Distance-Sensitive Routing and Information Brokerage in Sensor Networks

Stefan Funke   Leonidas J. Guibas   An Nguyen   Yusu Wang
Department of Computer Science
Stanford University, Stanford, CA 94305
E-mail: sfunke, guibas, anguyen, yusu@graphics.stanford.edu.

***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 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 sent quickly to those who seek it. Here, efficiency is an issue for both the producers (limiting the redundancy of where information is stored) as well as the consumers (keeping the query time low). 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 the consumers do not know the locations of the producers they seek. Our algorithms rely purely on the communication topology of the sensor network and do not require any geographic location information. In the process we introduce a new routing scheme that is of interest in its own right because it provides constant-factor approximations to the optimal paths. We give theoretical proofs of the efficiency of our scheme, as well as experimental results that further demonstrate its performance and suggest its practicality even for large scale sensor networks.**

I. 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 communications backbone to issue queries and collect data, or even in processing 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 of certain types, can find out about each other and exchange the desired information.

Information brokerage is closely coupled to 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 over which to retrieve it. As sensor networks scale to larger sizes, the issue of *information locality* becomes more important. It is natural to expect that a consumer node seeking certain types of data will be more interested in the availability of such data collected near its current location. This is because such data can be expected to be available at lower communication cost/delay, but also 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 relevant information is proportional to the network distance to where that information was collected. We want to have this property 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. Directed diffusion [13] 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. Further comparisons appear in the related work section.

We propose a scheme for accomplishing this distance-sensitive information brokerage based on a landmark-based hierarchical clustering of the network that is extended with certain 'sideways' links. A byproduct of this clustering is a hierarchical node naming scheme that is of interest in its own right. For a network of n nodes, we are able to keep routing information of size $O(\log n)$ per node so that we can do point-to-point routing in the network using our hierarchical names. Furthermore, we can guarantee both message delivery and *distance-sensitivity*: we always find a path, and the length of that path is provably to within a small constant factor of the true shortest path in the network (in our experiments the actual ratio is very close to 1).

Although cluster hierarchies have been used many times in network problems (think TCP/IP) and go back for decades, the one presented in this paper has a number of unique properties that make it appealing. First of all the hierarchy is built using only connectivity information among the nodes; we do not assume that nodes know their geographic coordinates (as is the case in GHTs)—so it is not a spatial hierarchy. Each node effectively names itself based on its network neighborhood, in a multiresolution fashion. Second, we are not aware of any previous work that can guarantee constant-factor approximation in the quality of the hierarchy routing paths. The main innovation that makes this possible in our case are the sideways links mentioned above—these bypass the problem that in any hierarchy pairs of nodes separated by the top level cuts can be near in the network graph sense but far in terms of the hierarchy. Finally, even though a hierarchy is involved, our routing and brokerage scheme is quite load-balanced and robust. With networks consisting of small untethered nodes, such as sensor networks, power conservation is always an issue. In such settings, hierarchies are always suspect, because it is natural to expect that nodes higher up in the hierarchy will carry a disproportional amount of the system load. Though the names are hierarchical, the way information storage and routing work in our system prevents load imbalance from occurring—a landmark node high in the hierarchy is used as a shared way point only when we are sufficiently far from it. For the same reason, the failure of a node high up in the hierarchy does not affect many more paths compared to the failure of a node in a much lower level. In fact, our routing scheme allows a packet that gets stuck temporarily due to node or link failure to recover and be routed towards its destination following only *local rules*—this provides for good robustness and performance in case of network volatility.

In all producer-consumer brokerage schemes there is a trade-off between the time and space cost of information diffusion when producer nodes record new data and have new detections, vs. the query time cost that consumer nodes have to pay to discover this information. In this work we try to minimize the amount of work/storage that producers have to invest so that they can be discovered within a consumers' budget—that is so as to meet the *distance-sensitive information brokerage* condition. 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; furthermore we assume that only a small fraction of the network nodes have information to be made available to the network (corresponding to sightings/measurements of rather exceptional events).

Related Work

Hierarchies for addressing and routing within networks have been used for a long time and form of course the basis of the standard TCP/IP protocols. The basic idea is to define a tree-like hierarchy of clusters based on the inter-node distances in the network. Then this tree structure is used to derive unique addresses for all nodes, and local routing schemes can be based on these coordinates. Many previous hierarchical approaches have designated nodes as *gateways* to route between clusters [2], [22], causing unbalanced node traffic, as well as making routing sensitive to node failures. There is also an inherent problem with such approaches as the resulting paths might be arbitrarily longer than the shortest paths, if two nearby nodes are put into different clusters at a high level in the cluster hierarchy (see Figure 2). Solutions have been proposed to overcome this problem by introducing cross-branch links in the clustering hierarchy (e.g. [28]), but still no rigorous theoretical analysis proving that the resulting paths are almost optimal has been conducted.

