The 28th International Conference on Distributed Computing Systems

Locality Sensitive Information Brokerage in Distributed Sensor Networks

Hong Lu, Anxiao (Andrew) Jiang, Steve Liu
Department of Computer Science
Texas A&M University

Abstract

In sensor network applications, sensors often need to retrieve data from each other. Information brokerage is a scheme that stores data (or index files of data) at rendezvous nodes, so that every sensor can efficiently finds the data it needs. A very useful property for information brokerage is locality sensitivity, which means that a sensor close the original source of the data should also be able to retrieve the data with a small communication cost. Given the locality sensitivity requirement, the key is to design an information brokerage scheme that minimizes the storage cost.

In this paper, we present a locality sensitive information brokerage scheme. It is designed for general locality-sensitive requirements, which include the linear data-retrieval cost (a frequently studied scenario) as a special case. We also prove that for a large class of networks, in the scenario of linear data-retrieval cost, our scheme achieves the asymptotically optimal storage cost. The result also proves the optimality of a few other schemes in the literature.

1 Introduction

Wireless sensor networks have been widely deployed for data collection, in-network processing and distributed control. In those networks, sensor often need to retrieve data from each other to fulfill their functions. When a sensor (consumer) retrieves the data acquired by another sensor (producer), it does not know the location of the producer. In fact, since sensors often have limited memories, they often cannot afford to remember the location of the other sensors in a large network, because the size of such information grows linearly in the size of the network. This poses a challenge for data retrieval. A simple solution is for the consumer to flood the whole network to find out where the producer is and retrieve its data, but flooding consumes too much energy and is not appropriate for large sensor networks. Therefore, an efficient information brokerage scheme that enables efficient data retrieval is needed.

An information brokerage scheme stores data replicas at rendezvous nodes so that consumers can retrieve them efficiently. An information brokerage scheme defines both a publish process and a data-retrieval process. Here publish means that the producer transmits its data to its rendezvous nodes, and stores them there. When a consumer needs to retrieve the data, it sends out a query message, which is routed along a predesigned path until it reaches a rendezvous node. The data-retrieval process is also called data lookup. There are three types of costs associated with an information brokerage scheme: the *publish cost* (the communication cost for transmitting the data replicas from a producer to its rendezvous nodes), the *storage cost* (the number of rendezvous nodes storing the data replicas), and the *lookup cost* (the communication cost for a consumer to retrieve the data). Since a producer publishes its data only once, but the data replicas need to be persistently stored and many consumer may access the data, the latter two costs are relatively more important.

A very useful property for information brokerage schemes is *locality sensitivity*, which means that consumer sensors that are closer to the producer should also be able to retrieve the data more efficiently. A frequently studied scenario is linear lookup cost, which means that the lookup cost for a consumer should be linear in the shortest-path communication cost between the consumer and the corresponding producer (e.g., in [1, 12]). There are several reasons why a locality-sensitive information brokerage scheme is desirable. Sensors closer to each other are more likely to have correlated data and tasks and should collaborate closer to perform network functions. It is beneficial to reduce the lookup cost for them. What's more, when the ratio of the lookup cost and the shortest-path consumer-producer communication cost is bounded, the total

1063-6927/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDCS.2008.36

lookup cost for the network becomes close to the optimal lookup cost for any data-retrieval pattern. In addition, a locality-sensitive scheme is more robust to network partitions if rendezvous nodes store index files of data, because a closer producer and rendezvous nodes are more likely to be in the same network component as the consumer.

Locality sensitive distributed services have been studied in various contexts in the literature [1–3, 5, 7–13]. Most of the locality-sensitive schemes assume the linear lookup cost, which we extend in this paper. Also, the optimality of the proposed schemes still needs exploration. In this paper, we present a locality-sensitive information brokerage scheme for dense sensor networks. The scheme is for general locality-sensitive requirements, which include the linear lookup cost as a special case. We prove that for a large family of networks – growth lower-bounded networks – the scheme achieves the asymptotically optimal storage cost for linear lookup cost. The results also proves the optimality of a few other schemes in the literature.

2 Related Work

There have been quite a few papers on information brokerage schemes. A well known approach is the Geographic Hash Table (GHT) [10], where a hash function is used to determine the rendezvous locations to store data. The same hash function is used by both producers and consumers, so that the consumers know where to find the data. GHT is not a locality-sensitive scheme. Locality-sensitive information brokerage has been studied in [12], where the authors proposed a novel information brokerage scheme that stores the data replicas on curves instead of isolated nodes. The curve along which a producer replicates its data is designed in a way that guarantees it to intersect the lookup path of any consumer. Additional information storage schemes based on landmarks, etc., have been proposed in [5, 7].

