Distance-Sensitive Information Brokerage in Sensor Networks

Stefan Funke[1] Leonidas J. Guibas[1] An Nguyen[1] Yusu Wang[2]

[1] Department of Computer Science
Stanford University, Stanford, CA 94305
[2] Department of Computer Science and Engineering
The Ohio State University, Columbus, OH 43210

Abstract. In a sensor network information from multiple nodes must usually be aggregated in order to accomplish a certain task. A natural way to view this information gathering is in terms of interactions between nodes that are *producers* of information, e.g., those that have collected data, detected events, etc., and nodes that are *consumers* of information, i.e., nodes that seek data or events of certain types. Our overall goal in this paper is to construct efficient schemes allowing consumer and producer nodes to discover each other so that the desired information can be delivered quickly to those who seek it. Here, efficiency means both limiting the redundancy of where producer information is stored, as well as bounding the consumer query times. We introduce the notion of *distance-sensitive information brokerage* and provide schemes for efficiently bringing together information producers and consumers at a cost proportional to the separation between them — even though neither the consumers nor the producers know about each other beforehand.
Our brokerage scheme is generic and can be implemented on top of several hierarchical routing schemes that have been proposed in the past, provided that they are augmented with certain key sideway links. For such augmented hierarchical routing schemes we provide a rigorous theoretical performance analysis, which further allows us to prove worst case query times and storage requirements for our information brokerage scheme. Experimental results demonstrate that the practical performance of the proposed approaches far exceeds their theoretical (worst-case) bounds. The presented algorithms rely purely on the topology of the communication graph of the sensor network and do not require any geographic location information.

1 Introduction

Early sensor networks were primarily data acquisition systems, where the information collected by the sensor nodes was to be aggregated and routed to a central base station. Newer generations of sensor networks, however, act more as peer-to-peer systems, where arbitrary nodes in the network may wish to collect information about measurements and events elsewhere in the network. Furthermore, the information needed may be quite specific, with only a very small amount of sensor data being relevant for any particular query. This peer-to-peer view is necessitated as sensor networks expand to serve multiple geographically dispersed users, as more powerful mobile nodes move through a static sensor network and use it as a communication backbone to issue queries and collect data, or even to process complex queries, where sensor nodes may find it necessary to issue sub-queries themselves. The basic problem this creates is that of *information brokerage*: how *producers* of information, e.g., nodes that have collected data, detected events, etc., and *consumers* of information, i.e., nodes that seek data or events of certain types, can find out about each other and exchange the desired information.

Information brokerage is closely coupled with node naming and routing: even if we know the exact location of the information we want in the network, we still need to discover a good path for its retrieval. As sensor networks scale to larger sizes, the issue of *information locality* becomes more important. It is natural to expect that a consumer will be more interested in data collected near its current location. This is because such data can be accessed at lower communication cost/delay, and because in almost all sensor network applications local information has higher value and relevance to the task at hand than remote information. The main problem studied in this paper is what we call *distance-sensitive information brokerage* — information brokerage where the cost for a consumer node to discover a certain piece of information is proportional to the network distance to where that information was collected. We want to have this property of *distance sensitivity* even though neither the consumer node nor the producer node involved in the information exchange know directly anything about the location of the other node in the network. Current common information brokerage schemes *do not* have this property. For example, directed diffusion [14] performs flooding and thus in a 2-D network will visit $O(d^2)$ nodes to reach a distance d from a consumer (sink), while geographic hash tables (GHT) [24] may hash the information quite far away from a nearby producer-consumer pair.

Our goal in this paper is to develop a unified framework for routing and information brokerage which can provide provable guarantees for both good (almost-optimal) routing paths and distance-sensitive information brokerage. Note that in all producer-consumer brokerage schemes there is a trade-off between the time and space effort for information diffusion when producer nodes record new data and have new detections, vs. the query time of consumer nodes to discover this information. In this work we aim at minimizing the amount of work/storage that producers have to invest so that they can be discovered within the consumers' budget, that is, to be *distance-sensitive*. We focus on the static case typical for sensor networks where nodes do not move during the lifetime of the network, though links may fail or nodes may die. We also assume that, at any one time, only a small fraction of the network nodes have information to be made available to the network (such as sightings/measurements of rather exceptional events).

Related work. Hierarchies for addressing and routing within networks have been used for a long time and form the basis of the standard TCP/IP protocols. The basic idea is to define a tree-like hierarchy of node clusters, based on the inter-node distances in the network. This tree structure is then used to derive unique addresses for all nodes, based on which local routing schemes can be developed. Many previous hierarchical approaches have designated nodes as *gateways* to route between clusters [1, 22], causing unbalanced node traffic, as well as making routing sensitive to gateway node failures. Furthermore, since hierarchies partition the network, there can be nearby nodes that end up in different clusters even at the top level, causing the routing paths to be possibly much longer than the true shortest paths, violating distance sensitivity. Solutions have been proposed for this difficulty by introducing cross-branch links in the clustering hierarchy by Tsuchiya [28] and others, and empirical studies have established their effectiveness. For the case where the underlying communication graph has constant doubling dimension [11], it has been shown very recently in [12] (although in a different context) that paths of bounded dilation can be achieved. Chan *et al.* [3] present a different hierarchical framework to achieve slightly better results, but their construction is based on a probabilistic argument and a non-trivial derandomization thereof.

