ARTICLE IN PRESS

Available online at www.sciencedirect.com
ScienceDirect
Pervasive and Mobile Computing ▮ (▮▮▮▮) ▮▮▮–▮▮▮
pervasive and mobile computing
www.elsevier.com/locate/pmc

Efficient data dissemination using locale covers☆

Sandeep Gupta*, Jinfeng Ni, Chinya V. Ravishankar

Department of Computer Science and Engineering, University of California, Riverside, CA 92521, United States

Received 26 February 2007; received in revised form 6 August 2007; accepted 7 November 2007

Abstract

Location-dependent data are central to many emerging applications, ranging from traffic information services to sensor networks. The standard pull- and push-based data dissemination models become unworkable since the data volumes and number of clients are high.

We address this problem using *locale covers*, a subset of the original set of locations of interest, chosen to include at least one location in a suitably defined neighborhood of any client. Since location-dependent values are highly correlated with location, a query can be answered using a location close to the query point. Typical closeness measures might be Euclidean distance, or a k-nearest neighbor criterion.

We show that location-dependent queries may be answered satisfactorily using locale covers. Our approach is independent of locations and speeds of clients, and is applicable to mobile clients.

We also introduce a nested locale cover scheme that ensures fair access latencies, and allows clients to refine the accuracy of their information over time. We also prove two important results: one regarding the greedy algorithm for sensor covers and the other pertaining to randomized locale covers for k-nearest neighbor queries.

Published by Elsevier B.V.

Keywords: Location-dependent data; Locale covers; Dissemination; Accuracy; Latency

☆ This paper is an extended version of [S. Gupta, J. Ni, C.V. Ravishankar, Efficient data dissemination using locale covers, in: Proceedings of the 2005 ACM CIKM International Conference on Information and Knowledge Management, Bremen, Germany, 2005].

* Corresponding author.
E-mail addresses: sandeep@cs.ucr.edu (S. Gupta), jni@cs.ucr.edu (J. Ni), ravi@cs.ucr.edu (C.V. Ravishankar).

1574-1192/$ - see front matter Published by Elsevier B.V.
doi:10.1016/j.pmcj.2007.11.001

1. Introduction

The growth in mobile and pervasive computing has led to growing interest in location-specific information services, in which mobile clients seek real-time data relevant to their current location. Unfortunately, a pull-based dissemination model is unworkable, since the typically large number of clients would overwhelm the server with pull requests. Push-based models are better, and broadcasting is already being used for this purpose. For example, XM Traffic & Weather [26] is a satellite-based service to providing real-time traffic flow and incident information, as well as weather data to vehicles. Similarly, Microsoft's Smart Personal Objects Technology (SPOT) [17] initiative aims to deliver a wide variety of location services, including traffic, and points of interest (gas stations, movie theaters, and so on), through an FM broadcast channel. However, data volumes tend to be large, so even broadcasting [13] must be used cautiously, since it serializes items. A client may have to wait a very long time before data of interest appears in the broadcast.

1.1. Challenges

We briefly discuss some challenges in accommodating mobility in such an environment.

1.1.1. High data volumes

The volume of data may be too high, given the available bandwidth, so that broadcasting all data may cause long broadcast cycles and client waiting time, as well as consume excessive bandwidth. Besides, longer waits require clients to remain in listen mode for longer, wasting battery power.

1.1.2. Mobility

Tracking each client's location may require high communication overhead, or even be impossible in applications such the satellite-based service Sirius [20] where bidirectional communication is not available. Consequently, scheduling-based approaches such as [22,1, 12,3] are infeasible in this environment, since they assume knowledge of the access patterns or requests of each mobile client.

1.1.3. Fairness and access latency

Say the locations of shopping centers are of interest to a large set of mobile clients, each of whom requires information about the shopping centers in its vicinity. Broadcasting data in arbitrary order causes access latencies to be as long as the full broadcast cycle. In some data dissemination models [1,22], clients communicate their access and mobility patterns to the server, which then schedules data item broadcasts to achieve the fairness among clients. However, it is impractical to require such communication, due to power limitations at clients, and their large number, or because the server is a satellite [20]. Our scheme allows all clients to obtain approximate but satisfactory data quickly.

1.1.4. Scalable push-based dissemination

Pull allows servers to remain stateless, but this model is inappropriate for our application scenarios. First, the huge number of clients would overload the server with pull requests.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

Second, for many applications, such as the satellite-based service Sirius [20], bidirectional communication is not available for clients to send pull requests to the server. The push model is better, but requires servers to be stateful, and to keep track of what data is to be sent to which client, and at what time. Servers must also commit resources to initiating and maintaining connections, so that push does not scale well with the number of clients. Our approach allows scalable push.

1.2. Our contributions

We propose efficient data-dissemination schemes using the novel concept of *locale covers*, to address the challenges we have outlined. Given a criterion for closeness, a locale cover includes information about some site close to every client, regardless of its position.

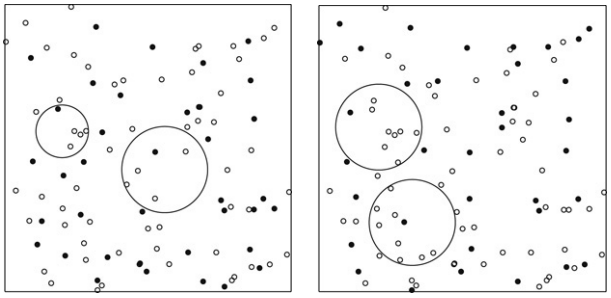
We also extend this idea to define *nested locale covers*: A nested locale cover is a set of locale covers $\{L_1, \ldots, L_m\}$ such that each site appears in some locale cover L_i. We give efficient and practical algorithms to find small size locale covers and nested locale covers under different closeness criteria, namely k-nearest neighbor and Euclidean distance. In this scheme the broadcast cycle is divided into several subcycles such that each subcycle guarantees to partially satisfy all clients and at the end of the broadcast the entire set is broadcasted.

In our schemes, the server needs no knowledge of the locations of clients, or their mobility and data-access patterns. The accuracy of the approximation can be controlled by defining the locale appropriately. Dissemination schemes using nested locale covers ensure fair access latencies. Nested locale covers allow clients to refine the accuracy of their information over time. However, it seems from our experiments that the nested locale covers are able to satisfy clients at a performance marginally lower than that of random broadcast. It is yet unclear to us whether all such schemes must suffer from lowered performance, or whether better schemes might exist than the one proposed here.

We make the following contributions.

(1) We present an $O(n^2k)$ algorithm for finding locale cover under k-nearest neighbor closeness criteria (Section 3.2). Previously known algorithms for this problem had $O(n^3)$ time complexity. We also give a partitioning based heuristic to improve its scalability (Section 3.3).
(2) We demonstrate through extensive experiments that the proposed technique is able to obtain locale covers of approximately $2n/k$ for the dataset used, regardless of distribution (Section 6.2).
(3) We solve the problem of locale cover under Euclidean closeness criteria by reducing it to a sensor cover problem [10] (Section 4.2.1).
(4) For the greedy algorithm of [10] we theoretically quantify the relationship between the size of cover and fraction of uncovered region (Section 4.2.2). This result directly characterizes the quality of locale cover obtained through a greedy algorithm.
(5) We present randomized algorithms to compute nested locale cover.
(6) We also give probabilistic guarantees on the quality of cover obtained by the proposed algorithm (Section 5).

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001



(a) 5 nearest sites. (b) 0.125-disk.

Fig. 1. Examples of locale covers.

Our notion of locale cover is very general and is likely to be applicable beyond the domain of data dissemination, for example, in spatio-temporal data mining, and approximate indices.