Due to these shortcomings and the fact that some preparation or initialization of the network is necessary to allow for addresses and routing information to be established, other routing schemes have been proposed. We give a brief review here; see [30] for a more detailed survey. Popular examples are so-called geographic routing schemes ([3], [14], [15]), which provide local rules based on the geographic position of the network nodes to forward messages towards their target. These schemes have problems in the presence of holes since in such cases, packets might get stuck in local minima of the distance function to the target. Schemes like GPSR ([14]) or the one proposed by Bose et al [3] 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 [16] 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 [5], [21], [23] are aware of the geographic positions. However, a robust routing scheme with guaranteed performance for 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 [6], [13], [17], [25], [31], 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 (substance A) and is commonly known to every node in the network. Upon detection of A, a producer will then send a message to node v, indicating its possession of some data of type σ. Any consumer can then obtain these data by first visiting v to find out who owns it and then retrieving the information 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 in the GLS [18] 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 type of information. A consumer can then retrieve this information 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[1]. There are also several other approaches focusing on data aggregation, multi-resolutional storage, or database-like queries [10], [8], [19]. They all suffer from the above two problems, however.

A recent study [4] has analyzed GLS, GHT, and other protocols for their performance when employed as *location services* in *mobile ad hoc networks*; it shows that GLS exhibits a rather large overhead induced by the hierarchy involved, especially in highly volatile network environments even for networks of moderate size. We want to emphasize that the scenario we are considering is quite different: we assume a static network of sensor nodes which do not move, although nodes or links may die; we also assume that relatively few nodes (producers) have information to be made available to the network (corresponding to sightings/measurements of rather exceptional events). This is in contrast to the hierarchy use as location service, where essentially *every* node in the network has information (namely its location) to distribute in the network.

In very recent work submitted to this conference [1], an approach combining Geographic Hash Tables with landmark-based routing via the GLIDER [5] scheme has been proposed. Different from our method, their solution is based on a two-level decomposition of the network and does not provide guarantees for storage or query times. Whether our hierarchical method or their approach based is more suitable depends on the topology of the communication graph as well as the structure and statistics of data collection operations vs. queries.

Distance-Sensitive Information Brokerage

In this paper we develop an augmented hierarchical decomposition scheme for routing and information brokerage based simply on the topology of the communication graph. While similar routing hierarchies have been around for many years, this paper is the first to give provable guarantees on not only the reachability, but also the quality of the produced paths—as well as good estimates on the storage requirements in the network to support the routing scheme. In particular, we prove that the point-to-point routing path produced by

[1]We remark that the original GLS paper did not directly address the information brokerage application, but their approach could be extended to do so; the ID service presented in their paper, which provides a mapping of unique node IDs to geographic coordinates, can be seen as a special case of information brokerage where σ is simply the node ID.

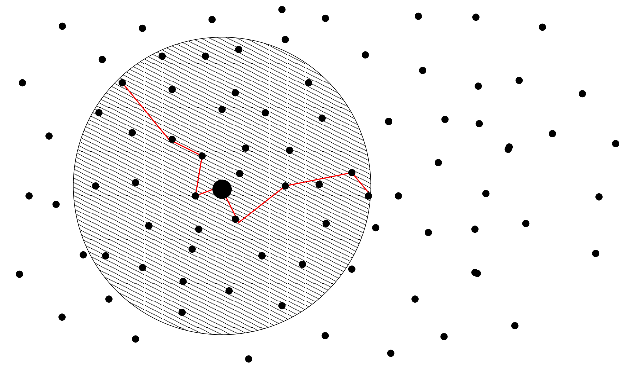


Fig. 1: Information about the interior of the shaded disk can be gathered by inspecting only 8 nodes (a number proportional to the radius of the disk) instead of all 38 nodes contained in the disk.

our scheme is always within a small constant factor of the shortest hop distance between the source and the destination (i.e., it is *distance-sensitive*). Our routing scheme requires only $O(\log n)$ bits per node to store auxiliary routing information. Furthermore it performs gracefully against failure of a limited number of network nodes, due to the absence of any backbone or hub structure. The techniques we use to provide guaranteed-quality paths as well as stability against network failures are fairly general and can also be applied to other hierarchical decompositions provided certain natural conditions are satisfied. In particular, existing decompositions like [2], [22] can be augmented using our techniques.

The information brokerage scheme is built on top of the above routing structure at no extra overhead. To our knowledge, just as for routing, it is also the first approach that works for arbitrary communication graphs and also provides theoretical performance guarantees in terms of effort required both on the producer and consumer sides. In particular, after the producer stores references to its data at a small number of nodes (in a multi-resolution manner), any consumer can retrieve the information 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 (Figure 1). This kind of *range query*[2] can be useful for implementing efficient in-network processing and data aggregation. We are not aware of any other scheme that efficiently support this type of query. Similar to the routing scheme, our information brokerage benefits from the topology-based hierarchical decomposition and the absence of any backbone or hub structure, thus exhibiting balanced information diffusion and robustness against node failures.

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 much lower levels. During routing, nodes in the hierarchy act as landmarks toward which packets are forwarded, but they

[2]We want to note that we use the term *range query* differently here from work like [8], [19], where the 'range' refers to a range in data space and not in the space of sensor locations.

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 landmark. Just like any other nodes, the failure of a node in high level of a hierarchy only has local effect on the routing and information brokerage capabilities of the sensor network.