Since sensor networks are embedded in the physical world, several routing schemes attempt to exploit naming and routing using this host space to gain efficiency and avoid expensive preprocessing. For example, geographic routing schemes [2, 15, 16] name nodes by their geographic coordinates and provide local rules for forwarding messages towards their target. Such schemes may have problems in the presence of holes, in which case packets might get stuck in local minima of the distance function to the target. Protocols like GPSR ([15]) or the one proposed by Bose et. al. [2] have been developed to alleviate these problems, and indeed they can guarantee delivery of the packets by performing a perimeter routing step around network holes, after an appropriate planarization of the network graph. Still, the paths might be considerably worse than the true shortest paths. In particular, Kuhn et. al. in [17] show that if the shortest path has distance d, *any* geographic routing algorithm might produce a path of length $\Omega(d^2)$. Furthermore, in many situations, it is challenging to obtain geographic coordinates. Various approaches have been developed for cases in which either a few nodes [26] or no nodes [6, 21, 23] are aware of the geographic positions. However, a robust routing scheme with proven guaranteed performance for sensor networks with arbitrary underlying topology is still missing.

At the same time, in the past few years, sensor networks have started to serve more as information processing mechanisms instead of simply as data collection tools [7, 14, 18, 25, 29], requiring more sophisticated operations, such as data aggregation and range queries. From this point of view, the location of the sensor that owns information becomes less important than the information itself. This explains the rise of various *data-centric* information storage and retrieval schemes [27]. A representative is the Geographic Hashing Table (GHT) approach [24], where each type σ of information (like the measured occurrence of substance A) is mapped to a specific node v by using a geographic hash function which depends only on the information type σ and is commonly known to every node in the network. Upon detection of A, a producer sends a message to node v, indicating its possession of some data of type σ. Any consumer can then obtain the data by first visiting v to find out who owns it and then retrieving it from the owner (many variations are possible). This node v, however, might be far away from both the producer and the consumer even when the producer and the consumer are very close in the network. The problem can be partly alleviated by the GLS [19] approach (originally proposed for providing location services on mobile nodes), where a producer performs an information diffusion process by sending a message to a *list* of server nodes determined by its location and the data type. A consumer can then retrieve this data in time proportional to the distance to the producer *in the underlying hierarchy*, which in the case of GLS is a positional quad-tree. But since — as in the case of hierarchy-based routing strategies — nearby nodes might be far away in the hierarchy, the consumer still does not experience distance-sensitive query times. Furthermore this assumes the availability of geographic location information and an auxiliary ID server structure that has to be precomputed to allow for information brokerage[3]. Very recently, an approach [5] combining Geographic Hash Tables with landmark-based routing via the GLIDER [6] scheme has been proposed. There are also several other approaches focusing on data aggregation, multi-resolutional storage, or database-like queries [10, 8, 20]. They all suffer from the above two problems, however.

Closest to our approach is the work of [4], where the authors developed a location/address lookup service called L^+ for mobile nodes in a landmark-based routing scenario aiming at distance sensitivity in the queries, that is, the time required to lookup the address of a target node should be proportional to its distance. Their construction could also be extended to an information brokerage scheme (though they did not do so). Their simulation results show the effectiveness of the approach, even in a mobile scenario, but no rigorous theoretical analysis was conducted, which this paper aims to provide.

Our contributions. In this paper we analyze augmented hierarchical decomposition schemes for routing and information brokerage built on only the topology of the communication graph. While such routing hierarchies have been around for many years and a related theoretical analysis has been very recently found in a different context [12], this paper is the first to present a *unified generic* framework for both routing and information brokerage, and this framework as well as the accompanying analysis can be applied to various implementations of hierarchical decompositions (such as those in [1, 9, 22]).

Our framework can be applied to networks with arbitrary communication graphs and guarantees distance-sensitivity of both routing as well as information brokerage. If the shortest path metric of the communication graph has bounded doubling dimension [11], we can guarantee that the routing tables that

[3] The GLS paper did not address the information brokerage application, though the ID service presented there, which provides a mapping of unique node IDs to geographic coordinates, can be seen as a special case of information brokerage.

need to be installed at every node have size only $O(\log n)$ bits. A metric space has *bounded doubling dimension* if any ball with radius R in the metric space can be covered by a constant number of balls with radius $R/2$. In practice, sensor networks frequently experience low-level link and node volatility. We show through simulations that our routing protocol is robust: it performs gracefully against the failure of a small fraction of network nodes, due to the absence of any backbone or hub structure.

Our information brokerage scheme is built on top of the presented routing structure at no extra overhead. To our knowledge, this is the first approach that works for arbitrary communication graphs and has theoretical performance guarantees in terms of the effort on both the producer and consumer sides. In particular, after the producer stores references to its data at a small number ($O(\log n)$ in case of metrics of bounded doubling dimension) of nodes (in a multi-resolution manner), any consumer can retrieve it in a distance-sensitive way. Furthermore, by visiting only $O(d)$ nodes, the consumer can collect all occurrences of a particular type of data within a neighborhood of radius d. This kind of *range query*[4] can be useful for implementing efficient in-network processing and data aggregation. We are not aware of any other scheme that efficiently supports this type of query. Our information brokerage scheme inherits both load-balanced information diffusion and robustness against node failures from the employed routing scheme. All these results are backed by an evaluation in simulation which indicates that the practical performance of the analyzed schemes is significantly better than their theoretical guarantees.

We would like to emphasize again that even though a hierarchy is used in our approach, nodes high up in the hierarchy do not get any additional load comparing to nodes in lower levels. During routing, nodes in the hierarchy act as landmarks toward which packets are forwarded, but they only stay as landmarks when the packets are still far away from them, so a breakdown of a node in some sense *does not* hinder its function as a routing waypoint. Just like any other nodes, the failure of a node in high level of a hierarchy only has a local effect on the routing and information brokerage capabilities of the sensor network.