Information brokerage is related to the distributed object locating and routing (DoLR) schemes for efficient file sharing in networks, especially overlay networks. In the seminal work by Plaxton et al. [9], a randomized locality-sensitive DoLR solution, called PRR, was proposed for growth bounded networks. The schemes Tapestry [13] and Pastry [11] inherited the basic ideas of PRR, and focused more on the self organization of overlay networks. PRR, Tapestry and Pastry all have linear lookup cost in expectation. LAND [2] is a DoLR scheme for growth bounded networks. It achieves a deterministic $1+\epsilon$ stretch lookup cost in the worst case.

Information brokerage is also related to location service in mobile networks, where a node uses the data stored in location servers to learn the locations of other nodes. The study of location service was pioneered by Awerbuch and Peleg [3]. In their work, the network was modeled by a general graph, and the approach was based on hierarchical regional directories. LLS [1] adopted the ideas in [3] on tracking, and obtained a locality-sensitive scheme for grid-like networks. STALK [4] and MLS [6] achieved a similar tracking performance. They also addressed the issues of fault tolerance and robust performance. The above schemes all have linear lookup costs. GLS is another interesting scheme [8]. It is based on the hierarchical decomposition of the network using a quad-tree structure. Although its lookup cost is not locality-sensitive in the rigorous sense, it does limit the search process within a smaller region, and thus achieves a performance that is nearly locality sensitive.

3 Basic Concepts and Terms

We model the sensor network as a graph $G = (V, E)$. Each edge has an associated communication cost. The length of a path is the summation of the communication cost of its edges. For two sensors u and v, the distance $d(u, v)$ is defined as the length of the shortest path between them. When the context is clear, we will also use d instead of $d(u, v)$. As in many existing works [1, 8, 10, 12], we assume that every sensor knows its own physical location. For simplicity, we assume that the Euclidean distance between two sensors is bounded from below by some constant δ.

Every sensor in the network can be both a producer (when it provides data) and a consumer (when it retrieves data). When a pair of producer and consumer sensors are discussed, we usually use u to denote the producer sensor, and use v to denote the consumer sensor. A producer u publishes its data by storing its data (or the index files of the data) in a set of rendezvous sensors, which we call the *mirrors of* u. When a consumer v needs to retrieve the data of u, v sends a query message along a pre-designed path, which is called the *lookup path of* v. If the set of mirrors of u and the lookup path of v share a common vertex w, we say that the mirrors of u and the lookup path of v *meet* at w. The design of an information brokerage schemes includes a policy that chooses the mirror nodes for each producer, and a policy that chooses the lookup path for each consumer.

For a producer u and a consumer v, let w be the first node in the lookup path of v where the look path and the mirrors of u meet. The lookup cost of v is defined as the communication cost (i.e., length) of the sub-path from v to w. That is, if the lookup path of v is $v \to v_1 \to v_2 \to \cdots \to v_i \to w \to \cdots$, then the lookup cost is $d(v, v_1) + d(v_1, v_2) + d(v_2, v_3) + \cdots + d(v_i, w)$. (In reality, since the query message needs to return to v, the actual communication cost is about twice the lookup cost. But that does not affect the analysis in the paper.)

Let $g(x)$ be a positive and monotonically increasing function. An information brokerage scheme is called *g-locality sensitive* if for every consumer v and every producer u, the lookup cost of v is at most $g(d(u, v))$. When $g(x)$ is linear in x, the information brokerage is said to have a linear lookup cost.

The publish process of a producer has two associated costs: the communication cost of transmitting the data replicas to the mirror nodes, and the cost of storing the data replicas in those mirrors. In this paper, we focus on the storage cost, because the data are persistently stored in the mirrors and the storage cost is substantial. We do not study the communication cost here because a producer only needs to transmit its data once, and the cost is comparatively negligible. For an information brokerage scheme, we define the *storage cost* of a producer u as the number of mirrors it has, and denote it by $f(u)$. The storage cost of an information brokerage scheme is defined as $\max_{u \in V} f(u)$. That is, we focus on the maximum number of mirrors a producer has.

Given the function $g(x)$, the key to a g-locality sensitive information brokerage scheme is to minimize the storage cost.

4 Grid-based Information Brokerage

In this section, we study information brokerage for grid networks, where the x and y coordinates of each grid node are a pair of integers. (Without loss of generality, the minimum distance between two grid nodes is assumed to be one.)