All our theoretical results are backed by an experimental evaluation which indicates that the practical performance of our proposed schemes is significantly better than their theoretical guarantees.

Outline

The paper is organized as follows. We introduce the notion of a hierarchical decomposition as a generic hierarchical frame work for organizing a sensor network in Section II. We then propose an efficient and robust routing scheme, and an information brokerage system with guaranteed performance under the hierarchical decomposition framework in Section III. The performance of our routing and information brokerage system is experimentally evaluated in Section IV by extensive simulations. Finally, we conclude and discuss possible extensions to our scheme in Section V.

II. Hierarchical Decompositions and Discrete Center Hierarchies

In this section, we introduce the notion of a *hierarchical decomposition* (HD) constructed on an arbitrary sensor network in which geographic coordinates of 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 describe how to construct the HD we use in this paper efficiently and distributedly on an arbitrary sensor network given only its connectivity graph. The construction uses the discrete center hierarchy from [9].

A. 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 will mainly focus on that case. We also remark that the constraint of all leaves being at the same level 0 is not mandatory and could be removed, but we make this assumption for simplicity of the presentation and also because the hierarchical decomposition we are going to use will have this property.

B. The Discrete Center Hierarchy

When the location of each node in a sensor network is known, a quadtree based on geometric location of the nodes, as used in GLS [18], gives a hierarchical decomposition. Each cluster in the decomposition is the set of nodes in some square tile of the quadtree.

We are interested in a hierarchical decomposition of a sensor network in which no geometric information is available. Given only the connectivity of the nodes in a sensor network, we borrow the concept of discrete center hierarchy (DCH) from [9] and [7] to get a hierarchical decomposition. For a set of nodes R, a set of *discrete centers* with radius r from R is defined as a set $R' \subset R$ such that the hop distance between any pairs of nodes in R' is more than r and any nodes in R is within r hops from some node in R'. Intuitively, the set of discrete centers is a good sampling of the original set at a certain scale. For any given node x in R, there may be more than one node in R' that covers x. We select arbitrarily one node x' among those and call x' the *parent* of x and x a *child* of x'.

Given a sensor network S, we construct the *levels* $S_0, S_1, \cdots$ as following. Set $S_0 = S$, and for each $i > 0$, let S_i be a set of discrete centers of S_{i-1} with radius 2^i, see Figure 2. Note that once $|S_i| = 1$ then $|S_j| = 1$ for all $j \geq i$, so we can stop the construction as soon as we encounter a level with only one node in it. Note also that if a node is in some level S_i then it is also in all levels S_j where $0 \leq j \leq i$. To avoid ambiguity, we call a node a *center* when referring to it as a node in some specific level.

The parent-child relationship of the discrete centers induces a tree structure in the set of centers $\cup_{0 \leq i < h} S_i$. We call this tree a *discrete center hierarchy* (DCH) of the sensor field. We also call S_i where $0 \leq i < h$ the *levels of the DCH*.

Let s be a center in S_i. As all children of s in S_{i-1} are within a hop distance of at most 2^i from s, it is easy to see that all descendant of s are within at most 2^{i+1} hops from s. Thus,

Lemma 1 *The descendants of a center in level i form a cluster of diameter 2^{i+2}.*

Corollary 1 *A DCH of a sensor network is a HD in which each cluster in the decomposition is the set of descendants of some center in the DCH.*

C. Construction of the DCH

The DCH can be constructed in a bottom-up manner. We assume that each node in the sensor network has a unique ID. During the construction, each node maintains the highest level in which it participates, and a routing table to help with the routing during the construction. Each node initially has an empty routing table and has the highest level of 0.

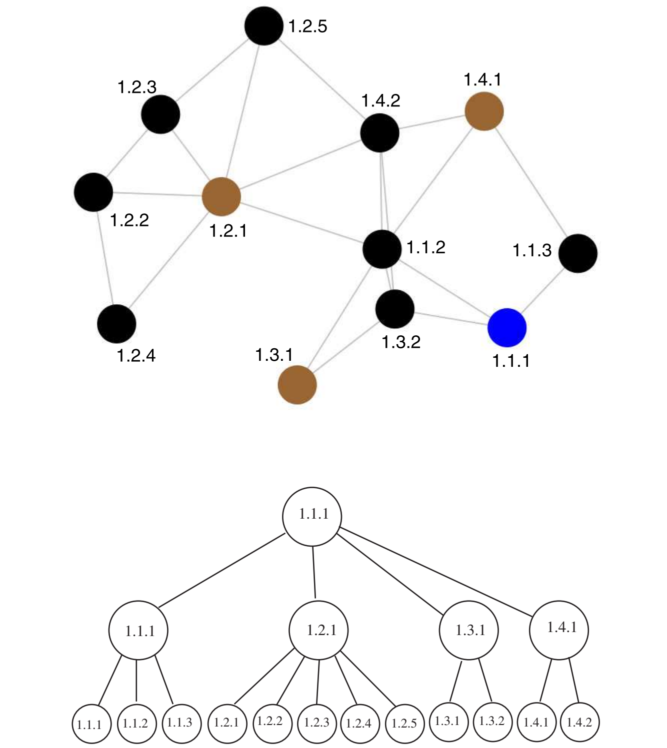


Fig. 2: A DCH on a sensor network and the naming of network nodes using that DCH.