Outline. In Section 2, we introduce the notion of a hierarchical decomposition as a generic hierarchical framework for organizing a sensor network, and then describe our routing scheme and an information brokerage system under this framework. The performance of our framework is experimentally evaluated in Section 3 by extensive simulations. Finally, we conclude and discuss possible extensions to our scheme in Section 4.

2 Distance-Sensitive Routing and Information Brokerage

In this section, we first introduce the notion of a *hierarchical decomposition* (HD) constructed on an arbitrary sensor network in which geographic coordinates of

[4] We remark that we use the term *range query* differently here from work like [8, 20], where 'range' refers to a range in data space and not in the space of sensor locations.

the nodes may not be available. As it becomes clear later, a HD captures all the necessary properties we need for routing and information brokerage. We then show how to use a HD of the communication graph for efficient routing and for distance sensitive information brokerage.

2.1 Hierarchical Decompositions of Graphs

In the following we consider an undirected, weighted, connected graph of n nodes with node distances induced by the shortest path metric. We call a tree H of height h a hierarchical decomposition (HD) of S if

- each node $c \in H$ is associated with a set of nodes $S_c \subseteq S$ (*cluster*),
- for the root $r \in H$ (which is at level $h-1$) we have $S_r = S$,
- all leaves of H have the same level 0,
- for any node $c \in H$ at level k, we have that the cluster S_c associated with c has diameter less than $\alpha \cdot 2^k$ for some constant α,
- if $c \in H$ has children $c_1 \dots c_l$, we have $S_c = \biguplus S_{c_i}$

In case of an unweighted graph (i.e. edge weights are all 1), the diameter of a connected n-node graph is at most n, and one can construct a hierarchical decomposition of height at most $h = 1 + \lceil \log n \rceil \leq 2 + \log n$. The following discussion focuses on this case for simplicity. We also remark that the constraint of all leaves being at the same level 0 is not mandatory and could be removed. We make this assumption for simplicity of the presentation and also because the hierarchical decomposition we use in our experiments has this property.

There are different ways to construct a HD. In our implementation we use a specific HD based on the discrete center hierarchy (DCH) from [9], which is also similar to the hierarchy in [12]. It can be built efficiently and distributedly on an arbitrary sensor network given only its connectivity graph (details omitted for lack of space).

2.2 Routing Using Hierarchical Decompositions

First we provide a routing scheme based on a hierarchical decomposition H of a communication graph. Important properties which we want to ensure are:

- **scalability:** the routing information that an individual node of the network has to store should be small compared to the network size; the load on the nodes should be distributed in a balanced fashion (so we disallow dedicated hub nodes or backbones).
- **efficiency:** the path generated by our routing scheme between nodes v and w should be only a constant factor worse than the optimal shortest path in the communication network.
- **robustness:** the impact of nodes or links failing should be limited and local. In particular, packets that get temporarily stuck due to node or link failures should be able to recover using local rules.

The scheme we provide has all these three properties.

An addressing scheme Let $\Delta_{\max}$ be the maximum degree of H. We number the children $c_1, \ldots c_o$ with $o \leq \Delta_{\max}$ of a node c arbitrarily, and define the following addresses for the nodes of the tree:

- the root r has as address the h-dimensional vector $f(r) := (1, 0, 0, \ldots, 0)$ (remember h is the height of H)
- a node $c' \in H$ at level k which is child c_i of parent c is assigned the address $f(c') := f(c) + i \cdot e_{h-k}$, where e_{h-k} is the h-dimensional unit vector with a 1 at the $(h-k)$-th position

Essentially this constructs an IP-type address for each node of the tree and hence also for each cluster in the hierarchical decomposition. The entries in the vector are bounded by the maximum number of children $\Delta_{\max}$.

Connecting levels and efficient routing Let us now extend the addressing scheme to an efficient routing protocol. For that we need the notion of *neighboring clusters* of a node:

Definition 1. *A cluster L at level k is called a* neighboring cluster *of a node v if there exists a node $q \in L$ such that $d(v, q) \leq \alpha \cdot 2^{k+1}$.*

Note that while the number of neighboring cluster of a given node in each level may be larger than the number $\Delta_{\max}$ in the decomposition tree, in 'well-behaved' decompositions, we expect this number to be small. In particular, when the hop distance between the nodes is a metric with bounded doubling dimension [11] and a DCH is used as a hierarchical decomposition, the maximum number λ_{max} of neighboring clusters of a node in any given level is a constant, and a node has at most a total of $O(h) = O(\log n)$ neighboring clusters. From now on, we assume this is the case (so $\Delta_{\max}$ and $\lambda_{\max}$ are constants) to simplify our presentation.

We let each node in the sensor network store its distances to all of its neighboring cluster. The routing of a message to a node w from v can then be done as follow. Node v first looks up its neighboring clusters and find the smallest cluster L containing w. v then locally forwards the message for w to *any* node closer to L than v.

We claim that the length of the path when the message arrives at the smallest cluster containing w is at most a constant times longer than the optimal, shortest path from v to w in the communication graph. If our hierarchical decomposition has singleton clusters associated with the leaves, this implies a complete path from v to w which is a constant factor approximation of the shortest path. In particular, we have the following result (proof omitted for lack of space).

Lemma 1. *Let v, w be two nodes in the network. Then the path generated by the above routing scheme from v to the cluster of the lowest level containing w has length at most $4 \cdot d_{vw}$ where d_{vw} denotes the shortest path distance between v and w in the communication graph.*

Since there are typically many close-to-shortest paths from a given node to a cluster, this routing scheme has a *natural robustness* against node or link failures. And even if none of the immediate neighbors is closer to the target cluster, inspecting a slightly larger *local* neighborhood most of the time results in a successful forwarding of the message towards the target cluster.