In the following discussion, we assume that for any two nodes u and v,

$$d(u, v) \le c \cdot |uv|, \tag{1}$$

where $|uv|$ is the Euclidean distance between u and v, and $c \ge 1$ is some constant. (c is called the stretch factor.) That is, we assume the communication cost is at most a constant times the Euclidean distance between two nodes. Although for a grid network, it appears natural to let c be $\sqrt{2}$ or a similar value, we choose to let c be a general constant for an important reason. The grid network is an approximation of the underlying physical sensor network with its own routing layer, and the routing stretch factor of the routing layer is not necessarily $\sqrt{2}$ or some other given constant.

We now present a locality-sensitive information brokerage scheme for the grid network, which we call GIB (grid-based information brokerage). We first present a concise version of the scheme for linear lookup cost, which achieves a logarithmic storage cost. We then present its generic version for general locality-sensitivity requirements.

4.1 Information Brokerage Scheme for Linear Lookup Cost

The baseline version of the grid-based information brokerage scheme is designed for linear lookup cost. Specifically, $g(x) = s \cdot x$, where s is an arbitrary positive number called the brokerage *stretch factor*.

The scheme for linear lookup cost adopts a well known technique called exponentially growing neighborhoods, which was also explored by a number of related papers [1, 2, 4, 6, 9, 11, 13]. Although the baseline GIB scheme differs from these schemes in details, it is not meant to be a radically new design. Rather, it serves as a basis for the discussion of the more flexible general version to be presented in the next subsection.

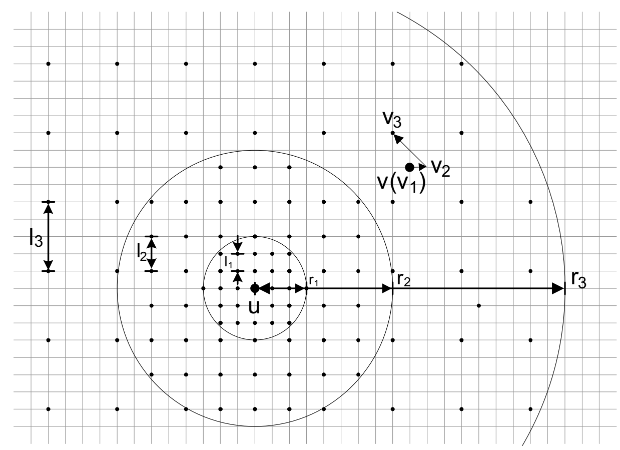

Figure 1. The 1-, 2-, 3-mirrors of node u, and the 1-, 2-, 3-neighbor of node v.

First, we introduce the notions of i-lattice, i-resolution, i-neighbor, i-radius, and i-mirror. For $i = 1, 2, 3 \cdots$, let $l_i = 2^{i-1}$ be an exponentially increasing sequence. l_i is called the i-resolution. Then, the *i-lattice* nodes are defined to be the nodes whose x and y coordinates are both multiples of the *i-resolution* l_i. For any node, the closest (in Euclidean distance) i-lattice node is called its i-neighbor. (Ties are broken arbitrarily.)

For $i = 1, 2, 3 \cdots$, let r'_i be defined as:

$$r'_i = \sqrt{2}c \cdot (2^{i+1} - 1)/s, \quad (2)$$

and let

$$r_i \quad = \quad r'_i + c \cdot l_i/\sqrt{2}. \quad (3)$$

r_i is called the *i-radius*. For a producer u, the set of i-lattice nodes within Euclidean distance r_i to u are called u's *i-mirrors*.

Let $I(d)$ denote the smallest integer such that $r'_{I(d)} \geq d$. By (2), we get

$$I(d) = \lceil \log(\frac{s \cdot d}{\sqrt{2}c} + 1) \rceil - 1 \quad (4)$$

The information brokerage scheme for linear lookup cost is as follows:

- *Publish policy:* For every producer u, for $i = 1, 2, 3 \cdots$, let M_i denote the set of i-mirrors of u. The producer u stores its data replicas in the nodes in $M_1 \cup M_2 \cup M_3 \cdots$.
- *Lookup policy:* For every consumer v, for $i = 1, 2, 3 \cdots$, let v_i denote the i-neighbor of v. The lookup path of v is $v_1 \rightarrow v_2 \rightarrow v_3 \cdots$

We illustrate some of the above terms and concepts in Figure 1. The 1-, 2- and 3-mirrors of a node u, are shown in the figure with black dots. The figure also shows the 1-, 2- and 3- of a node v. Since $l_1 = 1$, v_1 and v are the same node. In this figure, v_3 is the first mirror of u that v's lookup path meets. So v will retrieve the data of u (or the index files of the data) from the mirror node v_3.