The DCH is constructed in rounds. In round i, suppose that S_i has already been constructed, and we wish to compute level S_{i+1}. We let each node in S_i to do a restricted flooding, sending out its ID and its highest level to all nodes within 2^{i+1} hops from it. During the flooding, each node in the sensor network records to its routing table the IDs it receives together with the hop distances from the nodes with those IDs. Once the flooding is done, each node in the sensor network discovers all nodes in S_i that are within 2^{i+1} hops from it, and it can route messages to those nodes using routing tables stored in the network. The nodes in S_i can then employ some leader selection algorithm [20], [29] to select a maximal subset S_{i+1} of S_i that are at least 2^{i+1} hops away from each other.

For all practical sensor networks, the hop distance function between the nodes is a metric with small (bounded) doubling dimension [11]. For such networks, as the nodes in S_i are at least 2^i hops away from each other, each node in the sensor field is within 2^{i+1} hops from only a small constant number of nodes in S_i. The cost (per node in the sensor network) of flooding and selecting nodes in S_{i+1} is thus constant. Given that there are only at most $\log n + 2$ levels in a sensor network, the cost per node of the DCH construction is only a constant multiple of $\log n$. Experiment results show that the constants involved are quite small.

III. Routing and Information Brokerage

Given a hierarchical decomposition of the communication graph, we show that we can use it for efficient routing, yielding paths that are guaranteed to be within a constant factor of the shortest paths, and for distance sensitive information brokerage.

A. Routing Using Hierarchical Decompositions

In the following we will 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; furthermore 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 will ensure these three properties. In particular, each node will need to store only $O(\log n)$ items of information locally and paths generated are at most 4 times longer than the optimal shortest paths in the network. Note that the quality of the paths is much better in our experimental evaluation than these theoretical bounds indicate: most paths generated are only slightly longer than the shortest path possible.

1) 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, see Figure 2. The entries in the vector are bounded by the maximum number of children—a small number in general.

2) 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 $d(v, q) \leq \alpha \cdot 2^{k+1}$.*

A node v maintains h lists of *neighboring* clusters, one for each level in the hierarchy. The number of neighbors typically is larger than the number $\Delta_{\max}$ in the decomposition tree, but 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 number of neighbors is a constant.

For each of the neighboring clusters, v stores its address as well as the minimum distance to the boundary (nearest node) of the respective cluster. So if $\lambda_{\max}$ denotes the maximum number of neighbors a node might have in a level, each node v has to store an address and distance to at most $h \cdot \lambda_{\max}$ clusters. If the underlying communication graph is a unit-disk graph, we have $h = O(\log n)$ and for constant $\lambda_{\max}$, every node has to store information about $O(\log n)$ clusters.

How can some node v use this information to route a message to node w? Node v inspects its lists of neighboring clusters by increasing level. Remember that the address of a cluster at level i has the first $(h - i)$ entries non-zero and the last i entries set to 0. Let k be the level where the non-zero prefix of one of the neighboring clusters in v's list is also a prefix of the target address. That is, we have a neighboring cluster $C_N^{(k)}$ at level k with address $f(C_N^{(k)}) = (a_1, a_2, \ldots, a_{h-k}, 0, \ldots, 0)$ and $f(w) = (a_1, a_2, \ldots, a_{h-k}, a_{h-k+1}, \ldots a_h)$. Let $C^{(k)}$ be the cluster at level k that contains v. Node v now sends the message together with the target address towards the boundary of $C_N^{(k)}$. This can be done greedily by always choosing an adjacent node that is closer to $C_N^{(k)}$, since all nodes in $C^{(k)}$ (and possibly other clusters on the way to $C_N^{(k)}$) have stored this information. Since there are typically several neighbors that are closer to the target cluster, this provides for 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. Let v' be the first node that this route leads to in cluster $C_N^{(k)}$. We have that $f(v') = (a_1, a_2, \ldots, a_k, a_{k+1}, \ldots)$, that is $f(v')$ and $f(w)$ agree now in at least the first $k+1$ entries. Then recursively, using the same rules, inspecting v''s list etc., the message is sent further towards the destination w. 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.

Lemma 2 *Let v, w be two nodes in the network. Then the path generated by the above routing scheme from v to the cluster of 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.*

Proof: Let μ_{vw} denote the length of the generated path from v to the smallest cluster containing w. We now show $\mu_{vw} \leq 4 \cdot d_{vw}$. Assume node v has identified a neighboring cluster $C_N^{(k)}$ at the lowest level k. Since $C_N^{(k-1)}$ is not neighboring cluster of v, we have that $d_{vw} > \alpha \cdot 2^k$. If we have stored minimum distances to neighboring clusters, the first leg of the generated path which forwards the message to the boundary of the neighboring cluster $C_N^{(k)}$ has length at most $\alpha \cdot 2^{k+1} \leq 2 \cdot d_{vw}$. Observe that for the following legs, the level in which a matching neighboring cluster is found is strictly decreasing. Hence the sum of the lengths of the following legs decreases always by a factor of 2. The lemma follows immediately. ■

It is instructive to compare our routing scheme that relies entirely on the connectivity graph structure with schemes that exploit geographic coordinates. The latter effectively follow geodesics on the underlying manifold in which the nodes are embedded. As such, they are less affected by changes in the graph connectivity. But exactly because of that, they less aware of the geodesic structure intrinsic to the graph, and can be significantly suboptimal when the two geodesic structures differ (as in the presence of large holes in the network). A blend of the two may be an appropriate topic for future research.

