

Dynamic Data Broadcasting with Traffic Awareness

Chih-Lin Hu and Ming-Syan Chen
Department of Electrical Engineering
National Taiwan University
Taipei, Taiwan, R.O.C.

E-mail: clhu@arbor.ee.ntu.edu.tw; mschen@cc.ee.ntu.edu.tw

Abstract

Data dissemination has significantly served as a scalable data delivery mechanism in wireless networks. However, even though the broadcast traffic has the nature of dynamic changes, most previous research efforts were elaborated upon the premise of static workloads and access patterns without having proper traffic awareness. In this paper, we address the existence of client impatience and accordingly devise an on-line traffic awareness mechanism based on a novel selective deferment and reflection technique (SDR) to estimate the dynamic workloads and access patterns in a granularity of a broadcast cycle. In comparison with prior probing and feedback approaches, our design is of practical usefulness in that it has low complexity and is light-weight without performance degradation. With various dynamic traffic scenarios, the experimental results show that with an increasing/decreasing workload, the real access frequency distribution is bounded by two specific estimated distributions. This fact in turn suggests us to employ a trigonometric tuning method to further enhance the estimation. In addition, we examine that the mean difference between the estimated access frequency distribution and the real one is very small, consequently indicating the feasibility and reliability of our proposed data broadcast mechanism with traffic awareness.

1 Introduction

With the limited bandwidth capacity, *data broadcasting* is a promising mechanism for the scalable information dissemination in wireless network environments [2][5][12]. As its paradigm, a server applies a broadcast program and delivers all data items in the database to clients through a shared medium periodically. The benefits are twofold: (1) in the client side, data broadcasting achieves a total saving of all clients' battery energy by avoiding per access request transmission, and (2) an information server can mod-

erately mitigate the inherent performance and scalability problems. Thus, the access time and the energy consumption are primary performance measures in a data broadcast system. Accordingly, there are three major categories in the data broadcast research community. First, broadcast scheduling aims to arrange the contents (a series of items) of the broadcast data so as to minimize the mean access time [2][9][13][17][19]. Second, indexing broadcast data can facilitate the clients' tuning on the broadcast medium, and can thus save the energy consumption as well as reduce access duration [7][11][14]. Third, a hybrid data delivery exploits the commonality of access interests to minimize the average access time by delivering hot (i.e., popular) data on the push (i.e., broadcast) bandwidth and cold (i.e., unpopular) data on the pull (i.e., on-demand) bandwidth [3][10][18]. In addition, the caching and pre-fetching mechanisms complement these three categories [1][22].

Note that in prior studies, there is a crucial weakness that a server does not possess *traffic awareness* against the nature of dynamic traffic. Explicitly, a server is unable to perceive the dynamic changes of workload, client population and access commonality because a client, in a sense, is passive and does not respond to his access, which is a challenging issue for a server to adjust the broadcast contents. Regardless, the previous research efforts were mainly based on that the workloads and access patterns are static and the prior knowledge of traffic changes is available. As a result, in order to attain dynamic adaptiveness and performance improvement, the probing and the feedback/piggyback techniques have been called for to cope with this issue [6][10][13][18]. A probing technique intentionally halts the broadcast service for a certain time period, which will in turn cause the clients to request data by the pull mode; meanwhile, the server can understand the push access interests from their corresponding pull requests. However, a *broadcast miss* will lead to many bursty requests and may congest the uplink channel. Incidentally, the partial probing and the sampling techniques are presented to alleviate the possible performance degradation [8]. As for the feedback technique, there exists an op-

timistic assumption that all clients will willingly keep their broadcast access statistics and report them actively, requiring further justifications in practice. Moreover, an extra structure is needed to represent the feedback information [16], which will complicate the design of a data broadcast system. Since the mobile devices are usually resource-limited, the cost of extra computation is very undesirable. Therefore, it is disputable to employ these two techniques. More supplements and related works will be reviewed in Section 2.

In pursuit of the traffic awareness, we devise a selective deferment and reflection technique (SDR) by exploiting the behavior of client impatience to collect and detect the dynamic workloads and access patterns on the broadcast channel. Basically, it is derived from the observation that a client usually has a limited patience for his push access [15]. That is, a client will alternatively send an *impatient (pull) request* for a push item when the waiting time exceeds his patience. By counting the impatient requests on the uplink channel, the server can calculate the access frequency proportion of push items on the broadcast channel. In addition, the server will deliberately generate a “single item” broadcast miss, which causes the clients to declare their demands for that item. The basic idea of the SDR technique is as follows. First, the SDR technique heuristically selects a push item and defers its broadcast for a certain time period. At the same time, the clients do not know this intended deferment and hence continue their waiting for this item. After the corresponding waiting time expires, the clients will take this instance as a single item broadcast miss and subsequently change their access to this item from the push mode to the pull mode. Then, the server can count the exact access frequency for this item and further use it as the *reflective base* to estimate the access frequency in accordance with the relative proportion of the number of impatient requests for each push item. On the other hand, the exact access frequency of each cold item is directly available by counting the regular pull requests. Consequently, with the combination of the access frequencies of hot and cold items, the SDR technique can attain the estimated access frequency distribution, thereby providing the traffic awareness required to the server.