If every bit counts Assuming that the maximum number of neighboring clusters $\lambda_{\max}$ and the maximum number of children $\Delta_{\max}$ in the HD is a constant, each node has to store address and distance of $O(h)$ clusters. In case of a communication graph with unit edge weights and singleton clusters at the leaves, $h = \Omega(\log n)$ and hence each node has to store $\Omega(\log^2 n)$ bits: the address of a cluster has $h = \Omega(\log n)$ components and the bit-complexity of the distance value stored for a neighboring cluster might be $\Omega(\log n)$ as well.

If every bit of space is relevant, we can do still better. Let us first consider a more efficient way to store the addresses of neighboring clusters of a node v. The key observation here is the trivial fact that if at level $k-1$ some cluster L is a neighbor of v, then in level k its parent $p(L)$ is also a neighbor. That means if a node has already stored the address of $p(L)$ it can store the neighbor L at level $k-1$ at additional cost of only $\log \Delta_{\max}$ bits. Hence for constant $\lambda_{\max}, \Delta_{\max}$, the addresses of all neighbors at all levels can be stored using $O(\log n)$ bits. The need for $\Omega(\log n)$ bits per distance value per neighbor can easily be reduced to a constant by just remembering *one edge* to an adjacent node in the communication graph that is closer to the neighboring cluster instead of the actual distance.

Corollary 1. *The routing scheme can be implemented by storing* $O(\log n)$ *bits per node in the network.*

We remark that the above addressing scheme as well as the neighboring information can be computed efficiently by restricted flooding during the initialization stage. It only increases the construction cost slightly over that of DCH's.

2.3 Efficient Information Brokerage using Hierarchical Decompositions

Given some fixed HD H, we now show how to achieve efficient distance-sensitive information brokerage based on the routing scheme described above. Let $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ denote the discrete set of all data items possibly produced or queried in the sensor net. Some properties of a desirable information brokerage system include:

- **load balance:** no node should have the burden of providing lookup-information for many different types of data items;
- **efficiency:** references to a certain type of data should be stored at a small number of nodes, and the time required for node w to access data produced by node u should be proportional to the distance between u and w.
- **robustness:** failure of nodes or links should only increase the time to store or retrieve a certain data item, but not make storage/retrieval impossible.

Our information brokerage system exploits the routing structure described above and to meet these desiderata.

First assume that we have a hash function $\mu : \Sigma \times HD \to S$, such that given any data item $\sigma \in \Sigma$ and a cluster $L \in HD$, we can compute a unique sensor node $\mu(\sigma, L) \in S$ within this cluster. Furthermore, $\mu(\sigma, L)$ can be accessed from any node in cluster L (of diameter D) within $2D$ hops using our routing structure (we will describe one such hash function at the end of this section). We call $\mu(\sigma, L)$ the *information server* of data item σ in cluster L. Our information brokerage system relies on collecting and distributing some synopses of data items to a small set of information servers.

Information diffusion Suppose a node (producer) u has data item $\sigma \in \Sigma$. Recall that u is contained in h clusters of the tree H (i.e., its ancestors), one in each level. Let $L(u, d)$ be the cluster containing u at level d, and $L_d^1, \ldots, L_d^l$ the neighboring clusters of $L(u, d)$. The producer u sends a message (containing its own address and some synopses of σ) to the information server $\mu(\sigma, L_d^j)$ associated with each of these L_d^j's for all $0 \le d < h$, see Figure 1. If the maximum number of neighbors at each level and the maximum degree of H are constants, then a producer will store a synopses of σ at $O(h) = O(\log n)$ nodes. Since the routing structure already specifies how to access all these neighboring clusters, no extra per-node storage is required to implement the diffusion process.

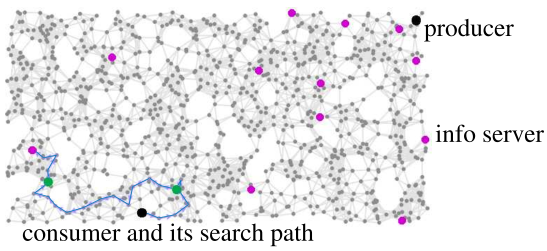


Fig. 1: There are exponentially fewer information servers away from the producer (upper right). The consumer (bottom center) looks for information at information servers in exponentially bigger clusters containing itself.

The length of the paths to the information servers decreases geometrically with decreasing level in the hierarchy. Hence the total number of hops to send synopses to all information servers, i.e., the *communication cost* for the producer, is dominated by the length of the paths to the information servers in the highest level of the hierarchy. For graphs of constant doubling dimension, this cost is in the order of the diameter of the network. We summarize this in the following lemma:

Lemma 2. *In the above diffusion scheme a producer of some data item $\sigma \in \Sigma$ distributes σ to $O(\log n)$ nodes in the network at a total cost of $O(D)$ hops, where D denotes the diameter of the network.*

Information retrieval When a consumer w wants to access some data item σ, it will look for it in growing neighborhoods, namely, in clusters $L(w,i)$'s in increasing order of i, where $L(w,i)$ denotes the ancestor of w at level i, see Figure 1. More precisely, it starts from w, and visits information servers $\mu(\sigma, L(w,1))$, $\mu(\sigma, L(w,2))$, ..., in sequential order to check whether the data item σ is there. Note that unlike the producer which sent out messages in different branches to various information servers, the consumer will only follow one path, and return as soon as it finds the information sought. The following lemma guarantees the distance sensitivity of our method (proof omitted from this extended abstract).