3) 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 $C_N^{(k-1)}$ is a neighbor of v, then in level k its parent $p(C_N^{(k-1)})$ is also a neighbor. That means if a node has already stored the address of $p(C_N^{(k-1)})$ it can store the neighbor $C_N^{(k-1)}$ 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 2 *The routing scheme can be implemented by storing $O(\log n)$ bits per node in the network.*

Finally, we remark that the above addressing scheme as well as the neighboring information can be computed by restricted flooding (similar to the one in the DCH construction) during the initialization stage. It only increases the construction cost slightly over that of DCH's. We also emphasize that the discrete centers do not form any backbone structure during the routing process: once the addresses are assigned, they behaved just like any other node in the network. Therefore high-level discrete centers do not bear more traffic load than low-level ones during routing.

B. 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 only a small number of nodes, and the time required for a 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.

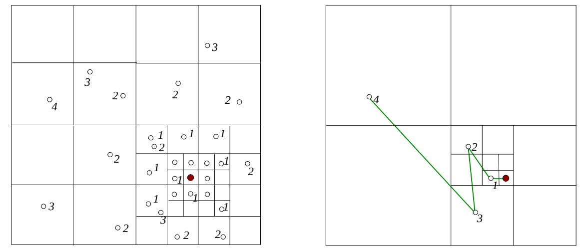
Our information brokerage system exploits the routing structure described above and reasonably meets these desiderata.

First assume that we have a hashing function $\mu : \Sigma \times HD \rightarrow S$, 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 only current routing structure (we will describe one such 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.

1) 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 $0 \le d < h$. The process is illustrated in Figure 3 (a). 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.

The length of the paths to the information servers decreases geometrically with decreasing level in the hierarchy. Therefore, the total number of hops to send synopses to all these information servers, i.e., the *communication cost* for the producer, is dominated by the length of the paths to the information severs in the highest level of the hierarchy. For graphs of low doubling dimension (number of neighbors $\lambda_{\max}$ is small), this cost is in the order of the diameter of the network. We summarize this in the following Lemma:

Lemma 3 *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.*



(a) (b)

Fig. 3: We use a quadtree as the underlying HD in this example. In (a), empty nodes are information servers at different levels (as indicated by the number, level 0 neighbors are not indexed) for the producer (the solid node). One can imagine that the producer will extend a path to each of these empty dots. In (b), the consumer (solid node) follows the thick path to query for information. It might stop at any time along this path, as soon as it locates the information.

2) Information retrieval: When a consumer w wants to access a particular 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. More precisely, it starts from w, and visits information servers $\mu(\sigma, L(w, 1))$, $\mu(\sigma, L(w, 2))$, $\ldots$, in sequential order to check whether the data item σ is there. See Figure 3 (b) for an illustration. 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 immediately when as soon as it finds the information sought.

The following lemma guarantees the distance sensitivity of our method.

Lemma 4 *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.*

Proof: Let d be the lowest level that $L(w, d)$ is a *neighbor* of u, where $L(v, i)$ is the ancestor cluster of sensor node v at level i. Since $L(w, d-1)$ is not a neighbor of u, $d_{uw} > \alpha \cdot 2^d$. It follows that during the information diffusion process, the producer u must have sent a message to node $\mu(\sigma, L(w, d))$ in cluster $L(w, d)$. Since node w inspects clusters $L(w, i)$ in increasing order of level i, it will visit at most d levels before it reaches the node $\mu(\sigma, L(w, d))$. In Section III-B.4, we will show that $\mu(\sigma, L)$ can be accessed from any point in cluster L at level i in at most $\alpha \cdot 2^{i+1}$ number of hops. Therefore, the overall number of time steps it takes to reach $\mu(\sigma, L(w, d))$ from w is

$$\sum_{i=0}^{d} \alpha \cdot 2^{i+1} \le \alpha \cdot 2^{d+2} \le 4 \times d_{uw},$$

■

We would like to emphasize that the bound obtained in practice is lower, very close to 1 in most cases.

a) Remark.: It is often the case 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.

3) 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.

Lemma 5 *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 $RC(w, r; \sigma) \le s \le RC(w, 4r; \sigma)$.*

Proof: First, for any producer u such that $d_{uw} \le r$, obviously $L(w, d)$ is a neighboring cluster for u at level d. The left inequality then follows.

