remaining slots are insufficient to broadcast the next selected group, the server

TABLE 2
Simulation Parameters Description

notation	meaning	value
AG	the amount of actual groups	100, 200, 400
VG	the amount of virtual groups	100, 200, 400
b	the number of broadcast bandwidth slots	100, 200, 300, 400, 500
λ	the mean number of dynamic/static items in a time unit	$10 \sim 30$
θ	the skew coefficient in the Zipf distriubtion	$0.0 \sim 1.6$
W_A	the workload of actual groups in a time unit	$variable = AG \times \lambda$
W_V	the workload of virtual groups in a time unit	$variable = VG \times \lambda$
R	the times of the broadcast cycle	$variable = 1 \sim 10$

will never look for a group that has lower schedule priority and a smaller item production rate to take those remanent slots. Therefore, the server ensures that all groups in P are relatively hotter than others in U^C.

5 PERFORMANCE EVALUATION

Section 5.1 models the simulation environment. Section 5.2 specifies the performance metrics. Section 5.3 inspects the GID mechanism in comparison with the unicast model. The LSAFC technique with various loaning policies is examined in Section 5.4. In Section 5.5, several experiments are conducted to demonstrate the efficacy of the GID+LSAFC mechanism. Finally, Section 5.6 summarizes significant remarks.

5.1 Simulation Environment

Table 2 lists the simulation parameters description. In accordance with Definitions 1 and 2, the aggregate of the traffic workload in a time unit is the summation of $W_A = \sum_{AG_i} \|AG_i\| \times \gamma_i$ and $W_V = \sum_{VG_i} \|VG_i\|$. Given with b broadcast slots, the server generates the broadcast program by the relative group popularity. Because an AG_i has $GP_i = \|AG_i\| \times \gamma_i$, the simulator tunes the ratio $\frac{\gamma_i}{\|AG_i\|}$ in the range of $\left[\frac{1}{4}, \frac{1}{2}, 1, 2, 4\right]$ to examine the influence of various traffic factors.

5.1.1 Access Frequency Distribution

W_A and W_V are generated by a Zipf distribution with a skew coefficient θ [42], expressed as

$$p_i = \left(\frac{1}{i}\right)^{\theta} / \sum_{1 \le i \le L} \left(\frac{1}{i}\right)^{\theta},$$

which is used to model the skewed access frequency distribution [7], [17], [20], [29], [39]. The access frequency λ_i for an item d_i is $p_i \cdot \lambda$. The distribution becomes skewer as θ increases and reduces to a uniform distribution as $\theta = 0$.

5.1.2 Dynamic Traffic Generation

We describe the generations of dynamic workload and dynamic access pattern below. A joint change of workload and access pattern is also considered in this simulation:

- *Dynamic workload*: The increase or the decrease of dynamic W_A is mapped to the geometric series with a rate $-1 < \alpha < 1$, that is,

  $$W_A = \{W_{A_0}(1+\alpha)^0, W_{A_0}(1+\alpha)^1, \ldots\}.$$

  In addition, the dynamic W_V is generated similarly.
- *Dynamic access pattern*: Given with a destined skew coefficient θ_2, the skew access pattern in the ith run is generated by the Zipf distribution with $\theta_i = \theta_1 + \frac{i}{R}(\theta_2 - \theta_1)$ and $1 \le i \le R$.

5.2 Performance Metrics

This section specifies the server-centric performance metrics to evaluate the GID mechanism and the LSAFC technique. Note that the user-centric metrics, e.g., mean access time, adopted usually in the traditional model, are not applicable in dynamic data dissemination contexts, while data are produced dynamically. In contrast, the user-central metrics are used to evaluate the static optimization of the indexing and the scheduling techniques in the traditional model where the broadcast schedule or data contents are static and predetermined in advance of each broadcast cycle.

5.2.1 Measure of Dynamic Message Traffic

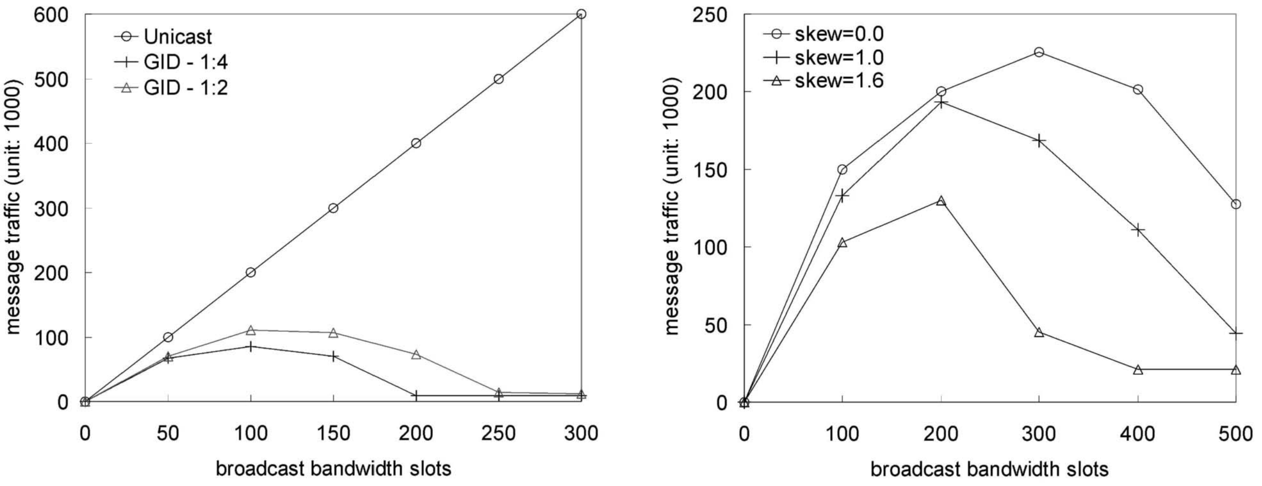
Based on an extended hybrid data broadcast model, the GID+LSAFC mechanism broadcasts dynamic data items of hot groups by the push way, delivers dynamic items of cold groups by the pull way, and disseminates the queued items by either the pull way or the push way with slot preemption. Accordingly, the measure of dynamic message traffic within a cycle time includes the broadcast traffic of hot actual groups and virtual groups in P, the pull traffic of cold actual groups and virtual groups, and the extra traffic caused by delivering queued items in the end of a broadcast cycle. The respective mathematical expressions are given below.