Lemma 1. *Let u be a producer and v be a consumer. If $d(u,v) \leq r'_i$ for some integer $i \geq 1$, then the mirrors of u and the lookup path of v meet at the node v_i, where v_i denotes the i-neighbor of v.*

Proof. By equation (1), $d(v, v_i) \leq c \cdot |vv_i|$. By the definition of i-neighbor, we have $|vv_i| \leq l_i/\sqrt{2}$. Therefore, $d(v, v_i) \leq c \cdot l_i/\sqrt{2}$. Since $d(u,v) \leq r'_i$, $d(u, v_i) \leq d(u,v) + d(v, v_i) \leq r'_i + c \cdot l_i/\sqrt{2} = r_i$. So v_i is one of u's i-mirrors. Therefore, the mirrors of u and the lookup path of v meet at v_i. □

It should be clear from the above proof why we chose r_i as $r'_i + c \cdot l_i/\sqrt{2}$ in equation (3). It is one way to get a necessary condition for Lemma 1 to hold.

We can see from Lemma 1 that for a producer-consumer pair u and v, the mirrors of u and the lookup path of v must both contain the $I(d(u,v))$-neighbor of v, $v_{I(d(u,v))}$. Therefore, when v sends a query packet along its lookup path up to the node $v_{I(d(u,v))}$, it can find not only the data of u, but also the data of any other producer with distance $d(u,v)$ from v.

Theorem 1. *Let n be the number of nodes in the grid network. In the information brokerage scheme, every producer stores data in $O(\log n)$ mirrors. For every consumer v and producer u, the lookup cost for v is at most $s \cdot d(u,v)$.*

Proof. We first consider the storage cost. Consider a generic producer u. By the definition of r_i, it is simple to see to that $r_i = O(2^i)$, and thus $r_i/l_i = O(1)$. So the number of i-mirrors of u is less than $\pi r_i^2/(l_i^2)$, which is upper bounded by a constant. The diameter of the grid network is $O(\sqrt{n})$. By equation (4), for the i-mirrors of u, the maximum value of i grows as $O(\log n)$ when n increases. Therefore, the producer u stores data in $O(\log n)$ mirrors.

We now consider the lookup cost of a generic consumer v. Let D denote the diameter of the grid network. And consider the ranges $[r'_0 = 1, r'_1]$, $[r'_1, r'_2]$, ..., $[r'_{I(D)-1}, r'_{I(D)}]$. Let u be a producer, and we will show that the lookup cost for v is at most $s \cdot d(u,v)$. Let i be the integer such that $d(u,v) \in [r'_{i-1}, r'_i]$. For the convenience of discussion, let v_0 denote the same node as v. By Lemma 1, the mirrors of u and the lookup path of v meet at v_i, and the lookup cost for v is:

$$\begin{aligned} \sum_{j=1}^{i} d(v_{j-1}, v_j) &\leq \sum_{j=1}^{i} (d(v, v_{j-1}) + d(v, v_j)) \\ &\leq \sum_{j=1}^{i} 2d(v, v_j) \leq \sum_{j=1}^{i} 2c \cdot |vv_j| \end{aligned}$$

By the definition of v_j, $|vv_j| \leq l_j/\sqrt{2}$, therefore,

$$\sum_{j=1}^{i} d(v_{j-1}, v_j) \leq \sum_{j=1}^{i} 2c \frac{l_j}{\sqrt{2}} = \sqrt{2}c \sum_{j=1}^{i} l_j.$$

Let U_i be the right hand side of the above equation, that is,

$$U_i = \sqrt{2}c \cdot \sum_{j=1}^{i} l_j. \quad (5)$$

The analysis above shows that if the distance $d(u,v)$ falls in the range $[r'_{i-1}, r'_i]$, then the lookup cost for v is at most U_i. For the information brokerage scheme discussed here, $l_j = 2^{j-1}$. By plugging it into equation (5), we get

$$U_i = \sqrt{2}c \cdot \sum_{j=1}^{i} l_j = \sqrt{2}c \cdot \sum_{j=1}^{i} 2^{j-1} = \sqrt{2}c(2^i - 1) = s \cdot r'_{i-1}. \quad (6)$$

Since $r'_{i-1} \leq d(u,v)$, we have $U_i \leq s \cdot d(u,v)$. Therefore, the lookup cost for v is upper bounded by $s \cdot d(u,v)$. So the information brokerage scheme is locality sensitive. □