To prove the right inequality, note that if u sends a message to $\mu(\sigma, L(w, d))$, then the distance between u and cluster $L(w, d)$ is at most $\alpha \cdot 2^d$. Since w is contained in $L(w, d)$ and the diameter of $L(w, d)$ is at most $\alpha \cdot 2^d$, it follows that $d_{uw} \le \alpha \cdot 2^{d+1} = 4r$. ■

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. Note that the guarantee of such approximate range counting comes from the distance sensitivity of our routing and information brokerage scheme.

4) Hash function $\mu(\sigma, L)$: 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 map 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 (roughly, $O(h \log n)$ bits per-node memory) on top of whatever routing structure exploited. Our goal is to define $\mu(\sigma, L)$ so that we can compute and route to it efficiently based purely on the routing framework we have.

In particular, if L is a leaf node, then $\mu(\sigma, L)$ is simply the sensor node contained in L. Otherwise, let the address of L be $(a_h, a_{h-1}, \ldots, a_{h-d}, 0, \ldots, 0)$, and c the number of children of L in HD. Given any hash function g that generate a random number based on σ and $(a_h, \ldots, a_{h-d}, 0, \ldots, 0)$, compute

$$a_{h-d-1} = g(\sigma, L) \mod c \ ,$$

and define $\mu(\sigma, L)$ recursively as $\mu(\sigma, L')$, where L' is the a_{h-d-1}'th child of L (i.e, the address for L' is $(a_h, \ldots, a_{h-d}, a_{h-d-1}, 0, \ldots, 0)$). There are various choices for choosing g. A simple one that we use is as follows (given σ is already mapped to some fixed integer ID_σ randomly).

$$g(\sigma, L) = \left(\mathrm{ID}_\sigma - \sum_{i=h-d}^{h} a_i \right) \mod p,$$

where p is a large prime number.

Note that this definition of $\mu(\sigma, L)$ also specifies a routing path to access it: one can first route to any node in L at level d, compute the child id a_{h-d-1}, route towards that child, and recurse. The number of hops to route from any node in L to $\mu(\sigma, L)$ is obviously bounded by $\sum_{i=1}^{d} \alpha \cdot 2^i \le \alpha \cdot 2^{d+1}$.

5) Robustness: The routing components of our information brokerage scheme inherit the robustness properties of the routing scheme; in particular, messages that 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.

IV. 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.

A. 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. For some of the test data, subsequently nodes were 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 4 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.

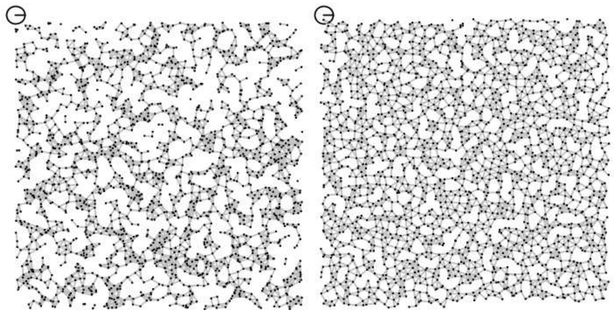


Fig. 4: Sensor fields with node density 0.02 nodes/m^2. Random sensor field (left) has degree 6.26 and has many small holes. Perturb grid field (right) has lower degree, 5.18, yet has more edges and looks much more regular.

In the following subsections, we fix the communication radius of the nodes to 10 m. The number of nodes in the network varies from 200 to 20000, and different (average) degrees of the communication graphs are obtained by adjusting the density of the network (from $.02$ to $.04$ nodes/m^2). We compare our routing scheme with GPSR, and our information brokerage scheme with GHT by assuming that the geographic locations of sensors are known. However, it is important to note that this geographic information is not used in our approach. The underlying planar graph for GPSR is the Gabriel graph.

We would like to point out that our routing scheme and information brokerage continue to work well even on networks with low node density, while the path quality of GPSR degrades quickly when the node density becomes low. When making the comparison between our approach and GPSR and GHT (which relies on GPSR), we restrict ourselves to reasonably dense networks in our experiments—a favorable case for GPSR.

B. Evaluation of Routing Strategies

a) Routing quality.: In a first experiment we evaluate the quality of the paths produced by our routing scheme. In particular, 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 I the average and standard deviation of the quality of these 1000 paths.

We note that our HD routing scheme always produces near-optimal paths regardless of the node density. The practical constant is much better than the bound of 4 we could prove.

b) Network initialization and routing scalability.: Here, we fix the node density at 0.02 nodes/m^2 (i.e., average degree is roughly 6), vary the number of nodes from 200 to 20000

Avg. deg.	Qual. of HD avg	Qual. of HD std	Qual. of GPSR avg	Qual. of GPSR 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

TABLE I: Comparison of the quality of paths from HD and from GPSR. HD routing gives close to optimal paths.

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 5. In particular, note that the maximal storage at a node in the network is quite reasonable, merely about twice the average per-node storage.