GID+LSAFC and the pull way for the queued items:

$$T = \sum_{\forall AG_i \in U^H} \left(\left(S_i' - Q_i \right) \times 1 + Q_i \cdot \|AG_i\| \right) + \sum_{\forall VG_i \in U^H} 1 + \sum_{\forall AG_i \in U^C} (\gamma_i \cdot \|AG_i\|) + \sum_{\forall VG_i \in U^C} \|VG_i\|.$$

GID+LSAFC and the push way for the queued items with slot preemption:

$$T = \sum_{\forall AG_i \in U^H} S_i' + \sum_{\forall VG_i \in U^H} 1 + \sum_{\forall AG_i \in U^C} (\gamma_i \cdot \|AG_i\|) + \sum_{\forall VG_i \in U^C} \|VG_i\| + \Delta T,$$



(a)

(b)

Fig. 5. (a) Various $\frac{\gamma_i}{\|AG_i\|}$ (experimental baseline: $\lambda = 10$, $AG = VG = 100$, and $\theta = 0.8$), and $\mathrm{ratio} - \mathrm{r} : \|\mathrm{AG}\|$. (b) skew access pattern (experimental baseline: $\lambda = 5$ and $AG = VG = 200$), and skew coefficient.

where ΔT is the possible decrease/increase of message traffic caused by the slot preemption. Particularly, the slot preemption decreases the number of broadcast slots for the broadcast program in the next broadcast cycle; nevertheless, the message traffic may decrease or increase due to the dynamic traffic changes. Hence, the relative performance is examined in the simulation.

5.2.2 Percent of Loaned Slots

Under the same traffic conditions, the percent of loaned slots is used to assess the relative superiority among the SL, IL, and GL loaning policies.

$$\text{Percent of loaned slots} = \frac{\text{the number of loaned slots}}{\text{the length of the broadcast cycle}} \times 100\%.$$

Note that the percent of loaned slots indicates the robustness of a loaning policy and the overhead of slot maintenance and computation. Explicitly, the computation overhead is the cost to determine a slot loaner, and the maintenance of slot allocation causes extra overhead of state space.

5.3 Evaluation of Basic GID

In this section, we conduct several experiments to evaluate the scalability of the GID mechanism in terms of the sensitivity to dynamic traffic factors: item production rate, workload and skew access pattern, as considered in the simulation.

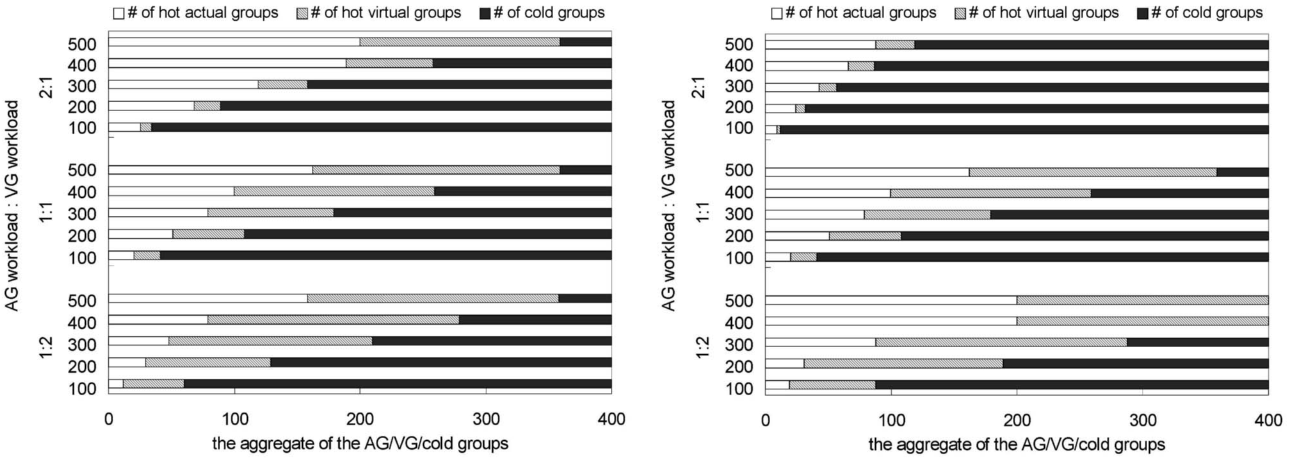
In Fig. 5a, the linear increase of message traffic by unicasting dynamic items obviously suggests the superiority of the GID mechanism. With the increment of broadcast bandwidth, the broadcast program maintains a larger number of groups and delivers the group-specific data on the broadcast channel. Correspondingly, the tendency of the increase of message traffic is diminished. Eventually, the message traffic has no changes when the broadcast bandwidth is enough to accommodate dynamic data of all groups. In addition, Fig. 5b presents the experimental results regarding the sensitivity to the skew access pattern. With a skew access pattern, most workload congregates in few groups of relatively larger popularity. The message traffic can be reduced rapidly if the available bandwidth is enough for broadcasting these groups. It is also depicted in Fig. 5b that the skewer the access pattern, the lower the message traffic.