Fig. 1(a) and (b) depict instances of locale covers when locales are defined in terms of the k-neighbor and Euclidean distance criteria, respectively. In Fig. 1(a), the small circles (dark or transparent) represent a collection of one hundred sites distributed over a normalized unit square region. If a client's locale is set as its five nearest sites, the dark sites form a locale cover, since they include at least one of the five nearest sites for every point in the region. Fig. 1(b) shows the corresponding solution when locales are defined as disks of fixed size ($r = 0.125$).

The rest of the paper is organized as follows. Section 2 provides a brief overview of concepts developed in this work. Sections 3 and 4 illustrate techniques to obtain locale cover for the k-nearest neighbor and Euclidean distance. In Section 5, we discuss nested locale covers for each of these metrics. Finally, in Section 6, we provide experimental results showing that our techniques are efficient and result in small locale covers.

2. Our approach: Locale covers

Central to our approach is the notion of locale cover, an idea interesting in its own right.

2.1. Locale covers

If P is some predicate defining closeness, the *P-locale* of a point $p \in \mathbb{R}^2$ is the set of all points q so that $P(p, q)$ holds. Given a region $R \subset \mathbb{R}^2$, a set of "sites" $X = \{s_1, s_2, \ldots, s_n\} \subseteq R$, and a proximity predicate P, the *P-locale cover* of R with respect to X is a subset $L \subseteq X$, so that, for any $p \in R$, $P(p, s)$ holds for at least one site $s \in L$. Thus, L covers all possible locales in R.

To illustrate this concept, consider the following game between two players A and B. A knows a set X of n points on the plane. Two points are neighbors if they are sufficiently close, by some agreed-upon criterion. Player B has limited memory, and can store only a subset of X. Player A picks a point $p \in \mathbb{R}^2$, and challenges B to produce at least one neighbor of p. For B to always win, it must store at least the locale cover of X.

A locale cover will depend on the region, the set of sites, and the closeness criterion, *but, surprisingly, not on the query point itself.* It is this feature that allows our dissemination

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

scheme to service arbitrarily large number of mobile clients. Also, as we shall demonstrate in this paper, locale covers of small size can be obtained even for stringent closeness criteria. Hence each dissemination cycle is of small duration, thereby reducing the access latency, with just marginal error in the accuracy of the client's information.

2.1.1. Definitions

Given a set of n sites $X = \{s_1, s_2, \ldots, s_n\}$, we now define the locale coverage problem under two criteria for closeness: k-nearest neighbor (k-NN) and Euclidean distance.

Definition. The *compass* $\widehat{R}$ of a region R is the set of sites whose locations are in R or on its boundary.

For example, in Fig. 1(a) and (b), the compass of each disk is the set of sites within the disk or on its boundary

Definition. A *k-domain* D is a disk with $|\widehat{D}| = k$. That is, its compass has cardinality k.

For example, each of the two disks in Fig. 1(a) defines a 5-domain, since each disk has exactly 5 sites within it.

Definition. An *r-disk* D is a disk of fixed radius r.

Next we define locale covers under different notions of proximity.

Definition (*k-Domain Locale Cover*). $L \subseteq X$ is a k-domain locale cover if $\widehat{D} \cap L \neq \emptyset$ for all k-domains D.

For example, the set of dark sites in Fig. 1(a) is a 5-domain locale cover; it includes at least one of the five nearest sites for every point in this region.

Definition (*r-Disk Locale Cover*). $L \subseteq X$ is a r-disk locale cover if $\widehat{D} \cap L \neq \emptyset$ for all r-disks D.

For example, the set of dark sites in Fig. 1(b) forms a 0.125-disk locale cover; it contains at least one site from within *any* non-empty 0.125-disk that one may draw in the region.

2.2. Nested locale covers

In some data dissemination models [1,22], clients communicate their access and mobility patterns to the server, which then schedules data item broadcasts so as to minimize the variance in access latency among clients. However, it is impractical to require such communication, due to the large number and power limitations at clients, or because the server is a satellite [20].

We introduce the concept of *nested locale covers* to ensure fairness with respect to access latency among mobile clients. A nested locale cover is a set of locale covers $\{L_1, \ldots, L_m\}$ such that each site appears in some locale cover L_i. A client receives data about at least one of its neighbors in each L_i, and about all its neighbors across all m covers. The client can refine the information it has after each cover appears.

The length of the broadcast cycle is the sum of the lengths of the individual nested locale covers (see Fig. 2). The challenge is to find nested locale covers that minimize $\sum_{1 \leq i \leq m} |L_i|$.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

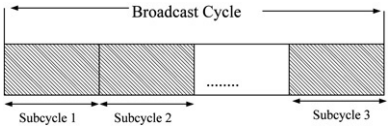


Fig. 2. A nested broadcast cycle of m subcycles. Each subcycle broadcasts a locale cover.

3. k-domain locale covers

The k-domain locale cover problem is to find an $L \subseteq X$ which hits the compass of all k-domains. That is, if $\mathcal{F}$ be a family of subsets defined as follows: $\mathcal{F} = \{\widehat{D} | D \text{ is disk}, |\widehat{D}| = k\}$, then the hitting set of $\mathcal{F}$ is a k-domain locale cover.

We first argue, using results from machine learning theory [23], that there exists small size locale cover regardless of the spatial distribution of the sites. Then we describe our algorithm for efficiently finding small size k-domain locale cover. The algorithm runs in two phases. In the first phase, it generates $\mathcal{F}$. The second phase computes the hitting set for $\mathcal{F}$.

3.1. Existence of small size k-domain locale cover

We first describe the notion of VC-dimension [24] and then discuss its connection with k-domain locale Covers. A *range space* is a pair $(X, \mathcal{R})$, where $\mathcal{R}$ denotes the family of subsets over X. Members of $\mathcal{R}$ are referred to as ranges. Further, let $\mathcal{C}$ denote the family of subsets such that each $C \in \mathcal{C}$ is the compass for some disk D.

The Vapnik–Chervonenkis or *VC-dimension* of a range space $(X, \mathcal{R})$ is the cardinality of the largest subset of $A \subseteq X$ for which $\{A \cap R \mid R \in \mathcal{R}\}$ is the power set of A.

Haussler and Welzl [11] defined the notion of an ϵ-net. Formally, for a range space $(X, \mathcal{R})$ and $0 < \epsilon < 1$, a subset $T \subset X$ is called an ϵ-net for X with respect to $\mathcal{R}$ iff for every range $R \in \mathcal{R}$ with $|R| > \epsilon|X|$, we have $R \cap T \neq \emptyset$.

Lemma 3.1 (*[11,15]*). *Let $S = (X, R)$ denote a set system with VC-dimension less than d then there exists a ϵ-net of S of size $cd/\epsilon \log 1/\epsilon$ where c is an absolute constant.*

Further it is shown in [4] that there is a positive constant c such that a random sample of size $c/\epsilon^2(d \log d/\epsilon + \log 1/\delta)$ is an ϵ-net with probability at least $1 - \delta$. Later Matousek proved the following result:

Lemma 3.2 (*[16]*). *Let $0 < \epsilon \leq 1$. Given range space $(X, \mathcal{C})$, there exists an ϵ-net for R of size at most $60\lceil 1/\epsilon \rceil - 10$.*

A k-domain locale cover is an ϵ-net for range space $(X, \mathcal{F})$ where $\epsilon = k/n$ and $n = |X|$. Since $\mathcal{F} \subseteq \mathcal{C}$, it follows from 3.2 there exists that a k-domain locale cover of size of $O(n/k)$.

Therefore, an algorithm for finding ϵ-nets is also an algorithm for k-domain locale covers. The algorithm by Matousek [16] generates an ϵ-net of size $O(1/\epsilon)$, but is impractical for a large set X since its running cost is $O(n^3)$.

Here we present an $O(n^2k)$ algorithm for finding k-domain locale cover. We also give a partitioning based heuristic to improve its scalability. Further, we demonstrate

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

through extensive experiments that the proposed technique is able to obtain locale cover of approximately $2n/k$ for the dataset used, regardless of distribution.