As mentioned, since the increased traffic due to a broadcast miss is confined to the range of a single deferred item, we comment that the SDR technique can avoid the scalability problem in the probing technique, and can provide an opportunity to serve as an on-line traffic awareness procedure. Within a deferment period, the server is able to calculate the access frequencies of all items simultaneously and adjust the broadcast contents dynamically for the next broadcast program so as to achieve better performance. This feature distinguishes this paper from others. Note that adaptiveness is either absent or sensitive to a dedicated traffic factor in the previous works. With quantitative analyses, the SDR technique is shown to be amenable to dynamic traffic awareness.

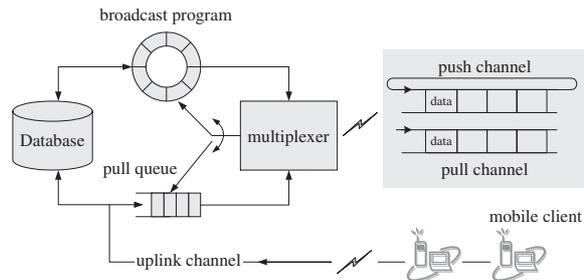


Figure 1. A hybrid data delivery model.

To provide more insights into the SDR technique, we implement a system simulator with a variety of dynamic traffic factors and conduct several experiments for performance studies. It is shown by the experimental results that given an increasing or decreasing workload with a static access pattern, the real access frequency distribution is bounded by two estimated distributions which are calculated with two specific reflective bases. In view of this, we further devise a trigonometric tuning method to optimize the estimated access frequency distribution. In addition, we find that the SDR technique with a suggested reflective base is able to generate an access frequency distribution very close to the real one, showing the feasibility and reliability of our proposed data broadcast mechanism with traffic awareness.

The rest of the paper is organized as follows. Section 2 first models the data broadcast environment and gives some supplements and related works. Section 3 studies the SDR technique with its analysis and discussions. Section 4 describes the simulation model and demonstrates the experimental results. This paper concludes with Section 5.

2 Preliminary

Section 2.1 presents a hybrid data delivery model where the traffic awareness is considered. Section 2.2 reviewed some prior studies with their problems to this model.

2.1 A Hybrid Data Delivery Environment

As depicted in Figure 1, the broadcast bandwidth is partitioned into an upward channel and a downward channel. The former delivers clients’ pull requests, and the latter corresponds to a series of interleaved data slots of equal size. Data slots are further classified as the push or pull mode. Logically, we view the push slots as a push channel and the pull slots as a pull channel, and these slots are multiplexed into a downward channel. The wireless information server contains a database including all broadcast items which are classified as hot or cold ones according to their previous access frequencies. The server broadcasts hot items over push slots periodically and delivers cold items by pull slots in re-

sponse to clients' explicit access requests from the uplink channel. The assumptions adopted in this study are as follows: (1) the bandwidth of an uplink channel is constant, (2) each item is self-identified and read-only, (3) a data slot is either in the pull or push mode, and can alternatively be switched, (4) the broadcast cycle is tunable, (5) the data classification policy is based on their relative access frequencies, and (6) a client will submit an impatient pull request for a hot item when his waiting time has expired. This model is similar to those in [3][6][18], while having a graceful extension for the design of a general traffic awareness mechanism.

2.2 Related Work and Supplement

In a push delivery, the access time, i.e., a half of the broadcast cycle on average, is dominated by the number of items in the database, whereas a pull response time in a pull delivery can be realized by an M/M/1 queueing model. Rather, a hybrid delivery is called for to strike a compromise among the trade-offs; however, the adaptiveness of data classification and bandwidth allocation against dynamic traffic changes is a critical challenge. Without the traffic awareness mechanism, a hybrid delivery is not reliable. In the previous researches, few works were adaptive to dedicated traffic factors. In [3], the slot allocation and the contents of push and pull sets are static. The work in [18] firstly applied the broadcast miss, the original probing technique, to calculate the access statistics, but might suffer from the scalability problem. In contrast, the work in [6][10] used the feedback/piggyback technique to append the access statistics information into the clients' regular pull requests, which complicates the design of a data broadcast system. As for broadcast scheduling, the work in [21] devised an on-line approach to pick the next broadcast item and also applied an off-line algorithm to generate a fixed-length broadcast program. In [19], a priority index policy is proposed to select the next broadcast item. Other on-line scheduling works in [4][20] considered the number of pull requests and the duration time from the last broadcast, to determine the most "profitable" broadcast item. In addition, the work in [23] presented a statistic *maximum likelihood estimation* to estimate the effectiveness of static scheduling. Finally, the notion of "impatient user" was originally addressed in [15] for broadcast scheduling, but only to improve the service ratio. Comparatively, our work exploits the "client's impatience" to estimate the access frequency distribution. It is noteworthy that we aim at the design of a traffic awareness technique with all-inclusive traffic factors to offer a foundation for broadcast scheduling, indexing and the hybrid data delivery.

3 SDR Traffic Awareness

The abstraction of the SDR traffic awareness mechanism consists of the adaptive module, the pull procedure and the

push procedures. The adaptive module can dynamically adjust data classification and bandwidth allocation. The pull and push procedures are mutual and responsible for the collections of the traffic information. The pull procedure responds to the pull requests for cold data; on the other hand, the push procedure maintains a broadcast program including all hot data. For a push access, a client can submit alternatively an impatient pull request when the waiting time exceeds his patience. Hence, the server has the exact number of regular pull requests for each cold item and the number of impatient requests for each hot item. Furthermore, the server defers the broadcast of a selected item purposely for a broadcast cycle; meanwhile, the pull procedure can count the exact number of requests for this deferred item. Accordingly, the reflection estimation can calculate the access frequencies of hot items. Consequently, with access frequencies of cold and hot items, the server can be aware of traffic changes and perform adaptive data broadcast.