Further, the GID mechanism is examined under dynamic workloads which are generated with various ratios of $\frac{W_A}{W_V}$ and $\frac{\gamma_i}{\|AG_i\|}$. Fig. 6 shows the measured results of group classification. In Fig. 6a, in case that γ_i is constant as the value in the initial workload, the number of hot virtual groups in the broadcast program decreases when the workload W_A is increased by the incremental ratio of $\frac{W_A}{W_V}$. Oppositely, the number of hot actual groups increases slightly. On the other hand, as illustrated in Fig. 6b, in case of a constant $\|AG_i\|$, both of the respective amounts of hot actual and hot virtual groups decrease when the ratio of $\frac{W_A}{W_V}$ increases. This phenomenon is explained as follows: Most workload belongs to actual groups, and each AG_i needs a larger slot quota corresponding to the increase of γ_i. In contrast, there are few virtual groups of higher group popularity in this case. Therefore, the total amount of hot groups decreases in response to the incremental $\frac{W_A}{W_V}$, but increases under the decrement of $\frac{W_A}{W_V}$.

5.4 Evaluation of LSAFC with Various Loaning Policies

Several experiments have been conducted to synthetically evaluate the LSAFC technique with different loaning policies in terms of the percent of loaned slots. The numerical results are listed in Table 3.

We notice that using prior traffic patterns to generate a reliable broadcast program, as assumed commonly, is paradoxical in dynamic data dissemination environments. It is showed in Table 3 that the IL policy has the poorest performance. Particularly, when the tendency of traffic



(a)

(b)

Fig. 6. (a) $\frac{W_A}{W_V}$ versus constant γ, and constant item production rate. (b) $\frac{W_A}{W_V}$ versus constant $\|AG_i\|$ (experimental baseline: $\frac{\gamma_i}{\|AG_i\|} = 1$, $\lambda = 6$, $AG = VG = 200$, and $\theta = 0.6$), and constant number of clients.

changes and the slot utilization are similar to the past patterns, the server can allocate each hot group a slot quota near to the actual need within a broadcast cycle. Due to the unpredictable input sequence of dynamic data, however, dynamic slot allocation with the IL policy is inefficient. In contrast, the SL policy takes the temporary traffic patterns into consideration and attains the best performance. On the other hand, we deliberate that the GL policy is prominent with its simplicity and less computation overhead, although the SL policy is better than the GL in a degree of $1 \sim 2\%$. Therefore, the GL policy is considered when the broadcast cycle is lengthy, or when the overhead of slot allocation is critical.

Furthermore, as the comparison with parts 1, 2, and 6 in Table 3, the amount of loaned slots decreases under a higher skew traffic ($\theta = 0.8 > 0.6 > 0.2$). When the skew access pattern is stable, some slots may remain, if the workload decreases, as the comparison between parts 2 and 3, and parts 5 and 6. In case the access pattern becomes skewer drastically, the percent is decreased with more available bandwidth slots. Since most workload congregates to few hot groups, other groups have lower possibility to loan slots. In addition, while scheduling the broadcast program, the server ensures that the groups in the broadcast program are relatively hotter than other cold groups. A group of lower schedule priority cannot be taken to digest the remanent $b - L$ slots. Consequently, the length of the broadcast cycle L is slightly smaller or equal to b as shown in Table 3.

TABLE 3
The Measured Results by the LSAFC Technique with the GL, SL, and IL Loaning Policies (Experimental Baseline: $AG = VG = 200$, $\frac{\gamma_i}{\|AG_i\|} = 1$, and $\lambda = 10$)