3.2. Relating $\mathcal{F}$ to order-k Voronoi diagrams

We now show that $\mathcal{F}$ can be directly derived from the order-k Voronoi diagrams. If $X = \{s_1, s_2, \ldots, s_n\}$ is a given set of sites, its order-1 Voronoi diagram is the collection of Voronoi cells $V(s_i)$ for each s_i, defined as the set of points $q \in \mathbb{R}^2$ such that $\text{dist}(q, s_i) < \text{dist}(q, s_j), \forall j \neq i$.

The order-k Voronoi diagram generalizes this notion. Subset $P \subseteq X$ with cardinality k induces a k-order Voronoi cell if inside this cell, the set of k nearest neighbors of every point is exactly the set P. Formally, the order-k Voronoi cell induced by P is $V^k(P) = \{q \in \mathbb{R}^2 \mid \forall p \in P, \forall p' \in X \setminus P, \text{dist}(q, p) < \text{dist}(q, p')\}$.

Theorem 3.3 shows an elegant relation between k-domains of compass k and order-k Voronoi diagrams.

Theorem 3.3. *Let X be a set of sites, and let $\mathcal{F} = \{F \mid F \subseteq X, |F| = k,$ and there is a disk $D \subseteq \mathbb{R}^2$ with $\widehat{D} = F\}$. There is a bijection between elements $F \in \mathcal{F}$ and cells in the order-k Voronoi diagram for X.*

Proof. Consider a subset $F \in \mathcal{F}$. By definition, there is a D with $\widehat{D} = F$, so that F is the set of k-nearest neighbors of the center q of disk D. By the definition of order-k Voronoi diagrams, point q belongs to the Voronoi cell of F.

Conversely, let V be a Voronoi cell of k-subset F and q be a point within V. By definition of order-k VD, F is the set of q's k-nearest neighbors. Let r be the distance from q to the farthest site in F. Obviously, the disk centered at q with radius r contains exactly the sites in F. Further, no site $X \setminus F$ lies inside D. Therefore, $\widehat{D} = F$, so that $F \in F$. □

This theorem gives us a way to compute $\mathcal{F}$ by computing the order-k Voronoi diagram. We can infer that the size of $\mathcal{F}$ is reasonable, and contains $O(nk)$ subsets, since the order-k Voronoi diagram has $O(nk)$ Voronoi cells [6]. Several algorithms have been proposed to compute order-k Voronoi diagrams. (See the survey by Boissonnat [7] and references therein.) We use the randomized incremental algorithm described by Aurenhammer et al. [6]. The algorithm inserts the given sites in a random order one at a time, while maintaining the order-k Voronoi diagram at each stage of insertion. The algorithm has expected cost $O(nk^2 + nk \log^2 n)$, and requires optimal space $O(nk)$ [6,9].

3.3. Obtaining k-domain locale cover from $\mathcal{F}$

Clearly, the hitting set of $\mathcal{F}$ is a k-disk locale cover over X. We describe two methods to efficiently compute the hitting set: one using a greedy method, and the other using random sampling. Let subset $F \in \mathcal{F}$ be *covered* by $R \subseteq X$ if $F \cap R \neq \emptyset$. The generic step of our greedy algorithm is to simply pick the site that belongs to the highest number of sets not yet covered until all elements of $\mathcal{F}$ are covered. If n is the number of sites and $O(nk)$ is the size of $\mathcal{F}$, the complexity of this algorithm is $O(n^2k)$. If the number of sites is large, then the size of $\mathcal{F}$ will also be large. Therefore, a naive implementation for finding the locale cover performs poorly.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

3.3.1. An efficient greedy method with partitioning

We now develop a divide and conquer approach for finding a hitting set for a given $\mathcal{F}$, based on the observation that each set in $\mathcal{F}$ is constrained by some spatial region. For each $S \subset X$, let M_S denote the minimum bounding rectangle (MBR) over the sites of S. We decompose region R into m (preferably disjoint) rectangles $R_1, R_2, \ldots, R_m$. Let us define $\mathcal{F}_i \subseteq \mathcal{F}$ as the subfamily that contains subsets $F \in \mathcal{F}$ whose MBRs intersect region R_i.

We first find a locale cover L_i for each subfamily $\mathcal{F}_i$, and obtain a locale cover L for $\mathcal{F}$ as the union of the locale covers for the subfamilies, that is, $L = \cup L_i$.

Now, if the $\mathcal{F}_i$ are such that only a few members of $\mathcal{F}$ appear in more than one $\mathcal{F}_i$, then the size of locale cover obtained by this method is likely to be close to that obtained using a naive greedy approach. The sites in each F are physically close to each other, since they are within the compass of some disk. Therefore, only subsets with MBRs spanning multiple subregions R_i will be replicated across several $\mathcal{F}_i$. Clearly, this is likely to occur only for a small number of MBRs.

This divide-and-conquer computation will be most efficient if the $|\mathcal{R}_i|$ are of equal size $|\mathcal{F}|/m$. We use the capacitated clustering algorithm [2] with inter-cluster distance [5] as metric to obtain rectangles R_i of roughly equal size. Our experiments with real datasets yield locale covers of size close to that obtained by the naive greedy approach, but at a fraction of its cost.

3.3.2. A random sampling approach

Bronnimann [8] has proposed an iterative randomized algorithm to find hitting sets of almost optimal size for ranges with finite VC-dimension. Unfortunately, this approach requires an efficient ϵ-net finder. The only known algorithm to find ϵ-nets for this problem has $O(n^3)$ time complexity [16], making this approach impractical.

Consequently, we use a simple algorithm that randomly picks a site and appends to the set L until L hits all sets in $\mathcal{F}$. The running time of this algorithm is $O(|L| \times |\mathcal{F}|)$ where L is the resulting locale cover.

To the best of our knowledge, it is an open question as to whether the greedy or randomized approach performs better. The iterative randomized algorithm of [8] has better theoretical bounds, although the greedy method seems to outperform it in practice.

3.4. A running example

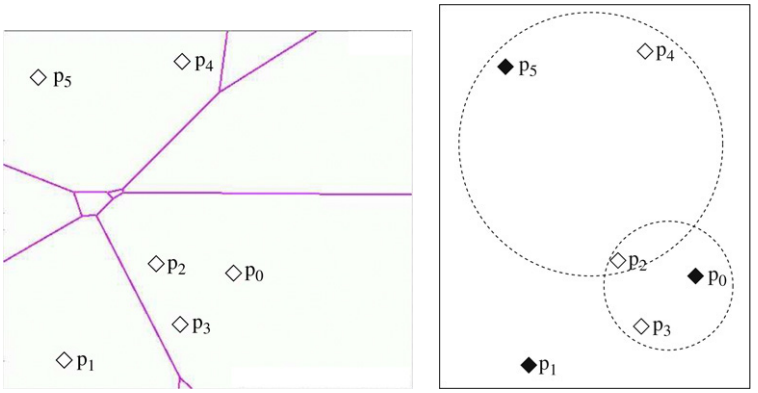
The pseudo code for obtaining locale cover is described in Algorithm 1.

Algorithm 1 Pseudo Code for k-domain locale Cover

Require: a set X of n sites
1: Compute the k-order Voronoi diagram of X using algorithms in [6]
2: Compute $\mathcal{F} = \{P | V^k(P)$ *is a Voronoi cell in order k diagram*, $\forall P \subset X\}$
3: Compute L the hitting set for $\mathcal{F}$ using either of the techniques in Section 3.3
4: Return L as k-domain Locale Cover

Next we illustrate the algorithm through a simple example. Fig. 3(a) shows a set of six sites $X = \{p_0, p_1, \ldots, p_5\}$. First, we use the randomized incremental algorithm [6] and

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001



(a) Order-3 Voronoi diagram. (b) 3-domain locale cover.