3.1 Notation and Premise

Time is *slotted* equally and each time slot t_u is equal to an item slot interchangeably. The server has a database D that contains a number of $m = |D|$ data items of equal size s . Let the request arrival rate λ_i of each item d_i form a Poisson process with an aggregate arrival rate $\lambda = \sum_{i=1}^m \lambda_i$ where $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_m$ in a t_u . After data classification, the pull item set U^c includes cold items d_1, d_2, \dots, d_k and the push item set U^h includes hot items $d_{k+1}, d_{k+2}, \dots, d_m$. The aggregate arrival rates are $\lambda^c = \sum_{i=1}^k \lambda_i$ and $\lambda^h = \sum_{i=k+1}^m \lambda_i$ respectively. The length of a broadcast cycle L is the number of data slots used to deliver all items in U^h at least once according to a specific scheduling policy. To maintain the generality, the SDR applies a *flat* broadcast program \mathcal{P} for a fundamental basis in comparison with others. Thus, L is equal to the number of hot items, $m - k$, in U^h in a hybrid data delivery. In addition, we assume that the index of a broadcast program \mathcal{P} can be transmitted to the clients ahead of the push data either through the same downward channel or a distinct channel, so that a client can decide to access an item by the pull/push mode. When a client wants to access an item on the push channel, this client has a patience ω in waiting for this item. The number of impatient requests for item d_i is denoted as γ_i in a time unit, and its aggregate in a broadcast cycle is γ_i^L . In practice, the total number of the accumulative impatient requests is independent of whether an index is provided or not.

3.2 Pull Access Calculation

During a broadcast cycle, in a t_u , the server can count the exact access frequency λ_i of each d_i in U^c from an uplink channel. Thus, the sum of access requests for d_i in L is $\lambda_i^L = \sum_{1 \leq j \leq L} \lambda_i(j)$ where j indicates the j -th slot in the

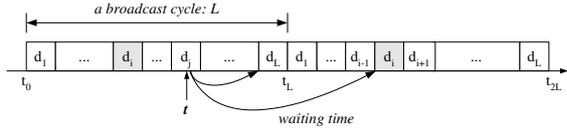


Figure 2. The client impatience with his impatient request.

broadcast cycle, and the mean access frequency for d_i is

$$\bar{\lambda}_i = \sum_{1 \leq j \leq L} \lambda_i(j)/L = \lambda_i^L/L, \quad 1 \leq i \leq k, \quad (1)$$

where j indicates the j -th slot in L . Moreover, the ratio of mean access frequency among pull items is

$$\bar{\lambda}_1 : \dots : \bar{\lambda}_i : \dots : \bar{\lambda}_k = \lambda_1^L/L : \dots : \lambda_i^L/L : \dots : \lambda_k^L/L,$$

where $d_i \in U^c$ and $1 \leq i \leq k$. Moreover, the total pull access frequency in a L is $\sum_{1 \leq i \leq k} \sum_{1 \leq j \leq L} \lambda_i(j)$.

3.3 Push Access Estimation

This subsection presents an explicit procedure to estimate the access frequencies of push items by three phases: (1) modeling the behavior of client impatience, (2) the selective deferment, and (3) the reflective estimation.

3.3.1 Client Impatience Modeling

Since each client has a patience in waiting for the arrival of a push item in the broadcast program \mathcal{P} , the client will submit an impatient request for that item over the uplink channel if the waiting time exceeds his patience. Let a random variable x present a client's patience and be independent of other clients' patience. As x is an exponential distribution with a mean patience ω , the probability density function of x is given by $f_\gamma(x) = \frac{1}{\omega} e^{-\frac{x}{\omega}}$ where $0 \leq x < \infty$, and the distribution function of x is $F_\gamma(x) = 1 - e^{-\frac{x}{\omega}}$ where $0 \leq x < \infty$, or otherwise $F_\gamma(x) = 0$.

Figure 2 illustrates the behavior of a client's push access. There are two possible situations: (1) the item will be broadcast later in this broadcast cycle, (2) the item has been broadcast in this broadcast cycle so that the client has to wait until its broadcast in the next broadcast cycle. Given that a broadcast cycle includes L slots and the arrival of a client with interest in item d_L is located in the j -th slot, this client has to wait $(L-1) - j + \frac{1}{2}$ slots where $\frac{1}{2}$ indicates an average access delay in the j -th slot. However, with a push access for d_i , because item d_i has already been broadcast in this cycle, the client has to wait $L+i-j+\frac{1}{2}$ slots until the d_i 's broadcast in the next cycle. Accordingly, the probability that a client will generate an impatient request for d_i can be derived as

$$\mathcal{F}_\gamma(i, j) = \begin{cases} 1 - e^{-\frac{1}{\omega}(i-j-\frac{1}{2})}, & i > j, \\ 1 - e^{-\frac{1}{\omega}(L+i-j-\frac{1}{2})}, & i \leq j, \end{cases} \quad (2)$$

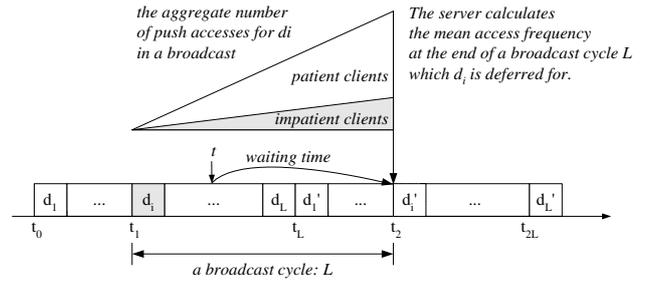


Figure 3. The selective and deferment reflection (SDR).