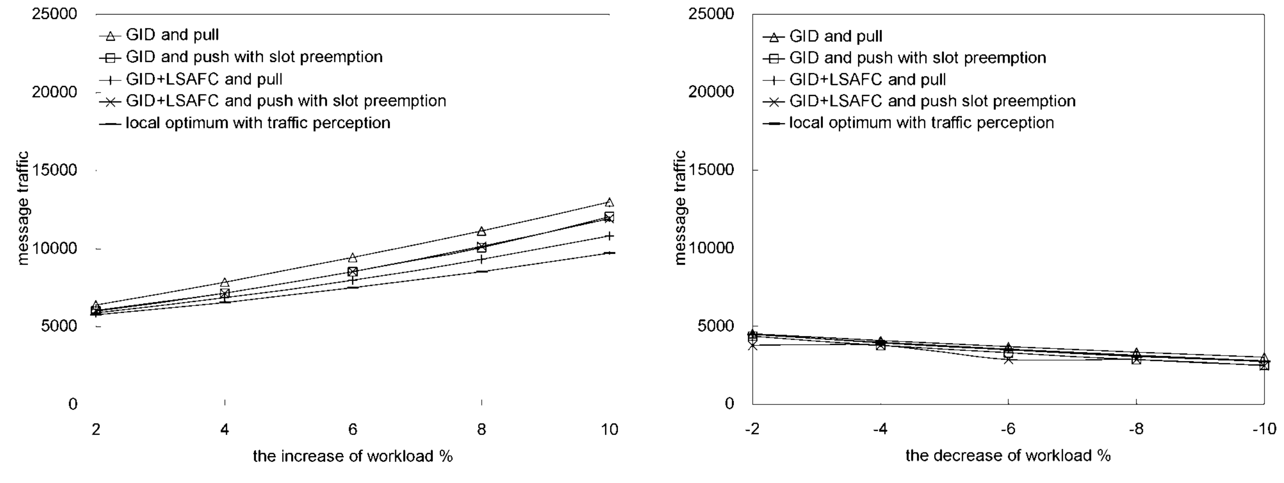
	b	$\theta_1 : \theta_2$	$W_1 : W_2$	AG# : VG#	L	GL (%)	SL (%)	IL (%)
1	100	0.2:0.6	1:2	22:27	99	28.3	27.8	31.0
	300	0.2:0.6	1:2	67:93	300	25.3	26.1	31.0
	500	0.2:0.6	1:2	114:152	500	23.0	22.1	29.0
2	100	0.6:0.2	1:2	14:16	96	3.1	3.4	3.1
	300	0.6:0.2	1:2	60:75	300	12.9	13.8	12.8
	500	0.6:0.2	1:2	119:155	500	22.0	22.1	23.3
3	100	0.6:0.2	1:1.4	14:16	96	0	0	0
	300	0.6:0.2	1:1.4	60:75	300	3.7	3.7	3.8
	500	0.6:0.2	1:1.4	119:155	500	14.1	14.2	15.5
4	100	0.6:1	1:1	14:16	96	16.2	15.5	19.0
	300	0.6:1	1:1	60:75	300	15	13.7	18.9
	500	0.6:1	1:1	119:155	500	8.4	8.4	11.7
5	100	0.2:0.8	1:0.9	22:27	99	25.5	24.6	27.4
	300	0.2:0.8	1:0.9	67:93	300	20.7	19.4	27.1
	500	0.2:0.8	1:0.9	114:152	500	13.5	13.1	19.4
6	100	0.8:0.2	1:0.9	12:13	99	0	0	0
	300	0.8:0.2	1:0.9	61:66	298	4.4	4.4	4.5
	500	0.8:0.2	1:0.9	128:133	499	3.8	3.8	4.1

5.5 Evaluation of GID+LSAFC

This section presents the investigation on the GID+LSAFC under dynamic workload and dynamic skew access pattern. The measured message traffic is compared with the local optimum, that is, the comparison baseline which is available only if the server could perceive dynamic traffic patterns in the beginning of each broadcast cycle.

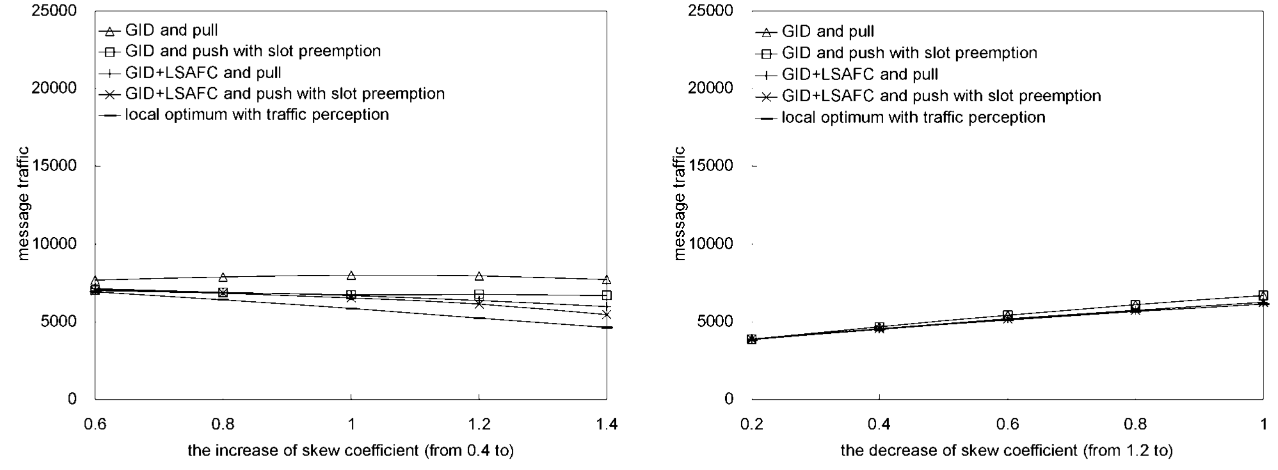
5.5.1 Dynamic Workload

Fig. 7 depicts the experimental results under the increasing and the decreasing workloads. Observe that the GID+LSAFC mechanism is able to reduce the message traffic prominently. According to the loan-based feedback, the GID mechanism is able to perform the adaptation of group popularity, group classification, broadcast program, and slot allocation in the end of the broadcast cycle. Regarding the delivery of excess items, it is shown in Fig. 7a that the pull way has the better performance when the workload increases. In contrast, as depicted in Fig. 7b, the push way with slot preemption is the best when the workload decreases; noteworthily, it can minimize the message traffic to be lower than the local optimum based on the local traffic pattern within the current cycle. Explicitly, the decreasing workload will cause the number of hot items to be smaller than the total slot quota in the next broadcast. Thus, slot preemption will not result in the



(a) (b)

Fig. 7. Comparison under dynamic workload (experimental baseline: $R = 10$, $b = 200$, $\lambda = 20$, $AG = VG = 200$, $\frac{\gamma_i}{\|AG_i\|} = 1$, and $\theta = 0.6$).