4.2 Information Brokerage Scheme for General Lookup Cost

We now generalize the information brokerage scheme for general lookup cost functions. Note that in the basic information brokerage scheme for linear lookup cost, the design is based on i-radius r_i and i-resolution l_i. To achieve the linear lookup cost, both l_i and r_i are set as exponential sequences. When given a more general lookup cost function $g(x)$, instead of using exponential sequences, we will use better sequences for l_i and r_i. The basic idea is to formulate the information brokerage problem as an optimization program, and obtain the optimal values of the r_i and l_i sequences by solving it. The optimization program enforces the locality sensitivity constraint and minimizes the storage cost. Specifically, the optimization program is as follows:

$$
\begin{array}{llll}
\min & \sum_{i=1}^{m} \pi r_i^2 / l_i^2 & & (7) \\
s.t. & r_i = r'_i + c \cdot l_i/\sqrt{2} & \forall 1 \leq i \leq m & (8) \\
 & U_i = \sqrt{2} c \sum_{j=1}^{i} l_j & \forall 1 \leq i \leq m & (9) \\
 & r'_i \geq 0, l_i \geq 0 & \forall 1 \leq i \leq m & (10) \\
 & U_i \leq g(r'_{i-1}) & \forall 1 \leq i \leq m & (11) \\
 & r'_0 = 1, r'_m \geq D & & (12)
\end{array}
$$

In the above formulation, m is a variable that denotes the maximum index for the i-mirrors and i-neighbors. The meaning of the above formulation should be clear, because the equalities and inequalities in the formulation all have counterparts in the analysis of the basic scheme for linear lookup cost. The objective function in the formulation, $\sum_{i=1}^{m} \pi r_i^2 / l_i^2$, is a good estimation of a publisher's storage cost. It is asymptotically equivalent to the number of mirrors that the producer has. The equations (8) and (9) correspond to the definitions of U_i and r_i in the previous equations (3) and (5). The inequality (10) is the non-negativity constraint. The inequalities (11) and (12) enforce the locality-sensitivity constraint, that is, the $g(d(u,v))$ lookup cost.

To rigorously prove that the solution of the above program gives a g-locality sensitive information brokerage scheme, we can apply the same arguments as in the proof of Theorem 1. For Theorem 1, it has been shown that U_i is an upper bound of the lookup cost for a producer-consumer pair whose distance falls in the range $[r'_{i-1}, r'_i]$. For general lookup cost function, in the above program, the constraint (11) ensures that this upper bound does not exceed $g(r'_{i-1})$, thus enforcing the g-locality sensitivity for producer-consumer pairs whose distance is in the range $[r'_{i-1}, r'_i]$. Since the above fact holds for every i ($1 \leq i \leq m$), and $r'_m \geq D$, the g-locality sensitivity is achieved for all producers and consumers. In summary, the solution of the above formulation yields an information brokerage scheme that has a g-locality sensitive lookup cost and optimized storage cost.

We now discuss how to solve the GIB program. First, observe that it is a non-linear program, since the objective function (7) and the locality sensitivity constraint (11) are both non-linear. Second, notice that the number m is also an unknown variable. Both characteristics make the solution to the program highly non-trivial. We use the following intuitive approach to solve the program: try a range of reasonable values of m; for each value of m, use standard non-linear programming techniques to solve the more constrained problem. In the end, we choose the value of m that presents the best performance and the corresponding non-linear programming solution.

Choosing the best value for m is very important. Given m, the program can be generally solved very efficiently. We present our experience by using some example experiments. All the experimental results were obtained by the Matlab optimization toolbox on a laptop with a 1.3GHz Intel Pentium M processor and 1Gb memory.

In the following, we show the results for information brokerage with a quadratic lookup function $g(x) = x^2$. Table 1 lists the l_i sequence and the corresponding storage cost f (the objective function in the program) for $m = 1, 2, 3, ..., 8$,and $D = 100$. For succinctness, the sequences r_i, r'_i and U_i are not shown in the table.

m	f	l_i
1	32046	1
2	286.9	1,21.9
3	118.3	1,5.5,71.4
4	100.1	1, 3.4, 15.0, 100
5	102.0	1, 3.1, 10.1, 34.0, 100
6	106.7	1, 3.0, 9.8,31.5, 86.0, 100
7	111.7	1, 3.0, 9.6, 29.8, 77.0, 100, 100
8	116.9	1, 3.0, 9.5, 28.6, 71.0, 100, 100, 100

Table 1. solution of the GIB program with $g(x) = x^2$ and $D = 100$

As shown in Table 1, when m increases from 1 to 8, the storage cost f first decreases monotonically and then increases. The optimal storage cost $f = 100.1$ is obtained when $m = 4$. Recall that the locality sensitivity of the grid-based information brokerage scheme is achieved by segmenting the distance between producers and consumers (that is, the range $[1, D]$) into m sub-ranges. Table 1 shows that both over-segmentation and under-segmentation will result in high storage costs, and the optimal segmentation lies in between the two extremes.