Fig. 3. Find 3-Domain Locale Cover for a set of 6 sites.

get the order-3 Voronoi diagrams, shown in Fig. 3(a). In Fig. 3(a), each Voronoi cell corresponds to a unique 3-domain (Theorem 3.3), such all points within it have the same three nearest sites. For example, the lower-right region is the Voronoi cell corresponding to the 3-domain $\{p_0, p_2, p_3\}$. That is, $\{p_0, p_2, p_3\}$ are the three nearest sites for any point chosen within it. From this order-3 Voronoi diagram, we get the family of 3-domains as $\mathcal{F} = \{\{p_1, p_2, p_3\}, \{p_0, p_2, p_3\}, \{p_0, p_1, p_3\}, \{p_0, p_4, p_5\}, \{p_0, p_3, p_4\}, \{p_2, p_3, p_5\}, \{p_0, p_2, p_4\}, \{p_0, p_2, p_5\}, \{p_1, p_3, p_5\}, \{p_2, p_4, p_5\}, \{p_1, p_4, p_5\}, \{p_1, p_4, p_5\}\}$.

We next run the greedy algorithm[1] to find the hitting set $\{p_0, p_1, p_5\}$ for $\mathcal{F}$ This hitting set forms the 3-domain locale cover for X. In Fig. 3(b), it is guaranteed for any client, regardless of position, that at least one of his top 3 nearest sites appears in the 3-domain locale cover, For example, the three sites nearest to a client located at the center of the big circle will be $\{p_2, p_5, p_4\}$. Of these, we see that p_5 belongs to the 3-domain locale cover.

3.5. Handling insertions and deletions

In this section, we briefly discuss techniques to update k-domain locale cover under insertions and deletions. We assume that a high-order Voronoi diagram on X is available. When a new site s is added, we first find the k-order Voronoi diagram for $X \cup s$, from the Voronoi diagram of X, using the algorithm of [6]. The cost of insertion is small under some simple assumptions regarding insertion order, and is equal to the number of Voronoi regions destroyed. Next, we examine all newly created k-domain subsets, and if any of these subsets is not covered by L, we add the site s to L.

If a site s being deleted belongs to the locale cover, we find all newly created subsets F' that are not covered by $L \setminus s$, and run the greedy algorithm only over these subsets to find L'. The final local cover now is $L = L \cup L' \setminus s$.

One drawback of this approach is that the locale cover may grow linearly with the number of updates. We propose to rebuild the locale cover if size of the current locale

[1] We do not use partitioning here for this simple example.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

cover is more than $c * n/k$, where c is a system constant. Through our experiments we see that greedy locale cover is close to $2 * n/k$. Therefore a reasonable choice of c is 2.

4. r-disk locale covers

We give two algorithms to compute r-disk locale covers. The first is similar in flavor to k-domain locale covers, but has $O(n^3)$ complexity. The second is more efficient, and is obtained by reducing it to a sensor cover problem [10].

4.1. Naive algorithm

Following the approach in Section 3.3.1, we will first build a family of subsets $\mathcal{F}$ such that the hitting set $L \subset X$ for $\mathcal{F}$ is an r-disk locale cover. Clearly, in this case, $\mathcal{F}$ is defined by

$$\mathcal{F} = \{\widehat{D} \mid D \text{ is an } r\text{-disk}\}. \tag{1}$$

This definition is useless in constructing $\mathcal{F}$, since there is an infinite number of r-disks. Instead, we apply the following theorem to find an $O(n^3)$ algorithm for generating $\mathcal{F}$. We say that a site *touches* a region R if its distance to the boundary of R is negligibly small.

Theorem 4.1. *Given an r-disk D, we can find another r-disk D' such that $\widehat{D} = \widehat{D'}$, and D' touches at least two sites on its boundary.*

Proof. Move D to obtain D'' so that it touches a point p_1. Rotate D'' around p_1 until it touches another point p_2. This defines D'. □

We can now construct an $O(n^3)$ algorithm. We form all pairs of sites, and draw a circle of radius r that touches each such pair. The compass of each such circle will be a set in the family $\mathcal{F}$.

4.2. r-disk locale cover via sensor cover

Here we first describe the sensor cover problem and then present its relationship with r-disk locale cover.

4.2.1. Sensor covers

Consider a set of n sensors $X = \{s_1, \ldots, s_n\}$ deployed within some region R. Each s_i has a sensing region $R(s_i)$, typically a circle of radius r centered at s_i. A sensor cover of X is a subset $\{s_{i_1}, \ldots, s_{i_m}\}$ such that $R \subset R(s_{i_1}) \cup \cdots \cup R(s_{i_m})$. The region R can be fully monitored by activating only these sensors, reducing power consumption and extending network life.

Finding a minimum cover for a set of sensors X is NP-hard, so a heuristic was proposed in [10] based on the notion of *subelement*, which is the intersection of a set of sensing regions. The heuristic computes the sensor cover by choosing the sensor that covers the most uncovered subelements until all the subelements have been covered. Unfortunately, generating subelements takes $O(n^3)$ time, where n is the number of sensors, so this

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

approach may be impractical when n is large. The work in [10] therefore discretizes the entire region into a $k \times k$ grid, and computes the sensor cover by iteratively selecting the sensor that covers the most uncovered grid points. We refer to this approach as *Discrete-Greedy*. This approach cannot guarantee that the entire region will be covered, since some non-grid points may not have been covered. Although the uncovered fraction of the region can be reduced by discretization space at a finer granularity, no guarantees can be made for the uncovered fraction.

However, this work did not give the relationship between the fraction of uncovered region and size of cover. Here we theoretically quantify the relationship between the size of cover and fraction of uncovered region.

Let $X = \{s_1, s_2, \ldots s_n\}$ be the set of points for which an r-disk locale cover L must be computed. Place n sensors, one at every location $s_i \in X$. If $SC \subseteq X$ is an r-sensor cover for X, SC is clearly also an r-disk locale cover. There is a mapping between our naive algorithm (see Section 4.1) and the subelement-based greedy algorithm of complexity $O(n^3)$ presented in [10]. Moreover, there is one-to-one correspondence between subsets $F \in \mathcal{F}$ and subelements in the algorithm of [10].

The main algorithm of [10] (*Discrete-Greedy* in Section 4.2.1) achieves efficiency by avoiding the costly computation of subelements. However, the sensor cover thus obtained may not cover the entire region.

4.2.2. Analysis of Discrete-Greedy algorithm

Theorem 4.2 shows that a large fraction of the region is covered by the first few sensors in discrete-greedy.

Theorem 4.2. *Given a region R of unit area and a set X of n sensors, let Ω denote the size of optimal sensor cover for R. Let δ be the area of uncovered region after the first ω iterations of the greedy algorithm. Then*

$$\delta \leq \mathrm{e}^{-\frac{\omega}{\Omega}}. \tag{2}$$

Proof. Let R_i be the uncovered area after the ith iteration of the greedy algorithm. R_i can be covered by at most Ω disks. Therefore, there exists at least one sensor which covers at least R_i/Ω area. Hence, $R_{i+1} \leq R_i(1-1/\Omega)$. Since $R_0 = 1$ and $R_\omega = \delta$. $\delta \leq (1-1/\Omega)^\omega$. Hence, $\delta \leq \mathrm{e}^{-\omega/\Omega}$. □

4.2.3. Cardinality of the cover

We were able to guarantee small-sized k-domain locale covers regardless of the spatial distribution of the sites. No such guarantees are possible for r-disk locale covers. In fact, if the distance between the closest pair in X is greater than r, then the locale cover must contain all the points of X.