where $1 \leq i, j \leq L$, the item is scheduled in the i -th slot, and the client starts to access at j -th slot. Let $\lambda_i(j)$ be the access rate for d_i in the j -th slot. Then, the arrival rate of impatient requests for d_i is given as

$$\gamma_i(j) = \lambda_i(j) \cdot \mathcal{F}_\gamma(i, j), \quad (3)$$

and the aggregate of γ_i in a broadcast cycle is

$$\gamma_i^L = \sum_{1 \leq j \leq L} \lambda_i(j) \cdot \mathcal{F}_\gamma(i, j). \quad (4)$$

We observe that if every client has an equal expected patience ω , the summation of the impatient probabilities in a broadcast cycle is the same for each push item. Hence, γ_i^L is in proportion to $\sum_{1 \leq j \leq L} \lambda_i(j)$. In addition, despite of not knowing various $\lambda_i(j)$, instead of resolving the λ_i , the server can calculate the mean of push access rate at the end of a broadcast cycle since γ_i^L is available from the uplink channel. We can have

$$\gamma_i^L = \sum_{1 \leq j \leq L} \bar{\lambda}_i \cdot \mathcal{F}_\gamma(i, j) = \bar{\lambda}_i \cdot (L - e^{-\frac{L^2}{2\omega}}), \quad (5)$$

where $(L - e^{-\frac{L^2}{2\omega}})$ is a constant in a broadcast cycle. Furthermore, it is important to note that the ratio of the relative access frequencies of push items is equal to the ratio of the numbers of their impatient requests in a broadcast cycle.

$$\gamma_{k+1}^L : \dots : \gamma_i^L : \dots : \gamma_m^L = \bar{\lambda}_{k+1} : \dots : \bar{\lambda}_i : \dots : \bar{\lambda}_m, \quad (6)$$

where $k+1 \leq i \leq m$ and $d_i \in U^h$.

3.3.2 Selective Deferment

The ratio in Equation (6), is not informative enough for the server to perform adaptation to comply with the traffic changes. In that case, the server employs a selective deferment to produce useful information for further reflective estimation, rather than to evaluate ω and $\lambda_i(j)$ directly. The server intentionally makes a temporary "single item" broadcast miss, compelling the clients to disclose their interests in push data. Explicitly, the server selects a single item and defers it for a broadcast cycle. Whether the interested clients

are patient or not, they will submit pull requests. Then, the server can calculate the access frequency of this item. Figure 3 illustrates the selective deferment procedure. According to the “equal spacing” property that for an item d_i , the interval s_i in two consecutive broadcasts is fixed and equal to s_j where $d_i \neq d_j$ and d_i and d_j are in the same \mathcal{P} . The interval $[t_0, t_L]$ is equal to the broadcast cycle L . At t_1 the server selects an item d_i and deliberately defers its broadcast in the next broadcast cycle $[t_L, t_{2L}]$. Since the clients are not aware of the deferment, for those clients who are interested in d_i and arrive before t_1 , they are satisfied in this broadcast cycle. After t_1 , all clients interested in d_i will be satisfied at t_2 , except some clients whose patience is less than the waiting time $t_2 - t_1$. In $[t_1, t_2]$, by Equation (3) the server will receive a number of $\lambda_i(j) \cdot \mathcal{F}_r(i, j)$ impatient requests for d_i and thus γ_i^L is available at t_2 . Moreover, when those patient clients find that d_i is absent at t_2 , and they will submit regular pull requests for d_i . Therefore, the server can obtain the exact access frequency λ_i^L of d_i by adding γ_i^L and the number of the regular pull requests for d_i .

3.3.3 Reflection Estimation

In $[t_1, t_2]$, d_i is selectively deferred as a reflective base, and simultaneously other impatient frequencies of push data $\langle \gamma_{k+1}, \gamma_{k+2}, \dots, \gamma_m \rangle$ can be available similarly. From Equation (6) with λ_i^L , the SDR is able to estimate the access frequency of each push item reflectively as follows.

$$\bar{\lambda}_x = \frac{\gamma_x}{\gamma_i} \cdot \frac{\lambda_i^L}{L} = \frac{\gamma_x}{\gamma_i} \cdot \bar{\lambda}_i, \quad k+1 \leq x \leq m. \quad (7)$$

By Equations (1) and (7), the server can further obtain the dynamic access frequency distribution $\Phi(x)$ of all data in a broadcast cycle as follows.

$$\langle \bar{\lambda}_1, \dots, \bar{\lambda}_k, \frac{\gamma_{k+1}}{\gamma_i} \cdot \bar{\lambda}_i, \dots, \frac{\gamma_{i-1}}{\gamma_i} \cdot \bar{\lambda}_i, \bar{\lambda}_i, \frac{\gamma_{i+1}}{\gamma_i} \cdot \bar{\lambda}_i, \dots, \frac{\gamma_m}{\gamma_i} \cdot \bar{\lambda}_i \rangle,$$

where $\bar{\lambda}_i = \Phi(i)$ is the mean access frequency of the reflective base. In addition, the estimated mean \overline{W}_{load} of traffic workload is $\sum_{\forall d_i \in U^c} \bar{\lambda}_i + \sum_{\forall d_i \in U^h} \frac{\gamma_x}{\gamma_i} \cdot \bar{\lambda}_i$.