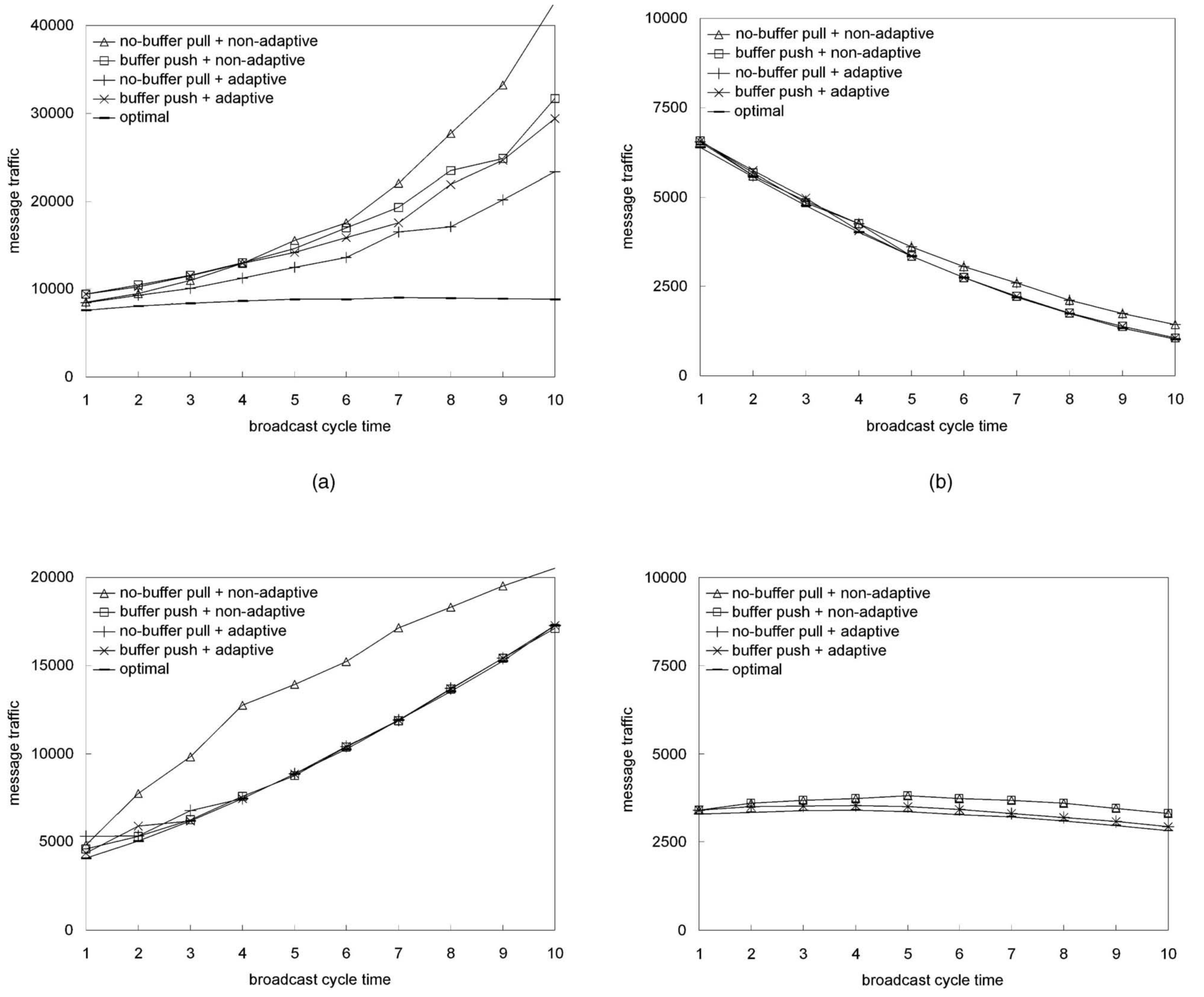
(a) (b)

Fig. 8. Comparison under dynamic skew access pattern (experimental baseline: $R = 10$, $b = 200$, $\lambda = 20$, $AG = VG = 200$, and $\frac{\gamma_i}{\|AG_i\|} = 1$).

increase of message traffic. In addition, as plotted in Fig. 7a, when the slot preemption is used, the message traffic in the GID mechanism is close to that in the GID+LSAFC mechanism. However, it is noted that the former significantly incurs about $4 \sim 5$ times the buffer size required for temporarily keeping the excess items in the latter. Furthermore, using slot preemption to broadcast queued items is less susceptible to dynamic workload and may not lead to a drastic increase of message traffic. In contrast, the message traffic by unicasting the queued items is critical and related to the number of queued items and the item popularity. Consequently, the GID+LSAFC mechanism is able to get the best performance, while the workload increases. Otherwise, the GID+LSAFC mechanism with the slot preemption is more profitable under a decreasing workload.