4.3. Handling insertions and deletions

Since the r-disk locale cover algorithm is straightforward, updates are easy to handle. For any inserted site s (or circle, as site corresponds to circle in its dual), we examine the

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

grid points that are covered by this site and if there exists a grid point covered only by s then we append s to the locale cover.

Alternatively if the site being deleted belongs to the r-disk cover, find the grid points covered only by this site and run the greedy algorithm for only these uncovered grid points.

5. Nested locale covers

Larger locale covers will clearly be needed as the desired accuracy increases. Designing locale covers to meet the requirements of the most demanding client would be wasteful and unfair since average wait time increases with the size of the broadcast cycle. *Nested locale covers* ensure fairness with respect to access latency. A nested locale cover is a set of locale covers $\{L_1, \ldots, L_m\}$ such that each site appears in some locale cover L_i. A client receives data about at least one of its neighboring sites in each L_i, and about all its neighbors if it listens to all the L_i. In the rest of the section we discuss algorithms to minimize the length of the broadcast cycle $\sum |L_i|$.

5.1. The Rand + Greedy algorithm

If $P_i \subseteq X$ is any subset of sites, let $\mathcal{F}_i$ and $\overline{\mathcal{F}}_i$ be the families of subsets in $\mathcal{F}$ covered and not covered by P_i, respectively. That is, $P_i \cap F_j = \emptyset, \forall F_j \in \overline{\mathcal{F}}_i$. Further, let $H(\overline{\mathcal{F}}_i) \subseteq X$ denote the hitting set for subfamily $\overline{\mathcal{F}}_i$ obtained by the greedy algorithm (see Section 3.3.1).

We desire a series of m locale covers $L_1, L_2, \ldots L_m$, such that $\cup L_i = X$. We begin by distributing the elements of X randomly into m bins to create a partition $P_1, P_2, \ldots P_m$ of roughly equal sized sets. Next, we compute $\overline{\mathcal{F}}_i$ and then use the greedy approach of Section 3.3.1 to obtain $H(\overline{\mathcal{F}}_i)$. The locale cover is now obtained as $L_i = P_i \cup H(\overline{\mathcal{F}}_i)$.

5.1.1. Analysis of Rand + Greedy approach for k-domain locale covers

We now need to show that $H(\overline{\mathcal{F}}_i)$ is small. Instead, we will show that $\overline{\mathcal{F}}_i$ is small for k-domain locale covers, which in turn implies that $H(\overline{\mathcal{F}}_i)$ is small.

Theorem 5.1. *Let $P_1, \ldots, P_m$ be buckets, with $m = k/\log k$. Let all points of X be distributed across these buckets with equal probability. Then,* $\Pr[|\overline{\mathcal{F}}_i| > (1+\delta)n] \leq e^{-\delta^2 n/4}$.

Proof. If P_j is any partition, $\Pr[x \in P_j] = 1/m$ for $x \in X$. As in Section 3.2, let $\mathcal{F} = \{F_1, F_2, \ldots, F_t\}$ where $t = O(nk)$, and each F_i has cardinality k. Consider a $F_i = \{p_i^1, \ldots, p_i^k\}$. Let X_{ij} be a random variable that is 1 when F_i not covered by partition P_j. Therefore, $\Pr[X_{ij} = 1] = p = (1 - 1/m)^k = 1/k$. Clearly, $\overline{\mathcal{F}}_j = \{F_i | X_{ij} = 1\}$. Hence, $E[|\overline{\mathcal{F}}_j|] = pt = n$. Using Chernoff bounding, we obtain $\Pr[|\overline{\mathcal{F}}_j| > (1+\delta)n] \leq e^{-\delta^2 n/4}$. □

5.1.2. Analysis of Rand + Greedy approach for r-disk locale covers

Consider a set $X = \{s_1, \ldots, s_n\}$ of n points uniformly distributed inside a unit square. Let $P_j \subset X$ be a random sample. We ask whether P_j a good r-disk locale cover.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

Our algorithm for finding an r-disk locale cover (see Section 4) first generates a family of subset $\mathcal{F}$ whose hitting set is an r-disk locale cover. We are interested in $|\overline{\mathcal{F}}_j|/|\mathcal{F}|$, the fraction of subsets of $\mathcal{F}$ that is not hit by random sample P_j.

Let $\mathcal{F} = \{F_1, F_2, \ldots, F_t\}$, and let P_j be a random sample of X, where $|P_j| = \omega$. We show that the probability that any given $F_i \in \mathcal{F}$ is not hit by P_j decreases exponentially with ω.

Theorem 5.2. *Let $P_1, \ldots, P_m$ be buckets, and let each point of X be distributed across these buckets with equal probability. Let $c = \pi r^2$, and $|P_j| = \omega$. Then*

$$\Pr[F_i \cap P_j = \emptyset] = (1 - c)^{\omega}. \tag{3}$$

Proof. F_i denotes the set of points inside an r-disk D. Since any point of X will be inside D with probability $c = \pi r^2$, it follows that $\Pr[|F_j| = y] = \binom{n}{y} c^y (1 - c)^{n-y}$.

For a random sample P_j with size of ω, the probability that F_i is not hit by P_j, given $|F_i| = y$, is

$$\Pr[F_i \cap P_j = \emptyset \mid |F_i| = y] = \binom{n - y}{\omega} \Big/ \binom{n}{\omega}, \tag{4}$$

so $\Pr[F_i \cap P_j = \emptyset] = \sum_{y=0}^{n-\omega} \left(\binom{n-y}{\omega} / \binom{n}{\omega}\right) \binom{n}{y} c^y (1 - c)^{n-y} = (1 - c)^{\omega}$. □

From Theorem 5.2, we conclude that the fraction of subsets of $\mathcal{F}$ that is not hit by the random sample P_j will decrease exponentially with $|P_j|$. This ensures that our Rand + Greedy algorithm will hit most of the subsets with a small random sample, while the greedy approach will find additional points to hit the rest of the subsets.

6. Performance evaluation

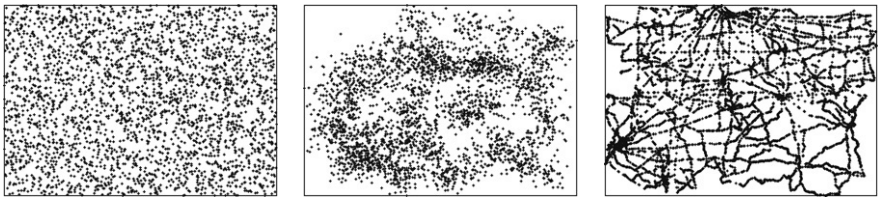
This section presents the experimental results for the proposed technique of locale cover, using both synthetic and real datasets. Synthetic datasets, UN30k and GN30k, have 30,000 sites within a unit square having random and Gaussian distribution respectively.

Real dataset CH represents 7972 road intersections in Chicago. Dataset CA, obtained from [19], contains 3257 traffic stations monitoring traffic speed on the freeways across the state of California. Dataset SH, obtained from [18], contains 6769 shopping centers across the state of California. Sites in dataset CH are relatively uniformly distributed, while CA and SH are highly skewed. Fig. 4(a)–(d) depict these datasets.

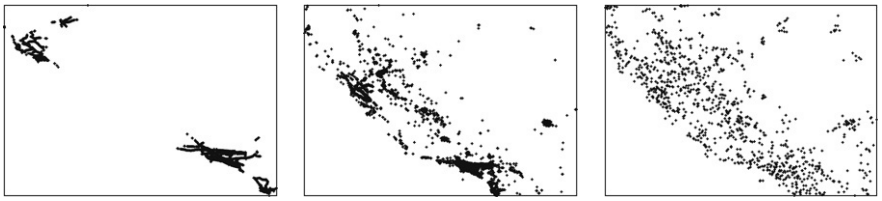
6.1. Experiment setup

In our experiments we use average rank and relative discrepancy as performance criteria. Given a broadcast subset B of S, let $N(c, B)$ be the nearest neighbor in B for some client c. The rank, $R(c, B)$, is the rank of $N(c, B)$ in the set S. That is, if $N(c, B)$ is tth nearest neighbor of c in S, then $R(c, B) = t$. Average rank of a broadcast set B is the mean of rank values of the entire client population.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001



(a) UN30k. (b) GN30k. (c) CH.