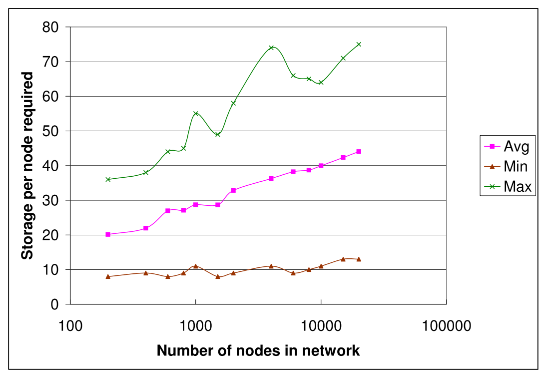
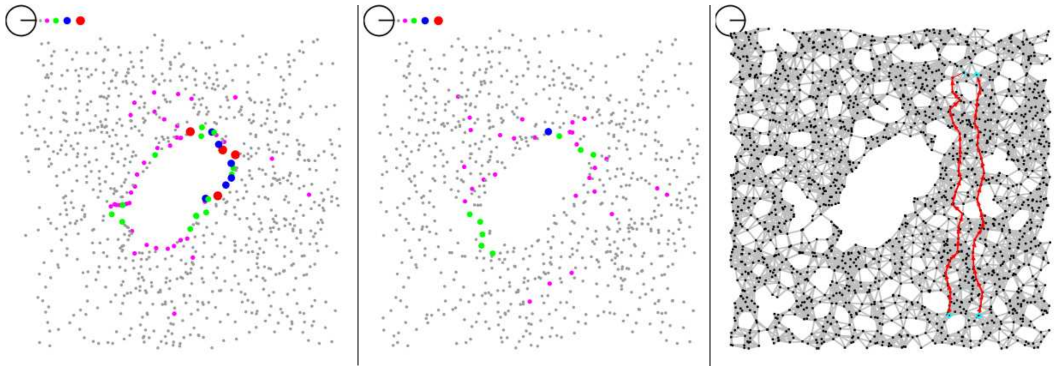


Fig. 5: The storage required growths slowly when the network becomes large.

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.

c) 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 toward a cluster reaches any of its nodes, the package starts heading toward a different cluster. So these cluster heads do not form a backbone structure, nor do they create bottlenecks in network traffic. Figure 6 (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 6 (a) and (b), where the size of the sensor field is 2000 with average degree 9.5. Our



(a) (b) (c)

Fig. 6: Hotspots comparisons for (a) GPSR and (b) HD scheme. Larger dots mean nodes with higher traffic loads. In (c), two paths generated by HD scheme with nearby sources and destinations.

HD scheme produces many fewer higher load nodes (larger dots in the picture).

d) 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 7. During the routing process, if a node finds that the next sensor on its shortest path to some cluster L is dead, it simply chooses another one at most 5 hops away from itself and with smaller distance to L. The result shows that the performance is gracefully degraded when the node failure rate increases.

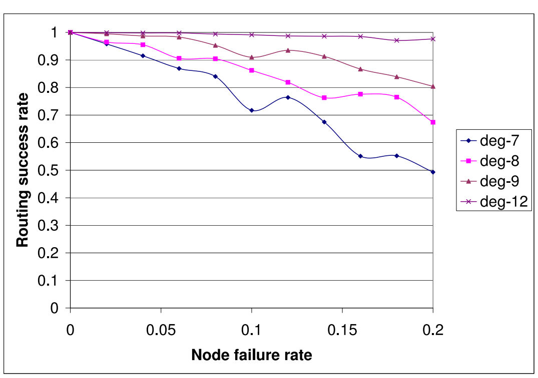


Fig. 7: The success rate of routing vs. node depletion for sensor fields with various average degree.

C. Evaluation of the Information Brokerage Scheme

In this section, we evaluate the performance of our information brokerage scheme built upon the HD routing structure.

e) 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 II, 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 4,5 show the path quality for the consumer, 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), showing that our information brokerage system is indeed distance sensitive.

Size	# Info. servers		Query time	
	avg	std	avg	std
200	9.95	0.21	1.08	0.66
400	15.9	0.26	0.89	0.67
600	25.4	1.02	0.77	0.57
1000	31.5	4.00	0.83	0.57
2000	47.0	7.47	0.73	0.47
4000	61.2	11.3	0.73	0.42
6000	69.5	13.6	0.79	0.46
8000	79.0	16.5	0.77	0.39
10000	84.3	15.6	0.76	0.39

TABLE II: Performance of brokerage

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 8, 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.

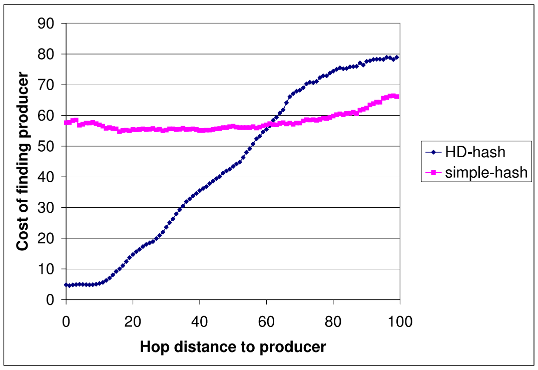


Fig. 8: Number of hops to information server using HD and number of hops in a shortest path to an ideally random information server.