5.5.2 *Dynamic Skew Access Pattern*

Several experiments are conducted to evaluate the relative performance under dynamic skew access patterns. Fig. 8 illustrates the experimental results in the increment and the decrement of skew coefficients. Generally speaking, more workload congregates in fewer groups with the increment of skew coefficient. The overall message traffic is reduced by allocating broadcast slots first to the relatively hotter groups. In addition, there are several observations regarding the delivery of excess items as follows: First, as depicted in Fig. 8a, the extra message traffic caused by delivering the excess queued items reduces gradually in reverse proportion to the incremental skew coefficient. Deductively, there may be no extra pull message traffic in case of more bandwidth or a drastic increment of skew coefficient since all excess items can be satisfied by loaning slots alternatively. Second, the push way with slot preemption has a substantial reduction of message traffic in comparison with the pull way. Particularly, given a broadcast program, more slots will remain when the skew access pattern becomes skewer. Hence, the slot preemption in the next cycle will not much affect the dissemination of dynamic data in the next broadcast program. On the other hand, as compared with the case under dynamic workload, the variation of message traffic is more sensitive to the incremental skew coefficient due to the higher frequency of dynamic slot allocation. In



(a)

(b)

(c)

(d)

Fig. 9. Comparison under various joint changes of dynamic workload and skew access pattern (experimental baseline: $\lambda = 25$, $AG = VG = 200$, and $\frac{\gamma_i}{\|AG_i\|} = 1$). (a) $W = +8\%$, $\mathrm{theta} = 0.4$ to 1.2, and $b = 200$. (b) $W = -8\%$, $\mathrm{theta} = 0.4$ to 1.2, and $b = 200$. (c) $W = +8\%$, $\mathrm{theta} = 1.2$ to 0.4, and $b = 200$. (d) $W = -8\%$, $\mathrm{theta} = 1.2$ to 0.4, and $b = 200$.

contrast, in the case of the decremental skew coefficient, the slot allocation based on the feedback in the last broadcast cycle is sufficient to sustain dynamic item production rates. Remarkably, the GID+LSAFC mechanism with slot preemption is able to reduce the message traffic prominently under dynamic skew access patterns. Further, the pull way is able to attain similar performance to that of the push way when the access pattern becomes smoother.

5.6 Summary and Discussion

To make a synthetic comparison, we further investigate the relative performance under the joint changes of dynamic workload and skew access pattern. In light of extensive experiments in the simulation, several remarks are made below.

The GID+LSAFC mechanism has demonstrated the scalability for dynamic data dissemination in terms of message traffic. We have evaluated the design of the GID mechanism and the LSAFC technique against the performance impacts of the system-designed factors with various traffic conditions. The efficiency and robustness of the GL, IL, and SL loaning policies are comparatively investigated in terms of the percent of loaned slots and the overhead of slot maintenance. It is shown that the dynamic slot allocation is more efficient, while the temporary traffic patterns during the cyclic broadcast is considered, instead of the past traffic patterns.

As shown by the experimental results, remarkably, the GID+LSAFC mechanism achieves the best performance by using the pull way to disseminate queued items, while dynamic workload increases. In contrast, the GID+LSAFC mechanism with slot preemption for broadcasting queued items is more profitable in response to the incremental skew of dynamic access pattern. Furthermore, the performance difference between the GID mechanism and the GID+LSAFC mechanism, as illustrated in Fig. 9, is more discernible under a joint change of dynamic workload and access pattern. Comparatively, the LSAFC technique enables the GID mechanism to perform dynamic slot allocation efficiently for broadcasting hot items as soon as possible. The message traffic caused by delivering excess queued items will thus be reduced. Therefore, the above observations show the advantages and efficacy of the GID+LSAFC mechanism.

6 CONCLUSIONS

The work in this paper has devised the GID+LSAFC mechanism, essentially an adaptive information dissemination framework, by exploiting the potential of data broadcasting. The GID mechanism not only supports dynamic information dissemination, but is also backward compatible with the conventional data broadcast scheme. Considering both of the client-oriented and the server-oriented traffic factors, we have further designed an online LSAFC technique to sustain dynamic traffic changes. Based on the LSAFC technique, the GID mechanism is able to perform the adaptations of broadcast slot allocation, group association, popularity and classification, and broadcast program. Extensive simulations have been conducted to evaluate the performance of the GID+LSAFC mechanism. The experimental results have shown that the GID mechanism is very scalable and attains a substantial reduction of dynamic message traffic. In addition, the LSAFC technique is able to complement the GID mechanism functionally, therefore enhancing the performance and making the GID mechanism more efficient and robust.

ACKNOWLEDGMENTS

The authors are supported in part by the Ministry of Education Project No. 89-E-FA06-2-4, and the National Science Council Project No. NSC 92-2213-E-002-001 and NSC 92-2213-E-002-010, Taiwan, Republic of China.

REFERENCES

[1] The 3rd Generation Partnership Project, http://www.3gpp.org, 2003.

[2] S. Acharya, R. Alonso, M.J. Franklin, and S.B. Zdonik, "Broadcast Disks: Data Management for Asymmetric Communications Environments," *Proc. ACM Int'l Conf. Management of Data,* pp. 199-210, May 1995.

[3] S. Acharya, M.J. Franklin, and S.B. Zdonik, "Prefetching from Broadcast Disks," *Proc. 12th IEEE Int'l Conf. Data Eng.,* pp. 276-285, Feb. 1996.