(d) CA. (e) SH. (f) SH client.

Fig. 4. Datasets and clients.

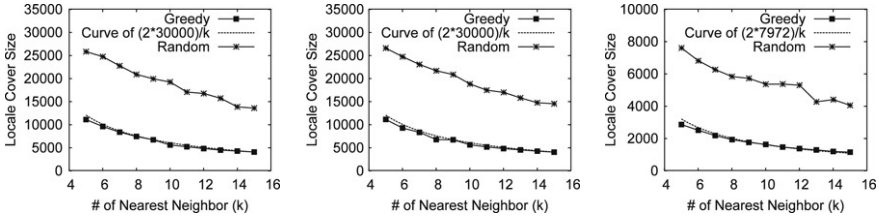
The relative discrepancy compares the difference between the nearest client in the complete dataset with that of nearest client in the broadcast set. Let L be the locale cover, RS be a random subset of S, d be the distance from client c to its nearest site in the complete dataset S. Let d_L and d_{RS} denote the distance from the client c to the nearest site in L and RS, respectively. The relative discrepancies for locale cover and the random broadcast are $\delta_{LC} = \frac{d_L - d}{d}$ and $\delta_{RS} = \frac{d_{RS} - d}{d}$, respectively.

Our experiments use distributions of client locations more realistic than a uniform distribution. In general, clients are likely to be in the vicinity of sites, so that they will be clustered in a fashion similar to that of the sites. For example, we are likely to find more clients in densely populated areas than in sparsely populated ones. We generate such client distributions as a two-step process. We first pick a point p uniformly in plane and keep p as the location of a client with probability $C_1^{-C_2 * d}$, where C_1 and C_2 are positive constants, and d is the distance from p to its nearest site in S. In effect, the density of clients is exponentially distributed with respect to the distance from the closest site. Fig. 4(f) shows 1% of clients used for the dataset SH.

Our first set of experiments compare two alternative schemes, which we call *random cover* and *STR-cover*, respectively, each with the same number m of sites. Random cover is simply a random subset of size m. The sites in STR-cover are chosen more judiciously. The set of sites is partitioned into clusters using STR-clustering [14], and we pick one site from each cluster. The STR-clustering algorithm [14] is an efficient clustering algorithm for R-tree pack loading, and runs in two phases. First, we sort the sites along X-axis and divide them into $\sqrt{m}$ tiles. Next, sites in each tile are sorted along Y-coordinate, and divided into $\sqrt{m}$ clusters, If there are less than m clusters, we randomly pick additional sites until we get a subset of m sites.

Our second set of experiments measure performance of nested locale cover in the scenario where clients are interested in all k-nearest sites. We compare nested locale covers

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001



(a) UN30k. (b) GN30k. (c) CH.

Fig. 5. Sizes vs. k for k-domain locale covers.

with random broadcast (where all sites are broadcasted in some random order). For this set of experiments, we use the k-relative discrepancy metric. This metric is derived from the relative discrepancy metric, by replacing all distances in it with the *sum* of distances from the client c to all its top-k nearest sites.

6.2. Size of k-domain locale cover

Our first set of experiments evaluated the size of locale covers varying the value of k in the k-domain locale coverage problem. The locale cover is obtained using our partition-based greedy approach (see Section 3.3.1). Fig. 5(a)–(c) show the sizes of locale covers for different k, for datasets UN30k, GN30k and CH. In these figures, the dashed line represents the curve of $\frac{2n}{k}$, for purposes of comparison. Clearly, we are able to obtain locale covers of approximately $\frac{2n}{k}$ for all the datasets, regardless of distribution. This is actually better than Matousek's theoretical bound for this problem [16], which is at most $O(\frac{60n}{k})$.

6.2.1. Greedy and random compared

We also compared the greedy approach with the simple random sampling technique.[2] Fig. 5(a)–(c) show this comparison for the UN30k, GN30k and CH dataset. We see that the greedy approach outperforms simple random sampling by a factor of three or four. These experimental results agree with Bronnimann's finding [8] that although the randomized method has a better theoretical bound, the greedy approach performs well in practice. (See also Fig. 6.)

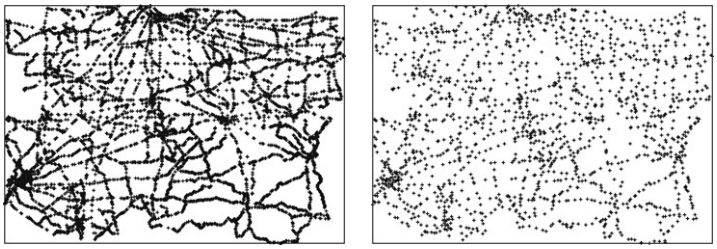
The results for r-disk are quite similar to k-domain locale cover, and thus are omitted for the sake of space.

6.3. Performance measurement

Fig. 7(a)–(c) compare the average relative discrepancy of locale cover with the random cover and STR-cover. Fig. 8(a)–(c) compare the average rank of locale covers with the random cover and STR-cover. Locale covers show smaller relative discrepancy and rank,

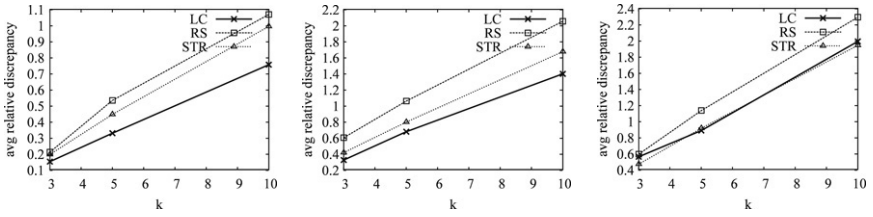
[2] Random Sampling is different from the random broadcast scheme. In random broadcast scheme we choose a subset of fixed size. In the random sampling we randomly pick sites one by one until the chosen subset satisfies to be a locale cover.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001



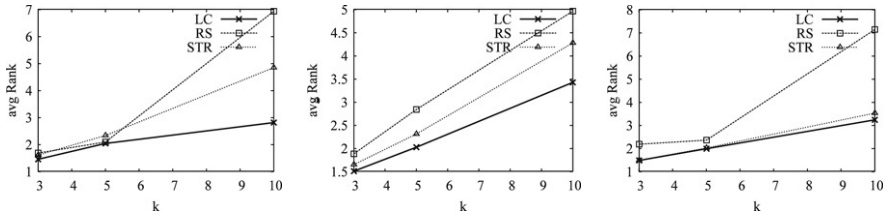
(a) CH. (b) 10-domain locale cover.

Fig. 6. Dataset CH and the 10-domain locale cover.



(a) CA. (b) CH. (c) SH.

Fig. 7. Average discrepancy of locale cover, random cover and STR-cover vs k for different dataset.



(a) CA. (b) CH. (c) SH.