3.4 SDR Analysis and Characteristic

We present several characteristics of the SDR technique with their theoretical and quantitative analyses under various traffic conditions. Without the loss of generality, the SDR technique assumes neither that the workload is in proportion to client population nor that the access pattern is static, in comparison with other previous works. First, Theorem 1 shows the accuracy of the SDR technique under a static traffic, and Lemma 1 presents a supplement similar to other works. Then, the other theorems and lemmas are derived to investigate the SDR technique with dynamic traffic changes. Proofs of theorems are omitted in this paper to save the page space.

Theorem 1 Given a static traffic, the estimated access frequency distribution $\Phi(x)$ by the SDR is equivalent to the real distribution $\Psi(x)$.

Lemma 1 Given a \mathcal{P} with a mean patience ω , γ_i^L is determined by the client population if the access probability p_i for an item d_i is static and a client can access a push item only if this client has no pending push access.

Theorem 2 Let \mathcal{P} be a flat program with a length of L . If λ_i in U^h is increasing in ascending order. As d_1 , the first item in \mathcal{P} , is selected as the reflective base, then $\Phi_1(i) \gtrsim \Psi(i)$ with $2 \leq i \leq L$.

Theorem 3 Let \mathcal{P} be a flat program with a length of L . If λ_i in U^h is decreasing and d_i is the reflective base, $\Phi_i(x) \lesssim \Psi(x)$ with $1 \leq x \leq L$.

Theorem 4 Given d_i is the reflective base in \mathcal{P} , $\Phi_i(x)$ has at least a cross with $\Psi(x)$ where $\Phi_i(k) = \Psi(k)$ and $k = i$.

With a prior knowledge of the access frequency distribution and the workload, the server can determine the item whose λ_i^L / γ_i^L is the smallest as the reflective base. Lemma 2 provides a guideline for the SDR technique to select the reflective base. In the next section, we will compare the estimated results with Lemma 2 and Property 1.

Lemma 2 Let \mathcal{P} be flat with a length of L . If the server selects d_i , whose λ_i^L / γ_i^L is the smallest, to be the reflective base, then $\Phi_i(x)$ has a cross with $\Psi(x)$ at the $x = i$. Furthermore, we can have $\lambda_i^L = \Phi_i(i) \lesssim \Psi(i)$ if λ_i is continuously increasing during this broadcast cycle. In contrast, $\lambda_i^L = \Phi_i(i) \gtrsim \Psi(i)$ if λ_i is continuously decreasing during this broadcast cycle.

Property 1 Given that \mathcal{P} is flat, the item of the smallest λ_i^L / γ_i^L is either the middle item $d_{\lceil \frac{L}{2} \rceil}$ or an item very close to $d_{\lceil \frac{L}{2} \rceil}$ if \mathcal{P} is scheduled in an order by the access frequency, or otherwise the server can sort these items by their respective numbers of impatient requests, and take $d_{\lceil \frac{L}{2} \rceil}$ as the reflective base if \mathcal{P} is scheduled randomly.

3.5 Trigonometric Tuning SDR Estimation

According to Properties 2 and 3, it is interesting that “weighting $\bar{\lambda}'_i$ and $\bar{\lambda}''_i$ ” can possibly improve the estimation. By Theorems 2, 3 and 4, in an ascending/descending workload with a constant access pattern, $\Phi_1(i)$ can be slightly higher/lower than $\Psi(i)$ correspondingly, whereas $\Phi_{\lceil \frac{L}{2} \rceil}(i)$ can be slightly lower/higher than $\Psi(i)$. Hence, we further devise a tuning method based on the semi-monotonic feature of a trigonometric function to enhance the SDR estimation.

Property 2 We can narrow the inaccuracy of the estimation of $\bar{\lambda}_i$ to a small range of a distance between $\bar{\lambda}_i$ and $\bar{\lambda}_i'$ estimated reflectively by deferring d_1 and $d_{\lceil \frac{L}{2} \rceil}$ respectively.

Property 3 Given an increasingly or decreasingly dynamic workload with a static access pattern, the average of $\Phi_1(x)$ and $\Phi_{\lceil \frac{L}{2} \rceil}(x)$, denoted by $\bar{\Phi}(x)$, has two cross points with $\Psi(x)$ if \mathcal{P} is scheduled in order of the access frequency.

With a \mathcal{P} , we can transform a series of item slots in a L into the range of $[0, \pi]$ in radians subsequently. Each slot is then mapped to an angle by a one-to-one mapping. For instance, the angle of d_1 is 0, the angle of $d_{\lceil \frac{L}{2} \rceil}$ is $\frac{\pi}{2}$ and for other d_i in \mathcal{P} , its angle δ_x is $x \cdot \frac{\pi}{L}$ where $1 \leq x \leq L$. Given that $\Phi_{FI}(x)$ is the estimation distribution by using d_1 as the reflective base and $\Phi_{MI}(x)$ is reflected by the middle item $d_{\lceil \frac{L}{2} \rceil}$, we can have a tuned distribution $\Phi_{\Delta}(x)$ as

$$\Phi_{\Delta}(x) = \cos^2 \delta_x \cdot \Phi_{FI}(x) + \sin^2 \delta_x \cdot \Phi_{MI}(x), \quad (8)$$