The general information brokerage scheme presented here not only works for non-linear lookup functions, but also outperforms the previous scheme when the lookup cost function is linear. Recall that a scheme with linear lookup cost is one whose lookup cost function is $g(x) = s \cdot x$. The comparison is illustrated in Table 2, where the storage cost f and the l_i sequence are shown. The parameters are $D = 100$, and $s = 0.25, 0.5, 1, 2$.

	baseline		generic	
s	f	l_i	f	l_i
0.25	6729	1,2,4	6517.6	1,2.4,4.8
0.5	2614.5	1,2,4,8	2387.3	1,3,7.8
1	987.6	1,2,4,8,16	869.0	1,2.8,7.5,17.2
2	389.5	1,2,4,8,16,32	328.8	1,3.3,10.3,29.3

Table 2. The storage cost and i-resolution sequence of the baseline and generic GIB schemes.

The following are two observations on the data in Table 2. First, the storage cost of the generic GIB scheme is indeed consistently lower than that of the baseline GIB scheme. Second, the baseline GIB scheme tends to over-segment the distance range $[1, D]$ between producers and consumers. In particular, for $s = 1$ and $s = 2$, the baseline GIB scheme divides the range into 4 and 5 sub-ranges, while the generic GIB scheme achieves better performance by dividing it into only 3 and 4 sub-ranges.

5 Optimality of the Information Brokerage Scheme

In this section, we analyze the optimal performance of information brokerage for linear lookup cost. We show that for a wide family of networks called growth lower-bounded networks, the logarithmic storage cost is a lower bound for any information brokerage scheme that achieves linear lookup cost. Since growth lower-bounded networks includes the grid networks as a special case, our information brokerage achieves the asymptotically optimal performance.

The notion of *bounded growth* is based on the concept of *ball packing*. A *ball* $B_r(v)$ with center node v and radius r is the set of nodes within distance r from v. Let $\mathcal{B}$ be a set of balls each with radius r, and let $U \subseteq V$ be a subset of nodes in the network. $\mathcal{B}$ is called a *r-packing* of U if $\cup_{B \subseteq \mathcal{B}} B \subseteq U$ and $\forall B, B' \in \mathcal{B}$, $B \cap B' = \Phi$. A network is *growth lower-bounded* or simply *growth bounded* with growth rate α if for any $v \in V$ and any r, r' such that $r' \leq r$, there exists a r'-packing of at least $(r/r')^\alpha$ balls for $B_r(v)$.

Consider an arbitrary information brokerage scheme with linear lookup cost. Without loss of generality, we assume that the minimum distance between two nodes is one. Let s be the stretch factor in the linear lookup cost function $g(x)$. An important quantity in the following analysis is the *zooming factor*, which is defined as:

$$z = z(s, \alpha) = (1.5s)^{\frac{\alpha}{\alpha-1}}.$$

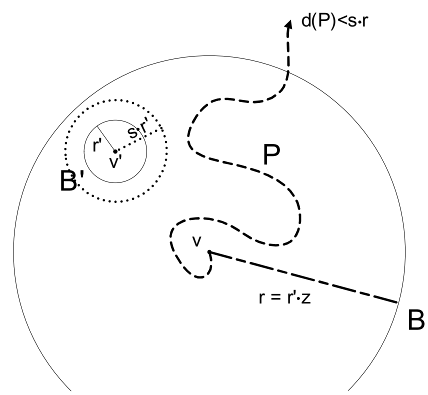

Figure 2. For any ball and any (lookup) path starting from its center, there is a big enough ball contained in it and the lookup path does not intersect it.

Before presenting the following lemma (Lemma 2), we introduce a few more terms. The distance $d(v, P)$ between a node v and a path P is the minimum distance between v and any node in P. We say that "a path crosses a ball" if the intersection of the path and the ball is non-empty.

Lemma 2. *Let v be an arbitrary node in a growth-bounded network. Let $P = (v, w_1, w_2, ...)$ be a path whose length $d(P)$ is at most $s \cdot r$. Then, there must exist a node v' such that $B_{r'}(v') \subseteq B_r(v)$ and $d(v', P) > s \cdot r'$, where $r' = r/z$.*

Proof. By the basic property of the bounded growth of the network, there is a $1.5sr'$-ball packing of cardinality

$$(\frac{r}{1.5sr'})^{\alpha} = (\frac{z}{1.5s})^{\alpha} = (\frac{1.5s^{\alpha/(\alpha-1)}}{1.5s})^{\alpha} = (1.5s)^{\frac{\alpha}{\alpha-1}} = z$$

for any ball of radius r. Therefore, there is such a packing for $B = B_r(v)$. Shrink the radius of each ball in this packing from $1.5sr'$ to sr', and we obtain an sr'-ball packing $\mathcal{B}$ for B of cardinality z. Clearly, the distance between any two balls in this packing is at least sr', as shown in Figure 3.