Lemma 3. *If node w wants to retrieve the data item $\sigma \in \Sigma$ associated with node u, this request can be completed in $O(d_{uw})$ time steps, where d_{uw} denotes the distance between u and w.*

We would like to emphasize that the constant in the O-notation experienced in practice is very close to 1. Furthermore it frequently happens that when a node observes an event, nearby nodes also observe the same event. To prevent multiple hashing for the same data item, each node can attempt to first retrieve the same item from its local neighborhood using the information retrieval process. If the item is not found, it can start the information diffusion process.

Approximate range counting and reporting Let $\mathrm{RC}(w, r; \sigma)$ denote the number of occurrences of data item σ detected by some sensor at most r hops away from w. Our information brokerage system can also be used to perform approximate range counting or range reporting for a consumer. In particular, we have the following lemma (proof omitted).

Lemma 4. *Let s be the number of distinct messages about σ received at node $\mu(\sigma, L(w,d))$, where $d = \lceil \log(r/\alpha) \rceil$, then $RC(w, r; \sigma) \le s \le RC(w, 4r; \sigma)$.*

In other words, by visiting the information server $\mu(\sigma, L(w,d))$ directly, a consumer w is guaranteed to collect all sources that have information about data item σ within roughly distance 2^d to itself. If the consumer only wants to know the number of such sources (range counting), it can simply return. Otherwise, if it wishes to report all such data (range reporting), it can then route to each of these sources and fetch the data.

Hash function $\boldsymbol{\mu(\sigma, L)}$ We still have to define the hash function $\mu(\sigma, L)$ that maps any given data item σ to an information server in a particular cluster L. Ideally, in order to have a good load balance, this function should distribute the information servers uniformly to all nodes contained in L for various $\sigma \in \Sigma$.

One possible choice of this function is deployed in the GLS approach, where each sensor node in the sensor network has a unique id (an integer). Given an data item $\sigma \in \Sigma$, assume there is a function that maps σ to one of this id, say ID_σ, randomly. The information server $\mu(\sigma, L)$ is then defined as the node $s \in L$ with smallest id that is greater than ID_σ. However, in order to identify and reach node s, one has to build extra structure (which needs about $O(h \log n)$ bits per-node memory) on top of whatever routing structure exploited.

A simpler way to define $\mu(\sigma, L)$ is to use σ as the seed of some pseudo-random function and traverse the HD tree downward randomly according to that function. $\mu(\sigma, L)$ is then defined as the leaf node reached in the process. Note that this definition of $\mu(\sigma, L)$ also gives a routing path to access it. We can first route to any node in L, using σ to determine the pseudo-random subcluster L' of L that contains $\mu(\sigma, L)$, then recursively route toward $\mu(\sigma, L') = \mu(\sigma, L)$. The number of hops to route from any node in L to $\mu(\sigma, L)$ is obviously bounded by $\sum_{i=1}^{d} \alpha \cdot 2^i \leq \alpha \cdot 2^{d+1}$.

Robustness The routing components of our information brokerage scheme inherit the robustness properties of the routing scheme; in particular, messages that are unable to make progress due to node or link failures can recover by simple *local* rules and be eventually forwarded to their destinations. Furthermore, recovery after failure of an information server is possible by querying the information server one level higher, incurring only a constant factor overhead.

3 Experimental Results

We implemented the discrete center hierarchy in java to experimentally evaluate the performance of our proposed schemes. Currently our implementation simulates the network at the graph level only. While it does not mimic network attributes like packet loss or delay, we are quite confident that the results reported here give a good indication about the usefulness of our approach in practical scenarios.

3.1 Data Generation and Implemented Algorithms

All our measurements were carried out on a unit disk graph of nodes that were spread *uniformly at random* in a unit square, and subsequently nodes may have been removed to simulate the presence of large holes in the network topology. Note that this "unit disk graph" is not required in our approach—our system can take an arbitrary graph as input. We also want to emphasize that in this model, as long as the average node degree is below about 10, a large fraction of the unit disk graph is not connected, and the dilation factor is rather large, i.e. the shortest path in the graph between two nodes is much larger than their Euclidean distance. This is quite different in the 'skewed grid' model where the node positions are determined by randomly perturbing points on a grid by some rather small amount. There the unit-disk graph is almost always connected for any grid-width slightly smaller than 1 (which corresponds to an average degree of 4 or more) and the geometric dilation is very small. See Figure 2 for an example. While our scheme performs much better in the skewed grid model as compared to the random model, we feel the latter provide more insight on realistic scenarios, and present all our results in this random model.

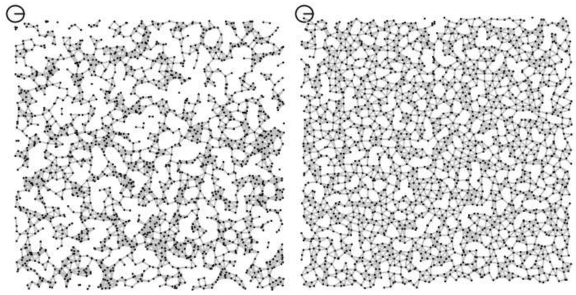


Fig. 2: Random sensor field (left) has degree 6.26 and has many holes, some of which are quite large. Skewed grid field (right) has lower degree, 5.18, yet is much more regular.

In the following subsections, we compare our routing scheme with GPSR, and our information brokerage scheme with GHT by assuming that the geographic locations of sensors are known (although not used in our approach). The underlying planar graph for GPSR is the Gabriel graph.