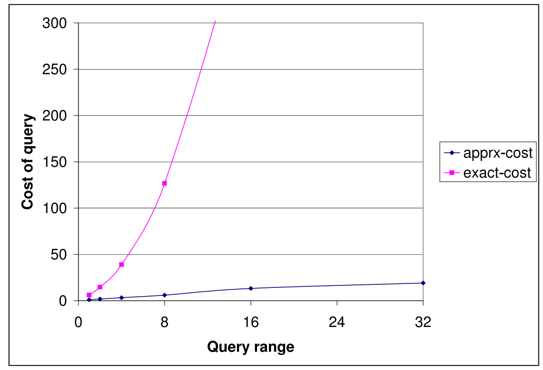


Fig. 9: Approximate query cost is very low compared to the naive flooding

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 result is shown in Table III. 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 III: The average/standard deviation of the number of times that a sensor serves as an information server for some data item.

f) 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 9 where the query cost in our approach (the 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.

g) 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 10. 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. The success rate can be further improved by checking neighbors when all fail though we have not incorporated that in our current system.

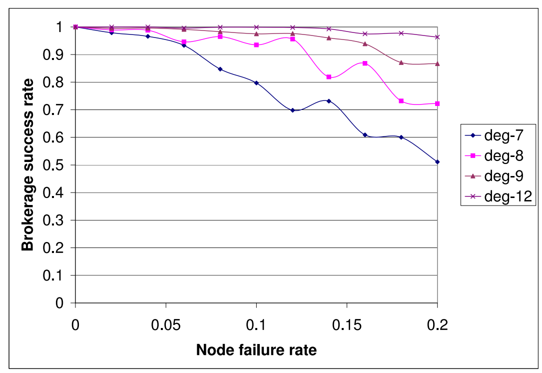


Fig. 10: The success rate of information brokerage under nodes depletions.

V. Conclusion and Discussion

In this paper we have presented an approach for augmenting hierarchical decompositions of communication graphs with sideways links, so as to allow for routing with guaranteed path quality. Our method is general and in particular works for hierarchical decompositions of communication graphs constructed in the absence of any geographical location information. The guarantee for almost optimal routing paths comes at an additional cost of only $O(\log n)$ bits of storage per network node. Unlike other hierarchical routing schemes, ours 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. The main contribution of this paper is a novel *distance-sensitive information brokerage* scheme based on the augmented hierarchical decomposition. Here we view the network nodes as *producers* of information (of data they sense or collect) that has to be made available for queries, i.e., for *consumers*. In this work we have emphasized the consumers' perspective and aimed for query times that are *distance-sensitive* in the sense that the time to answer a query should be proportional to the distance from the respective information producer (even though the location of the producer is not known to the consumer). To allow for such efficient queries, a data diffusion step has to be implemented by the information producers, in which a synopsis of the producer information is spread across the network and stored at $O(\log n)$ other locations. By defining an appropriate hash function, no node has to serve as information server for too many producers. 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 theoretical proofs of their performance guarantees, which is also reflected in the experimental results.

A more detailed network simulation is being planned. 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 [12]. 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 2nd author wishes to thank Scott Shenker for many useful discussions. The authors also gratefully acknowledge the support of NSF grants CCR-0204486 and CNS-0435111, as well as the DoD Multidisciplinary University Research Initiative (MURI) program administered by the Office of Naval Research under Grant N00014-00-1-0637.

REFERENCES

[1] Landmark-based information storage and retrieval in sensor networks. *Submitted to this conference.*

[2] E. M. Belding-Royer. Multi-level hierarchies for scalable ad hoc routing. *Wireless Networks*, 9(5):461–478, 2003.

[3] 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.

[4] S. M. Das, H. Pucha, and Y. C. Hu. Performance comparison of scalable location services for geographic ad hoc routing. In *24th Conference of the IEEE Communication Society (INFOCOM)*, 2005.

[5] 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.

[6] W. F. Fung, D. Sun, and J. Gehrke. COUGAR: The network is the database. In *SIGMOD Conference*, 2002.

[7] J. Gao, L. Guibas, and A. Nguyen. Distributed proximity maintenance in ad hoc mobile networks. In *IEEE International Conference on Distributed Computing in Sensor System (DCOSS)*, 2005.

[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] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] J. Kulik, W. Heinzelman, and H. Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *5th ACM MOBICOM*, Seattle, WA, August 1999.

[18] 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.

[19] 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.

[20] T. Moscibroda and R. Wattenhofer. Efficient computation of maximal independent sets in unstructured multi-hop radio networks. In *1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, 2004.

[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] P.-J. Wan, K. M. Alzoubi, and O. Frieder. Distributed construction of connected dominating set in wireless ad hoc networks. *Mob. Netw. Appl.*, 9(2):141–149, 2004.

[30] J. Widmer, M. Mauve, H. Hartenstein, and H. Füßler. Position-based routing in ad hoc wireless networks. pages 219–232, 2003.

[31] F. Zhao and L. Guibas. *Wireless Sensor Networks: An Information processing approach*. Elsevier/Morgan-Kaufmann, 2004.