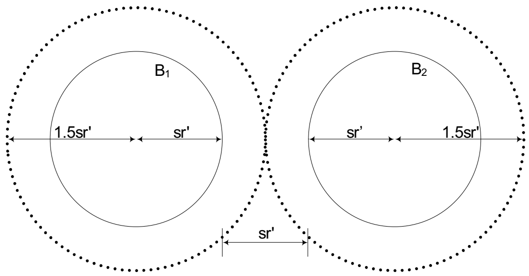

Figure 3. Two balls B_1 and B_2 in an sr'-ball packing obtained from a $1.5sr'$-ball packing, where the distance between B_1 and B_2 is at least sr'.

Consider the number balls in $\mathcal{B}$ that P can cross. Since $d(P) \leq s \cdot r$ and the minimum inter-ball distance of $\mathcal{B}$ is at least $s \cdot r'$, P may cross at most $s \cdot r/(s \cdot r') = r/r' = z$ balls in $\mathcal{B}$. Since there are at least z balls in $\mathcal{B}$, there is at least one ball that P cannot cross. Let v' be the center of this ball. Obviously $B_{s \cdot r'}(v') \subseteq B$ because $B_{s \cdot r'}(v')$ is a member of the ball packing $\mathcal{B}$. Also, $B' = B_{r'}(v') \subseteq B$ too because $B_{r'}(v') \subseteq B_{sr'}(v')$. By our choice of v', P does not cross $B_{sr'}(v')$, which implies that $d(u', P) > sr'$. Therefore, v' is the node we have been searching for. □

Theorem 2. *Let B be an arbitrary ball of radius R in a growth bounded network. For any information brokerage scheme with the linear lookup cost function $g(x) = s \cdot x$, there must exist a node $u \in B$ such that its storage cost is $\Omega(\log R)$.*

Proof. The proof is based on the following construction of a sequence of balls $B_1, B_2, ...$ with exponentially decreasing radius. See Figure 4 for an illustration. First, let v_1 be the center of B. Let $B_1 = B_{r_1}(v_1)$. Let P_1 be the maximum segment of v_1's lookup path such that $d(P_1) \leq s \cdot r_1$. By Lemma 2, there is a node v_2 such that ball $B_2 = B_{r_2}(v_2) \subseteq B_1$ and $d(v_2, P_1) > s \cdot r_2$, where $r_2 = r_1/z$. Let P_2 be the maximum segment of v_2's lookup path such that $d(P_2) \leq s \cdot r_2$. Notice that $P_1 \cap P_2 = \Phi$, that is, P_1 and P_2 are mutually disjoint, because $d(v_2, P_1) > s \cdot r_2$ and $d(P_2) < s \cdot r_2$. Again by Lemma 2, there is a node v_3 such that ball $B_3 = B_{r_3}(v_3) \subseteq B_2$ and $d(v_3, P_2) > s \cdot r_3$, where $r_3 = r_2/z$. By repeating this process in $I = \log R/\log z$ steps, we will have a sequence of nodes $v_1, v_2, ..., v_I$, a sequence of balls $B_1, B_2, ..., B_I$ and a sequence of paths $P_1, P_2, ..., P_I$. By the above construction, $B_I \subseteq B_{I-1} \subseteq ... \subseteq B_2 \subseteq B_1$, and $P_1, P_2, ..., P_I$ are mutually disjoint.

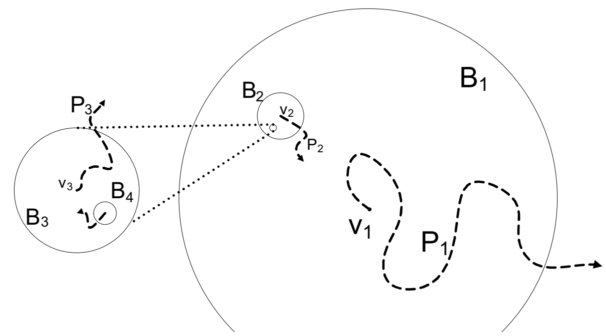

Figure 4. Non-intersecting paths $P_1, P_2, P_3...$, and balls $B_1 \supseteq B_2 \supseteq B_3...$ with exponentially decreasing radii.