3.2 Evaluation of Routing Strategies

Routing quality. In the first experiment we evaluate the quality of the paths produced by our routing scheme. We fix a sensor field of 2000 nodes and vary the average degree from roughly 6 to 12. We then select at random 1000 pairs of nodes from the largest connected component of the sensor field. We compute the paths between the nodes in these pairs using the HD as well as using GPSR. For each of such path, its *quality* is defined as the ratio between its length and the shortest distance between its two endpoints. We show in Table 1 (a) the average and standard deviation of the quality of these 1000 paths. Note that our HD routing scheme always produces near-optimal paths regardless of the node density. The practical constant is much better than the worst case bound of 4 we could prove.

Avg.	Qual. of HD		Qual. of GPSR	
deg.	avg	std	avg	std
6.21	1.08	0.18	4.91	6.79
6.80	1.06	0.11	4.04	7.41
7.39	1.05	0.09	3.25	7.02
7.93	1.09	0.16	2.04	3.10
8.53	1.07	0.12	1.59	2.22
8.94	1.07	0.10	1.51	2.42
9.82	1.07	0.10	1.35	1.62
10.2	1.06	0.09	1.31	1.80
10.8	1.05	0.09	1.44	2.90
12.0	1.06	0.11	1.15	1.30

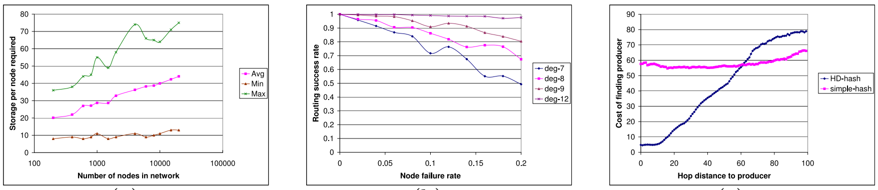
(a)

Size	# Info. servers		Query time		Total time	
	avg	std	avg	std	avg	std
200	10.8	0.42	1.22	0.82	2.28	1.31
400	17.0	0.21	0.93	0.63	1.91	1.06
600	28.7	1.36	0.74	0.59	1.73	0.92
1000	34.8	4.36	0.74	0.49	1.71	0.76
2000	50.9	8.22	0.73	0.45	1.76	0.70
4000	65.2	11.8	0.75	0.39	1.69	0.56
6000	75.7	13.5	0.78	0.43	1.79	0.64
8000	83.5	14.8	0.79	0.43	1.83	0.65
10000	85.9	15.5	0.77	0.38	1.77	0.60

(b)

Table 1: (a) Quality of paths from HD and from GPSR. (b) Performance of brokerage.

Network initialization and routing scalability. Here, we fix the average degree at roughly 6, vary the number of nodes from 200 to 20000 and compute the per-node storage for the HD routing structure. As expected, the number of entries in the routing table needed at each node grows very slowly, see Figure 3 (a). In particular, the maximal storage at a node in the network is quite reasonable, merely about twice the average per-node storage.



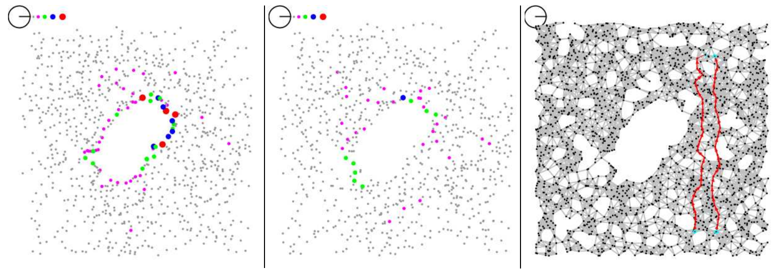
(a) (b) (c)

Fig. 3: (a) The storage required growths slowly when the network becomes large. (b) The success rate of routing vs. node depletion for sensor fields with various average degree. (c) Number of hops to information server using HD and number of hops in a shortest path to an ideally random information server.

We note that the cost of initializing the network, i.e., the number of messages sent to establish our hierarchical decomposition, is directly proportional to the storage at each node. As the storage cost is low, the cost of network initialization is also low.

Hot spots. Even though our implementation of the HD uses cluster heads, they are not special nodes in the network In a typical route, the moment a package heading towards a cluster reaches any of its nodes, the package starts heading towards a different cluster. So these cluster heads do not form a backbone structure, nor do they create bottlenecks in network traffic. Figure 4 (c) gives an example where two routes with nearby sources and destination nodes stay separate during their course. On the other hand, when large holes are present in the network, nodes close to holes will naturally have a heavier burden, as our HD paths approximate the shortest paths well. Still, our paths do not hug the holes in the sensor net as tightly as GPSR paths do, as shown in Figure 4 (a) and (b) for a sensor field with 2000 nodes and average degree 9.5. Our HD scheme produces many fewer higher load nodes (larger dots in the picture).

Robustness. To measure the performance of our routing scheme under node depletions, we start with a graph with average degree of $7, 8, 9$, or 12, build the HD routing structure on top of it, and then randomly remove a small percentage of sensors (from 2% to 20%) from the sensor field. We then pick 1000 pairs of live nodes at random, and show the success rate of routing between these pairs in Figure 3 (b). During the routing process, if a node finds that the next sensor on its shortest path to some cluster L is dead, it locally floods a neighborhood of nodes at most 5 hops away from itself to find a node with smaller distance to



(a) (b) (c)

Fig. 4: Hotspots comparisons for (a) GPSR and (b) HD scheme. Larger dots are nodes with higher traffic loads. In (c), two paths generated by HD scheme with nearby sources and destinations.