[4] S. Acharya, M.J. Franklin, and S.B. Zdonik, "Balancing Push and Pull for Data Broadcast," *Proc. ACM Int'l Conf. Management of Data,* pp. 183-194, May 1997.

[5] S. Acharya and S. Muthukrishnan, "Scheduling On-Demand Broadcasts: New Metrics and Algorithms," *Proc. ACM MobiCom,* pp. 43-54, 1998.

[6] D. Aksoy and M. Franklin, "Rxw: A Scheduling Approach for Large-Scale On-Demand Data Broadcast," *IEEE/ACM Trans. Networking,* vol. 7, no. 6, pp. 846-860, 1999.

[7] M.H. Ammar and J.W. Wong, "On the Optimality of Cyclic Transmission in Teletext Systems," *IEEE Trans. Comm.,* vol. 11, pp. 1159-1170, 1987.

[8] A. Bar-Noy, R. Bhatia, J. Naor, and B. Schiber, "Minimizing Service and Operation Costs of Periodic Scheduling," *Proc. Ninth ACM-SIAM Symp. Discrete Algorithms (SODA '98),* pp. 11-20, Jan. 1998.

[9] A. Bar-Noy, J. Naor, and B. Schiber, "Pushing Dependent Data in Clients-Providers-Servers Systems," *Proc. ACM MobiCom '00,* pp. 222-230, 2000.

[10] D. Barbara, "Mobile Computing and Database—A Survey," *IEEE Trans. Knowledge and Data Eng.,* vol. 11, no. 1, pp. 108-117 Jan./Feb. 1999.

[11] T. Bowen et al., "The Datacycle Architecture," *Comm. ACM,* vol. 35, no. 12, pp. 71-81, 1992.

[12] J. Cai and K. -L. Tan, "Tuning Integrated Dissemination-Based Information Systems," *Data & Knowledge Eng.,* vol. 30, no. 1, pp. 1-21, 1999.

[13] Y.-I. Chang and C.-N. Yang, "A Complementary Approach to Data Broadcasting in Mobile Information Systems," *Data & Knowledge Eng.,* vol. 40, no. 2, pp. 181-194, Feb. 2002.

[14] M.-S. Chen, P.S. Yu, and K.-L. Wu, "Optimizing Index Allocation for Sequential Data Broadcasting in Wireless Mobile Computing," *IEEE Trans. Knowledge and Data Eng.,* vol. 15, no. 1, Jan./Feb. 2003.

[15] R. Deolasee, A. Kathar, A. Panchbudhe, K. Ramamritham, and P. Shenoy, "Dissemination of Dynamic Data," *Proc ACM SIGMOD '01,* p. 599, May 2001.

[16] H. Dykeman, M. Ammar, and J. Wong, "Scheduling Algorithms for Videotex Systems under Broadcast Delivery," *Proc. IEEE Int'l Conf. Comm.,* pp. 1847-1851, 1986.

[17] M.J. Franklin and S.B. Zdonik, "Dissemination-Based Information Systems," *Data Eng. Bull.,* vol. 19, no. 3, pp. 20-30, 1996.

[18] S. Hameed and N.H. Vaidya, " Efficient Algorithms for Scheduling Data Broadcast," *ACM/Baltzer Wireless Networks,* vol. 5, no. 3, pp. 183-193, 1999.

[19] C.-L. Hu and M.-S. Chen, "Adaptive Balanced Hybrid Data Delivery for Multi-Channel Data Broadcast," *Proc. IEEE Int'l Conf. Comm.,* Apr. 2002.

[20] C.-L. Hu and M.-S. Chen, "Dynamic Data Broadcasting with Traffic Awareness," *Proc. 22nd IEEE Int'l Conf. Distributed Computing Systems,* July 2002.

[21] Q. Hu, D.L. Lee, and W.-C. Lee, "Dynamic Data Delivery in Wireless Communication Environments," *Proc. ER '98 Workshops Mobile Data Access,* pp. 218-229, 1998.

[22] Q. Hu, W.-C. Lee, and D.L. Lee, "Indexing Techniques for Wireless Data Broadcast under Data Clustering and Scheduling," *Proc. Eighth Int'l Conf. Information and Knowledge Management,* 1999.

[23] T. Imielinski and B.R. Badrinath, "Data Management for Mobile Computing," *SIGMOD Record,* vol. 22, no. 1, pp. 34-39, 1993.

[24] T. Imielinski and S. Viswanathan, "Adaptive Wireless Information Systems," *Proc. ACM Special Interest Group on DataBase Systems,* pp. 19-41, 1994.

[25] 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.

[26] Y.-B. Lin, "A Multicast Mechanism for Mobile Networks," *IEEE Comm. Letter,* vol. 5, no. 11, pp. 450-452, 2001.

[27] S. Paul, *Multicasting on the Internet and Its Applications.* Boston: Kluwer Academic Publishers, 1998.

[28] W.-C. Peng and M.-S. Chen, "Dynamic Generation of Data Broadcasting Programs for a Broadcast Disk Array in a Mobile Computing Environment," *Proc. Ninth ACM Int'l Conf. Information and Knowledge Management,* Nov. 2000.

[29] W.-C. Peng and M.-S. Chen, "Efficient Channel Allocation Tree Generation for Data Broadcasting in a Mobile Computing Environment," *Wireless Networks,* vol. 9, no. 2, pp. 117-129, 2003.