We can now show that v_I is the node we are searching for. To verify this, let $u = v_I$ be a producer. It is straightforward to see that $\forall 1 \leq i \leq I$, $u \in B_i$ because $u \in B_I$ and $B_I \subseteq B_i$. The fact that $u \in B_i$ implies that $d(u, v_i) \leq r_i$. Since the lookup stretch factor is s, and P_i is the maximum segment of the v_i's lookup path satisfying $d(P_i) \leq s \cdot r_i$, u must have at least one mirror on P_i ($\forall 1 \leq i < I$) in order for v_i to locate it with a cost no more than $s \cdot d(u, v_i)$. Since $P_1, P_2, ...P_{I-1}$ are mutually non-intersecting, the total number of u's mirrors in B is at least $I - 1 = \log R/\log z - 1 = \Theta(\log R)$. □

Theorem 2 shows that for a growth-bounded network of diameter D, for any information brokerage scheme with linear lookup cost, the maximum number of storage cost for a node is $\Omega(\log D)$, which is $\Omega(\log n)$ for a (square) grid network, since the number of nodes n in it is $O(D^2)$. So we have the following theorem.

Theorem 3. *The information brokerage scheme GIB achieves asymptotically optimal storage cost for grid networks with linear lookup cost functions.*

We would like to note that this result also proves the optimality of a few other information dispersion methods in literature such as [1].

6 Concluding Remarks

This paper presents an information brokerage scheme GIB, whose lookup cost can be bounded by both linear and more general cost functions. To our best knowledge, this is the first that such general lookup cost functions are studied for locality-sensitive information brokerage. It proves that an logarithmic storage cost is asymptotically optimal for schemes with linear lookup cost, which meets the performance of our proposed scheme for grid networks. We are interested in studying the information brokerage schemes for more general network models, and for more flexible query functions. Those remain as our future research.

References

[1] Ittai Abraham, Danny Dolev, and Dahlia Malkhi. Lls: a locality aware location service for mobile ad hoc networks. In *DIALM-POMC '04: Proceedings of the 2004 joint workshop on Foundations of mobile computing*, pages 75–84, New York, NY, USA, 2004. ACM Press.

[2] Ittai Abraham, Dahlia Malkhi, and Oren Dobzinski. Land: stretch (1 + e) locality-aware networks for dhts. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 550–559, Philadelphia, PA, USA, 2004. Society for Industrial and Applied Mathematics.

[3] Baruch Awerbuch and David Peleg. Concurrent online tracking of mobile users. *SIGCOMM Comput. Commun. Rev.*, 21(4):221–233, 1991.

[4] Murat Demirbas, Anish Arora, Tina Nolte, and Nancy Lynch. Brief announcement: Stalk: a self-stabilizing hierarchical tracking service for sensor networks. In *PODC '04: Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*, pages 378–378, New York, NY, USA, 2004. ACM Press.

[5] Qing Fang, Jie Gao, and Leonidas J. Guibas. Landmark-based information storage and retrieval in sensor networks. In *INFOCOM '06: Proceedings of the 25th Conference on Computer Communications.*

[6] Roland Flury and Roger Wattenhofer. Mls:: an efficient location service for mobile ad hoc networks. In *MobiHoc '06: Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing*, pages 226–237, New York, NY, USA, 2006. ACM Press.

[7] Stefan Funke and Imran Rauf. Information brokerage via location-free double rulings. In *ADHOC-NOW '07: the 6th International Conference on Ad-Hoc, Mobile, and Wireless Networks*, volume 4686 of *Lecture Notes in Computer Science*, pages 87–100. Springer, 2007.

[8] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A scalable location service for geographic ad hoc routing. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 120–130, New York, NY, USA, 2000. ACM Press.

[9] C. Greg Plaxton, Rajmohan Rajaraman, and Andrea W. Richa. Accessing nearby copies of replicated objects in a distributed environment. In *SPAA '97: Proceedings of the ninth annual ACM symposium on Parallel algorithms and architectures*, pages 311–320, New York, NY, USA, 1997. ACM Press.

[10] Sylvia Ratnasamy, Brad Karp, Scott Shenker, Deborah Estrin, Ramesh Govindan, Li Yin, and Fang Yu. Data-centric storage in sensornets with ght, a geographic hash table. *Mob. Netw. Appl.*, 8(4):427–442, 2003.

[11] Antony Rowstron and Peter Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, 2001.

[12] Rik Sarkar, Xianjin Zhu, and Jie Gao. Double rulings for information brokerage in sensor networks. In *MobiCom '06: Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 286–297, New York, NY, USA, 2006. ACM Press.

[13] Ben Y. Zhao, John D. Kubiatowicz, and Anthony D. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and. Technical report, Berkeley, CA, USA, 2001.