L. The result shows that the performance is gracefully degraded when the node failure rate increases.

3.3 Evaluation of the Information Brokerage Scheme

Brokerage quality. The efficiency of a brokerage system includes both the number of messages (i.e., # information servers) that a producer needs to replicate, and the number of hops that a consumer needs to access before locating the data it needs (i.e., the query time). In Table 1 (b), we vary the size of the sensor field from 200 to $10,000$ nodes. For each sensor field, we randomly choose $1,000$ producer/consumer pairs, with each pair producing/requesting a random data item. Columns 2,3 show the average number and standard deviation of information servers for each producer. Columns 4 and 5 show the quality of the path from the consumer to the information server, defined as the ratio between query time using our scheme (i.e., the path length to the respective information server) and the shortest hop distance between the producer and the consumer. We see from the table that this ratio is always close to 1.0 (in fact, in most cases it is smaller than 1.0, since the information server can be even closer than the producer). If the data is large, the producers may not replicate their data and only leave their addresses at the information servers. In that case, a consumer after locating the desired information server must further route to the producers to get the data itself. The quality of the brokerage path is then defined as the ratio between the path length from the consumer to the producer obtained using our scheme over the shortest path length between the consumer and producer. Column 6 and 7 in Table 1 (b) show this ratio in our experiment. In all cases, it is always small, around 2.0.

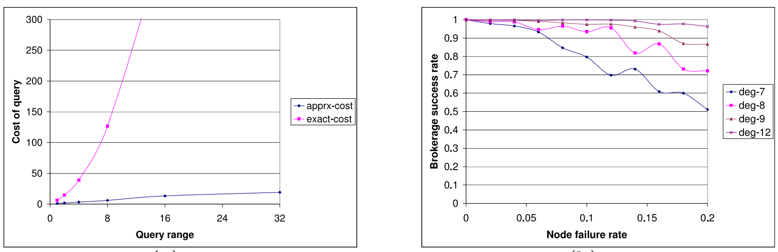
While the number of replications used by a particular producer is higher in our system than in the GHT approach, the query time can be much smaller, especially when the consumer is closer to the producer. This phenomenon is illustrated in Figure 3 (c), where we compare the query time in our scheme (lower curve) with the length of the shortest path to an ideally random information server (upper curve). Note that the latter is in fact a lower bound of the query time for any scheme using GHT for information brokerage. The query time for GHT using GPSR as the underlying routing scheme may be much longer than this shortest path, due to the path quality returned by GPSR. In short, our system is attractive for scenarios where there are multiple queries for the same data, as the overhead for the producer is then amortized.

It is also important to keep the distribution of information servers for different data items as uniform as possible. To test this, we let each sensor in the network produce a different data item, and record for each node the number of times that it serves as an information server for some data item. The results are in Table 2. The distribution of information servers observed is reasonably good compared to a distribution obtained by a centralized uniformly random hash function.

Size	200	400	600	1000	4000	6000	8000
Avg.	10.0	16.0	25.5	31.5	47.2	61.2	69.7
Std.	14.4	28.1	36.4	52.7	75.4	90.8	99.7

Table 2: The average/standard deviation of the number of times that a sensor serves as an information server for some data item for various network sizes.

Approximate range counting. One important application of our information brokerage system is for approximate range counting, such as reporting all horses detected within some distance r from a particular sensor. When r is quite small, flooding is simple and effective. However, the number of nodes accessed in the flooding approach increases quadratically as r increases. This is illustrated in Figure 5 (a) where the query cost in our approach (lower curve) increase in a linear manner, while that for flooding (upper curve) increases quadratically. The size of the sensor field in this example is $2,000$ with an average degree of 6.1.



(a) (b)

Fig. 5: (a) Approximate query cost is very low compared to the naive flooding. (b) The success rate of information brokerage under nodes depletions.

Robustness. Again, we fix a sensor field of 2000 nodes with various average degree, compute the HD routing structure, and remove a portion of sensors randomly (from 2% to 20%) from the field. We then randomly choose a set of producer/consumer pair, each generating/seeking a random data item. The resulting success rates are shown in Figure 5 (b). The brokerage system is slightly more robust than the routing scheme, which is not surprising: as the robustness of our information brokerage system comes partly from the robustness of the routing scheme, and partly from the fact that even if a query fails to route to a particular information server, it can simply go to the information server one level up.

4 Conclusion and Discussion

In this paper we have presented a unified framework for efficient routing and distance-sensitive information brokerage based on augmented hierarchical decompositions of communication graphs. In particular, for communication graphs of constant doubling dimensions, the guarantee for almost optimal routing paths comes at an additional cost of only $O(\log n)$ bits of storage per network node. Our routing scheme does not rely on a dedicated backbone or hub structure, and hence performs rather well when some network nodes fail while providing a natural load balance between routing paths.

Our novel *distance-sensitive information brokerage* scheme is built based on the above routing scheme. We showed how information producers can diffuse pointers to their information to $O(\log n)$ other locations and in turn how information consumer can exploit this stored data to retrieve the information they want in a distant sensitive manner. As an application, our brokerage infrastructure allows for range queries with specified radius that take time proportional to the radius instead of time proportional to the area of the relevant range region. All our procedures come with rigorous proofs of their worst-case performance guarantees, and the experimental results for both routing and information brokerage show considerably better performance than the worst case considerations in the theoretical analysis, (but the latter in some way explain this good behavior in practice).

In future work, it might be interesting to view the problem also from a producer's perspective. In particular, we can try to trade off the producer's effort to make its information available against the consumer's effort to obtain that information. The exact tradeoff can depend on the relative frequencies of data collection operations vs. queries, in the style of [13]. Furthermore, even though the main focus of this paper has been the static case where sensor nodes do not move over the lifetime of the network, it might be interesting to extend our approach to allow for efficient routing and information brokerage in the presence of mobile sensor nodes. Also we believe that the use of our distance-sensitive range queries can lead to interesting new in-network data-aggregation and processing algorithms.