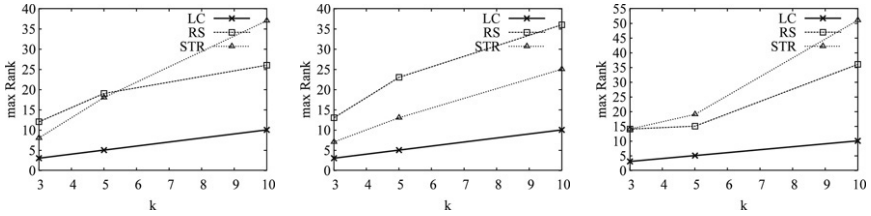
Fig. 8. Average rank of locale cover, random cover and STR-cover vs k for different dataset.

and provide much closer sites to clients. Furthermore, as Fig. 9(a)–(c) show, locale covers guarantee that any client gets at least one of the top-k nearest sites, so that the maximum rank for locale covers is equal to k. The random cover and STR-cover do not provide such guarantees, and the maximum rank can be arbitrarily high. It is interesting to note that STR-cover generally yields smaller average rank than random cover, yet much larger maximum rank than random cover in some cases. The reason is that neither random cover nor STR-cover considers *every* client when picking the sites to broadcast. Hence, there are likely to be clients who get unsatisfactory answers from these methods.

6.4. Nested locale covers

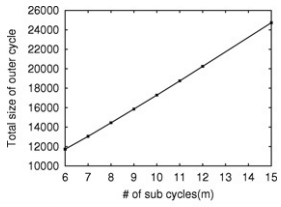
As discussed in Section 5, nested locale covers ensure fairness with respect to access latency. A nested locale cover consists of m subcycles, each subcycle broadcasting a locale

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001



(a) CA. (b) CH. (c) SH.

Fig. 9. Max rank of locale cover, random cover and STR-cover vs k for different dataset. Locale cover's maxRank is bounded.



Total size of locale covers

Fig. 10. Nested broadcast.

cover, and all sites are guaranteed to be included in the full broadcast cycle. A client, at the end of every subcycle, may choose to turn off the channel, since he is now aware that the current answer set intersects with the top-k nearest neighbors. Observe that in an arbitrary broadcast a client has to listen to the entire broadcast, to ensure that he has results from the top-k sites.

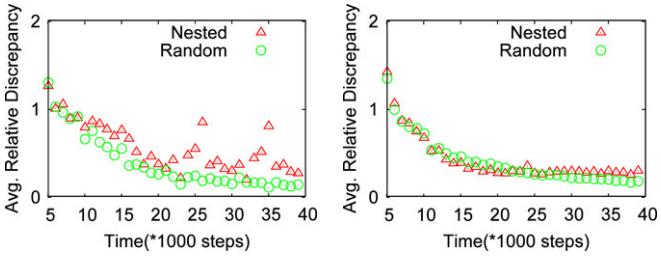
Here we first study the size of the nested locale cover with respect to the number of subcycles. If the number of subcycles m is small, each subcycle will be clearly be long. As m increases, the subcycles become shorter, but the total length of the nested broadcast cycle increases. We found that for $m \leq k$, the average length of subcycles approaches $\frac{2n}{k}$, while the total length of nested cycle is approximately $2n$.

Fig. 10 shows the length of broadcast cycle for k-domain locale covers for $k = 10$ with the CH dataset. The x-axis shows the value of m, while the y-axis shows the total length of outer broadcast cycle for each m.

Next we simulated a broadcasting environment in which a site is disseminated at each simulation step. Clients arrive at different stages of the simulation and leave after receiving their top-k sites. We measured the average quality of answer (specifically average rank and average k-relative discrepancy) from a nested broadcast cycle to that of random broadcast at each step of the simulation.

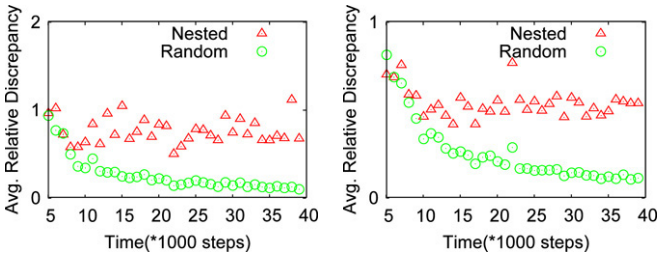
The simulation runs for 40,000 steps. Starting with an empty set of listeners, a client is added at each step of the simulation with probability $1/2$. Their location is sampled from the distribution model described for the locale cover experiment. At each step, a site from the broadcast scheme is disseminated to the current client pool. The broadcast resumes from the beginning once it's finished.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001



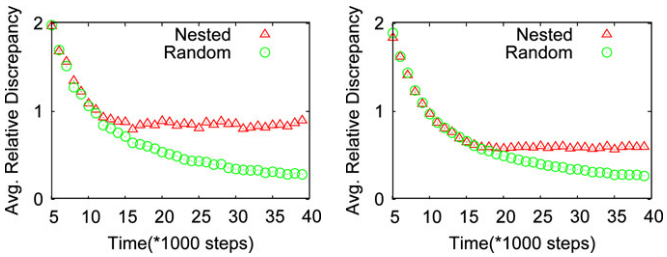
(a) $k = 5$. (b) $k = 10$.

Fig. 11. [SH Dataset] Progression of relative discrepancy with respect to time for nested and random broadcast scheme.



(a) $k = 5$. (b) $k = 10$.

Fig. 12. [CA Dataset] Progression of relative discrepancy with respect to time for nested and random broadcast scheme.



(a) $k = 5$. (b) $k = 10$.

Fig. 13. [CH Dataset] Progression of relative discrepancy with respect to time for nested and random broadcast scheme.

Fig. 11(a), (b) show the progression of average k-relative discrepancy for the nested locale cover and the random locale cover for the SH dataset. Each point on the plot shows the average relative discrepancy after broadcast of 1000 sites. The average relative discrepancy of nested locale cover is higher than that of random broadcast. As k increases, nested locale cover shows marginally lower performance than random broadcast. The average relative discrepancy performs comparably to the random broadcast scheme for larger k. Datasets CA and CH show similar progression of relative discrepancy in Figs. 12(a) and 13(b).

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

Although nested locale covers introduce higher relative discrepancy for users interested in *all* top-k nearest sites, we observe that nested locale covers make these guarantees *regardless of user distribution*. In contrast, traditional broadcasting techniques make no such guarantees; some users may get their answer very quickly, while other unlucky users may have to wait for the whole broadcast cycle to get the answer. We believe this feature make nested locale covers a good candidate in satellite broadcasting applications.

7. Related work

Data dissemination using wireless broadcasting has been extensively studied for location-based services within the research and commercial communities.

7.1. Air indexing in wireless environments

To reduce power consumption, Imielinski et al. [13] proposed *air indexing* techniques, which pre-compute an index and interleave it with the data on the channel, enabling the device to go into active mode only when the requested data arrives. Several air indices for querying spatial data have been proposed, for example, the *D-trees* [28] for point-queries, the *grid-partition index* [30] for nearest neighbor queries, the *sorted lists* [29] for general k-NN queries and the *exponential index* [27].

We focus on finding a small subset to broadcast such that every user will receive data from at least one neighbor. In contrast, existing work typically broadcasts all data, and focuses on building indices for the data. Such indexing schemes are complementary to our work; we can use them to minimize the turning time and access latency for a locale cover.

7.2. Broadcast scheduling

The characteristics of location-based services make conventional broadcast scheduling approaches inappropriate. Since mobile clients move fast (100 km/h say), their access patterns change rapidly. Models such as broadcast disks [1] are not suitable for location-based services, since they assume static client access patterns.

Some recent models [12,3,25,22,21] incorporate feedback from clients to accommodate dynamic access patterns. However, such work focuses mainly on scheduling, assuming access probabilities are already known to the server. In location-based services, tracking the locations of mobile clients and estimating the access patterns can require high overhead, or be impossible. The work in [25] uses random sampling to obtain an accurate picture of client interest patterns, and to answer queries approximately, and comes closest in spirit to our work, but differs in several respects. Clients in [25] are interested in documents, not location-dependent data. Consequently, their method must actually estimate client access patterns by on-line sampling, rather than infer correlations due to proximity. Also, they do not guarantee that the broadcast data will satisfy all client queries, even if only approximately.