where $\cos^2 \delta_x + \sin^2 \delta_x = 1$. We use sin and cos functions because the value of sin function is gradually up to 1 with the argument from 0 to $\frac{\pi}{2}$, whereas a cos function has an inverse trend from 1 to 0. Because $\Phi_{MI}(x)$ has a cross with $\Psi(x)$ at $d_{\lceil \frac{L}{2} \rceil}$, we apply sin to $\Phi_{MI}(x)$. Likewise, we apply cos function to $\Phi_{FI}(x)$ whose cross point with $\Psi(x)$ is at d_1 . In light of this, we can tune the estimated value within the range between $\Phi_{FI}(x)$ and $\Phi_{MI}(x)$ with two cross points at the point $\delta_x = \frac{\pi}{2}$ where $\sin \delta_x = 1$ and $\cos \delta_x = 0$ with $\Phi_{\Delta}(\frac{L}{2}) = \Phi_{MI}(\frac{L}{2}) = \Psi(\frac{L}{2})$ and at the point $\delta_x = 0$ with $\Phi_{\Delta}(1) = \Phi_{FI}(1)$. Furthermore, we will investigate Equation (8) and present the simulation results in Section 4.

4 Simulation and Results

Section 4.1 describes the simulation environment, including traffic generation, candidates of the reflective bases, accuracy measure of estimation, etc. Subsequently, the SDR technique is inspected by various traffic scenarios and the simulation results are summarized in respective subsections.

4.1 Simulation Model

Let the discrete space R_z of the one-to-one mapping comprise λ_x s of all items in U^h . Then, we generate R_z by the Zipf distribution with a skew coefficient θ where $\lambda = \sum_{d_x \in R_z} \lambda_x$ and $\lambda_x = \lambda \cdot p_x = \lambda \cdot (\frac{1}{x})^\theta / \sum_{1 \leq x \leq L} (\frac{1}{x})^\theta$. In this simulation, we design two dynamic traffic generators, *partially dynamic traffic* and *fully dynamic traffic*. The former is the general case where the workload increases/decreases gradually or the access frequencies of some items vary in ascending/descending order. The latter can be viewed as the bursty traffic where the access pattern

Table 1. Simulation parameters description

notation	meaning	value
m	the number of items in \mathcal{P}	50 ~ 1000
λ	access frequency in a time unit	5 ~ 30
θ	Zipf's skew coefficient	+2.0 ~ -5.0
ω	client patience time	50 ~ 3000
L	the length of a broadcast cycle	50 ~ 1000

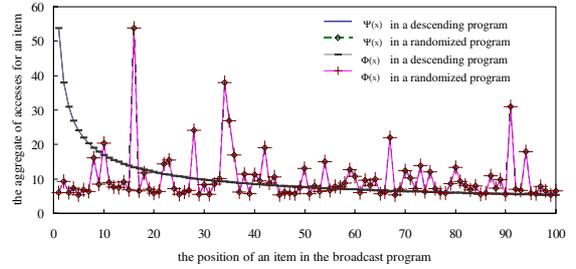


Figure 4. The estimation by the SDR under a static traffic.

in each slot is assigned arbitrarily by a Zipf distribution with a range of skew coefficients. Note that by the “equal space property,” we generate a broadcast program \mathcal{P} and schedule the items either in descending order by the access frequency or in an arbitrary order. With the ordered items in \mathcal{P} , three candidates for the reflective base are investigated in our study: (1) the first item (d_{FI}), scheduled in the broadcast program, (2) the middle item (d_{MI}), scheduled in the broadcast program, and (3) the last item (d_{LI}) in a broadcast program. To determine the reflective base, we utilize the difference mean and the variance to measure the estimation accuracy. Let Λ_x be the difference between the estimated and accurate access frequencies of an item d_x , and the mean Λ_x is $\mu = E(\Lambda_x) = \sum_{R_z} |\lambda_x^L - \gamma_x^L| / L$ and the variance is $\sigma^2 = E[(\Lambda - \mu)^2] = \sum_{R_z} (\Lambda_x - \mu)^2 / L$. Table 1 lists the simulation parameters.

4.2 Static Traffic

Figure 4 depicts the estimated curves by the SDR under a static traffic with $\omega = 200$, $L = 100$, $\lambda = 10$ and $\theta = 0.5$. Given a broadcast program \mathcal{P} of descending order, the SDR can obtain an estimated access frequency distribution $\Phi(x)$ which exactly matches the real $\Psi(x)$ for whichever item is selected as the reflective base. This is because γ_i^L is determined by the summation of $\mathcal{F}_\gamma(i, j)$ in L as λ_i is static. As ω is constant, γ_i^L is in proportion to λ_i^L . Comparatively, given a \mathcal{P} of random order, $\Phi(x)$ is also exactly the same with $\Psi(x)$. The results show that under a static traffic, the estimation by the SDR is regardless of the order by which \mathcal{P} is scheduled.

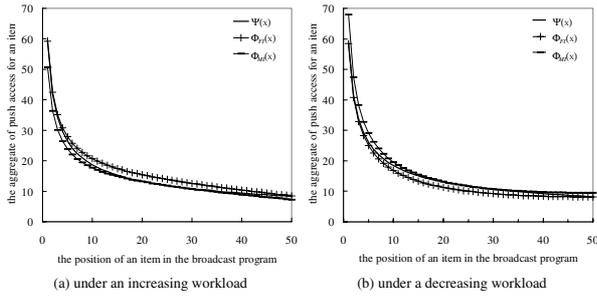


Figure 5. The real $\Psi(x)$ versus the estimated $\Phi_{FI}(x)$ and $\Phi_{MI}(x)$, reflected by d_{FI} and d_{MI} .