Acknowledgements

The authors would like to thank the anonymous referees for their constructive comments. The authors gratefully acknowledge the support of DoD Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under Grant N00014-00-1-0637, NSF grants FRG-0354543 and CNS-0435111, and the Max Planck Center for Visual Computing. The 2nd author also wishes to thank Scott Shenker for many useful discussions.

References

1. E. M. Belding-Royer. Multi-level hierarchies for scalable ad hoc routing. *Wireless Networks*, 9(5):461–478, 2003.
2. P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. *Wireless Networks*, 7(6):609–616, 2001.
3. H. T.-H. Chan, A. Gupta, B. M. Maggs, and S. Zhou. On hierarchical routing in doubling metrics. In *SODA '05: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 762–771, Philadelphia, PA, USA, 2005. Society for Industrial and Applied Mathematics.
4. B. Chen and R. Morris. L+: Scalable landmark routing and address lookup for multi-hop wireless networks, March 2002.
5. Q. Fang, J. Gao, and L. J. Guibas. Landmark-based information storage and retrieval in sensor networks. In *Proc. of the 25th Conference of the IEEE Communication Society (INFOCOM'06)*, April 2006.
6. Q. Fang, J. Gao, L. J. Guibas, V. de Silva, and L. Zhang. GLIDER: Gradient landmark-based distributed routing for sensor networks. In *24th Conference of the IEEE Communication Society (INFOCOM)*, 2005.
7. W. F. Fung, D. Sun, and J. Gehrke. COUGAR: The network is the database. In *SIGMOD Conference*, 2002.
8. J. Gao, L. J. Guibas, J. Hershberger, and L. Zhang. Fractionally cascaded information in a sensor network. In *3rd Int'l. Sympos. Information Processing in Sensor Networks (IPSN)*, pages 311–319, 2004.
9. J. Gao, L. J. Guibas, and A. Nguyen. Deformable spanners and applications. In *Proc. of the 20th ACM Symposium on Computational Geometry (SoCG'04)*, pages 179–199, June 2004.
10. B. Greenstein, D. Estrin, R. Govindan, S. Ratnasamy, and S. Shenker. DIFS: A distributed index for features in sensor networks. In *1st IEEE International Workshop on Sensor Network Protocols and APplications Anchorage*, 2003.
11. A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proc. IEEE Symposium on Foundations of Computer Science*, 2003.
12. M. Herlihy and Y. Sun. Distributed transactional memory for metric-space networks. In *Proc. International Symposium on Distributed Computing (DISC 2005)*, pages 324–338, 2005.
13. Y. Huang and H. Garcia-Molina. Publish/subscribe in a mobile enviroment. In *MobiDe '01: Proceedings of the 2nd ACM international workshop on Data engineering for wireless and mobile access*, pages 27–34, New York, NY, USA, 2001. ACM Press.

14. C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.*, 11(1):2–16, 2003.
15. B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *6th ACM Int'l. Conf. Mobile Computing and Networking (MobiCom)*, pages 243–254, 2000.
16. F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric ad-hoc routing: of theory and practice. In *PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 63–72, New York, NY, USA, 2003. ACM Press.
17. F. Kuhn, R. Wattenhofer, and A. Zollinger. Asymptotically optimal geometric mobile ad-hoc routing. In *Proc. Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial-M). ACM Press*, pages 24–33, 2002.
18. J. Kulik, W. Heinzelman, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *5th ACM MOBICOM*, Seattle, WA, August 1999.
19. J. Li, J. Jannotti, D. S. J. D. Couto, D. R. Karger, and R. Morris. A scalable location service for geographic ad hoc routing. In *6th ACM Int'l. Conf. Mobile Computing and Networking (MobiCom)*, pages 120–130, 2000.
20. X. Li, Y. J. Kim, R. Govindan, and W. Hong. Multi-dimensional range queries in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 63–75, New York, NY, USA, 2003. ACM Press.
21. J. Newsome and D. Song. GEM: Graph embedding for routing and data-centric storage in sensor networks without geographic information. In *1st Int'l Conf. Embedded networked sensor systems*, pages 76–88, 2003.
22. R. Ramanathan and M. Steenstrup. Hierarhically-organized, multihop mibile wireless networks for quality-of-service support. *Mobile Networks and Applications*, 3(1):101–119, 1998.
23. A. Rao, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *9th ACM Int'l. Conf. Mobile Computing and Networking (MobiCom)*, pages 96–108, 2003.
24. S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. GHT: A geographic hash table for data-centric storage in sensornets. In *1st ACM Workshop on Wireless Sensor Networks and Applications*, pages 78–87, 2002.
25. N. Sadagopan, B. Krishnamachari, and A. Helmy. Active query forwarding in sensor networks. *Ad Hoc Networks*, 3(1):91–113, 2005.
26. A. Savvides, C. C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *7th ACM Int'l. Conf. Mobile Computing and Networking (MobiCom)*, pages 166–179, 2001.
27. S. Shenker, S. Ratnasamy, B. Karp, R. Govindan, and D. Estrin. Data-centric storage in sensornets. *ACM SIGCOMM Computer Communication Review*, 33(1):137–142, 2003.
28. P. F. Tsuchiya. The landmark hierarchy: a new hierarchy for routing in very large networks. In *Proceedings on Communications architectures and protocols*, pages 35–42, 1988.
29. F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information processing approach.* Elsevier/Morgan-Kaufmann, 2004.