[30] E. Pitoura and R. Chrusanthis, "Scalable Processing of Read-Only Transactions in Broadcast Push," *Proc. IEEE Int'l Conf. Distributed Computing Systems,* Sept. 1999.

[31] E. Pitoura and R. Chrusanthis, "Multiversion Data Broadcast," *IEEE Trans. Computers,* vol. 51, no. 10, pp. 1224-1230, Oct. 2002.

[32] E. Pitoura, R. Chrusanthis, and K. Ramamritham, "Characterizing the Temporal and Semantic Coherency of Broadcast-Based Data Dissemination," *Proc. Ninth Int'l Conf. Database Theory,* 2003.

[33] A.K. Salkintzis, "A Survey of Mobile Data Networks," *IEEE Comm. Surveys,* vol. 2, no. 3, pp. 2-18, 1999.

[34] N. Schabanel, "The Data Broadcast Problem with Preemption," *Proc. 17th Int'l Symp. Theoretical Computer Science (STACS '00),* pp. 181-192, Feb. 2000.

[35] J. Shanmugasundaram et al., "Efficient Concurrency Control for Broadcast Environment," *Proc. ACM SIGMOD,* 1999.

[36] K. Stathatos, N. Roussopoulos, and J.S. Baras, "Adaptive Data Broadcast in Hybrid Networks," *Proc. 23rd Int'l Conf. Very Large Data Bases,* Aug. 1997.

[37] C.-J. Su, L. Tassiulas, and V.J. Tsotras, "Broadcast Scheduling for Information Distribution," *ACM/Baltzer Wireless Networks,* vol. 5, no. 2, pp. 137-147, 1999.

[38] K.-L. Tan and B.C. Ooi, *Data Dissemination in Wireless Computing Environments.* Boston: Kluwer Academic Publishers, 2000.

[39] N.H. Vaidya and S. Hameed, "Scheduling Data Broadcast in Asymmetric Communication Environments," *ACM/Baltzer Wireless Networks,* vol. 5, no. 3, pp. 171-182, 1999.

[40] J.W. Wong, "Broadcast Delivery," *Proc. IEEE,* vol. 76, no. 12, pp. 1566-1577, Dec. 1988.

[41] K.-L. Wu, P.S. Yu, and M.-S. Chen, "Energy-Efficient Caching for Bandwidth-Limited Wireless Mobile Computing," *Proc. 12th IEEE Int'l Conf. Data Eng.,* pp. 335-343, Feb. 1996.

[42] G.K. Zipf, *Human Behaviour and the Principle of Least Effort.* Reading, Mass.: Addison-Wesley, 1949.

Chih-Lin Hu received the BS degree in computer science from National ChengChi University, the MS degree from National ChungHsing University, and the PhD degree in National Taiwan University, respectively, in 1997, 1999, and 2003. He had the honor of receiving the best paper award at the IEEE ICPADS-2000. His research interests include mobile agent technology, broadcast information management, and active networks.

Ming-Syan Chen received the BS degree in electrical engineering from National Taiwan University, Taipei, Taiwan, and the MS and PhD degrees in computer, information, and control engineering from The University of Michigan, Ann Arbor, in 1985 and 1988, respectively. Dr. Chen is currently the chairman of the Graduate Institute of Communication Engineering and also a professor in the Electrical Engineering Department, National Taiwan University, Taipei, Taiwan. He was a research staff member at IBM Thomas J. Watson Research Center, Yorktown Heights, New York, from 1988 to 1996. His research interests include database systems, data mining, mobile computing systems, and multimedia networking, and he has published more than 160 papers in his research areas. In addition to serving as a program committee member in many conferences, Dr. Chen served as an associate editor of *IEEE Transactions on Knowledge and Data Engineering* on data mining and parallel database areas from 1997 to 2001, is on the editorial board of *VLDB Journal*, *Journal of Information Science and Engineering*, and *Journal of the Chinese Institute of Electrical Engineering*, was a distinguished visitor of IEEE Computer Society for Asia-Pacific from 1998 to 2000, and program chair of PAKDD-02 (Pacific Area Knowledge Discovery and Data Mining), program vice-chair of VLDB-2002 (Very Large Data Bases) and ICPP 2003, general chair of the Real-Time Multimedia System Workshop in 2001, program chair of the IEEE ICDCS Workshop on Knowledge Discovery and Data Mining in the World Wide Web in 2000, and program cochair of the International Conference on Mobile Data Management (MDM) in 2003, the International Computer Symposium (ICS) on Computer Networks, Internet and Multimedia in 1998 and 2000, and the ICS on Databases and Software Engineering in 2002. He was a keynote speaker on Web data mining for the International Computer Congress in Hong Kong, 1999, a tutorial speaker on Web data mining for the DASFAA-1999 and on parallel databases for the 11th IEEE International Conference on Data Engineering in 1995, and also a guest coeditor for the *IEEE Transactions on Knowledge and Data Engineering* on a special issue for data mining in December 1996. He holds, or has applied for, 18 US patents and seven ROC patents in the areas of data mining, Web applications, interactive video playout, video server design, and concurrency and coherency control protocols. He is a recipient of the NSC (National Science Council) Distinguished Research Award in Taiwan and the Outstanding Innovation Award from IBM Corporate for his contribution to a major database product, and also received numerous awards for his research, teaching, inventions, and patent applications. He coauthored with his students for their works which received ACM SIGMOD Research Student Award and Long-Term Thesis Award. Dr. Chen is a senior member of IEEE and a member of ACM.