4.3 Dynamic Workload & Trigonometric Tuning

Referring to the analyses in Section 3.5, we simulate the SDR with a downward/upward workload and various client patience, and evaluate the trigonometric tuning method.

Figure 5(a) depicts the result with an increasing workload from $\lambda = 10$ to $\lambda = 20$, $\theta = 0.5$, $L = 50$, and $\omega = 100$ in a broadcast cycle. When d_{FI} is the reflective base, the estimated $\Phi_{FI}(x)$ is slightly higher than the real $\Psi(x)$ and has the maximal distance between-in at the middle item. Oppositely, $\Phi_{MI}(x)$ is slightly lower than $\Psi(x)$ and has a cross at d_{MI} when the server takes d_{MI} as the reflective base. The reason is that for an item d_i which has been broadcast in this cycle, an earlier interested client has a higher probability to submit an impatient request for d_i than another interested client who arrives later. Even though λ increases gradually, the probability of submitting an impatient request for d_i is lower in the end of the broadcast cycle because of a shorter waiting time. In addition, it is noted that the item d_{LI} in the end of a broadcast program faces the same circumstance with d_{FI} in that a broadcast program is cyclic. Inductively, this circumstance is symmetric with d_{MI} as the symmetric point. In view of this, we can understand that if an item is scheduled earlier than d_{MI} , the rate of its push accesses to impatient requests can be more than those of other items scheduled lately. Comparatively, Figure 5(b) illustrates the result with a decreasing workload from $\lambda = 20$ to $\lambda = 10$. Inversely, $\Phi_{FI}(x)$ is lower than $\Psi(x)$, whereas $\Phi_{MI}(x)$ is up $\Psi(x)$ with a cross point at d_{MI} . This is because that for an item that has been broadcast, the earlier a client arrives, the higher probability this client will submit an impatient request. Since λ decreases gradually, the rate of λ_{FI}^L to γ_{FI}^L is the smallest in a broadcast cycle and the estimated number of accesses for another item is lower than its real number. In contrast, $\Phi_{MI}(x)$ is mostly larger than $\Psi(x)$.

Noticeably, Figure 5 demonstrates that $\Psi(x)$ is bounded by $\Phi_{FI}(x)$ and $\Phi_{MI}(x)$ with an increasing/decreasing workload if the access pattern does not change. According

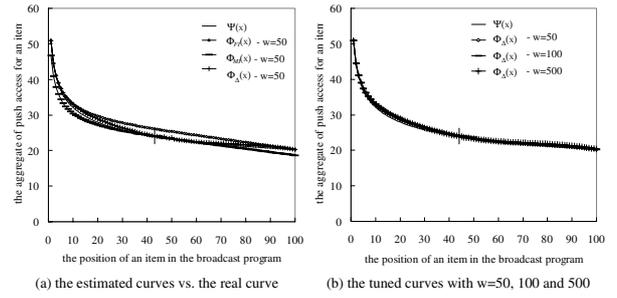


Figure 6. The real $\Psi(x)$ versus the estimated $\Phi_{\Delta}(x)$ by the trigonometric tuning.

to Equation (8), we can further obtain a tuned estimation $\Phi_{\Delta}(x)$. Figure 6 displays the synthetic comparison among $\Phi_{FI}(x)$, $\Phi_{MI}(x)$, $\Phi_{\Delta}(x)$ and $\Psi(x)$ with λ from 20 to 30, $L = 100$ and $\theta = 0.2$. As displayed in Figure 6(a), with a mean difference 0.603, $\Phi_{MI}(x)$ is relatively closer to $\Psi(x)$ than $\Phi_{FI}(x)$ which is above $\Psi(x)$ with a mean difference 1.217. However, after the trigonometric tuning, we have $\Phi_{\Delta}(x)$ very close to $\Psi(x)$ with a mean difference 0.353. Figure 6(b) depicts the results by the trigonometric tuning versus various ω under a dynamic workload. By Equation (7), although $\Phi(x)$ by the SDR with a less information of client impatience can slightly deviate from $\Psi(x)$. However, in this case even with a large ω , the estimated result $\Phi_{\Delta}(x)$ is still close to the real distribution $\Psi(x)$, showing very good stability of the SDR technique.

4.4 Dynamic Access Pattern

Table 2 lists the experimental results with the suggested reflective bases. We examine d_{MI} under a heavy workload with 2000 accesses in a broadcast cycle. In most cases, d_{MI} is suggested as the reflective base and the corresponding mean difference is smaller than 1, except few extreme cases with excessive changes of access patterns. For those cases in the sides of the diagonal, their mean differences are about 0.2, meaning that the difference between the real and the estimated aggregate of push accesses in a broadcast cycle is only about 40. Although there are few cases whose reflective bases are suggested with the d_{LI} , we find that their mean differences are very close to those by d_{MI} . Therefore, we observe that the mean difference will be larger if the item of a larger change of access frequency is selected as the reflective base.

5 Conclusions

In this paper, we have devised a novel selective deferment and reflection (SDR) technique which can be aware of the dynamic traffic changes. Compared to prior probing

Table 2. The suggested reflective bases, where $L = 200$, $\lambda = 10$, $\omega = 200$ and θ from -5 to 1 .