The data to broadcast are also dynamic (traffic data, for example), so timely transmission to mobile clients is important. Previous scheduling work [12,3,22], typically schedule and broadcast all the data items, without using correlations among the data items.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

We broadcast a small subset instead of the whole dataset, and need no knowledge of client access patterns.

8. Conclusions

We introduce the notion of locale cover, and present several novel formulations and variants of the data dissemination problem for location-dependent data in broadcasting environments. Our schemes choose a small subset of sites that include a site in the neighborhood of all clients, regardless of their number or distribution. This method significantly reduce broadcast bandwidth and access latencies for clients, and scales well with the number of users and sites. Our experiments confirm the applicability and efficiency of our schemes.

Our notion of locale cover is very general and is likely to be applicable beyond the domain of data dissemination, for example, in spatio-temporal data mining, and approximate indices. We intend to investigate these possibilities in future work.

Acknowledgments

We would like to thank Dimitrios Gunopulos for several discussions and useful comments. This work was supported in part by grants from Tata Consultancy Services, Inc., and a matching grant from the MICRO program of the University of California.

References

[1] S. Acharya, R. Alonso, M. Franklin, S. Zdonik, Broadcast disks: Data management for asymmetric communication environments, in: Proceedings of SIGMOD, 1995, pp. 199–210.
[2] S. Ahmadi, I. Osman, Greedy random adaptive memory programming search for the capacitated clustering problem, Eur. J. Oper. Res. (2003).
[3] D. Aksoy, M. Franklin, Scheduling for large-scale on-demand data broadcasting, in: Proceedings of IEEE INFOCOM, 1998, pp. 651–659.
[4] N. Alon, J. Spencer, The Probabilistic Method, Wiley-Interscience Publication, New York, 1992.
[5] T. Asano, B. Bhattacharya, M. Keil, F. Yao, Clustering algorithms based on minimum and maximum spanning trees, in: Proceedings of the Fourth Annual Symposium on Computational Geometry, ACM Press, 1988, pp. 252–257.
[6] F. Aurenhammer, O. Schwarzkopf, A simple on-line randomized incremental algorithms for computing higher order voronoi diagrams, in: Proceedings of ACM Symp. on Computational Geometry, North Conway, New Hampshire, US, June 1991, pp. 142–151.
[7] J.-D. Boissonnat, O. Devillers, M. Teillaud, A semi-dynamic construction of higher order voronoi diagrams and its randomized analysis, Algorithmica 9 (4) (1993) 329–356.
[8] H. Bronnimann, M.T. Goodrich, Almost optimal set covers in finite vc-dimension: (Preliminary version), in: Proceedings of the Tenth Annual Symposium on Computational Geometry, ACM Press, 1994, pp. 293–302.
[9] J. Flototto, M. Yvinec, Order-k voronoi diagrams, in: Proceedings of 17th European Workshop on Computational Geometry, Berlin, Germany, March 2001.
[10] H. Gupta, S.R. Das, Q. Gu, Connected sensor cover: Self-organization of sensor networks for efficient query execution, in: Proceedings of the 4th ACM MobiHoc, Annapolis, MD, June 2003, pp. 189–200.
[11] D. Haussler, E. Welzl, Epsilon-nets and simplex range queries, Discrete Comput. Geom. 2 (1987) 127–151.
[12] Q. Hu, D.-L. Lee, W.-C. Lee, Dynamic data delivery in wireless communications environments, in: Proceedings of Workshop on Mobile Data Access, 1998, pp. 213–224.

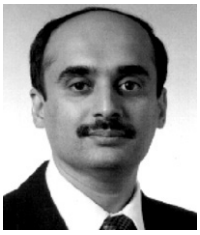
Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001

[13] T. Imielinski, S. Viswanathan, B. Badrinath, Data on air: Organization and access, IEEE TKDE 9 (3) (1997) 353–372.
[14] M.E.J. Leutenegger, S.T. Lopez, Str: A simple and efficient algorithm for r-tree packing, in: Proceedings of the 13th ICDE, 1997.
[15] J. Matousek, Lectures on Discrete Geometry, vol. 212, Springer-Verlag, New York, 2002.
[16] J. Matousek, R. Seidel, E. Welzl, How to net a lot with little: Small ϵ-nets for disks and halfspaces, in: Proceedings of the Sixth Annual Symposium on Computational Geometry, ACM Press, 1990, pp. 16–22.
[17] Microsoft. Microsoft spot. http://www.msndirect.com.
[18] Navteq. Navteq. http://www.navteq.com/.
[19] PeMS. Freeway performance measurement system. http://pems.eecs.berkeley.edu/Public/.
[20] SIRIUS. Sirius radio. http://www.siriusradio.com/.
[21] N.H. Vaidya, S. Hameed, Data broadcast scheduling: On-line and off-line algorithms, Tech. Rep, 96-017, Dept. of Computer Science, Texas A&M University, 1996.
[22] N.H. Vaidya, S. Hameed, Scheduling data broadcast in asymmetric communication environments, Wireless Netw. 5 (1999) 171–182.
[23] V. Vapnik, Statistical Learning Theory, Wiley-Interscience, 1998.
[24] V. Vapnik, A. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, Theory Probab. Appl. 16 (1971) 264–280.
[25] W. Wang, C.V. Ravishankar, Adaptive data broadcasting in asymmetric communication environments, in: Proceedings of IDEAS, 2004, pp. 27–36.
[26] XM. Xm traffic, weather. http://www.xmradio.com.
[27] J. Xu, W.-C. Lee, X. Tang, Exponential index: A parameterized distributed indexing scheme for data on air, in: MobiSYS '04: Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services, ACM Press, 2004, pp. 153–164.
[28] J. Xu, B. Zheng, W.-C. Lee, D.L. Lee, Energy efficient index for querying location-dependent data in mobile broadcast environments, in: Proceedings of the 19th International Conference on Data Engineering, ICDE, Bangalore, India, March 2003, pp. 239–252.
[29] B. Zheng, W.-C. Lee, D.L. Lee, Search k nearest neighbors on air, in: Proceedings of the 4th International Conference On Mobile Data Management, Melbourne, Australia, January 2003, pp. 181–195.
[30] B. Zheng, J. Xu, W.-C. Lee, D.L. Lee, Energy-conserving air indexes for nearest neighbor search, in: Proceedings of 9th International Conference on Extending DataBase Technology, EDBT'04, 2004, pp. 48–66.

Dr. Sandeep Gupta received the B.S. degree in computer science from the Indian Institute of Technology (IIT), Guwahati and the Ph.D. degree in computer science from the University of California, Riverside in 2000 and 2006, respectively. He is currently a postdoctoral fellow at the San Diego SuperComputing Center. His research interest include databases, datamining, algorithms, and computational geometry.

Dr. Jinfeng Ni is a staff software engineer in IBM's Silicon Valley Laboratory. He holds a Bachelor and Master degree in computer science from the University of Science and Technology of China, and a Ph.D. in Computer Science from the University of California, Riverside. His research interests include spatio-temporal databases, XML and security.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001



Chinya V. Ravishankar is a Professor of Computer Science and Associate Dean in the Bourns College of Engineering at the University of California, Riverside. Between 1986–1999, he was on the Faculty of the Electrical Engineering and Computer Science Department at the University of Michigan–Ann Arbor.

Prof. Ravishankar's research is currently in the areas of Databases, Networking, and Security. He holds an undergraduate degree in Chemical Engineering from the Indian Institute of Technology, Bombay, and a Ph.D. in Computer Science from the University of Wisconsin — Madison. He is a Senior Member of the Institute of Electrical and Electronics Engineers, and a member of the Association for Computing Machinery.

Please cite this article in press as: S. Gupta, et al., Efficient data dissemination using locale covers, Pervasive and Mobile Computing (2008), doi:10.1016/j.pmcj.2007.11.001