θ	-5	-4	-3	-2	-1	0.2	0.4	0.6	0.8	1.0
-5	-	LI	LI	LI	MI	MI	MI	MI	MI	MI
-4	LI	-	LI	MI	MI	MI	MI	MI	MI	MI
-3	LI	LI	-	MI	MI	MI	MI	MI	MI	MI
-2	LI	MI	MI	-	MI	MI	MI	MI	MI	MI
-1	MI	MI	MI	MI	-	MI	MI	MI	MI	MI
0.2	MI	MI	MI	MI	MI	-	MI	MI	MI	MI
0.4	MI	MI	MI	MI	MI	MI	-	MI	MI	MI
0.6	MI	MI	MI	MI	MI	MI	MI	-	MI	MI
0.8	MI	MI	MI	MI	MI	MI	MI	MI	-	MI
1.0	MI	MI	MI	MI	MI	MI	MI	MI	MI	-

and feedback approaches, our design is of low complexity and light-weight without degrading the corresponding performance. We have conducted performance analysis with various dynamic traffic scenarios. It has been shown by the experimental results that the estimated access frequency distribution is very close to the real one. Furthermore, we have found that with an increasing/decreasing workload, if the access pattern is static, the real access frequency distribution is bounded by two specific estimated distributions with d_{FI} and d_{MI} as the reflective bases. In view of this, we have devised a trigonometric tuning method to further optimize the estimation. Consequently, our proposed SDR traffic awareness mechanism is very feasible for dynamic data broadcasting to cope with the nature of a changing traffic.

References

- [1] S. Acharya, M. J. Franklin, and S. B. Zdonik. Prefetching from broadcast disks. In *Proceedings of IEEE ICDE'96*, pages 276–285, February 1996.
- [2] S. Acharya, et al. Broadcast disks: Data management for asymmetric communications environments. In *Proceedings of ACM SIGMOD'95*, pages 199–210, May 1995.
- [3] S. Acharya, et al. Balancing push and pull for data broadcast. In *Proceedings of ACM SISMOD'97*, pages 183–194, May 1997.
- [4] D. Aksoy and M. Franklin. Rxw: A scheduling approach for large-scale on-demand data broadcast. *IEEE/ACM Transactions on Networking*, 7(6):846–860, 1999.
- [5] D. Barbará. Mobile computing and database - a survey. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):108–117, 1999.
- [6] J. Cai and K.-L. Tan. Tuning integrated dissemination-based information systems. *Data & Knowledge Engineering*, 30(1):1–21, 1999.
- [7] M.-S. Chen, P. S. Yu, and K.-L. Wu. Indexed sequential data broadcasting in wireless mobile computing. In *Proceedings of IEEE ICDCS'97*, pages 124–131, 1997.
- [8] J. Fernandez-Conde and K. Ramamritham. Adaptive dissemination of data in time-critical asymmetric communication environments. *to appear in ACM/Baltzer Mobile Networks and Applications*.
- [9] S. Hameed and N. H. Vaidya. Efficient algorithms for scheduling data broadcast. *ACM/Baltzer Wireless Networks*, 5(3):183–193, 1999.
- [10] Q. Hu, D. L. Lee, and W.-C. Lee. Dynamic data delivery in wireless communication environments. In *Proceedings of ER'98 Workshops on Mobile Data Access*, volume 1552 of LNCS, pages 218–229. Springer, 1998.
- [11] Q. Hu, W.-C. Lee, and D. L. Lee. Indexing techniques for wireless data broadcast under data clustering and scheduling. In *Proceedings of ACM CIKM'99*, 1999.
- [12] T. Imielinski and B. R. Badrinath. Mobile wireless computing: Challenges in data management. *Communications of ACM*, 37(10):18–28, 1994.
- [13] T. Imielinski and S. Viswanathan. Adaptive wireless information systems. In *Proceedings of the ACM Special Interest Group on DataBase Systems*, pages 19–41, 1994.
- [14] T. Imielinski, S. Viswanathan, and B. R. Badrinath. Data on air: Organization and access. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):353–372, May/June 1997.
- [15] S. Jiang and N. H. Vaidya. Scheduling data broadcast to "impatient" users. In *Proceedings of ACM MobiDE'99*, pages 52–59, August 1999.
- [16] J. Jing, O. Bukhres, A. K. Elmargamid, and R. Alonso. Bit-sequences: A new cache invalidation method in mobile environments. *ACM/Baltzer Mobile Networks and Applications*, 2(2), 1997.
- [17] W.-C. Peng and M.-S. Chen. Dynamic generation of data broadcasting programs for a broadcast disk array in a mobile computing environment. In *Proceedings of ACM CIKM-2000*, November 2000.
- [18] K. Stathatos, et al. Adaptive data broadcast in hybrid networks. In *Proceedings of VLDB'97*, pages 326–335, August 1997.
- [19] C.-J. Su, et al. Broadcast scheduling for information distribution. *ACM/Baltzer Wireless Networks*, 5(2):137–147, 1999.
- [20] W. Sun, et al. A self-adaptive scheduling algorithm of on-demand broadcasts. In *Proceedings of the 4th ACM MSWIM'01*, pages 139–145, July 2001.
- [21] N. H. Vaidya and S. Hameed. Scheduling data broadcast in asymmetric communication environments. *ACM/Baltzer Wireless Networks*, 5(3):171–182, 1999.
- [22] K.-L. Wu, P. S. Yu, and M.-S. Chen. Energy-efficient caching for bandwidth-limited wireless mobile computing. In *Proceedings of IEEE ICDE'96*, February 1996.
- [23] J. X. Yu, T. Sakata, and K.-L. Tan. Statistical estimation of access frequencies in data broadcasting environments. *ACM/Baltzer Wireless Networks*, 22(6):89–98, 2000.